CXFS™ Version 2 Software Installation and Administration Guide（日本語版）

007–4016–015JP

制作スタッフ

著作 Lori Johnson

編集 Rick Thompson、Susan Wilkening

イラスト Chrystie Danzer、Chris Wengelski

製作 Glen Traefald

協力エンジニア Rich Altmaier、Neil Bannister、François Barbou des Places、Ken Beck、Felix Blyakher、Laurie Costello、Mark Cruciani、Dave Ellis、Brian Gaffey、Philippe Gregoire、Dean Jansa、Erik Jacobson、Dennis Kender、Chris Kirby、Ted Kline、Dan Knappe、Kent Koeninger、Linda Lait、Bob LaPreze、Steve Lord、Aaron Mantel、Troy McCorkell、LaNet Merrill、Terry Merth、Nate Pearlstein、Bryce Petty、Alain Renaud、John Relph、Elaine Robinson、Dean Roehrich、Wesley Smith、Kerm Steffenhagen、Paddy Sreenivasan、Andy Tran、Rebecca Underwood、Connie Waring、Geoffrey Wehrman

COPYRIGHT

© 1999–2002 Silicon Graphics, Inc. All rights reserved. このマニュアルの別の箇所に示されているとおり、提供されている部分の著作権はサード・パーティが保持している場合があります。この電子ドキュメントの内容の一部または全部について、Silicon Graphics, Inc. から事前に文書による許諾を得ずに、いかなる方法でも複製または頒布したり、派生的な文書を作成することはできません。

LIMITED RIGHTS LEGEND

The electronic (software) version of this document was developed at private expense; if acquired under an agreement with the USA government or any contractor thereto, it is acquired as "commercial computer software" subject to the provisions of its applicable license agreement, as specified in (a) 48 CFR 12.212 of the FAR; or, if acquired for Department of Defense units, (b) 48 CFR 227-7202 of the DoD FAR Supplement; or sections succeeding thereto. Contractor/manufacturer is Silicon Graphics, Inc., 1600 Amphitheatre Pkwy 2E, Mountain View, CA 94043-1351, USA.

商標および帰属

Silicon Graphics、SGI、SGI ロゴ、IRIS、IRIX、O2、Octane、Onyx、Onyx2、および Origin は Silicon Graphics, Inc. の登録商標です。CXFS、FailSafe、IRISconsole、IRIS FailSafe、FDDIXPress、NUMAlink、Octane2、Performance Co-Pilot、SGI FailSafe、Trusted IRIX、および XFS は Silicon Graphics, Inc. の商標です。

Brocade は Brocade Communication Systems, Inc. の商標です。Digi は Digi International, Inc. の商標です。FLEXlm は GLOBEtrotter, Inc. の商標です。Java および Sun は Sun Microsystems, Inc. の登録商標です。Solaris は Sun Microsystems, Inc. の商標です。Linux は Linus Torvalds の登録商標です。Legato NetWorker は Legato Systems, Inc. の登録商標です。Netscape は Netscape Communications Corporation の商標です。UNIX は The Open Group の米国およびその他の国々における登録商標です。VERITAS は VERITAS Software Corporation の商標です。Windows および Windows NT は Microsoft Corporation の登録商標です。X/Open は X/Open Company Ltd. の登録商標です。

カバー・デザイン Sarah Bolles、Sarah Bolles Design、Dany Galgani、SGI Technical Publications

このガイドでの新機能

メモ:このリリースでは、再設定の機能はデフォルトで無効になっており、回復の機能はスタンバイ・ノードを使用している場合のみサポートされています。

スタンバイ・ノードは、特定のファイルシステムのメタデータ・サーバとして使用可能なノードとして設定されているが、そのファイルシステムを使用するアプリケーションを現在は実行していない、メタデータ・サーバとして使用可能な管理ノードです。回復の機能を使用するには、特定のファイルシステムのメタデータ・サーバとして使用可能なノードでアプリケーションを実行しないでください。アクティブなメタデータ・サーバがシステムによって選択された後、そのアクティブなメタデータ・サーバとクライアントのみのノードでそのファイルシステムを使用するアプリケーションを実行できます。

再設定および回復の機能は完全に実装されていますが、関連する問題が多数存在するため、現在のリリースでは完全にサポートされていません。これらの機能を使用してもデータの整合性が損なわれることはありませんが、クラスタ・ノードのパニックやハングが発生する可能性があります。再設定および回復の機能は、このような問題が解決された時点で、今後のリリースで完全にサポートされる予定です。

この更新版には、以下の新しい情報が含まれています。

- IRIX ノードは、CXFS クライアントのみのノードになることができるようになりました。つまり、IRIX ノードは、CXFS およびクラスタ・サービスの最小限の実装を実行し、CXFS クラスタ・データベースのコピーを含みません。クライアントのみのノードには、`cxfs_client` ソフトウェア製品がインストールされます。

  この変更により、CXFS 管理ノードという用語も導入されました。CXFS 管理ノードとは、`cluster_admin` ソフトウェア製品がインストールされるノードで、このノードでは、クラスタ管理タスクを実行したり、クラスタ・データベースのコピーを含むことができます。メタデータ・サーバとして実行するノードは、CXFS サーバとして使用可能な管理ノードとしてインストールする必要があります。その他のすべてのノードはクライアントのみのノードとしてインストールすることが推奨されます。

  ノードを定義する際、ノードの重みを指定する必要性はなくなりました。これは、「ノードの機能(Node Function)」フィールドで置換され、「サーバとして使用可能な管理(Server-capable Admin)」、「クライアント管理(Client Admin)」、または「クライアントのみ(Client-Only)」を選択することができます。Solaris ノードおよび Windows ノードに対しては、「クライアントのみ(Client-Only)」が自動的に選択されます。同様のフィールドは、`cmgr`(1M) コマンドを実行した場合にも表示されます。

  以下を参照してください。

  – 44 ページの「CXFS 管理ノードへの IRIX ソフトウェアのインストール」

- 109 ページの「GUI を使ったノードの定義」
- 148 ページの「cmgr を使ったノードの定義」
- 202 ページの「既存のノードのクライアントのみのノードへの切替え」

6.5.18f にアップグレードする場合、既存の IRIX ノードは、重みが 1 の場合、デフォルトで「サーバとして使用可能な管理(Server-capable Admin)」として割当てられます。

このバージョンでは、メンバーシップに対して使用される CXFS カーネル・メンバーシップおよびクラスタ・データベース・メンバーシップという概念も定義されます。8 ページの「用語」を参照してください。

- 新しいシステム調整可能パラメータ
  - cxfs_relocation_ok で、再設定の機能を有効または無効にすることができます。再設定の機能は、このリリースではデフォルトで無効になっており、有効にすることは推奨されません。
  - cxfsd_min および cxfsd_max で、各 CXFS ファイルシステムで実行する cxfsd スレッドの最小数と最大数を指定できます。

  222 ページの「システム調整可能パラメータ」を参照してください。
- 新しいコマンド
  - cxfs_info(1M) は、クラスタ、ノード、およびファイルシステムに関するステータス情報を提供し、クライアントのみのノードから実行されます。240 ページの「cxfs_info を使ったクラスタ、ノード、またはファイルシステムのステータスの確認」を参照してください。
  - cxfsdump(1M) は、CXFS 設定情報を収集します。256 ページの「cxfsdump を使用したクラスタ設定の収集」を参照してください。
- 1 つの CXFS クラスタでは、最大 32 個のノードがサポートされています。そのうち最大 16 個のノードは CXFS 管理ノードに、その他のすべてのノードはクライアントのみのノードになることができます。ノードは CXFS クライアント管理ノードとして定義するよう選択できますが、メタデータ・サーバとして使用可能なノードのみを CXFS サーバとして使用可能な管理ノードとして設定することが推奨されます。また、定足数の計算の目的では、サーバとして使用可能なノードの数を奇数にすることが推奨されます。
- XVM および CXFS のグラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) は、1 つに統合されました (89 ページの「GUI の概要」を参照してください)。このガイドでは、GUI によって提供される XVM 固有タスクの概要について説明します。これらのタスクについての詳細は、『XVM Volume Manager Administrator's Guide』を参照してください。

  以下のファイルシステム・タスクが GUI で提供され、このガイドで説明されています。
  - 132 ページの「ファイルシステムの作成」

  - 134 ページの「ファイルシステムの拡張」
  - 138 ページの「ファイルシステムをローカルでマウント」
  - 139 ページの「ローカル・ファイルシステムのアンマウント」
  - 139 ページの「ファイルシステム・マウント情報の削除」

- CXFS および Trusted IRIX の使用についてのアドバイス。235 ページの 第8章「Trusted IRIX と CXFS」を参照してください。

- クライアントのみのノードとしての Microsoft Windows 2000 システムのサポート。このガイドでは、Microsoft Windows NT ノードおよび Microsoft Windows 2000 ノードの両方を総称して *Windows* と記述します（情報が両方に同等に適用される場合）。片方のタイプのノードだけに適用される情報の場合は、適切なタイプのシステム名が明記されます。

改訂情報

バージョン	説明
001	1999 年 9 月 IRIX 6.5.6f リリースで CXFS 1.1 製品をサポート
002	1999 年 10 月 IRIX 6.5.6f リリースで CXFS 1.1 製品をサポート
003	1999 年 12 月 IRIX 6.5.7f リリースで CXFS 製品をサポート
004	2000 年 3 月 IRIX 6.5.8f リリースで CXFS 製品をサポート
005	2000 年 6 月 IRIX 6.5.9f リリースで CXFS 製品をサポート
006	2000 年 9 月 IRIX 6.5.10f リリースで CXFS 製品をサポート
007	2001 年 1 月 IRIX 6.5.11f リリースで CXFS 製品をサポート
008	2001 年 3 月 IRIX 6.5.12f リリースで CXFS 製品をサポート
009	2001 年 6 月 IRIX 6.5.13f リリースで CXFS 製品をサポート
011	2001 年 9 月 IRIX 6.5.14f リリースで CXFS 製品をサポート（社内の採番方法によりバージョン 010 はありません）
012	2001 年 12 月 IRIX 6.5.15f リリースで CXFS Version 2 製品をサポート
013	2002 年 3 月 IRIX 6.5.16f リリースで CXFS Version 2 製品をサポート

014	2002 年 6 月 IRIX 6.5.17f リリースで CXFS Version 2 製品をサポート
015	2002 年 9 月 IRIX 6.5.18f リリースで CXFS Version 2 製品をサポート

目次

このガイドについて xxvii

関連ドキュメント xxvii

出版物の入手方法 xxviii

表記規則 xxix

ご意見とお問合わせ先 xxix

1. CXFS とは 1

CXFS とは 1

XFS と CXFS の比較 2

サポートされている XFS の機能 3

CXFS を使用すべきケース 4

パフォーマンスに関する留意事項 4

ネットワーク・ファイルシステムと CXFS ファイルシステムの比較 5

ネットワーク・ファイルシステム 5

CXFS ファイルシステム 6

機能 6

制約 7

クラスタ環境 8

用語 8

クラスタ 8

ノード 9

プール 9

クラスタ・データベース 10

ノードの機能 10

メンバーシップ 16

プライベート・ネットワーク 16

再設定 17

回復 18

異常ノードの分離 20

I/O 制限 21

シリアル・ハードウェア・リセット 24
クラスタ・データベースと CXFS クライアント 27
メンバーシップ定足数 28
CXFS カーネル・メンバーシップ定足数 28
定足数の計算 28
定足数の変更の例 30
ネットワークの分割の例 30
クラスタ・データベース・メンバーシップ定足数 32
サーバとして使用可能なノードを 2 つだけ含むクラスタでの回復の問題 33
メタデータ・サーバの機能 34
システム表示 36
ハードウェアとソフトウェアのサポート 36
必要条件 36
互換性 38
推奨事項 39
IRIS FailSafe 同時実行の概要 39
クラスタ・マネージャ・ツールの概要 40

2. IRIX システム: CXFS ソフトウェアのインストールとシステムの準備 43

CXFS 管理ノードへの IRIX ソフトウェアのインストール 44
クライアントのみのノードへの IRIX ソフトウェアのインストール 49
IRIX システム・ファイルの設定 51
ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf 51
CXFS 管理ノードの /etc/services 54
CXFS 管理ノードの /etc/config/cad.options 54
CXFS 管理ノードの /etc/config/fs2d.options 55
例 1 57
例 2 58
(オプション) /.rhosts 58
(オプション) /etc/exports 59

(オプション) 自動再起動用の設定 60
ネットワーク・インタフェースの設定 60
IRIX ノードのシリアル・ポートの設定 62
システムの再起動 63
システムのテスト 63
プライベート・ネットワーク・インタフェース 64
CXFS 管理ノードのシリアル・ハードウェア・リセット接続 64
アップグレードのためのファイルシステム定義の切替え 66
6.5.12f 以前からのアップグレード 66
すべての IRIX ノードが 6.5.14f 以降にアップグレードされた状態での実行 66

3. クラスタの初期設定 67

初期設定の推奨事項 67
準備手順 70
ライセンスの確認 70
クラスタ・デーモンの開始 71
クラスタ・デーモンが実行されていることの確認 71
CXFS 管理ノードのホスト名の判断 72
`chkconfig` フラグが on であることの確認 72
GUI を使った設定 73
GUI の起動 73
GUI を使った新規クラスタの設定 74
GUI を使った新規 CXFS ファイルシステムの設定 76
`cmgr`(1M) コマンドを使った設定 77

4. CXFS の GUI タスク・リファレンス 89

GUI の概要 89
GUI の起動 90
GUI ウィンドウ 92
GUI の機能 94

GUI ウィンドウのレイアウト 95
「ファイル」メニュー 95
「編集」メニュー 95
「タスク」メニュー 96
「ヘルプ」メニュー 96
「ドメイン」メニュー 97
コマンド・ボタンを使ったショートカット 97
「表示」メニュー 99
表示または変更するアイテムの選択 99
コンポーネントの詳細の表示 99
タスクの実行 100
ドラッグアンドドロップを使った XVM 設定 100
PCP を使った I/O パフォーマンスの解析 101
ボリューム・トポロジーの構築 101
ディスクの設定 102
詳細の表示 102
GUI と xvm(1M) コマンドの重要な相違点 102
アイコンと状態の説明 103
設定ガイド・タスク 106
GUI を使って CXFS 用に既存の FailSafe クラスタを再設定 107
既存のクラスタの変更 108
クラスタ・ノードの修正またはアップグレード 108
GUI を使ったノード・タスク 108
GUI を使ったノードの定義 109
GUI を使ったノードの定義例 114
GUI を使ったクラスタ内のノードの追加と削除 117
GUI を使ったノードのリセット 117
GUI を使ったノード定義の変更 118
GUI を使った FailSafe ノードの CXFS 用への切替え 120

GUI を使ったノードの削除 121
GUI を使ったノードの接続性のテスト 121
GUI を使ったノードの表示 121
GUI を使ったクラスタ・タスク 122
GUI を使ったクラスタの定義 122
GUI を使ったクラスタ定義の変更 123
GUI を使った FailSafe クラスタの CXFS 用への切替え 123
GUI を使ったクラスタの削除 124
GUI を使ったクラスタの表示 124
GUI を使ったクラスタ・サービス・タスク 125
GUI を使った CXFS サービスの開始 125
GUI を使った CXFS サービスの停止（通常 CXFS シャットダウン） 125
GUI を使ったタイブレーカー・ノードの設定 126
GUI を使ったログの設定 126
GUI を使ったログ・グループ定義の表示 127
GUI を使ったログ・グループの設定 127
GUI を使ったローカル・ノードのメンバーシップ取消し 128
GUI を使ったローカル・ノードのメンバーシップ許可 128
GUI を使ったスイッチと I/O 制限タスク 129
GUI を使ったスイッチの定義 129
GUI を使ったスイッチ定義の変更 130
GUI を使ったスイッチ・ポート情報の更新 131
GUI を使ったスイッチ定義の削除 131
GUI を使ったノードの I/O 制限の強化 131
GUI を使ったノードの I/O 制限の緩和 131
GUI を使ったファイルシステム・タスク 132
ファイルシステムの作成 132
ファイルシステムの拡張 134
GUI を使った CXFS ファイルシステムの定義 135

GUI を使った CXFS ファイルシステムの変更 136
GUI を使った CXFS ファイルシステムのマウント 137
GUI を使った CXFS ファイルシステムのアンマウント 138
ファイルシステムをローカルでマウント 138
ローカル・ファイルシステムのアンマウント 139
GUI を使った CXFS ファイルシステムの削除 139
ファイルシステム・マウント情報の削除 139
GUI を使った CXFS ファイルシステムのメタデータ・サーバの再設定 140

5. CXFS の cmgr タスク・リファレンス 141

cmgr(1M) 概要 142
ヘルプの利用 142
プロンプト・モードの使用 142
アクションの完了とキャンセル 143
スクリプト・ファイルの使用 144
cmgr 内部からのシェルの呼出し 146
コマンド・ラインでのサブコマンドの入力 146
テンプレート・スクリプト 147
cmgr を使ったデフォルトの設定 147
cmgr を使ったノード・タスク 148
cmgr を使ったノードの定義 148
cmgr を使ったノードの変更 156
パーティションの設定例 157
異常階層の変更 158
cmgr を使ったノードのリセット 160
cmgr を使ったノードの CXFS または FailSafe への切替え 161
cmgr を使ったノードの削除 162
cmgr を使ったノードの表示 163
cmgr を使ったノードの接続性のテスト 165
cmgr を使ったシリアル接続のテスト 166

cmgr を使ったクラスタ・タスク 166
cmgr を使ったクラスタの定義 166
cmgr を使ったクラスタの変更 168
cmgr を使ったクラスタの CXFS または FailSafe への切替え 169
cmgr を使ったクラスタの削除 170
cmgr を使ったクラスタの表示 171
cmgr を使ったクラスタ・サービス・タスク 172
cmgr を使った CXFS サービスの開始 172
cmgr を使った CXFS サービスの停止 172
cmgr を使ったタイブレーカー・ノードの設定 173
cmgr を使ったログの設定 174
cmgr を使ったログ・グループ定義の表示 174
cmgr を使ったログ・グループの設定 175
cmgr を使ったログ・グループの変更 176
cmgr を使ったローカル・ノードのメンバーシップ取消し 176
cmgr を使ったローカル・ノードのメンバーシップ許可 177
cmgr を使った CXFS ファイルシステム・タスク 177
cmgr を使った CXFS ファイルシステムの定義 178
cmgr を使った CXFS ファイルシステムのマウント 183
cmgr を使った CXFS ファイルシステムのアンマウント 184
cmgr を使った CXFS ファイルシステムの変更 185
cmgr を使ったファイルシステムのメタデータ・サーバの再設定 189
cmgr を使った CXFS ファイルシステムの削除 189
cmgr を使ったスイッチと I/O 制限タスク 189
cmgr を使ったスイッチの定義 190
cmgr を使ったスイッチ定義の変更 190
cmgr を使ったノードの I/O 制限の強化 191
cmgr を使ったノードの I/O 制限の緩和 191
cmgr を使ったスイッチ・ポート情報の更新 192

cmgr を使ったスイッチ定義の削除 . . . 192
cmgr を使ったスイッチの表示 . . . 192
cmgr を使ったスイッチのステータスの照会 . . . 193
スクリプトの例 . . . 194
cmgr スクリプトの自動作成 . . . 196

6. 管理とメインテナンス . . . 201

既存のノードのクライアントのみのノードへの切替え . . . 202
NFS エクスポート・スクリプト . . . 203
lofs ファイルシステムのアンマウント . . . 204
telnet と I/O 制限の使用 . . . 205
fsr(1M) の使用 . . . 205
CXFS クラスタでの cron(1) の使用 . . . 205
階層ストレージ管理 (HSM: Hierarchical Storage Management) システムの使用 . . . 205
ファイルシステムのアクティブなメタデータ・サーバの検出 . . . 206
GUI を使用したメタデータ・サーバの検出 . . . 206
cluster_status を使用したメタデータ・サーバの検出 . . . 208
clconf_info を使用したメタデータ・サーバの検出 . . . 208
メタデータ・サーバの回復 . . . 209
データベースと CXFS のシャットダウン . . . 209
クラスタ・データベースのシャットダウン . . . 210
ノードのステータスとクラスタ・データベースのシャットダウン . . . 210
クラスタ・データベースの再起動 . . . 211
通常 CXFS シャットダウン . . . 211
ノードのステータスと通常 CXFS シャットダウン . . . 212
通常 CXFS シャットダウンを実行しない場合 . . . 212
通常 CXFS シャットダウン後のクラスタへの再設定 . . . 212
強制 CXFS シャットダウン: ローカル・ノードのメンバーシップ取消し . . . 213
ノードのステータスと強制 CXFS シャットダウン . . . 213
強制 CXFS シャットダウン後のクラスタへの再設定 . . . 214

シリアル・ハードウェア・リセット機能と強制 CXFS シャットダウン . . . 214
再起動時の CXFS の再開の回避 . . . 215
ログ・ファイル管理 . . . 215
すべてのログ・ファイルのローテーション . . . 215
大容量のログ・ファイルのローテーション . . . 216
ボリューム管理 . . . 216
ディスク管理 . . . 217
バックアップ . . . 217
NFS . . . 217
Quota . . . 218
Samba . . . 218
ファイルシステムのメインテナンス . . . 218
ファイルシステムのマウント . . . 218
ファイルシステムのアンマウント . . . 219
ファイルシステムの拡張 . . . 220
ダンプと復元 . . . 220
クラスタ・データベースのバックアップと復元 . . . 221
chkconfig フラグ . . . 222
システム調整可能パラメータ . . . 222

7. IRIS FailSafe との同時実行 . . . 227

CXFS と FailSafe を同時に実行する理由 . . . 227
同時実行できるリリース・レベル . . . 228
同時実行クラスタのサイズ . . . 228
クラスタのタイプ . . . 228
メタデータ・サーバ・ノードのタイプ . . . 230
GUI の違い . . . 230
切替え . . . 230
ネットワーク・インタフェース . . . 230
CXFS タイブレーカー・ノードと同時実行 . . . 231

メタデータ・サーバとフェイルオーバー・ドメイン 231
CXFS FailSafe のリソース・タイプ 231

8. Trusted IRIX と CXFS 235
CXFS と Trusted IRIX のインストールのヒント 235
強制アクセス・コントロール 236

9. ステータスのモニタ 237
ログ・ファイル 237
クラスタのステータス 238
GUI を使ったクラスタのステータスの確認 238
cluster_status を使ったクラスタのステータスの確認 239
clconf_info を使ったクラスタのステータスの確認 240
cmgr を使ったクラスタのステータスの確認 240
cxfs_info を使ったクラスタ、ノード、またはファイルシステムのステータスの確認 240
ノードのステータス 241
GUI を使ったノードのステータスのモニタ 242
cmgr を使ったクラスタのステータスの照会 244
cluster_status を使ったノードのステータスのモニタ 244
cmgr を使ったシステム・コントローラへの ping の実行 244
cmgr を使ったシリアル・ハードウェア・リセット回線のモニタ 244
XVM 統計 245
I/O 制限のステータス 246

10. トラブルシューティング 249
トラブルシューティング方法 249
ツールについての理解 249
物理ストレージ・ツール 250
クラスタ設定ツール 250
クラスタ制御ツール 251

ネットワーク・ツール 253
クラスタまたはノードのステータス・ツール 253
パフォーマンス・モニタ・ツール 254
カーネルのステータス・ツール 255
ログ・ファイル 256
cxfsdump を使用したクラスタ設定の収集 256
クラスタのステータスの識別 258
問題の検出 259
問題の回避 261
適切な開始 262
残っているクラスタの削除 262
クラスタ・データベース・メンバーシップ定足数の安定性 263
設定の統一性 263
ノードの機能の適切な定義 263
GUI の使用 263
ログ・ファイル名とサイズ 264
Netscape と Brocade スイッチの GUI 264
未書込みエクステントの追跡と排他書込みトークンに関するパフォーマンスの問題 264
未書込みエクステントの追跡 264
排他書込みトークン 265
crontab ファイルによる過剰なファイルシステム・アクティビティの回避 265
システム容量の効果的な使用 266
ノード ID またはクラスタ ID を変更する前の再起動 266
使用されていないノードの削除 267
強制シャットダウン後の CXFS の再開 267
シリアル・ハードウェア・リセット回線の取外し 267
xfs_repair の適切な使用 268
一般的な問題 270
ファイルシステムにアクセスできない 270

GUI が実行されない . . . 270
ログ・ファイルがディスク容量を消費しすぎる . . . 270
ノードを定義できない . . . 271
システムがハングする . . . 271
ノードは検出されるが、メンバーシップに設定されない . . . 271
セル ID カウントと“ Membership Delivered” メッセージ . . . 271
ログインできない . . . 272
ファイルシステムの I/O エラー . . . 272
クラスタの名前を変更後、ファイルシステムをマウントできない . . . 272
GUI に無効なファイルシステムが表示される . . . 273
エラー・メッセージの理解 . . . 273
通常メッセージ . . . 273
clconfd デーモンが異常終了する場合 . . . 275
論理スワップ・スペース不足 . . . 276
クラスタ名 ID 欠如エラー . . . 276
CXFS メンバーシップの取消し . . . 277
ライセンス・エラー . . . 277
IP アドレス・エラー . . . 278
SYSLOG エラー . . . 278
cli エラー . . . 279
clconfd エラー . . . 280
crsd エラー . . . 283
cmond エラー . . . 283
fs2d エラー . . . 284
一般メッセージ . . . 286
ログ・ファイル・エラー . . . 286
cad メッセージ . . . 287
cli メッセージ . . . 288
crsd エラー . . . 289

fs2d エラー 290
修正処置 291
CXFS サービスの再開 291
クラスタ・データベースのクリア 291
再起動 292
2 ノード・クラスタの回復 293
ノードをクラスタに再設定しない再起動 294
クラスタ・インフラストラクチャ・デーモンの停止と再開 295
クラスタ・データベースの再作成 295

付録A. CXFS ソフトウェアのアーキテクチャ 297
デーモン 297
通信パス 300
同時実行クラスタ内の通信パス 305
読取りおよび書込み時のメタデータの流れ 306

付録B. CXFS プライベート・ネットワークの IP フィルタリングの例 311

付録C. 初期設定チェックリスト 315

用語集 319

索引 331

図一覧

図1-1 プールとクラスタの概念 9
図1-2 均等に分散されているメタデータ・サーバ 13
図1-3 複数のメタデータ・サーバ 14
図1-4 1 つのメタデータ・サーバ 15
図1-5 再設定と回復 19
図1-6 I/O 制限 23
図1-7 シリアル・ハードウェア・リセットを使用したクラスタの例 25
図1-8 ネットワークの分割による定足数とアクティブなメタデータ・サーバの変化 31
図1-9 最小限の CXFS およびクラスタ・データベース・メンバーシップの概念 33
図3-1 CXFS マネージャ 74
図4-1 CXFS マネージャ GUI ウィンドウ 93
図4-2 ノードの詳細が表示されている CXFS マネージャ GUI 94
図4-3 ノード定義の例 115
図4-4 システム・コントローラ設定の例 116
図6-1 メタデータ・サーバが表示されているウィンドウ 207
図7-1 クラスタとノード・タイプの組合わせ 229
図9-1 ノードのステータス 243
図9-2 pmgxvm グラフ 246
図A-1 特定の管理ノード内の通信 300
図A-2 特定の管理ノード内のデーモン通信 301
図A-3 プール内のノード間の通信 302
図A-4 クラスタ内にない管理ノードの通信 303
図A-5 管理ノードとクライアントのみのノード間の通信 304
図A-6 同時実行時の特定の IRIX ノード内の管理通信 305
図A-7 同時実行時の特定の IRIX ノード内のデーモン通信 306

図A-8 書込み時のメタデータの流れ 307
図A-9 クライアントAでの書込み後のクライアントBでの読取り時のメタデータの流れ . . . 308
図A-10 クライアントAでの読取り後のクライアントBでの読取り時のメタデータの流れ . . . 309

表一覧

表2-1 fs2d.options ファイル・オプション 55
表4-1 GUI プラットフォーム 91
表4-2 コマンド・ボタン 97
表4-3 アイコンの説明 103
表4-4 状態の説明 106
表5-1 システム・コントローラ・タイプ 151
表6-1 システム調整可能パラメータ 223
表10-1 SYSLOG エラー・メッセージの形式 279
表10-2 ログ・エラー・メッセージの形式 287
表A-1 CXFS のデーモンとスレッド 298

このガイドについて

このガイドは、ストレージ・エリア・ネットワーク (SAN: Storage Area Network) で実行される CXFS について説明しています。ここでは、『IRIX Admin: Disks and Filesystems』で説明されている XFS ファイルシステムをすでに十分に理解しており、『XVM Volume Manager Administrator's Guide』を利用できることを想定しています。

CXFS クラスタのインストールと設定を行う前に、このマニュアル全体、特に 249 ページの 第10章「トラブルシューティング」をよくお読みください。

関連ドキュメント

以下のドキュメントには、補足情報が記載されています。

- 『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』
- 『EL Serial Port Server Installation Guide』(Digi International によって提供されています)
- 『EL Serial Port Server Installation Guide Errata』
- 『FDDIXPress Administration Guide』
- 『IRISconsole Administrator's Guide』(CXFS では Silicon Graphics O2 ワークステーションは CXFS ノードとしてサポートされていないため、CXFS リセット・サーバとして使用できません)
- 『IRIS FailSafe Version 2 Administrator's Guide』
- 『IRIX 6.5 Installation Instructions』
- 『IRIX Admin: Disks and Filesystems』
- 『IRIX Admin: Networking and Mail』
- 『Personal System Administration Guide』
- 『NIS Administrator's Guide』
- 『Performance Co-Pilot User's and Administrator's Guide』
- 『Performance Co-Pilot Programmer's Guide』
- 『Personal System Administration Guide』

- 『SGI TP9400 RAID Owner's Guide』
- 『SGI TP9400/9500 RAID Administration Guide』
- 『SGI Total Performance 9100 Storage System Owner's Guide』
- 『Trusted IRIX Read Me First Notice』
- 『Trusted IRIX/CMW Security Features User's Guide』
- 『XVM Volume Manager Administrator's Guide』

CXFS では以下のマン・ページが提供されています。

- `cdbBackup`(1M)
- `cdbRestore`(1M)
- `cmond`(1M)
- `cmgr`(1M)
- `cluster_status`(1M)
- `crsd`(1M)

出版物の入手方法

SGI のドキュメントは、以下の方法で入手できます。

- SGI Technical Publications Library `http://docs.sgi.com` を参照します。さまざまな形式が用意されています。このライブラリには、オンライン・ブック、リリース・ノート、マン・ページ、およびその他の情報の最新セットが幅広く含まれています。
- InfoSearch が SGI システムにインストールされている場合は、InfoSearch を使用できます。InfoSearch は、オンライン・ブック、リリース・ノート、およびマン・ページの限定されたセットを提供するオンライン・ツールです。IRIX システムで、Toolchest から「ヘルプ(Help)」->「InfoSearch」を選択するか、またはコマンド・ラインに「`infosearch`」と入力できます。
- リリース・ノートは、コマンド・ラインに「`grelnotes`」または「`relnotes`」と入力しても参照できます。
- マン・ページは、コマンド・ラインに「`man` *title*」と入力しても参照できます。

表記規則

メモ:このガイドでは、Microsoft Windows NT ノードおよび Microsoft Windows 2000 ノードの両方を総称して *Windows* と記述します（情報が両方に同等に適用される場合）。片方のタイプのノードだけに適用される情報の場合は、適切なタイプのシステム名が明記されます。

このドキュメントでは、以下の表記規則が使用されています。

表記規則	意味
`command`	この固定スペース・フォントは、コマンド、ファイル、ルーチン、パス名、シグナル、メッセージ、プログラミング言語の構造などのリテラル項目を示します。
variable	イタリック体は、変数のエントリ、および定義される語や概念を示します。
`user input`	この太字体の固定スペース・フォントは、対話型セッションでユーザが入力するリテラル項目を示します。出力は、太字体ではない固定スペース・フォントで示されます。
[ ]	コマンドやディレクティブ行のオプション部分は、角括弧で囲みます。

このガイドでは、Microsoft Windows NT ノードおよび Microsoft Windows 2000 ノードの両方を総称して *Windows* と記述します（情報が両方に同等に適用される場合）。片方のタイプのノードだけに適用される情報の場合は、適切なタイプのシステム名が明記されます。

このガイドでは、*IRIS FailSafe* を *FailSafe* と表記することがあります。

ご意見とお問合わせ先

このマニュアルの技術的正確性、内容、または構成についてご意見等ございましたら、弊社までお問合わせください。コメントいただくマニュアルのタイトルとドキュメント番号も必ず一緒にお聞かせいただくようお願いいたします。オンラインの場合、ドキュメント番号はマニュアルの最初の部分に記載されています。印刷マニュアルの場合は、各ページの下部にドキュメント番号が記載されています。

ご連絡の際は、以下のいずれかの方法をご利用いただけます。

- 電子メールの場合は、以下のアドレスまでお送りください。

  techpubs@sgi.com

- 次の Technical Publications Library の Web ページからの場合は、「Feedback」オプションをクリックしてください。

http://docs.sgi.com

- お客様相談窓口までご連絡いただき、SGI の問題追跡システムへの入力をお申付けください。
- 郵送の場合は、次の住所までお送りください。
  Technical Publications
  SGI
  1600 Amphitheatre Parkway, M/S 535
  Mountain View, California 94043–1351, USA
- FAX の場合は、+1 650 932 0801 の「Technical Publications」宛にお送りください。

いただいたコメントには迅速確実に対応いたします。

CXFS とは

この章では、以下の内容について説明します。

- 「CXFS とは」
- 2 ページの「XFS と CXFS の比較」
- 5 ページの「ネットワーク・ファイルシステムと CXFS ファイルシステムの比較」
- 8 ページの「クラスタ環境」
- 36 ページの「ハードウェアとソフトウェアのサポート」
- 39 ページの「IRIS FailSafe 同時実行の概要」
- 40 ページの「クラスタ・マネージャ・ツールの概要」

メモ: CXFS クラスタのインストールと設定を行う前に、このマニュアル全体、特に 249 ページの 第10章「トラブルシューティング」をよくお読みください。

CXFS とは

CXFS はクラスタ化された XFS で、ハイパフォーマンス・コンピューティング環境向けのクラスタ・ファイルシステムです。

CXFS を使用することで、コンピュータのグループは、高いパフォーマンスを保ったまま、一貫性のある方法で複数のホストに渡る XFS ファイルシステム、およびストレージ・デバイスを共有できます。CXFS は、ファイバ・チャネルなどのストレージ・エリア・ネットワーク (SAN: Storage Area Network) ディスク上で動作します。SAN は、サーバとストレージ・デバイスで構成される非常に高速でスケーラブルなネットワークで、ストレージ・リソースの統合、データ・アクセスと可用性の向上、およびストレージ管理の一元化を実現できます。CXFS ファイルシステムは、CXFS 管理ソフトウェアによってクラスタ全体にマウントされます。このファイルシステムをマウントするすべてのノードで、ファイルシステム内のすべてのファイルを使用可能です。

CXFS と IRIS FailSafe は、同じインフラストラクチャを使用します。

XFS と CXFS の比較

CXFS は、XFS と同じファイルシステム構造を使用します。CXFS ファイルシステムは、標準的な XFS ファイルシステムの作成に使用するコマンドと同じ mkfs(1M) コマンドを使用して最初に作成します。

XFS ファイルシステムと CXFS ファイルシステムの最も大きな違いは、ファイルシステムのマウントと管理の方法です。

- XFS の場合
  - ファイルシステムは、mount(1M) コマンドで直接マウントするか、/etc/fstab のエントリにより起動時にシステムにマウントさせるか、またはファイルシステム・マネージャを利用してマウントします。
  - ファイルシステムは 1 つのホスト上に 1 つだけ存在します。
  - /etc/fstab ファイルには、ファイルシステムに関する静的な情報が含まれます。詳細については、fstab(4) のマン・ページを参照してください。
- CXFS の場合
  - ファイルシステムは、CXFS マネージャのグラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) または cmgr(1M) コマンドを使用してマウントします。
  - ファイルシステムは、クラスタ内のすべてのホスト(ノード)から使用可能です。CXFS ファイルシステムは、CXFS 管理ソフトウェアによってクラスタ全体にマウントされます。ファイルシステム内のすべてのファイルは、このファイルシステムをマウントするすべてのホストで参照できます。
  - メタデータ(ファイルの名前、サイズ、場所、アクセス権など、ファイルを記述する情報)の更新の調整は、クラスタ内のすべてのノードに代わって特定のノード(メタデータ・サーバと呼ばれる)が行います。各 CXFS ファイルシステムには 1 つのアクティブなメタデータ・サーバがあり、1 つのクラスタには複数のアクティブなメタデータ・サーバを含めることができます(各 CXFS ファイルシステムに 1 つずつ)。
  - ファイルシステムの情報は、ファイルシステム、ノード、およびクラスタに関する固定の静的な設定情報が含まれるクラスタ・データベース (CDB: Cluster Database) に格納されます。CXFS クラスタ・デーモンは、プール内のすべての CXFS 管理ノードへのクラスタ・データベースの同期された複数のコピーの配布を管理します。管理者は、CXFS マネージャ GUI または cmgr(1M) コマンドを使用して、データベースを表示および変更できます。

GUI には、クラスタの静的な状態と動的な状態が表示されます。たとえば、ファイルシステムのマウントが有効に設定されているという静的な情報がデータベースに含まれている場合、以下のいずれかを示す動的な情報が GUI に表示されます。

- ファイルシステムがマウントされていることを示すアイコン(静的な状態と動的な状態が一致)
- ファイルシステムをマウントする準備はできているが、CXFS サービスが開始されていないために手順を完了できないことを示すアイコン(静的な状態と動的な状態が一致しないが、現在の状況では予想された状態)
- ファイルシステムがマウントされているべき(CXFS サービスは開始済み)なのにマウントされていないことを示すエラー(赤い)アイコン(静的な状態と動的な状態が一致せず、問題がある)

以下のコマンドを使用して、クラスタの状態を表示することもできます。

- `cmgr`(1M)(CXFS 管理ノードで使用される): 静的なクラスタの状態を表示します。
- `clconf_info` および `cluster_status`(1M)(CXFS 管理ノードで使用される): 静的なクラスタの状態と動的なクラスタの状態の両方を表示します。
- `cxfs_info` コマンド: クライアントのみのノードで、ステータス情報を提供します。

– 情報は `/etc/fstab` ファイルには格納されません。ただし、CXFS ファイルシステムは、実際には `/etc/mtab` ファイルに現れます。CXFS では、情報は代わりにクラスタ・データベースに格納されます。

サポートされている XFS の機能

XFS の機能のうち CXFS にもあるものは以下のとおりです。

- 信頼性が高く高速(1 秒以下の精度)なログベースのファイルシステムの回復機能。
- 64 ビットのスケーラビリティによる 1 ファイルあたり最大 900 万テラバイト(9 エクサバイト)のファイル・サイズ上限。
- 高速性: 高帯域幅(1 秒あたり数メガバイト)、高トランザクション・レート(1 秒あたりの I/O)、および高速なメタデータ操作。
- メタデータ領域の動的な割当て。
- Quota。Quota は、クラスタ内のどこからでも通常の XFS ファイルシステムと同じように管理できます。
- Filesystem Reorganizer(断片化解消プログラム)。このプログラムは、特定のファイルシステムの CXFS メタデータ・サーバから実行する必要があります。`fsr_xfs`(1M) のマン・ページを参照してください。

- ファイル・アクセス権とアクセス・コントロール・リスト (ACL: Access Control List) を使用したファイルへのアクセスの制限。また、論理ユニット (lun: logical unit) のマスク、または物理的なケーブル配線を使用して、特定のホストからSAN 内の特定のディスクのセットへのアクセスを拒否することもできます。

CXFS は、これらの基本的な XFS の機能を保ったまま、I/O をディスクとホストの間で直接分散します。効率的な XFS I/O パスでは非同期バッファ技術が採用されており、できるかぎり書込みを遅らせて不必要な物理 I/O を避けます。これにより、ファイルシステムは、データ領域を効率的、かつ多くの場合連続的に割当てることができます。一般的には、データは大容量の連続チャンクに割当てられるので、持続的に高い帯域幅が得られます。

XFS のディレクトリ構造は B ツリーに基づいているので、ディレクトリ内のファイルの数が数万～数十万に増加しても高速な応答時間を維持できます。

XFS の機能についての詳細は、『IRIX Admin: Disks and Filesystems』を参照してください。

CXFS を使用すべきケース

CXFS は、一般的なファイルシステムへの高帯域幅アクセスが必要なアプリケーションを実行する複数のホストが存在する場合に使用します。

CXFS は、以下の条件下で最高のパフォーマンスを発揮します。

- データ I/O 操作が 16 KB を超える場合
- 特定のファイルに対して読取りや書込みを実行するすべてのプロセスが同じホスト上に存在する場合
- 複数のホスト上の複数のプロセスが同じファイルを読取る場合
- 複数のホスト上の複数のプロセスの読取りと書込みにダイレクトアクセス I/O が使用される場合
- 大容量のファイルや大規模なファイル・アクセスを使用している場合

ほとんどのファイルシステム負荷では、上記のシナリオは大容量のファイル・アクセスを意味します。そのため、CXFS は高速なローカル・ファイル・パフォーマンスを実現できます。また、CXFS は、メタデータ I/O の量よりもデータ I/O の量が多い場合にも便利です。CXFS ではデータがネットワークを経由しないため、NFS よりも高速です。

パフォーマンスに関する留意事項

以下の状況では、CXFS が最適なパフォーマンスを発揮できない可能性があります。このような場合は、CXFS の使用を十分に検討する必要があります。

- ローカル・ホスト上のファイルにのみアクセスする場合。

- NFS を介して CXFS ファイルシステムをエクスポートする場合。CXFS メタデータ・サーバからエクスポートを実行した方が、CXFS クライアントからエクスポートしたときよりも大幅にパフォーマンスが向上する点に注意してください。
- 以下の場合など、CXFS でのアクセスがネットワーク・ファイルシステムと同程度に低速になる場合。
  - ファイルの容量が小さい
  - 帯域幅が低い
  - 大量のメタデータが転送される

  CXFS を経由したメタデータ操作は、ローカル・ファイルシステム上での操作よりも完了に時間がかかることがあります。以下に、メタデータのトランザクションの例を示します。
  - ファイルの開閉
  - ファイル・サイズの変更（一般的にはファイル・サイズの増加）
  - ファイルの作成と削除
  - ディレクトリの検索

  また、バッファ I/O を使用して同じファイルへの読書きを行っている複数のホスト上の複数のプロセスも、CXFS の使用時の方がローカル・ファイルシステムの使用時より低速になる可能性があります。パフォーマンスにこのような差が生じるのは、分散されているファイル・バッファの間でコヒーレンシを維持するためです。バッファに格納されている共有ファイルに書込みを行うと、ほかのホストでバッファされている（そのファイルに関する）データが無効になってしまいます。
- 分散型アプリケーションが、メモリ・マップされた共有ファイルに書込む場合。

ネットワーク・ファイルシステムと CXFS ファイルシステムの比較

ネットワーク・ファイルシステムと CXFS ファイルシステムで実行される機能の多くは同じですが、ここでは、パフォーマンスと機能の面での重要な違いについて説明します。

ネットワーク・ファイルシステム

ローカル・エリア・ネットワーク (LAN: Local Area Network) 上でリモート・ファイルにアクセスすると、ローカル・ファイルにアクセスした場合よりも大幅に低速になる場合があります。トランザクション・レートと帯域幅を大きく低下させる遅れの原因は、ネットワーク・ハードウェアとソフトウェアです。LAN ベースのネットワーク・ファイルシステムのクライアント/サーバ・アーキテクチャでは、このような遅れを回避するのは困

難です。これらの遅れは、LAN の帯域幅とレーテンシ上の制限、およびデータ・サーバを経由する共有パスが原因で発生します。

ほとんどの既存の共有ファイルシステムの速度は、LAN の帯域幅の制限を受けます。これは、ローカル・ディスクまたは共有ディスクに接続した複数のディスク・チャネル全体で得られる帯域幅よりも数段遅い場合があります。帯域幅のレートは、何階層ものネットワーク・プロトコルとサーバ・ソフトウェアによっても制限されます。

複数のクライアントが、中央のファイルサーバを経由する必要があるデータを待っている場合は、共有ファイルサーバがパフォーマンス上のボトルネックになることがあります。たとえば、NFS および Samba サーバは、サーバに接続されているディスクからデータを読取って UDP/IP または TCP/IP パケットにコピーしてから、LAN を通じてクライアント・ホストにデータを送信します。このサーバに多くのクライアントが同時にアクセスすると、サーバの応答性が低下します。

CXFS ファイルシステム

CXFS はクラスタ XFS ファイルシステムで、ネットワーク・ファイルシステムと同様の論理的なファイル共有が可能ですが、パフォーマンスと機能の面で大きな利点があります。CXFS は SAN 上で動作しますが、クラスタ内の各ホストは、共有されているディスクのセットに対して直接の高速なデータ・チャネルを利用できます。

機能

CXFS には、以下のような独自の機能があります。

- ディスク・アクセスのためのピアツーディスク・モデル。共有ファイルは、クラスタ内のすべてのホストでローカル・ファイルとして扱われます。各ホストは、このディスクに対してローカル・ディスクと同等の速度で読取りと書込みを行うことができます。データは、ディスクから、I/O を要求したホストに直接渡され、データ・サーバや LAN を経由することはありません。データ・パスの点では、各ホストは SAN 上のピアであり、どのシステムでも、共有ディスクに高速に直接アクセスできるデータ・パスを利用できます。

  したがって、SAN にディスク・チャネルやストレージを追加することで、帯域幅を拡張できます。大規模なシステムでは、ギガバイトにまで帯域幅を拡張でき、さらに毎秒数 10 ギガバイトにも拡張可能です。これは、1～100 MB/秒の LAN 上を流れるネットワーク・ファイルシステムと比較するとはるかに優れています。

  このピアツーディスクのデータ・パスにより、大部分の LAN ベースの共有ファイルシステムに見られるファイル・サーバのデータ・パスのボトルネックも解消されます。

- 各ホストは、ローカルに接続されているディスクと同じように共有ディスクをバッファできます。これらの分散バッファのコヒーレンシは CXFS によって維持されるので、XFS ファイルシステムの先進的なバッファ技術をそのまま利用できます。

- ファイルシステムのフラットで単一のシステム表示。これは、ファイルシステムを共有しているすべてのホストで同一で、特定のホストに依存することはありません。パス名は通常の POSIX パス名 (`/u/username/directory`) です。

  メタデータ・サーバをあるノードから別のノードに移動したり、メタデータ・サーバの名前を変更したり、メタデータ・サーバの追加や交換を行っても、パスは変わりません。このため、管理者にとってもユーザにとってもストレージの管理が簡単になります。1 つのホスト上の複数のプロセスでも、複数のホストに分散されているプロセスでも、ファイルシステムは同じように表示され、各ホストのパフォーマンスも同等です。

  パス名にファイル・サーバの名前が含まれる一般的なネットワーク・ファイルシステムとはこの点が異なります。この違いは、指定されているサーバを通じてディスクにアクセスする LAN ファイルシステムに余分な階層があるのと比べて、ダイレクトツーディスク I/O を使用する SAN のアーキテクチャが単純であることによるものです。

- POSIX、System V、および BSD インタフェースを含む完全な UNIX ファイルシステム・インタフェース。ここには、強制および任意レコード・ロックなどのファイルシステム・セマンティックスが含まれており、特殊なレコード・ロック・ライブラリは必要ありません。

制約

CXFS には、以下のような制約があります。

- 一部のファイルシステム・セマンティックスは適切ではないため、共有ファイルシステムではサポートされません。たとえば、root ファイルシステムは共有ファイルシステムとして適切ではありません。Root ファイルシステムは特定のホストに属しており、各特定のホストの特性に合わせてシステム・ファイルが設定されています。
- 名前付きパイプを使用するすべてのプロセスが同じノードに存在する必要があります。
- 階層ストレージ管理 (HSM: Hierarchical Storage Management) アプリケーションはメタデータ・サーバ上で実行する必要があります。

CXFS でサポートされていない XFS の機能は以下のとおりです。

- リアルタイム・ファイルシステム
- I/O レート保証
- ファイルへのスワップ
- マウント・オプションのネスト。IRIX ノードでは、CXFS ファイルシステム上にマウント・ポイントをネストすることはできません。つまり、既存の CXFS ファイルシステム上に IRIX XFS または CXFS ファイルシステムをマウントすることはできません。

クラスタ環境

この節では、以下の内容について説明します。

- 8 ページの「用語」
- 20 ページの「異常ノードの分離」
- 27 ページの「クラスタ・データベースと CXFS クライアント」
- 28 ページの「メンバーシップ定足数」
- 34 ページの「メタデータ・サーバの機能」
- 36 ページの「システム表示」

CXFS デーモン、通信パス、およびメタデータの流れについての詳細は、297 ページの付録A「CXFS ソフトウェアのアーキテクチャ」を参照してください。

用語

この節では、CXFS の理解に必要な用語を定義します。319 ページの「用語集」も参照してください。

クラスタ

クラスタは、単一のコンピュータ・リソースとして連携動作するよう設定されたシステム（ノード）の集合です。クラスタは、単純な名前とクラスタ ID で識別されます。複数のオペレーティング・システムが実行されているクラスタは、マルチ OS クラスタと呼ばれます。

ノードの特定のプールから形成されるのは、1 つのクラスタだけです。

ディスクまたは論理ユニット (LUN: Logical Unit) は、クラスタの名前をディスク（または LUN）に記録することによってクラスタに割当てられます。したがって、複数のクラスタ内のマシンから（ファイバ・チャネル接続を通じて）任意のディスクにアクセスできるようにするには、このようなクラスタに一意な名前が必要です。クラスタのメンバーからメッセージが相互に送信される際、クラスタ ID によってそのクラスタが識別されます。したがって、2 つのクラスタで通信用に同じネットワークを共有するようにする場合は、このようなクラスタに一意なクラスタ ID が必要です。マルチ OS クラスタに対しては、クラスタがネットワークを共有する場合、名前と ID の両方が一意でなければなりません。

クラスタ名およびクラスタ ID の上記の制約と、クラスタが作成されたらクラスタ名とクラスタ ID を（そのクラスタを削除して再作成しないかぎり）変更できないという理由により、構成内の各クラスタに対しては一意の名前とクラスタ ID を選択することが推奨されます。

ノード

ノードはオペレーティング・システム (OS: Operating System) の 1 個のイメージで、通常は個々のコンピュータです。この意味でのノードという用語は、SGI Origin 3000 または SGI 2000 システムのノードとは異なる意味を持ちます。)

特定のノードは、1 つだけのプール、つまり 1 つだけのクラスタのメンバーになることができます。

プール

プールは、特定のクラスタを形成できるノードの集合です。特定のプールから設定できるのは 1 つのクラスタのみです。使用可能なすべてのノードをクラスタに含める必要はありません。ほかのプールを存在させることはできますが、各プールはお互いから分離した状態になり、ノードやクラスタ定義は共有されません。

プールは、まず、特定の CXFS 管理ノード (`cluster_admin` がインストールされるノード) に接続し、CXFS GUI または `cmgr`(1M) コマンドを使用してクラスタ・データベース内でそのノードを定義すると形成されます。その後、最初のノードに接続されたままの状態でその他のノードを定義することにより、それらのノードをプールまたはプール内にすでにあるその他の CXFS 管理ノードに追加できます。別のノードに接続して定義を行うと、別のプールが作成されてしまいます。

図1-1 に、プールとクラスタの概念を示します。

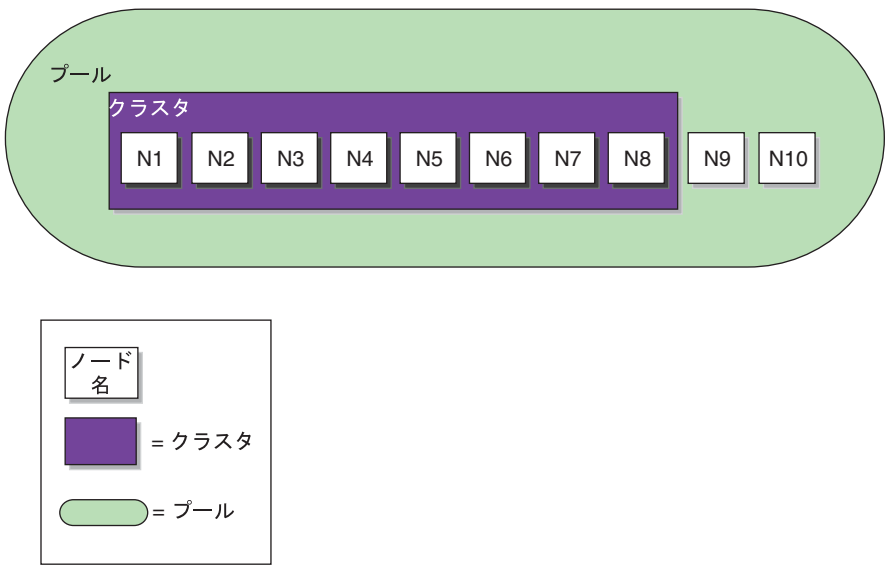


図1-1 プールとクラスタの概念

クラスタ・データベース

クラスタ・データベースには、ファイルシステム、ノード、およびクラスタに関する固定の設定情報が含まれます。クラスタ管理デーモンは、プール内のすべての CXFS 管理ノードへのクラスタ・データベース (CDB: Cluster Database) の配布を管理します。

このデータベースはファイルの集合で構成されており、CXFS マネージャ GUI と、cmgr(1M)、cluster_status(1M)、clconf_info、および cxfs_info(1M) コマンドを使用して、データベースの内容を表示および変更することができます。GUI は 1 つの CXFS 管理ノードに接続されており、cmgr、cluster_status、および clconf_info コマンドは 1 つの CXFS 管理ノード上で実行される必要があります。cxfs_info コマンドは、どのノードでも使用できます。

ノードの機能

ノードには、以下のいずれかの機能があります。

- CXFS クライアント管理ノード (IRIX のみ)

  これは、cluster_admin ソフトウェア製品がインストールされるノードで、CXFS クラスタ管理デーモンの完全なセット (fs2d、clconfd、crsd、cad、および cmond) が含まれます。これらのデーモンにより、このノードでは、クラスタ管理タスクを実行したり、クラスタ・データベースのコピーを含むことができます。デーモンについての詳細は、297 ページの付録A「CXFS ソフトウェアのアーキテクチャ」を参照してください。

  クラスタ管理タスクは、CXFS 管理ノードで実行されている cmgr(1) コマンドを使用するか、または CXFS マネージャ・グラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) を使用して CXFS 管理ノードに接続することによって実行します。詳細については、90 ページの「GUI の起動」を参照してください。

- メタデータ・サーバとして使用可能な管理ノード (IRIX のみ)

  このノードも cluster_admin のインストールによって作成されますが、管理タスクに加えて、クラスタのアクティビティとメタデータを調整することもできます。

  メタデータは、ファイルの名前、サイズ、場所、アクセス権など、ファイルを記述する情報です。通常、メタデータは、XFS 内にある 1 ファイルあたり 512 バイト程度の小さなファイルです。メタデータは、ファイルの内容であるデータとは異なります。データのサイズは、数メガバイトまたは数ギガバイトになる場合があります。

  各 CXFS ファイルシステムでは、1 つのノードがそのファイルシステムのメタデータの更新を行います。このノードをメタデータ・サーバと呼びます。サーバとして使用可能なノードとして定義されたノードだけが、メタデータ・サーバとなることができます。

  特定の CXFS ファイルシステムのメタデータ・サーバとして使用可能なノードとして複数の CXFS 管理ノードを定義することができますが、アクティブなメタデータ・サーバとして選択されるのは、各ファ

イルシステムに対して 1 つのノードのみです。クラスタ内には、複数のアクティブなメタデータ・サーバを存在させることができます（各 CXFS ファイルシステムに対して 1 つ）。

CXFS ファイルシステムをマウントするその他のノードは、CXFS クライアントと呼ばれます。CXFS 管理ノードは、その設定方法、およびアクティブなメタデータ・サーバになるよう選択されたかどうかに基づいて、メタデータ・サーバまたは CXFS クライアントのいずれかとして機能できます。

メモ: メタデータ・サーバおよび CXFS クライアントを、ネットワーク・ファイルシステムで使用される従来のデータ・パス型のクライアント/サーバ・モデルと混同しないように注意してください。プライベート Ethernet ネットワーク上でメタデータ・サーバを通じて送信されるのはメタデータ情報だけです。データは、ファイバ・チャネル接続を通じて CXFS クライアントのディスクに対して直接送受信されます。

- CXFS クライアントのみのノード（IRIX、Solaris、または Windows）

  このノードは、CXFS およびクラスタ・サービスの最小限の実装を実行するノードです。この最適化されたインフラストラクチャにより、クラスタとの十分な統合が提供され、クライアントのみのノードでは安全に CXFS ファイルシステムをマウントできます。

  Solaris ノードおよび Windows ノードは、常に CXFS クライアントのみのノードとして設定されます。IRIX ノードは、`cxfs_client` ソフトウェア・パッケージをインストールして作成した場合、クライアントのみのノードです。

  クライアントのみのノードを CXFS メタデータ・サーバにしたり、クライアントのみのノードでクラスタ管理を実行することはできません。

  Solaris クライアントのみのノードおよび Windows クライアントのみのノードについての詳細は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

- スタンバイ・ノード（IRIX のみ）

  このノードは、特定のファイルシステムのメタデータ・サーバとして使用可能なノードとして設定されているが、そのファイルシステムを使用するアプリケーションを現在は実行していない、サーバとして使用可能な管理ノードです。

すべての IRIX ノードで同じバージョンの IRIX オペレーティング・システムを実行することが理想的です。ただし、シームレスなアップグレードのためには、IRIX ノードで連続した 2 つのバージョンの IRIX OS のいずれかが実行されていることが最小限の必要条件となります。たとえば、6.5.16f と 6.5.17f は同じクラスタ内で同時に実行できますが、6.5.15f と 6.5.17f を同じクラスタ内で実行することはできません。連続した 2 つのバージョンを実行する機能は、6.5.12f リリースで最初に提供されました。

メモ: マルチ OS クラスタでは、すべての IRIX ノードで IRIX 6.5.16f 以上が実行されている必要があります。

各 CXFS マルチ OS リリースは、特定の IRIX リリースにのみ適用されます。IRIX ノードをアップグレードする際は、クラスタ内の Solaris ノードと Windows ノードも適切な CXFS マルチ OS リリースにアップグレードする必要があります。

詳細については、製品のリリース・ノートおよび『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

以下の図に、メタデータ・サーバとクライアントに対して使用できるさまざまな設定を示します。メタデータ・サーバとして使用可能なノードは CXFS 管理ノードになる必要があります。その他のノードはクライアントのみのノードになることができます。

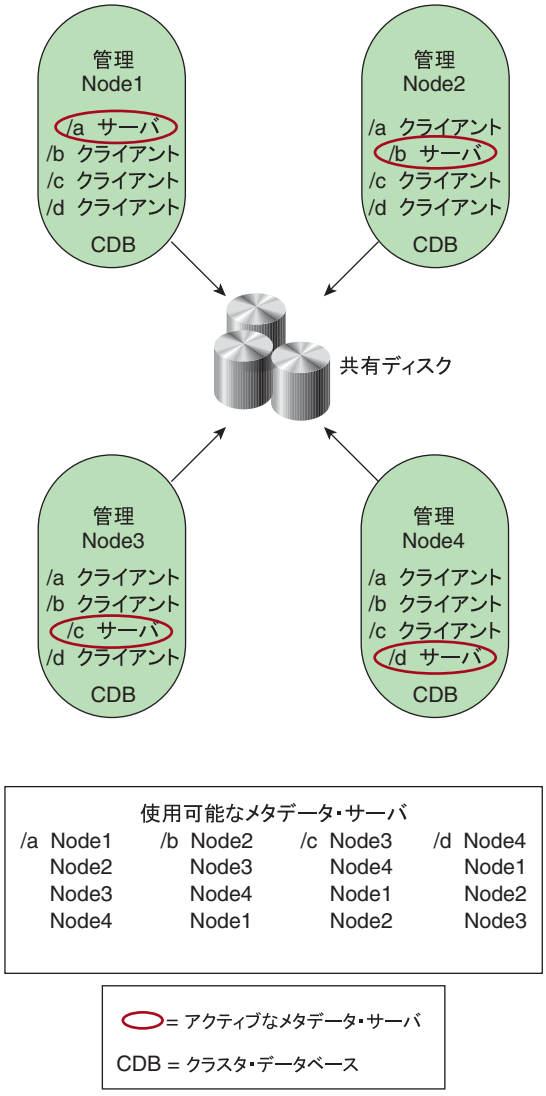


図1-2 均等に分散されているメタデータ・サーバ

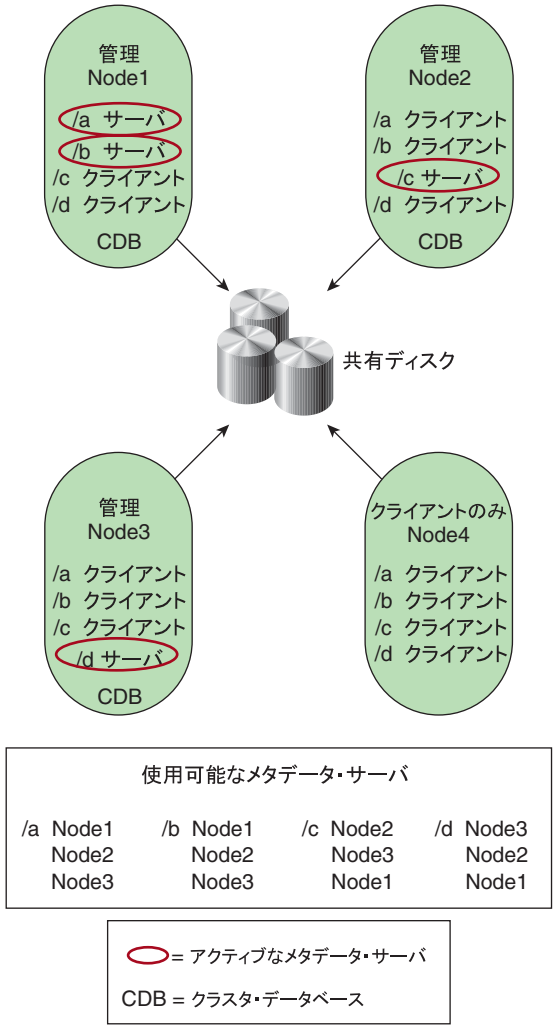


図1-3 複数のメタデータ・サーバ

図1-3 では、Node4 はクライアントのみのノードなので、サポートされている 任意の OS を実行することができます。このノードは、メタデータ・サーバとして使用可能なノードではありません。

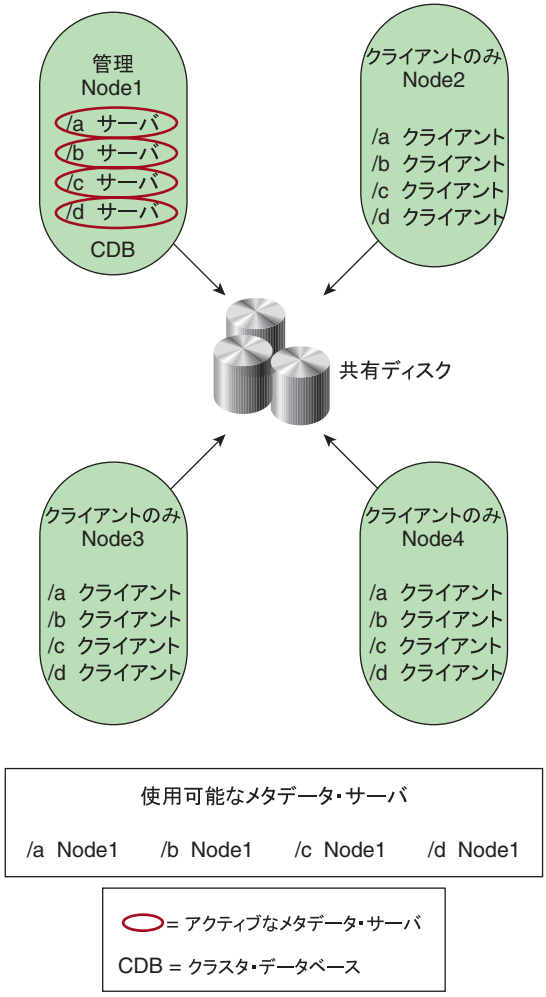


図1-4 1 つのメタデータ・サーバ

図1-4 では、Node2、Node3、および Node4 はクライアントのみのノードなので、サポートされている 任意の OS を実行することができます。これらのノードは、メタデータ・サーバとして使用可能なノードではありません。

メンバーシップ

メンバーシップには、以下のタイプがあります。

- CXFS カーネル・メンバーシップは、ファイルシステムをアクティブに共有できるクラスタ内の CXFS ノードのグループで、メンバーシップとハートビートを管理する CXFS カーネルによって決定されます。CXFS カーネル・メンバーシップは、クラスタ内で定義されているノードのサブセットの場合もあります。クラスタ内のすべてのノードを CXFS カーネル・メンバーシップに含めることができます。

  ノードは起動処理中に CXFS カーネル・メンバーシップを要求し、要求が受付けられると、クラスタのファイルシステムをアクティブに共有できます。メンバーシップは定足数とは異なります。28 ページの「CXFS カーネル・メンバーシップ定足数」を参照してください。

- クラスタ・データベース・メンバーシップ（`fs2d` メンバーシップまたはユーザ空間メンバーシップとも呼ばれる）は、相互にアクセス可能な CXFS 管理ノードのグループです。CXFS クライアントのみのノードは、クラスタ・データベース・メンバーシップに含めることはできません。

詳細については、28 ページの「メンバーシップ定足数」および297 ページの付録A「CXFS ソフトウェアのアーキテクチャ」を参照してください。

CXFS カーネル・メンバーシップおよびクラスタ・データベース・メンバーシップは、*FailSafe* メンバーシップとは異なります。FailSafe についての詳細は、『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。

プライベート・ネットワーク

プライベート・ネットワークは、クラスタ通信専用のネットワークで、管理者はアクセスできますが、ユーザはアクセスできません。

CXFS は、メタデータ・トラフィックのためにプライベート・ネットワークを使用します。クラスタ・ソフトウェアは、プライベート・ネットワークを使用して、クラスタ設定を機能させるために必要なハートビート・メッセージとコントロール・メッセージを送信します。ハートビートのタイミングのわずかな変動でも、問題が発生する可能性があります。ハートビート・メッセージの受信に遅れがあった場合、クラスタ・ソフトウェアは、ノードが応答していないと判断して CXFS カーネル・メンバーシップを取消します。このため、設定によってはノードがリセットまたは切断されることがあります。

ネットワーク装置を再起動すると、クラスタ内のノードの通信が切断される場合があり、その結果、CXFS カーネル・メンバーシップまたはクラスタ・データ・メンバーシップ、あるいはその両方が失われる可能性があります。ノード間の通信が切断されると、そのクラスタはパフォーマンス低下状態になったり、シャットダウンすることがあります。プライベート・ネットワークを使用するとネットワークのトラフィックが抑えられるため、不要なリセットや切断を回避するために役立ちます。また、パブリック・ネットワークのメッセージング・プロトコルではパケットの覗き見（不正な表示）やなりすまし（ネットワーク上の特定のマシンを別のマシンに見せかけること）を防止できないので、アクセスが制限されたネットワークの方が、ユーザ・アクセスのあるネットワークよりも安全です。

したがって、パブリック・ネットワークのパフォーマンスやセキュリティ特性が原因でクラスタに問題が発生する可能性があるという理由や、ハートビートがタイミングに大きく依存するという理由から、プライベート・ネットワークは必須です。

ハートビートおよびコントロール・ネットワークがすべてのノードに接続されており、すべてのノードがそのネットワークに対して同じサブネットを使用するよう設定されている必要があります。

注意: プライベート・ネットワークにネットワーク上の問題がある場合は、CXFS を使用する前に解決してください。

ネットワーク・セグメントおよびパーティション設定についての詳細は、30 ページの「ネットワークの分割の例」を参照してください。プライベート・ネットワークに対する IP フィルタリングの使用についての詳細は、311 ページの付録B「CXFS プライベート・ネットワークの IP フィルタリングの例」を参照してください。

再設定

メモ: このリリースでは、再設定の機能はデフォルトで無効になっており、回復の機能はスタンバイ・ノードを使用している場合のみサポートされています。

再設定および回復の機能は完全に実装されていますが、関連する問題が多数存在するため、現在のリリースでは完全にサポートされていません。これらの機能を使用してもデータの整合性が損なわれることはありませんが、クラスタ・ノードのパニックやハングが発生する可能性があります。再設定および回復の機能は、このような問題が解決された時点で、今後のリリースで完全にサポートされる予定です。

再設定とは、管理操作のために、あるノードから別のノードにメタデータ・サーバを移動するプロセスのことです。最初のノード上のその他のサービスが中断されることはありません。

CXFS カーネル・メンバーシップには、再設定による影響はありません。ただし、メタデータ・サーバの再設定中は、ファイルシステムのパフォーマンスが低下することがあります。

以下に、再設定が実行される場合の例を示します。

- システム管理者が、GUI または `cmgr` を使用してメタデータ・サーバを再設定した場合
- FailSafe `CXFS` リソースによってメタデータ・サーバが再設定された場合
- システム管理者が、メタデータ・サーバ上の CXFS ファイルシステムをアンマウントした場合
- メタデータ・サーバを再設定するコマンドがアプリケーションによって発行された場合

回復

メモ: 回復の機能は、このリリースではスタンバイ・ノードでのみサポートされています。

回復とは、あるノード上のサービスが中断した場合に、最初のノードから別のノードにメタデータ・サーバが移動されるプロセスのことです。

以下に、回復が実行される場合の例を示します。

- メタデータ・サーバのパニックが発生した場合
- メタデータ・サーバがロック・アップし、メタデータ・クライアントでハートビートのタイムアウトが発生した場合
- メタデータ・サーバがハートビート・ネットワークから切断された場合

図1-5 に、メタデータ・サーバの再設定と回復の違いを示します。各 CXFS ファイルシステムでアクティブなメタデータ・サーバは 1 つだけです。クラスタ内には、複数のアクティブなメタデータ・サーバを存在させることができます（各 CXFS ファイルシステムに対して 1 つ）。

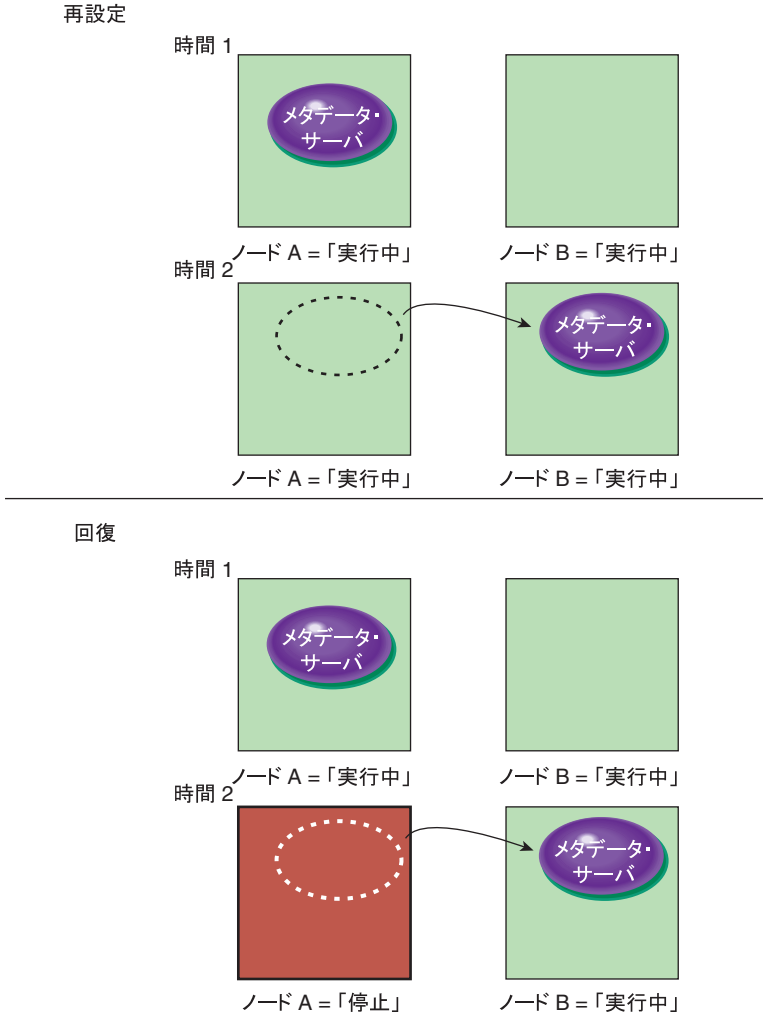

再設定
時間 1
メタデータ・サーバ
ノード A =「実行中」
ノード B =「実行中」
時間 2
メタデータ・サーバ
ノード A =「実行中」
ノード B =「実行中」
回復
時間 1
メタデータ・サーバ
ノード A =「実行中」
ノード B =「実行中」
時間 2
メタデータ・サーバ
ノード A =「停止」
ノード B =「実行中」

図1-5 再設定と回復

異常ノードの分離

CXFS では、以下の方法を使用して、異常ノードを分離します。

- I/O 制限: 問題のあるノードを SAN から分離します。これにより、問題のあるノードが I/O デバイスにアクセスできなくなるため、共有 CXFS ファイルシステムのデータが破壊されることがなくなります。I/O 制限は、クラスタ内の任意のノードに適用できます。制限が適用されると、クラスタの残りは、直ちに回復を開始することができます。ただし、I/O 制限では、応答しないノードをクラスタに返すことはできません。この問題を解決するには、システム管理者による操作が必要になります。

  SGI 以外のプラットフォームではリセット回線がサポートされていないため、Solaris および Windows ノードのデータを保護するには、I/O 制限が必要です。マルチ OS CXFS クラスタで I/O 制限をサポートするには、SGI が販売、サポートする Brocade ファイバ・チャネル・スイッチが必須です。

  メモ: I/O 制限は、ゾーニングとは異なります。

  制限とは、ホストと共有クラスタ・リソースとの間に障壁を組み立てることを意味する一般的なクラスタ用語です。

  ゾーニングとは、スイッチ（障壁）の論理サブネットを定義する機能のことで、ホストやメディアを特定のゾーンに含めたり、特定のゾーンからそれらを除外する機能も付いています。ホストは、そのゾーンに含まれているメディアにのみアクセスできます。ゾーニングは、制限を実装することのできる 1 つの方法です。

  ただし、ゾーニングの実装は複雑で、スイッチ全体での可用性は統一されません。このため、ホストのファイバ・チャネル・ポートを有効または無効にする、より単純な形式の制限の実装が、代替として選択されました。

- シリアル・ハードウェア・リセット: システム・コントローラに接続されているシリアル回線を通じてシステム・リセットを実行します。この方法は、IRIX ノードにのみ適用されます。

- I/O 制限とシリアル・ハードウェア・リセット: 問題のあるノードから SAN へのアクセスを無効にします。そのノードが正常に制限された場合、ノードが IRIX ノードであれば非同期リセットが実行され、リセット確認を待たずに回復が開始されます。

- CXFS シャットダウン: CXFS カーネル・メンバーシップが失われた場合に、ノードの CXFS カーネルベースのサービスを停止します。正常なクラスタは、ノードにシャットダウンする余裕を与えるために回復の開始を遅らせます。

IRIX ノードでは、CXFS が単にノードの機能の一部で、システムを再起動するよりも CXFS にアクセスできなくなった方が問題が少ない場合、データの整合性を保護するために I/O 制限を使用することができます。たとえば、CXFS クライアントでもある大規模なコンピュータ・サーバの場合などが例として挙げられます。CXFS が主要なアクティビティで、早急にオンラインに戻す必要がある場合（CXFS ファイルサーバの場合など）は、IRIX ノードで I/O 保護のためにシリアル・ハードウェア・リセットを使用することがで

きます。ただし、I/O 制限では、応答しないノードをクラスタに返すことはできません。この問題を解決するには、システム管理者による操作が必要になります。

Solaris および Windows ノードでは、データの整合性を保護するための唯一の選択肢は、I/O 制限です。

どの方法を使用するかを決定する命令のセットである異常時のアクション階層を定義することによって、これらの方法をどのように実装するかを指定することができます。109 ページの「GUI を使ったノードの定義」および 148 ページの「cmgr を使ったノードの定義」を参照してください。異常時のアクション階層を定義していない場合は、デフォルトで、シリアル・ハードウェア・リセットと CXFS シャットダウンが実行されます。

この節の以降の部分では、I/O 制限とシリアル・ハードウェア・リセットについてさらに詳細に説明します。CXFS シャットダウンについての詳細は、211 ページの「通常 CXFS シャットダウン」を参照してください。

I/O 制限

I/O 制限は、以下の機能を提供します。

- クラスタから除外されたノードが I/O を利用できなくすることにより、データの整合性を保持する
- 特定の状況において、除外されたノードのリセットを待つのではなく、直ちに処理を続行できる正常クラスタの回復を高速化する

ノードが CXFS カーネル・メンバーシップに設定されると、そのホスト・バス・アダプタ (HBA: Host Bus Adapter) のワールドワイド・ノード名 (WWNN: Worldwide Node Name) がクラスタ・データベースに格納されます。ノードに問題がある場合は、telnet プロトコルを使用して I/O 制限ソフトウェアから適切な Brocade ファイバ・チャネル・スイッチにメッセージが送信され、ポートが無効に設定されます。

注意: I/O 制限を正常に実行するには、常に telnet ポートを空けておく必要があります。

Brocade ファイバ・チャネル・スイッチにより、問題のあるノードが、対応する HBA を通じてストレージ・エリア・ネットワーク (SAN: Storage Area Network) のリソースと通信することが禁止されます。これについては、23 ページの図1-6 で説明します。

クラスタ化されていない LUN または SAN 内のデバイスにユーザがアクセスする必要がある場合、これらの LUN またはデバイスは、明示的に制限からマスクされている HBA を通じてアクセスまたはマウントする必要があります。IRIX ノードの場合に制限から HBA を除去する方法についての詳細は、129 ページの「GUI を使ったスイッチの定義」および 190 ページの「cmgr を使ったスイッチの定義」を参照してください。Solaris ノードおよび Windows ノードの場合については、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

回復を行うには、影響を受けるノードを CXFS カーネル・メンバーシップから取消し、制限された HBA を経由する I/O パスを使用しているすべてのファイルシステムをアンマウントした後、クラスタに再設定する必要があります。この処理は制限の回復と呼ばれ、自動的に開始されます。設定されている異常時の

アクション階層によっては、制限の回復が開始される前に、ノードがリセット(再起動)される場合があります。異常時のアクション階層の設定についての詳細は、148 ページの「cmgr を使ったノードの定義」および 109 ページの「GUI を使ったノードの定義」を参照してください。

制限されたノードを CXFS カーネル・メンバーシップに再設定するには、現在のクラスタのリーダがノードの制限を緩和して、その XVM ボリュームの再検査とファイルシステムの再マウントを可能にする必要があります。ノードを CXFS カーネル・メンバーシップに再設定できなかった場合、そのノードは制限されたままになることがあります。制限とは、影響を受けるノードではなく、Brocade ファイバ・チャネル・スイッチに適用される操作なので、この状態は、ノードが再起動されたかどうかには関係ありません。したがって、場合によっては、制限を手動で緩和しなければならないことがあります。詳細については、131 ページの「GUI を使ったノードの I/O 制限の緩和」および 191 ページの「cmgr を使ったノードの I/O 制限の緩和」を参照してください。

注意: I/O 制限を使用する場合、HBA の制限を強化すると、それ以降は、制限を緩和するまでその HBA を通じた SAN との I/O を実行できなくなることを理解しておいてください。これには以下が含まれます。

- カーネル・ドライバのキューに入っている I/O。I/O 上のユーザ・プロセスとアプリケーションがブロックされ、完了待ちになる可能性があります。これらのプロセスは、UNIX エラー・コード EIO を返します。
- SAN 内のクラスタ化されていない(ローカルの)論理ユニット (LUN: Logical Unit)、またはテープ・ストレージ・デバイスなどのファイバ・チャネル・デバイスに対し、影響を受ける HBA を通じて発行された I/O。

詳細については、129 ページの「GUI を使ったスイッチと I/O 制限タスク」および 189 ページの「cmgr を使ったスイッチと I/O 制限タスク」を参照してください。

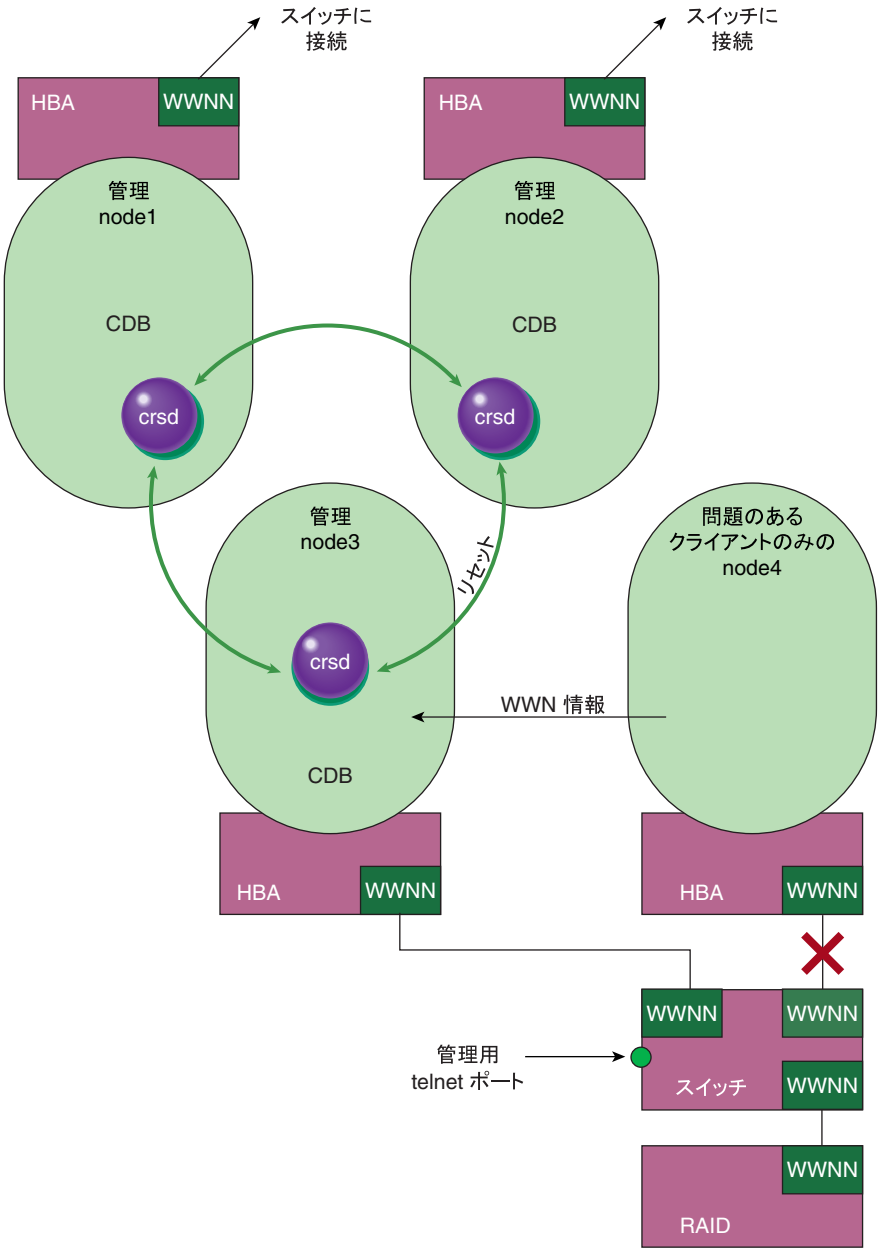

スイッチに
接続
スイッチに
接続
HBA
WWNN
HBA
WWNN
管理
node1
管理
node2
CDB
CDB
crsd
crsd
管理
node3
問題のある
クライアントのみの
node4
リセット
crsd
WWN 情報
CDB
HBA
WWNN
HBA
WWNN
WWNN
WWNN
管理用
telnet ポート
スイッチ
WWNN
WWNN
RAID

図1-6 I/O 制限

メモ: I/O 制限は、FailSafe ノードには使用できません。FailSafe ノードでは、シリアル・ハードウェア・リセットが必要です。

シリアル・ハードウェア・リセット

特定のエラー状況では、回復手順の一環として、クラスタから IRIX ノードを除外しなければならない場合があります。これは、システム・コントローラに接続されているシリアル回線を通じたシステム・リセットによって行わます。

図1-7 に、シリアル・ハードウェア・リセット機能と Ethernet シリアル・ポート・マルチプレクサを使用したクラスタにおける CXFS のハードウェア・コンポーネントの例を示します。

メモ: データの整合性を保証するため、サーバとして使用可能なノードが 2 つだけのクラスタでは、シリアル・ハードウェア・リセット機能、あるいは I/O 制限とスイッチの使用が必須です。また、サーバとして使用可能なすべてのノードに対しては、シリアル・ハードウェア・リセット機能、あるいは I/O 制限とスイッチの使用を強くお薦めします。より大きなクラスタでは、サーバとして使用可能なノードを奇数個使用し、ノードの 2 つのみがサーバとして使用可能な場合は、シリアル・ハードウェア・リセット回線を使用するか、I/O 制限とスイッチを使用する必要があります。33 ページの「サーバとして使用可能なノードを 2 つだけ含むクラスタでの回復の問題」を参照してください。

リセット接続の接続設定は、IRIS FailSafe と同じです。詳細については、日本 SGI の営業担当員、もしくは営業フリーダイヤルまたは管理サービスまでお問合わせください。

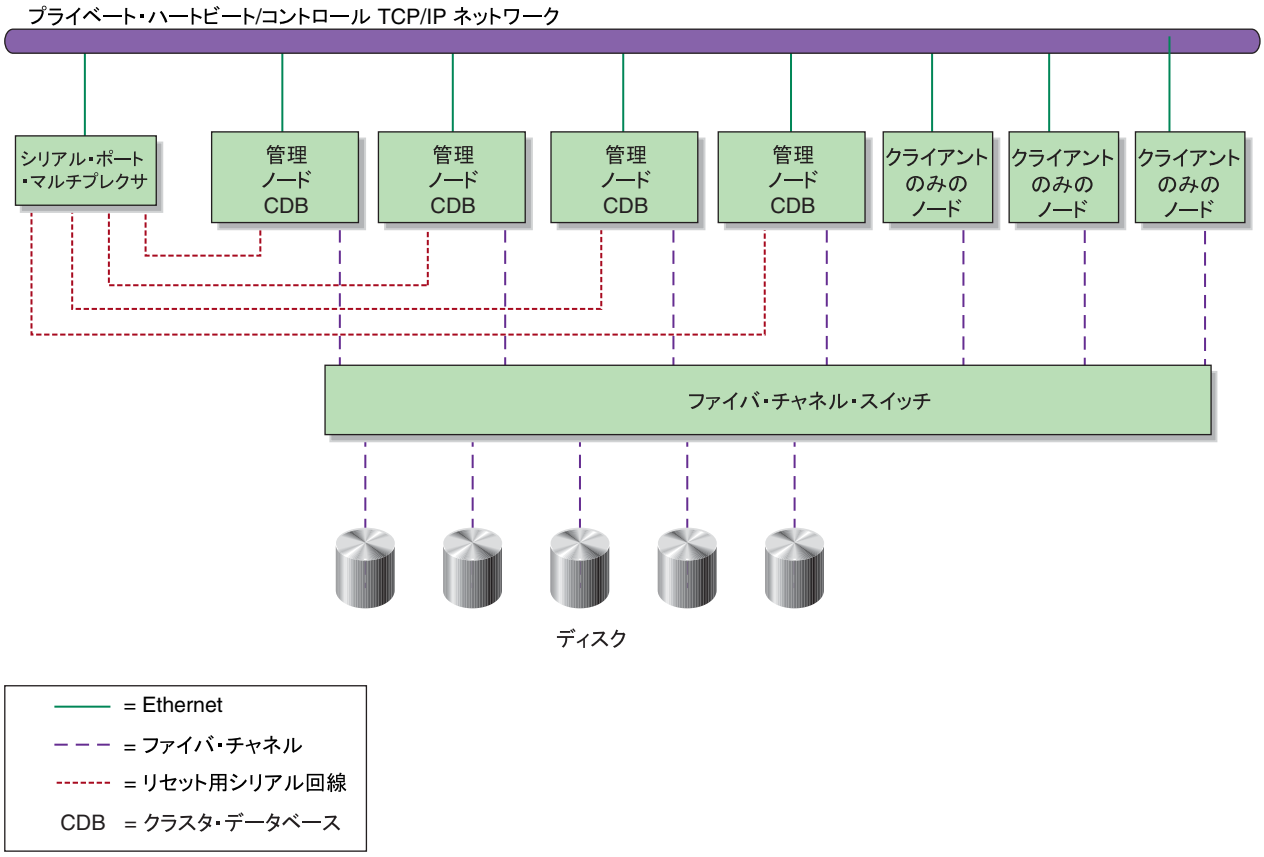


図1-7 シリアル・ハードウェア・リセットを使用したクラスタの例

クラスタとの接続が切断されたノードは、強制的に共有ディスクへのアクセスを中止します。シリアル・ハードウェア・リセット・ハードウェアがない場合、以下のことを仮定すると、これだけでは共有ディスクのデータの整合性を十分に保証できない可能性があります。

- ノードは通信が切断されたことを検出できます。つまり、このノードは、エラーのために切断を検出できない状態ではありません。
- ノードは、意図したとおりにタイムリーに切断を検出します。つまり、このノードのエラーによって検出が遅れることはありません。

ただし、めったに発生しない特定のエラー状況でデータの整合性を保証するため、特に、サーバとして使用可能なノード偶数個で構成されるクラスタの場合は、シリアル・ハードウェア・リセットに必要なハードウェアを使用することが推奨されています。マルチ OS クラスタを実行している場合は、I/O 制限を使用して、データの整合性を保護する必要もあります。

データの整合性を保証するため、サーバとして使用可能なノードが 2 つだけのクラスタでは、シリアル・ハードウェア・リセット機能、あるいは I/O 制限とスイッチの使用が必須です。また、サーバとして使用可能なすべてのノードに対しては、シリアル・ハードウェア・リセット機能、あるいは I/O 制限とスイッチの使用を強くお薦めします。より大きなクラスタでは、サーバとして使用可能なノードを奇数個使用し、ノードの 2 つのみがサーバとして使用可能な場合は、シリアル・ハードウェア・リセット回線を使用するか、I/O 制限とスイッチを使用する必要があります。IRIS FailSafe ではリセット機能は必須です。

最悪の状況は、ノードが通信の切断を検出しないにもかかわらず共有ディスクへのアクセスを引続き許可しており、データが破壊されてしまうような状況です。たとえば、ソフトウェアまたはハードウェアの異常が原因で、クラスタ内のあるノードがクラスタのその他のノードと通信できなくなっており、クラスタからアクティブなメンバーとして認識されていないにもかかわらず、引続き共有ディスクにアクセスできるような場合です。

この場合、シリアル・ハードウェア・リセットを行えば、エラーが検出された瞬間、かつこのノードがクラスタから除外された状態を回復する前に、その他のいずれかのノードが異常ノードによるディスクへのアクセスを中止し、異常ノードがこれ以上のアクティビティを行わないようにすることができます。

既存の CXFS カーネル・メンバーシップが 2 つの半数（それぞれが、サーバとして使用可能なノードの総数の半分を持つ）に分割されるような本当のネットワーク分割が発生した場合は、以下のような状況が発生します。

- CXFS タイブレーカー・ノード、およびシリアル・ハードウェア・リセットまたは制限が設定されている場合は、タイブレーカー・ノードのある方の半数がもう一方の半数をリセットまたは制限します。タイブレーカー・ノードのない方は、CXFS サービスを強制的にシャットダウンします。
- CXFS タイブレーカー・ノードがなくてもリセットまたは制限が設定されている場合は、各半数がヒューリスティック遅延を使用して他方の半数のリセットまたは制限を試みます。一方の半数はリセットに成功して続行され、他方はリセットまたは制限レースに負けて再起動または制限されます。

- CXFS タイブレーカー・ノードがなく、リセットまたは制限も設定されていない場合は、一方がレースに勝って他方をリセットすることを想定して、両方の半数がリセットを遅らせます。この場合は、どちらの半数もリセットまたは制限されないので両方が動作し続け、データが破壊される可能性があります。

  このような状況を避けるため、少なくともタイブレーカー・ノード、あるいはリセットまたは制限機能のいずれかを設定してください。ただし、サーバとして使用可能なノードが 2 つだけのクラスタ内のタイブレーカー・ノードに異常が発生したり、管理者が CXFS サービスを停止した場合は、その他のノードによって強制シャットダウンが実行され、CXFS ファイルシステムはすべてアンマウントされます。

リセットに失敗した方の半数が CXFS カーネル・メンバーシップを形成しようとしてネットワーク分割が解消されない場合は、サーバとして使用可能なノードの数が半分になって、初期 CXFS カーネル・メンバーシップを形成できなくなり、1 つのクラスタに 2 つの CXFS カーネル・メンバーシップができるのを防止します。

シリアル・ハードウェア・リセット接続には、以下の方法があります。

- 2 つのノードで構成されるクラスタは、シリアル・ハードウェア・リセット回線で直接接続できます。
- 3 つ以上のノードで構成されるクラスタは、シリアル・ポート・マルチプレクサで接続する必要があります。各 IRIX ノードは、ノードのリセット機能を持つノードであるオーナー・ホストを持つよう定義されます。

詳細については、日本 SGI の営業担当員、もしくは営業フリーダイヤルまたは管理サービスまでお問合わせください。

クラスタ・データベースと CXFS クライアント

分散型クラスタ・データベース (CDB: Cluster Database) は、CXFS クラスタの管理の中心です。データベースの同期された複数のコピーが、プール内の CXFS 管理ノード全体で保持されます (つまり、`cluster_admin` ソフトウェア・パッケージがインストールされます)。特定の CXFS マネージャ GUI タスクまたは `cmgr` タスクでは、別のタスクを開始する前に、CXFS クラスタ・デーモンによって関連する変更がクラスタ・データベースに適用され、各 CXFS 管理ノードに変更が配布されている必要があります。

プール内のクライアントのみのノードでは、完全なクラスタ・データベースの同期されたローカル・コピーは保持されません。代わりに、CXFS 管理ノードで実行されているデーモンの 1 つによって、関連するデータベース情報がそれらのノードに提供されます。CXFS 管理ノードの集合が変更されている場合は、別のノードによって、クライアントのみのノードの更新が行われる場合があります。

メンバーシップ定足数

メンバーシップの各タイプには別々の定足数のメカニズムがあります。この節では、以下の内容について説明します。

- 「CXFS カーネル・メンバーシップ定足数」
- 32 ページの「クラスタ・データベース・メンバーシップ定足数」
- 33 ページの「サーバとして使用可能なノードを 2 つだけ含むクラスタでの回復の問題」

CXFS カーネル・メンバーシップ定足数

設計上、アクティブなメタデータ・サーバは 1 つのファイルシステムにつき 1 つだけです。ただし、クライアントとメタデータ・サーバの間でメタデータ情報を転送するために使用されるハートビート・ネットワークまたはコントロール・ネットワークに問題があると、問題が発生する可能性があります。ネットワーク異常などの何らかの理由でハートビート・ネットワークまたはコントロール・ネットワークが半分に分割されると、ネットワークが 2 つの小さなネットワーク(セグメント)になってしまうことがあります。このような場合は、CXFS カーネル・メンバーシップ定足数によって、一方のメタデータ・サーバだけが SAN 上の CXFS ファイルシステムのメタデータの部分に書込みを行うようになります。

定足数の計算

CXFS カーネル・メンバーシップ定足数は、CXFS カーネル・メンバーシップに参加する、サーバとして使用可能なノードの数を、クラスタ内で定義されている、サーバとして使用可能なすべてのノードの総数と比較することによって計算されます。

メモ: クライアント管理ノードとクライアントのみのノードは、カーネル・メンバーシップ定足数を計算する際に考慮に入れられません。

最適なパフォーマンスを得るには、サーバとして使用可能な管理ノードを偶数個、CXFS クラスタに含めてください。

ほとんどの場合、メタデータ・サーバにならない IRIX ノードは、CXFS クライアント管理ノードではなく、クライアントのみのノードとしてインストールすることをお薦めします。

少なくとも 1 つのノードを、サーバとして使用可能な管理ノードとして定義しなければなりません。また、メタデータ・サーバとして使用可能なノードになることを希望するすべてのノードは、サーバとして使用可能な管理ノードとして定義する必要があります。

初期 CXFS カーネル・メンバーシップ定足数では、クラスタを形成するには、サーバとして使用可能なノードの過半数 (>50%) が使用可能でなければなりません。すべてのノードがサーバとして使用可能である大規模なクラスタの場合、クラスタを形成するには、大部分のノードがクライアント管理ノードまたは

クライアントのみのノードである場合よりもさらに多くのノードが使用可能でなければなりません。既存の CXFS カーネル・メンバーシップ定足数を維持するには、クラスタで定義されている、サーバとして使用可能なノードの半数 (50%) が必要です。

シリアル・ハードウェア・リセットまたは制限を使用しない場合は、CXFS タイブレーカー・ノードを設定して、ネットワーク分割が発生した際に複数のクラスタが生じるのを避けてください。

メモ: データの整合性を保証するため、サーバとして使用可能なノードが 2 つだけのクラスタでは、シリアル・ハードウェア・リセット機能の使用が必須です。また、サーバとして使用可能なすべてのノードに対しては、シリアル・ハードウェア・リセット機能の使用を強くお薦めします。より大きなクラスタでは、サーバとして使用可能なノードを奇数個使用し、ノードの 2 つのみがサーバとして使用可能な場合は、シリアル・ハードウェア・リセット回線を使用するか、I/O 制限とスイッチを使用する必要があります。

Solaris および Windows ノードでは、データの整合性の保護のために I/O 制限をサポートするには、Brocade スイッチが必須です。

CXFS タイブレーカー・ノードは、CXFS で識別されるノードで、クラスタ内のちょうど半数のサーバ・ノードが動作していて相互に通信できる場合に、クラスタの CXFS カーネル・メンバーシップを計算する処理で使用されます。デフォルトの CXFS タイブレーカー・ノードはありません。

ただし、サーバとして使用可能なノードが 2 つだけのクラスタ内の CXFS タイブレーカー・ノードに異常が発生したり、管理者が CXFS サービスを停止した場合は、その他のノードによって強制シャットダウンが実行され、CXFS ファイルシステムはすべてアンマウントされます。

ハートビートまたはコントロールに使用されているネットワークが半分に分割された場合は、サーバとして使用可能なノードの定足数を持っているネットワークの部分と、CXFS タイブレーカー・ノードだけがクラスタに残ります。定足数の一部ではないハートビート・ネットワークまたはコントロール・ネットワークの部分にあるノードは、クラスタから除外されます。したがって、ハートビート・ネットワークまたはコントロール・ネットワークが半分に分割された場合は、SAN を通じて同時に同じ CXFS メタデータにアクセスするハートビート/コントロール・ネットワークのどちらの部分にも、アクティブなメタデータ・サーバがなくなります。

CXFS カーネル・メンバーシップは、通常のメッセージとハートビートでそれ自体をモニタします。異常が発生した場合、違反ノードは CXFS カーネル・メンバーシップから削除されるかまたはリセットされ、以降はこれらのノードが共有リソースにアクセスすることはできなくなります。CXFS カーネル・メンバーシップから削除されたことを通信異常の発生によって検出したノードは、共有ディスクへのアクセスを強制的に終了し、最初にリセットされなかった場合は CXFS サービスを停止します。

CXFS カーネル・メンバーシップで可能なノードの数を変更して、新しいノードを組込めるようクラスタを拡大したり、CXFS カーネル・メンバーシップから除外されたノードを削除することができます。現在停止中で今後も使用できない、サーバとして使用可能なノードを削除すると、定足数の形成に必要な、サーバとして使用可能なノードの数が減ります。

定足数の変更の例

以下に、CXFS カーネル・メンバーシップ定足数を変更する場合の例を示します。

1 つのプライベート・ネットワーク上に、CXFS サーバとして使用可能な管理ノードが 6 つ（A、B、C、D、E、および F）のプールがあるとします。

- C は CXFS タイブレーカー・ノード
- A、B、C、D、および E は CXFS クラスタのメンバー

この条件の場合、初期 CXFS カーネル・メンバーシップ定足数に必要なノードの最低数は 3 つのノード (>50%) です。

B がシャットダウンしたり、CXFS カーネル・メンバーシップから外れると、クラスタ内の残りのノードは A、C、D、E の 4 つになります。この状態では、CXFS カーネル・メンバーシップ定足数を維持するための要件 (50%) がまだ満たされているので、このクラスタは引続き使用可能です。

ネットワークの分割の例

図1-8 に、ルータがダウンしてハートビート・ネットワークまたはコントロール・ネットワークが実質的に 2 つに分割された状態を示します。ネットワーク・セグメント 2 上のノード（ノード D および E）には CXFS タイブレーカー・ノードが含まれていないので定足数がありません。したがって、これらのノードは切断されます。ネットワーク・セグメント 1 では、その他の使用可能な 2 つのメタデータ・サーバになれるノードの一方がアクティブになり、クラスタにはネットワーク・セグメント 1 上のシステムだけが含まれることになります。ルータの修復後も、ネットワーク・セグメント 2 上にあったノードは、これらのノード上でクラスタ・サービスを再開するまでは切断されたままです。

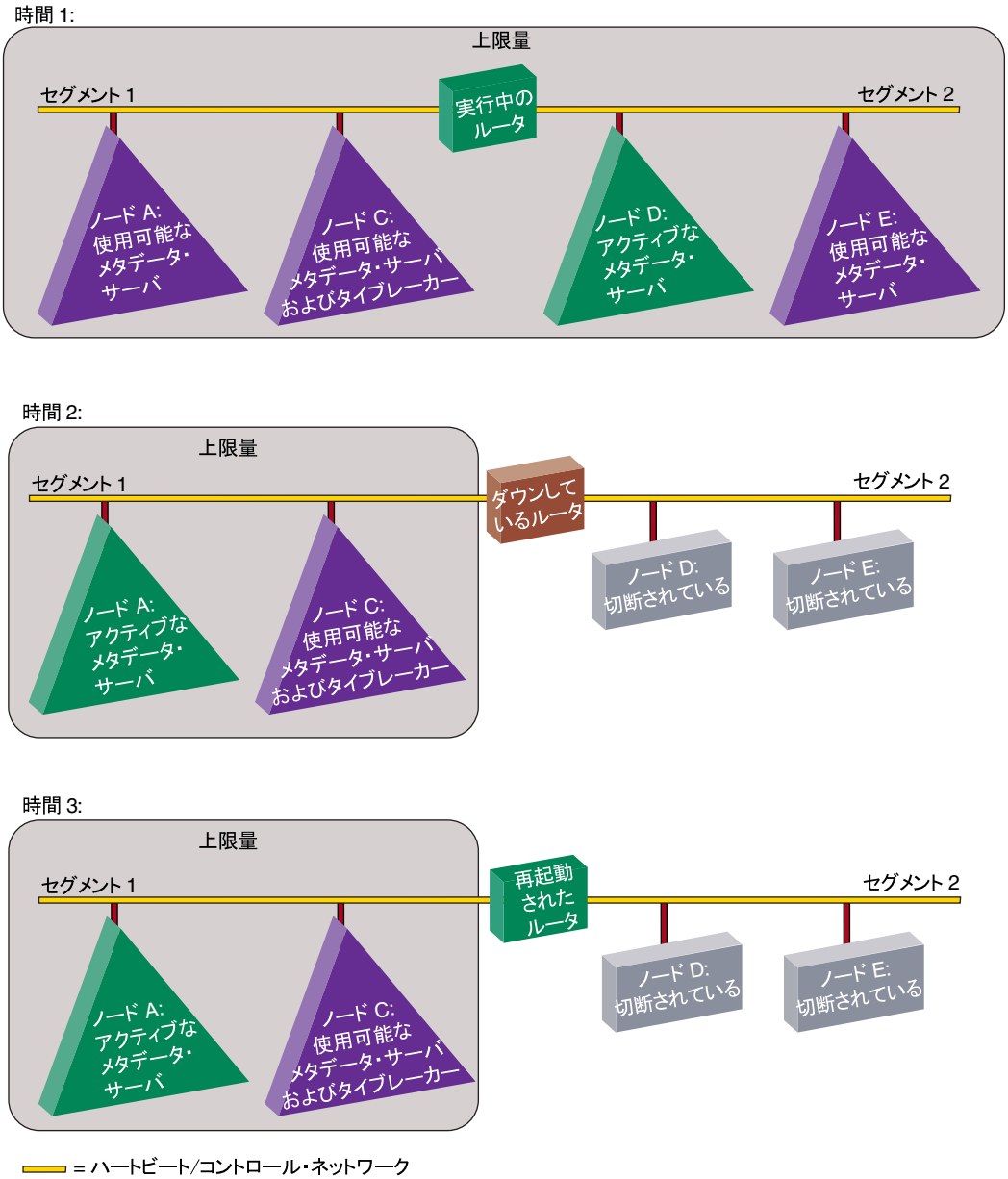


図1-8 ネットワークの分割による定足数とアクティブなメタデータ・サーバの変化

クラスタ・データベース・メンバーシップ定足数

CXFS クラスタ管理デーモンの完全なセットと、完全なクラスタ・データベースの配布されたコピーが含まれるのは、CXFS 管理ノードだけです。クライアントのみのノードには、最小限のデータベース情報のみ含まれます。

クラスタ・データベース・メンバーシップ定足数では、プール内の CXFS 管理ノードの半数 (50%) がクラスタ・データベースの更新を受信できる場合に初期クラスタを形成できます。

メモ: これは、CXFS カーネル・メンバーシップとは異なります。CXFS カーネル・メンバーシップの場合、初期クラスタを開始するには、サーバとして使用可能な CXFS 管理ノードの過半数 (>50%) がクラスタ内で使用可能である必要があります。CXFS カーネル・メンバーシップは、サーバとして使用可能なノードの 50% だけで維持できます。

図1-9 に、最小限の CXFS カーネル・メンバーシップおよびマルチ OS クラスタのクラスタ・データベース・メンバーシップの概念を示します。

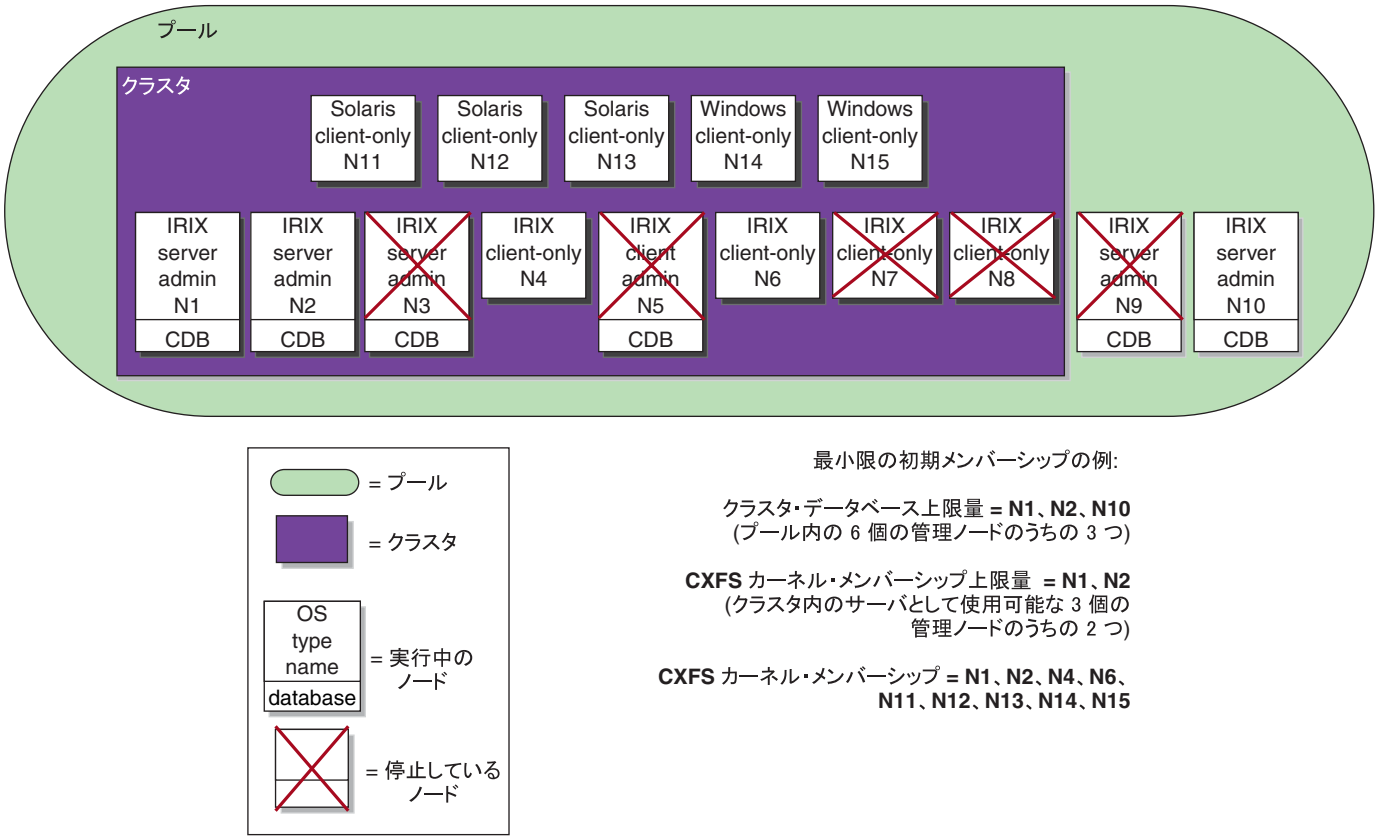


図1-9 最小限の CXFS およびクラスタ・データベース・メンバーシップの概念

サーバとして使用可能なノードを 2 つだけ含むクラスタでの回復の問題

実作業用の環境では、サーバとして使用可能なノードを少なくとも 3 つ含むクラスタが推奨されています。

ただし、サーバとして使用可能なノード偶数個で構成される(特に、サーバとして使用可能なノードを 2 つだけ含む場合)実作業用のクラスタを使用する場合は、以下のいずれかを行う必要があります。

- シリアル・ハードウェア・リセット回線を使用して、データを確実に保護し、1 つのノードだけがエラー状態で実行されるよう保証します。データの整合性を保証するため、サーバとして使用可能なノードが 2 つだけのクラスタでは、リセット機能の使用が必須です。また、サーバとして使用可能なすべてのノードに対しては、リセット機能の使用を強くお薦めします。より大きなクラスタでは、サーバとして使用可能なノードを奇数個使用し、ノードの 2 つのみがサーバとして使用可能な場合は、シリアル・ハードウェア・リセット回線を使用するか、I/O 制限とスイッチを使用する必要があります。

メモ: Solaris および Windows ノードでは、データの整合性の保護のために I/O 制限をサポートするには、SGI が販売、サポートする Brocade スイッチが必須です。

- CXFS タイブレーカー・ノードを設定します。この結果、タイブレーカー・ノードが停止すると、クラスタ・サービスとファイルシステムが失われます。

ただし、これらの方法を使用しても、サーバとして使用可能なノードが 2 つだけのクラスタには回復と再設定の点で特有の問題があります。

シリアル・ハードウェア・リセット回線を持つ、サーバとして使用可能なノードが 2 つだけのクラスタがあり、一方のノードに問題が発生したとします。この場合、以下のような状況が発生します。

- 両方のノードがサーバとして使用可能で、CXFS タイブレーカー・ノードがない場合

  CXFS クライアント・ノードはサーバのパニックまたはリセットを切抜けます。また、プライベート・ネットワーク上のメタデータ・サーバとの接続の切断を切抜ける場合もあります。接続が切断された場合は、クライアントとメタデータ・サーバの両方が他方のノードのリセットを試み、最初にリセットに成功したノードが残り、サーバになります。

- 両方のノードがサーバとして使用可能で、メタデータ・サーバが CXFS タイブレーカー・ノードの場合

  既存の CXFS カーネル・メンバーシップ定足数は、サーバとして使用可能なノードの 50% があれば維持できますが、このメタデータ・サーバはタイブレーカー・ノードなので、CXFS クライアント・ノードはメタデータ・サーバの異常を切抜けることができません。

- 両方のノードがサーバとして使用可能で、CXFS クライアント・ノードが CXFS タイブレーカー・ノードの場合

  サーバのパニックまたはリセット、およびプライベート・ネットワーク上のメタデータ・サーバとの接続の切断のどちらについても、CXFS クライアントは切抜けます。この CXFS クライアントは、メタデータ・サーバとして使用可能なノードとしてリストされている場合、アクティブなメタデータ・サーバになります。CXFS クライアントがリストにない場合は、ファイルシステムがアンマウントされます。

詳細については、20 ページの「異常ノードの分離」および 28 ページの「定足数の計算」を参照してください。

メタデータ・サーバの機能

メタデータ・サーバは、以下のようなクラスタ調整機能を実行します。

- メタデータのログへの記録
- ファイルのロック

- バッファのコヒーレンシ
- ファイルシステムのブロック割当て

メタデータに対する CXFS の要求は、すべて TCP/IP ネットワーク上でメタデータ・サーバを通じてルーティングされ、メタデータの変更はすべてメタデータ・サーバに送信されます。メタデータ・サーバは、XFS の先進的なジャーナル機能を使用して、メタデータの変更をログに記録します。一般的にメタデータのサイズは小さいので、メタデータのトラフィックに対しては Fast Ethernet ローカル・エリア・ネットワーク (LAN: Local Area Network) の帯域幅で十分です。

ディスクとの直接のデータ操作と比較すると、CXFS メタデータ・サーバへの操作の頻度は通常は高くありません。たとえば、ファイルを開くと、メタデータ・サーバにファイルの情報が要求されます。ファイルが開いたら、通常、プロセスは何度でも読取りや書込みを行うことができ、追加のメタデータ要求は必要ありません。メタデータ操作は、ファイル・サイズなどファイルのメタデータ属性が変更されたときに実行されます。

この場合、以下の規則が適用されます。

- `cluster_admin` 製品がインストールされたすべてのノードは、サーバとして使用可能な管理ノードとして定義できます。
- 複数のファイルシステムのアクティブなメタデータ・サーバになることができるのは、クラスタ内にある、サーバとして使用可能な 1 つのノードだけです。
- それぞれが異なるセットのファイルシステムを持つ、サーバとして使用可能なノード複数個が、アクティブなメタデータ・サーバになることができます。ただし、特定のファイルシステムは、1 つのノードの 1 つのアクティブなメタデータ・サーバを使用します。
- サーバとして使用可能な CXFS 管理ノード複数個を特定のファイルシステムのメタデータ・サーバとして使用可能なノードに設定できますが、アクティブなメタデータ・サーバになるのは、そのファイルシステムをマウントするノードのうちの最初のノードだけです。特定のファイルシステムのメタデータ・サーバとして使用可能なノードのリストは順序付きですが、ネットワーク・レーテンシやその他の予測不能な遅れがあるため、どのノードがアクティブなメタデータ・サーバになるかは予想できません。
- アクティブな CXFS クライアントが存在する場合に、特定のファイルシステムのメタデータ・サーバとして使用可能な最後のノードが停止すると、すべてのクライアントがそのファイルシステムを使用できなくなります。メタデータ・サーバとして使用可能なほかのノードがリストにある場合は、回復が実行されます。詳細については、209 ページの「メタデータ・サーバの回復」を参照してください。
- CXFS ファイルシステムをエクスポートしてその他の NFS クライアントで使用する場合、最適なパフォーマンスを得るには、アクティブなメタデータ・サーバからそのファイルシステムをエクスポートする必要があります。CXFS ファイルシステムの NFS エクスポートについての詳細は、203 ページの「NFS エクスポート・スクリプト」を参照してください。

詳細については、306 ページの「読取りおよび書込み時のメタデータの流れ」を参照してください。

システム表示

CXFS は、複数のファイルシステムの単一のシステム表示を備えています。SAN の各ホストはすべて共有ディスクに直接アクセスでき、どのホストでもファイルへのパス名は同じです。CXFS では、ディスク・チャネルとストレージを追加して、ホストからディスクへの直接の帯域幅を増やすことで、必要に応じて共有ファイルシステムのパフォーマンスを拡張できます。CXFS 共有ファイルのパフォーマンスは、LAN の速度や、データが中央のファイル・サーバを通じて渡されることによるボトルネックのために制限されることはありません。CXFS 共有ファイルでは、ローカル・ディスク・アクセスと同等の速度に、クラスタの柔軟性、スケーラビリティ、および信頼性が統合されています。

ハードウェアとソフトウェアのサポート

この節では、CXFS の必要条件、互換性、および推奨事項について説明します。

必要条件

CXFS では、以下の要件が必要です。

- 以下のプラットフォームを使用して構成された SAN ハードウェア。メタデータ・サーバのために必要です。
  - SGI Origin 300 サーバ
  - SGI Origin 3000 シリーズ
  - SGI 2000 シリーズ
  - SGI Origin 200 サーバ
  - Silicon Graphics Onyx2 システム
  - Silicon Graphics Onyx 3000 シリーズ
  - Silicon Graphics Octane システム
  - Silicon Graphics Octane2 システム

メモ: サポートされているハードウェアについての詳細は、リリース資料に付属の Entitlement Sheet を参照してください。サポートされていないハードウェアを使用すると、CXFS ライセンスの違反となります。CXFS では Silicon Graphics O2 ワークステーションはサポートされていないため、CXFS シリアル・ハードウェア・リセット・サーバとして使用できません。また、CXFS では JBOD もサポートされていません。

- 『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』で定義されている CXFS クライアントのみのプラットフォーム。
- 各ノードに接続された 100baseT TCP/IP のプライベート・ネットワーク。
- シリアル・ハードウェア・リセット回線、または I/O 制限と SGI が販売、サポートする Brocade スイッチ。サーバとして使用可能なノードが 2 つだけのクラスタには、このようなソリューションが必須です。より大きなクラスタでは、サーバとして使用可能なノードを奇数個使用することをお薦めします。
- I/O 制限をサポートするには、SGI が販売、サポートする Brocade スイッチが必須です。Solaris および Windows では、データの整合性を保護するために、I/O 制限が必要です。したがって、マルチ OS クラスタには Brocade スイッチが必要となります。
- RAID ハードウェア。
  - SGI TP9400
    - TP9400 4.0 CD で供給される 4774 または 4884 ユニットに必要なコントローラ・ファームウェアおよび NVSRAM ファイル。
    - Mojave コードを実行している必要があります（4884 に対するデフォルト）。
  - SGI TP9100
    - SGI が販売、サポートする Brocade スイッチを介してサポートされる 2 Gビットの TP9100、または、シリアル・ハードウェア・リセット回線を使用する IRIX ノードに対しては直接接続。
    - 次の条件に適合する 1 Gビットの TP9100: SGI が販売、サポートする Brocade スイッチのみ、リリース 4 ファームウェア、光ファイバ接続（その他の条件が適用される場合もあります）。
- CXFS ノード（特にメタデータ・サーバ）の適切な処理能力。必要な通信と I/O オーバーヘッドを処理できる必要があります。システムには 512 MB 以上の RAM が必要です。
- CXFS 用の FLEXlm ライセンス・キー。

  XVM は、ミラー化機能を備えています。クラスタ内の特定のノードからミラー化ボリュームにアクセスする場合は、XFS Volume Plexing ソフトウェア・オプションを購入し、FLEXlm ライセンスを取得してインストールする必要があります。ライセンスは、ミラー化ボリュームにアクセスするノードにのみインストールしてください。このライセンスの購入についての詳細は、日本 SGI の営業担当員までお問合わせください。

メモ: IRIX 6.5.15f 以降にアップグレードする、パーティションが設定された Origin 3000 および Onyx 3000 システムには、交換用ライセンスが必要です。IRIX 6.5.15f より前のバージョンでは、これらのパーティションが設定されたシステムには、システム内のすべてのパーティションで同じライセンスが使用されていました。詳細については、『Start Here/Welcome』および Web ページ `http://www.sgi.com/support/licensing/partitionlic.html` を参照してください。

- XVM ボリューム・マネージャ。IRIX リリースの一部として提供されています。

1 つのクラスタでは、最大 32 個のノードがサポートされています。そのうち最大 16 個のノードは、CXFS 管理ノードになることができます。

CXFS と IRIS FailSafe 2.1 以降の両方が実行されている(同時実行と呼ばれる)クラスタでは最大 32 個のノードがサポートされており、そのうち最大 8 個のノードで FailSafe を実行できます。FailSafe と同時に実行されている場合でも、プールとクラスタはそれぞれ 1 つだけです。設定についての詳細は、39 ページの「IRIS FailSafe 同時実行の概要」を参照してください。

すべての IRIX ノードで、IRIX 6.5.12f 以降の同バージョンまたは連続するバージョンの IRIX オペレーティング・システム (OS: Operating System) が実行されている必要があります(6.5.16f と 6.5.17f など)。

クライアントでは、IRIX、および『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』に記載されているその他のオペレーティング・システムが実行されている必要があります。それ以外のオペレーティング・システムが実行されているノードを含むクラスタ内の IRIX ノードでは、6.5.16f 以降が実行されている必要があります。

互換性

CXFS は、以下と互換性があります。

- DMF (Data Migration Facility) と TMF (Tape Management Facility)。
- Trusted IRIX/CMW (Compartmented Mode Workstation)。DMF (Data Migration Facility) および TMF (Tape Management Facility) を使用して、CXFS を SGI Trusted IRIX クラスタで使用できるようになりました。

  CXFS と Trusted IRIX を同時に実行したい場合は、クラスタ内のすべてのノードで Trusted IRIX を実行することが推奨されています。すべてのノードが IRIX ノードまたは Trusted IRIX ノードでなければなりません(マルチ OS クラスタでは Trusted IRIX を実行できません)。詳細については、235 ページの 第8章「Trusted IRIX と CXFS」を参照してください。
- IRIS FailSafe(同時実行)。39 ページの「IRIS FailSafe 同時実行の概要」および『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。

- IRISconsole。『IRISconsole Administrator's Guide』を参照してください。CXFS では、CXFS ノードとして Silicon Graphics O2 ワークステーションはサポートされていません。
- シリアル・ハードウェア・リセット機能に使用されるシリアル・ポート・マルチプレクサ。

推奨事項

CXFS を実行する場合は、以下が推奨されています。

- CXFS 管理ノード(`cluster_admin` ソフトウェア製品がインストールされる)にすることができるのは、メタデータ・サーバとして使用可能なノードになることを希望するノードだけです。CXFS クライアント管理ノードは、IRIS FailSafe との同時実行に必要な場合にのみ使用してください。その他のすべてのノードは、クライアントのみのノード(`cxfs_client` がインストールされる)にしてください。
- パフォーマンスおよび制御の点から、ハブではなくネットワーク・スイッチを使用してください。
- すべてのノードは、同じ物理ネットワーク・セグメント上に存在させてください。
- 実作業用には、サーバとして使用可能なノードが少なくとも 3 つのクラスタを設定してください。
- CXFS と Trusted IRIX または CMW を同時に実行したい場合は、クラスタ内のすべてのノードで Trusted IRIX を実行してください。また、クラスタのすべてのノードのユーザ ID、アクセス・コントロール・リスト (ACL: Access Control List)、および CAP 機能 (CAPABILITY) が同じになるようシステムを設定する必要があります。
- 長時間かかるジョブがある場合は、IRIX のチェックポイント機能と再起動機能を使用してください。詳細については、`cpr`(1) のマン・ページを参照してください。

設定および管理に関する推奨事項については、67 ページの「初期設定の推奨事項」を参照してください。

IRIS FailSafe 同時実行の概要

CXFS を使用することで、コンピュータのグループは、高いパフォーマンスを保ったまま、一貫性のある方法で大量のデータを共有できます。SGI IRIS FailSafe 製品には、高可用性サービスを提供するための一般的な機能が用意されています。

したがって、CXFS クラスタ内で FailSafe を使用して(同時実行とも呼ばれる)、CXFS ファイルシステム上で実行される高可用性サービス(NFS や Web など)を提供できます。この組合わせにより、クラスタ・システムにおいて、パフォーマンスの高い共有データ・アクセスが高可用性アプリケーションに提供されます。

CXFS 6.5.10 以降と IRIS FailSafe 2.1 以降(および関連するパッチ)は、同じシステムにインストールして実行できます。

同時実行クラスタ内のノードのサブセットは、FailSafe ノードとして使用するよう設定できます。したがって、同時実行クラスタでは、FailSafe を実行するノードを最大 8 つまで使用できます。

詳細については、227 ページの 第7章「IRIS FailSafe との同時実行」を参照してください。

クラスタ・マネージャ・ツールの概要

CXFS 製品は、以下のツールを備えています。

- CXFS マネージャ・グラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface)
- `cmgr`(1M)(`cluster_mgr` とも呼ばれる)
- `cluster_status`(1M)
- `clconf_info`

GUI は 1 つの CXFS 管理ノードに接続されており、その他のコマンドは 1 つの CXFS 管理ノード上で実行される必要があります。クライアントのみのノードに関する類似の情報は、`cxfs_info` コマンドを実行することによって取得できます。

CXFS 設定タスクは、GUI または `cmgr`(1M) クラスタ・マネージャ・コマンドをのいずれかを使用して実行できます。これらのツールを使用すると、クラスタ・データベースが更新され、メタデータおよびクラスタ設定の情報が永久に保存されます。

クラスタの設定とモニタを行う際にこれらのツールが使用する下層のソフトウェア・コマンド・ライン・インタフェース (CLI: Command Line Interface) は同じですが、GUI は、実作業用システムでは特に重要な以下の機能を備えています。

- 青いテキストをクリックすると、その概念や入力フィールドに関する詳細を参照できます。「ヘルプ (Help)」ボタンをクリックしてオンライン・ヘルプを利用することもできます。
- クラスタの状態が視覚的に表示されるため、ステータスやフェイルオーバーをすばやく認識できます。
- 状態が動的に更新されるので、継続的にシステムをモニタできます。
- クラスタ設定情報に変更を加える前に、構文が正しいかどうかに関してすべての入力値がチェックされます。どのタスクでも、「OK」をクリックするまでクラスタ設定は更新されません。
- タスクにより、設定および管理操作を順番に進めることができ、タスクを実行しながらクラスタ設定に実際に変更を加えることができます。
- これらのグラフィカルなツールは、任意の IRIX ワークステーション、または Windows や Linux のコンピュータとラップトップを含む Java が有効な Web ブラウザがインストールされた任意のコンピュータから、リモートで安全に実行できます。

cmgr(1M) コマンドの機能は GUI よりも限られており、1 つの CXFS 管理ノード（cluster_admin ソフトウェア製品がインストールされるノード）上だけでクラスタ・システムの設定と管理を行うことができます。また、ヘルプや書式付き出力は最低限しか用意されておらず、クエリーを実行した場合を除き、動的にステータスを参照することはできません。ただし、経験が豊富な管理者の方にとっては、基本的な設定タスクや実作業環境で独立した単一のタスクを実行する場合、またはスクリプトを実行して特定のクラスタ管理タスクを自動化する場合は、cmgr の方が便利かもしれません。build_cmgr_script(1M) コマンドを使用すると、クラスタ・データベースの内容に基づいて cmgr(1M) スクリプトを自動的に作成することができます。

関連する変更がプール内のオンライン・データベースのすべてのコピーに適用されると、GUI の表示領域が更新されます。GUI、または cmgr(1M)、cluster_status(1M)、および clconf_info コマンドを使用すると、データベースの状態を表示することができます。データベースは、直接アクセスすることはできないファイルの集合です。クライアントのみのノードでは、cxfs_info コマンドを使用できます。

詳細については、以下を参照してください。

- 89 ページの「GUI の概要」
- 142 ページの「cmgr(1M) 概要」
- 196 ページの「cmgr スクリプトの自動作成」
- 237 ページの 第9章「ステータスのモニタ」

IRIX システム: CXFS ソフトウェアのインストールとシステムの準備

注意: CXFS は複雑な製品です。CXFS が最適にインストールおよび設定されるよう、必ず日本 SGI から初期セットアップ・サービスをご購入ください。この章は、カスタマが直接使用することを目的としておらず、リファレンスとして提供されています。

IRIX システムにインストールするソフトウェアによって、CXFS 管理ノード(CXFS クラスタ・サービスおよびクラスタ・データベースの完全なセットが含まれる)または CXFS クライアントのみのノードになるかどうかが決定されます。メタデータ・サーバとして実行する予定のノードは、CXFS 管理ノードとしてインストールする必要があります。その他のすべてのノードは、クライアントのみのノードとしてインストールすることが推奨されます。

この章は、以下の手順で構成されています。

1. 44 ページの「CXFS 管理ノードへの IRIX ソフトウェアのインストール」
2. 49 ページの「クライアントのみのノードへの IRIX ソフトウェアのインストール」
3. 51 ページの「IRIX システム・ファイルの設定」
4. 60 ページの「(オプション) 自動再起動用の設定」
5. 60 ページの「ネットワーク・インタフェースの設定」
6. 62 ページの「IRIX ノードのシリアル・ポートの設定」
7. 63 ページの「システムの再起動」
8. 63 ページの「システムのテスト」
9. 66 ページの「アップグレードのためのファイルシステム定義の切替え」

これらの手順を完了したら、67 ページの 第3章「クラスタの初期設定」を参照してください。特定の設定タスクについての詳細は、89 ページの 第4章「CXFS の GUI タスク・リファレンス」および 141 ページの 第5章「CXFS の `cmgr` タスク・リファレンス」を参照してください。CXFS および Trusted IRIX のインストールについての詳細は、235 ページの 第8章「Trusted IRIX と CXFS」を参照してください。

CXFS クラスタのインストールと設定を行う前に、このマニュアル全体、特に 249 ページの 第10章「トラブルシューティング」をよくお読みください。IRIS FailSafe との同時実行を使用する場合は、『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。Solaris クライアントのみのノードおよび Windows クライアントのみのノードを使用する場合は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

CXFS 管理ノードへの IRIX ソフトウェアのインストール

CXFS メタデータ・サーバになる可能性のあるすべてのノードは、CXFS 管理ノードとしてインストールする必要があります。その他のすべてのノードは、クライアントのみのノードにすることが推奨されます。

CXFS 管理ノードに対する CXFS base CD のインストールには、約 30.3 MB のスペースが必要です。

CXFS 管理ノードに必要な IRIX ソフトウェアをインストールするには、以下の手順に従います。

1. プール内の各 CXFS 管理ノードで、『IRIX 6.5 Installation Instructions』に従って IRIX 6.5.18f にアップグレードします。

   特定のノードがアップグレードされたことを確認するには、次のコマンドを使用して、現在インストールされているシステムを表示します。

   ```
   # uname -aR
   ```

2. (シリアル・ポート・サーバのあるサイトの場合) 各 CXFS 管理ノードに、オペレーティング・システムに適したバージョンのシリアル・ポート・サーバ・ドライバをインストールします。この際、シリアル・ポート・サーバに付属の CD を使用します。インストールの後、システムを再起動します。

   詳細については、シリアル・ポート・サーバに付属のドキュメントを参照してください。

3. プール内の各 CXFS 管理ノードで、以下の作業を行います。

   a. CXFS ライセンス・キーをインストールします。ライセンス・キーが必要な製品を注文した場合、キーはインストール手順の説明と共に受注部門から自動的に電子メールで送信されます。この情報がない場合は、日本 SGI または最寄のサポート・プロバイダにお問合わせください。

      ライセンスが適切にインストールされている場合は、CXFS ソフトウェアのインストールが完了すると、`cxfslicense` コマンドから次の出力が表示されます。

      ```
      # /usr/cluster/bin/cxfslicense -d
      CXFS license granted.
      ```

CXFS ライセンスが適切にインストールされていない場合は、CXFS を実行すると、コンソールに次のエラーが表示されます。

```
Cluster services:CXFS not properly licensed for this host.  Run
        '/usr/cluster/bin/cxfslicense -d'
for detailed failure information.  After fixing the
license, please run '/etc/init.d/cluster restart'.
```

次の例のようなエラーが SYSLOG ファイルに表示されます。

```
Mar  4 12:58:05 6X:typhoon-q32 crsd[533]: <<CI> N crs 0> Crsd restarted.
Mar  4 12:58:05 6X:typhoon-q32 clconfd[537]: <<CI> N clconf 0>
Mar  4 12:58:05 5B:typhoon-q32 CLCONFD failed the CXFS license check.Use the
Mar  4 12:58:05 5B:typhoon-q32    '/usr/cluster/bin/cxfslicense -d'
Mar  4 12:58:05 5B:typhoon-q32 command to diagnose the license problem.
```

システムの CPU の数を増やすと、新しいライセンスが必要になることがあります。IRIX 6.5.15f 以降にアップグレードする、パーティションが設定された Origin 3000 および Onyx 3000 システムには、交換用ライセンスが必要です。IRIX 6.5.15f より前のバージョンでは、これらのパーティションが設定されたシステムには、システム内のすべてのパーティションのライセンスに同じ lmhostID が使用されていました。詳細については、6.5.15 の『Start Here/Welcome』および Web ページ http://www.sgi.com/support/licensing/partitionlic.html を参照してください。

ソフトウェア・ライセンスのインストールについての詳細は、『IRIX 6.5 Installation Instructions』ブックレットを参照してください。

b. CD-ROM #2 を CD-ROM ドライブに挿入します。

c. 次を入力して、挿入されている CD-ROM を読取るよう inst に指示します。

```
Inst> from /CDROM/dist
```

注意: inst -r オプションを使用して別の root にインストールしないでください。終了操作 (exitop) の中には、別の root への相対パス名を使用しないものがあります。このため、-r オプションを使用すると、メインの root ファイルシステムと別の root ファイルシステムの両方で問題が発生する可能性があります。詳細については、inst(1M) のマン・ページを参照してください。

d. 次のメッセージが表示されたら、<Enter> キーを押して CD-ROM を読取ります。

```
Install software from : [/CDROM/dist]
```

e. CXFS ソフトウェアをインストールします。

```
Inst> keep *
Inst> install cluster_admin
Inst> install cluster_cluster
Inst> install cluster_services
Inst> install cxfs
Inst> install eoe.books.xvm
Inst> install eoe.sw.xvm
Inst> install sysadm_base
Inst> install sysadm_cxfs
Inst> install sysadm_cluster
Inst> go
```

以下のサブシステムがインストールされます。

```
cluster_admin.man.man
cluster_admin.sw.base
cluster_control.man.man
cluster_control.sw.base
cluster_control.sw.cli
cluster_services.man.man
cluster_services.sw.base
cluster_services.sw.cli
cxfs.books.CXFS_AG
cxfs.sw.cxfs
cxfs.sw.xvm_cell
eoe.books.xvm
eoe.sw.xvm
sysadm_base.man.priv
sysadm_base.man.relnotes
sysadm_base.man.server
sysadm_base.sw.client
sysadm_base.sw.dso
sysadm_base.sw.priv
sysadm_base.sw.server
sysadm_cxfs.man.pages
sysadm_cxfs.man.relnotes
sysadm_cxfs.sw.client
sysadm_cxfs.sw.desktop
sysadm_cxfs.sw.server
sysadm_cxfs.sw.web
sysadm_cluster.man.relnotes
```

```
sysadm_cluster.sw.client
sysadm_cluster.sw.server
```

sysadm_base がインストールされると、tcpmux サービスが /etc/inetd.conf ファイルに追加されます。

メモ: root 以外のログインから CXFS マネージャ・グラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) を実行したい場合は、sysadmdesktop もインストールします。これにより、CXFS コマンドの実行に必要な特権を含むユーザ特権を付与できるコマンドが提供されます。sysadmdesktop をインストールする場合は、以下のサブシステムをインストールします。

```
sysadmdesktop.man.base
sysadmdesktop.man.relnotes
sysadmdesktop.sw.base
sysadmdesktop.sw.data
sysadmdesktop.sw.sysadm
```

---

4. GUI の Web ベースのバージョンを使用する場合、任意のプラットフォームで動作する Java が有効な Web ブラウザを使用して管理操作を実行するには、接続先の CXFS 管理ノードに sysadm_base.sw.client、sysadm_cxfs.sw.client、および sysadm_cxfs.sw.web サブシステムがインストールされている必要があります。これらのサブシステムは、手順 3e でインストールしたデフォルト・ソフトウェアの一部です。

   GUI の Web ベースのバージョンを使用する場合は、以下のいずれかがインストールされている必要もあります。

   - sgi_apache.sw.server
   - nss_enterprise.sw.server（Netscape CD-ROM から）

   これらのサブシステムの 1 つがまだインストールされていない場合は、適切な CD-ROM をロードして、該当するサブシステムをインストールします。

5. IRIX デスクトップ（クラスタ内またはクラスタ外のノードの場合がある）から GUI クライアントを実行する場合は、以下のサブシステムをインストールします。

```
Inst> keep *
Inst> install java_eoe.sw (SGI version 3.1.1)
Inst> install java_eoe.sw32
Inst> install sysadm_base.man
Inst> install sysadm_base.sw.client
Inst> install sysadm_cluster.sw.client
Inst> install sysadm_cxfs.man
Inst> install sysadm_cxfs.sw.client
```

```
Inst> install sysadm_cxfs.sw.desktop
Inst> go
```

注意: GUI は、Java Execution Environment 3.2 (Sun JRE 1.1.8) でのみ動作します。これは、IRIX 6.5.*x* リリースに付属する Java のバージョンです。

SGI の Web サイトには Java 2 も掲載されていますが、このバージョンの Java は GUI では使用できません。1.1.8 以外のバージョンの Java を使用すると、GUI は正しく実行できません。

6. ワークステーションが、Java をサポートする Web ブラウザから GUI クライアントを起動する IRIX マシンの場合は、IRIX 6.5.*x* CD から `java_plugin` サブシステムをインストールします。これは Runtime Plug-in for IRIX、Java Edition 1.1.1b で、JRE 1.1.8 をサポートしています。ただし、IRIX 上では Web ブラウザから GUI を起動する方法は推奨されていません。上記の手順 5 のように、IRIX デスクトップから GUI クライアントを実行する方法が推奨されています。

   `java_plugin` のすべてのサブシステムをインストールしようとすると、サブシステム (`java_plugin.sw.swing101`、`java_plugin.sw.swing102`、および `java_plugin.sw.swing103`) に互換性のないことが `inst` によって報告されます。これらの 3 つのサブシステムは GUI では使用されないので、インストールしないでください。

   Java Plug-in をインストールしたら、すべてのブラウザ・ウィンドウを閉じ、ブラウザを再起動する必要があります。

7. PCP (Performance Co-Pilot) を使用して XVM 統計を実行したい場合は、デフォルトの `pcp_eoe` サブシステムと共に、`pcp_eoe.sw.xvm` も選択してインストールします。これにより、PCP PMDA (XVM 統計をエクスポートするエージェント) が終了操作 (exitop) としてインストールされます。

8. `inst` を終了します。

   ```
   Inst> quit
   ```

   このプロセスは、完了までに数分かかる場合があります。

   ソフトウェアをインストールして `inst` インタフェースを終了すると、変更内容を適用するためにシステムを再起動するようプロンプトが表示されます。ただし、再起動は 63 ページの「システムの再起動」で説明する手順で行います。

クライアントのみのノードへの IRIX ソフトウェアのインストール

メモ: Solaris または Windows が実行されているノードへのソフトウェアのインストールについての詳細は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

必要な IRIX ソフトウェアをインストールするには、以下の手順に従います。

1. プール内の各 IRIX クライアントのみのノードで、『IRIX 6.5 Installation Instructions』に従って IRIX 6.5.18f にアップグレードします。

   特定のノードがアップグレードされたことを確認するには、次のコマンドを使用して、現在インストールされているシステムを表示します。

   ```
   # uname -aR
   ```

2. ((シリアル・ポート・サーバのあるサイトの場合)各ノードに、オペレーティング・システムに適したバージョンのシリアル・ポート・サーバ・ドライバをインストールします。この際、シリアル・ポート・サーバに付属の CD を使用します。インストールの後、システムを再起動します。

   詳細については、シリアル・ポート・サーバに付属のドキュメントを参照してください。

3. プール内の各 IRIX クライアントのみのノードで、以下の作業を行います。

   a. CXFS ライセンス・キーをインストールします。ライセンス・キーが必要な製品を注文した場合、キーはインストール手順の説明と共に受注部門から自動的に電子メールで送信されます。この情報がない場合は、日本 SGI または最寄のサポート・プロバイダにお問合わせください。

      ライセンスが適切にインストールされている場合は、`cxfslicense` コマンドから次の出力が表示されます。

      ```
      # /usr/cluster/bin/cxfslicense -d
      CXFS license granted.
      ```

      CXFS ライセンスが適切にインストールされていない場合は、CXFS を実行すると、コンソールに次のエラーが表示されます。

      ```
      Cluster services:CXFS not properly licensed for this host.  Run
              '/usr/cluster/bin/cxfslicense -d'
      for detailed failure information.  After fixing the
      license, please run '/etc/init.d/cluster restart'.
      ```

次の例のようなエラーが SYSLOG ファイルに表示されます。

```
Mar  4 12:58:05 6X:typhoon-q32 crsd[533]: <<CI> N crs 0> Crsd restarted.
Mar  4 12:58:05 6X:typhoon-q32 clconfd[537]: <<CI> N clconf 0>
Mar  4 12:58:05 5B:typhoon-q32 CLCONFD failed the CXFS license check.Use the
Mar  4 12:58:05 5B:typhoon-q32    '/usr/cluster/bin/cxfslicense -d'
Mar  4 12:58:05 5B:typhoon-q32 command to diagnose the license problem.
```

システムの CPU の数を増やすと、新しいライセンスが必要になることがあります。IRIX 6.5.15f 以降にアップグレードする、パーティションが設定された Origin 3000 および Onyx 3000 システムには、交換用ライセンスが必要です。IRIX 6.5.15f より前のバージョンでは、これらのパーティションが設定されたシステムには、システム内のすべてのパーティションのライセンスに同じ lmhostID が使用されていました。詳細については、6.5.15 の『Start Here/Welcome』および Web ページ http://www.sgi.com/support/licensing/partitionlic.html を参照してください。

ソフトウェア・ライセンスのインストールについての詳細は、『IRIX 6.5 Installation Instructions』ブックレットを参照してください。

b. CD-ROM #2 を CD-ROM ドライブに挿入します。

c. 次を入力して、挿入されている CD-ROM を読取るよう inst に指示します。

```
Inst> from /CDROM/dist
```

注意: inst -r オプションを使用して別の root にインストールしないでください。終了操作 (exitop) の中には、別の root への相対パス名を使用しないものがあります。このため、-r オプションを使用すると、メインの root ファイルシステムと別の root ファイルシステムの両方で問題が発生する可能性があります。詳細については、inst(1M) のマン・ページを参照してください。

d. 次のメッセージが表示されたら、<Enter> キーを押して CD-ROM を読取ります。

```
Install software from : [/CDROM/dist]
```

e. CXFS ソフトウェアをインストールします。

```
Inst> keep *
Inst> install cxfs
Inst> install cxfs_client
Inst> install eoe.books.xvm
Inst> install eoe.sw.xvm
```

```
Inst> go
```

以下のサブシステムがインストールされます。

```
cxfs.books.CXFS_AG
cxfs_client.man.man
cxfs_client.sw.base
cxfs.sw.cxfs
cxfs.sw.xvm_cell
eoe.books.xvm
eoe.sw.xvm
```

4. PCP (Performance Co-Pilot) を使用して XVM 統計を実行したい場合は、デフォルトの pcp_eoe サブシステムと共に、pcp_eoe.sw.xvm も選択してインストールします。これにより、PCP PMDA (XVM 統計をエクスポートするエージェント) が終了操作 (exitop) としてインストールされます。

5. inst を終了します。

   ```
   Inst> quit
   ```

   このプロセスは、完了までに数分かかる場合があります。

   ソフトウェアをインストールして inst インタフェースを終了すると、変更内容を適用するためにシステムを再起動するようプロンプトが表示されます。ただし、再起動は 63 ページの「システムの再起動」で説明する手順で行います。

IRIX システム・ファイルの設定

CXFS ソフトウェアをインストールするときは、システム・ファイルに関するいくつかの考慮事項があります。ネットワーク設定は非常に重要です。クラスタの各ノードは、その他のネットワーク・ルーティングを経由せずに、論理名と IP アドレスの両方で同じクラスタ内のすべてノードと相互に通信できなければなりません。つまり、適切な名前解決が重要です。このため、静的ルーティングが推奨されています。

この節では、IRIX ノードに対して行う必要がある必須の変更、およびオプションの変更について説明します。ほかのオペレーティング・システムが実行されているノードに必要な変更についての詳細は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf

注意: CXFS クラスタを設定する前に、以下の規則を理解しておくことが非常に重要です。

すべてのクラスタ・ノードには、以下のホスト名解決規則と推奨事項が適用されます。

- ホスト名は、アンダースコア (_) で始めたり、スペースを含めることはできません。
- /etc/sys_id ファイルの値は、/etc/hosts ファイルのノードのプライマリ・ホスト名 (/etc/hosts のノードの IP アドレスの直後のフィールド) に一致しなければなりません。このフィールドは、ホスト名または完全修飾ドメイン名で指定できます。

  以下の場合は、/etc/sys_id の名前を使用する必要があります。

  – CXFS GUI に初めてログインする場合の GUI のログイン・ウィンドウの「サーバ(Server)」フィールド

  – プールの最初のノードを定義する場合の「ホスト名(Hostname)」フィールド

  /etc/hosts ファイルの形式は次のとおりです。*primary_hostname* には、単純なホスト名または完全修飾ドメイン名を指定できます。

  IP_address primary_hostname aliases

  CXFS は、ハートビートまたはコントロール用にプライマリ・ネットワークのみを使用します。2 つ目のネットワークを使用するのは、同時実行クラスタで FailSafe も実行している場合だけにすることが推奨されます。

  たとえば、/etc/hosts に以下が含まれているとします。

```
# The public interface:
128.2.3.4  color-green.sgi.com color-green green

# The private interface:
192.0.1.1  color-green-private.sgi.com  color-green-private green-private
```

  /etc/sys_id ファイルには、ホスト名 color-green または完全修飾ドメイン名 color-green.sgi.com を含めることができますが、エイリアス green を含めることはできません。

  この場合、ログイン画面の「サーバ(Server)」フィールド、および「新規ノードの定義(Define a new node)」ウィンドウの「ホスト名(Hostname)」フィールドには、ホスト名 color-green または完全修飾ドメイン名 color-green.sgi.com を入力します。

メモ: /etc/sys_id の値を使用する必要があるのは、初期設定中に最初のノードにログインするときだけです。以降のノードには、green などのエイリアスを使用できます。

- nsd(1M) ネーム・サービス・デーモンを使用する場合は、NIS (Network Information Service) またはドメイン・ネーム・サービス (DNS: Domain Name Service) よりも先にローカル・ファイルがアクセス

されるようシステムを設定する必要があります。つまり、/etc/nsswitch.conf の hosts の行の最初に files がリストされていなければなりません。例は、次のとおりです。

```
hosts:      files nis dns
```

nis と dns の順序は CXFS では重要ではありません。files は最初にする必要があります。

/etc/config/netif.options ファイルでは、インタフェースの 1 つが /etc/sys_id ($HOSTNAME) の値と等しくなければなりません。

USN (Unified Name Service) およびネーム・サービス・デーモンについての詳細は、nsd(1M) のマン・ページを参照してください。

- /etc/nsswitch.conf または /etc/hosts ファイルを変更した場合は、nsadmin restart コマンドを使用して nsd を再開する必要があります。これにより、キャッシュもフラッシュされます。

  これらのファイルの変更後に nsd(1M) を再開しなければならない理由は、nsd ネーム・サービス・デーモンは実際に /etc/hosts の内容を読取って、その内容をより高速に検索できる形式でメモリ・キャッシュに配置するためです。したがって、変更を反映させて新しい /etc/hosts の情報を RAM キャッシュに配置するには、nsd を再開しなければなりません。/etc/nsswitch.conf を変更した場合は、管理対象のファイルのタイプ (hosts や passwd など)、情報を得るために呼出すサービス、およびこれらのサービスの呼出し順序を判断するために、nsd がこのファイルを再度読取る必要があります。

  クラスタ・サービスがアクティブなときは、クラスタで実行中のノードの IP アドレスや、クラスタの最初のノードの IP アドレスを変更することはできません。

- /etc/hosts ファイルで完全修飾ドメイン名を使用する場合は、すべて完全修飾ドメイン名を使用してください。特定のノードの /etc/sys_id で完全修飾ドメイン名を使用した場合、クラスタ内のノードの /etc/hosts ファイルにそのノードの IP またはホスト名の情報を定義するときは、すべてのノードでそのノードの完全修飾ドメイン名を使用する必要があります。

  通常、完全修飾ドメイン名を使用するかどうかは、クライアント (NFS など) でのクライアント・サーバ・プログラムの名前解決方法や、デフォルトの解決方法などに応じて決定します。

- 最初のノード (クラスタ・データベースを作成するノード) を定義した後で、/etc/nsswitch.conf ファイルのホスト名解決方法を変更した場合は、データベースを再度作成する必要があります。295 ページの「クラスタ・データベースの再作成」を参照してください。

- IRIS FailSafe との同時実行を使用する場合、/etc/sys_id の値を IP アドレス・エイリアスに関連付ける /etc/hosts のエントリは追加しないでください。この場合は、プライマリ・アドレスを使用しなければなりません。

CXFS 管理ノードの /etc/services

プール内の各 CXFS 管理ノードに cluster_admin 製品をインストールする前に、各 CXFS 管理ノードの /etc/services ファイルを編集して、sgi-cad および sgi-crsd のエントリを含めます。これらのプロセスに割当てられたポート番号は、プール内のすべてのノードで同じである必要があります。

メモ: /etc/services に sgi-cmsd および sgi-gcd を追加しなければならないことを示す inst メッセージが表示されます。この 2 つを両方とも追加する必要があるのは、FailSafe との同時実行の場合または FailSafe のみを実行している場合だけです。CXFS だけを実行している場合、sgi-cmsd は必要ありません。また、CXFS のクラスタ・サービスには sgi-cmsd は必要ありません。

以下に、sgi-cad および sgi-crsd の /etc/services エントリの例を示します。

```
sgi-crsd        7500/udp            # Cluster reset services daemon
sgi-cad         9000/tcp            # Cluster Admin daemon
```

CXFS 管理ノードの /etc/config/cad.options

各 CXFS管理ノードの /etc/config/cad.options ファイルには、cad プロセスの開始時にクラスタ管理デーモンによって読取られるパラメータのリストが含まれています。cad はクラスタ情報を提供します。

cad.options ファイルでは、以下のオプションを設定できます。

--append_log	cad ログ情報を cad ログ・ファイルに上書きするのではなく、追加します。
--log_file *filename*	cad ログ・ファイル名。これは、-lf *filename* として指定することもできます。
-vvvv	詳細度レベル。文字 v の数は、ログのレベルを示します。-v に設定すると、最小限のメッセージだけがログに記録されます。-vvvv に設定すると、最大限のメッセージがログに記録されます。

デフォルトのファイルでは、以下のオプションが指定されています。

```
-lf /var/cluster/ha/log/cad_log --append_log
```

以下に、中間レベルの詳細度を使用した /etc/config/cad.options ファイルの例を示します。

```
-vv -lf /var/cluster/ha/log/cad_nodename --append_log
```

デフォルトのログ・ファイルは /var/cluster/ha/log/cad_log です。ログ・ファイルがすでに存在する場合、エラーおよび警告メッセージはそのログ・ファイルに追加されます。

/etc/config/cad.options ファイルの内容は、cmgr(1M) コマンドまたは GUI を使用して変更することはできません。

初期設定時以外に cad.options ファイルを変更した場合、変更を反映させるには、cad プロセスを再開する必要があります。これは、ノードを再起動するか、または次のコマンドを入力することで実行できます。

```
# /etc/init.d/cluster restart
```

実行中のクラスタでこのコマンドを実行した場合、クラスタは動作し続けますが、GUI と cad(1M) デーモンの接続は切断され、再接続するよう GUI によってプロンプトが表示されます。

ライセンスについての詳細は、44 ページの「CXFS 管理ノードへの IRIX ソフトウェアのインストール」および 49 ページの「クライアントのみのノードへの IRIX ソフトウェアのインストール」を参照してください。

CXFS 管理ノードの /etc/config/fs2d.options

各 CXFS 管理ノードの /etc/config/fs2d.options ファイルには、プロセスの開始時に fs2d デーモンによって読取られるパラメータのリストが含まれています。fs2d デーモンは、プール内のすべての CXFS 管理ノードへのクラスタ・データベース (CDB: Cluster Database) の配布を管理します。

表2-1 に、fs2d.options ファイルで設定できるオプションを示します。

表2-1 fs2d.options ファイル・オプション

オプション	説明
-logevents *event name*	選択されたイベントをログに記録します。使用されるイベントの名前には、all、internal、args、attach、chandle、node、tree、lock、datacon、trap、notify、access、 storage などがあります。デフォルトは all です。
-logdest *log destination*	ログの記録先を設定します。使用されるログの記録先には、all、stdout、stderr、syslog、logfile などがあります。複数の記録先が指定されている場合、ログ・メッセージはそれらのすべてに書込まれます。logfile を指定した場合は、-logfile オプションも指定しないと効果がありません。デフォルトは logfile です。
-logfile *filename*	ログ・ファイル名を設定します。デフォルトは /var/cluster/ha/log/fs2d_log です。

オプション	説明
-logfilemax *maximum size*	ログ・ファイルの最大サイズ(バイト単位)を設定します。ファイルが最大サイズを超過した場合は、既存の filename.old が削除された後、現在のファイルの名前が filename.old に変更され、新しいファイルが作成されます。単一のメッセージが複数のファイルに分割されることはありません。-logfile が設定されている場合、デフォルトは 10000000 です。
-loglevel *loglevel*	ログ・レベルを設定します。使用されるログ・レベルには、always、critical、error、warning、info、moreinfo、freq、morefreq、trace、busy などがあります。デフォルトは info です。
-trace *trace_class*	選択されたイベントをトレースします。使用されるトレース・クラスには、all、rpcs、updates、transactions、monitor などがあります。このオプションを指定した場合は、-tracefile または -tracelog、あるいはその両方も指定してください。イベントの 1 つまたは複数のクラスのトレースを要求しても、-tracefile または -tracelog、あるいはその両方を指定しないかぎり、トレースは実行されません。デフォルトは transactions です。
-tracefile *filename*	トレース・ファイル名を設定します。デフォルトはありません。
-tracefilemax *maximum_size*	トレース・ファイルの最大サイズ(バイト単位)を設定します。ファイルが最大サイズを超過した場合は、既存の filename.old が削除された後、現在のファイルの名前が filename.old に変更され、新しいファイルが作成されます。
-[no]tracelog	-tracelog の場合はログの記録先をトレースし、-notracelog の場合はログの記録先をトレースしません。このオプションを設定すると、トレース・メッセージはログの記録先に送信されます。トレース・ファイルがある場合は、このファイルにもトレース・メッセージが書込まれます。デフォルトは -tracelog です。
-[no]parent_timer	-parent_timer の場合は親が存在すると終了し、-noparent_timer の場合は親が存在すると終了しません。デフォルトは -noparent_timer です。
-[no]daemonize	-daemonize の場合はデーモンとして実行し、-nodaemonize の場合はデーモンとして実行しません。デフォルトは -daemonize です。
-l	デーモンとして実行しません。

オプション	説明
-h	使用法メッセージを出力します。
-o help	使用法メッセージを出力します。

これらのオプションに対してデフォルト値を使用すると、レベル info 以下のすべてのログ・メッセージと、トランザクション・イベントのすべてのトレース・メッセージが /var/cluster/ha/log/fs2d_log ファイルに送信されるよう、システムが設定されます。ファイル・サイズが 10 MB に達した場合、このファイルは拡張子が .old で同じ名前のファイルに移動され、ログは同じ名前の新しいファイルにロールオーバーされます。単一のメッセージが複数のファイルに分割されることはありません。

初期設定時以外に fs2d.options ファイルを変更した場合、変更を反映させるには、fs2d プロセスを再開する必要があります。これは、CXFS 管理ノードを再起動するか、または次のコマンドを入力することで実行できます。

```
# /etc/init.d/cluster restart
```

実行中のクラスタでこのコマンドを実行した場合、クラスタは動作し続けますが、GUI と cad(1M) デーモンの接続は切断され、再接続するよう GUI によってプロンプトが表示されます。

例 1

以下に、ログおよびトレース情報を次のように送信する /etc/config/fs2d.options ファイルの例を示します。

- すべてのログ・イベントは /var/adm/SYSLOG に送信されます。
- RPC、更新、およびトランザクションのトレース情報は /var/cluster/ha/log/fs2d_ops1 に送信されます。

  ファイルのサイズが 100,000,000 バイトを超えると、このファイルの名前は /var/cluster/ha/log/fs2d_ops1.old に変更され、新しい /var/cluster/ha/log/fs2d_ops1 ファイルが作成されます。単一のメッセージが複数のファイルに分割されることはありません。

（改行して読みやすくしてあります。）

```
-logevents all -loglevel trace -logdest syslog -trace rpcs
-trace updates -trace transactions -tracefile /var/cluster/ha/log/fs2d_ops1
-tracefilemax 100000000
```

例 2

以下に、すべてのログとトレース・メッセージを 1 つのファイル /var/cluster/ha/log/fs2d_chaos6 に送信する /etc/config/fs2d.options ファイルの例を示します。このファイルでは、最大サイズが 100,000,000 バイトに指定されています。-tracelog は、トレース結果をログ・ファイルに送信します。

(改行して読みやすくしてあります。)

```
-logevents all -loglevel trace -trace rpcs -trace updates
-trace transactions -tracelog -logfile /var/cluster/ha/log/fs2d_chaos6
-logfilemax 100000000 -logdest logfile.
```

(オプション) /.rhosts

CXFS に付属する接続性診断テストを使用する場合、 rsh(1) などのリモート・コマンドを実行するには、各 IRIX ノードの /.rhosts ファイルでクラスタ内のすべてのノードが相互にアクセスできるようにします。接続性テストでは、ローカル・ノードからすべてのノード、およびすべてのノードからローカル・ノードに、ping(1) コマンドを実行します。リモート・ノードで ping を実行する場合、CXFS は rsh (1) (ユーザ root) を使用します。たとえば、cxfs0 、cxfs1 、および cxfs2 の 3 つのノードで構成されるクラスタがあるとします。各 IRIX ノードの /.rhosts ファイルは、以下のようになります (ノード名はプロンプトで示されています)。

```
cxfs0# cat /.rhosts
cxfs1 root
cxfs1-priv root
cxfs2 root
cxfs2-priv root

cxfs1# cat /.rhosts
cxfs0 root
cxfs0-priv root
cxfs2 root
cxfs2-priv root

cxfs2# cat /.rhosts
cxfs0 root
cxfs0-priv root
cxfs1 root
cxfs1-priv root
```

（オプション）/etc/exports

各ノードの /etc/exports ファイルは、NFS クライアントにエクスポートされているファイルシステムを記述します。CXFS マウント・ポイントが exports ファイルに含まれている場合、CXFS マウント後にファイルシステムが再エクスポートされないかぎり、空のマウント・ポイントがエクスポートされます。

（オプション）自動再起動用の設定

ノードがリセットされたりノードの電源がオンになったときに自動的にノードを再起動したい場合は、以下のように各 IRIX ノードの起動パラメータの AutoLoad 変数を yes に設定する必要があります。

```
# nvram AutoLoad yes
```

この設定が推奨されていますが、CXFS では必須ではありません。

この変数の設定は、次のコマンドを使用してチェックできます。

```
# nvram AutoLoad
```

ネットワーク・インタフェースの設定

ネットワークを設定するときは、以下の点に注意してください。

- クラスタのすべてのノードは、ルーティングが指定されていなくても、IP アドレスと論理名を使用して直接通信できなければなりません。
- プライベート・ネットワークは、ハートビートおよびコントロール・ネットワーク専用にします。このネットワークではその他の負荷はサポートされません。
- ハートビートおよびコントロール・ネットワークがすべてのノードに接続されており、すべてのノードがそのネットワークに対して同じサブネットを使用するよう設定されている必要があります。

ネットワーク・インタフェースを設定するには、以下の手順に従います。

1. ネーム・サービスを使用可能にします。ローカル名前解決を使用する必要があります。DNS または NIS を使用している場合でも、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』の定義に従って、ノードのすべての IP アドレスとホスト名を IRIX ノードの /etc/hosts に追加してください。例は、次のとおりです。

   ```
   190.0.2.1 server1-company.com server1
   190.0.2.3 stocks
   190.0.3.1 priv-server1
   190.0.2.2 server2-company.com server2
   190.0.2.4 bonds
   190.0.3.2 priv-server2
   ```

   続いて、これらのすべての IP アドレスをクラスタのその他のノードの /etc/hosts に追加する必要があります。

   詳細については、hosts(4)、named(1M)、dns(7P)、および nis(7P) のマン・ページと、『IRIX Admin: Networking and Mail』および『NIS Administrator's Guide』を参照してください。

メモ: NIS や DNS をノードの IP アドレス・ルックアップ専用に使用すると、NIS や DNS サービスの信頼性が低下した場合に可用性が低下します。

51 ページの「ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf」を参照してください。

2. 1 つのノードで、ノードのインタフェースとそれらの IP アドレスを /etc/config/netif.options ファイルに追加します。

   例は、次のとおりです。

   ```
   if1name=ec0
   if1addr=$HOSTNAME
   ```

   $HOSTNAME は、/etc/hosts に示されている IP アドレスのエイリアスです。51 ページの「ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf」を参照してください。

   インタフェースがほかにもある場合、インタフェースの名前と IP アドレスは、以下のような行に表示されます。

   ```
   if2name=
   if2addr=
   ```

   この例では、コントロール・ネットワーク名および IP アドレスは以下のとおりです。

   ```
   if3name=ec3
   if3addr=priv-$HOSTNAME
   ```

   この例のコントロール・ネットワーク IP アドレスである priv-$HOSTNAME は、/etc/hosts に示されている IP アドレスのエイリアスです。

3. ノードに 9 つ以上のインタフェースがある場合は、if_num の値をインタフェースの数に変更します。インタフェースが 8 つ未満の場合、行は以下のようになります。

   ```
   if_num=8
   ```

4. その他のノードに対して手順 1〜3 を繰返します。

5. ルートがコントロール・ネットワークを通じてアドバタイズされるよう、各 IRIX ノードの /etc/config/routed.options ファイルを編集します。オプションのリストについては、routed(1M) のマン・ページを参照してください。

   例は、次のとおりです。

   ```
   -q -h -Prdisc_interval=45
   ```

これらのオプションは、以下を実行します。

- ルートのアドバタイズをオフにします。
- ホストまたはポイントツーポイント・ルートがアドバタイズされないようにします(同じ方向のネットワーク・ルートがある場合)。
- Router Discovery Advertisement が送信される公称周期を 45 秒(およびそれらの寿命を 135 秒)に設定します。

6. 各 CXFS 管理ノードで電子メール・エイリアスを設定します。この電子メール・エイリアスは、CXFS クラスタ外のユーザやクラスタ内のその他のノードのユーザにクラスタが移行したことを通知する電子メールを送信するためのものです。たとえば、cxfs1 および cxfs2 という 2 つのノードがある場合は、cxfs1 で、/etc/aliases ファイルに次の行を追加します。

   ```
   cxfs_admin:operations@console.xyz.com,admin_user@cxfs2.xyz.com
   ```

   cxfs2 で、 /usr/lib/aliases ファイルに次の行を追加します。

   ```
   cxfs_admin:operations@console.xyz.com,admin_user@cxfs1.xyz.com
   ```

   選択したエイリアス(この場合は cxfs_admin)が、システムの設定時にメールの送信先アドレスに使用する値になります。この例では、operations がクラスタ外のユーザで、admin_user が CXFS 管理ノードのユーザです。

   メモ: /usr/lib/aliases の編集が終わったら、newaliases(1M) コマンドを実行する必要があります。

   ---

7. FDDI が使用されている場合は、FDDIXpress リリース・ノートおよび『FDDIXPress Administration Guide』の説明に従って、新しい FDDI ステーションの設定と確認を完了します。

IRIX ノードのシリアル・ポートの設定

ある CXFS 管理ノードが別の CXFS 管理ノードをリセットするよう設定されている場合は、シリアル・ハードウェア・リセット・シリアル・ケーブルが接続されている tty ポートに対して getty プロセスをオフにする必要があります。この作業は、リセットを受信するノードではなく、リセットを実行する IRIX ノードで実行してください。このためには、各 CXFS 管理ノードで以下の手順を実行します。ほかのオペレーティング・システムが実行されているノードが含まれるクラスタを使用している場合は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

1. シリアル・ハードウェア・リセット回線に使用するポートを決定します。
2. ファイル /etc/inittab を開いて編集します。

3. 手順 1 のポート番号について、右側のコメントを参照してそのポートの行を検索します。
4. この行の 3 つ目のフィールドを `off` に変更します。例は、次のとおりです。

```
t2:23:off:/sbin/getty -N ttyd2 co_9600          # port 2
```

5. ファイルを保存します。
6. 以下のコマンドを入力して、変更を反映します。

```
# killall getty
# init q
```

メモ: IRISconsole システムで実行されるリセット・デーモンを使用してクラスタを設定する場合、IRISconsole にはリセット・ポートを設定しないでください。CXFS システムが実行しているリセット・デーモンと競合する可能性があります。CXFS では CXFS ノードとして Silicon Graphics O2 ワークステーションはサポートされていないため、CXFS シリアル・ハードウェア・リセット・サーバとして使用できません。

システムの再起動

各 IRIX ノードで次のコマンドを実行して、ノードを再起動します。

```
# reboot
```

シャットダウン・プロセスによって `autoconfig` (1M) が実行され、変更が反映されたカーネルが生成されます。

詳細については、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

システムのテスト

この節では、以下の内容について説明します。

- 「プライベート・ネットワーク・インタフェース」
- 64 ページの「CXFS 管理ノードのシリアル・ハードウェア・リセット接続」

プライベート・ネットワーク・インタフェース

プール内の各 IRIX ノードの各プライベート・ネットワークに対して、次を入力します。*nodeIPaddress* は、ノードの IP アドレスです。

```
# /usr/etc/ping -c 3 nodeIPaddress
```

以下のような標準の ping(1) 出力が表示されます。

```
PING IPaddress (190.x.x.x: 56 data bytes
64 bytes from 190.x.x.x: icmp_seq=0 tt1=254 time=3 ms
64 bytes from 190.x.x.x: icmp_seq=1 tt1=254 time=2 ms
64 bytes from 190.x.x.x: icmp_seq=2 tt1=254 time=2 ms
```

ping が失敗した場合は、以下の手順に従います。

1. ifconfig を使用して、ネットワーク・インタフェースが UP に設定されていることを確認します。例は、次のとおりです。

```
# /usr/etc/ifconfig ec3
ec3: flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,FILTMULTI,MULTICAST>
inet 190.x.x.x netmask 0xffffff00 broadcast 190.x.x.x
```

   出力の最初の行にある UP は、インタフェースが設定されていることを示します。

2. ケーブルが正しく接続されていることを確認します。

各 IRIX ノードでこの手順を繰返します。

CXFS 管理ノードのシリアル・ハードウェア・リセット接続

シリアル・ハードウェア・リセット接続をテストするには、以下の手順に従います。

1. IRIX ノードとシリアル・ポート・マルチプレクサの電源がオンになっていることを確認します。
2. プール内の CXFS 管理ノードの 1 つで cmgr(1M) コマンドを開始します。

   ```
   # cmgr
   ```

3. クラスタ全体で CXFS サービスを停止します。

   ```
   stop cx_services for cluster clustername
   ```

   例は、次のとおりです。

   ```
   cmgr> stop cx_services for cluster cxfs6-8
   ```

ノードが正常に非アクティブ状態に移行してCXFSプロセスが終了するまで待ちます。このプロセスには数分かかる場合があります。

4. 以下のいずれかを入力して、シリアル接続をテストします。

   - クラスタ全体をテストするには、次を入力します。

     test serial in cluster *clustername*

     例は、次のとおりです。

```
cmgr> test serial in cluster cxfs6-8
Status: Testing serial lines ...
Status: Checking serial lines using crsd (cluster reset services) from node cxfs8
Success: Serial ping command OK.

Status: Checking serial lines using crsd (cluster reset services) from node cxfs6
Success: Serial ping command OK.

Status: Checking serial lines using crsd (cluster reset services) from node cxfs7
Success: Serial ping command OK.

Notice: overall exit status:success, tests failed:0, total tests executed:1
```

   - 個々のノードをテストするには、次を入力します。

     test serial in cluster *clustername* node *machinename*

     例は、次のとおりです。

```
cmgr> test serial in cluster cxfs6-8 node cxfs7
Status: Testing serial lines ...
Status: Checking serial lines using crsd (cluster reset services) from node cxfs6
Success: Serial ping command OK.

Notice: overall exit status:success, tests failed:0, total tests executed:1
```

   - pingだけを使用して個々のノードをテストするには、次を入力します。

     admin ping node *nodename*

     例は、次のとおりです。

```
cmgr> admin ping node cxfs7

ping operation successful
```

5. コマンドが失敗した場合は、すべてのケーブルが正しく接続されていることを確認してから、再度コマンドを実行します。
6. クラスタのほかのノードでこの手順を繰返します。

アップグレードのためのファイルシステム定義の切替え

CXFS ファイルシステム設定の構造は、IRIX 6.5.13f のリリースで変更されています。6.5.13f リリースにアップグレードすることにより、古い構造から新しい構造に自動的に切替わります。ただし、最初に 6.5.13f のインストールと実行を行わずに 6.5.12f 以前から直接アップグレードする場合は、CXFS ファイルシステム定義を手動で切替える必要があります。

6.5.12f 以前からのアップグレード

メモ: 6.5.13f 以降からアップグレードする場合は、この節の指示に従う必要はありません。ファイルシステム定義は、6.5.13f の実行中に自動的かつ透過的に切替わります。

6.5.12f 以前からアップグレードすると、CXFS ファイルシステムがマウントされなくなり、GUI または `cmgr` のクエリーに表示されないことがわかります。以前の CXFS ファイルシステム定義をすべて新しい形式に切替えるには、CXFS クラスタ内の 6.5.14f 以降のノードの 1 つで次のコマンドを実行するだけです。

```
# /usr/sysadm/privbin/cxfsfilesystemUpgrade
```

このコマンドを実行すると、CXFS ファイルシステムが GUI および `cmgr` の出力に表示されるようになり、それらのステータスが `enabled` で、CXFS サービスがアクティブであればマウントされます。

注意: この切替え操作は一度だけ行い、2 回以上実行しないでください。ファイルシステムに変更を加えた後に 2 回目の `cxfsfilesystemUpgrade` を実行すると、すべての変更内容が失われます。

すべての IRIX ノードが 6.5.14f 以降にアップグレードされた状態での実行

クラスタ内のすべての IRIX ノードを 6.5.14f 以降にアップグレードしたら、誤って `cxfsfilesystemUpgrade` コマンドを再度実行してしまった場合に以前の CXFS ファイルシステム定義によって新しい定義が上書きされるのを防ぐために、これらの古い定義を破棄することが推奨されています。以前の CXFS ファイルシステム定義を破棄するには、次を入力します。

```
# /usr/cluster/bin/cdbutil -c "delete #cluster#clustername#Cellular#FileSystems"
```

クラスタの初期設定

この章では、グラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) または `cmgr`(1M) コマンドを使用してクラスタを初期設定するための推奨事項と、必要な手順の概要を説明します。315 ページの付録C「初期設定チェックリスト」で提供されているワークシートを使用することもできます。既存の IRIS FailSafe クラスタから切替える場合は、107 ページの「GUI を使って CXFS 用に既存の FailSafe クラスタを再設定」を参照してください。

この章は、43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」の手順が実行済みであることを想定しています。また、タスク・リファレンスおよび『XVM Volume Manager Administrator's Guide』の詳しい説明を参照しています。

初期インストールの場合は、GUI の設定ガイド・タスクを使用することが強く推奨されています。73 ページの「GUI を使った設定」を参照してください。また、クラスタを設定する前に、249 ページの 第10章「トラブルシューティング」を含むこのマニュアル全体をよくお読みください。

CXFS を正しく実行するには、各ノードにライセンスがインストールされていなければなりません。システムの CPU の数を増やすと、新しいライセンスが必要になることがあります。43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。ほかのオペレーティング・システムについての詳細は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

初期設定の推奨事項

以下に推奨事項を示します。

- プライベート・ネットワークにネットワーク上の問題がある場合は、CXFS を使用する前に解決してください。
- パフォーマンスおよび制御の点から、ハブではなくネットワーク・スイッチを使用してください。
- すべてのノードは、同じ物理ネットワーク・セグメント上に存在させてください。
- 実作業用には、サーバとして使用可能なノードが少なくとも 3 つで、合計は 32 個以内のノード、そのうち 16 個のノードは管理ノードになることのできるクラスタを設定してください。
- 大規模なクラスタの場合は、サーバとして使用可能な最初の 3 つのノードを定義してから次の設定手順に進むことが推奨されています。つまり、正常な小さいクラスタを作成してから残りのノードを追加してください。

- 新しいクラスタは、同じ IRIX リリースがインストールされている IRIX ノードを使用して作成してください。クラスタが機能している場合は、選択したノードを後から連続したリリースにアップグレードすることができます。
- CXFS と Trusted IRIX/CMW を実行したい場合は、クラスタ内のすべてのノードで Trusted IRIX を実行してください。また、クラスタのすべてのノードのユーザ ID、アクセス・コントロール・リスト (ACL: Access Control List)、および CAP 機能 (CAPABILITY) が同じになるようシステムを設定する必要があります。
- `find`(1) コマンドの使用を避けることにより、不要なメタデータ・トラフィックを避けてください。CXFS ファイルシステムはクラスタ内のすべてのノードでローカルと見なされるので、`find` 実行中にノードが同じファイルシステムに同時にアクセスしようとすると、ファイルシステム・アクティビティが過剰に生成される場合があります。クラスタの 1 つのメタデータ・サーバ上でのみ `find` を実行するには、ノードの `crontab` ファイルを編集してください。
- CXFS ファイルシステムで `xfs_repair`(1M) を使用する前に、必ず日本 SGI の技術サポートに問合わせてください。`xfs_repair` は、ほかのすべてのクラスタ・ノードでファイルシステムがアンマウントされていることを確認した場合にのみ、メタデータ・サーバ上だけで使用してください。

  `xfs_repair` を使用する場合は、正常にアンマウントされたファイルシステム上でのみ実行してください。ファイルシステムが正常にアンマウントされていない場合、コミットされていないメタデータ・トランザクションがログに記録され、`xfs_repair` によって削除されます。通常は、これにより `xfs_repair` からの一部のデータやメッセージが失われ、ファイルシステムが破壊されたように見えます。

  システム・クラッシュまたはファイルシステムのシャットダウンの直後に `xfs_repair` を実行すると、ファイルシステムに不正なログが記録される可能性があります。データの損失を防ぐには、`xfs_repair` を実行する前に、必ずファイルシステムのマウントとアンマウントを行ってください。CXFS によるアンマウントの後、および `xfs_repair` の実行前にファイルシステムをローカルでマウントおよびアンマウントしても、問題は発生しません。
- サーバとして使用可能なノードは奇数個使用してください。
- CXFS 管理ノードは奇数個使用してください。
- メインテナンスの前にクラスタ・サービスをシャットダウンしてください。無効にされたノードは CXFS カーネル・メンバーシップの計算に使用されないので、この操作によって、定足数が失われるのを防止できます。
- 現在のリリースでは、回復を避けてください。

メモ: このリリースでは、再設定の機能はデフォルトで無効になっており、回復の機能はスタンバイ・ノードを使用している場合のみサポートされています。

再設定および回復の機能は完全に実装されていますが、関連する問題が多数存在するため、現在のリリースでは完全にサポートされていません。これらの機能を使用してもデータの整合性が損なわれることはありませんが、クラスタ・ノードのパニックやハングが発生する可能性があります。再設定および回復の機能は、このような問題が解決された時点で、今後のリリースで完全にサポートされる予定です。

ノードをシャットダウンする前に、以下の操作を行ってください。

1. CXFS GUI または `cmgr`(1M) コマンドを使用して、すべてのホストからファイルシステムをアンマウントする
2. クラスタ・サービスをシャットダウンする

- CXFS ファイルシステムの強制アンマウント機能を有効にしてください。これは、デフォルトではオフになっています。特に、ファイルシステムをアンマウントしなければならない状況では、この機能を有効にすると CXFS クラスタの安定性が向上することが多くのサイトで確認されています。

  IRIX ノードでは、この機能は `umount -k` オプションを使用します。`-k` オプションは、該当するファイルシステムのファイルまたは現在のディレクトリを開いているプロセスを強制終了した後、アンマウントします。つまり、I/O が単独で終了するまで待機した結果、アンマウントを失敗させてしまうのではなく、ファイルシステムにアクセスする I/O を終了し、ファイルシステムを直ちにアンマウントできるようにします。

  この機能は、CXFS GUI の次のメニューから利用できます。

  「タスク(Tasks)」
  > 「ファイルシステム(Filesystems)」
  > 「ファイルシステムのアンマウント(Unmount a Filesystem)」

  また、 `cmgr`(1M) コマンドを使用してこの機能を指定し、ファイルシステムを定義することもできます。

  138 ページの「GUI を使った CXFS ファイルシステムのアンマウント」、178 ページの「`cmgr` を使った CXFS ファイルシステムの定義」、および `umount`(1M) のマン・ページと、Solaris の `fuser`(1M) のマン・ページを参照してください。

- ファイルシステム断片化解消プログラムは使用しないでください。IRIX ノードでは、IRIX `fsr`(1M) コマンドは、コマンドの実行対象のファイルシステムのメタデータ・サーバ上でのみ使用できます。

- クラスタ内のいずれのノードの `crontab`(1) ファイル内にも `find`(1) コマンドを含めないでください。CXFS ファイルシステムはローカル・ファイルシステムのように動作するため、各ノードで検索が行わ

れます。したがって、find を使用すると、システムの速度が低下し、一時的にメタデータ・サーバのメモリが大量に消費されます。

- I/O 制限を使用する場合は、ファイバ・チャネル・スイッチの telnet ポートを常に空けておく必要があります。スイッチに対して telnet を実行した後、セッションを接続したままにしないでください。

準備手順

メモ: 管理は、CXFS 管理ノード(cluster_admin ソフトウェア・パッケージがインストールされているノード)に接続されている GUI、または cmgr(1M) コマンドを CXFS 管理ノードで使用することによって実行する必要があります。

以下の手順を完了して、初期クラスタの設定準備を整えます。

- 「ライセンスの確認」
- 71 ページの「クラスタ・デーモンの開始」
- 71 ページの「クラスタ・デーモンが実行されていることの確認」
- 72 ページの「CXFS 管理ノードのホスト名の判断」
- 72 ページの「chkconfig フラグが on であることの確認」

設定作業中は、ログ・ファイルにさまざまな情報メッセージが示されます。273 ページの「通常メッセージ」を参照してください。

ライセンスの確認

cxfslicense コマンドの -d オプションを使用して、CXFS ライセンスがあることを確認します。例は、次のとおりです。

```
# /usr/cluster/bin/cxfslicense -d
CXFS license granted.
```

ライセンスが適切にインストールされている場合は、各 IRIX ノードの /var/flexlm/license.dat ファイルに FEATURE CXFS という行も示されます。

メモ: /var/flexlm/license.dat ファイルは各ノードに固有なので、ノード間でコピーすることはできません。

ソフトウェア・ライセンスのインストールについての詳細は、『IRIX 6.5 Installation Instructions』ブックレットを参照してください。

クラスタ・デーモンの開始

CXFS 管理ノードで次を入力して、クラスタ・デーモンを開始します。

```
# /etc/init.d/cluster start
```

クラスタ・デーモンが実行されていることの確認

ソフトウェアを初めてインストールするときは、以下のデーモンが実行されている必要があります。

- fs2d
- cmond
- cad
- crsd

CXFS サービスを開始すると、clconfd デーモンも開始されます。

実行されているデーモンを判断するには、次を入力します。

```
# ps -ef | grep cluster
```

以下に、初期デーモンだけが実行されている場合の出力の例を示します。ここでは、見やすいように空白を削除してデーモンを強調表示してあります。

```
cxfs6 # ps -ef | grep cluster
root 31431     1 0 12:51:36 ?     0:14 /usr/lib32/cluster/cbe/fs2d /var/cluster/cdb/cdb.db #
root 31456 31478 0 12:53:01 ?     0:03 /usr/cluster/bin/crsd -l
root 31475 31478 0 12:53:00 ?     0:08 /usr/cluster/bin/cad -l -lf /var/cluster/ha/log/cad_log --append_log
root 31478     1 0 12:53:00 ?     0:00 /usr/cluster/bin/cmond -L info -f /var/cluster/ha/log/cmond_log
root 31570 31408 0 14:01:52 pts/0 0:00 grep cluster
```

これらのプロセスが表示されない場合は、ログを開いて、考えられる問題を確認します。デーモンを再開する必要がある場合は、次を入力します。

```
# /etc/init.d/cluster start
```

詳細については、295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」および 297 ページの「デーモン」を参照してください。

CXFS 管理ノードのホスト名の判断

クラスタを初期設定する場合、GUI にログインするときとプールの最初のノードを定義するときは、IP アドレスまたは /etc/sys_id の値(/etc/hosts のノードの IP アドレスのプライマリ名に一致していなければならない)を使用する必要があります。これは、GUI が CXFS 管理ノードに接続されている場合に実行してください。hostname(1) コマンドを実行すると、/etc/sys_id の値が表示されます。例は、次のとおりです。

```
# hostname
cxfs6
```

また、nsd(1M) を使用する場合は、NIS (Network Information Service) またはドメイン・ネーム・サービス (DNS: Domain Name Service) よりも先にローカル・ファイルがアクセスされるようシステムを設定する必要があります。51 ページの「ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf」を参照してください。

注意: これらのファイルを適切に設定すること、およびプールで定義されている最初のノードに対してはプライマリ名を入力することが非常に重要です。以降のノードの定義にはエイリアスを使用できます。43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。

chkconfig フラグが on であることの確認

以下のフラグが on に設定されていることが chkconfig(1M) の出力に表示されることを確認します。

```
# chkconfig
        Flag                State
        ====                =====
        cluster             on
        cxfs_cluster        on
```

このように表示されない場合は、on に設定して再起動します。

例は、次のとおりです。

```
# init 1
# /etc/chkconfig cluster on
# /etc/chkconfig cxfs_cluster on
# init 2
```

GUI を使った設定

GUI を使ってクラスタを初期設定するには、以下を実行します。

- 70 ページの「準備手順」
- 73 ページの「GUI の起動」
- 74 ページの「GUI を使った新規クラスタの設定」
- 76 ページの「GUI を使った新規 CXFS ファイルシステムの設定」

クラスタの表示は、どの CXFS 管理ノードに GUI を接続するかによって変わります。変更を加えたときは、次の変更を行う前に、表示領域に変更内容が表示されるまで待ってください。表示領域に変更内容が表示されるまでは、その変更がクラスタ全体に伝わったことは保証されていません。変更は、特定の時点の GUI の 1 つのインスタンスだけから行ってください。2 つ目のインスタンスから変更を行うと、最初のインスタンスで加えた変更が上書きされる場合があります。

GUI の起動

次を入力して、CXFS マネージャを起動します。

```
# /usr/sbin/cxfsmgr
```

GUI の起動方法はほかにもあります。詳細については、90 ページの「GUI の起動」を参照してください。

接続先の CXFS 管理ノードの名前と `root` パスワードを入力します。

注意: プールで定義されている最初のノードでは、「サーバ(Server)」フィールドで `/etc/sys_id` の値を使用することが非常に重要です。51 ページの「ホスト名解決: `/etc/sys_id`、`/etc/hosts`、`/etc/nsswitch.conf`」を参照してください。

クラスタをまだ定義してないので、クラスタが登録されていないことが通知されますが、これはこの時点では正常なメッセージです。

図3-1 に、「CXFS マネージャ(CXFS Manager)」ウィンドウを示します。

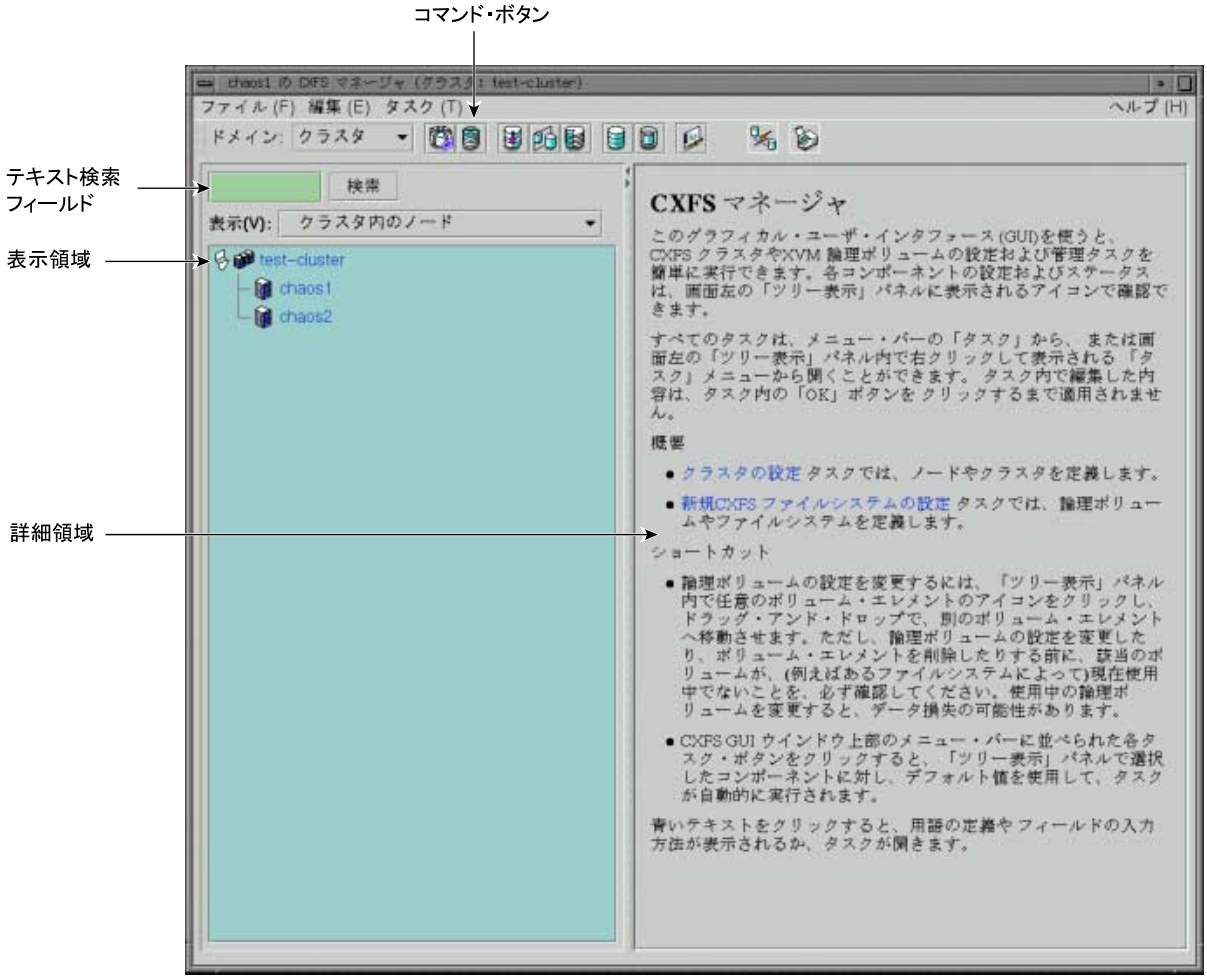


図3-1 CXFS マネージャ

GUI を使った新規クラスタの設定

メモ: CXFS タスク内で青いテキストをクリックすると、その概念や入力フィールドに関する詳細を参照できます。いずれのタスクでも、「OK」をクリックするまでクラスタ設定は更新されません。

「設定ガイド(Guided Configuration)」メニューの「新規クラスタの設定(Set Up a New Cluster)」タスクを使用すると、新しいクラスタを作成するために必要な手順を実行できます。このタスクには、ほかの箇所で詳しく説明されているタスクが含まれます。

以下を実行します。

1. 「ノードの定義(Define a Node)」をクリックして、接続先の CXFS 管理ノードを定義します。この最初のノードの定義では、 `/etc/sys_id` に示されているホスト名が使用されます。72 ページの「CXFS 管理ノードのホスト名の判断」を参照してください。109 ページの「GUI を使ったノードの定義」を参照してください。

2. （オプション）左側の表示領域に最初のノードのアイコンが表示されたら、手順 2「ノードの定義(Define a Node)」をクリックして、クラスタ内のその他のノードを定義します。ホスト名と IP アドレスのペアおよびネットワークの優先度は、クラスタ内の各ノードに対して同じでなければなりません。109 ページの「GUI を使ったノードの定義」を参照してください。

   メモ: 表示領域にこのノードが表示されるまで、別のノードは定義しないでください。ノードを追加するのが早すぎると(データベースにノードを含めることができる前にノードを追加すると)、エラーが発生します。

   各ノードに対してこの手順を繰返します。大規模なクラスタの場合は、サーバとして使用可能な最初の 3 つのノードを定義してから次の手順に進むことが推奨されています。つまり、正常な小さいクラスタを作成してから残りのノードを追加してください。

3. 「クラスタの定義(Define a Cluster)」をクリックして、クラスタ定義を作成します。122 ページの「GUI を使ったクラスタの定義」を参照してください。表示領域にクラスタが表示されていることを確認します。「表示(View)」->「クラスタ内のノード(Nodes in Cluster)」を選択します。

4. 表示領域にクラスタのアイコンが表示されたら、「クラスタ内のノードの追加と削除(Add/Remove Nodes in Cluster)」をクリックして、新しいクラスタにノードを追加します。117 ページの「GUI を使ったクラスタ内のノードの追加と削除」を参照してください。

   「次へ(Next)」をクリックして、タスクの 2 番目の画面に移動します。

5. （オプション）「接続性のテスト(Test Connectivity)」をクリックして、ノードが物理的に接続されていることを確認します。121 ページの「GUI を使ったノードの接続性のテスト」を参照してください。このテストでは、`/etc/.rhosts` ファイルが正しく設定されている必要があります。58 ページの「（オプション）/.rhosts」を参照してください。

6. I/O 制限を使用する場合は、クラスタ内で Brocade スイッチを定義します。Solaris および Windows ノードには、データの整合性を保護するために I/O 制限が必要です。

7. 「CXFS サービスの開始(Start CXFS Services)」をクリックします。125 ページの「GUI を使った CXFS サービスの開始」を参照してください。

8. 「閉じる(Close)」をクリックします。「閉じる」をクリックすると、タスクは終了します。これは、タスクを元に戻すものではありません。

GUI を使った新規 CXFS ファイルシステムの設定

メモ: CXFS タスク内で青いテキストをクリックすると、その概念や入力フィールドに関する詳細を参照できます。いずれのタスクでも、「OK」をクリックするまでクラスタ設定は更新されません。

「新規 CXFS ファイルシステムの設定(Set Up a New CXFS Filesystem)」タスクを使用すると、新しいファイルシステムを作成してクラスタ内のすべてのノードにマウントするために必要な手順を実行できます。このタスクには、ほかの箇所で詳しく説明されているタスクが含まれます。

以下を実行します。

1. CXFS サービスがまだ開始されていない場合は、「CXFS サービスの開始(Start CXFS Services)」をクリックします。現在のステータスは、タスク・リンクの下に表示されています。125 ページの「GUI を使った CXFS サービスの開始」を参照してください。

2. 「ディスクのラベル付け(Label Disks)」をクリックします。

   メモ: ディスクは、ラベルが付けられる前に初期化する必要があります。出荷時のセットアップ中にディスクが初期化されていない場合は、`fx`(1M) コマンドを使用して初期化してください。

   XVM タスクについての詳細は、『XVM Volume Manager Administrator's Guide』を参照してください。

3. 物理ストレージを定義するスライスを、ラベル付きのディスクで作成します。「ディスクのスライス(Slice Disks)」をクリックします。

4. 希望のタイプのファイルシステム(ストライプ、ミラー、または結合)を作成します。

5. 「ファイルシステムの作成(Make the Filesystem)」をクリックします。デフォルトのオプションを使用しない場合は、「サイズの指定(Specify Sizes)」をクリックして、次のページに移動します。詳細については、`mkfs`(1M) のマン・ページと、『IRIX Admin: Disks and Filesystems』および『XVM Volume Manager Administrator's Guide』を参照してください。

6. 「CXFS ファイルシステムの定義(Define a CXFS Filesystem)」をクリックします。このタスクを使用して、新規ファイルシステムを定義し、メタデータ・サーバとして使用可能なノードの順序付きリストと、このファイルシステムのクライアント・ノードのリストを設定できます。135 ページの「GUI を使った CXFS ファイルシステムの定義」を参照してください。

7. 「CXFS ファイルシステムのマウント(Mount a CXFS Filesystem)」をクリックします。このタスクを使用して、クラスタ内のすべてのノードにファイルシステムをマウントできます。137 ページの「GUI を使った CXFS ファイルシステムのマウント」を参照してください。

各ファイルシステムに対して上記の手順を繰返します。

cmgr(1M) コマンドを使った設定

メモ: 初期インストールの場合は、GUI の設定ガイド・タスクを使用することが強く推奨されています。73 ページの「GUI を使った設定」を参照してください。

cmgr(1M) コマンドについての詳細は、マン・ページおよび 141 ページの 第5章「CXFS の cmgr タスク・リファレンス」を参照してください。

cmgr(1m) コマンドを使ってクラスタを初期設定するには、以下を実行します。

1. 70 ページの「準備手順」の指示に従います。
2. クラスタの一部にできるノードを定義します。ホスト名と IP アドレスのペアおよびネットワークの優先度は、クラスタ内の各ノードに対して同じでなければなりません。148 ページの「cmgr を使ったノードの定義」を参照してください。

   大規模なクラスタの場合は、最初の 3 つの CXFS 管理ノードを定義してから次の手順に進むことが推奨されています。つまり、正常な小さいクラスタを作成してから残りのノードを追加します。

   以下に、3 つのノードを定義する順序の例を示します。プロンプトに対してデフォルト値を使用するには、<Enter> キーを押します。このマニュアルの例では、<Enter> キーは示されていません。

   cxfs6 という名前の最初のノードを定義するには、以下を入力します。

```
cxfs6 # /usr/cluster/bin/cmgr -p
Welcome to SGI Cluster Manager Command-Line Interface

cmgr> define node cxfs6
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ?
Is this a FailSafe node <true|false> ? false
Is this a CXFS node <true|false> ? true
Node Function <server_admin|client_admin|client_only> ? server_admin
Operating System  <IRIX|Solaris|Windows> ? irix
Node ID[optional]?
```

```
Partition ID[optional] ? (0)
Do you wish to define failure hierarchy[y/n]:n
Reset type <powerCycle>  ? (powerCycle)
Do you wish to define system controller info[y/n]:y
Sysctrl Type <msc|mmsc|l2> ? (msc)
Sysctrl Password[optional] ? ( )
Sysctrl Status <enabled|disabled> ? enabled
Sysctrl Owner ? cxfs8
Sysctrl Device ? /dev/ttyd2
Sysctrl Owner Type <tty> ? (tty)
NIC 1 - IP Address ? cxfs6

Successfully defined node cxfs6
```

cxfs7 という名前の 2 番目のノードを定義するには、以下を入力します。

```
cmgr> define node cxfs7
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ?
Is this a FailSafe node <true|false> ? false
Is this a CXFS node <true|false> ? true
Node Function <server_admin|client_admin|client_only> ? server_admin
Operating System <IRIX|Solaris|Windows> ? irix
Node ID[optional] ?
Partition ID[optional] ? (0)
Do you wish to define failure hierarchy[y/n]:n
Reset type <powerCycle>  ? (powerCycle)
Do you wish to define system controller info[y/n]:y
Sysctrl Type <msc|mmsc|l2> ? (msc)
Sysctrl Password[optional] ? ( )
Sysctrl Status <enabled|disabled> ? enabled
Sysctrl Owner ? cxfs6
Sysctrl Device ? /dev/ttyd2
Sysctrl Owner Type <tty> ? (tty)
Number of Network Interfaces ? (1)
NIC 1 - IP Address ? cxfs7

Successfully defined node cxfs7
```

cxfs8 という名前の 3 番目のノードを定義するには、以下を入力します。

```
cmgr> define node cxfs8
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ?
Is this a FailSafe node <true|false> ? false
Is this a CXFS node <true|false> ? true
Node Function <server_admin|client_admin|client_only> ? server_admin
Operating System <IRIX|Solaris|Windows> ? irix
Node ID[optional] ?
Partition ID[optional] ? (0)
Do you wish to define failure hierarchy[y/n]:n
Reset type <powerCycle>  ? (powerCycle)
Do you wish to define system controller info[y/n]:y
Sysctrl Type <msc|mmsc|l2> ? (msc)
Sysctrl Password[optional] ? ( )
Sysctrl Status <enabled|disabled> ? enabled
Sysctrl Owner ? cxfs7
Sysctrl Device ? /dev/ttyd2
Sysctrl Owner Type <tty> ? (tty)
Number of Network Interfaces ? (1)
NIC 1 - IP Address ? cxfs8

Successfully defined node cxfs8
```

これで、プールに 3 つのノードが定義されました。これを確認するには、次を入力します。

```
cmgr> show nodes in pool

3 Machine(s) defined
        cxfs6
        cxfs7
        cxfs8
```

ノード cxfs6 の内容を表示するには、次を入力します。

```
cmgr> show node cxfs6

Logical Machine Name: cxfs6
Hostname: cxfs6.americas.sgi.com
Operating System: irix
Node Is FailSafe: false
Node Is CXFS: true
Node Function: server_admin
Nodeid: 13203
Partition id: 0
Reset type: powerCycle
System Controller: msc
System Controller status: enabled
System Controller owner: cxfs8
System Controller owner device: /dev/ttyd2
System Controller owner type: tty
ControlNet Ipaddr: cxfs6
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 1
Node Weight: 1
```

3. クラスタを定義してノードを追加します。166 ページの「cmgr を使ったクラスタの定義」を参照してください。

   たとえば、cxfs6-8 という名前のクラスタを定義して、定義済みのノードを追加するには、以下を入力します。

```
cmgr> define cluster cxfs6-8
Enter commands, you may enter "done" or "cancel" at any time to exit

Is this a FailSafe cluster <true|false> false ?
Is this a CXFS cluster  <true|false> true ?
```

```
Cluster Notify Cmd [optional] ?
Cluster Notify Address [optional] ?
Cluster mode <normal|experimental>[optional]
Cluster ID ? 22

No nodes in cluster cxfs6-8

Add nodes to or remove nodes from cluster cxfs6-8
Enter "done" when completed or "cancel" to abort

cxfs6-8 ? add node cxfs6
cxfs6-8 ? add node cxfs7
cxfs6-8 ? add node cxfs8
cxfs6-8 ? done
Successfully defined cluster cxfs6-8
```

異常時のアクション階層は、異常が発生した場合に使用する方法を決定する命令のセットです。制限を含む階層が設定されている場合は、ここでスイッチを定義することができます。詳細については、189 ページの「cmgr を使ったスイッチと I/O 制限タスク」を参照してください。

詳細については、166 ページの「cmgr を使ったクラスタの定義」を参照してください。

クラスタとその内容を確認するには、以下を入力します。

```
cmgr> show clusters

1 Cluster(s) defined
        cxfs6-8

cmgr> show cluster cxfs6-8
Cluster Name: cxfs6-8
Cluster Is FailSafe: false
Cluster Is CXFS: true
Cluster ID: 22
Cluster CX mode: normal

Cluster cxfs6-8 has following 3 machine(s)
        cxfs6
        cxfs7
        cxfs8
```

スクリプトを使用したこの手順の例については、194 ページの「スクリプトの例」を参照してください。

4. 次を入力して、クラスタの CXFS サービスを開始します。

   start cx_services for cluster *clustername*

   例は、次のとおりです。

   ```
   cmgr> start cx_services for cluster cxfs6-8

   CXFS services have been activated in cluster cxfs6-8
   ```

   このアクションにより CXFS サービスが開始され、システムの再起動時に常に CXFS サービスが自動的に再開されるよう設定されます。

   メモ: GUI または cmgr(1M) を使用して CXFS サービスを停止すると、自動再開機能はオフになります。自動再開機能をオンに戻すには、CXFS サービスを再度開始する必要があります。

   クラスタが動作していることを確認するには、次の cmgr(1M) コマンドを使用できます。

   show status of cluster *clustername*

   例は、次のとおりです。

   ```
   cmgr> show status of cluster cxfs6-8

   Cluster (cxfs6-8) is not configured for FailSafe


   CXFS cluster state is ACTIVE.
   ```

   また、clconf_info(1M) コマンドを使用することもできます。例は、次のとおりです。

```
cxfs6 # /usr/cluster/bin/clconf_info
Membership since Wed May 16 14:42:48 2001
Node          NodeId     Status    Age   Incarnation     CellId
cxfs7          12812       UP      0             0          1
cxfs6          13203       UP      0             0          0
cxfs8          14033       UP      0             0          2
0 CXFS FileSystems
```

   詳細については、171 ページの「cmgr を使ったクラスタの表示」を参照してください。

5. クラスタ内のいずれかの CXFS 管理ノードのシェル・ウィンドウを開き、fx(1M) コマンドを使用して、ディスク・ドライブのボリューム・ヘッダを作成します。詳細については、『IRIX Admin: Disks and Filesystems』を参照してください。

6. XVM 論理ボリュームを作成します。シェル・ウィンドウで、xvm(1M) コマンド・ライン・インタフェースまたは XVM GUI を使用します。詳細については、『XVM Volume Manager Administrator's Guide』を参照してください。

7. ファイルシステムを作成します。シェル・ウィンドウで、mkfs(1M) コマンドまたは XVM GUI を使用します。詳細については、『XVM Volume Manager Administrator's Guide』を参照してください。

8. cmgr(1M) の define cxfs_filesystem サブコマンドを使用して、ファイルシステムをマウントします。177 ページの「cmgr を使った CXFS ファイルシステム・タスク」を参照してください。

   以下に、fs1 ファイルシステムのメタデータ・サーバとして使用可能な 2 つのノードの例を示します。クラスタの開始時に cxfs6 (ランク 0 の優先サーバ) が動作していなかったり、異常が発生したり、クラスタから削除された場合は、cxfs7 (ランク 1) が使用されます。また、クラスタ内のすべてのノードにデフォルトでマウントされるが、cxfs8 には明示的にマウントされないファイルシステム (Default Local Status enabled) も表示されます。

   メモ: 特定のファイルシステムのメタデータ・サーバのリストは順序付きですが、ネットワーク・レーテンシやその他の予測不能な遅れがあるため、起動サイクル中にどのノードが実際にサーバになるかは予想できません。

   以下を実行します。

```
cmgr> define cxfs_filesystem fs1 in cluster cxfs6-8

(Enter "cancel" at any time to abort)

Device ? /dev/cxvm/d76lun0s0
Mount Point ? /mnts/fs1
Mount Options[optional] ?
Use Forced Unmount ? <true|false> ? false
Default Local Status <enabled|disabled> ? (enabled)

DEFINE CXFS FILESYSTEM OPTIONS

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
```

```
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1

No current servers

Server Node ? cxfs6
Server Rank ? 0

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1
Server Node ? cxfs7
Server Rank ? 1


        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:5

No disabled clients

Disabled Node ? cxfs8
```

```
        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:7

Current settings for filesystem (fs1)

CXFS servers:
        Rank 0          Node cxfs6
        Rank 1          Node cxfs7

Default local status: enabled

No explicitly enabled clients

Explicitly disabled clients:
        Disabled Node: cxfs8

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:9
Successfully defined cxfs_filesystem fs1

cmgr> define cxfs_filesystem fs2 in cluster cxfs6-8
```

```
(Enter "cancel" at any time to abort)

Device ? /dev/cxvm/d77lun0s0
Mount Point ? /mnts/fs2
Mount Options[optional] ?
Use Forced Unmount ? <true|false> ? false
Default Local Status <enabled|disabled> ? (enabled)

DEFINE CXFS FILESYSTEM OPTIONS

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1

Server Node ? cxfs8
Server Rank ? 0

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:7

Current settings for filesystem (fs2)
```

```
CXFS servers:
        Rank 0          Node cxfs8

Default local status: enabled

No explicitly enabled clients

No explicitly disabled clients

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:9
Successfully defined cxfs_filesystem fs2
```

クラスタ cxfs6-8 の変更内容を表示するには、次を入力します。

```
cmgr> show cxfs_filesystems in cluster cxfs6-8

fs1
fs2
```

9. cmgr を終了するには、次を入力します。

```
cmgr> quit
```

CXFS の GUI タスク・リファレンス

この章では、CXFS マネージャ・グラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) について説明します。また、CXFS タスクについての詳細と、XVM タスクの概要も説明します。XVM タスクについての詳細は、『XVM Volume Manager Administrator's Guide』を参照してください。

この章は、以下の節で構成されています。

- 「GUI の概要」
- 106 ページの「設定ガイド・タスク」
- 108 ページの「GUI を使ったノード・タスク」
- 122 ページの「GUI を使ったクラスタ・タスク」
- 125 ページの「GUI を使ったクラスタ・サービス・タスク」
- 129 ページの「GUI を使ったスイッチと I/O 制限タスク」
- 132 ページの「GUI を使ったファイルシステム・タスク」

メモ: CXFS を正しく実行するには、各ノードにライセンスがインストールされていなければなりません。ライセンスを正しくインストールせずにソフトウェアをインストールすると、エラーが発生し、CXFS マネージャ GUI を使用することはできません。ライセンスについての詳細は、43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。IRIX 以外のオペレーティング・システムが実行されているノードのライセンスについての詳細は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

GUI の概要

GUI では、CXFS ファイルシステムと XVM 論理ボリュームの設定および管理を行うことができます。また、ステータスや構造を表すアイコンを表示することもできます。

この節では、GUI の概要を説明します。

- 「GUI の起動」
- 92 ページの「GUI ウィンドウ」

- 94 ページの「GUI の機能」
- 103 ページの「アイコンと状態の説明」

GUI の起動

GUI を起動するには、以下のいずれかの方法を使用します。

- CXFS GUI クライアント・ソフトウェア (`sysadm_cxfs.sw.client`) とデスクトップ・サポート・ソフトウェア (`sysadm_cxfs.sw.desktop`) がインストールされている IRIX システムで、次のコマンド・ラインを入力します。

  ```
  # /usr/sbin/cxfsmgr
  ```

  クラスタを定義する前に `cxfsmgr` を呼出すと、エラー・メッセージが表示されます。クラスタの定義中の場合、このエラー・メッセージは無視してかまいません。

  `cxdetail` コマンドおよび `cxtask` コマンドは、`cxfsmgr` と同じ機能を実行します。これらのコマンドの名前は、従来の操作性を維持するために保持されています。

- CXFS GUI クライアント・ソフトウェア (`sysadm_cxfs.sw.client`) とデスクトップ・サポート・ソフトウェア (`sysadm_cxfs.sw.desktop`) がインストールされている IRIX システムで、Toolchest から以下を選択します。

  「システム(System)」
  　>「CXFS マネージャ(CXFS Manager)」

  Toolchest に CXFS のエントリを表示するには、CXFS のインストール後に Toolchest を再起動する必要があります。Toolchest を再起動するには、以下のコマンドを入力します。

  ```
  # killall toolchest
  # /usr/bin/X11/toolchest &
  ```

  ホストにログインするよう求めるダイアログ・ボックスが表示されます。

- Microsoft Windows、Linux、またはその他のプラットフォーム上の Web ブラウザで、`http://`*`server`*`/CXFSManager/` と入力し（*server* はプール内の CXFS 管理ノードの名前）、<Enter> を押します。表示される Web ページで CXFS マネージャのアイコンをクリックします。

  この方法で CXFS マネージャを起動できるのは、Java プラグインをインストールしてすべての Java プロセスを終了してから、ブラウザを再起動して Java を有効にした場合だけです。このアイコンが表示されるまでに長い時間がかかる場合は、「プラグインなし(non plug-in)」リンクをクリックすることができますが、ブラウザ固有の Java で実行することにより、操作上の不具合が発生する可能性があります。

この方法で GUI を起動できるのは、IRIX 以外のオペレーティング・システムから GUI を実行する場合です。IRIX システムで GUI を実行している場合は、Toolchest または `cxfsmgr` コマンドを使用する方法が推奨されています。

表4-1 に、GUI を起動、接続、および表示できるプラットフォームについて説明します。

表4-1 GUI プラットフォーム

GUI モード	GUI を起動する場所	GUI を接続する場所	GUI が表示される場所
`cxfsmgr`(1) または Toolchest	`sysadm_cxfs.sw.client` および `sysadm_cxfs.sw.desktop` ソフトウェアがインストールされている IRIX システム（SGI 2000 シリーズや SGI O2 ワークステーションなど）。ただし、CXFS では CXFS ノードとして Silicon Graphics O2 ワークステーションはサポートされていないため、CXFS シリアル・ハードウェア・リセット・サーバになることはできないことに注意してください。	クラスタ管理に使用する、プール内の CXFS 管理ノード。	SGI O2 ワークステーション
Web	Web ブラウザと Java 1.1 Plug-in がインストールされ有効になっているシステム。	クラスタ管理に使用する、プール内の CXFS 管理ノード。	Web ブラウザがインストールされているのと同じシステム。

クラスタの表示は、どの CXFS 管理ノードに GUI を接続するかによって変わります。変更を加えたときは、次の変更を行う前に、表示領域に変更内容が表示されるまで待ってください。表示領域に変更内容が表示されるまでは、その変更がクラスタ全体に伝わったことは保証されていません。表示領域の場所については、93 ページの図4-1を参照してください。クラスタ・データベースに変更を加えると、その都度、すべての CXFS 管理ノードにクラスタ全体のステータス情報が送信されます。

すべてのタスクを実行するために必要な特権を持つことができるよう、GUI には `root` としてログインする必要があります。ただし、IRIX Interactive Desktop System Administration (`sysadmdesktop`) 製品の一部である特権マネージャを使用すると、システム管理者が一部またはすべての特権をその他の任意のユーザに許可できます。詳細については、『Personal System Administration Guide』を参照してください。

メモ: 変更は、特定の時点で実行されている 1 つの GUI プロセスだけから行ってください。2 つ目の GUI プロセス(`cxfsmgr` の 2 つ目の呼出し)から変更を行うと、最初のインスタンスで加えた変更が上書きされる場合があります。ただし、「ファイル(File)」メニューからアクセスした複数の「CXFS マネージャ(CXFS Manager)」ウィンドウは、すべて同じアプリケーション・プロセスの一部なので、これらのどのウィンドウからでも変更を加えることができます。

GUI ウィンドウ

図4-1 に、「CXFS マネージャ(CXFS Manager)」ウィンドウを示します。また、図4-2 に、詳細領域の特定のコンポーネントの詳細を示します。表示領域を使用したステータスのモニタの詳細、およびアイコンとカラーの説明については、238 ページの「クラスタのステータス」を参照してください。

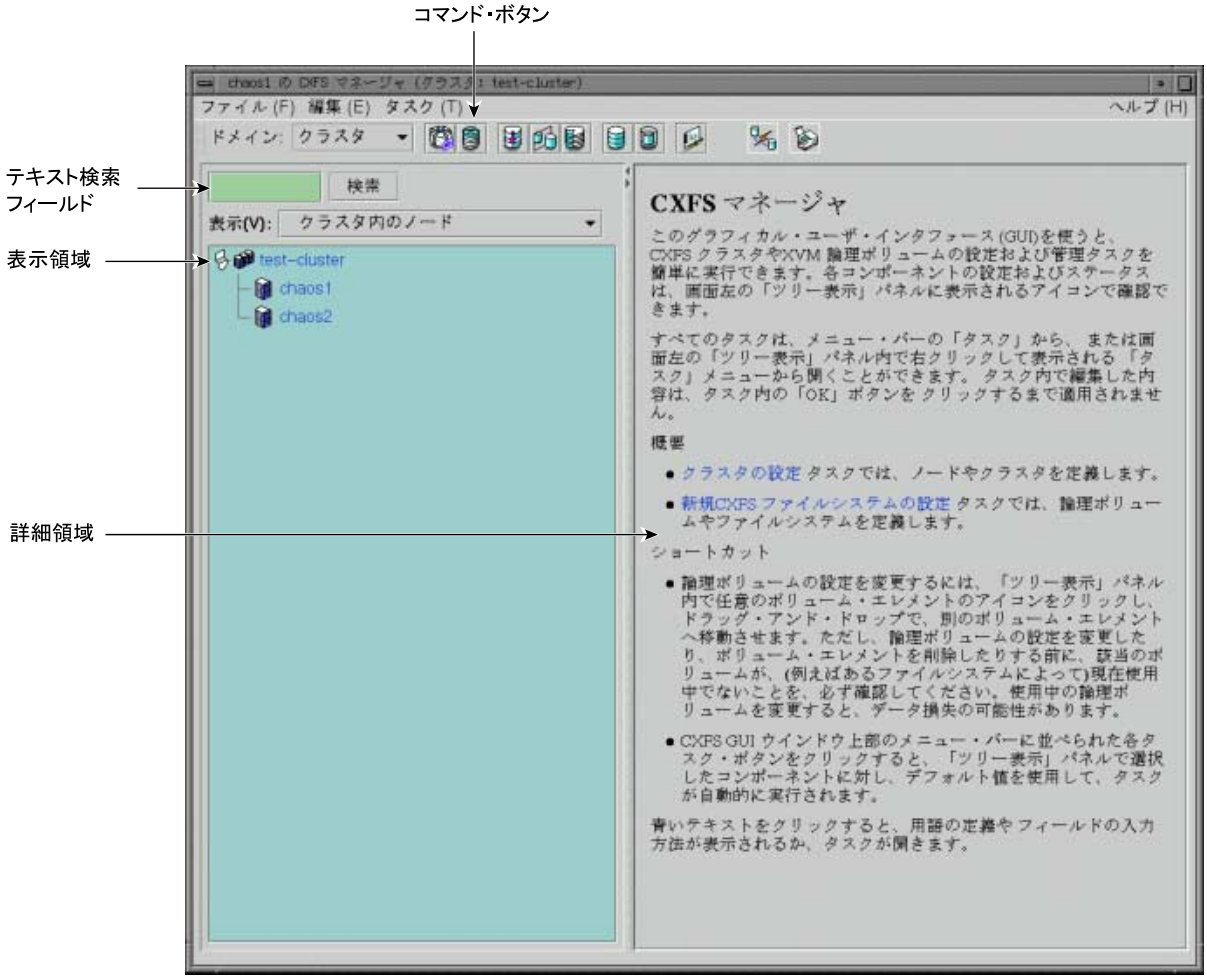


図4-1 CXFS マネージャ GUI ウィンドウ

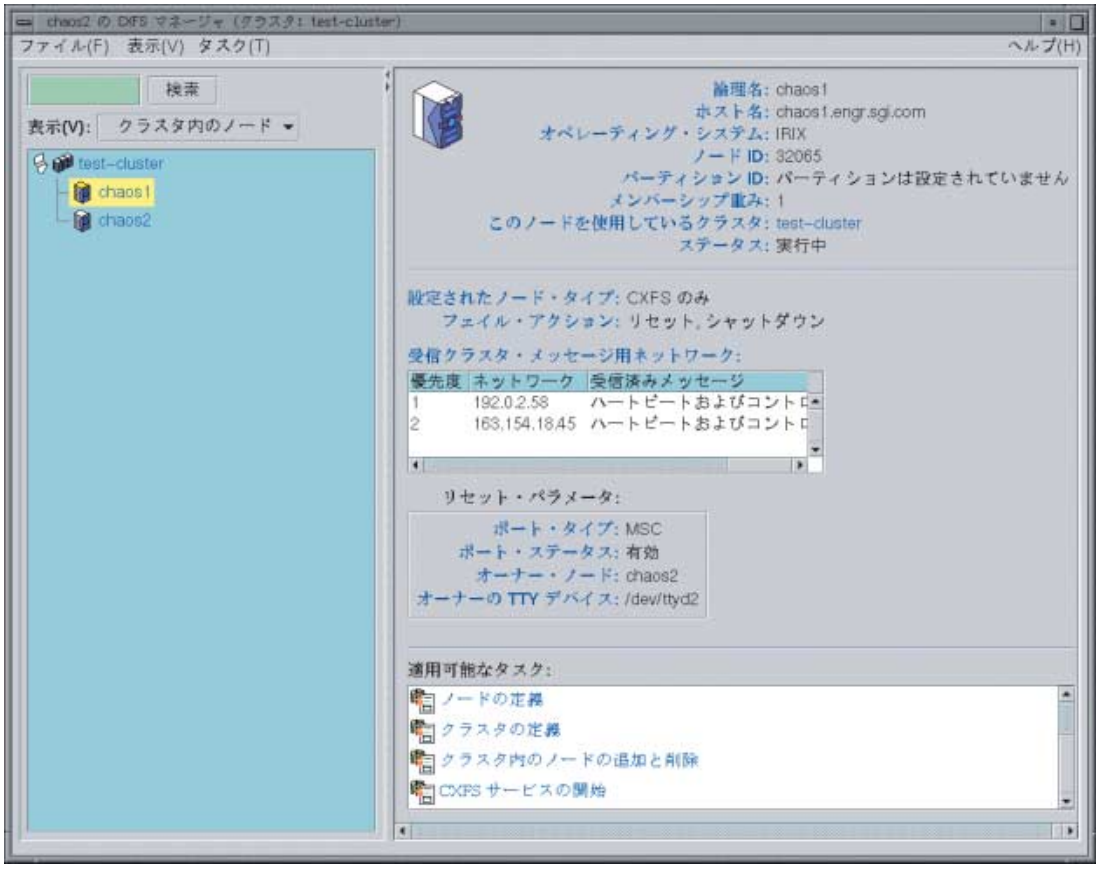


図4-2 ノードの詳細が表示されている CXFS マネージャ GUI

GUI の機能

CXFS マネージャ GUI を使用すると、単一のポイントから CXFS クラスタ全体を管理できます。これは、CXFS クラスタの設定と管理を行いやすくするツールへのアクセスを提供します。

- タスクを使用して、CXFS クラスタの個々のコンポーネントを設定およびモニタできます。
- 設定ガイド・タスクは、より大きな目的を達成するために一まとめにされたタスクのグループで構成されます。たとえば、「新規クラスタの設定(Set Up a New Cluster)」を使用すると、新規クラスタの作成手順を順番に実行でき、タイトルをクリックするだけで必要な個々のタスクを起動できます。

この節では、以下の内容について説明します。

- 「GUI ウィンドウのレイアウト」
- 「「ファイル」メニュー」
- 95 ページの「「編集」メニュー」
- 96 ページの「「タスク」メニュー」
- 96 ページの「「ヘルプ」メニュー」
- 97 ページの「「ドメイン」メニュー」
- 97 ページの「コマンド・ボタンを使ったショートカット」
- 99 ページの「「表示」メニュー」
- 100 ページの「タスクの実行」
- 100 ページの「ドラッグアンドドロップを使った XVM 設定」
- 101 ページの「PCP を使った I/O パフォーマンスの解析」
- 101 ページの「ボリューム・トポロジーの構築」
- 102 ページの「ディスクの設定」
- 102 ページの「詳細の表示」
- 102 ページの「GUI と xvm(1M) コマンドの重要な相違点」

GUI ウィンドウのレイアウト

デフォルトでは、ウィンドウは表示領域と詳細領域の 2 つのセクションに分かれています。ウィンドウの中央にある矢印を使用して画面を切り替えることができます。

「ファイル」メニュー

「ファイル(File)」メニューでは、GUI の現在のインスタンスの複数のウィンドウ、/var/adm/SYSLOG システム・ログ・ファイル、および /var/sysadm/salog システム管理ログ・ファイル(GUI からアクセスしたコマンドが示されている)を表示できます。また、現在のウィンドウを閉じたり、GUI を完全に終了することもできます。

「編集」メニュー

「編集(Edit)」メニューでは、表示領域の内容を展開および縮小できます。また、自動的に画面を展開して、プールまたはクラスタに追加された新しいノードを反映するよう選択することもできます。さらに、この

メニューを使用して、表示領域のすべてのアイテムを選択したり、現在選択されているアイテムをクリアすることもできます。

「タスク」メニュー

「タスク(Tasks)」メニューには以下が含まれます。

- 「設定ガイド(Guided Configuration)」: クラスタの設定、ファイルシステムの定義、ボリュームの作成、ステータスのチェック、および既存のクラスタの変更を行うためのタスクが含まれます。
- 「ノード(Nodes)」: ノードを定義および管理するためのタスクが含まれます。
- 「クラスタ(Cluster)」: クラスタを定義および管理するためのタスクが含まれます。
- 「クラスタ・サービス(Cluster Services)」: CXFS サービスの開始と停止、CXFS タイブレーカー・ノードの設定、ログの設定、およびローカル・ノードの CXFS カーネル・メンバーシップの許可と取消しを行うことができます。
- 「スイッチと I/O 制限(Switches and I/O Fencing)」: ファイバ・チャネル・スイッチの定義の設定および I/O 制限の管理を行うためのタスクが含まれます。
- 「ディスク(Disks)」: XVM ディスク管理タスクが含まれます。
- 「ボリューム・エレメント(Volume Elements)」: XVM ボリューム・エレメントを作成、削除、変更、および管理するためのタスクが含まれます。
- 「ファイルシステム(Filesystems)」: ファイルシステムの定義と管理、およびメタデータ・サーバの再設定を行うためのタスクが含まれます。

  メモ: 再設定機能は完全に実装されていますが、関連する問題が多数存在するため、CXFS ではサポートされていません。これらの機能を使用してもデータの整合性が損なわれることはありませんが、クラスタ・ノードのパニックやハングが発生する可能性があります。この機能は、このような問題が解決された時点で完全にサポートされる予定です。再設定の機能は、このリリースではデフォルトで無効になっています。

- 「タスクの検索(Find Tasks)」: キーワードを使用して特定のタスクを検索できます。

「ヘルプ」メニュー

「ヘルプ(Help)」メニューには、GUI の概要と、アイコンへのキーが表示されます。青いテキストをクリックすることによって、その特定のアイテムのヘルプを表示することもできます。

「ドメイン」メニュー

「ドメイン(Domain)」メニューは、このセッションで作成する XVM ボリュームをローカル・ドメインまたはクラスタ・ドメインのどちらに存在させるかを選択する場合に使用します。ドメインについての詳細は、『XVM Volume Manager Administrator's Guide』を参照してください。

コマンド・ボタンを使ったショートカット

GUI ウィンドウの上部に沿って配置されているコマンド・ボタンを使用すると、タスクをすばやく実行できます。ボタンをクリックすると、対応するタスクがデフォルト値を使用して実行されます。通常、タスク・ウィンドウは表示されません。デフォルトをオーバーライドするには、「タスク(Tasks)」メニューからタスクを起動します。表4-2 に、使用可能なショートカットの概要と、これらのタスクについての詳細を示します。『XVM Volume Manager Administrator's Guide』を参照してください。

表4-2 コマンド・ボタン

ボタン	タスク
	ラベルのない特定のディスクに、ラベルを付けます。選択したディスクに外部ディスクまたはラベル付きのディスク、あるいはその両方が含まれている場合、「ディスクのラベル付け(Label Disks)」タスクが実行されます。
	デフォルトの入力値として選択されたディスクで「ディスクのスライス(Slice Disks)」タスクを実行します。
	一時名を持つ結合を作成します。
	一時名を持つミラーを作成します。

ボタン	タスク
	一時名を持つストライプを作成します。
	一時名を持つサブボリュームを作成します。
	一時名を持つボリュームを作成します。
	PCP (Performance Co-Pilot) XVM I/O モニタの pmgxvm(1) をサーバで開始し、X Windows を通じてローカル管理ステーションに表示します。
	選択されたボリューム・エレメントを現在の親から分離します。
	スライス以外の選択されたボリューム・エレメントを削除したり、選択されたディスクから直接ラベルを削除したり、選択されたコンポーネントに適切な削除タスクを実行します。

「表示」メニュー

表示したい内容は、「表示(View)」メニューから選択します。

- クラスタ内のノード
- プール内のノード(つまり、定義されているすべてのノード)
- ファイルシステム
- クラスタ・ボリューム・エレメント
- ローカル・ボリューム・エレメント
- ディスク
- Brocade ファイバ・チャネル・スイッチ

表示または変更するアイテムの選択

アイテムを選択するには、以下の方法を使用できます。

- 一度に 1 つのアイテムを選択するには、単にクリックします。
- アイテムのブロックを選択するには、<Shift> キーを押しながらクリックします。
- 任意のアイテムの選択を切替えるには、<Ctrl> キーを押しながらクリックします。

また、「検索(Find)」テキスト・フィールドに名前を入力した後、<Enter> キーを押すか「検索(Find)」ボタンをクリックして、1 つまたは複数のアイテムを選択することもできます。

コンポーネントの詳細の表示

コンポーネントの詳細を表示するには、表示領域でその名前をクリックします。99 ページの「表示または変更するアイテムの選択」を参照してください。

コンポーネントの設定とステータスの詳細は、右側の詳細領域に表示されます。詳細領域の下部には、「適用可能なタスク(Applicable Tasks)」リストが現れます。このリストには、コンポーネントの設定の詳細を評価した後で起動できるタスクが表示されます。タスクを起動するには、タスク名をクリックします。選択したコンポーネントに基づいて、デフォルト値がタスク・ウィンドウに表示されます。

詳細領域でアイテムの詳細を参照するには、その名前を選択します。すると、選択した名前は青色で表示され、新しいウィンドウに詳細が表示されます。用語集に定義されている用語も、青色で表示されます。

タスクの実行

個々のタスクを実行するには、以下の手順に従います。

1. 「タスク(Tasks)」メニューからタスク名を選択するか、または表示領域でマウスの右ボタンをクリックします。たとえば、以下のように選択します。

   「タスク(Tasks)」
   >「設定ガイド(Guided Configuration)」
   >「新規クラスタの設定(Set Up a New Cluster)」

   タスク・ウィンドウが表示されます。

   メモ: 青いテキストをクリックすると、その概念や入力フィールドに関する詳細を参照できます。

   ---

2. 該当するフィールドに情報を入力して「OK」をクリックし、タスクを完了します。タスクの中には複数のページで構成されるものもあります。このような場合は、「次へ(Next)」をクリックして次のページに移動し、そのページで情報を入力してから「OK」をクリックします。

   メモ: いずれのタスクでも、「OK」をクリックするまでクラスタ設定は更新されません。

   タスクが正常に完了したことを確認するダイアログ・ボックスが表示されます。

3. 必要に応じて、タスクの起動を続行します。

ドラッグアンドドロップを使った XVM 設定

GUI を使用すると、ドラッグアンドドロップを使用して、ボリューム・トポロジーを構築し、XVM ディスクを管理できます。

注意: ドラッグアンドドロップを使用してボリューム・エレメントを再構築する際は、常に注意してください。これは、ドラッグアンドドロップを使用することによって、ボリューム・エレメントに存在するデータが失われる可能性があるためです。データを失う可能性が高いことが予測された場合、GUI によってユーザに警告が表示されます。ただし、マウントされているファイルシステムにボリュームが関連付けられていない場合は、そのボリュームが重要なデータを保持できるかどうかについて、`xvm`(1M) コマンドまたは GUI のいずれによっても判断されません。

---

2 つの GUI ウィンドウ間では、ドラッグアンドドロップすることはできません。また、CXFS マネージャと IRIX Interactive Desktop Personal System Administration ウィンドウ間でのドラッグアンドドロップも不可能です。さらに、ショートカット・コマンド・ボタンにドラッグアンドドロップすることもできません。

ドラッグアンドドロップを使用したボリューム・トポロジーの構築とディスクの設定についての詳細は、『XVM Volume Manager Administrator's Guide』を参照してください。

PCP を使った I/O パフォーマンスの解析

パフォーマンスを解析するには、PCP を起動するボタンをクリックします。97 ページの「コマンド・ボタンを使ったショートカット」を参照してください。結果として表示される PCP ウィンドウには、読取りと書込みの I/O アクティビティを示す色の付いた LED と共に、すべてのボリュームが表示されます。任意の LED の上にカーソルを置き、スペースバーを押すと、LED の値/カラーの凡例と、対応する XVM ボリュームまたはボリューム・エレメントの読取りレートや書込みレートの現在の値を示すウィンドウが表示されます。任意の LED でマウスの中央ボタンをクリックすると、XVM の読取りと書込みの I/O アクティビティ・グラフとディスク・アクティビティの 3D グラフィカル表示を表示するその他のツールを起動できる場所からメニューを取得できます。

ボリューム・トポロジーの構築

論理ボリュームを再設定するには、以下を実行します。

- 希望の表示を選択します。

  「表示(View)」
  >「クラスタ・ボリューム・エレメント(Cluster Volume Elements)」

  または、

  「表示(View)」
  >「ローカル・ボリューム・エレメント(Local Volume Elements)」

- ボリューム・エレメントのアイコンを選択します。
- 選択したアイコンを別のボリューム・エレメントのアイコンにドラッグアンドドロップします。

アイコンは、ドロップ先として有効な場合、ドラッグする際に青色になります。ドラッグするときにマウスのカーソルが表示領域の上部または下部に到達すると、画面は自動的にスクロールされます。

異なるタイプの複数のボリューム・エレメントで操作するには、ドラッグアンドドロップを使用できます。たとえば、アイテムを選択して任意の「アタッチされていない(Unattached)」見出しにドラッグすることにより、このカテゴリに属するアイテムが選択されていない場合でも、ボリューム・エレメントのいくつかのタイプを分離できます。また、異なるタイプの複数のアイテムを選択し、親に接続することができます。たとえば、2 つの結合と 1 つのストライプを選択した後でドラッグアンドドロップを使用し、親結合に接続することが可能です。

選択されている(強調表示されている)ボリューム・エレメントをクリックし、テキスト・フィールドに新しい名前を入力することによって、ボリューム・エレメントの名前は変更することができます。

ディスクの設定

ドラッグアンドドロップを使用してディスクにラベルを付けたり、ディスクからラベルを削除するには、以下を選択します。

「表示(View)」
>「ディスク(Disks)」

その後、ラベルのないディスクを選択した後で「ラベル付きのディスク(Labeled Disks)」見出しにドラッグアンドドロップするか、ラベル付きのディスクを選択した後で「ラベルのないディスク(Unlabeled Disks)見出しにドラッグアンドドロップします。

「ドメイン(Domain)」フィールドが「クラスタ(Cluster)」に設定されている場合、ドラッグアンドドロップを使用してディスクの所有権を譲渡できます。これを行うには、「ディスク(Disks)」表示でディスクを選択した後、「クラスタ・ディスク(Cluster Disks)」見出しにドラッグアンドドロップします。

メモ: データを失う危険性は、ディスクの所有権の譲渡の方がディスクの奪取よりも低いと言えます。

選択されている(強調表示されている)ディスクをクリックし、結果として表示される名前テキスト・フィールドに名前を入力することによって、ディスクにはラベルを付けることができます。

詳細の表示

用語定義、入力する内容の詳細、またはアイテム設定詳細を表示したり、タスクを起動するには、青いテキストをクリックします。

一般的に、青いテキストをクリックすると、以下のいずれかが表示されます。

- 用語定義
- 入力詳細
- アイテム詳細
- 選択されたタスク・ウィンドウ

GUI と `xvm`(1M) コマンドの重要な相違点

ボリューム以外のボリューム・エレメントが作成または分離された場合、そのボリューム・エレメントに関連付けられたボリュームとサブボリュームが自動的に作成されます。生成されたこのボリュームには、明示的に名前を付けることができます。ボリューム名はラベル領域に格納され、マシンを再起動しても変わらなくなります。

GUI では、明示的に名前が付けられていないボリュームとサブボリュームは表示されません。また、このようなボリュームとサブボリュームの子は、「使用可能」または「アタッチされていない」として GUI に表示されます。それとは対照的に、xvm(1M) コマンドでは、すべてのボリュームとサブボリュームが表示されます。

GUI では、明示的に名前が付けられていないボリュームにあるファイルシステムは表示されますが、このようなボリュームは「なし(None)」としてリストされます。システムによって自動的に生成された、一時名を持つボリュームとサブボリュームは、アタッチされていないボリューム・エレメントのフル・パスで示されますが（例: /vol96/datav）、そうでない場合は、GUI で無視されます。

データを失う危険性を低くするには、GUI を使用する際にボリュームに明示的に名前を付けることが推奨されます。明示的に名前を付けなかったボリュームを xvm(1M) コマンドを使用して作成した場合は、次に進む前に、xvm ツールを使用して、ボリュームに永続的な名前を割当てることができます。このようにすると、データを失う危険性を低くすることができます。

アイコンと状態の説明

以下の表に、CXFS マネージャ GUI で使用されるアイコンと状態を説明します。

表4-3 アイコンの説明

アイコン	実体
	ノード
	クラスタ
	表示領域で展開されているツリー
	表示領域で縮小されているツリー

アイコン	実体
	スイッチ
	XVM ディスク
	ラベルのないディスク
	外部ディスク
	スライス
	ボリューム
	サブボリューム
	結合
	ミラー

アイコン	実体
	ストライプ
	スロット
	ローカル・ファイルシステム
	CXFS ファイルシステム
	書込みコピー
	リポジトリ
	スナップショット

表4-4 状態の説明

アイコン	状態
	(グレーのアイコン)非アクティブ、不明、またはオフライン(CXFS サービスがアクティブでない可能性があります)
	(青いアイコン)マウントが有効(CXFS サービスがアクティブでない可能性があります)
	(青いアイコン)オンライン、使用準備済み、実行中、またはマウント済み(エラーなし)
	(緑色のスワッチ)開いている、使用中
	(オレンジ色の矢印)ミラーの再生中
	(赤いアイコン)エラーが検出されている、停止、またはマウント済み(エラーあり)

設定ガイド・タスク

この節では、以下の設定ガイド・タスクについて説明します。

- 107 ページの「GUI を使って CXFS 用に既存の FailSafe クラスタを再設定」
- 108 ページの「既存のクラスタの変更」
- 108 ページの「クラスタ・ノードの修正またはアップグレード」

74 ページの「GUI を使った新規クラスタの設定」、76 ページの「GUI を使った新規 CXFS ファイルシステムの設定」、および 238 ページの「GUI を使ったクラスタのステータスの確認」も参照してください。XVM 設定ガイド・タスクについての詳細は、『XVM Volume Manager Administrator's Guide』を参照してください。

GUI を使って CXFS 用に既存の FailSafe クラスタを再設定

メモ: CXFS タスク内で青いテキストをクリックすると、その概念や入力フィールドに関する詳細を参照できます。いずれのタスクでも、「OK」をクリックするまでクラスタ設定は更新されません。

「CXFS 用に既存の FailSafe クラスタを再設定(Set Up an Existing FailSafe Cluster for use with CXFS)」タスクを使用すると、既存の IRIS FailSafe ノードおよびクラスタを CXFS に切替えるために必要な手順を実行できます。このタスクには、ほかの箇所で詳しく説明されているタスクが含まれます。FailSafe もインストールされている場合のみ、このタスクは CXFS GUI に表示されます。

FailSafe および CXFS のデータベースは共通のものが 1 つだけ存在します。特定のノードが両方の製品に適用される場合は、変更内容が両方の製品に対して適切であることを確認してください。

以下を実行します。

1. 「FailSafe クラスタの CXFS 用への切替え(Convert a FailSafe Cluster for use with CXFS)」をクリックします。これにより、クラスタ・タイプは `CXFS and FailSafe` に変わります。123 ページの「GUI を使った FailSafe クラスタの CXFS 用への切替え」を参照してください。
2. FailSafe GUI を使用して、切替えるノードで高可用性 (HA: High Availability) サービスを停止します。『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。
3. CXFS GUI を使用して、2 番目のハートビートおよびコントロール・ネットワークをノード定義に追加します。118 ページの「GUI を使ったノード定義の変更」を参照してください。
4. 「FailSafe ノードの CXFS 用への切替え(Convert a FailSafe Node for use with CXFS)」をクリックして、ローカル・ノード(接続しているノード)に切替えます。切替えたノードのタイプは、`CXFS and FailSafe` または `CXFS` になります。120 ページの「GUI を使った FailSafe ノードの CXFS 用への切替え」を参照してください。
5. 「FailSafe ノードの CXFS 用への切替え(Convert a FailSafe Node for use with CXFS)」をクリックして、別のノードに切替えます。切替えたいノードに対して、この手順を繰返します。
6. 「CXFS サービスの開始(Start CXFS Services)」をクリックします。

既存のクラスタの変更

このタスクでは、既存のクラスタを編集するさまざまな方法を説明します。変更(クラスタ管理者にイベントを通知する方法の変更など)は、CXFS サービスがアクティブなときでも実行できます。ただし、接続性をテストするときは、まずクラスタ・サービスを停止してください。ファイルシステムは、変更を加える前にアンマウントする必要があります。

以下を参照してください。

- 123 ページの「GUI を使ったクラスタ定義の変更」
- 76 ページの「GUI を使った新規 CXFS ファイルシステムの設定」
- 136 ページの「GUI を使った CXFS ファイルシステムの変更」
- 109 ページの「GUI を使ったノードの定義」
- 121 ページの「GUI を使ったノードの接続性のテスト」
- 117 ページの「GUI を使ったクラスタ内のノードの追加と削除」

クラスタ・ノードの修正またはアップグレード

このタスクを使用すると、クラスタからノードを削除するために必要な手順を実行できます。このタスクは、以下の手順で構成されています。

- 125 ページの「GUI を使った CXFS サービスの停止(通常 CXFS シャットダウン)」
- ノードで必要なメインテナンスを実行します。これは、必要な場合にのみ行ってください(117 ページの「GUI を使ったノードのリセット」を参照してください)。
- 125 ページの「GUI を使った CXFS サービスの開始」
- 表示領域で、クラスタ・コンポーネントの状態をモニタします。238 ページの「GUI を使ったクラスタのステータスの確認」を参照してください。

GUI を使ったノード・タスク

この節では、GUI を使用してノードを定義、変更、削除、表示、およびリセットする方法について説明します。

メモ:「新規クラスタの設定(Set Up a New Cluster)」設定ガイド・タスクを使用すると、クラスタおよびノードを設定するために必要なタスクを実行できます。74 ページの「GUI を使った新規クラスタの設定」を参照してください。

GUI を使ったノードの定義

定義する最初のノードは、クラスタ管理を実行するために、GUI を接続した CXFS 管理ノードでなければなりません。定義する最初のノードに対しては、/etc/sys_id に示されているホスト名を使用しなければなりません。

メモ: CXFS タスク内で青いテキストをクリックすると、その概念や入力フィールドに関する詳細を参照できます。いずれのタスクでも、「OK」をクリックするまでクラスタ設定は更新されません。

ノードを定義するには、以下を実行します。

1. 「ホスト名(Hostname)」: mynode.company.com(すべてのノードで解決される場合は、mynode に省略できます)など、定義するノードのホスト名を入力します。プール内で定義する最初のノードに対しては、/etc/sys_id に示されているホスト名を使用する必要があります。また、このノードは CXFS 管理ノードでなければなりません。これは、最初のノードに対してデフォルトで入力されます。/etc/sys_id ファイルに示されているとおりにホスト名を表示するには、hostname(1) コマンドを使用します。51 ページの「ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf」を参照してください。

2. 「論理名(Logical Name)」: ホスト名と同じ名前、ホスト名の簡略名(lilly など)、またはまったく別の名前(nodeA など)を入力します。論理名は、アンダースコア (_) で始めたり、空白として使用される文字を含めたり、255 文字より長くすることはできません。

   メモ: ノードの名前を変更するには、そのノードを削除した後、新しいノードを定義しなければなりません。

3. 「オペレーティング・システム(Operating System)」: 定義するノードで実行されているオペレーティング・システムの名前を選択します。Windows NT または Windows 2000 に対しては、「Windows」を選択します。

   IRIX ノードには、選択したノードの機能およびインストールしたソフトウェアに応じて、サーバとして使用可能な管理ノード、クライアント管理ノード、または CXFS クライアントのみのノードを指定できます。Solaris および Windows ノードは、常に CXFS クライアントのみのノードです。

IRIX ノードはリセットすることができます。IRIX を選択した場合は、2 ページ目でシリアル・ハードウェア・リセット情報を入力できます。

定義済みのノードのオペレーティング・システムを後から変更することはできません。オペレーティング・システムを変更するには、そのノードを削除して、新しい名前を持つ新しいノードを定義する必要があります。

4. 「ノードの機能(Node Function)」: IRIX ノードに対して、以下のいずれかを選択します。

   - 「サーバとして使用可能な管理(Server-capable Admin)」: メタデータ・サーバとして使用することを希望するノードです。このノードは、`cluster_admin` 製品をインストールして作成される必要もあります(この結果、ノードでは、クラスタ・データベースを含み、クラスタ管理タスクを実行できます)。実際にアクティブなメタデータ・サーバになるには、特定のファイルシステムのメタデータ・サーバとして使用可能なノードのリストからこのノードを選択しなければなりません。このタイプのノードは、CXFS カーネル・メンバーシップ定足数およびクラスタ・データベース定足数を形成する際にカウントされます。

   - 「クライアント管理(Client Admin)」: `cluster_admin` 製品がインストールされ、その結果、クラスタ・データベースを含み、クラスタ管理タスクを実行できるノードです。このタイプのノードは、クラスタ・データベース定足数を形成する際にカウントされますが、CXFS カーネル・メンバーシップ定足数を形成する際にはカウントされません。

   - 「クライアントのみ(Client-Only)」: `cxfs_client` ソフトウェア製品がインストールされるノードです。クライアントのみのノードは、CXFS メタデータ・サーバになったり、クラスタ管理を実行することはできず、メンバーシップ定足数を形成する際にカウントされません。

   Solaris および Windows ノードは、自動的に「クライアントのみ(Client-Only)」として選択されます。

5. 「受信クラスタ・メッセージ用ネットワーク(Networks for Incoming Cluster Messages)」: 以下を実行します。

   - 「ネットワーク(Network)」: プライベート・ネットワークの IP アドレスまたはホスト名を入力します。ホスト名は、 `/etc/hosts` ファイルで解決されなければなりません。ネットワークの優先度は、クラスタ内の各ノードに対して同じでなければなりません。ホスト名の使用についての詳細は、51 ページの「ホスト名解決: `/etc/sys_id`、`/etc/hosts`、`/etc/nsswitch.conf`」を参照してください。プライベート・ネットワークが必要である理由については、16 ページの「プライベート・ネットワーク」を参照してください。

     GUI では複数のネットワークを入力できますが、CXFS が使用するのは優先度が 1 のネットワークだけで、このネットワークはフェイルオーバーされません。FailSafe にも適用されるようクラスタを切替える場合以外(FailSafe では 2 つのネットワークが必要)は、追加のネットワークを定義しないでください。

   - 「受信メッセージ(Messages to Accept)」:「ハートビートおよびコントロール(Heartbeat and Control)」を選択します。

メモ: ネットワークを一時的に定義して、ネットワークでメッセージを受信しないようにする場合は、「なし(None)」設定を使用できます。詳細については、8 ページの「クラスタ環境」を参照してください。

- ネットワークをリストに追加するには、「追加(Add)」をクリックします。

  後でこのネットワークを変更する場合は、リストでそのネットワークをクリックして選択し、「変更(Modify)」をクリックします。

  リストからネットワークを削除するには、リストでそのネットワークをクリックして選択し、「削除(Delete)」をクリックします。

6. 「ノード ID(Node ID)」: (IRIX ではオプション) プール内のノード全体で一意な、1〜32767 の範囲の整数。IRIX ノードに対して数字を指定しないと、CXFS によって ID が計算されます。

   IRIX ノードでは、デフォルトの ID は、マシンのシリアル・ナンバーなどのマシン固有の情報を基にした 5 桁の数字で、連続する数字ではありません。IRIX 以外のオペレーティング・システムが実行されているノードでは、ノード ID を指定する必要があります。

   ノードを定義した後にノード ID 番号を変更しないでください。デフォルトの CXFS タイブレーカー・ノードはありません。詳細については、28 ページの「メンバーシップ定足数」を参照してください。

7. 「パーティション ID(Partition ID)」: (オプション) パーティションが設定された Origin 3000 システムのパーティションを一意に定義します。システムにパーティションが設定されていない場合、このフィールドは空のままにします。パーティション ID の値を決定するには、`mkpart`(1M) コマンドを使用します (以下を参照してください)。

   「次へ(Next)」をクリックして、次の画面に移動します。

8. 「異常時のアクション(Fail Action)」: 異常ノードの処理方法を決定するアクションのセット。2 番目のアクションは最初のアクションが失敗した場合にのみ実行され、3 番目のアクションは、最初のアクションと 2 番目のアクションが失敗した場合にのみ実行されます。

   使用可能なアクションは、以下のように、ノードのオペレーティング・システムに応じて決まります。

   - 「制限(Fence)」: 問題のあるノードからストレージ・エリア・ネットワーク (SAN: Storage Area Network) へのアクセスを無効にします。このアクションは、すべてのノードに対して使用できます。
   - 「フェンスリセット(FenceReset)」: 問題のあるノードから SAN へのアクセスを無効にします。そのノードが正常に制限された場合は、ノードの非同期リセットも実行します。回復は、リセット確認を待たずに開始されます。このアクションは、IRIX ノードのみに対して使用できます。

- 「リセット(Reset)」: システム・コントローラに接続されているシリアル回線を通じてシステム・リセットを実行します。このアクションは、IRIX ノードのみに対して使用できます。
- 「シャットダウン(Shutdown)」: CXFS カーネル・メンバーシップが失われた場合に、ノードの CXFS カーネルベースのサービスを停止します。正常なクラスタは、ノードにシャットダウンする余裕を与えるために回復の開始を遅らせます。このアクションは、すべてのノードに対して必要です。

IRIX ノードでは、CXFS が単にノードの機能の一部で、システムを再起動するよりも CXFS にアクセスできなくなった方が問題が少ない場合(CXFS クライアントでもある大規模なコンピュータ・サーバの場合など)、データの整合性を保護するために I/O 制限を使用することができます。CXFS が主要なアクティビティで、早急にオンラインに戻す必要がある場合(CXFS ファイル・サーバの場合など)は、I/O 保護のためにシリアル・ハードウェア・リセットを使用することができます。

Solaris および Windows ノードでは、データの整合性を保護するための唯一の選択肢は、I/O 制限です。

メモ: Solaris および Windows ノードでは、データの整合性を保護するために I/O 制限が必要です。I/O 制限をサポートするには、SGI が販売、サポートする Brocade スイッチが必須であるため、マルチ OS CXFS クラスタには Brocade スイッチが必要です。

IRIX ノードのデフォルトの異常時のアクション階層は、「リセット、シャットダウン(Reset, Shutdown)」です。ほかのノードのデフォルトは、「シャットダウン(Shutdown)」です。

9. 「リセット(Reset)」または「フェンスリセット(FenceReset)」を含む異常階層を選択した場合は、次の情報を指定します。
   - このノード:
     - 「ポート・タイプ(Port Type)」:「L1」(Origin 300、Origin 3200C、Onyx 300、および Onyx 3200C システム用 L1 システム・コントローラ)、),「L2」(Origin 3400、Origin 3800、NUMAlink モジュール付き Origin 300、および Onyx 3000 シリーズ用 L2 システム・コントローラ)、,「MSC」(Origin 200、Onyx2 deskside、および SGI 2100、2200 deskside システム用モジュール・システム・コントローラ)、または「MMSC」(Rackmount SGI 2400、SGI 2800、および Onyx2 システム用マルチモジュール・システム・コントローラ)を選択します。
     - 「ポート・パスワード(Port Password)」: システム・コントローラ・ポートのパスワードで、ノードの `root` パスワードや PROM パスワードではありません。ノードによっては、システム管理者がこのパスワードを設定していない可能性があります。システム・コントローラ・ポート・パスワードを設定または変更する場合は、ご使用のノードのハードウェア・マニュアルを参照してください。
     - 「一時的に無効なポート(Temporarily Disable Port)」: シリアル・ハードウェア・リセット情報を指定して、現時点ではリセット機能を使用できないようにする場合は、このボックスをオンにします。このボックスがオンになっていると、CXFS はノードをリセットできません。

- オーナー(リセット・コマンドを送信するノード):
  - 「論理名(Logical Name)」: シリアル・ハードウェア・リセット・コマンドを送信するノードの名前。シリアル・ケーブルは、システム・コントローラ・ポートを通じて、定義されているノードとオーナー・ノードに物理的に接続されていなければなりません。また、ランタイム時に、ノードは CXFS プールで定義される必要があります。

    プルダウン・リストから論理名を選択したり、まだ定義されていないノードの論理名を入力できます。ただし、CXFS のノードは、ノードの接続性診断タスクを実行する前に定義しなければなりません。
  - 「TTY デバイス(TTY Device)」: システム・コントローラが接続されているオーナー・ノードの端末ポート (TTY) の名前(`/dev/ttyd2` など)。オーナー・ノードからノードをリモートで制御できるよう、ケーブルのもう一方はこのノードのシステム・コントローラ・ポートに接続します。

  シリアル・ハードウェア・リセット機能をまったく使用しない場合は、「システム・コントローラ・パラメータの設定(Set System Controller Parameters)」ボックスをクリックして、選択を解除します(オフにします)。IRIX 以外のオペレーティング・システムが実行されているノードに対しては、このボックスの選択を解除してください。

10. 「OK」をクリックします。

ネットワーク・インタフェース・フィールドへの入力としては、ホスト名または IP 番号を使用できます。ただし、ホスト名を使用するにはノードに DNS が必要となるため、実際の IP 番号を使用することをお薦めします。

メモ: 最初のノードのアイコンが表示領域で表示されるまで、2 番目のノードは追加しないでください。クラスタ・データベースに変更を加えると、その都度、すべての CXFS 管理ノードにクラスタ全体のステータス情報が送信されます。したがって、設定内での CXFS 管理ノードの数が多くなるほど送信に時間がかかります。

パーティション ID を決定するには、mkpart コマンドを使用できます。

- -n オプションは、パーティション ID をリストします(システムにパーティションが設定されていない場合は 0 になります)。
- -l オプションは、さまざまなパーティションのブリックをリストします(GUI では *rack#.slot#* 形式を使用してください)。

例は、次のとおりです(読みやすくするために出力を一部省略してあります)。

```
# mkpart -n
Partition id = 1
# mkpart -l
partition: 3 = brick: 003c10 003c13 003c16 003c21 ...
partition: 1 = brick: 001c10 001c13 001c16 001c21 ...
```

「パーティション ID(Partition ID)」フィールドには、以下のいずれかを入力できます。

```
1
001.10
```

GUI を使ったノードの定義例

以下の図に、新しいノードの定義例を示します。

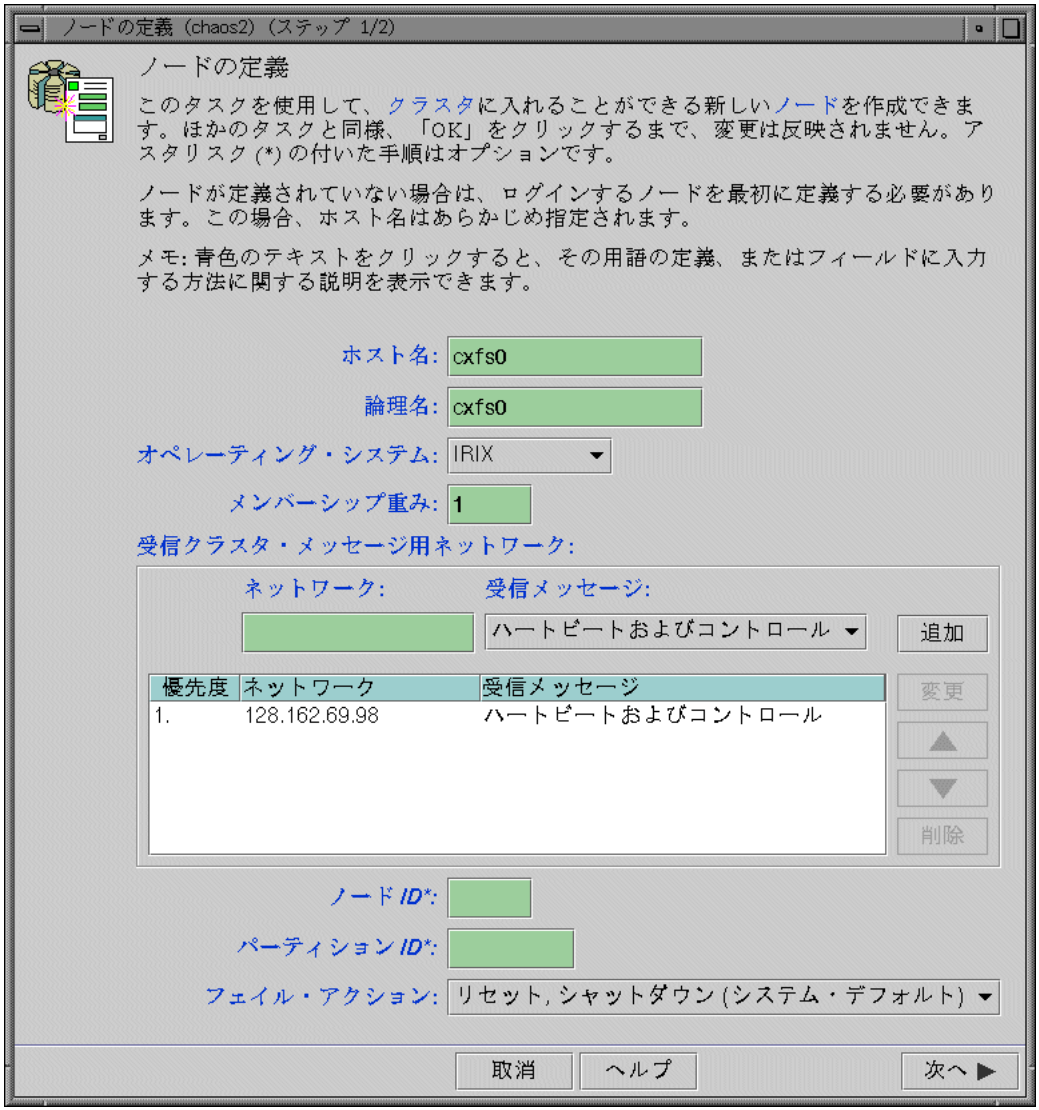

ノードの定義 (chaos2) (ステップ 1/2)
ノードの定義
このタスクを使用して、クラスタに入れることができる新しいノードを作成できます。ほかのタスクと同様、「OK」をクリックするまで、変更は反映されません。アスタリスク (*) の付いた手順はオプションです。
ノードが定義されていない場合は、ログインするノードを最初に定義する必要があります。この場合、ホスト名はあらかじめ指定されます。
メモ: 青色のテキストをクリックすると、その用語の定義、またはフィールドに入力する方法に関する説明を表示できます。
ホスト名: cxfs0
論理名: cxfs0
オペレーティング・システム: IRIX
メンバーシップ重み: 1
受信クラスタ・メッセージ用ネットワーク:
ネットワーク:
受信メッセージ:
ハートビートおよびコントロール
追加
優先度
ネットワーク
受信メッセージ
1.
128.162.69.98
ハートビートおよびコントロール
変更
削除
ノード ID*:
パーティション ID*:
フェイル・アクション: リセット, シャットダウン (システム・デフォルト)
取消
ヘルプ
次へ ▶

図4-3 ノード定義の例

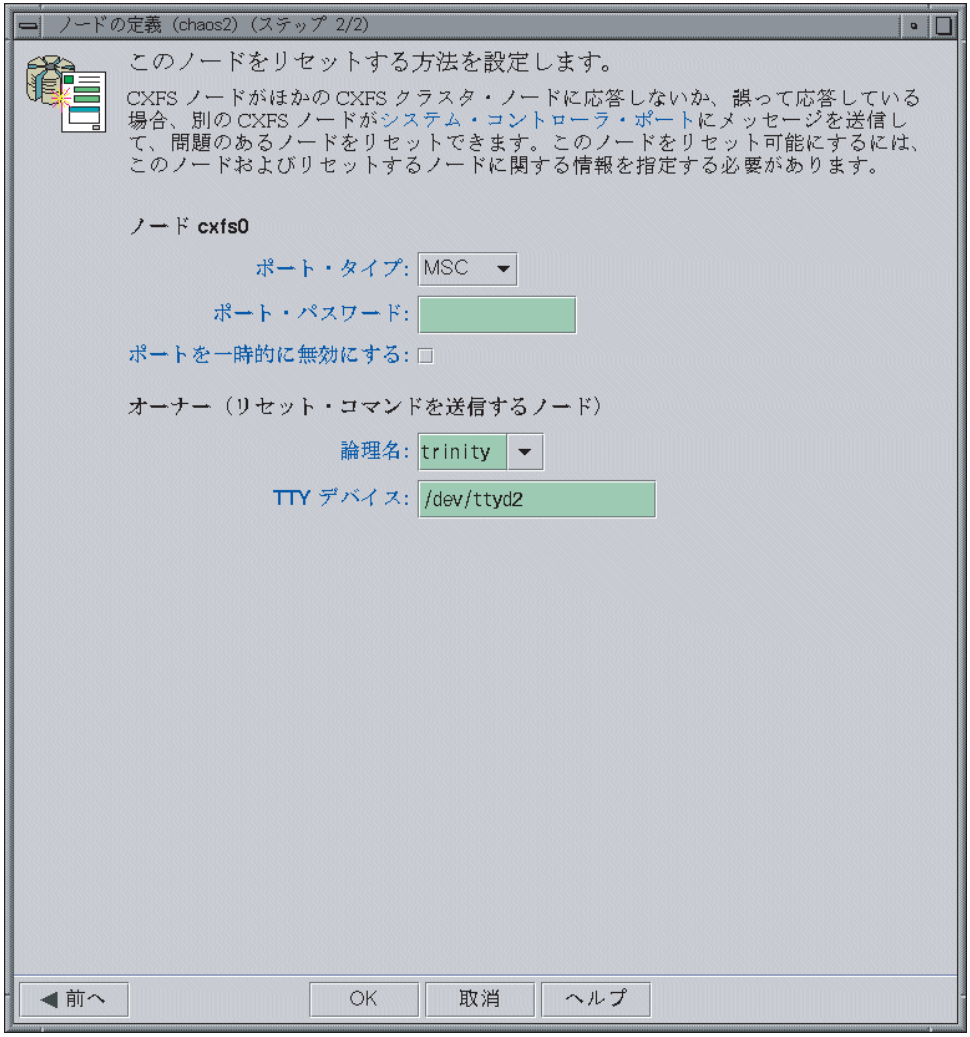


図4-4 システム・コントローラ設定の例

GUI を使ったクラスタ内のノードの追加と削除

プールにノードを追加して、クラスタを定義したら、クラスタ内に含めるノードを指定できます。

メモ: クラスタのアイコンが表示領域に表示されるまで、ノードを追加したり削除しないでください。「表示(View)」の選択肢は、「クラスタ内のノード(Nodes in Cluster)」に設定してください。

以下を実行します。

1. 希望のノードを追加または削除します。
   - ノードを追加するには、その論理名を「利用可能なノード(Available Nodes)」プルダウン・メニューから選択し、「追加(Add)」をクリックします。ノード名は、「クラスタに追加するノード(Nodes to Go into Cluster)」リストに表示されます。利用可能なノードをすべて選択するには、「すべて追加(Add All)」をクリックします。
   - ノードを削除するには、その論理名を「クラスタに追加するノード(Nodes to Go into Cluster)」画面でクリックします。すると、論理名が強調表示されます。その後、「削除(Remove)」をクリックします。
2. 「OK」をクリックします。

GUI を使ったノードのリセット

GUI を使用して、クラスタ内の IRIX ノードをリセットすることができます。これにより、指定したノードのシステム・コントローラ・ポートにリセット・コマンドが送信されます。ノードがリセットされると、クラスタ内のほかのノードで変更が検出され、アクティブなクラスタからこのノードが削除されます。ノードは、再起動すると、CXFS カーネル・メンバーシップに再設定されます。

ノードをリセットするには、以下を実行します。

1. 「リセットするノード(Node to Reset)」: リセットするノードをプルダウン・リストから選択します。
2. 「OK」をクリックします。

GUI を使ったノード定義の変更

ノードの名前またはノードのオペレーティング・システムを変更するには、そのノードを削除した後、新しいノードを定義しなければなりません。

ノードに関するその他の情報を変更するには、以下を実行します。

1. 「論理名(Logical Name)」: ノードの論理名をプルダウン・リストから選択します。論理名を選択すると、このノードの情報がさまざまなフィールドに入力されます。

2. 「受信クラスタ・メッセージ用ネットワーク(Networks for Incoming Cluster Messages)」: GUI では複数のネットワークを入力できますが、CXFS が使用するのは優先度が 1 のネットワークだけで、このネットワークはフェイルオーバーされません。FailSafe にも適用されるようクラスタを切替える場合以外(FailSafe では 2 つのネットワークが必要)は、追加のネットワークを定義しないでください。ネットワークの優先度は、クラスタ内の各ノードに対して同じでなければなりません。

   - 「ネットワーク(Network)」: 受信クラスタ・メッセージ用ネットワークを追加するには、IP アドレスまたはホスト名を「ネットワーク」テキスト・フィールドに入力し、「追加(Add)」をクリックします。
   - すでにリストに存在しているネットワークを変更するには、そのネットワークをリストでクリックして選択します。その後、「変更(Modify)」をクリックします。これにより、ネットワークは、リストからテキスト入力領域に移動されます。これで、ネットワークを変更できます。ネットワークをリストに追加し直すには、「追加(Add)」をクリックします。
   - ネットワークを削除するには、そのネットワークを優先度リストでクリックして選択します。その後、「削除(Delete)」をクリックします。
   - ネットワークの優先度を変更するには、そのネットワークを優先度リストでクリックして選択します。その後、上下の矢印をクリックして、リスト内でネットワークを別の位置に移動します。

   メモ: ネットワークを一時的に定義して、ネットワークでメッセージを受信しないようにする場合は、「なし(None)」設定を使用できます。詳細については、8 ページの「クラスタ環境」を参照してください。

   - 「パーティション ID(Partition ID)」: (オプション)パーティションが設定された Origin 3000 システムのパーティションを一意に定義します。システムにパーティションが設定されていない場合、このフィールドは空のままにします。パーティション ID の値を決定するには、`mkpart`(1M) コマンドを使用できます(以下を参照してください)。

3. 「FailSafe の設定を維持する(Keep FailSafe Settings)」: FailSafe で使用できるよう CXFS 管理ノードが定義されている場合は、CXFS で使用できるようこのノードを再設定できます。オプションとして、FailSafe で使用できないようノードを再設定することもできます。今後このノードを FailSafe で使用しない場合は、このオプションをオフにします。

4. 「異常時のアクション(Fail Action)」: 異常ノードの処理方法を決定するアクションのセットを指定します。2 番目のアクションは最初のアクションが失敗した場合にのみ実行され、3 番目のアクションは、最初のアクションと 2 番目のアクションが失敗した場合にのみ実行されます。

   使用可能なアクションは、以下のように、ノードのオペレーティング・システムに応じて決まります。

   - 「制限(Fence)」: 問題のあるノードからストレージ・エリア・ネットワーク (SAN: Storage Area Network) へのアクセスを無効にします。制限機能を使用すると、CXFS カーネル・メンバーシップをシリアル・ハードウェア・リセットより高速に回復することができます。このアクションは、すべてのノードに対して使用できます。
   - 「フェンスリセット(FenceReset)」: 問題のあるノードから SAN へのアクセスを無効にします。そのノードが正常に制限された場合は、ノードの非同期リセットも実行します。回復は、リセット確認を待たずに開始されます。このアクションは、IRIX ノードのみに対して使用できます。
   - 「リセット(Reset)」: システム・コントローラに接続されているシリアル回線を通じてシステム・リセットを実行します。このアクションは、IRIX ノードのみに対して使用できます。
   - 「シャットダウン(Shutdown)」: CXFS カーネル・メンバーシップが失われた場合に、ノードの CXFS カーネルベースのサービスを停止します。正常なクラスタは、ノードにシャットダウンする余裕を与えるために回復の開始を遅らせます。このアクションは、すべてのノードに対して必要です。

   IRIX ノードのデフォルトの異常時のアクション階層は、「リセット、シャットダウン(Reset, Shutdown)」です。ほかのノードのデフォルトは、「シャットダウン(Shutdown)」です。

5. 次のページに移動するには、「次へ(Next)」をクリックします。必要に応じて、値を変更します。
6. 「OK」をクリックします。

パーティション ID の値を決定するには、`mkpart` コマンドを使用できます。

- `-n` オプションは、パーティション ID をリストします(システムにパーティションが設定されていない場合は 0 になります)。
- `-l` オプションは、さまざまなパーティションのブリックをリストします(GUI では *rack#.slot#* 形式を使用してください)。

例は、次のとおりです(読みやすくするために出力を一部省略してあります)。

```
# mkpart -n
Partition id = 1
# mkpart -l
partition: 3 = brick: 003c10 003c13 003c16 003c21 ...
partition: 1 = brick: 001c10 001c13 001c16 001c21 ...
```

「パーティション ID(Partition ID)」フィールドには、以下のいずれかを入力できます。

```
1
001.10
```

GUI を使った FailSafe ノードの CXFS 用への切替え

FailSafe もインストールされている場合のみ、このタスクは CXFS GUI に表示されます。このタスクは、CXFS 管理ノードにのみ適用されます。

既存の IRIS FailSafe ノード(タイプ `FailSafe`)は、以下のタイプのいずれかに切替えることができます。

- `CXFS and FailSafe`
- `CXFS`

以下を実行します。

1. FailSafe GUI を使用して、切替えるノードで HA サービスを停止します。『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。
2. CXFS GUI を使用して、2 番目のハートビートおよびコントロール・ネットワークをノード定義に追加します。118 ページの「GUI を使ったノード定義の変更」を参照してください。
3. 以下の情報を入力します。
   - 「論理名(Logical Name)」: ノードの論理名をプルダウン・リストから選択します。
   - 「FailSafe の設定を維持する(Keep FailSafe Settings)」:
     - タイプ `CXFS and FailSafe` に切替えるには、チェックボックスをオンにします。
     - タイプ `CXFS` に切替えるには、チェックボックスをオフのままにします。
   - 「OK」をクリックします。

メモ: ノードの名前を変更する場合は、そのノードを削除した後、新しいノードを定義しなければなりません。

その他のパラメータを変更するには、118 ページの「GUI を使ったノード定義の変更」を参照してください。変更内容が FailSafe と CXFS の両方に対して適切であることも確認してください。

FailSafe に適用されるよう CXFS ノードを切替えるには、`cmgr`(1M) コマンドまたは FailSafe GUI を使用します。FailSafe GUI についての詳細は、『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。

GUI を使ったノードの削除

プールからノードを削除する前に、クラスタからそのノードを削除してください。詳細については、123 ページの「GUI を使ったクラスタ定義の変更」を参照してください。

ノードを削除するには、以下を実行します。

1. 「削除するノード(Node to Delete)」: 削除するノードの論理名をプルダウン・リストから選択します。
2. 「OK」をクリックします。

GUI を使ったノードの接続性のテスト

「ノードの接続性テスト(Test Node Connectivity)」画面では、ホスト間の `rsh`(1) アクセスが必要です。`/.rhosts` ファイルには、接続性をテストするホストとローカル・ホストを含めなければなりません。

接続性をテストするには、「CXFS マネージャ(CXFS Manager)」から以下を実行します。

1. 適切なラジオ・ボタンをクリックして、ネットワーク別またはシリアル接続別のどちらでテストするかを選択します。
2. テストするノードをプルダウン・リストから選択し、「追加(Add)」をクリックして、テスト・リストにそのノードを追加します。

   テストするノードのリストからノードを削除するには、その論理名をクリックして選択し、「削除(Delete)」をクリックします。
3. テストを開始するには、「テストの開始(Start Tests)」をクリックします。テストを停止するには、「テストの停止(Stop Tests)」をクリックします。
4. 別のテストを実行するには、「出力をクリア(Clear Output)」をクリックしてステータス画面をクリアし、手順 3 から新たに実行します。
5. ウィンドウを閉じるには、「閉じる(Close)」をクリックします。

GUI を使ったノードの表示

ノードを定義したら、表示領域の「表示(View)」の選択肢を使用して、以下を表示できます。

- 「プール内のノード(Nodes in Pool)」: 定義されているすべてのノードを表示します。
- 「クラスタ内のノード(Nodes In Cluster)」: 特定のクラスタのメンバーであるノードのみ表示します。

詳細なステータスおよび設定情報を表示するには、任意の名前またはアイコンをクリックします。

GUI を使ったクラスタ・タスク

以下のタスクを使用して、クラスタを定義、変更、削除、および表示することができます。

メモ:「新規クラスタの設定(Set Up a New Cluster)」設定ガイド・タスクを使用すると、クラスタおよびノードを設定するために必要なタスクを実行できます。74 ページの「GUI を使った新規クラスタの設定」を参照してください。

GUI を使ったクラスタの定義

クラスタは、プライベート・ネットワークによって互いに結合されたノードの集合です。クラスタは、単純な名前で識別されます。特定のノードは、1 つだけのクラスタのメンバーにできます。

クラスタを定義するには、以下を実行します。

1. 以下の情報を入力します。

   - 「クラスタ名(Cluster Name)」: クラスタの論理名。論理名は、アンダースコア (_) で始めたり、空白として使用される文字を含めたり、255 文字より長くすることはできません。ネットワークを共有するクラスタには、一意の名前を付ける必要があります。
   - 「クラスタ ID(Cluster ID)」: ネットワーク内で一意な 1〜128 の範囲の数字。クラスタ ID は、ネットワーク上に存在する可能性のあるほかのクラスタからクラスタ情報を受付けないようにするために、オペレーティング・システム・カーネルによって使用されます。カーネルは通信にデータベースを使用しないため、クラスタ通信を確認するためにクラスタ ID が必要です。カーネルのこの情報は、いったん初期化したら変更できません。このため、クラスタを定義したらクラスタ ID は変更しないでください。ネットワークを共有するクラスタには、一意の ID が必要です。
   - 「クラスタ・モード(Cluster Mode)」: クラスタは、通常、デフォルトの通常 (`Normal`) モードに設定します。

     試験中 (`Experimental`) にモードを設定すると、ノード異常を発生させることなくクラスタをデバッグできるよう、CXFS カーネル・メンバーシップ・コードでハートビートがオフになります。これは、短時間だけネットワークを切断したい場合(クラスタ・ネットワーク・アクティビティがほかになく、ハートビートがなくても異常が検出されるという条件で)や、CXFS ノードでカーネル・デバッガ(ハートビートを停止させます)を入力したい場合などに便利です。試験中 (`Experimental`) モードは、デバッグする場合にのみ使用することをお薦めします。
   - 「管理者に通知(Notify Administrator)」(クラスタおよびノードのステータス変更について):
     - 「メールで(By e-mail)」: これを選択した場合、電子メール・プログラム(デフォルトでは `/usr/sbin/Mail`)と、識別された管理者の電子メール・アドレスを指定する必要があります。複数のアドレスを指定するには、各アドレスをカンマで区切ります。CXFS では、ノードま

たはクラスタのステータスが変更されるたびに、電子メールがそれらのアドレスに送信されます。アドレスを指定しなかった場合、通知は送信されません。

- 「コマンドで(By other command)」: これを選択した場合、ノードまたはクラスタのステータスが変更されるたびに実行されるコマンドを指定する必要があります。
- 「なし(Never)」: これを選択した場合、通知が送信されないことが指定されます。

2. 「OK」をクリックします。

GUI を使ったクラスタ定義の変更

クラスタの状態の変更をクラスタ管理者に通知する方法を変更するには、以下を実行します。

1. 以下の情報を入力します。
   - 「クラスタ名(Cluster Name)」: プルダウン・リストから選択します。
   - 「クラスタ・モード(Cluster Mode)」: クラスタは、通常、デフォルトの通常 (`Normal`) モードに設定します。試験中 (`Experimental`) モードについての詳細は、122 ページの「GUI を使ったクラスタの定義」を参照してください。
   - 「管理者に通知(Notify Administrator)」: 希望の通知方法を選択します。詳細については、122 ページの「GUI を使ったクラスタの定義」を参照してください。
2. 「OK」をクリックします。

クラスタを構成しているノードを変更するには、117 ページの「GUI を使ったクラスタ内のノードの追加と削除」を参照してください。

メモ: クラスタの名前を変更する場合は、そのクラスタを削除した後、新しいクラスタを定義しなければなりません。CXFS サービスがノードで開始されている場合は、サービスを再起動するか、クラスタの名前を変更するときにクラスタ ID 番号を再使用してください。

ただし、CXFS ファイルシステムがすでに定義されている場合にクラスタの名前を変更すると、CXFS は、それらのファイルシステムをマウントできなくなるので注意してください。詳細については、272 ページの「クラスタの名前を変更後、ファイルシステムをマウントできない」を参照してください。

GUI を使った FailSafe クラスタの CXFS 用への切替え

FailSafe もインストールされている場合のみ、このタスクは CXFS GUI に表示されます。

既存の IRIS FailSafe クラスタ(つまり、タイプ `FailSafe`)からの情報を切替えて、CXFS(つまり、タイプ `CXFS and FailSafe` またはタイプ `CXFS`)に適用されるクラスタを作成するには、以下を実行します。

1. 以下の情報を入力します。
   - 「クラスタ名(Cluster Name)」: プルダウン・リストから選択します。
   - 「クラスタ ID(Cluster ID)」: ネットワーク内で一意な 1〜128 の範囲の数字を入力します。クラスタ ID は、ネットワーク上に存在する可能性のあるほかのクラスタからクラスタ情報を受付けないようにするために、オペレーティング・システム・カーネルによって使用されます。カーネルは通信にデータベースを使用しないため、クラスタ通信を確認するためにクラスタ ID が必要です。カーネルのこの情報は、いったん初期化したら変更できません。このため、クラスタを定義したらクラスタ ID は変更しないでください。
2. 「OK」をクリックします。

クラスタは、IRIS FailSafe と CXFS の両方に適用されるようになります。クラスタを構成しているノードを変更するには、117 ページの「GUI を使ったクラスタ内のノードの追加と削除」を参照してください。

メモ: クラスタの名前を変更する場合は、そのクラスタを削除した後、新しいクラスタを定義しなければなりません。

GUI を使ったクラスタの削除

ノードが含まれているクラスタは削除できません。したがって、最初に、クラスタに含まれているノードをクラスタから移動してください。詳細については、117 ページの「GUI を使ったクラスタ内のノードの追加と削除」を参照してください。

クラスタを削除するには、以下を実行します。

1. 「削除するクラスタ(Cluster to Delete)」: クラスタの名前が選択されます。
2. 「OK」をクリックします。

GUI を使ったクラスタの表示

「表示(View)」の選択肢からは、チェックする要素を選択できます。クラスタの詳細を表示するには、クラスタ名またはアイコンをクリックします。すると、ステータスと設定の情報が、右側の詳細領域に現れます。

また、`cluster_status` コマンドを使用することもできます。237 ページの 第9章「ステータスのモニタ」を参照してください。

GUI を使ったクラスタ・サービス・タスク

以下のタスクを使用して、CXFS クラスタ・サービスの開始と停止、およびログの設定を行うことができます。

GUI を使った CXFS サービスの開始

CXFS サービスを開始し、システムが再起動されるたびに CXFS サービスが自動的に再開されるよう設定するには、以下を実行します。

1. 「アクティブにするノード(Node(s) to Activate)」:「すべてのノード(`All Nodes`)」、または CXFS サービスを開始する個々のノードを選択します。
2. 「OK」をクリックします。

GUI を使った CXFS サービスの停止(通常 CXFS シャットダウン)

ノードで CXFS サービスが停止されると、ファイルシステムは、自動的にそのノードからアンマウントされます。

CXFS サービスを一時的に停止する(つまり、再起動するとサービスが再開されるよう設定されている場合に再開できるようにする)には、GUI 外のシェル・ウィンドウで次のコマンド・ラインを使用します。

```
# /etc/init.d/cxfs stop
```

以下の手順を実行することによって、指定したノードまたはクラスタで CXFS を停止し、再起動によって CXFS サービスが再開されないようにすることができます。

メモ:この方法を使用して CXFS サービスを停止した場合、ノードが再起動されてもサービスは再開されません。

---

1. 以下の情報を入力します。
   - 「強制(Force)」: エラーが発生しても(通常は、エラーが発生すると停止操作を防止します)、CXFS サービスを強制的に停止する場合は、「強制」チェックボックスをオンにします。
   - 「停止するノード(Node(s) to Deactivate)」:「すべてのノード(`All Nodes`)」、または CXFS サービスを停止する個々のノードを選択します。

     ノードで CXFS サービスを停止すると、そのノードはファイルシステムにアクセスできなくなります。そのノードがファイルシステムのメタデータ・サーバとして動作している場合、メタデータ・サーバとして使用可能なノードのリストに含まれる別のノードが選択されます。このプロセス中は、ファイルシステムのクライアントで遅れが発生します。

2. 「OK」をクリックします。プロセスが完了するには、数分かかる場合があります。

ノードで CXFS サービスを停止すると、そのノードは、クラスタのアクティブなメンバーではなくなります。CXFS サービスは、システムが再起動されても再開されません。

注意: CXFS サービスは、`shutdown`(1M) コマンドまたは `reboot`(1M) コマンドを使用する前に停止してください。CXFS サービスがアクティブなときに `shutdown` または `reboot` を実行すると、クラスタのほかのノードによってノード異常と解釈され、そのノードに対して強制的に回復が実行されます。

GUI を使ったタイブレーカー・ノードの設定

CXFS タイブレーカー・ノードは、サーバとして使用可能なノードのちょうど半数が実行中で、互いに通信可能である場合に、CXFS カーネル・メンバーシップ定足数が維持されるかどうかを決定します。デフォルトの CXFS タイブレーカー・ノードはありません。詳細については、28 ページの「メンバーシップ定足数」を参照してください。

注意: サーバとして使用可能なノードが 2 つのクラスタ内の CXFS タイブレーカー・ノードに異常が発生したり、管理者が CXFS サービスを停止した場合は、その他のノードによって強制シャットダウンが実行され、CXFS ファイルシステムはすべてアンマウントされます。

データの整合性を保証するため、すべての CXFS システム(特に、サーバとして使用可能なノードが 2 つのクラスタ)に対してシリアル・ハードウェア・リセットまたは I/O 制限、あるいはその両方を使用することをお薦めしています。IRIS FailSafe では、リセットは必須です。

現在の CXFS タイブレーカー・ノードは、クラスタの詳細表示に表示されます。

CXFS タイブレーカー・ノードを設定するには、以下を実行します。

1. 「タイブレーカー・ノード(Tie-Breaker Node)」: 希望のノードをリストから選択します。現在、CXFS タイブレーカー・ノードがある場合は、デフォルトでそのノードが選択されます。

   CXFS タイブレーカー・ノードの設定を解除するには、「なし(None)」を選択します。

2. 「OK」をクリックします。

GUI を使ったログの設定

CXFS では、各 CXFS デーモンのログを維持します。CXFS は、通常の処理と重大なエラーの両方を、`/var/adm/SYSLOG` のほかに各ログ・グループの個々のログにも記録します。維持するログのレベルに基づいて、ログはカスタマイズすることができます。

注意: ログ・ファイルの名前は変更しないでください。名前を変更すると、エラーが発生します。

ログ設定を定義するときは、次の情報を指定します。

- 「ログ・グループ(Log Group)」: 同じログ設定を使用する 1 つまたは複数の CXFS プロセスのセット。通常、ログ・グループは、`crsd` などの 1 つの CXFS デーモンに対応します。
- 「ログ・レベル(Log Level)」: 関連付けられているログ・グループのログ・ファイルに CXFS が書込むログ・メッセージの量を制御する数字。
- 「ログ・ファイル(Log File)」: メッセージのログを記録するファイル。

237 ページの「ログ・ファイル」も参照してください。

GUI を使ったログ・グループ定義の表示

ログ・グループ定義を表示するには、以下を実行します。

1. 「ログ・グループ(Log Group)」: 表示するログ・グループをメニューから選択します。

   これを選択すると、そのログ・グループの現在のログ・レベルとログ・ファイルがタスク・ウィンドウに表示されます。必要に応じて、このウィンドウでこれらの設定を変更できます。

2. 「OK」をクリックします。

GUI を使ったログ・グループの設定

ログ・グループを設定するには、「ログの設定(Set Log Configuration)」タスク内の以下を実行します。

1. 適切な情報を入力します。

   - 「ログ・グループ(Log Group)」: ログ・グループをプルダウン・リストから選択します。ログ・グループは、同じログ設定に基づいて同じログ・ファイルにログを記録するプロセスのセットです。各 CXFS デーモンは、ログ・グループを 1 つ作成します。設定は、`cli` ログ・グループと `crsd` ログ・グループについてはプール内のすべてのノードに適用され、`clconfd` ログ・グループと `diags` ログ・グループについてはクラスタ内のすべてのノードに適用されます。
   - 「ログ・レベル(Log Level)」: ログ・レベルを選択します。これは、ログの量を指定します。

注意: 「デフォルト(Default)」ログ・レベルは非常に冗長であるため、これを使用すると、ディスク容量の問題が発生する場合があります。この場合は、低いログ・レベルを選択できます。215 ページの「ログ・ファイル管理」、54 ページの「CXFS 管理ノードの `/etc/config/cad.options`」、および 55 ページの「CXFS 管理ノードの `/etc/config/fs2d.options`」も参照してください。

値は、以下のとおりです。

- 「オフ(Off)」: ログには記録されません。
- 「最小(Minimal)」: 重大なエラーと通常の操作の通知をログに記録します(これらのメッセージのログは、/var/adm/SYSLOG ファイルにも記録されます)。
- 「情報(Info)」:「最小(Minimal)」通知と警告をログに記録します。
- 「デフォルト(Default)」: すべての「情報(Info)」メッセージとその他の通知をログに記録します。
- 「デバッグ 0(Debug 0)」～「デバッグ 9(Debug 9)」: 数字の大きさに応じて、ログに記録されるデバッグ情報(データ構造を含む)が増えます。

cmgr(1M) コマンドでは、これらのログ・レベルを示すために、数字のセットを使用します。175 ページの「cmgr を使ったログ・グループの設定」を参照してください。

2. 「ログ・ファイル(Log File)」: この値は変更しないでください。
3. 「OK」をクリックします。

GUI を使ったローカル・ノードのメンバーシップ取消し

ローカル・ノードの CXFS カーネル・メンバーシップは、強制 CXFS シャットダウンの実行が必要な場合(209 ページの「データベースと CXFS のシャットダウン」を参照してください)など、エラーが発生したときにのみ取消すことをお薦めします。

ローカル・ノードの CXFS カーネル・メンバーシップを取消すには、以下を実行します。

1. 「ローカル・ノード(Local Node)」: ポップアップ・ウィンドウに表示されるローカル・ノードの名前を確認します。
2. 「OK」をクリックして、タスクを完了します。

このタスクの結果は、クラスタのその他の部分でノード異常と解釈されます。このため、クラスタのその他の部分では、CXFS カーネル・メンバーシップ定足数が失われたために異常が発生したり、異常ノードのリセットが決定される場合があります。リセットを回避するには、ノード定義を変更して、システム・コントローラ・ステータスを無効にできます。

GUI を使ったローカル・ノードのメンバーシップ許可

強制 CXFS シャットダウンを必要とする問題を解決したら、ローカル・ノード(GUI が接続しているノード)の CXFS カーネル・メンバーシップを許可する必要があります。これによって、そのノードにクラスタ内の

メンバーシップの再適用を許可します。強制 CXFS シャットダウンは、手動またはカーネルによって実行できます。詳細については、209 ページの「データベースと CXFS のシャットダウン」を参照してください。

以下の状況では、ローカル・ノードの CXFS カーネル・メンバーシップをアクティブに許可する必要があります。

- 128 ページの「GUI を使ったローカル・ノードのメンバーシップ取消し」で説明されているように、手動取消しの後
- コンソールまたは /var/adm/SYSLOG のエラー・メッセージによって指示された場合
- カーネルによって取消しが実行された後。この状況は、/var/adm/SYSLOG 内の次のメッセージによって示されます。

```
Membership lost - withdrawing from cluster
```

ローカル・ノードの CXFS カーネル・メンバーシップを許可するには、以下を実行します。

1. 「ローカル・ノード(Local Node)」: ポップアップ・ウィンドウに表示されるローカル・ノードの名前を確認します。
2. 「OK」をクリックして、タスクを完了します。

GUI を使ったスイッチと I/O 制限タスク

以下のタスクを使用して、CXFS クラスタ用の Brocade ファイバ・チャネル・スイッチの設定、および I/O 制限の実行を行うことができます。

メモ: Solaris および Windows ノードでは、データの整合性を保護するために I/O 制限が必要です。I/O 制限をサポートするには、SGI が販売、サポートする Brocade スイッチが必須であるため、マルチ OS CXFS クラスタには Brocade スイッチが必要です。

GUI を使ったスイッチの定義

このタスクを使用して、SGI が販売、サポートする新しい Brocade ファイバ・チャネル・スイッチを定義し、クラスタ内で I/O 制限をサポートできます。以下を実行します。

1. 次の情報を入力します。
   - 「スイッチ名(Switch Name)」: Brocade ファイバ・チャネル・スイッチのホスト名を入力します。これは、スイッチの IP アドレスを判断するために使用されます。

- 「ユーザ名(Username)」: Brocade ファイバ・チャネル・スイッチに `telnet` メッセージを送信する場合に使用するユーザ名を入力します。デフォルトでは、この値は `admin` です。
- 「パスワード(Password)」: 指定した「ユーザ名(Username)」フィールドのパスワードを入力します。
- 「マスク(Mask)」: Brocade ファイバ・チャネル・スイッチ内の制限されないポートのリストを示す、(64 ビット・ポート・ビットマップを表す) 16 進の文字列を入力します。ポートには、ゼロから番号が付けられます。ビット `i` がゼロ以外の場合、`i` に対応するポートは、常に制限操作から除外されます。たとえば、`A4` というマスクは、3 番目、6 番目、および 8 番目のポート(ポート番号 2、5、および 7) が制限を受けないことを示します。

  CXFS 管理ノードでは、使用可能な HBA を自動的に検出し、「制限(Fence)」または「フェンスリセット(FenceReset)」の異常時のアクションが選択されている場合は、すべてのファイバ・チャネル HBA の制限をデフォルトでオフにします。ただし、マスクされた HBA は制限されません。マスキングを使用すると、SAN に接続されているがクラスタと共有されていないデバイスの制限を防止し、CXFS のステータスに関係なく、常にこれらのデバイスを利用できます。このため、テープ・ストレージへのアクセスに使用される HBA、またはローカル (クラスタ化されていない)・デバイスにアクセスするためだけに使用される HBA をマスクすることができます。

2. 「OK」をクリックして、タスクを完了します。

GUI を使ったスイッチ定義の変更

このタスクを使用して、既存の Brocade ファイバ・チャネル・スイッチ定義を変更できます。以下を実行します。

1. 次の情報を入力します。
   - 「スイッチ名(Switch Name)」: 変更する Brocade ファイバ・チャネル・スイッチのホスト名を選択します。
   - 「ユーザ名(Username)」: Brocade ファイバ・チャネル・スイッチに `telnet` メッセージを送信する場合に使用するユーザ名を入力します。デフォルトでは、この値は `admin` です。
   - 「パスワード(Password)」: 指定した「ユーザ名(Username)」フィールドのパスワードを入力します。
   - 「マスク(Mask)」: 制限されないポートのリストを示す 64 ビット・ビットマップを入力します。詳細については、129 ページの「GUI を使ったスイッチの定義」を参照してください。
2. 「OK」をクリックして、タスクを完了します。

GUI を使ったスイッチ・ポート情報の更新

このタスクを使用して、ホスト・バス・アダプタ (HBA: Host Bus Adapter) と Brocade ファイバ・チャネル・スイッチのポートのマッピングを更新できます。スイッチを再設定したり、ポートを追加する場合は、このコマンドを実行する必要があります。「OK」をクリックして、タスクを完了します。

GUI を使ったスイッチ定義の削除

このタスクを使用して、既存の Brocade ファイバ・チャネル・スイッチ定義を削除できます。以下を実行します。

1. 「スイッチ名(Switch Name)」: 削除するファイバ・チャネル・スイッチのホスト名を選択します。
2. 「OK」をクリックして、タスクを完了します。

GUI を使ったノードの I/O 制限の強化

このタスクを使用して、ノードの I/O 制限を強化することができます。I/O 制限を強化すると、そのノードは SAN から分離されます。そして、CXFS から `telnet` プロトコルを通じて Brocade スイッチにメッセージが送信され、ポートが無効に設定されます。ノードが分離された後は、そのノードによって共有 CXFS ファイルシステムのデータが破壊されることはありません。

以下を実行します。

1. 「制限を強化するノード(Raise Fence for Node)」: 分離するノードの名前を選択します。「制限(Fence)」または「フェンスリセット(FenceReset)」の異常時のアクションを使用して設定されているノードのみ選択できます。
2. 「OK」をクリックして、タスクを完了します。

GUI を使ったノードの I/O 制限の緩和

このタスクを使用して、ポートを再度有効にすることにより、特定のノードの I/O 制限を緩和させることができます。I/O 制限を緩和すると、ノードは SAN に再接続して、共有 CXFS ファイルシステムにアクセスできるようになります。

以下を実行します。

1. 「制限を緩和するノード(Lower Fence for Node)」: 再接続するノードを選択します。「制限(Fence)」または「フェンスリセット(FenceReset)」の異常時のアクションを使用して設定されているノードのみ選択できます。
2. 「OK」をクリックして、タスクを完了します。

GUI を使ったファイルシステム・タスク

以下のタスクを使用して、CXFS ファイルシステムを共有 XVM ボリュームとして設定できます。これらの共有ボリュームには、CXFS クラスタ内のすべてのノードから直接アクセスできます。各ボリュームは、それぞれのデバイス名で識別されます。各ボリュームには、クラスタ内の各ノードと同じマウント・ポイントが必要です。

メモ:「新規 CXFS ファイルシステムの設定(Set Up a New CXFS Filesystem)」設定ガイド・タスクを使用すると、新しい CXFS ファイルシステムを設定するために必要な手順を実行できます。76 ページの「GUI を使った新規 CXFS ファイルシステムの設定」を参照してください。

ファイルシステムの作成

このタスクを使用して、オンラインであっても開いていないボリュームでファイルシステムを作成できます。

注意:「OK」をクリックすると、ターゲット・ボリュームに存在するデータはすべて消去されます。

ファイルシステムを作成するには、以下を実行します。

1. 次の情報を入力します。

   - 「ドメイン(Domain)」: 作成するボリューム・エレメントを所有するドメインを選択します。GUI が接続されているノードでのみ使用するようボリューム・エレメントまたはディスクが定義されている場合は、「ローカル(Local)」を選択し、クラスタ内の複数のノードで使用するよう定義されている場合は、「クラスタ(Cluster)」を選択します。

   - 「ボリューム・エレメント(Volume Element)」: ファイルシステムを作成するボリュームを選択するか、ファイルシステムに対して親ボリュームが使用されるボリューム・エレメントを選択します。メニューには、使用可能なボリューム・エレメントのみがリストされます。ボリューム以外のボリューム・エレメントが作成または分離された場合、システムでは、そのボリューム・エレメントに関連付けられたボリュームとサブボリュームが自動的に作成されます。自動的に生成されたボリュームに明示的に名前を付けなかった場合、GUI ではその子のみが表示されます。

   - 「サイズの指定(Specify Sizes)」: ファイルシステムに対するデフォルトのオプション(データ領域サイズ、ログ・サイズ、リアルタイム・セクション・サイズなど)を変更するには、このボックスをオンにします。

     デフォルトでは、データ領域サイズがデータ・サブボリュームのサイズに等しいファイルシステムが作成されます。ボリュームにログ・サブボリュームが含まれている場合、ログ・サイズは、このログ・サブボリュームのサイズに設定されます。また、ボリュームにリアルタイム・サブボリュームが

含まれている場合、リアルタイム・セクション・サイズは、このリアルタイム・サブボリュームのサイズに設定されます。

2. 「サイズの指定(Specify Sizes)」ボックスをオンにした場合は、「次へ(Next)」をクリックして 2 ページ目に移動し、そのページで次の情報を入力します。これらのフィールドについての詳細は、`mkfs_xfs`(1M) のマン・ページを参照してください。

   - 「ブロック・サイズ(Block Size)」: ファイルシステムの基本ブロック・サイズ(バイト単位)を選択します。
   - 「ディレクトリ・ブロック・サイズ(Directory Block Size)」: ファイルシステムの命名(ディレクトリ)領域のサイズ(バイト単位)を選択します。
   - 「i ノード・サイズ(Inode Size)」: i ノード割当てに使用されるブロックの数(バイト単位)を入力します。i ノード・サイズは、「ブロック・サイズ(Block Size)」の値の半分を超えることはできません。
   - 「最大 i ノード・スペース(Maximum Inode Space)」: i ノードに割当てることのできる、ファイルシステムの領域の最大割合を入力します。デフォルトは 25% です。値を 0 に設定すると、ファイルシステム全体が i ノード・ブロックになります。
   - 「未書込みエクステントのフラグ付け(Flag Unwritten Extents)」: 未書込みエクステントにフラグを付けるには、このボックスをオンにします。未書込みエクステントにフラグが付けられている場合、ファイルシステムの書込みパフォーマンスは、事前に割当てられているファイル・エクステントに関してマイナスの影響を受けます。これは、ファイルの範囲のエクステント・フラグを切替えるために、ファイルシステム・トランザクションがさらに必要となるためです。

     フラグ付け機能をサポートしていないオペレーティング・システム・バージョンでファイルシステムを使用する必要がある場合は、この機能を(ボックスをオフにすることによって)無効にしてください。
   - 「データ領域サイズ(Data Region Size)」: ファイルシステムのデータ領域のサイズを、512 バイト・ブロックの数として入力します。この数は、通常、データ・サブボリュームのサイズに等しくなります。データ・サブボリュームのサイズよりも小さい領域をファイルシステムが使用する場合にのみ、0 以外のサイズを指定してください。
   - 「ログ・サブボリュームを使ってログを作成(Use Log Subvolume for Log)」: ファイルシステムのログ・セクションを XVM 論理ボリュームのログ・サブボリュームに書込む必要があることを指定するには、このボックスをオンにします。ボリュームに ログ・サブボリュームが含まれていない場合、ログ・セクションはデータ・サブボリュームのデータ・セクションの断片になります。
   - 「ログ・サイズ(Log Size)」: ファイルシステムのログ・セクションのサイズを、512 バイト・ブロックの数として入力します。ログ・サブボリュームのサイズよりも小さい領域をログが使用する場合にのみ、0 以外のサイズを指定してください。

- 「リアルタイム・セクション・サイズ(Real-Time Section Size)」: ファイルシステムのリアルタイム・セクションのサイズを、512 バイト・ブロックの数として入力します。この値は、通常、リアルタイム・サブボリュームのサイズに等しくなります(存在する場合)。real-time サブボリュームのサイズよりも小さい領域をリアルタイム・セクションが使用する場合にのみ、0 以外のサイズを指定してください。

3. 「OK」をクリックします。

ファイルシステムの拡張

このタスクを使用して、マウントされたファイルシステムを拡張できます。

メモ: ファイルシステムを拡張するには、ファイルシステムがマウントされている論理ボリュームのサイズをまず増やす必要があります。「ミラーまたは結合をボリューム・エレメントに挿入する(Insert Mirrors or Concats above Volume Elements)」タスクを起動して結合を追加したり、ドラッグアンドドロップ・メカニズムを使用して既存の結合にスライスを接続することが可能です。既存のボリューム・エレメントへのスライスの追加は、そのボリューム・エレメントまたはボリューム・エレメントの先祖のデータのレイアウトが変更された場合は実行できません。詳細については、『XVM Volume Manager Administrator's Guide』を参照してください。

ファイルシステムを拡張するには、以下を実行します。

1. 次の情報を入力します。

   - 「ファイルシステム(Filesystem)」: 拡張するファイルシステムの名前を選択します。使用可能なファイルシステムのリスト内容は、XFS スーパーブロックを含むブロック・デバイスを検索することによって決定されます。

   - 「サイズの指定(Specify Sizes)」: ファイルシステムに対するデフォルトのオプション(データ領域サイズ、ファイルシステムに対してすでに存在する場合はログ・サイズ、およびリアルタイム・セクション・サイズなど)を変更するには、このオプションをオンにします。

     デフォルトでは、データ領域サイズがデータ・サブボリュームのサイズに等しいファイルシステムが作成されます。ボリュームにログ・サブボリュームが含まれている場合、ログ・サイズは、このログ・サブボリュームのサイズに設定されます。また、ボリュームにリアルタイム・サブボリュームが含まれている場合、リアルタイム・セクション・サイズは、このリアルタイム・サブボリュームのサイズに設定されます。

2. 「サイズの指定(Specify Sizes)」ボックスをオンにした場合は、「次へ(Next)」をクリックして 2 ページ目に移動します。これらのフィールドについての詳細は、`mkfs_xfs`(1M) のマン・ページを参照してください。

   - 「データ領域サイズ(Data Region Size)」: ファイルシステムのデータ領域のサイズを、512 バイト・ブロックの数として入力します。この数は、通常、データ・サブボリュームのサイズに等しくなり

ます。データ・サブボリュームのサイズよりも小さい領域をファイルシステムが使用する場合にのみ、0 以外のサイズを指定してください。

- 「ログ・サイズ(Log Size)」: ファイルシステムのログ・セクションのサイズを、512 バイト・ブロックの数として入力します。ログ・サブボリュームのサイズよりも小さい領域をログが使用する場合にのみ、0 以外のサイズを指定してください。このオプションは、ファイルシステムに ログ・サブボリュームがある場合にのみ表示されます。
- 「リアルタイム・セクション・サイズ(Real-Time Section Size)」: ファイルシステムのリアルタイム・セクションのサイズを、512 バイト・ブロックの数として入力します。この値は、通常、リアルタイム・サブボリュームのサイズに等しくなります(存在する場合)。real-time サブボリュームのサイズよりも小さい領域をリアルタイム・セクションが使用する場合にのみ、0 以外のサイズを指定してください。このオプションは、ファイルシステムにリアルタイム・サブボリュームがある場合にのみ表示されます。

3. 「OK」をクリックします。

GUI を使った CXFS ファイルシステムの定義

このタスクでは、ディスク・ドライブのボリューム・ヘッダ、XVM 論理ボリューム、およびファイルシステムが作成されていることを想定しています。詳細については、73 ページの「GUI を使った設定」を参照してください。

ファイルシステムのメタデータ・サーバを定義するには、以下を実行します。

1. 次の情報を入力します。
   - 「デバイス名(Device Name)」: これは、CXFS クラスタ内のすべてのノード間で共有される XVM ボリュームのデバイス名です。名前は必ず /dev/cxvm/ で始まるので、この情報はデフォルトで提供されます。GUI では、名前の最後の部分(cxvm 以降)が、論理ファイルシステム名として自動的に使用されます。異なる論理名を使用するには、cmgr コマンドを使用します(178 ページの「cmgr を使った CXFS ファイルシステムの定義」を参照してください)。
   - 「マウント・ポイント(Mount Point)」: この値は、指定した XVM ボリュームが接続されるディレクトリです。このディレクトリ名は、スラッシュ (/) で始まる必要があります。詳細については、mount(1M) のマン・ページを参照してください。
   - (オプション)「マウント・オプション(Mount Options)」: これらのオプションは、mount(1M) コマンドに渡され、指定した XVM ボリュームへのアクセスを制御するために使用されます。使用可能なオプションのリストについては、fstab(4) のマン・ページを参照してください。
   - 「メタデータ・サーバ(Metadata Servers)」: メタデータ・サーバとして動作できる CXFS 管理ノードのリスト。サーバのリストに CXFS 管理ノードを追加するには、その名前をプルダウン・ノード・リ

ストから選択し、「追加(Add)」をクリックします。リストされているノードをすべて選択するには、「すべて追加(Add All)」をクリックします。

サーバのリストからノードを削除するには、その名前をリストでクリックして選択し、「削除(Remove)」をクリックします。

メモ: サーバの順序は重要です。リストの最初のノードが優先メタデータ・サーバになります。論理名をクリックして選択した後、矢印ボタンをクリックして、使用される順にサーバを並べ替えます。

しかし、ネットワーク・レーテンシやその他の予測不能な遅れがあるため、起動サイクル中にどのサーバが実際にサーバになるかは予想できません。このため、リスト内で最初に使用可能なノードが、メタデータ・サーバとして使用されます。

- 「マウント可能なノード(Enable Mount on)」: クラスタ内のすべてのノード、または 2 ページ目で指定する、選択された CXFS 管理ノードのリストのいずれかを選択します。ファイルシステムは、常に現在のメタデータ・サーバにマウントされます。
- 「ノードがクラスタに後で追加される場合(If Nodes are Added to the Cluster Later)」: このオプションは、後でクラスタに追加される可能性のあるすべてのノードにファイルシステムをマウントすることを許可します。このオプションはデフォルトで選択されます。
- 上で「選択されたノードのみ(Only Selected Nodes)」を選択した場合は、「次へ(Next)」ボタンをクリックして、タスクの 2 ページ目に移動します。

  「選択されたノード(Selected Nodes)」: 「ノード(Node)」リストから希望のノードを選択できます。また、「すべて追加(Add All)」ボタンを使用して、すべてのノードを選択することもできます。これは、「クラスタ内に現在あるすべてのノード(All Nodes Currently in Cluster)」を選択するのと同じです。

2. 「OK」をクリックします。

GUI を使った CXFS ファイルシステムの変更

メモ: マウント済みのファイルシステムを変更することはできません。

既存のファイルシステムを変更するには、以下を実行します。

1. 次の情報を入力します。
   - 「変更するファイルシステム(Filesystem to Modify)」: ファイルシステムをプルダウン・メニューから選択します。これにより、さまざまなフィールドに、選択したファイルシステムの情報が表示されます。

- 「マウント・ポイント(Mount Point)」および「マウント・オプション(Mount Options)」: 選択したファイルシステムに対して表示されている情報を、必要に応じて変更します。テキストを消去するには、<Backspace> キーを使用してテキストを削除するか、テキストを選択して上書き入力します。
- (オプション)「マウント・オプション(Mount Options)」: これらのオプションは、`mount`(1M) コマンドに渡され、指定した XVM ボリュームへのアクセスを制御するために使用されます。使用可能なオプションのリストについては、`fstab`(4) のマン・ページを参照してください。
- 「メタデータ・サーバ(Metadata Servers)」:
  - サーバのリストからノードを削除するには、その名前をクリックした後、「削除(Delete)」をクリックします。
  - サーバのリストに新しい CXFS 管理ノードを追加するには、そのノードをプルダウン・ノード・リストから選択し、「追加(Add)」をクリックします。CXFS 管理ノードをすべて選択するには、「すべて追加(Add All)」を選択します。
  - サーバの優先度を再度並べ替えるには、その名前をクリックして選択した後、必要に応じて矢印ボタンをクリックします。
- 「マウント可能なノード(Enable Mount on)」: クラスタ内のすべてのノード、または 2 ページ目で指定する、選択されたノードのリストのいずれかを選択します。ファイルシステムは、常に現在のメタデータ・サーバにマウントされます。
- 「ノードがクラスタに後で追加される場合(If Nodes are Added to the Cluster Later)」: このオプションは、後でクラスタに追加される可能性のあるすべてのノードにファイルシステムをマウントすることを許可します。このオプションはデフォルトで選択されます。
- 上で「選択されたノードのみ(Only Selected Nodes)」を選択した場合は、「次へ(Next)」ボタンをクリックして、タスクの 2 ページ目に移動します。

  「選択されたノード(Selected Nodes)」: 「ノード(Node)」リストから希望のノードを選択できます。また、「すべて追加(Add All)」ボタンを使用して、すべてのノードを選択することもできます。これは、「クラスタ内に現在あるすべてのノード(All Nodes Currently in Cluster)」を選択するのと同じです。

2. 「OK」をクリックします。

GUI を使った CXFS ファイルシステムのマウント

既存のファイルシステムをクラスタ内のすべてのノードにマウントするには、以下を実行します。

1. 「マウントするファイルシステム(Filesystem to Mount)」: ファイルシステムをプルダウン・メニューから選択します。

2. 「OK」をクリックします。

CXFS サービスがアクティブではない場合、ファイルシステムのマウントは完全には成功しません。ファイルシステムはマウント準備済みとしてマークされ、「ファイルシステムのマウント(Mount a Filesystem)」タスクに警告メッセージが表示されます。ファイルシステムは、CXFS サービスを開始するまで、実際にマウントされません。詳細については、125 ページの「GUI を使った CXFS サービスの開始」を参照してください。

GUI を使った CXFS ファイルシステムのアンマウント

ファイルシステムをクラスタ内のすべてのノードからアンマウントするには、以下を実行します。

1. 次の情報を入力します。
   - 「アンマウントするファイルシステム(Filesystem to Unmount)」: アンマウントするファイルシステムを選択します。
   - 「強制(Force)」: IRIX ノードに対し `umount -k` オプションを使用して強制的にアンマウントを実行するには、「強制」チェックボックスをクリックします。`-k` オプションは、該当するファイルシステムのファイルまたは現在のディレクトリを開いているプロセスを強制終了した後、アンマウントします。force オプションは、デフォルトではオフです。詳細については、`umount`(1M) のマン・ページを参照してください。
2. 「OK」をクリックします。

ファイルシステムをローカルでマウント

このタスクを使用して、GUI が接続されているノード(ローカル・ノード)にのみファイルシステムをマウントできます。

ファイルシステムをローカルでマウントするには、以下を実行します。

1. 次の情報を入力します。
   - 「マウントするファイルシステム(Filesystem to Mount)」: マウントするファイルシステムを選択します。使用可能なファイルシステムのリスト内容は、XFS スーパーブロックを含むブロック・デバイスを検索することによって決定されます。
   - 「マウント・ポイント(Mount Point)」: 選択したファイルシステムがマウントされるディレクトリを指定します。
   - (オプション)「マウント・オプション(Mount Options)」: `mount`(1M) コマンドに渡される必要のあるオプションを指定します。使用可能なオプションについての詳細は、`fstab`(4) のマン・ページを参照してください。

2. デフォルトでは、ファイルシステムは、システムが起動するたびに再マウントされます。ただし、このボックスをオフにした場合は、このタスクを明示的に使用した場合にのみマウントが行われます。
3. 「OK」をクリックします。

詳細については、mount(1M) のマン・ページを参照してください。

ローカル・ファイルシステムのアンマウント

ファイルシステムをローカル・ノードからアンマウントするには、以下を実行します。

注意: ファイルシステムをアンマウントすると、そのファイルシステムに関連付けられているボリュームが再構築され、データが失われる可能性があります。

1. 次の情報を入力します。
   - 「アンマウントするファイルシステム(Filesystem to Unmount)」: アンマウントするファイルシステムを選択します。
   - 「マウント情報の削除(Remove Mount Information)」: /etc/fstab ファイルからマウント・ポイントを削除するには、このチェックボックスをクリックします。すると、ファイルシステムは、次回再起動されてもアンマウントされたままになります。このアイテムは、マウント・ポイントが /etc/fstab に現在保存されている場合にのみ使用可能です。
2. 「OK」をクリックします。

GUI を使った CXFS ファイルシステムの削除

現在マウントされているファイルシステムは、削除できません。ファイルシステムをアンマウントするには、138 ページの「GUI を使った CXFS ファイルシステムのアンマウント」を参照してください。

アンマウントされたファイルシステムを完全に削除するには、以下を実行します。

1. 「削除するファイルシステム(Filesystem to Delete)」: ファイルシステムの名前をプルダウン・リストから選択します。
2. 「OK」をクリックします。

ファイルシステム・マウント情報の削除

このタスクを使用して、ローカル・ファイルシステムのマウント情報を /etc/fstab から削除できます。

メモ: ファイルシステムは、ボリュームに存在したままとなります。

以下を実行します。

1. 「ファイルシステム名(Filesystem Name)」: マウント情報を削除するファイルシステムを選択します。使用可能なファイルシステムのリスト内容は、XFS スーパーブロックを含むブロック・デバイスを検索することによって決定されます。
2. 「OK」をクリックします。

GUI を使った CXFS ファイルシステムのメタデータ・サーバの再設定

メモ: 再設定機能は完全に実装されていますが、関連する問題が多数存在するため、CXFS ではサポートされていません。これらの機能を使用してもデータの整合性が損なわれることはありませんが、クラスタ・ノードのパニックやハングが発生する可能性があります。この機能は、このような問題が解決された時点で完全にサポートされる予定です。再設定の機能は、このリリースではデフォルトで無効になっています。

`cxfs_relocation_ok` systune を使用してカーネルで再設定が明示的に有効になっている場合、ファイルシステムのメタデータ・サーバは、リスト内のメタデータ・サーバとして使用可能なその他のノードに再設定することができます。ファイルシステムは、GUI が接続されているシステムにマウントされている必要があります。

1. 次の情報を入力します。
   - 「ファイルシステム(Filesystem)」: 希望のファイルシステムをリストから選択します。
   - 「現在のメタデータ・サーバ(Current Metadata Server)」: 現在のメタデータ・サーバが表示されます。
   - 「新しいメタデータ・サーバ(New Metadata Server)」: 希望のノードをリストから選択します。

     選択されたサーバでは、「CXFS サービスの開始(Start CXFS Services)」タスクを実行すると、選択したファイルシステムへのアクセスを調整する役割を果たします。125 ページの「GUI を使った CXFS サービスの開始」を参照してください。
2. 「OK」をクリックして、タスクを完了します。

CXFS カーネル・メンバーシップには、再設定による影響はありません。ただし、メタデータ・サーバの再設定中は、ファイルシステムのパフォーマンスが低下することがあります。

CXFS の cmgr タスク・リファレンス

この章では、以下の内容について説明します。

- 142 ページの「cmgr(1M) 概要」
- 147 ページの「cmgr を使ったデフォルトの設定」
- 148 ページの「cmgr を使ったノード・タスク」
- 166 ページの「cmgr を使ったクラスタ・タスク」
- 172 ページの「cmgr を使ったクラスタ・サービス・タスク」
- 177 ページの「cmgr を使った CXFS ファイルシステム・タスク」
- 189 ページの「cmgr を使ったスイッチと I/O 制限タスク」
- 194 ページの「スクリプトの例」
- 196 ページの「cmgr スクリプトの自動作成」

クラスタを設定するために実行しなければならないタスクの概要については、77 ページの「cmgr(1M) コマンドを使った設定」を参照してください。

タスクは、特定の段階を追って実行する必要があります。たとえば、パーティション ID を変更するには、そのノード名をまず識別しなければなりません。

また、cluster_status ツールを使用して、curses モードでステータスを表示することもできます。237 ページの 第9章「ステータスのモニタ」を参照してください。

メモ: CXFS を正しく実行するには、各ノードにライセンスがインストールされていなければなりません。ライセンスを正しくインストールせずにソフトウェアをインストールすると、cmgr(1M) コマンドは使用できません。ライセンスについての詳細は、43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。ライセンス、および IRIX 以外のオペレーティング・システムが実行されているノードについての詳細は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

---

cmgr(1M) 概要

cmgr(1M) コマンドを使用するには、CXFS 管理ノード上で root としてログインしている必要があります。続いて以下のいずれかを入力します。

```
# /usr/cluster/bin/cmgr
```

または

```
# /usr/cluster/bin/cluster_mgr
```

このコマンドを入力すると、次のメッセージとコマンド・プロンプト (cmgr>) が表示されます。

```
Welcome to SGI Cluster Manager Command-Line Interface

cmgr>
```

ヘルプの利用

コマンド・プロンプトが表示されたら、以下のサブコマンドを入力できます。? または help と入力すると、いつでも cmgr のヘルプを表示できます。

プロンプト・モードの使用

cmgr(1M) の -p オプションは、CXFS コンポーネントの定義と変更を行う管理コマンドに必要な値を入力するよう求めるプロンプトを表示します。プロンプト・モードでの実行は、以下のいずれかの方法で行うことができます。

- コマンド・ラインで -p オプションを指定します。

  ```
  # cmgr -p
  ```

- 次の例のように、cmgr を起動した後で set prompting on コマンドを実行します。

  ```
  cmgr> set prompting on
  ```

  この方法を使用した場合は、個々のサブコマンドを実行するときにプロンプト・モードとそれ以外のモードを切替えることができます。プロンプト・モードを終了するには、次を入力します。

  ```
  cmgr> set prompting off
  ```

以下に、プロンプト・モードで尋ねられる可能性がある質問の例を示します(実際に尋ねられる質問は、前の質問への回答に応じて変わります)。

```
cmgr> define node nodename
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ?
Is this a FailSafe node <true|false> ?
Is this a CXFS node <true|false> ?
Operating System <IRIX|Solaris|Windows> ?
Node ID ?[optional]
Partition ID ?[optional] (0)
Do you wish to define failure hierarchy[y/n]:
Reset type <powerCycle> ? (powerCycle)
Do you wish to define system controller info[y/n]:
Sysctrl Type <msc|mmsc|l2|l1>? (msc)
Sysctrl Password[optional] ? ( )
Sysctrl Status <enabled|disabled> ?
Sysctrl Owner ?
Sysctrl Device ?
Sysctrl Owner Type <tty> ? (tty)
NIC 1 - IP Address ?
```

このタスクについての詳細は、148 ページの「cmgr を使ったノードの定義」を参照してください。

アクションの完了とキャンセル

クラスタのコンポーネントを作成または変更する場合は、以下のいずれかのコマンドを入力できます。

- cancel: 現在のモードを中止して、変更を破棄します。
- done: 現在の定義または変更を実行して、cmgr プロンプトに戻ります。

スクリプト・ファイルの使用

-f オプションを使用して入力ファイルを指定することで、cmgr コマンドを連続して実行できます。

```
cmgr -f input_file
```

または、ファイルの 1 行目に次を含めておき、そのファイルをスクリプトとして実行することができます。

```
#!/usr/cluster/bin/cmgr -f
```

ファイルの各行は、有効な cmgr コマンド・ライン、コメント行 (# で始まる)、または空白行でなければなりません。

メモ: 複数レベルのコマンドを完了するために done コマンド・ラインを含める必要があります。また、ファイルの最後には quit コマンド・ラインを含める必要があります。

入力ファイルのいずれかの行が失敗すると、cmgr は終了します。以下のように、-i オプションを -f オプションと共に使用すると、行の失敗を無視するよう選択して処理を続行できます。

```
cmgr -if input_file
```

または、以下のようにスクリプトの 1 行目にこのオプションを含めることもできます。

```
#!/usr/cluster/bin/cmgr -if
```

メモ: cmgr コマンド・ラインをスクリプトの 1 行目として使用するときに -i を含める場合は、必ずまったく同じ構文 (-if) を使用してください。

たとえば、ファイル /tmp/showme に次が含まれているとします。

```
cxfs6# more /tmp/showme
show clusters
show nodes in cluster cxfs6-8
quit
```

この場合は、次のコマンドを実行して出力を得ることができます。

```
cxfs6# /usr/cluster/bin/cmgr -if /tmp/showme

1 Cluster(s) defined
        cxfs6-8


Cluster cxfs6-8 has following 3 machine(s)
        cxfs6
        cxfs7
        cxfs8
```

または、スクリプトの 1 行目に cmgr コマンド・ラインを含めて、そのスクリプトに実行アクセス権を指定し、showme 自体を実行できます。

```
cxfs6# more /tmp/showme
#!/usr/cluster/bin/cmgr -if
#
show clusters
show nodes in cluster cxfs6-8
quit

cxfs6# /tmp/showme

1 Cluster(s) defined
        cxfs6-8


Cluster cxfs6-8 has following 3 machine(s)
        cxfs6
        cxfs7
        cxfs8
```

完全なクラスタを定義する場合についての例は、194 ページの「スクリプトの例」を参照してください。

cmgr 内部からのシェルの呼出し

cmgr(1M) 内部からシェルを呼出すには、次を入力します。

```
cmgr> sh
cxfs6#
```

シェルを終了して cmgr> プロンプトに戻るには、次を入力します。

```
cxfs6# exit
cmgr>
```

コマンド・ラインでのサブコマンドの入力

一部の cmgr サブコマンドは、次の形式を使用してコマンド・ラインに直接入力できます。

```
cmgr -c "subcommand"
```

subcommand には、適切なオペランドと共に以下のいずれかを入力できます。

- admin: ノードのリセットなど、特定のアクションを実行できるようにします。
- delete: クラスタまたはノードを削除します。
- help: ヘルプ情報を表示します。
- show: クラスタまたはノードに関する情報を表示します。
- start: CXFS サービスを開始し、再起動時に CXFS サービスが自動的に再開されるよう設定します。
- stop: CXFS サービスを停止し、再起動時に CXFS サービスが自動的に再開されないよう設定します。
- test: 接続性をテストします。

たとえば、クラスタに関する情報を表示するには、次を入力します。

```
# cmgr -c "show clusters"
1 Cluster(s) defined
      eagan
```

詳細については、cmgr(1M) のマン・ページを参照してください。

テンプレート・スクリプト

/var/cluster/cmgr-templates ディレクトリには、システムのさまざまなコンポーネントを設定するために変更可能な cmgr スクリプトのテンプレートが用意されています。

各テンプレート・ファイルには、特定のオブジェクトを作成するために必要な cmgr コマンドのリストに加え、各フィールドを説明するコメントも含まれています。また、オプションのフィールドのデフォルト値も提供されています。

/var/cluster/cmgr-templates ディレクトリには、クラスタおよびノードを作成するために以下のテンプレートが含まれています。

- cmgr-create-cluster
- cmgr-create-node

CXFS 設定を作成するには、複数のテンプレートを 1 つのファイルに結合して、生成されたスクリプトを実行できます。

メモ: 複数のテンプレート・スクリプトの情報を結合してクラスタ設定を準備する場合、各テンプレート・スクリプトの末尾にある quit は、最後の quit を除いて削除してください。1 つの cmgr スクリプトでは、quit 行は 1 つしか使用しないでください。

たとえば、3 ノード設定の場合、3 つの cmgr-create-node ファイルと、1 つの cmgr-create-cluster ファイルを結合します。

cmgr を使ったデフォルトの設定

デフォルトのクラスタおよびノードを設定して、cmgr の現在のセッションの設定プロセスを簡略化できます。名前を明示的に指定しないと、デフォルトが使用されます。デフォルト値を指定するには、以下のコマンドを使用できます。

```
set cluster  clustername
set node hostname
```

clustername および *hostname* は、論理名です。論理名は、アンダースコア (_) で始めたり、空白として使用される文字を含めたり、255 文字より長くすることはできません。

現在のデフォルトを表示するには、次を使用します。

```
show set defaults
```

例は、次のとおりです。

```
cmgr> set cluster cxfs6-8
cmgr> set node cxfs6
cmgr> show set defaults
Default cluster set to: cxfs6-8

Default node set to: cxfs6
Default cdb set to: /var/cluster/cdb/cdb.db
Default resource_type is not set
Extra prompting is set off
```

cmgr を使ったノード・タスク

この節では、cmgr を使用してノードを定義、変更、削除、表示、およびリセットする方法について説明します。

メモ: クラスタ・データベースに変更を加えると、その都度、すべての CXFS 管理ノードにクラスタ全体のステータス情報が送信されます。したがって、設定内での CXFS 管理ノードの数が多くなるほど送信に時間がかかります。

---

cmgr を使ったノードの定義

ノードを定義するには、以下のコマンドを使用します。

```
define node logical_hostname
    set hostname to hostname
    set nodeid to nodeID
    set node_function to server_admin|client_admin|client_only
    set partition_id to partitionID
    set reset_type to powerCycle
    set sysctrl_type to msc|mmsc|l2|l1 (based on node hardware)
    set sysctrl_password to password
    set sysctrl_status to enabled|disabled
    set sysctrl_owner to node_sending_reset_command
    set sysctrl_device to /dev/ttyd2
    set sysctrl_owner_type to tty_device
    set is_failsafe to true|false
    set is_cxfs to true|false
```

```
set operating_system to irix|solaris|windows
set weight to 0|1 (no longer needed)
add nic IP_address_or_hostname (if DNS)
        set heartbeat to true|false
        set ctrl_msgs to true|false
        set priority to integer
remove nic IP_address_or_hostname (if DNS)
set hierarchy to [system][fence][reset][fencereset][shutdown]
```

上記のコマンド書式について、次の点に注意してください。

- *logical_hostname* は、ホスト名（mynode.company.com など）と同じであるか、ホスト名の簡略名（mynode など）、またはまったく別の名前（nodeA など）です。論理名は、アンダースコア (_) で始めたり、空白として使用される文字を含めたり、255 文字より長くすることはできません。
- hostname は、定義されているノードで hostname(1) コマンドを実行することによって返されるホスト名です。プール内のその他のノードでは、/etc/hosts または名前解決メカニズムによって、このホスト名を正しく解決できなければなりません。*hostname* のデフォルトは、*logical_hostname* の値であるため、*logical_hostname* に対してホスト名またはホスト名の簡略名以外の値を使用する場合は、このコマンドの値を入力してください。
- nodeid は、プール内のノード全体で一意な、1～32767 の範囲の整数です。ノードを定義した後にノード ID 番号を変更しないでください。
  - IRIX ノードでは、この値はオプションです。数字を指定しないと、CXFS によって自動的に ID が計算されます。デフォルトの ID は、マシンのシリアル・ナンバーなどのマシン固有の情報を基にした 5 桁の数字で、連続する数字ではありません。
  - Solaris または Windows ノードでは、一意の値を指定する必要があります。
- node_function は、IRIX ノードの機能を指定します。以下のいずれかを入力します。
  - server_admin: メタデータ・サーバとして使用することを希望するノードです。このノードは、cluster_admin 製品をインストールして作成される必要もあります（この結果、ノードでは、クラスタ・データベースを含み、クラスタ管理タスクを実行できます）。実際にアクティブなメタデータ・サーバになるには、特定のファイルシステムのメタデータ・サーバとして使用可能なノードのリストからこのノードを選択しなければなりません。このタイプのノードは、CXFS カーネル・メンバーシップ定足数およびクラスタ・データベース定足数を形成する際にカウントされます。
  - client_admin: cluster_admin 製品がインストールされ、その結果、クラスタ・データベースを含み、クラスタ管理タスクを実行できるが、メタデータ・サーバにはなることのできないノードです。このタイプのノードは、クラスタ・データベース定足数を形成する際にカウントされますが、CXFS カーネル・メンバーシップ定足数を形成する際にはカウントされません。

– client_only: cxfs_client ソフトウェア製品がインストールされるノードです。クライアントのみのノードは、CXFS メタデータ・サーバになったり、クラスタ管理を実行することはできず、メンバーシップ定足数を形成する際にカウントされません。

Solaris および Windows ノードは、自動的に client-only として指定されます。

- partition_id は、パーティションが設定された Origin 3000 システムのパーティションを一意に定義します。set partition_id コマンドはオプションです。Origin 3000 システムにパーティションが設定されていない場合は、このコマンドをスキップしたり、0 を入力することができます。

メモ: Origin 3000 システムでは、mkpart(1M) コマンドを使用してこの値を決定します。

– -n オプションは、パーティション ID をリストします(システムにパーティションが設定されていない場合は 0 になります)。

– -l オプションは、さまざまなパーティションのブリックをリストします(cmgr では *rack#.slot#* 形式を使用してください)。

例は、次のとおりです(読みやすくするために出力を一部省略してあります)。

```
# mkpart -n
Partition id = 1
# mkpart -l
partition: 3 = brick: 003c10 003c13 003c16 003c21 003c24 003c29 ...
partition: 1 = brick: 001c10 001c13 001c16 001c21 001c24 001c29 ...
```

「パーティション ID(Partition ID)」フィールドには、以下のいずれかを入力できます。

```
1
001.10
```

パーティション ID の設定を解除するには、0 または none という値を使用します。

- reset_type は、powerCycle に設定する必要があります。
- sysctrl_type はシステム・コントローラ・タイプで、表5-1 に示すように、IRIX ノードのハードウェアに基づいています。

表5-1 システム・コントローラ・タイプ

l1	l2	mmsc	msc
Origin 300	Origin 3400	SGI 2400 Rackmount	Origin 200
Origin 3200C	Origin 3800	SGI 2800 Rackmount	Onyx2 Deskside
Onyx 300	Origin 300（NUMAlink モジュール付き）	Onyx2 Rackmount	SGI 2100 Deskside
Onyx 3200C	Onyx 3000 シリーズ		SGI 2200 Deskside

- sysctrl_password はシステム・コントローラ・ポートのパスワードで、ノードの root パスワードや PROM パスワードではありません。ノードによっては、システム管理者がこのパスワードを設定していない可能性があります。システム・コントローラ・パスワードを設定または変更する場合は、ご使用のノードのハードウェア・マニュアルを参照してください。

- sysctrl_status では、システム・コントローラに関する情報を指定できますが、この値を disabled（つまり、CXFS ではノードをリセットできないことを意味する）に設定すると、一時的にシリアル・ハードウェア・リセットを無効にできます。CXFS がノードをリセットできるようにするには、enabled と入力します。IRIX 以外のオペレーティング・システムが実行されているノードでは、これを disabled に設定してください。

- sysctrl_device は、/dev/ttyd2 に設定する必要があります。

- sysctrl_owner は、システム・コントローラ・ポートを介してこのノードをリセットできるノードの論理名です。ノードがハートビート・メッセージに応答していないこと、または要求に正しく応答していないことが検出された場合、このノードは別のノードをリセットできます。オーナーのシリアル・ポートの 1 つと、定義されているノードのシステム・コントローラ・ポートがシリアル・ハードウェア・リセット・ケーブルで物理的に接続されている必要があります。オーナーは、プール内のノードにしてください。まだ定義されていないノードの名前を指定できますが、オーナーは、ノードの接続性診断タスクを実行する前、さらにクラスタがアクティブになる前に、ノードとして定義する必要があります。

- sysctrl_owner_type は、システム・コントローラが接続されているオーナー・ノードの端末ポート (TTY) の名前（/dev/ttyd2 など）です。一方の端からノードをリモートで制御できるよう、ケーブルのもう一方はこのノードのシステム・コントローラ・ポートに接続します。

- このノードで CXFS だけを実行している場合は、is_cxfs を true に、is_failsafe を false にそれぞれ設定します。同時実行クラスタ内にあるこのノードで CXFS と FailSafe の両方を実行している場合は、両方の値を true に設定してください。

- operating_system は、irix、solaris、または windows に設定できます。Windows NT または Windows 2000 に対しては、windows を使用します。

  is_failsafe を true に設定した場合、operating_system の値は irix と想定されます。solaris または windows を指定した場合、weight は 0 と想定されます。operating_system および is_failsafe または weight に互換性のない値を指定すると、define コマンドは失敗します。

- weight は 0 または 1 のいずれかで、CXFS カーネル・メンバーシップを決定する際に特定の CXFS 管理ノードが持つ票の数を指定します。この情報は、現在では「ノードの機能(Node Function)」フィールドによって設定されるため、このコマンドは必要なくなりました。

  メモ: set weight コマンドを使用して 0 または 1 以外の重みを設定することは可能ですが、このように設定することはお薦めできません。追加の重みは必要ありません。

- nic は、プライベート・ネットワークの IP アドレスまたはホスト名です。ホスト名は、/etc/hosts ファイルで解決されなければなりません。

  ネットワーク・インタフェースは 8 つまで指定できますが、優先後が最も高いネットワークのみが使用されます(フェイルオーバーはありません)。このネットワークはプライベートにする必要があります。16 ページの「プライベート・ネットワーク」を参照してください。

  ネットワークの優先度は、クラスタ内の各ノードに対して同じでなければなりません。ホスト名の使用についての詳細は、51 ページの「ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf」を参照してください。

- hierarchy は、異常時のアクション階層を定義します。これにより、異常ノードの処理方法が決まります。最大 3 つのオプションを指定できます。2 番目のオプションは最初のオプションが失敗した場合にのみ実行され、3 番目のオプションは、最初のオプションと 2 番目のオプションの両方が失敗した場合にのみ実行されます。オプションは、空白ではなくカンマで区切る必要があります。以下に、選択できるオプションを示します。

  - system: ノードに関するすべての階層情報をデータベースから削除し、システムのデフォルトが使用されるようにします。システムのデフォルトは、reset,shutdown を入力した場合と同じです。つまり、IRIX ノードに対してリセットが実行され、リセットが失敗した場合、またはノードで Solaris または Windows が実行されている場合は、CXFS サービスが停止されます。system オプションを指定した場合、その他の階層オプションは指定できません。

  - fence: 問題のあるノードからストレージ・エリア・ネットワーク (SAN: Storage Area Network) へのアクセスを無効にします。制限機能を使用すると、CXFS カーネル・メンバーシップをリセットより高速に回復することができます。このアクションは、すべてのノードに対して使用できます。

    IRIX ノードでは、CXFS が単にノードの機能の一部で、システムを再起動するよりも CXFS にアクセスできなくなった方が問題が少ない場合(CXFS クライアントでもある大規模なコンピュータ・

サーバの場合など)、データの整合性を保護するために I/O 制限を使用することができます。CXFS が主要なアクティビティで、早急にオンラインに戻す必要がある場合 (CXFS ファイルサーバの場合など) は、IRIX ノードで I/O 保護のためにシリアル・ハードウェア・リセットを使用することができます。

Solaris および Windows ノードでは、データの整合性を保護するための唯一の選択肢は、I/O 制限です。

メモ: Solaris および Windows ノードでは、データの整合性を保護するために I/O 制限が必要です。I/O 制限をサポートするには、SGI が販売、サポートする Brocade スイッチが必須であるため、マルチ OS CXFS クラスタには Brocade スイッチが必要です。

- `fencereset`: 問題のあるノードから SAN へのアクセスを無効にします。そのノードが正常に制限された場合は、IRIX ノードの非同期リセットも実行します。リセット確認を待たずに回復が開始されます。リセット・アクションは、IRIX ノードのみに対して使用できます。
- `reset`: システム・コントローラに接続されているシリアル回線を通じてリセットを実行します。このアクションは、IRIX ノードのみに対して使用できます。
- `shutdown`: CXFS カーネル・メンバーシップが失われた場合に、ノードの CXFS カーネルベースのサービスを停止します。正常なクラスタは、ノードにシャットダウンする余裕を与えるために回復の開始を遅らせます。このアクションは、すべてのノードに対して使用できます。

制限アクションが失敗した場合のみリセットを実行するには、次を指定します。

```
set hierarchy fence,reset
```

メモ: `shutdown` が指定されておらず、その他のアクションが失敗した場合、CXFS サービスは、CXFS カーネル・メンバーシップを提供するノードにより、ローカルで強制的にシャットダウンされます。

制限および非同期リセットを実行するには、次を指定します。

```
set hierarchy fencereset
```

システムのデフォルト (`reset,shutdown`) に戻すには、次を指定します。

```
set hierarchy system
```

詳細については、28 ページの「定足数の計算」および 109 ページの「GUI を使ったノードの定義」を参照してください。

デフォルト情報を使用するには、プロンプト・モードで <Enter> キーを押します。例では、<Enter> キーは示されていません。一般情報については、109 ページの「GUI を使ったノードの定義」を参照してください。以下に、プロンプトの概要を示します。

```
cmgr> define node logical_hostname
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ? hostname
Is this a FailSafe node <true|false> ? true|false
Is this a CXFS node <true|false> ? true
Node Function <server_admin|client_admin|client_only> ? node_function
Operating System <IRIX|Solaris|Windows> ?OS_type
Node ID ?[optional] node_ID
Partition ID ?[optional] (0)partition_ID
Do you wish to define failure hierarchy[y/n]:y|n
Do you wish to define system controller info[y/n]:y|n
Reset type <powerCycle> ? (powerCycle)
Do you wish to define system controller info[y/n]:y|n
Sysctrl Type <msc|mmsc|l2|l1>? (msc) model (based on node hardware)
Sysctrl Password[optional] ? ( )password
Sysctrl Status <enabled|disabled> ? enabled|disabled
Sysctrl Owner ? node_sending_reset_command
Sysctrl Device ? /dev/ttyd2
Sysctrl Owner Type <tty> ? (tty) tty_device
NIC 1 - IP Address ? IP_address_or_hostname (if DNS)
```

通常モードの場合:

```
# /usr/cluster/bin/cmgr
Welcome to SGI Cluster Manager Command-Line Interface

cmgr> define node foo
Enter commands, you may enter "done" or "cancel" at any time to exit

? set is_failsafe to false
? set is_cxfs to true
? set operating_system to irix
? set node_function to server_admin
? set hierarchy to fencereset,reset
? add nic 111.11.11.111
Enter network interface commands, when finished enter "done" or "cancel"

NIC 1 - set heartbeat to true
NIC 1 - set ctrl_msgs to true
NIC 1 - set priority to 1
NIC 1 - done
```

```
? done
```

プロンプト・モードの場合:

```
# /usr/cluster/bin/cmgr -p
Welcome to SGI Cluster Manager Command-Line Interface

cmgr> define node foo
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ?
Is this a FailSafe node <true|false> ? false
Is this a CXFS node <true|false> ? true
Operating System <IRIX|Solaris|Windows> ? irix
Node Function <server_admin|client_admin|client_only> server_admin
Node ID[optional]?
Partition ID ? [optional] (0)
Do you wish to define failure hierarchy[y|n]:y
Hierarchy option 0 <System|FenceReset|Fence|Reset|Shutdown>[optional] ? fencereset
Hierarchy option 1 <System|FenceReset|Fence|Reset|Shutdown>[optional] ? reset
Hierarchy option 2 <System|FenceReset|Fence|Reset|Shutdown>[optional] ?
Reset type <powerCycle>  ? (powerCycle)
Do you wish to define system controller info[y/n]:n
NIC 1 - IP Address ? 111.11.11.111
```

以下に、プロンプト・モードで Solaris ノードを定義する場合の例を示します(Solaris ノードであるため、デフォルトの ID は提供されません。また、client_only でなければならないため、ノードの機能を指定するようプロンプトは表示されません。

```
cmgr> define node solaris1
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ?
Is this a FailSafe node <true|false> ? false
Is this a CXFS node <true|false> ? true
Operating System <IRIX|Solaris|Windows> ? solaris
Node ID ? 7
Do you wish to define failure hierarchy[y/n]:y
Hierarchy option 0 <System|FenceReset|Fence|Reset|Shutdown>[optional] ? fence
Hierarchy option 1 <System|FenceReset|Fence|Reset|Shutdown>[optional] ?
NIC 1 - IP Address ? 163.154.18.172
```

cmgr を使ったノードの変更

既存のノードを変更するには、以下のコマンドを入力します。

```
modify node logical_hostname
    set hostname to hostname
    set nodeid to nodeID
    set partition_id to partitionID
    set reset_type to powerCycle
    set sysctrl_type to msc|mmsc|l2|l1 (based on node hardware)
    set sysctrl_password to password
    set sysctrl_status to enabled|disabled
    set sysctrl_owner to node_sending_reset_command
    set sysctrl_device to /dev/ttyd2
    set sysctrl_owner_type to tty_device
    set is_failsafe to true|false
    set is_cxfs to true|false
    set weight to 0|1
    add nic IP_address_or_hostname (if DNS)
            set heartbeat to true|false
            set ctrl_msgs to true|false
            set priority to integer
    remove nic IP_address_or_hostname (if DNS)
```

```
set hierarchy to [System] [Fence][Reset][FenceReset][Shutdown]
```

メモ: 特定のノードの operating_system の設定を変更することはできません。変更すると、エラーが発生します。誤って不正なオペレーティング・システムを指定した場合は、該当するノードを削除してから再度定義する必要があります。

ノードの機能は変更できません。IRIX ノードのノードの機能を変更するには、そのノードを削除してから再定義する必要があります(必要に応じてソフトウェア製品を再インストールしなければならない場合もあります)。Solaris または Windows ノードのノードの機能は、常に client_only です。

コマンドは、ノードの定義に使用されるものと同じです。ノード定義時に指定した情報は、ノード ID を除いてすべて変更可能です。コマンドについての詳細は、148 ページの「cmgr を使ったノードの定義」を参照してください。

注意: ノードを定義した後にノード ID 番号を変更しないでください。

パーティションの設定例

以下に、Origin 3000 システムのパーティション設定の例を示します。

```
# cmgr
Welcome to SGI Cluster Manager Command-Line Interface

cmgr> modify node n_preston
Enter commands, when finished enter either "done" or "cancel"

n_preston ? set partition_id to 1
n_preston ? done

Successfully modified node n_preston
```

プロンプト表示ありでこの機能を実行するには、以下を入力します。

```
# cmgr -p
Welcome to SGI Cluster Manager Command-Line Interface

cmgr> modify node n_preston
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ? (preston.engr.sgi.com)
Is this a FailSafe node <true|false> ? (true)
Is this a CXFS node <true|false> ? (true)
Node ID[optional] ? (606)
Partition ID[optional] ? (0) 1
Do you wish to modify failure hierarchy[y/n]:n
Reset type <powerCycle> ? (powerCycle)
Do you wish to modify system controller info[y/n]:n
NIC 1 - IP Address ? (preston)

Successfully modified node n_preston

cmgr> show node n_preston
Logical Machine Name: n_preston
Hostname: preston.engr.sgi.com
Operating System: IRIX
Node Is FailSafe: true
Node Is CXFS: true
Node Function: client_admin
Nodeid: 606
Partition id: 1
Reset type: powerCycle
ControlNet Ipaddr: preston
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 1
```

パーティション ID の設定を解除するには、0 または none という値を使用します。

異常階層の変更

以下に、ノード perceval の異常階層をシステムのデフォルトから fencereset,reset,shutdown に変更し、再びシステム・デフォルトに戻す場合の例を示します。

```
cmgr> modify node perceval
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ? (perceval.engr.sgi.com)
Is this a FailSafe node <true|false> ? (false)
Is this a CXFS node <true|false> ? (true)
Node ID[optional] ? (803)
Partition ID[optional] ? (0)
Do you wish to modify failure hierarchy[y/n]:y
Hierarchy option 0 <System|FenceReset|Fence|Reset|Shutdown>[optional] ?fencereset
Hierarchy option 1 <System|FenceReset|Fence|Reset|Shutdown>[optional] ?reset
Hierarchy option 2 <System|FenceReset|Fence|Reset|Shutdown>[optional] ?shutdown
Reset type <powerCycle> ? (powerCycle)
Do you wish to modify system controller info[y/n]:n
NIC 1 - IP Address ? (163.154.18.173)

Successfully modified node perceval

cmgr> show node perceval
Logical Machine Name: perceval
Hostname: perceval.engr.sgi.com
Operating System: IRIX
Node Is FailSafe: false
Node Is CXFS: true
Node Function: client_admin
Nodeid: 803
Node Failure Hierarchy is: FenceReset Reset Shutdown
Reset type: powerCycle
ControlNet Ipaddr: 163.154.18.173
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 1
```

システムのデフォルトに戻すには、以下を入力します。

```
cmgr> modify node perceval

Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ? (perceval.engr.sgi.com)
Is this a FailSafe node <true|false> ? (false)
```

```
Is this a CXFS node <true|false> ? (true)
Node ID[optional] ? (803)
Partition ID[optional] ? (0)
Do you wish to modify failure hierarchy[y/n]:y
Hierarchy option 0 <System|FenceReset|Fence|Reset|Shutdown>[optional] ?
(FenceReset) system
Reset type <powerCycle> ? (powerCycle)
Do you wish to modify system controller info[y/n]:n
NIC 1 - IP Address ? (163.154.18.173)
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false> ? (true)
NIC 1 - (use network for control messages) <true|false> ? (true)
NIC 1 - Priority <1,2,...> ? (1)

cmgr> show node perceval
Logical Machine Name: perceval
Hostname: perceval.engr.sgi.com
Operating System: IRIX
Node Is FailSafe: false
Node Is CXFS: true
Node Function: client_admin
Nodeid: 803
Reset type: powerCycle
ControlNet Ipaddr: 163.154.18.173
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 1
```

メモ: 異常階層に対してシステムのデフォルトを設定した場合、show ではステータスは表示されません。

cmgr を使ったノードのリセット

CXFS が実行されている場合、次のコマンドを使用して IRIX ノードをリセットできます。

```
admin reset node hostname
```

このコマンドは、CXFS デーモンを使用して、指定したノードをリセットします。

CXFS デーモンが実行されていない場合でも、admin reset コマンドの standalone オプションを使用すると、クラスタ内のノードをリセットできます。

```
admin reset standalone node hostname
```

リセット・デバイス名とタイプおよびシステム・コントローラのタイプがデータベースでまだ定義されていない場合（たとえば、リセットを実行するノードを定義する前にリセットをテストしたい場合など）、以下のコマンドを使用して、システム・コントローラへの接続またはノードのリセットを実行できます。

```
admin ping dev_name tty of dev_type /dev/ttyd2 with sysctrl_type msc|mmsc|l2|l1

admin reset dev_name tty of dev_type /dev/ttyd2 with sysctrl_type msc|mmsc|l2|l1
```

コマンド要素についての詳細は、148 ページの「cmgr を使ったノードの定義」を参照してください。

このコマンドは、CXFS デーモンを経由しません。

cmgr を使ったノードの CXFS または FailSafe への切替え

既存の FailSafe ノードを CXFS にも適用するよう切替えるには、modify コマンドを使用して設定を変更します。

メモ: 対応する高可用性 (HA: High Availability) サービスまたは CXFS サービスがアクティブな場合は、ノードの FailSafe または CXFS をオフにできません。まず、ノードのサービスを停止してください。

通常モードの場合:

```
cmgr> modify node cxfs6
Enter commands, when finished enter either "done" or "cancel"

cxfs6 ? set is_FailSafe to true
cxfs6 ? done

Successfully modified node cxfs6
```

プロンプト・モードの場合:

```
cmgr> modify node cxfs6
Enter commands, you may enter "done" or "cancel" at any time to exit

Hostname[optional] ? (cxfs6.americas.sgi.com)
Is this a FailSafe node <true|false> ? (false) true
Is this a CXFS node <true|false> ? (true)
Node ID[optional] ? (13203)
Partition ID[optional] ? (0)
Do you wish to modify failure hierarchy[y/n]:n
Reset type <powerCycle> ? (powerCycle)
Do you wish to modify system controller info[y/n]:n
```

```
NIC 1 - IP Address ? (163.154.18.172)
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false> ? (true)
NIC 1 - (use network for control messages) <true|false> ? (true)
NIC 1 - Priority <1,2,...> ? (1)

Successfully modified node cxfs6
```

cmgr を使ったノードの削除

ノードを削除するには、次のコマンドを使用します。

```
delete node hostname
```

ノードを削除できるのは、現在そのノードがクラスタの一部でない場合だけです。クラスタに現在ノードが含まれている場合は、そのクラスタをまず変更して、ノードを削除しなければなりません。

たとえば、次のように設定されている cxfs6-8 というクラスタがあるとします。

```
cmgr> show cluster cxfs6-8
Cluster Name: cxfs6-8
Cluster Is FailSafe: true
Cluster Is CXFS: true
Cluster ID: 20
Cluster HA mode: normal
Cluster CX mode: normal


Cluster cxfs6-8 has following 3 machine(s)
        cxfs6
        cxfs7
        cxfs8
```

cxfs8 ノードを削除するには、プロンプト・モードで以下を実行します（CXFS サービスがそのノードで停止されていると想定しています）。

```
cmgr> modify cluster cxfs6-8
Enter commands, when finished enter either "done" or "cancel"

Is this a FailSafe cluster <true|false> ? (false)
Is this a CXFS cluster <true|false> ? (true)
Cluster Notify Cmd [optional] ?
Cluster Notify Address [optional] ?
Cluster CXFS mode <normal|experimental>[optional] ? (normal)
```

```
Cluster ID ? (20)

Current nodes in cluster cxfs6-8:
Node - 1: cxfs6
Node - 2: cxfs7
Node - 3: cxfs8

Add nodes to or remove nodes from cluster cxfs6-8
Enter "done" when completed or "cancel" to abort

cxfs6-8 ? remove node cxfs8
cxfs6-8 ? done
Successfully modified cluster cxfs6-8

cmgr> show cluster cxfs6-8
Cluster Name: cxfs6-8
Cluster Is FailSafe: false
Cluster Is CXFS: true
Cluster ID: 20
Cluster CX mode: normal


Cluster cxfs6-8 has following 2 machine(s)
        cxfs6
        cxfs7
```

プールから cxfs8 を削除するには、次を入力します。

```
cmgr> delete node cxfs8

IMPORTANT: NODE cannot be deleted if it is a member of a cluster.
The LOCAL node can not be deleted if some other nodes are still defined.


Deleted machine (cxfs6).
```

cmgr を使ったノードの表示

ノードを定義したら、次のコマンドを使用して、そのノードのパラメータを表示できます。

```
show node hostname
```

例は、次のとおりです。

```
cmgr> show node cxfs6
Logical Machine Name: cxfs6
Hostname: cxfs6.americas.sgi.com
Operating System: IRIX
Node Is FailSafe: false
Node Is CXFS: true
Node Function: server_admin
Nodeid: 13203
Reset type: powerCycle
ControlNet Ipaddr: 163.154.18.172
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 1
```

定義されているすべてのノードのリストは、次のコマンドを使用して表示できます。

```
show nodes in pool
```

例は、次のとおりです。

```
cmgr> show nodes in pool

3 Machine(s) defined
        cxfs8
        cxfs6
        cxfs7
```

指定したクラスタに対して定義されているすべてのノードのリストは、次のコマンドを使用して表示できます。

```
show nodes [in cluster clustername]
```

例は、次のとおりです。

```
cmgr> show nodes in cluster cxfs6-8

Cluster cxfs6-8 has following 3 machine(s)
        cxfs6
        cxfs7
        cxfs8
```

デフォルトのクラスタが指定されている場合は、このコマンドを使用する際にクラスタを指定する必要はありません。例は、次のとおりです。

```
cmgr> set cluster cxfs6-8
cmgr> show nodes

Cluster cxfs6-8 has following 3 machine(s)
        cxfs6
        cxfs7
        cxfs8
```

cmgr を使ったノードの接続性のテスト

cmgr を使用して、クラスタでのネットワークの接続性をテストすることができます。このテストでは、指定したノードが、ノードで設定されている各インタフェースを通じて互いに通信できるかどうかをチェックします。CXFS が実行されている場合、このテストは行われません。このテストを実行するには、/etc/.rhosts ファイルが正しく設定されている必要があります。58 ページの「(オプション) /.rhosts」を参照してください。

クラスタ内のノードに対するネットワークの接続性をテストするには、次のコマンドを使用します。

```
test connectivity in cluster clustername [on node nodename1 node nodename2 ...]
```

例は、次のとおりです。

```
cmgr> test connectivity in cluster cxfs6-8 on node cxfs7
Status: Testing connectivity...
Status: Checking that the control IP_addresses are on the same networks
Status: Pinging address cxfs7 interface ef0 from node cxfs7 [cxfs7]
Notice: overall exit status:success, tests failed:0, total tests executed:1
```

このテストでは、最初のエラーが発生するとエラー・メッセージが表示され、応答しなかったノードが示されます。このテストを実行した後でエラー・メッセージを受取った場合は、ifconfig コマンドを使用して、ネットワーク・インタフェースが設定されていることを確認します。例は、次のとおりです(改行して読みやすくしてあります)。

```
# /usr/etc/ifconfig ef0
ef0: flags=405c43 <UP,BROADCAST,RUNNING,FILTMULTI,MULTICAST,CKSUM,DRVRLOCK,IPALIAS>
inet 128.162.89.39 netmask 0xffff0000 broadcast 128.162.255.255
```

出力の最初の行にある UP は、インタフェースが設定されていることを示します。

ネットワーク・インタフェースが設定されている場合は、ネットワーク・ケーブルが正しく接続されていることを確認し、テストを再び実行します。

cmgr を使ったシリアル接続のテスト

64 ページの「CXFS 管理ノードのシリアル・ハードウェア・リセット接続」を参照してください。

cmgr を使ったクラスタ・タスク

この節では、cmgr(1M) を使用してクラスタを定義、変更、削除、および表示する方法について説明します。また、CXFS サービスを開始および停止する方法についても説明します。

cmgr を使ったクラスタの定義

cmgr を使用してクラスタを定義するときは、同じコマンドを使用して、クラスタの定義およびクラスタへのノードの追加を行います。一般情報については、122 ページの「GUI を使ったクラスタの定義」を参照してください。

クラスタを定義するには、以下のコマンドを使用します。

```
define cluster clustername
    set is_failsafe to true|false
    set is_cxfs to true|false
    set clusterid to clusterID
    set notify_cmd to notify_command
    set notify_addr to email_address
    set ha_mode to normal|experimental
    set cx_mode to normal|experimental
    add node node1name
    add node node2name
    ...
```

上記のコマンド書式について、次の点に注意してください。

- cluster は、クラスタの論理名です。論理名は、アンダースコア (_) で始めたり、空白として使用される文字を含めたり、255 文字より長くすることはできません。ネットワークを共有するクラスタには、一意の名前を付ける必要があります。
- CXFS だけを実行している場合は、is_cxfs を true に、is_failsafe を false にそれぞれ設定します。同時実行クラスタを実行している場合は、両方の値を true に設定します。
- clusterid は、ネットワーク内で一意な 1〜128 の範囲の数字です。クラスタ ID は、ネットワーク上に存在する可能性のあるほかのクラスタからクラスタ情報を受付けないようにするために、オペレーティング・システム・カーネルによって使用されます。カーネルは通信にデータベースを使用しないため、クラスタ通信を確認するためにクラスタ ID が必要です。カーネルのこの情報は、いったん初期

化したら変更できません。このため、クラスタを定義したらクラスタ ID は変更しないでください。ネットワークを共有するクラスタには、一意の ID が必要です。

- notify_cmd は、ノードまたはクラスタのステータスが変更されるたびに実行されるコマンドです。
- notify_addr は、クラスタおよびノードのステータス変更が通知されるアドレスです。複数のアドレスを指定するには、各アドレスをカンマで区切ります。CXFS では、ノードまたはクラスタのステータスが変更されるたびに、電子メールがそれらのアドレスに送信されます。アドレスを指定しなかった場合、通知は送信されません。notify_addr コマンドを使用する場合は、*notify_command* として電子メール・プログラム(/usr/sbin/Mail など)を指定する必要があります。
- set ha_mode コマンドおよび set cx_mode コマンドは、通常、normal に設定します。set cx_mode コマンドは CXFS にのみ適用し、set ha_mode コマンドは IRIS FailSafe にのみ適用します。

以下に、プロンプト表示ありのコマンドを示します。

```
cmgr> define cluster clustername
Enter commands, you may enter "done" or "cancel" at any time to exit

Is this a FailSafe cluster <true|false> ? true|false
Is this a CXFS cluster  <true|false> ? true|false
Cluster Notify Cmd [optional] ?
Cluster Notify Address [optional] ?
Cluster CXFS mode <normal|experimental>[optional] use_default_of_normal
Cluster ID ? cluster_ID
No nodes in cluster clustername

Add nodes to or remove nodes from cluster clustername
Enter "done" when completed or "cancel" to abort

clustername ? add node node1name
clustername ? add node node2name
...
clustername ? done
Successfully defined cluster clustername
```

クラスタは、デフォルトの normal モードに設定します。experimental にモードを設定すると、ノード異常を発生させることなくクラスタをデバッグできるよう、CXFS カーネル・メンバーシップ・コードでハートビートがオフになります。これは、短時間だけネットワークを切断したい場合(クラスタ・ネットワーク・アクティビティがほかになく、ハートビートがなくても異常が検出されるという条件で)に便利です。ただし、実作業用のクラスタでは experimental モードを使用しないでください。このモードは、日本 SGI のカスタマ・サポートから指示があった場合にのみ使用してください。SGI では、カスタマが experimental を使用することを推奨していません。

例は、次のとおりです。

```
cmgr> define cluster cxfs6-8
Enter commands, you may enter "done" or "cancel" at any time to exit

Is this a FailSafe cluster <true|false> ? false
Is this a CXFS cluster  <true|false> ? true
Cluster Notify Cmd [optional] ?
Cluster Notify Address [optional] ?
Cluster CXFS mode <normal|experimental>[optional]
Cluster ID ? 20

No nodes in cluster cxfs6-8

Add nodes to or remove nodes from cluster cxfs6-8
Enter "done" when completed or "cancel" to abort

cxfs6-8 ? add node cxfs6
cxfs6-8 ? add node cxfs7
cxfs6-8 ? add node cxfs8
cxfs6-8 ? done
Successfully defined cluster cxfs6-8
```

プロンプト表示なしでこれを行うには、以下を入力します。

```
cmgr> define cluster cxfs6-8
Enter commands, you may enter "done" or "cancel" at any time to exit

cluster cxfs6-8? set is_cxfs to true
cluster cxfs6-8? set clusterid to 20
cluster cxfs6-8? add node cxfs6
cluster cxfs6-8? add node cxfs7
cluster cxfs6-8? add node cxfs8
cluster cxfs6-8? done
Successfully defined cluster cxfs6-8
```

cmgr を使ったクラスタの変更

コマンドは、以下のとおりです。

```
modify cluster clustername
    set is_failsafe to true|false
```

```
set is_cxfs to true|false
set clusterid to clusterID
set notify_cmd to command
set notify_addr to email_address
set ha_mode to normal|experimental
set cx_mode to normal|experimental
add node node1name
add node node2name
...
remove node node1name
remove node node2name...
```

これらのコマンドは、define cluster コマンドと同じです。詳細については、166 ページの「cmgr を使ったクラスタの定義」および 122 ページの「GUI を使ったクラスタの定義」を参照してください。

メモ: クラスタの名前を変更する場合は、そのクラスタを削除した後、新しいクラスタを定義しなければなりません。CXFS サービスがノードで開始されている場合は、サービスを再起動するか、クラスタの名前を変更するときにクラスタ ID 番号を再使用してください。

ただし、CXFS ファイルシステムがすでに定義されている場合にクラスタの名前を変更すると、CXFS は、それらのファイルシステムをマウントできなくなるので注意してください。詳細については、272 ページの「クラスタの名前を変更後、ファイルシステムをマウントできない」を参照してください。

---

cmgr を使ったクラスタの CXFS または FailSafe への切替え

クラスタを切替えるには、以下のコマンドを使用します。

```
modify cluster clustername
  set is_failsafe to true|false
  set is_cxfs to true|false
  set clusterid to clusterID
```

- cluster は、クラスタの論理名です。論理名は、アンダースコア (_) で始めたり、空白として使用される文字を含めたり、255 文字より長くすることはできません。
- CXFS だけを実行している場合は、is_cxfs を true に、is_failsafe を false にそれぞれ設定します。同時実行クラスタを実行している場合は、両方の値を true に設定します。
- clusterid は、ネットワーク内で一意な 1〜128 の範囲の数字です。クラスタ ID は、ネットワーク上に存在する可能性のあるほかのクラスタからクラスタ情報を受付けないようにするために、オペレーティング・システム・カーネルによって使用されます。カーネルは通信にデータベースを使用しないた

め、クラスタ通信を確認するためにクラスタ ID が必要です。カーネルのこの情報は、いったん初期化したら変更できません。このため、クラスタを定義したらクラスタ ID は変更しないでください。

たとえば、FailSafe にも適用されるよう CXFS クラスタ cxfs6-8 を切替えるには、次を入力します。

```
cmgr> modify cluster cxfs6-8
Enter commands, when finished enter either "done" or "cancel"

cxfs6-8 ? set is_failsafe to true
```

クラスタは、そのノードに関してオンにするすべての機能（FailSafe または CXFS、あるいはその両方）をサポートしていなければなりません。つまり、クラスタがタイプ CXFS の場合は、クラスタの一部であるノードをタイプ FailSafe または CXFS and FailSafe に変更することはできません。ただし、ノードがクラスタのすべての機能をサポートする必要はありません。つまり、タイプ CXFS and FailSafe のクラスタでタイプ CXFS のノードを使用できます。

cmgr を使ったクラスタの削除

クラスタを削除するには、次のコマンドを使用します。

```
delete cluster clustername
```

ただし、ノードが含まれているクラスタは削除できません。したがって、ノードで CXFS サービスをまず停止した後、クラスタを再定義して、ノードが含まれないようにする必要があります。

通常モードの場合:

```
cmgr> modify cluster cxfs6-8
Enter commands, when finished enter either "done" or "cancel"

cxfs6-8 ? remove node cxfs6
cxfs6-8 ? remove node cxfs7
cxfs6-8 ? remove node cxfs8
cxfs6-8 ? done
Successfully modified cluster cxfs6-8

cmgr> delete cluster cxfs6-8

cmgr> show clusters

cmgr>
```

プロンプト・モードの場合:

```
cmgr> modify cluster cxfs6-8
Enter commands, you may enter "done" or "cancel" at any time to exit

Cluster Notify Cmd [optional] ?
Cluster Notify Address [optional] ?
Cluster mode <normal|experimental>[optional] ? (normal)
Cluster ID ? (55)

Current nodes in cluster cxfs6-8:
Node - 1: cxfs6
Node - 2: cxfs7
Node - 3: cxfs8

Add nodes to or remove nodes from cluster cxfs6-8
Enter "done" when completed or "cancel" to abort

cxfs6-8 ? remove node cxfs6
cxfs6-8 ? remove node cxfs7
cxfs6-8 ? remove node cxfs8
cxfs6-8 ? done
Successfully modified cluster cxfs6-8

cmgr> delete cluster cxfs6-8

cmgr> show clusters

cmgr>
```

cmgr を使ったクラスタの表示

クラスタとその内容を表示するには、以下のコマンドを使用します。

```
show clusters
show cluster clustername
```

例は、次のとおりです。

```
cmgr> show cluster cxfs6-8
Cluster Name: cxfs6-8
Cluster Is FailSafe: false
```

```
Cluster Is CXFS: true
Cluster ID: 22
Cluster CX mode: normal


Cluster cxfs6-8 has following 3 machine(s)
        cxfs6
        cxfs7
        cxfs8
```

cmgr を使ったクラスタ・サービス・タスク

以下のタスクでは、CXFS サービスを開始および停止する方法と、ログ・レベルを設定する方法について説明します。

cmgr を使った CXFS サービスの開始

CXFS サービスを開始し、システムが再起動されるたびに CXFS サービスが自動的に再開されるよう設定するには、以下のいずれかのコマンドを使用します。

```
start cx_services [on node hostname ] for cluster clustername

start cx_services for cluster clustername
```

例は、次のとおりです。

```
cmgr> start cx_services for cluster cxfs6-8
```

cmgr を使った CXFS サービスの停止

ノードで CXFS サービスが停止されると、ファイルシステムは、自動的にそのノードからアンマウントされます。

CXFS サービスを一時的に停止する(つまり、再起動するとサービスが再開されるよう設定されている場合に再開できるようにする)には、cmgr 外のシェル・ウィンドウで次のコマンド・ラインを使用します。

```
# /etc/init.d/CXFS stop
```

指定したノードまたはクラスタで CXFS サービスを停止し、再起動によって CXFS サービスが再開されないようにするには、次のコマンドを使用します。

```
stop cx_services [on node hostname]for cluster clustername [force]
```

例は、次のとおりです。

```
cmgr> stop cx_services on node cxfs6 for cluster cxfs6-8

CXFS services have been deactivated on node cxfs6 (cluster cxfs6-8)

cmgr> stop cx_services for cluster cxfs6-8
```

ノードで CXFS サービスを停止すると、そのノードは、クラスタのアクティブなメンバーではなくなります。

注意: CXFS サービスを停止すると、ノードは「非アクティブ(INACTIVE)」としてマークされ、再起動後にクラスタに再設定されなくなります。ノードがクラスタに再設定されるようにするには、cmgr または GUI を使用して CXFS サービスを再開しなければなりません。

cmgr を使ったタイブレーカー・ノードの設定

CXFS タイブレーカー・ノードは、サーバとして使用可能なノードのちょうど半数が互いに通信可能である場合に、CXFS カーネル・メンバーシップ定足数が維持されるかどうかを決定します。デフォルトの CXFS タイブレーカー・ノードはありません。

注意: サーバとして使用可能な 2 つのノードで構成されるクラスタ内の CXFS タイブレーカー・ノードに異常が発生したり、管理者が CXFS サービスを停止した場合は、その他のノードによって強制シャットダウンが実行され、CXFS ファイルシステムはすべてアンマウントされます。

データの整合性を保証するため、サーバとして使用可能なノードが 2 つだけのクラスタでは、シリアル・ハードウェア・リセット機能、あるいは I/O 制限とスイッチの使用が必須です。また、サーバとして使用可能なすべてのノードに対しては、シリアル・ハードウェア・リセット機能、あるいは I/O 制限とスイッチの使用を強くお薦めします。より大きなクラスタでは、サーバとして使用可能なノードを奇数個使用し、ノードの 2 つのみがサーバとして使用可能な場合は、シリアル・ハードウェア・リセット回線を使用するか、I/O 制限とスイッチを使用する必要があります。33 ページの「サーバとして使用可能なノードを 2 つだけ含むクラスタでの回復の問題」を参照してください。

Solaris および Windows ノードでは、データの整合性を保護するために I/O 制限が必要です。I/O 制限をサポートするには、Brocade スイッチが必須であるため、マルチ OS CXFS クラスタには Brocade スイッチが必要です。

CXFS タイブレーカー・ノードを設定するには、以下のように modify コマンドを使用します。

```
modify cx_parameters
[on node nodename] in cluster clustername
set tie_breaker to hostname
```

CXFS タイブレーカー・ノードの設定を解除するには、次のコマンドを使用します。

```
set tie_breaker to none
```

通常モードの場合:

```
cmgr> modify cx_parameters in cluster cxfs6-8
Enter commands, when finished enter either "done" or "cancel"

cxfs6-8 ? set tie_breaker to cxfs8
cxfs6-8 ? done
Successfully modified cx_parameters
```

プロンプト・モードの場合:

```
cmgr> modify cx_parameters in cluster cxfs6-8

(Enter "cancel" at any time to abort)

Tie Breaker Node ? (cxfs7) cxfs8
Successfully modified cx_parameters

cmgr> show cx_parameters in cluster cxfs6-8

_CX_TIE_BREAKER=cxfs8
```

cmgr を使ったログの設定

CXFS ログについての一般情報は、126 ページの「GUI を使ったログの設定」を参照してください。

cmgr を使ったログ・グループ定義の表示

ログ・グループ定義を表示するには、次のコマンドを使用します。

```
show log_groups
```

このコマンドを実行すると、現在定義されているすべてのログ・グループと共に、ログ・グループ名、ログ・レベル、およびログ・ファイルが表示されます。

cmgr を使ったログ・グループの設定

ログ・グループは、次のコマンドを使用して設定できます。

```
define log_group log_group on node adminhostname [in cluster clustername]
  set log_level to log_level
  add log_file log_file
  remove log_file log_file
```

上記のコマンド書式について、次の点に注意してください。

- log_group には、以下のいずれかの値を指定できます。

  clconfd
  cli
  crsd
  diags

- log_level には、以下のいずれかの値が含まれます。
  - 0: ログには記録されません。
  - 1: 重大なエラーと通常の操作の通知をログに記録します(これらのメッセージのログは、SYSLOG ファイルにも記録されます)。
  - 2:「最小(Minimal)」通知と警告をログに記録します。
  - 5〜7: 数字の大きさに応じて、ログに記録される詳細通知が増えます。
  - 10〜19: 数字の大きさに応じて、ログに記録されるデバッグ情報(データ構造を含む)が増えます。
- *log_file*

注意: ログ・ファイルの名前は変更しないでください。名前を変更すると、エラーが発生します。

ログ・レベル 5 を使用してログ・グループ cli をノード cxfs6 で定義する場合:

```
cmgr> define log_group cli on node cxfs6 in cluster cxfs6-8

(Enter "cancel" at any time to abort)

Log Level ? (11) 5

CREATE LOG FILE OPTIONS
```

```
        1) Add Log File.
        2) Remove Log File.
        3) Show Current Log Files.
        4) Cancel. (Aborts command)
        5) Done. (Exits and runs command)

Enter option:5
Successfully defined log group cli
```

cmgr を使ったログ・グループの変更

ログ・グループを変更するには、次のコマンドを使用します。

```
modify log_group log_group_name on node hostname [in cluster clustername]
```

ログ・グループの変更は、定義に使用するコマンドと同じコマンドを使用して行います。

たとえば、cli のログ・レベルを 10 になるよう変更するには、以下を入力します。

```
cmgr> modify log_group cli on node cxfs6 in cluster cxfs6-8

(Enter "cancel" at any time to abort)

Log Level ? (2) 10

MODIFY LOG FILE OPTIONS

        1) Add Log File.
        2) Remove Log File.
        3) Show Current Log Files.
        4) Cancel. (Aborts command)
        5) Done. (Exits and runs command)

Enter option:5
Successfully modified log group cli
```

cmgr を使ったローカル・ノードのメンバーシップ取消し

強制 CXFS シャットダウンの前などにローカル・ノードの CXFS カーネル・メンバーシップを取消すには、そのローカル・ノードで次を入力します。

```
admin cxfs_stop
```

このコマンドは、クラスタのその他の部分でノード異常と解釈されます。このため、クラスタのその他の部分では、CXFS カーネル・メンバーシップ定足数が失われたために異常が発生したり、異常ノードのリセットが決定される場合があります。リセットを回避するには、ノード定義を変更して、システム・コントローラ・ステータスを無効にできます。

cmgr を使ったローカル・ノードのメンバーシップ許可

ローカル・ノードの CXFS カーネル・メンバーシップを許可すると、そのノードに CXFS カーネル・メンバーシップの再適用を許可します。以下の状況では、ローカル・ノードの CXFS カーネル・メンバーシップをアクティブに許可する必要があります。

- 176 ページの「cmgr を使ったローカル・ノードのメンバーシップ取消し」で説明されているように、手動取消しの後
- コンソールまたは /var/adm/SYSLOG のエラー・メッセージによって指示された場合
- カーネルによって取消しが実行された後。この状況は、/var/adm/SYSLOG 内の次のメッセージによって示されます。

```
Membership lost - withdrawing from cluster
```

ローカル・ノードの CXFS カーネル・メンバーシップを許可するには、次のコマンドを使用します。

```
cmgr> admin cxfs_start
```

209 ページの「データベースと CXFS のシャットダウン」も参照してください。

cmgr を使った CXFS ファイルシステム・タスク

この節では、ファイルシステムを定義する方法、ファイルシステムをマウントできるノードとできないノードを指定する方法（*enabled* または *disabled* のノード）、およびマウントとアンマウントを実行する方法について説明します。

以下に該当する場合、特定のファイルシステムを特定のノードにマウントできます。

- ノードが以下のいずれかに該当する。
  - デフォルトのローカル・ステータスが有効になっており、ノードは明示的に無効なノードのファイルシステムのリストにない
  - デフォルトのローカル・ステータスが無効になっており、ノードは明示的に有効なノードのファイルシステムのリストにある

178 ページの「cmgr を使った CXFS ファイルシステムの定義」を参照してください。

- ファイルシステムのグローバル・ステータスが有効になっている。183 ページの「cmgr を使った CXFS ファイルシステムのマウント」を参照してください。

cmgr を使った CXFS ファイルシステムの定義

ファイルシステムと、ファイルシステムをマウントできるノードを定義するには、以下のコマンドを使用します。

```
define cxfs_filesystem logical_filesystem_name [in cluster clustername]
   set device_name to devicename
   set mount_point to mountpoint
   set mount_options to mount_options
   set force to true|false
   set dflt_local_status to enabled|disabled
   add cxfs_server admin_nodename
      set rank to 0|1|2|...
   add enabled_node nodename
   add disabled_node nodename
   remove cxfs_server admin_nodename
   remove enabled_node nodename
   remove disabled_node nodename
```

上記のコマンド書式について、次の点に注意してください。

- cxfs_filesystem には、任意の論理名を指定できます。論理名は、アンダースコア (_) で始めたり、空白として使用される文字を含めたり、255 文字より長くすることはできません。

  メモ: GUI 内では、デフォルトにデバイス名の最後の部分が使用されます。たとえば、デバイス名が /dev/cxvm/d76lun0s0 の場合、GUI によって、d76lun0s0 という論理ファイルシステム名が自動的に指定されます。GUI では、cmgr を使用して定義されたその他の論理名を受付けますが、論理名は変更できません。論理名を変更するには、cmgr を使用しなければなりません。

- device_name は、CXFS クラスタ内のすべてのノード間で共有される XVM ボリュームのデバイス名です。名前は /dev/cxvm/ で始まる必要があります。

- mount_point は、指定した XVM ボリュームが接続されるディレクトリです。このディレクトリ名は、スラッシュ (/) で始まる必要があります。詳細については、mount(1M) のマン・ページを参照してください。

- mount_options は、mount(1M) コマンドに渡されるオプションで、指定した XVM ボリュームへのアクセスを制御するために使用されます。使用可能なオプションのリストについては、fstab(4) のマン・ページを参照してください。

- IRIX ノードに対して umount -k オプション(Solaris ノードに対しては fuser -k)を使用して強制的にアンマウントを実行する場合は、force を true に設定する必要があります。-k オプションは、該当するファイルシステムのファイルまたは現在のディレクトリを開いているプロセスを強制終了した後、アンマウントします。force オプションは、デフォルトではオフです。詳細については、umount(1M) のマン・ページを参照してください。

- dflt_local_status は、指定されていないすべてのノードにファイルシステムをマウントできるか、それとも指定されていないノードにはファイルシステムをマウントできないかどうかを定義します。その後、必要に応じて add enabled_node コマンドまたは add disabled_node コマンドを使用し、デフォルトとは異なるノードを明示的に指定します。結果が同じになる組合わせは、複数あります。

  たとえば、システムにノードが 10 個あるとします(node1～node10)。この場合、以下の方法を使用できます。

  – ファイルシステムがすべてのノードにマウントされるようにし、さらに、後でクラスタに追加されたノードにもマウントされるようにする場合は、次のように指定します。

  ```
  set dflt_local_status to enabled
  ```

  – ファイルシステムが node5 以外のすべてのノードにマウントされるようにし、さらに、後でクラスタに追加されたノードにもマウントされるようにする場合は、以下のように指定します。

  ```
  set dflt_local_status to enabled
  add disabled_node cxfs5
  ```

  – ファイルシステムが node5 以外のすべてのノードにマウントされるようにし、後でクラスタに追加されたノードにはマウントされないようにする場合は、以下のように指定します。

  ```
  set dflt_local_status to disabled
  add enabled_node cxfs1
  add enabled_node cxfs2
  add enabled_node cxfs3
  add enabled_node cxfs4
  add enabled_node cxfs6
  add enabled_node cxfs7
  add enabled_node cxfs8
  add enabled_node cxfs9
  add enabled_node cxfs10
  ```

  – ファイルシステムが node5～node10 と、今後追加されるすべてのノードにマウントされるようにする場合は、以下のように指定します。

  ```
  set dflt_local_status to enabled
  add disabled_node cxfs1
  add disabled_node cxfs2
  ```

```
add disabled_node cxfs3
add disabled_node cxfs4
```

有効なノードにファイルシステムを実際にマウントするには、183 ページの「cmgr を使った CXFS ファイルシステムのマウント」を参照してください。

- cxfs_server は、メタデータ・サーバとして使用可能なノードのリストに対して、指定した CXFS 管理ノード名の追加または削除を行います。

以下に、fs1 ファイルシステムのメタデータ・サーバとして使用可能な 2 つのノードの例を示します。クラスタの開始時に cxfs6(ランク 0 の優先サーバ)が動作していなかったり、後に異常が発生したり、クラスタから削除された場合は、cxfs7(ランク 1)が使用されます。ファイルシステムは、すべてのノードにマウントされています。

メモ:特定のファイルシステムのメタデータ・サーバのリストは順序付きですが、ネットワーク・レーテンシやその他の予測不能な遅れがあるため、起動サイクル中にどのノードが実際にサーバになるかは予想できません。

通常モードの場合:

```
cmgr> define cxfs_filesystem fs1 in cluster cxfs6-8

cxfs_filesystem fs1 ? set device_name to /dev/cxvm/d76lun0s0
cxfs_filesystem fs1 ? set mount_point to /mnts/fs1
cxfs_filesystem fs1 ? set force to false
cxfs_filesystem fs1 ? add cxfs_server cxfs6
Enter CXFS server parameters, when finished enter "done" or "cancel"

CXFS server - cxfs6 ? set rank to 0
CXFS server - cxfs6 ? done
cxfs_filesystem fs1 ? add cxfs_server cxfs7
Enter CXFS server parameters, when finished enter "done" or "cancel"

CXFS server - cxfs7 ? set rank to 1
CXFS server - cxfs7 ? done
cxfs_filesystem fs1 ? set dflt_local_status to enabled
cxfs_filesystem fs1 ? done
Successfully defined cxfs_filesystem fs1

cmgr> define cxfs_filesystem fs2 in cluster cxfs6-8

cxfs_filesystem fs2 ? set device_name to /dev/cxvm/d76lun0s1
```

```
cxfs_filesystem fs2 ? set mount_point to /mnts/fs2
cxfs_filesystem fs2 ? set force to false
cxfs_filesystem fs2 ? add cxfs_server cxfs8
Enter CXFS server parameters, when finished enter "done" or "cancel"

CXFS server - cxfs8 ? set rank to 0
CXFS server - cxfs8 ? done
cxfs_filesystem fs2 ? set dflt_local_status to enabled
cxfs_filesystem fs2 ? done
Successfully defined cxfs_filesystem fs2
```

プロンプト・モードの場合:

```
cmgr> define cxfs_filesystem fs1 in cluster cxfs6-8

(Enter "cancel" at any time to abort)

Device ? /dev/cxvm/d76lun0s0
Mount Point ? /mnts/fs1
Mount Options[optional] ?
Use Forced Unmount ? <true|false> ? false
Default Local Status <enabled|disabled> ? (enabled)

DEFINE CXFS FILESYSTEM OPTIONS

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1

No current servers
```

```
Server Node ? cxfs6
Server Rank ? 0

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1
Server Node ? cxfs7
Server Rank ? 1


        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:9
Successfully defined cxfs_filesystem fs1

cmgr> define cxfs_filesystem fs2 in cluster cxfs6-8

(Enter "cancel" at any time to abort)

Device ? /dev/cxvm/d77lun0s1
Mount Point ? /mnts/fs2
Mount Options[optional] ?
Use Forced Unmount ? <true|false> ? false
Default Local Status <enabled|disabled> ? (enabled)
```

```
DEFINE CXFS FILESYSTEM OPTIONS

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:1

Server Node ? cxfs8
Server Rank ? 0

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:9
Successfully defined cxfs_filesystem fs2
```

cmgr を使った CXFS ファイルシステムのマウント

有効なノードにファイルシステムをマウントするには、次を入力します。

```
admin cxfs_mount cxfs_filesystem logical_filesystem_name [on node
nodename] [in cluster clustername]
```

このコマンドは、ファイルシステムのグローバル・ステータスを有効にします。つまり、*nodename* を指定した場合は、ローカル・ステータスが有効になります。ファイルシステムを特定のノードにマウントするには、

グローバル・ステータスとローカル・ステータスの両方が有効になっている必要があります。177 ページの「cmgr を使った CXFS ファイルシステム・タスク」を参照してください。

ノードは、define cxfs_filesystem コマンドと modify cxfs_filesystem コマンドを使用して、まず有効にする必要があります。178 ページの「cmgr を使った CXFS ファイルシステムの定義」および 185 ページの「cmgr を使った CXFS ファイルシステムの変更」を参照してください。

たとえば、グローバル・ステータスを enabled に設定して、f1 ファイルシステムをアクティブにするには、次を入力します。

```
cmgr> admin cxfs_mount cxfs_filesystem fs1 in cluster cxfs6-8
```

すると、ファイルシステムに対してローカル・ステータスが enabled になっているすべてのノードに、このファイルシステムがマウントされます。

ローカル・ステータスを enabled に変更するには、次を入力します。

```
cmgr> admin cxfs_mount cxfs_filesystem fs1 on node cxfs7 in cluster cxfs6-8
```

ファイルシステムのグローバル・ステータスが disabled の場合は、何も変わりません。ファイルシステムのグローバル・ステータスが enabled の場合は、ローカル・ステータスを変更した結果として、このファイルシステムがノードにマウントされます。

メモ: CXFS サービスがアクティブではない場合、ファイルシステムのマウントは完全には成功しません。ファイルシステムはマウント準備済みとしてマークされますが、CXFS サービスを開始するまでは実際にマウントされません。詳細については、172 ページの「cmgr を使った CXFS サービスの開始」を参照してください。

---

cmgr を使った CXFS ファイルシステムのアンマウント

ファイルシステムをアンマウントするには、次を入力します。

```
admin cxfs_unmount cxfs_filesystem filesystemname [on node nodename] [in cluster clustername]
```

modify cxfs_filesystem コマンドとは異なり、このコマンドはアクティブなファイルシステムで実行できます。

たとえば、グローバル・ステータスを disabled に設定して、f1 ファイルシステムを停止するには、次を入力します。

```
cmgr> admin cxfs_unmount cxfs_filesystem fs1 in cluster cxfs6-8
```

すると、ファイルシステムに対してローカル・ステータスが enabled になっているすべてのノードから、このファイルシステムがアンマウントされます。

ローカル・ステータスを disabled に変更するには、次を入力します。

```
cmgr> admin cxfs_unmount cxfs_filesystem fs1 on node cxfs7 in cluster cxfs6-8
```

ファイルシステムのグローバル・ステータスが disabled の場合は、何も変わりません。ファイルシステムのグローバル・ステータスが enabled の場合は、ローカル・ステータスを変更した結果として、このファイルシステムがノードからアンマウントされます。

cmgr を使った CXFS ファイルシステムの変更

メモ: マウント済みのファイルシステムを変更することはできません。

ファイルシステムを変更するには、以下のコマンドを使用します。

```
modify cxfs_filesystem logical_filesystem_name [in cluster clustername]
   set device_name to devicename
   set mount_point to mountpoint
   set mount_options to options
   set force to true|false
   set dflt_local_status to enabled|disabled
   add cxfs_server servername
     set rank to 0|1|2|...
   modify cxfs_server servername
     set rank to 0|1|2|...
   add enabled_node nodename
   add disabled_node nodename
   remove cxfs_server nodename
   remove enabled_node nodename
   remove disabled_node nodename
```

これらは、ファイルシステムの定義に使用されるコマンドと同じです。詳細については、178 ページの「cmgr を使った CXFS ファイルシステムの定義」を参照してください。

通常モードの場合:

```
cmgr> show cxfs_filesystem fs1 in cluster cxfs6-8

Name: fs1
Device: /dev/cxvm/d76lun0s0
Mount Point: /mnts/fs1
Forced Unmount: false
Global Status: disabled
```

```
Default Local Status: enabled

Server Name: cxfs6
       Rank: 0
Server Name: cxfs7
       Rank: 1
Disabled Client: cxfs8

cmgr> modify cxfs_filesystem fs1 in cluster cxfs6-8
Enter commands, when finished enter either "done" or "cancel"

cxfs_filesystem fs3 ? modify cxfs_server cxfs6
Enter CXFS server parameters, when finished enter "done" or "cancel"

Current CXFS server cxfs6 parameters:
        rank : 0
CXFS server - cxfs6 ? set rank to 2
CXFS server - cxfs6 ? done
cxfs_filesystem fs1 ? done

Successfully modified cxfs_filesystem fs1
cmgr> show cxfs_filesystem fs1 in cluster cxfs6-8

Name: fs1
Device: /dev/cxvm/d76lun0s0
Mount Point: /mnts/fs1
Forced Unmount: false
Global Status: disabled
Default Local Status: enabled

Server Name: cxfs6
       Rank: 2
Server Name: cxfs7
       Rank: 1
Disabled Client: cxfs8
```

プロンプト・モードの場合:

```
cmgr> show cxfs_filesystem fs1 in cluster cxfs6-8

Name: fs1
Device: /dev/cxvm/d76lun0s0
```

```
Mount Point: /mnts/fs1
Forced Unmount: false
Global Status: disabled
Default Local Status: enabled

Server Name: cxfs6
       Rank: 0
Server Name: cxfs7
       Rank: 1
Disabled Client: cxfs8

cmgr> modify cxfs_filesystem fs1 in cluster cxfs6-8

(Enter "cancel" at any time to abort)

Device ? (/dev/cxvm/d76lun0s0)
Mount Point ? (/mnts/fs1)
Mount Options[optional] ?
Use Forced Unmount ? <true|false>  ? (false)
Default Local Status <enabled|disabled> ? (enabled)

MODIFY CXFS FILESYSTEM OPTIONS

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:0

Current servers:
CXFS Server 1 - Rank: 0         Node: cxfs6
CXFS Server 2 - Rank: 1         Node: cxfs7

Server Node ? cxfs6
Server Rank ? (0) 2
```

```
        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:7

Current settings for filesystem (fs1)

CXFS servers:
        Rank 2          Node cxfs6
        Rank 1          Node cxfs7

Default local status: enabled

No explicitly enabled clients

Explicitly disabled clients:
        Disabled Node: cxfs8

        0) Modify Server.
        1) Add Server.
        2) Remove Server.
        3) Add Enabled Node.
        4) Remove Enabled Node.
        5) Add Disabled Node.
        6) Remove Disabled Node.
        7) Show Current Information.
        8) Cancel. (Aborts command)
        9) Done. (Exits and runs command)

Enter option:9
Successfully modified cxfs_filesystem fs3
```

cmgr を使ったファイルシステムのメタデータ・サーバの再設定

メモ: 再設定機能は完全に実装されていますが、関連する問題が多数存在するため、CXFS ではサポートされていません。これらの機能を使用してもデータの整合性が損なわれることはありませんが、クラスタ・ノードのパニックやハングが発生する可能性があります。この機能は、このような問題が解決された時点で完全にサポートされる予定です。再設定の機能は、このリリースではデフォルトで無効になっています。

cxfs_relocation_ok systune を使用してカーネルで再設定が明示的に有効になっている場合、メタデータ・サーバは、次のコマンドを使用して別のノードに再設定することができます(cmgr を実行しているシステムにファイルシステムがマウントされなければならない場合):

```
admin cxfs_relocate cxfs_filesystem filesystem_name to node nodename [in cluster clustername]
```

メモ: この機能は、ライブ・システムでのみ使用できます。

クラスタ cxfs6-8 内の fs1 のメタデータ・サーバを cxfs6 から cxfs7 に再設定するには、次を入力します。

```
cmgr> admin cxfs_relocate cxfs_filesystem fs1 to node cxfs7 in cluster cxfs6-8
```

CXFS カーネル・メンバーシップには、再設定による影響はありません。ただし、メタデータ・サーバの再設定中は、ファイルシステムのパフォーマンスが低下することがあります。

詳細については、185 ページの「cmgr を使った CXFS ファイルシステムの変更」を参照してください。

cmgr を使った CXFS ファイルシステムの削除

ファイルシステムを削除するには、次のコマンドを使用します。

```
delete cxfs_filesystem filesystemname [in cluster clustername]
```

例は、次のとおりです。

```
cmgr> delete cxfs_filesystem fs2 in cluster cxfs6-8
```

cmgr を使ったスイッチと I/O 制限タスク

以下のタスクを使用して、スイッチおよび I/O 制限を設定できます。一般情報については、21 ページの「I/O 制限」を参照してください。

メモ: Solaris および Windows ノードでは、データの整合性を保護するために I/O 制限が必要です。I/O 制限をサポートするには、Brocade スイッチが必須であるため、マルチ OS CXFS クラスタには Brocade スイッチが必要です。

cmgr を使ったスイッチの定義

新しいスイッチを定義してクラスタで I/O 制限をサポートするには、次のコマンドを使用します。

```
define switch switch_hostname username username password password [mask mask]
```

上記のコマンド書式について、次の点に注意してください。

- switch は、ファイバ・チャネル・スイッチのホスト名です。これは、スイッチの IP アドレスを判断するために使用されます。
- username は、スイッチに telnet メッセージを送信する場合に使用するユーザ名です。
- password は、指定した *username* のパスワードです。
- mask は、(64 ビット・ポート・ビットマップを表す)16 進の文字列で、スイッチ内の制限されないポートのリストを示します。ポートには、ゼロから番号が付けられます。ビット i がゼロ以外の場合、i に対応するポートは、常に制限操作から除外されます。たとえば、A4 というマスクは、3 番目、6 番目、および 8 番目のポート(ポート番号 2、5、および 7)が制限を受けないことを示します。

  IRIX ノードでは、使用可能な HBA を自動的に検出し、「制限(Fence)」または「フェンスリセット(FenceReset)」の異常時のアクションが選択されている場合は、すべてのファイバ・チャネル HBA の制限をデフォルトでオフにします。ただし、マスクされた HBA は制限されません。マスキングを使用すると、SAN に接続されているがクラスタと共有されていないデバイスの制限を防止し、CXFS のステータスに関係なく、常にこれらのデバイスを利用できます。このため、テープ・ストレージへのアクセスに使用される HBA、またはローカル(クラスタ化されていない)・デバイスにアクセスするためだけに使用される HBA をマスクすることができます。

例は、次のとおりです。

```
cmgr> define switch ptg-brocade username admin password password mask A4
```

cmgr を使ったスイッチ定義の変更

特定のファイバ・チャネル・スイッチのユーザ名、パスワード、またはマスクを変更するには、次のコマンドを使用します。

```
modify switch switch_hostname username username password password [mask mask]
```

引数は、190 ページの「cmgr を使ったスイッチの定義」の場合と同じです。

たとえば、スイッチ ptg-brocade のマスクを A4 から 0 に変更するには(つまり、スイッチのどのポートも制限から除外されません)、次を入力します。

```
cmgr> modify switch ptg-brocade username admin password password mask 0
```

cmgr を使ったノードの I/O 制限の強化

I/O 制限を強化すると、そのノードは SAN から分離されます。そして、CXFS から telnet プロトコルを通じて Brocade スイッチにメッセージが送信され、ポートが無効に設定されます。ノードが分離された後は、そのノードによって共有 CXFS ファイルシステムのデータが破壊されることはありません。次のコマンドを使用します。

```
admin fence raise [node nodename]
```

nodename は、分離するノードの名前です。

たとえば、デフォルトのノードを分離するには、次を入力します。

```
cmgr> admin fence raise
```

ノード Node3 を分離するには、次を入力します。

```
cmgr> admin fence raise node Node3
```

cmgr を使ったノードの I/O 制限の緩和

特定のノードの I/O 制限を緩和してポートを再度有効にし、ノードが SAN に接続して共有 CXFS ファイルシステムにアクセスできるようにするには、次のコマンドを使用します。

```
admin fence lower [node nodename]
```

nodename は、再接続するノードの名前です。

たとえば、デフォルトのノードにアクセスを提供するには、次を入力します。

```
cmgr> admin fence lower
```

ノード Node3 にアクセスを提供するには、次を入力します。

```
cmgr> admin fence lower node Node3
```

cmgr を使ったスイッチ・ポート情報の更新

クラスタ・データベース内の、ホスト・バス・アダプタ (HBA: Host Bus Adapter) とスイッチ・ポートのマッピングを更新するには、次のコマンドを使用します。

```
admin fence update
```

スイッチを再設定したり、ポートを追加する場合は、このコマンドを実行する必要があります。

cmgr を使ったスイッチ定義の削除

スイッチを削除するには、次のコマンドを使用します。

```
delete switch switch_hostname
```

switch_hostname は、ファイバ・チャネル・スイッチのホスト名です。これは、スイッチの IP アドレスを判断するために使用されます。

例は、次のとおりです。

```
cmgr> delete switch ptg-brocade
Successfully updated switch config.
```

cmgr を使ったスイッチの表示

システム内のスイッチを表示するには、次のコマンドを使用します。

```
show switches
```

特定のノードのスイッチを表示するには、次のコマンドを使用します。

```
show switch hostname
```

例は、次のとおりです。

```
cmgr> show switch ptg-brocade
  Switch[0]
      *Hostname ptg-brocade Username admin Password password Mask 0
      Vendor BROCADE Number of ports 8
            0 0000000000000000 Reset
            1 210000e08b0102c6 Reset
            2 210000e08b01fec5 Reset
            3 210000e08b019dc5 Reset
            4 210000e08b0113ce Reset
```

```
5 210000e08b027795 Reset thump
6 210000e08b019ef0 Reset
7 210000e08b022242 Reset
```

cmgr を使ったスイッチのステータスの照会

スイッチの各ポートのステータスを照会するには、次のコマンドを使用します。

```
admin fence query
```

例は、次のとおりです。

```
cmgr> admin fence query
  Switch[0] "brocade04" has 16 ports
    Port 4 type=FABRIC status=enabled  hba=210000e08b0042d8 on host o200c
    Port 5 type=FABRIC status=enabled  hba=210000e08b00908e on host cxfs30
    Port 9 type=FABRIC status=enabled  hba=2000000173002d3e on host cxfssun3
```

詳細を表示するには(デフォルトのクラスタ内のノードに接続されているポートだけでなく、スイッチのすべてのポートが表示されます)、次のコマンドを使用します。

```
admin fence query verbose
```

例は、次のとおりです。

```
cmgr> admin fence query verbose
  Switch[0] "brocade04" has 16 ports
    Port 0 type=FABRIC status=enabled  hba=2000000173003b5f on host UNKNOWN
    Port 1 type=FABRIC status=enabled  hba=2000000173003adf on host UNKNOWN
    Port 2 type=FABRIC status=enabled  hba=210000e08b023649 on host UNKNOWN
    Port 3 type=FABRIC status=enabled  hba=210000e08b021249 on host UNKNOWN
    Port 4 type=FABRIC status=enabled  hba=210000e08b0042d8 on host o200c
    Port 5 type=FABRIC status=enabled  hba=210000e08b00908e on host cxfs30
    Port 6 type=FABRIC status=enabled  hba=2000000173002d2a on host UNKNOWN
    Port 7 type=FABRIC status=enabled  hba=2000000173003376 on host UNKNOWN
    Port 8 type=FABRIC status=enabled  hba=2000000173002c0b on host UNKNOWN
    Port 9 type=FABRIC status=enabled  hba=2000000173002d3e on host cxfssun3
    Port 10 type=FABRIC status=enabled  hba=2000000173003430 on host UNKNOWN
    Port 11 type=FABRIC status=enabled  hba=200900a0b80c13c9 on host UNKNOWN
    Port 12 type=FABRIC status=disabled hba=0000000000000000 on host UNKNOWN
    Port 13 type=FABRIC status=enabled  hba=200d00a0b80c2476 on host UNKNOWN
    Port 14 type=FABRIC status=enabled  hba=1000006069201e5b on host UNKNOWN
    Port 15 type=FABRIC status=enabled  hba=1000006069201e5b on host UNKNOWN
```

スクリプトの例

次のスクリプトによって、タイプ CXFS の 3 ノード・クラスタが定義されます。ノードのタイプは、CXFS です。

メモ: この例では、CXFS で必要な 1 つのネットワーク・インタフェースのみを定義します。ここでは、簡潔にするためにホスト名を使用していますが、51 ページの「ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf」の規則に従わなければなりません。また、混乱を避けるために IP アドレスを代わりに使用することもできます。

この例には、システム・コントローラ定義は示されていません。

```
#!/usr/cluster/bin/cmgr -if
#
#Script to define a three-node cluster


define node cxfs6
        set hostname to cxfs6
        set is_cxfs to true
        set operating_system to irix
        set node_function to server_admin
        add nic cxfs6
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
done

define node cxfs7
        set hostname to cxfs7
        set is_cxfs to true
        set operating_system to irix
        set node_function to server_admin
        add nic cxfs7
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
done

define node cxfs8
```

```
        set hostname to cxfs8
        set is_cxfs to true
        set operating_system to irix
        set node_function to server_admin
        add nic cxfs8
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
done

define cluster cxfs6-8
        set is_cxfs to true
        set is_failsafe to true
        set clusterid to 20
        add node cxfs6
        add node cxfs7
        add node cxfs8
        done
quit
```

このスクリプトを実行すると、次のような出力が表示されます。

```
Successfully defined node cxfs6

Successfully defined node cxfs7

Successfully defined node cxfs8

Successfully defined cluster cxfs6-8
```

次のスクリプトによって、2 つのファイルシステムが定義されます。ファイルシステム `fs1` は、`cxfs8` 以外のすべてのノードにマウントされ、もう 1 つのファイルシステム `fs2` は、すべてのノードにマウントされます。

```
#!/usr/cluster/bin/cmgr -if
# Script to define two filesystems
# Define fs1, do not mount on cxfs8
define cxfs_filesystem fs1 in cluster cxfs6-8
set device_name to /dev/cxvm/d76lun0s0
set mount_point to /mnts/fs1
set force to false
add cxfs_server cxfs6
  set rank to 0
```

```
  done
add cxfs_server cxfs7
  set rank to 1
  done
set dflt_local_status to enabled
add disabled_node cxfs8
done
#
# Define fs2, mount everywhere
define cxfs_filesystem fs2 in cluster cxfs6-8
set device_name to /dev/cxvm/d76lun0s1
set mount_point to /mnts/fs2
set force to false
add cxfs_server cxfs8
set rank to 0
done
set dflt_local_status to enabled
done
```

cmgr スクリプトの自動作成

クラスタ・データベースを設定したら、build_cmgr_script(1M) コマンドを使用して、cmgr(1M) スクリプトをそのクラスタ・データベースの内容に基づいて自動的に作成できます。生成されたスクリプトには、以下が含まれます。

- ノード定義
- クラスタ定義
- スイッチ定義
- CXFS ファイルシステム定義
- パラメータ設定
- cmgr(1M) コマンドまたは GUI のいずれかを使用して行われた変更
- FailSafe 情報（同時実行クラスタ内においてのみ）

その後、生成されたスクリプトを必要に応じて使用し、cdbreinit を実行した後にクラスタを再作成できます。

デフォルトでは、生成されたスクリプトに次の名前が付けられます。

/tmp/cmgr_create_cluster_*clustername_processID*

-o オプションを使用して、別のパス名を指定することもできます。

build_cmgr_script [-o *script_pathname*]

詳細については、build_cmgr_script(1M) のマン・ページを参照してください。

例は、次のとおりです。

```
# /var/cluster/cmgr-scripts/build_cmgr_script -o /tmp/newcdb
Building cmgr script for cluster clusterA ...
build_cmgr_script: Generated cmgr script is /tmp/newcdb
```

スクリプト例ファイルの内容は、以下のとおりです。

```
#!/usr/cluster/bin/cmgr -f

# Node nodeE definition
define node nodeE
        set hostname to nodeE.americas.sgi.com
        set operating_system to IRIX
        set is_failsafe to false
        set is_cxfs to true
        set node_function to server_admin
        set nodeid to 5208
        set reset_type to powerCycle
        add nic nodeE
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
done

# Node nodeD definition
define node nodeD
        set hostname to nodeD.americas.sgi.com
        set operating_system to IRIX
        set is_failsafe to false
        set is_cxfs to true
        set node_function to server_admin
        set nodeid to 5181
```

```
        set reset_type to powerCycle
        add nic nodeD
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
done

# Node nodeF definition
define node nodeF
        set hostname to nodeF.americas.sgi.com
        set operating_system to IRIX
        set is_failsafe to false
        set is_cxfs to true
        set node_function to server_admin
        set nodeid to 5401
        set reset_type to powerCycle
        add nic nodeF
                set heartbeat to true
                set ctrl_msgs to true
                set priority to 1
        done
done

# Define cluster and add nodes to the cluster
define cluster clusterA
        set is_failsafe to false
        set is_cxfs to true
        set cx_mode to normal
        set clusterid to 35
done

modify cluster clusterA
        add node nodeD
        add node nodeF
        add node nodeE
done

set cluster clusterA

define cxfs_filesystem fs1
        set device_name to /dev/cxvm/fs1
```

```
        set mount_point to /fs1
        set force to false
        set dflt_local_status to enabled
        add cxfs_server nodeE
                set rank to 1
        done
        add cxfs_server nodeD
                set rank to 2
        done
        add cxfs_server nodeF
                set rank to 0
        done
done

define cxfs_filesystem fs2
        set device_name to /dev/cxvm/fs2
        set mount_point to /fs2
        set force to false
        set dflt_local_status to enabled
        add cxfs_server nodeE
                set rank to 1
        done
        add cxfs_server nodeD
                set rank to 2
        done
        add cxfs_server nodeF
                set rank to 0
        done
done

define cxfs_filesystem fs2
        set device_name to /dev/cxvm/fs2
        set mount_point to /fs2
        set force to false
        set dflt_local_status to enabled
        add cxfs_server nodeE
                set rank to 1
        done
        add cxfs_server nodeD
                set rank to 2
        done
        add cxfs_server nodeF
```

```
                set rank to 0
        done
done

# Setting CXFS parameters
modify cx_parameters
        set tie_breaker to none
done

quit
```

管理とメインテナンス

オフライン管理タスクは、プール内の CXFS 管理ノード（cxfs_cluster 製品がインストールされているノード）にログインしている場合、または GUI がプール内の CXFS 管理ノードに接続されている場合に、cmgr(1M) コマンドを使用して実行できます。ただし、ファイルシステムがマウントされている場合は、メタデータ・サーバから管理を実行しなければなりません（cmgr(1M) を使用したり、クライアントのみのノードに GUI を接続することはできません）。

CXFS と XFS において、以下は同じです。

- ディスクの概念
- ファイルシステムの概念
- ユーザ・インタフェース
- ファイルシステムの作成

これらのトピックについての詳細は、『IRIX Admin: Disks and Filesystems』を参照してください。

この章の残りの部分では、以下のトピックについて説明します。

- 202 ページの「既存のノードのクライアントのみのノードへの切替え」
- 203 ページの「NFS エクスポート・スクリプト」
- 205 ページの「telnet と I/O 制限の使用」
- 205 ページの「fsr(1M) の使用」
- 205 ページの「CXFS クラスタでの cron(1) の使用」
- 205 ページの「階層ストレージ管理 (HSM: Hierarchical Storage Management) システムの使用」
- 206 ページの「ファイルシステムのアクティブなメタデータ・サーバの検出」
- 209 ページの「メタデータ・サーバの回復」
- 209 ページの「データベースと CXFS のシャットダウン」
- 215 ページの「再起動時の CXFS の再開の回避」
- 215 ページの「ログ・ファイル管理」
- 216 ページの「ボリューム管理」

- 217 ページの「ディスク管理」
- 218 ページの「ファイルシステムのメインテナンス」
- 220 ページの「ダンプと復元」
- 221 ページの「クラスタ・データベースのバックアップと復元」
- 222 ページの「chkconfig フラグ」
- 222 ページの「システム調整可能パラメータ」

6.5.12f 以前のバージョンから直接アップグレードした場合は、ファイルシステム定義を新しい形式に手動で切替える必要があります。66 ページの「アップグレードのためのファイルシステム定義の切替え」を参照してください。

既存のノードのクライアントのみのノードへの切替え

6.5.18f にアップグレードする場合に、重み 1 の既存のノード（6.5.18f では、サーバとして使用可能なノードとして定義されています）をクライアントのみのノードに変更するときは、以下を実行する必要があります。

1. ノードが任意のファイルシステムに対してメタデータ・サーバとして使用可能なノードとしてリストされていないことを確認します。136 ページの「GUI を使った CXFS ファイルシステムの変更」または 185 ページの「cmgr を使った CXFS ファイルシステムの変更」を参照してください。
2. ノードで CXFS サービスを停止します。125 ページの「GUI を使った CXFS サービスの停止（通常 CXFS シャットダウン）」または 172 ページの「cmgr を使った CXFS サービスの停止」を参照してください。
3. ノードが含まれないようにクラスタを変更します。123 ページの「GUI を使ったクラスタ定義の変更」または 168 ページの「cmgr を使ったクラスタの変更」を参照してください。
4. ノード定義を削除します。121 ページの「GUI を使ったノードの削除」または 162 ページの「cmgr を使ったノードの削除」を参照してください。
5. ノードに cxfs_client パッケージをインストールします。49 ページの「クライアントのみのノードへの IRIX ソフトウェアのインストール」を参照してください。
6. ノードを再定義して、クライアントのみのノードの機能を使用します。109 ページの「GUI を使ったノードの定義」または 148 ページの「cmgr を使ったノードの定義」を参照してください。
7. ノードが含まれるようにクラスタを変更します。123 ページの「GUI を使ったクラスタ定義の変更」または 168 ページの「cmgr を使ったクラスタの変更」を参照してください。

8. ノードで CXFS サービスを開始します。125 ページの「GUI を使った CXFS サービスの開始」または 172 ページの「cmgr を使った CXFS サービスの開始」を参照してください。

NFS エクスポート・スクリプト

CXFS をインストールすると、以下のデフォルトのスクリプトが /var/cluster/clconfd-scripts ディレクトリに配置されます。

- cxfs-pre-mount
- cxfs-post-mount
- cxfs-pre-umount
- cxfs-post-umount

これらのスクリプトを使用することにより、/etc/exports にリストされている CXFS ファイルシステムが正常にマウントされている場合、NFS を使用して、これらのファイルシステムをエクスポートできます。これらのスクリプトは、/etc/exports ファイルで指定されている CXFS ファイルシステムのマウントまたはアンマウントの前後に、clconfd デーモンによって実行されます。ファイルの名前は、上記とまったく同じように指定しなければならず、root 実行アクセス権が必要です。ニーズに合わせて、これらのスクリプトを変更できます。

メモ: /etc/exports ファイルは、NFS クライアントにエクスポートされているファイルシステムを記述します。CXFS マウント・ポイントが exports ファイルに含まれている場合、cxfs-post-mount スクリプトを使用して CXFS マウント後にファイルシステムが再エクスポートされないかぎり、空のマウント・ポイントがエクスポートされます。

/etc/exports ファイルに、IRIS FailSafe によって管理されているファイルシステムを含めることはできません。

ファイルには以下の引数が渡されます。

- cxfs-pre-mount: ファイルシステム・デバイス名
- cxfs-post-mount: ファイルシステム・デバイス名および終了コード
- cxfs-pre-umount: ファイルシステム・デバイス名
- cxfs-post-umount: ファイルシステム・デバイス名および終了コード

ファイルシステム名はスクリプトに渡されるため、異なるファイルシステムに対して異なるアクションが取られるようスクリプトを作成できます。また、終了コードは post- ファイルに渡されるので、操作の結果(成功または失敗)に基づいて異なるアクションを取るようスクリプトを作成することが可能です。

clconfd デーモンは、これらのスクリプトの終了コードをチェックします。異常が発生した場合は(ゼロ以外)、以下が行われます。

- cxfs-pre-mount および cxfs-pre-umount の場合、対応するマウントまたはアンマウント操作は実行されません。
- cxfs-post-mount および cxfs-post-umount の場合、clconfd によって操作全体(-pre- スクリプトを含みます)が再試行されます。

これは、ファイルシステムをホストにマウントしたくない場合、そのファイルシステムに対して cxfs-pre-mount スクリプトが異常を返し、cxfs-post-mount スクリプトは成功を返す必要があることを意味しています。

lofs ファイルシステムのアンマウント

CXFS ファイルシステムをアンマウントする前に、CXFS ファイルシステムの lofs マウントをアンマウントする必要があります。lofs ファイルシステムをアンエクスポートしてローカルでアンマウントするには、次のようなスクリプトを使用できます。

```
#!/bin/ksh
#/var/cluster/clconfd-scripts/cxfs-pre-umount
echo "$0: Preparing to unmount CXFS file system \"$1\""
MNTPNT=`mount | grep "$1 " | cut -f 3 -d" "`
print "MNTPNT $MNTPNT"
if [ -n "${MNTPNT}" ] ; then
    lofslist=`mount | grep 'type lofs' | grep "${MNTPNT}" | nawk '{print $3}'`
    set -e
    for lofs in ${lofslist}
    do
        echo "$0: unmounting $lofs"
```

```
        umount -k $lofs
    done
    if /usr/etc/exportfs | /sbin/grep -q "${MNTPNT}" ; then
        echo "$0: unexporting $MNTPNT"
        /usr/etc/exportfs -u ${MNTPNT}
    fi
fi
```

telnet と I/O 制限の使用

ノードに問題がある場合は、telnet プロトコルを使用して I/O 制限ソフトウェアから適切なファイバ・チャネル・スイッチにメッセージが送信されます。スイッチで使用できるのは一度に 1 つの telnet セッションのみです。したがって、I/O 制限を使用する場合は、ファイバ・チャネル・スイッチの telnet ポートを常に空けておく必要があります。スイッチに対して telnet を実行後、セッションを接続したままにしないでください。

fsr(1M) の使用

fsr(1M) コマンドは、ファイルシステムのメタデータ・サーバでのみ使用できます。bulkstat システム・コールは、CXFS クライアントに対しては無効になっています。fsr は、ファイルシステムのアクティブなメタデータ・サーバだけに対して手動で使用してください。

CXFS クラスタでの cron(1) の使用

クラスタ内の複数のノードでファイルシステムへの負荷が高い同一のタスクを同時に実行すると、cron(1) デーモンによって CXFS ファイルシステムに多大な負荷がかけられる可能性があります。たとえば、find(1) コマンドを使用して、ファイルシステム内のファイルを検索するタスクなどです。

cron を使用して CXFS ファイルシステム上で開始されるタスクは、クラスタ内の単一ノード (アクティブなメタデータ・サーバを推奨) から起動することをお薦めします。

階層ストレージ管理 (HSM: Hierarchical Storage Management) システムの使用

CXFS では、XSDM (X/Open Data Storage Management Specification) とも呼ばれる DMAPI (Data Management Application Programming Interface) を通して、階層ストレージ管理 (HSM: Hierarchical Storage Management) 製品の使用をサポートします。HSM 製品の例は、DMF (Data Migration Facility) です。DMF は、現在 CXFS でサポートされている唯一の HSM 製品です。

HSM アプリケーションでは、すべての DMAPI インタフェース・コールをアクティブなメタデータ・サーバを通して行う必要があります。CXFS マウントされたファイルシステムとの DMAPI インタフェースは、CXFS クライアント・ノードでは提供されません。CXFS クライアントは、HSM アプリケーションとのすべての通信をメタデータ・サーバを通してルーティングします。このため、一般的には、HSM アプリケーションが CXFS メタデータ・サーバで実行されている必要があります。

HSM を CXFS で使用するには、以下を実行します。

- クラスタ内の各 CXFS 管理ノードに、`eoe.sw.dmi` をインストールします。クライアントのみのノードには、追加のソフトウェアは必要ありません。
- 管理するファイルシステムをマウントするときに、`dmi` オプションを使用します。この手順についての詳細は、135 ページの「GUI を使った CXFS ファイルシステムの定義」または 168 ページの「`cmgr` を使ったクラスタの変更」を参照してください。
- 管理する各ファイルシステムに対して、アクティブなメタデータ・サーバで HSM アプリケーションを開始します。

ファイルシステムのアクティブなメタデータ・サーバの検出

アクティブなメタデータ・サーバは、GUI、または `cluster_status`(1M) か `clconf_info` というコマンドを使用して検出できます。

GUI を使用したメタデータ・サーバの検出

以下を実行します。

1. 「表示(View)」->「ファイルシステム(Filesystems)」を選択します。
2. 表示領域で、表示するファイルシステムの名前をクリックします。アクティブなメタデータ・サーバの名前が、右側の詳細領域に表示されます。

図6-1 に、例を示します。

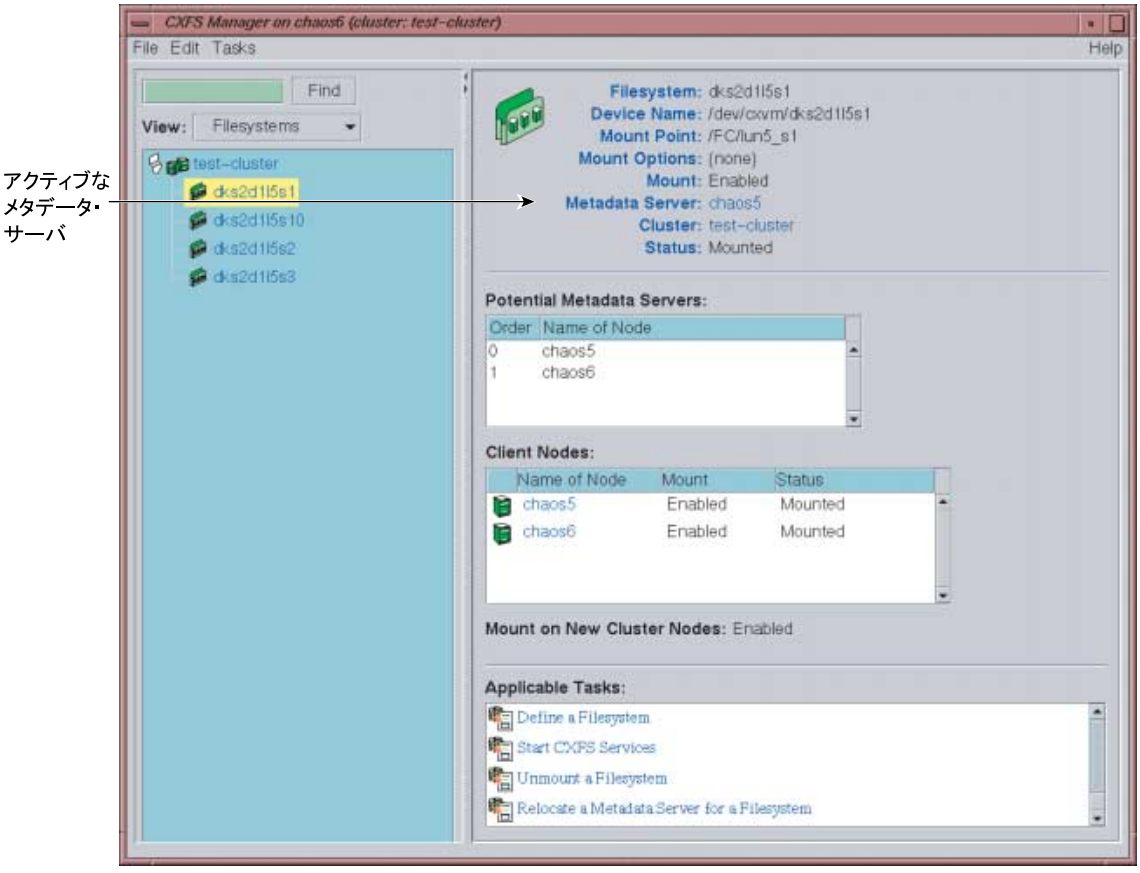


図6-1 メタデータ・サーバが表示されているウィンドウ

cluster_status を使用したメタデータ・サーバの検出

cluster_status コマンドを使用して、アクティブなメタデータ・サーバを検出できます。例は、次のとおりです。

```
# /var/cluster/cmgr-scripts/cluster_status

+ Cluster=cxfs6-8  FailSafe=Not Configured CXFS=ACTIVE                15:15:33
   Nodes =   cxfs6    cxfs7    cxfs8
FailSafe =
    CXFS =      UP       UP       UP

CXFS              DevName           MountPoint           MetaServer     Status
        /dev/cxvm/concat0             /concat0                cxfs7         UP
```

詳細については、239 ページの「cluster_status を使ったクラスタのステータスの確認」を参照してください。

clconf_info を使用したメタデータ・サーバの検出

clconf_info コマンドを使用して、特定のファイルシステムのアクティブなメタデータ・サーバを検出できます。たとえば、以下では、cxfs7 がメタデータ・サーバであることが示されています。

```
cxfs6 # clconf_info
Membership since Thu Mar  1 08:15:39 2001
Node         NodeId     Status    Age   Incarnation     CellId
cxfs6             6         UP      0             0          2
cxfs7             7         UP      0             0          1
cxfs8             8         UP      0             0          0
1 CXFS FileSystems
/dev/cxvm/concat0 on /concat0  enabled  server=(cxfs7)  2 client(s)=(cxfs8,cxfs6)
```

メタデータ・サーバの回復

メモ: このリリースでは、再設定の機能はデフォルトで無効になっており、回復の機能はスタンバイ・ノードを使用している場合のみサポートされています。

再設定および回復の機能は完全に実装されていますが、関連する問題が多数存在するため、現在のリリースでは完全にサポートされていません。これらの機能を使用してもデータの整合性が損なわれることはありませんが、クラスタ・ノードのパニックやハングが発生する可能性があります。再設定および回復の機能は、このような問題が解決された時点で、今後のリリースで完全にサポートされる予定です。

ファイルシステムのメタデータ・サーバとして動作しているノードがダウンした場合は、メタデータ・サーバとして使用可能なノードのリストに含まれる別のノードが新しいメタデータ・サーバとして選択されます。これは、ファイルシステムを定義する際に、メタデータ・サーバとして使用可能なノードが少なくとも 2 つはリストに含まれていることが前提となります。詳細については、135 ページの「GUI を使った CXFS ファイルシステムの定義」または 168 ページの「cmgr を使ったクラスタの変更」を参照してください。

選択されるメタデータ・サーバはファイルシステム・クライアントでなければなりません。その他のファイルシステム・クライアントでは、再設定プロセス中に遅れが発生します。各ファイルシステムの回復に要する時間は、アクティブな i ノードの数に依存します。全体の遅延時間は、各ファイルシステムの回復に要する時間の合計です。回復の際にファイルシステムがどの程度アクティブであるかに応じて、遅れの合計はファイルシステムあたり最高 7 分になる場合があります。

CXFS クライアントがダウンした場合、メタデータ・サーバは、そのクライアントをクリーンアップします。このプロセス中は、その他の CXFS クライアントで遅れが発生する場合があります。ダウンしたクライアントがトークンを保持していた場合、遅れはそのトークンによって異なります。クライアントがトークンを持っていなければ遅れはありません。別のクライアントの処理を進める上で取消さなければならないトークンをクライアントが保持している場合は、回復によって異常ノード・トークンが返されるまで、その他のクライアントは中断されます（たとえば、クライアントが書込みトークンを持っていて、別のクライアントが読取りを行いたい場合など）。遅れの実際の長さは、以下によって異なります。

- CXFS カーネル・メンバーシップの状況
- サーバがダウンしているかどうか
- このファイルシステムの回復に関連する回復順序でのサーバの位置

ダウンした CXFS クライアントは、すべてのメタデータ・サーバによってそのクライアントがクリーンアップされるまで、CXFS カーネル・メンバーシップに再設定できません。

データベースと CXFS のシャットダウン

この節では、以下の実行方法について説明します。

- 210 ページの「クラスタ・データベースのシャットダウン」
- 211 ページの「通常 CXFS シャットダウン」
- 213 ページの「強制 CXFS シャットダウン: ローカル・ノードのメンバーシップ取消し」

問題がある場合は、249 ページの 第10章「トラブルシューティング」を参照してください。状態についての詳細は、237 ページの 第9章「ステータスのモニタ」を参照してください。

クラスタ・データベースのシャットダウン

クラスタ・データベースのシャットダウンを行うと、クラスタ・データベースを管理する以下のユーザ空間デーモンが終了されます。

- `cad`
- `clconfd`
- `cmond`
- `crsd`
- `fs2d`

ノードでデータベースをシャットダウンしても、共有ファイルシステムへのアクセスは引続き使用可能で、ノードはクラスタのメンバーのままですが、データベースの更新は取得できません。ノードを再起動すると、すべてのサービスが再開されます。

クラスタ・データベースのシャットダウンを実行するには、次を入力します。

```
# /etc/init.d/cluster stop
```

また、起動時にデーモンが再開されるのを無効にする場合は、次を入力します。

```
# /etc/chkconfig cluster off
```

ノードのステータスとクラスタ・データベースのシャットダウン

クラスタ・データベースのシャットダウンは、ノードでメインテナンス操作を実行した後にそのノードを再起動し、「アクティブ(ACTIVE)」ステータスに戻す場合に行うのが適切です。

クラスタ・データベースのシャットダウンを実行した場合、ノードのステータスは「停止(DOWN)」になり、以下のような影響があります。

- DOWN ノードは引続きクラスタの一部と見なされますが、使用可能ではなくなります。

- DOWN ノードは、クラスタ・データベースの更新を取得しなくなります。ただし、ノードを再起動すると、すべての更新が通知されます。

  クラスタ・データベースが更新されないと、CXFS のカーネル部分がアクティブな場合に問題が発生することがあります。つまり、ノードが CXFS へのアクセスを持ち続けると、そのノードのカーネル・レベルでは更新を参照できなくなり、残りのノードがこれらのカーネル・レベルの更新を伝えても応答しなくなります。このため、クラスタは設定の更新に基づいて動作できなくなります。

クラスタ・データベースの再起動

クラスタ・データベースを再起動するには、次を入力します。

```
# /etc/init.d/cluster start
```

通常 CXFS シャットダウン

通常 CXFS シャットダウンは、ノードの CXFS サービスをすべて停止して、CXFS カーネル・メンバーシップ定足数から削除する場合に実行します。通常 CXFS シャットダウンにより、以下が行われます。

- ノードがアクティブなメタデータ・サーバであるファイルシステムを除き、すべてのファイルシステムがアンマウントされる。ノードがアクティブなメタデータ・サーバであるファイルシステムは、シャットダウン後はノードからアクセスできなくなります。
- クラスタ内のこのノードの CXFS カーネル・メンバーシップが終了される。
- 「非アクティブ(INACTIVE)」としてノードにマークが付けられる。

この結果、クラスタ・ディスクが使用できなくなり、クラスタ・データベースの更新はこのノードに伝えられなくなります。ノードを再起動しても、シャットダウン状態のままです。

CXFS サービスをシャットダウンしたノードが特定のファイルシステムのアクティブなメタデータ・サーバの場合、そのファイルシステムは、メタデータ・サーバとして使用可能なノードの 1 つとしてリストされている別のノードによって回復されます。詳細については、135 ページの「GUI を使った CXFS ファイルシステムの定義」または 168 ページの「cmgr を使ったクラスタの変更」を参照してください。選択されるサーバはファイルシステム・クライアントでなければなりません。その他のファイルシステム・クライアントでは、回復プロセス中に遅れが発生します。

メモ: CXFS シャットダウンを実行したノードがメタデータ・サーバとして使用可能な唯一のノードの場合(つまり、ファイルシステムに対してメタデータ・サーバとして使用可能なノードとしてリストされているノードがほかにない場合)、シャットダウンを実行する前に、CXFS GUI または cmgr(1M) コマンドを使用して、すべてのノードからファイルシステムをアンマウントする必要があります。

通常 CXFS シャットダウンを実行するには、次の cmgr(1M) コマンドを入力します。

```
cmgr> stop cx_services on node nodename for cluster clustername
```

また、GUI を使用することもできます。125 ページの「GUI を使った CXFS サービスの停止（通常 CXFS シャットダウン）」を参照してください。

メモ: このアクションにより、1 つのノードの CXFS サービスが停止され、そのノードの停止後に新しい CXFS カーネル・メンバーシップが形成されます。複数のノードでサービスを停止する場合は、このコマンドを複数回入力するか、GUI を使用してタスクを実行してください。

ノードでクラスタ・サービスをシャットダウンすると、そのノードは非アクティブとしてマークされ、CXFS カーネル・メンバーシップの計算時に使用されなくなります。241 ページの「ノードのステータス」を参照してください。

ノードのステータスと通常 CXFS シャットダウン

ノードで通常 CXFS シャットダウンを実行すると、その状態は「非アクティブ(INACTIVE)」になります。したがって、CXFS カーネル・メンバーシップ定足数の計算に影響はありません。211 ページの「通常 CXFS シャットダウン」を参照してください。

通常 CXFS シャットダウンを実行しない場合

以下の状況では、通常 CXFS シャットダウンを実行しないでください。

- ローカル・ノード上。このノードは、クラスタ・マネージャが実行されている CXFS 管理ノード、または GUI が接続されているノードです。
- ノード上で CXFS サービスを停止することによって、CXFS メンバーシップ定足数が失われた場合。

これらの状況において CXFS シャットダウンを実行する場合は、強制 CXFS シャットダウンを実行しなければなりません。213 ページの「強制 CXFS シャットダウン: ローカル・ノードのメンバーシップ取消し」を参照してください。

通常 CXFS シャットダウン後のクラスタへの再設定

ノードは、再起動後にクラスタに再設定されません。ノードは、GUI（125 ページの「GUI を使った CXFS サービスの開始」を参照してください）または次のコマンドを使用して、CXFS サービスを明示的に再度アクティブにした場合のみ、クラスタに再設定されます。

```
cmgr> start cx_services on node nodename for cluster clustername
```

強制 CXFS シャットダウン: ローカル・ノードのメンバーシップ取消し

強制 CXFS シャットダウンは、クラスタが CXFS カーネル・メンバーシップ定足数の条件を下回ることになってもローカル・ノードをシャットダウンする場合に行うのが適切です。

CXFS では以下が行われます。

- ローカル・ノードですべてのクラスタ・ファイルシステムがシャットダウンされる
- クラスタ・ファイルシステムにアクセスしようとすると、I/O エラーが発生する(手動によるファイルシステムのアンマウントが必要な場合があります)
- CXFS カーネル・メンバーシップからこのノードが削除される
- 「非アクティブ(DOWN)」としてノードにマークが付けられる

注意: 強制 CXFS シャットダウンを実行して、クラスタが CXFS カーネル・メンバーシップ定足数を下回った場合、クラスタに異常が発生する可能性があります。

アクティブなメタデータ・サーバで強制シャットダウンを行うと、メンバーシップが直ちに失われます。この時点で別のメタデータ・サーバによって引継ぎ(およびファイルシステムの回復)が行われなければなりません。さもないと、定足数が失われ、続いてすべてのノードで強制シャットダウンが行われます。

定足数を強制的に失わせる強制 CXFS シャットダウンを行った場合、クラスタの残りの部分は、同様にただちに強制シャットダウンを行わなければならないため、異常ノードをリセットしません。

強制 CXFS シャットダウンを実行するには、次の cmgr(1M) コマンドを入力して、ローカル・ノードの CXFS カーネル・メンバーシップを取消します。

```
cmgr> admin cxfs_stop
```

このアクションは、GUI を使用して実行することもできます。128 ページの「GUI を使ったローカル・ノードのメンバーシップ取消し」を参照してください。このアクションは、CXFS カーネル・メンバーシップ定足数が失われた後、カーネルによって自動的に実行することもできます。

ノードのステータスと強制 CXFS シャットダウン

強制 CXFS シャットダウンの後も、ノードは引続き設定されたクラスタの一部と見なされており、クラスタ・データベースを伝えるとき(これらのサービスはまだ実行されています)や、クラスタ・データベース (fs2d) メンバーシップ定足数を計算するとき(これにより、クラスタのその他の部分の定足数が失われ、その他のノードで強制シャットダウンが行われます)に考慮に入れられます。状態は、「非アクティブ(INACTIVE)」です。

データベースの整合性を保つには、このノードがアクセス可能なままで、クラスタ・インフラストラクチャ・デーモンを実行し続けることが重要です。特に、プール内の半数を超えるノードが停止していたり、イン

フラストラクチャ・デーモンを実行していない場合は、クラスタ・データベースの更新が伝えられらなくなり、結果として整合性が失われます。安全性の点から、今後も使用不可能なノードはクラスタとプールから削除してください。以下を参照してください。

- 117 ページの「GUI を使ったクラスタ内のノードの追加と削除」または 168 ページの「cmgr を使ったクラスタの変更」
- 121 ページの「GUI を使ったノードの削除」または 162 ページの「cmgr を使ったノードの削除」

強制 CXFS シャットダウン後のクラスタへの再設定

強制 CXFS シャットダウンの後は、ノードが再起動されるか、次の cmgr(1M) コマンドを入力してノードに対して明示的に CXFS カーネル・メンバーシップを許可するまで、ローカル・ノードは CXFS カーネル・メンバーシップを再開しません。

```
cmgr> admin cxfs_start
```

この手順は、GUI を使用して実行することもできます。128 ページの「GUI を使ったローカル・ノードのメンバーシップ許可」を参照してください。

強制シャットダウンを CXFS 管理ノードで実行した場合は、そのノードをクラスタに戻す前にノードで CXFS を再開する必要があります。ノードがクラスタ内にあってアクティブなことがまだクラスタ・データベースに示されているときにこれを行うと、CXFS カーネル・メンバーシップ・デーモンがノードによって再開されます。したがって、この操作は、データベースをリセットした後や、CXFS サービスを停止した後に行うことができます。

例は、次のとおりです。

```
cmgr> admin cxfs_start
```

シリアル・ハードウェア・リセット機能と強制 CXFS シャットダウン

注意: シリアル・ハードウェア・リセット機能を使用して IRIX ノードで強制シャットダウンを実行して、シャットダウンによってクラスタ定足数が失われなかった場合、ノードは適切なノードによってリセット(再起動)されます。

リセットについての詳細は、24 ページの「シリアル・ハードウェア・リセット」を参照してください。

再起動時の CXFS の再開の回避

chkconfig(1M) に対する cxfs_cluster フラグは、CXFS 管理ノード上の clconfd デーモンを制御します。cxfs_client フラグは、クライアントのみのノード上の cxfs_client デーモンを制御します。

これらのフラグをオフにした場合、デーモンは次回再起動時に開始されず、カーネルはクラスタに設定されません。システムやハードウェアのアップグレードなどのメインテナンス作業を行う場合にクラスタからノードを一時的に削除する場合は、ノードを再起動する前にオフにしておくと便利です。

CXFS 管理ノードでのクラスタ・データベースの再起動を回避するには、cluster オプションを off に設定します。

たとえば、CXFS 管理ノードで以下を実行します。

```
# /etc/chkconfig cxfs_cluster off
# /etc/chkconfig cluster off
# reboot
```

ログ・ファイル管理

ディスクがいっぱいにならないよう、ログ・ファイルは、少なくとも毎週ローテーションさせます。以下の節では、スクリプトの例を示します。ログ・レベルについての詳細は、127 ページの「GUI を使ったログ・グループの設定」を参照してください。

すべてのログ・ファイルのローテーション

すべてのファイルを新しい場所にコピーするには、/var/cluster/cmgr-scripts/rotatelogs スクリプトを実行できます。このスクリプトは、日付と月名を接尾辞として、ログ・ファイルを保存します。1 日にスクリプトを 2 回実行した場合、最新のログ・ファイルは、以前に保存されたコピーに追加されます。root crontab ファイルには、このスクリプトを毎週実行するためのエントリが含まれています。

以下に、スクリプトの構文を示します。

```
/var/cluster/cmgr-scripts/rotatelogs [-h] [-d|-u]
```

オプションが指定されていない場合、ログ・ファイルはローテーションされます。以下に、オプションを示します。

-h　　ヘルプ・メッセージを出力します。ログ・ファイルはローテーションされず、その他のオプションは無視されます。

-d　現在のログ・ファイルをローテーションする前に、1 週間前以前の保存されているログ・ファイルを削除します。このオプションと -u を同時に指定することはできません。

-u　現在のログ・ファイルをローテーションする前に、保存されているすべてのログ・ファイルを無条件に削除します。このオプションと -d を同時に指定することはできません。

大容量のログ・ファイルのローテーション

大容量のファイルを新しい場所にコピーするには、次のようなスクリプトを使用できます。新しい場所にあるファイルは、このスクリプトが実行されるたびに上書きされます。

```
#!/bin/sh
# Argument is maximum size of a log file (in characters) - default: 500000

size=${1:-500000}
find /var/cluster/ha/log -type f ! -name '*.OLD' -size +${size}c -print | while read log_file; do
        cp ${log_file} ${log_file}.OLD
        echo '*** LOG FILE ROTATION ' `date` '***' > ${log_file}
done
```

54 ページの「CXFS 管理ノードの /etc/config/cad.options」および 55 ページの「CXFS 管理ノードの /etc/config/fs2d.options」も参照してください。

ボリューム管理

CXFS では、XVM ボリューム・マネージャを使用します。XVM を使用することで、多数のディスクを高トランザクション・レート、高帯域幅、および高信頼性のファイルシステムに結合できます。CXFS では、以下を提供するために XVM を使用します。

- ディスクのストライプ
- ミラー化
- 結合
- 高度な回復機能

メモ: xvm(1M) コマンドは、CXFS 管理ノードで実行してください。CXFS デーモンを開始する前に XVM コマンドを実行しようとすると、警告メッセージが表示され、XVM のローカル・ドメインに移動します。

XVM のローカル・ドメイン内にいる場合はファイルシステムを定義できますが、後から CXFS を開始してもファイルシステムは表示されません。CXFS を開始すると、XVM はクラスタ・ドメインに切替わり、ファイルシステムは認識されなくなります。これは、ファイルシステムをローカル・ドメイン内で定義したためです。クラスタ・ドメインでファイルシステムを使用するには、give コマンドを使用しなければなりません。したがって、ボリュームは、クラスタ・ドメイン内で直接定義することをお薦めします。

詳細については、『XVM Volume Manager Administrator's Guide』を参照してください。

ディスク管理

この節では、バックアップ、NFS、Quota、および Samba に対する CXFS の違いについて説明します。

バックアップ

CXFS では、VERITAS NetBackup や Legato NetWorker などの商用バックアップ・パッケージを使用して、ローカル・エリア・ネットワーク (LAN: Local Area Network) から独立したバックアップを作成できます。これらのパッケージを使用すると、バックアップ作業をバックアップ・サーバに統合できると同時に、低速の LAN ではなくストレージ・エリア・ネットワーク (SAN: Storage Area Network) を通じてデータを渡すことができます。

たとえば、バックアップ・サーバとして指定された SAN 上のホストでバックアップ・パッケージを実行できます。このサーバでは、SAN ディスクに接続されているテープ・ドライブとのチャネル接続を使用できます。また、CXFS を通じてファイルシステムを表示したり、バックアップ・サーバを通じてディスクからテープ・ドライブに直接データを転送するバックアップ・アプリケーションを実行できます。

これにより、非常に容量の大きいファイルシステムの場合でも、バックアップ帯域幅をストレージ・サイズに合わせて拡張できます。ディスク・チャネルの数、バックアップ・サーバのサイズ、およびテープ・チャネルの数は、バックアップ帯域幅の条件を満たすよう増やすことができます。

NFS

NFS サーバを CXFS 上に配置して、クラスタの一部ではないコンピュータ・システムでファイルシステムを共有できます。これは任意のノードで実行できます。

Quota

XFS の Quota は、サポートされていますが、Quota のマウント・オプションは、ファイルシステムのすべてのマウントで同じでなければなりません。Quota は、クラスタ内のどこからでも XFS ファイルシステムと同じように管理できます。

Samba

CXFS 上で Samba を実行し、Windows マシンで CXFS をサポートしてファイルシステムにアクセスできるようにすることが可能です。

ファイルシステムのメインテナンス

通常、ファイルシステム情報は IRIX ノード上の `/etc/fstab` に格納されますが、CXFS ファイルシステム情報はクラスタ全体に関連しているため、複製されたクラスタ・データベースに格納されます。

管理者として、CXFS クラスタ・マネージャ・ツールを使用して CXFS ファイルシステム設定を行います。GUI についての詳細は 132 ページの「GUI を使ったファイルシステム・タスク」、`cmgr`(1M) についての詳細は 166 ページの「`cmgr` を使ったクラスタ・タスク」をそれぞれ参照してください。

情報は、クラスタ全体に自動的に同じ方法で伝えられます。クラスタ設定デーモンは、情報が利用可能になるとすぐに、この情報に基づいて各ノードでファイルシステムをマウントします。

CXFS ファイルシステムは、クラスタ内のすべてのノードで自動的にマウントされます。クラスタがアクティブな場合は、新しい CXFS ファイルシステムを設定に追加できます。

クラスタ設定内の変更がクラスタ設定デーモンによって検出されると、設定されているすべてのファイルシステムで `mount -a` コマンドと同等のコマンドが実行されます。

注意: ファイルシステムがマウントされている間は、CXFS ファイルシステム定義を変更したり削除しないでください。CXFS ファイルシステム定義を変更または削除する場合は、まずファイルシステムをアンマウントしてから行い、変更後はファイルシステムを再びマウントしてください。

ファイルシステムのマウント

マウント情報は、GUI の「ファイルシステムのマウント(Mount a Filesystem)」タスク(「新規ファイルシステムの設定(Set Up a New Filesystem)」設定ガイド・タスクの一部)、または `cmgr`(1M) のサブコマンド `modify` を使用して入力します。以下を参照してください。

- GUI を使用したマウントについての詳細は、76 ページの「GUI を使った新規 CXFS ファイルシステムの設定」および 135 ページの「GUI を使った CXFS ファイルシステムの定義」を参照してください。
- cmgr を使用した新しいファイルシステムの定義およびマウントについての詳細は、168 ページの「cmgr を使ったクラスタの変更」を参照してください。
- すでに定義されているが現在はアンマウントされているファイルシステムのマウントについての詳細は、178 ページの「cmgr を使った CXFS ファイルシステムの定義」を参照してください。

正しく定義およびマウントされていれば、CXFS ファイルシステムは、複製されたデータベースに収集された情報に基づいて、ローカル・クラスタ設定デーモン clconfd によって各ノードで自動的にマウントされます。ファイルシステム設定がデータベースに入力されたら、ユーザによる操作は必要ありません。

注意: CXFS ファイルシステムをマウントするために、mount(1M) コマンドは使用しないでください。このコマンドを使用すると、ほかのノードと異なる方法でファイルシステムが使用されるため、データが失われたり、破壊される可能性があります。

CXFS ファイルシステムは、クラスタ内のすべてのノードでマウントされているか、またはまったくマウントされていないかのいずれかでなければなりません。その他の場合は、ファイルシステムがノードごとに異なる方法でマウントされているため、データの損失や破壊につながる可能性があります。GUI および cmgr では、クラスタ内のノードのサブセットにファイルシステムをマウントすることはできません。

CXFS を使用している場合、マウント・ポイントはネストできません。つまり、/usr や /usr/home のように、ファイルシステム内にファイルシステムを含めることはできません。

ファイルシステムのアンマウント

CXFS ファイルシステムをアンマウントするには、GUI の「ファイルシステムのアンマウント(Unmount a Filesystem)」タスク、または cmgr のサブコマンド admin を使用します。詳細については、138 ページの「GUI を使った CXFS ファイルシステムのアンマウント」または 184 ページの「cmgr を使った CXFS ファイルシステムのアンマウント」を参照してください。

これらのタスクによって、クラスタ内のすべてのノードからファイルシステムがアンマウントされます。このアクションはすべてのノードでアンマウントを実行しますが、ファイルシステムが使用中の場合は、いくつかのノードで操作が失敗することもあります。アクティブなメタデータ・サーバでは、すべての CXFS クライアントがファイルシステムを正常にアンマウントするまで、アンマウントは成功しません。すべてのノードは、成功するまでアンマウントを再試行しますが、すべてのノードでファイルシステムがアンマウントされたことを示す一元化されたレポートはありません。すべてのノードからファイルシステムがアンマウントされていることを確認するには、以下のいずれかを実行します。

- ファイルシステムがアンマウントされていることを示すメッセージを、メタデータ・サーバの SYSLOG ファイルをチェックして確認する

- メタデータ・サーバで GUI または cmgr を実行し、サーバからファイルシステムを無効にした後、ファイルシステムが完全に無効になっていることが GUI で表示されるまで待つ（一部の CXFS クライアントにまだマウントされている場合はエラーになり、残っているクライアントが GUI に表示されます）

ファイルシステムの拡張

CXFS ファイルシステムを拡張するには、以下を実行します。

1. CXFS ファイルシステムをアンマウントします。詳細については、138 ページの「GUI を使った CXFS ファイルシステムのアンマウント」または 184 ページの「cmgr を使った CXFS ファイルシステムのアンマウント」を参照してください。
2. ファイルシステムを XFS ファイルシステムとしてマウントします。『IRIX Admin: Disks and Filesystems』を参照してください。
3. ファイルシステムを拡張するには、xfs_growfs(1M) コマンドを使用します。
4. umount(1M) コマンドを使用して、XFS ファイルシステムをアンマウントします。
5. ファイルシステムを CXFS ファイルシステムとしてマウントします。137 ページの「GUI を使った CXFS ファイルシステムのマウント」または 178 ページの「cmgr を使った CXFS ファイルシステムの定義」を参照してください。

ダンプと復元

ダンプおよび復元の手順は、アクティブなメタデータ・サーバから実行しなければなりません。xfsdump(1M) コマンドおよび xfsrestore(1M) コマンドでは、メタデータ・サーバだけで機能する特殊なシステム・コールが利用されます。

ダンプ・プロセス中は、ファイルシステムのアクティブなクライアントが存在してもかまいません。

クラスタ化された環境では、1 つの CXFS ファイルシステムに多くの CXFS クライアントとアクティブなメタデータ・サーバが同時に直接アクセスする場合があります。フェイルオーバーや単なるメタデータ・サーバの再割当てにより、時間の経過と共に、1 つのファイルシステムが多くのメタデータ・サーバで使用されるようになることがあります。したがって、xfsdump は、一貫したインベントリを保つために、過去のダンプのインベントリにアクセスしなければなりません。これは、この情報が別のノードにある場合にも該当します。以下のいずれかの方法を使用して、クラスタ内のメタデータ・サーバとして使用可能なノードからインベントリにアクセスできるようにすることをお薦めします。

- インベントリを共有ファイルシステムに移動する

  例は、次のとおりです。

– インベントリが現在含まれているノードで、以下を入力します。

```
# cp -r /var/xfsdump /shared_filesystem
# mv /var/xfsdump /var/xfsdump.bak
# ln -s ../shared_filesystem /var/xfsdump
```

– クラスタ内のその他のすべての IRIX ノードで、以下を入力します。

```
# mv /var/xfsdump /var/xfsdump.bak
# ln -s ../shared_filesystem /var/xfsdump
```

- NFS 共有ファイルシステムを使用して、ディレクトリをエクスポートする

  例は、次のとおりです。

  – インベントリが現在含まれているノードで、/var/xfsdump を /etc/exports に追加し、以下を入力します。

```
# exportfs -a
```

  – クラスタ内のその他のすべての IRIX ノードで、以下を入力します。

```
# mv /var/xfsdump /var/xfsdump.bak
# ln -s /hosts/hostname/var/xfsdump /var/xfsdump
```

メモ: 共有するのは /var/xfsdump ディレクトリで、/var/xfsdump/inventory ディレクトリではありません。インベントリがさまざまなノードに格納されている場合は、xfsinvutil(1M) を使用してそれらを単一の共通インベントリに結合し、クラスタ間でその共通インベントリを共有できます。

クラスタ・データベースのバックアップと復元

データベースを保存しておき、ディスクがクラッシュした場合や不正な変更が行われた場合などに保存時の状態に復元できるようにしたいときは、データベースのバックアップを実行することをお薦めします。

クラスタ・データベースのバックアップを実行するには、cdbBackup(1M) コマンドを使用します。

復元を行うには、以下を実行します。

1. クラスタ・サービスを停止します。
2. 古いデータベースを削除します。
3. cdbRestore コマンドを使用します。

詳細については、cdbRestore(1M) のマン・ページを参照してください。

chkconfig フラグ

メモ: これらのフラグは、通常は管理者が chkconfig コマンドを使用して操作することはなく、GUI または cmgr によって設定または設定の解除が行われます。これらのフラグでは、クラスタではなく、プロセスのみを制御します。クラスタを制御するプロセスを停止しても、クラスタは停止されません。プロセスを開始すると、CXFS サービスがデータベースでアクティブとマークされている場合のみ、クラスタが開始されます。

CXFS には、chkconfig(1M) コマンドに対して以下のフラグがあります。

- cluster(CXFS 管理ノードの場合)。ほかのクラスタ管理デーモン(複製されたクラスタ・データベースなど)を制御します。これがオフの場合、データベース・デーモンは次回再起動されるときに開始されず、ほかのノードでクラスタ設定を変更してもデータベースのローカル・コピーは更新されません。これにより、後で問題が発生することがあります。特に、ノードの過半数でデータベース・デーモンが実行されていない場合は、問題が発生する可能性が高くなります。データベース・デーモンが実行されていない場合、クラスタ・データベースにローカルでアクセスすることはできず、ノードはクラスタに設定されません。
- cxfs_cluster(CXFS 管理ノードの場合)。clconfd デーモン、およびシステム・シャットダウン中に cxfs_shutdown コマンドが使用されるかどうかを制御します。

  cxfs_shutdown コマンドは、再起動の前にクラスタからの取消しを試みます。取消しが試みられない場合、再起動は失敗と見なされ、その他のノードでは回復が必要となります。
- cxfs_client(クライアントのみのノードの場合)。cxfs_client デーモンを開始するかどうかを制御します。

システム調整可能パラメータ

表6-1 に、CXFS で使用できるシステム調整可能パラメータを示します。

表6-1 システム調整可能パラメータ

パラメータ	説明
cms_reset_timeout	リセット要求の認証に対する clconfd の待ち時間の秒数を指定します。0 を指定すると、待ち時間は無限になります。これはデフォルトです。ゼロ以外の値を設定し、待ち時間がタイムアウトになった場合、CXFS では、cms_reset_timeout_action パラメータによって指定されるアクションが実行されます。このパラメータは、ランタイム時に変更できます。 **注意:** タイムアウトの設定は、システムへの影響をよく理解した上で行ってください。このパラメータの変更は、一般的に推奨されていません。 このパラメータは、/var/sysgen/mtune/cell にあります。
cms_reset_timeout_action	clconfd がリセット要求(cms_reset_timeout によって決定されます)を確認しない場合に行われるアクションを指定します。cms_reset_timeout_action は、ランタイム時に変更でき、以下のいずれかに設定されます。 • 0 - リセット確認を待っているノードを強制的にクラスタから取消します。ノードが定足数を失ったときに起こる強制シャットダウンと同じです(デフォルト)。clconfd がまだ存在しており、正常に機能している場合は、カーネル cms デーモンが再開され、ノードではクラスタへの再設定が試みられます。 • 1 - ペンディングのリセットをすべてクリアして、続行します(つまり、確認をフェイクします)。 • 2 - ローカル・ノードをパニックさせます。 このパラメータは、/var/sysgen/mtune/cell にあります。

パラメータ	説明
cxfsd_min	各 CXFS ファイルシステムで実行する cxfsd スレッドの最小数を指定します。 cxfsd スレッドは、CXFS 内の遅延割当てバッファ用のディスク・ブロック割当てと、メタデータ・サーバによってローカル・キャッシュから削除されるファイルのバッファ・データのフラッシュを行います。スレッドは、ファイルシステムのマウント時に割当てられます。マウント時の cxfsd_min パラメータの値は、ファイルシステムがアンマウントされるまで有効です。 cxfsd_min の正しい値は、1～256 の範囲の整数です。デフォルトは 4 です。 このパラメータは、/var/sysgen/mtune/cxfs にあります。

パラメータ	説明
cxfsd_max	各 CXFS ファイルシステムで実行する cxfsd スレッドの最大数を指定します。マウント時の cxfsd_max パラメータの値は、ファイルシステムがアンマウントされるまで有効です。 cxfsd_max の正しい値は、8～65536 の範囲の整数です。デフォルトは 256 です。cxfsd_max の値は、cxfsd_min に指定した値未満にすることはできません。 このパラメータは、/var/sysgen/mtune/cxfs にあります。
cxfs_relocation_ok	再設定を有効または無効のどちらにするかを指定します。 • 0 - 再設定を無効にします。 • 1 - 再設定を有効にします。 **メモ:** 再設定はデフォルトで無効になっています。再設定を有効にしないことをお薦めします。 このパラメータは、/var/sysgen/mtune/cxfs にあります。

IRIS FailSafe との同時実行

この章では、以下の内容について説明します。

- 「CXFS と FailSafe を同時に実行する理由」
- 228 ページの「同時実行できるリリース・レベル」
- 228 ページの「同時実行クラスタのサイズ」
- 228 ページの「クラスタのタイプ」
- 230 ページの「メタデータ・サーバ・ノードのタイプ」
- 230 ページの「GUI の違い」
- 230 ページの「切替え」
- 230 ページの「ネットワーク・インタフェース」
- 231 ページの「CXFS タイブレーカー・ノードと同時実行」
- 231 ページの「メタデータ・サーバとフェイルオーバー・ドメイン」
- 231 ページの「CXFS FailSafe のリソース・タイプ」

305 ページの「同時実行クラスタ内の通信パス」も参照してください。

CXFS と FailSafe を同時に実行する理由

CXFS を使用することで、コンピュータのグループは、高いパフォーマンスを保ったまま、一貫性のある方法で大量のデータを共有できます。

SGI IRIS FailSafe 製品には、高可用性サービスを提供するための一般的な機能が用意されています。クラスタ内の IRIX ノードの 1 つ、またはそのノードのコンポーネントの 1 つに異常が発生した場合、その異常ノードの高可用性サービスはクラスタ内の別の IRIX ノードで再開されます。CXFS クライアントからは、代替ノードのサービスと、異常が発生する前の元のサービスは区別できず、元のサービスがクラッシュして、ただちに再開されたように見えます。つまり、CXFS クライアントでは、高可用性サービスが短時間中断されたことしかわかりません。

したがって、CXFS クラスタ内で FailSafe を使用して（同時実行とも呼ばれる）、CXFS ファイルシステム上で実行される高可用性サービス（NFS や Web など）を提供できます。この組合わせにより、クラスタ・システムにおいて、パフォーマンスの高い共有データ・アクセスが高可用性アプリケーションに提供されます。

同時実行できるリリース・レベル

CXFS 6.5.10 以降と IRIS FailSafe 2.1 以降（および関連するパッチ）は、同じシステムにインストールして実行できます。

同時実行クラスタのサイズ

同時実行クラスタ内の IRIX ノードのサブセットは、FailSafe ノードとして使用されるよう設定できます。したがって、同時実行クラスタでは、FailSafe を実行するノードを最大 8 つまで使用できます。

CXFS クラスタ内のすべてのノードでは CXFS が実行され、これらの IRIX ノードの最大 8 つでは FailSafe も実行できます。CXFS と FailSafe を実行している場合も、プール、クラスタ、およびクラスタ設定はそれぞれ 1 つだけです。

実作業用のクラスタは、サーバとして使用可能なノード 3 つ以上で設定することをお薦めします。最大で 32 のノードを使用して、最大 16 の CXFS 管理ノードを設定できます（シリアル・ハードウェア・リセット・ケーブルが接続されていて、サーバとして使用可能なノードが 2 つだけのクラスタはサポートされていますが、この設定には特有の問題があります。33 ページの「サーバとして使用可能なノードを 2 つだけ含むクラスタでの回復の問題」を参照してください）。

クラスタのタイプ

クラスタは、以下の 3 つのいずれかのタイプに設定できます。

- `FailSafe`。この場合、すべてのノードのタイプも `FailSafe` になります。これらのノードは IRIX ノードでなければなりません。
- `CXFS`。この場合、すべてのノードのタイプが `CXFS` になります。これらのノードには、IRIX ノード、または Solaris や Windows などのほかのオペレーティング・システムが実行されているノードを使用できます。
- `CXFS and FailSafe`（同時実行）。この場合、すべてのノードは、`CXFS`（IRIX ノード、またはほかのオペレーティング・システムが実行されているノード）と `CXFS and FailSafe`（IRIX ノード）が混在したタイプになり、アプリケーション・レベルの高可用性を得るための FailSafe と CXFS が使用されます。

メモ: FailSafe タイプのノードだけで同時実行クラスタを設定することも可能ですが、この設定はサポートされていません。

図7-1に、使用できる組合わせと使用できない組合わせをいくつか示します。

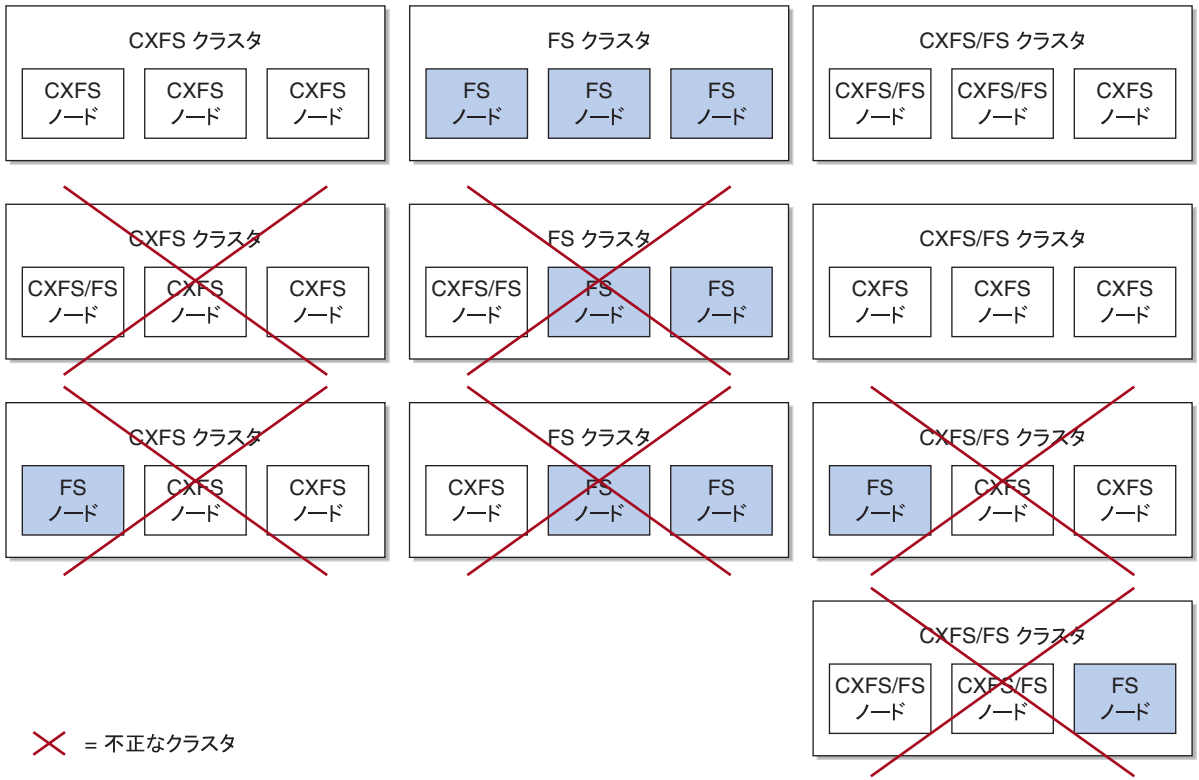


図7-1 クラスタとノード・タイプの組合わせ

メタデータ・サーバ・ノードのタイプ

使用可能なすべてのメタデータ・サーバ・ノードは、以下のいずれかのタイプでなければなりません。

- `CXFS`
- `CXFS and FailSafe`

GUI の違い

`cmgr`(1M) (`cluster_mgr`) コマンドは 1 つですが、CXFS と FailSafe にはそれぞれ別のグラフィカル・ユーザ・インタフェースがあります。CXFS 設定の管理は CXFS GUI で行い、FailSafe 設定の管理は FailSafe GUI で行ってください。`cmgr` ではどちらも管理できます。

切替え

CXFS GUI または `cmgr`(1M) を使用して、既存の `FailSafe` クラスタとノードのタイプを `CXFS` または `CXFS and FailSafe` に切替えることができます。FailSafe GUI を使用すると、並行して操作を実行できます。切替え後のノードは、アプリケーション・レベルの高可用性を提供するために FailSafe で使用したり、クラスタ・ファイルシステムを提供するために CXFS で使用できます。107 ページの「GUI を使って CXFS 用に既存の FailSafe クラスタを再設定」を参照してください。

ただし、以下の点に注意してください。

- 対応する高可用性 (HA) サービスまたは CXFS サービスがアクティブな場合は、ノードのタイプを変更することはできません。まず、ノードのサービスを停止してください。
- クラスタは、そのノードに対してオンにするすべての機能（FailSafe または CXFS、あるいはその両方）をサポートしていなければなりません。つまり、クラスタのタイプが `CXFS` の場合、すでにクラスタの一部であるノードのタイプを `FailSafe` に変更することはできません。ただし、ノードがクラスタのすべての機能をサポートする必要はありません。つまり、`CXFS and FailSafe` クラスタで `CXFS` ノードを使用することができます。

161 ページの「`cmgr` を使ったノードの CXFS または FailSafe への切替え」および 169 ページの「`cmgr` を使ったクラスタの CXFS または FailSafe への切替え」を参照してください。

ネットワーク・インタフェース

FailSafe では少なくとも 2 つのネットワーク・インタフェースが必要ですが、CXFS では、ハートビート・メッセージとコントロール・メッセージの両方に 1 つのインタフェースだけを使用します（CXFS GUI では、

ネットワークに対してハートビートまたはコントロールの一方だけを選択できるように見えますが、このように選択しないでください)。

同じノードで FailSafe と CXFS を使用する場合、CXFS に対しては優先度 1 のネットワークだけが使用されるので、このネットワークは、ハートビート・メッセージとコントロール・メッセージの両方を使用できるよう設定する必要があります。

メモ: CXFS は 2 番目のネットワークにフェイルオーバーされません。ノードが CXFS and FailSafe の場合、優先度 1 のネットワークに異常が発生すると、CXFS には異常が発生しますが、FailSafe サービスは 2 番目のネットワークに移動できます。

優先度 1 のネットワークがなくなったために CXFS がノードをリセットすると、そのノードは FailSafe によって FailSafe メンバーシップから削除されます。これにより、リソース・グループはクラスタ内の別の FailSafe ノードにフェイルオーバーされます。

CXFS タイブレーカー・ノードと同時実行

FailSafe ノードが 2 つだけの場合は、CXFS タイブレーカー・ノードを使用しないでください。

メタデータ・サーバとフェイルオーバー・ドメイン

メタデータ・サーバのリストは、フェイルオーバー・ドメインのリストに完全に一致する必要があります(名前と名前の順序の一致)。

CXFS FailSafe のリソース・タイプ

FailSafe には、CXFS ファイルシステムを使用するアプリケーションをフェイルオーバーさせるときに使用できる CXFS リソース・タイプが用意されています。CXFS リソースは、CXFS ファイルシステムに依存するリソースが含まれるリソース・グループに追加しなければなりません。CXFS リソース・タイプ名は、CXFS ファイルシステムのマウント・ポイントになります。

CXFS リソース・タイプは以下の特性を持ちます。

- CXFS ファイルシステムがローカル・ノードにマウントされるまで、CXFS ファイルシステムに依存するすべてのリソースを開始しません。
- CXFS リソース・タイプに対して start および stop アクション・スクリプトを実行しても、それぞれ CXFS ファイルシステムのマウントとアンマウントは行われません (start スクリプトは、CXFS ファイルシステムが使用可能になるまで待機します。stop スクリプトは何も実行しませんが、FailSafe では

このスクリプトが存在している必要があります)。CXFS ファイルシステムのマウントとアンマウントを行うには、ユーザは CXFS GUI または cmgr(1M) コマンドを使用する必要があります。

- 異常がないかどうか CXFS ファイルシステムをモニタします。
- CXFS メタデータ・サーバで実行する必要があるアプリケーションに対しては、CXFS タイプは、リソース・アプリケーション・フェイルオーバーの発生時に CXFS メタデータ・サーバを再設定します(オプション)。この場合、リソース・グループのアプリケーション・フェイルオーバー・ドメイン (AFD: Application Failover Domain) は、CXFS メタデータ・サーバおよびメタデータ・サーバのバックアップ・ノードで構成されている必要があります。

NFS サーバがエクスポートする CXFS ファイルシステムは、CXFS GUI または cmgr(1m) ツールを使用して、フェイルオーバー・ドメインのすべてのノードにマウントする必要があります。

たとえば、以下に、/FC/lun0_s6 にマウントされている CXFS ファイルシステムに基づいて NFS、CXFS、および statd_unlimited を作成するために使用するコマンドを示します (この例では、test-cluster という名前のクラスタが定義されていて、 cxfs-fp という名前のフェイルオーバー・ポリシーと、このポリシーに基づく cxfs-group という名前のリソース・グループが作成済みであることを想定しています。また、改行して読みやすくしてあります)。

```
cmgr> define resource /FC/lun0_s6 of resource_type CXFS in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to create with set command:

Type Specific Attributes - 1: relocate-mds


No resource type dependencies to add

resource /FC/lun0_s6 ? set relocate-mds to false
resource /FC/lun0_s6 ? done


==========================================

cmgr> define resource /FC/lun0_s6 of resource_type NFS in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to create with set command:

Type Specific Attributes - 1: export-info
Type Specific Attributes - 2: filesystem
```

```
No resource type dependencies to add

resource /FC/lun0_s6 ? set export-info to rw
resource /FC/lun0_s6 ? set filesystem to /FC/lun0_s6
resource /FC/lun0_s6 ? done


==========================================
cmgr> define resource /FC/lun0_s6/statmon of resource_type statd_unlimited in cluster
test-cluster
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to create with set command:

Type Specific Attributes - 1: ExportPoint


Resource type dependencies to add:

Resource Dependency Type - 1: NFS


resource /FC/lun0_s6/statmon ? set ExportPoint to /FC/lun0_s6
resource /FC/lun0_s6/statmon ? add dependency /FC/lun0_s6 of type NFS
resource /FC/lun0_s6/statmon ? done

==========================================
cmgr> define resource_group cxfs-group in cluster test-cluster
Enter commands, when finished enter either "done" or "cancel"

resource_group cxfs-group ? set failover_policy to cxfs-fp
resource_group cxfs-group ? add resource /FC/lun0_s6 of resource_type NFS
resource_group cxfs-group ? add resource /FC/lun0_s6 of resource_type CXFS
resource_group cxfs-group ? add resource /FC/lun0_s6/statmon of resource_type statd_unlimited
resource_group cxfs-group ? done
```

リソース・グループとフェイルオーバー・ドメインについての詳細は、『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。

Trusted IRIX と CXFS

DMF (Data Migration Facility) および TMF (Tape Management Facility) を使用して、CXFS を SGI Trusted IRIX クラスタで使用できるようになりました。

CXFS と Trusted IRIX を同時に実行したい場合は、クラスタ内のすべてのノードで Trusted IRIX を実行することが推奨されています。すべてのノードが IRIX ノードまたは Trusted IRIX ノードでなければなりません（マルチ OS クラスタでは Trusted IRIX を実行できません）。

CXFS と Trusted IRIX のインストールのヒント

実行する予定のソフトウェア製品（Trusted IRIX、CXFS、DMF、TMF など）をすべて同時にインストールすることをお薦めします。

これらの製品をインストールした後で、以下を行う必要があります。

1. システム・コンソールから、「System Maintenance」メニューに移動します。例は、次のとおりです。

   ```
   # init 0
   ```

   システムが自動的にマルチユーザ・モードで再起動するよう設定されている場合、このメニューを表示するには、Esc キーを押す必要があります。

2. メニューから 5 を選択して、コマンド・モニタを開始します。

   ```
   System Maintenance Menu

   1) Start System
   2) Install System Software
   3) Run Diagnostics
   4) Recover System
   5) Enter Command Monitor

   Option? 5
   ```

3. single コマンドを使用して、シングルユーザ・モードにします。

   ```
   >> single
   ```

4. プロンプトが表示されたら、パスワードを入力します。

5. root ディレクトリに移動します。

   ```
   # cd /
   ```

6. Trusted IRIX および CXFS に対して以下の属性を設定します。

   ```
   # suattr -C all+eip
   ```

7. Trusted IRIX 設定コマンドを実行します。これにより、ファイルに対して適切な拡張属性を設定します。

   ```
   # /etc/trix.config
   ```

詳細については、以下を参照してください。

- 『Trusted IRIX Read Me First Notice』
- 『Trusted IRIX/CMW Security Features User's Guide』

強制アクセス・コントロール

Trusted IRIX と IRIX が混在するクラスタの場合、IRIX CXFS クライアントは、Trusted IRIX サーバにアクセスする際に、その認証情報に関連付けられている強制アクセス・コントロール (MAC: Mandatory Access Control) ラベルを求められますが、このラベルはありません。この問題を解決するために、以下のいずれかの方法で MAC ラベルが提供されます。

- `eag:mac-ip=label` オプションを使用してファイルシステムをマウントし、IRIX CXFS クライアントに使用するラベルを指定できます。
- このマウント・オプションが使用されていない場合は、IRIX の元のノードに対する `rhost` データベース・エントリのデフォルトのラベルが使用されます。
- `rhost` データベース・エントリが使用できないか、無効の場合は、`msenlow, minthigh` というラベルが使用されます。

ステータスのモニタ

システムのステータスは、以下の方法で表示できます。

メモ: 管理タスクは、GUI が CXFS 管理ノード（`cluster_admin` ソフトウェア・パッケージがインストールされているノード）に接続されている場合は GUI を、CXFS 管理ノードにログインしている場合は、`cmgr`(1M) コマンドを使用して実行する必要があります。管理コマンドは、CXFS 管理ノードで実行してください。`cxfs_info`(1M) ステータス・コマンドは、クライアントのみのノードで実行します。

- `/var/cluster/ha/logs` のログ・ファイルをモニタする
- GUI または `tail`(1) コマンドを使用して、SYSLOG ファイルの最後を表示する
- GUI の表示領域または `cluster_status`(1M) コマンドを使用して、クラスタの状態を継続的に監視する
- GUI または `cmgr`(1M) コマンドのいずれかを使用して、個々のノードまたはクラスタのステータスを照会する
- `ls`(1) コマンドを使用して、ファイルシステムを手動でテストする
- PCP (Performance Co-Pilot) を使用してシステムをモニタする。PCP を使用すると、すべてのディスクとクラスタ内のすべてのノードの読取り/書込みスループットと I/O 負荷分散をモニタできます。アクティビティは、グラフィカルに表示したり、警告の生成に使用したり、後から解析できるようにアーカイブしておくことができます。また、XVM 統計もモニタできます。『Performance Co-Pilot User's and Administrator's Guide』と『Performance Co-Pilot Programmer's Guide』、および `dkvis`(1)、`pmie`(1)、`pmieconf`(1)、`pmlogger`(1) のマン・ページを参照してください。

  メモ: PCP パッケージの XVM 統計はデフォルトではインストールされないため、手動でインストールする必要があります。43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。

以下の節では、これらのタスクの一部を実行するための手順について説明します。

ログ・ファイル

問題に対してリストされる以下のログ・ファイルをモニタする必要があります。

- /var/adm/SYSLOG（システム・ログ。クラスタが形成されていることを示す Membership delivered メッセージを検索）
- /var/cluster/ha/log/cad_log（GUI および clconfd からのイベント）
- /var/cluster/ha/log/clconfd_*hostname*（カーネルのステータス）
- /var/cluster/ha/log/cli_*hostname*（コマンド・ライン・インタフェース・ログ）
- /var/cluster/ha/log/cmond_log（その他のデーモンのモニタ）
- /var/cluster/ha/log/crsd_*hostname*（デーモン・ログのリセット）
- /var/cluster/ha/log/diags_*hostname*（シリアルおよびネットワークの接続性のテストなどの診断ツールの出力）
- /var/cluster/ha/log/fs2d_log（クラスタ・データベース・メンバーシップのステータス）
- /var/sysadm/salog（システム管理ログ。GUI によって実行されたコマンドのリストが含まれます）

ディスクがログ・メッセージでいっぱいになる場合は、215 ページの「ログ・ファイル管理」を参照してください。

注意: ログ・ファイルの名前は変更しないでください。名前を変更すると、エラーが発生します。

クラスタのステータス

システムのステータスは、GUI、あるいは cluster_status(1M)、clconf_info、cmgr(1M)、または cxfs_info(1M) コマンドを使用してモニタできます。103 ページの「アイコンと状態の説明」も参照してください。

GUI を使ったクラスタのステータスの確認

クラスタの状態を継続的に監視する簡単な方法は、表示領域を使用して以下を選択することです。

「編集(Edit)」
　>「すべて展開(Expand All)」

クラスタのステータスは、以下のいずれかになります。

- 「アクティブ(ACTIVE)」: クラスタが実行中であることを意味します。

- 「非アクティブ(INACTIVE)」:「CXFS サービスの開始(Start CXFS Services)」タスクが実行されていないことを意味します。
- 「エラー(ERROR)」: 一部のノードが「停止(DOWN)」状態であることを意味します(つまり、クラスタは実行されているべきなのに、実行されていません)。
- 「不明(UNKNOWN)」: クエリーを実行しているノードで CXFS サービスが実行されていないために状態を判別できないことを意味します。詳細については、241 ページの「ノードのステータス」を参照してください。

cluster_status を使ったクラスタのステータスの確認

curses(3X) インタフェースを使用してクラスタをモニタするには、cluster_status コマンドを使用できます。たとえば、以下の例に、1 つのファイルシステムがマウントされており、ヘルプ・テキストが表示されている 3 ノード・クラスタを示します。

```
# /var/cluster/cmgr-scripts/cluster_status
+ Cluster=cxfs6-8  FailSafe=Not Configured CXFS=ACTIVE              15:15:33
   Nodes =   cxfs6    cxfs7    cxfs8
FailSafe =
    CXFS =      UP       UP       UP

CXFS              DevName           MountPoint           MetaServer     Status
        /dev/cxvm/concat0             /concat0                cxfs7         UP

                      +-------+ cluster_status Help +--------+
                      |  on s  - Toggle Sound on event       |
                      |  on r  - Toggle Resource Group View  |
                      |  on c  - Toggle CXFS View            |
                      |     h  - Toggle help screen          |
                      |     i  - View Resource Group detail  |
                      |     q  - Quit cluster_status         |
                      +--- Press 'h' to remove help window --+

------------------------------------------------------------------
cmd('h' for help) >
```

上記の例では、ノードまたはクラスタのステータスが変わると、サウンドがアクティブになります。r コマンドおよび i コマンドは CXFS とは関係がなく、FailSafe だけで使用できます。s 設定は、-m(ミュート)オプションと共に cluster_status を呼出すことによってオーバーライドできます。

次の出力では、CXFS クラスタ内が実行中であり、cxfs7 が /dev/cxvm/concat0 XVM ボリュームのメタデータ・サーバであることを示しています。

```
cxfs6# /var/cluster/cmgr-scripts/cluster_status

+ Cluster=cxfs6-8  FailSafe=Not Configured CXFS=ACTIVE              15:18:28
   Nodes =   cxfs6    cxfs7    cxfs8
FailSafe =
    CXFS =      UP       UP       UP

CXFS              DevName           MountPoint           MetaServer     Status
        /dev/cxvm/concat0             /concat0                cxfs7         UP
```

メモ: cluster_status コマンドを使用して表示できる CXFS ファイルの数は、最大 128 個までです。

clconf_info を使ったクラスタのステータスの確認

クラスタが実行中の場合、/usr/cluster/bin/clconf_info を使用して詳細情報を表示できます。

例は、次のとおりです。

```
cxfs6 # clconf_info
Membership since Thu Mar  1 08:15:39 2001
Node         NodeId     Status    Age   Incarnation     CellId
cxfs6             6         UP      0             0          2
cxfs7             7         UP      0             0          1
cxfs8             8         UP      0             0          0
1 CXFS FileSystems
/dev/cxvm/concat0 on /concat0  enabled  server=(cxfs7)  2 client(s)=(cxfs8,cxfs6)
```

cmgr を使ったクラスタのステータスの確認

ノードおよびクラスタのステータスを照会するには、次の cmgr(1M) コマンドを使用します。

```
cmgr> show status of cluster cluster_name
```

cxfs_info を使ったクラスタ、ノード、またはファイルシステムのステータスの確認

cxfs_info コマンドは、クラスタのステータス、ノードのステータス、およびファイルシステムのステータスに関する情報を提供します。cxfs_info(1M)は、クライアントのみのノードから実行されます。

- IRIX クライアントのみのノード

  ```
  /usr/cluster/bin/cxfs_info
  ```

- Solaris

  ```
  /usr/cxfs_cluster/bin/cxfs_info
  ```

- Windows（コマンド・プロンプトまたはエクスプローラから）

  ```
  \Program Files\CXFS\cxfs_info
  ```

-e オプションを使用すると、新しい情報が得られたときに画面を更新して継続的に情報を表示できます。更新の間に画面をクリアするには、-c オプションを使用します。出力の詳細度を下げるには、-q（表示抑制）オプションを使用します。

たとえば、cxfssun4 という Solaris ノードでは、以下のような出力になります。

```
cxfssun4 # /usr/cxfs_cluster/bin/cxfs_info
cxfs_client status [timestamp Sep 03 12:16:06 / generation 18879]

Cluster:
    sun4 (4) - enabled
Local:
    cxfssun4 (2) - enabled, state: stable, cms: up, xvm: up, fs: up
Nodes:
    cxfs27     enabled  up    1
    cxfs28     enabled  up    0
    cxfsnt4    enabled  up    3
    cxfssun4   enabled  up    2
    mesabi     enabled  DOWN  4
Filesystems:
    lun1s0     enabled  mounted          lun1s0               /lun1s0
    mirror0    disabled unmounted        mirror0              /mirror0
```

ノードのステータス

ノードのステータスを照会するには、そのノードの論理名を指定します。ノードのステータスは、以下のいずれかになります。

- 「実行中(UP)」: CXFS サービスが開始されており、ノードが CXFS カーネル・メンバーシップの一部であることを意味します。詳細については、28 ページの「メンバーシップ定足数」を参照してください。

- 「停止(DOWN)」: CXFS サービスが開始されており、ノードがクラスタの一部として定義されているにもかかわらず、このノードは現在の CXFS カーネル・メンバーシップ内に含まれていないことを意味します。
- 「非アクティブ(INACTIVE)」:「CXFS サービスの開始(Start CXFS Services)」タスクが実行されていないことを意味します。
- 「不明(UNKNOWN)」: クエリーを実行しているノードで CXFS サービスが実行されていないために状態を判別できないことを意味します。

状態の情報は、CXFS サービスが開始されている場合にのみ実行されるデーモンによって交換されます。ほかのノードのステータスを報告するには、指定した CXFS 管理ノードで CXFS サービスが実行されている必要があります。

たとえば、node2 の状態を表示するには、node1 で CXFS サービスが開始されていなければなりません。node1 で CXFS サービスが開始されていれば、クラスタ内のその他のすべてのノードの状態は正確に報告されます。ただし、node1 の CXFS サービスが開始されていない場合は、以下の状態が報告されます。

- このノード自体の状態としては「非アクティブ(INACTIVE)」が報告されます。これは、「CXFS サービスの開始(Start CXFS Services)」タスクが実行されていないと判断できるためです。
- その他のすべてのノードの状態としては「不明(UNKNOWN)」が報告されます。これは、その他のノードと情報を交換するために必要なデーモンが実行されていないので状態を判断できないためです。

以下の節では、ノードのステータスをモニタするためのさまざまな方法を説明します。240 ページの「cxfs_info を使ったクラスタ、ノード、またはファイルシステムのステータスの確認」も参照してください。

GUI を使ったノードのステータスのモニタ

ノードのステータスをモニタするには、表示領域を使用できます。「表示(View)」->「クラスタ内のノード(Nodes in Cluster)」を選択します。

CXFS または FailSafe、あるいはその両方にノードが適用されるかどうかを判断するには、画面でそのノード名をダブルクリックします。243 ページの図9-1 に、タイプ CXFS only のノードの例を示します。

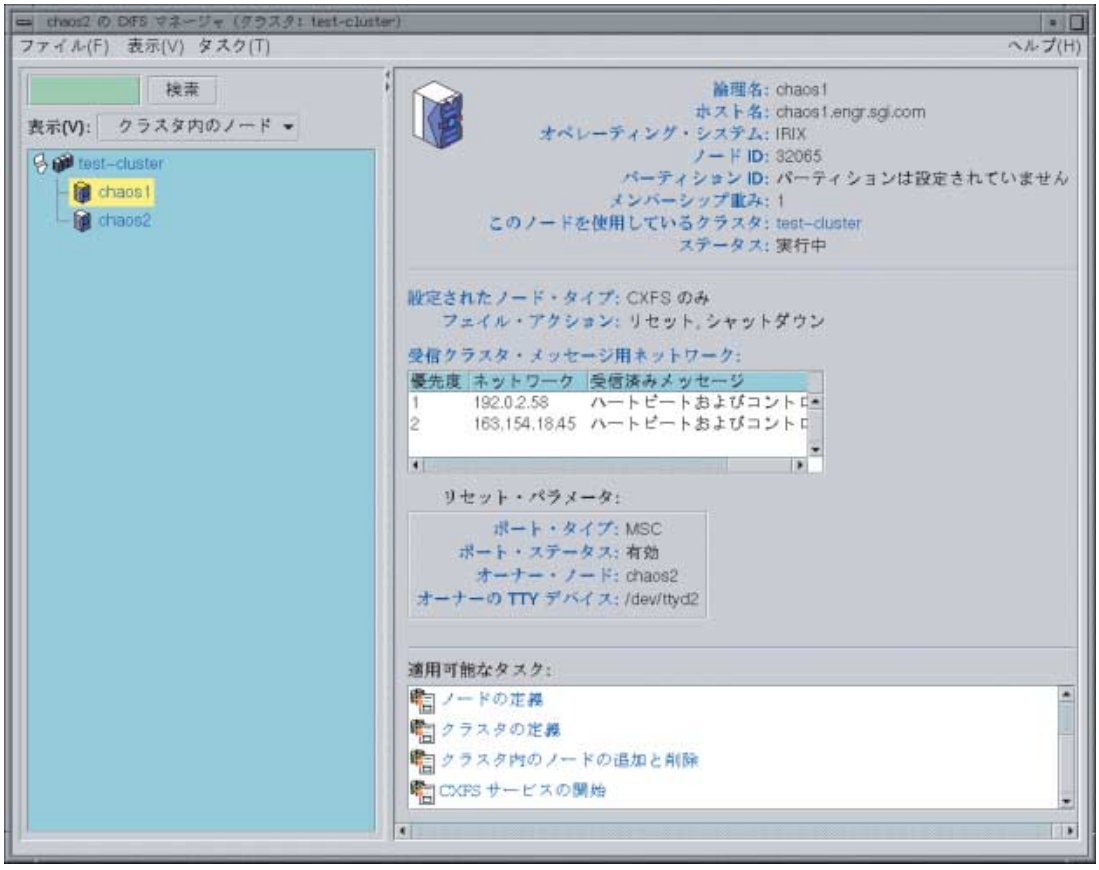


図9-1 ノードのステータス

cmgr を使ったクラスタのステータスの照会

ノードのステータスを照会するには、次の cmgr(1M) コマンドを使用します。

```
cmgr> show status of node node_name
```

cluster_status を使ったノードのステータスのモニタ

クラスタ内のノードのステータスをモニタするには、cluster_status コマンドを使用できます。たとえば、次の出力は、CXFS クラスタ内の 3 つのノードがすべて実行中であることを示しています。

```
cxfs6# /var/cluster/cmgr-scripts/cluster_status

+ Cluster=cxfs6-8  FailSafe=Not Configured CXFS=ACTIVE              15:15:33
   Nodes =   cxfs6    cxfs7    cxfs8
FailSafe =
    CXFS =      UP       UP       UP
```

c コマンドを off に切替えた場合、CXFS 行はなくなります。

cmgr を使ったシステム・コントローラへの ping の実行

CXFS が実行されている場合、次の cmgr(1M) コマンドを使用して、ノードのシステム・コントローラが応答しているかどうかを判断できます。

```
cmgr> admin ping node node_name
```

このコマンドは、CXFS デーモンを使用して、システム・コントローラが応答しているかどうかをテストします。

CXFS デーモンが実行されていない場合でも、admin ping コマンドの standalone オプションを使用すると、クラスタ内のノードのリセット接続性を確認できます。

```
cmgr> admin ping standalone node node_name
```

このコマンドは、ping コマンドを直接呼出して、指定したノードでシステム・コントローラが応答しているかどうかをテストします。

cmgr を使ったシリアル・ハードウェア・リセット回線のモニタ

ノードでシステム・コントローラに対して ping を実行するには、以下のように cmgr(1M) コマンドを使用できます（改行して読みやすくしてあります）。

```
cmgr> admin ping dev_name device_name of dev_type device_type
with sysctrl_type system_controller_type
```

XVM 統計

メモ:この機能を使用するには、pcp_eoe および pcp_eoe.sw.xvm パッケージがインストールされていることが前提となります。43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。

PCP (Performance Co-Pilot) を使用して、XVM 統計をモニタできます。PCP で XVM 統計をモニタするには、以下の統計の収集を有効にする必要があります。

- ローカル・ホストの統計の収集を有効にするには、以下を入力します。

  ```
  $ pmstore xvm.control.stats_on 1
  ```

- ローカル・ホストの統計の収集を無効にするには、以下を入力します。

  ```
  $ pmstore xvm.control.stats_on 0
  ```

XVM 統計は、以下の方法で収集できます。

- pcp_eoe.sw.monitor パッケージの pmval(1) コマンドを使用する。このコマンドは IRIX リリースに付属しており、使用可能なメトリックの PCP 名前空間内の xvm グループから、選択したメトリックの ASCII レポートを生成するために使用できます。
- PCP pcp.sw.monitor パッケージ(市販のオプション製品)に付属するオプションの pmgxvm(1) コマンドを使用する。

  pcp.sw.monitor パッケージを所有している場合は、pmchart(1) コマンドを使用して時系列のデータを動くグラフに表示することもできます。図9-2 に、例を示します。

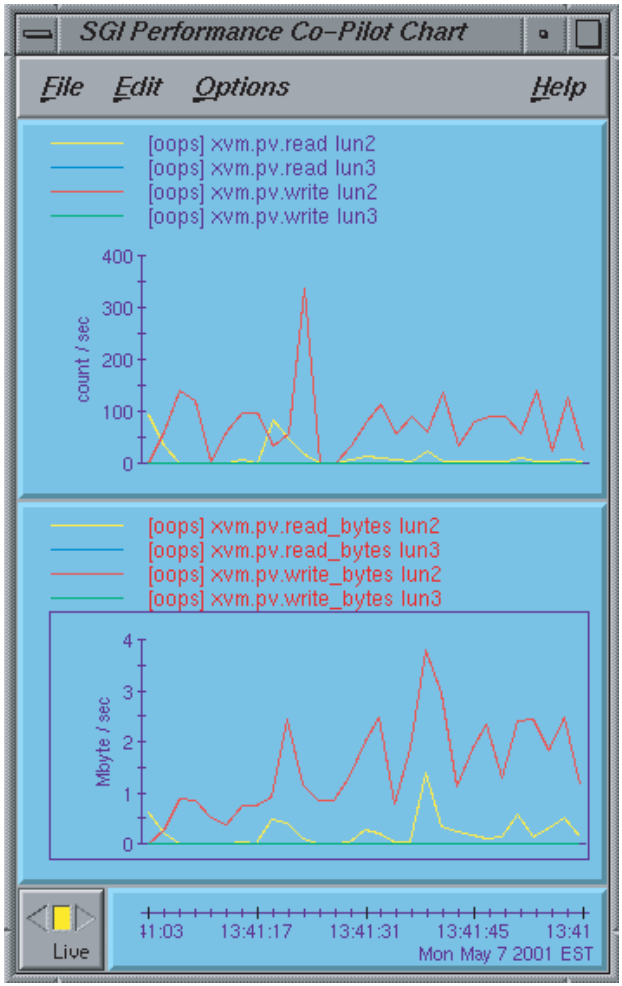


図9-2 pmgxvm グラフ

I/O 制限のステータス

現在の制限のステータスを確認するには、GUI の表示領域で「表示(View)」->「スイッチ(Switches)」を選択するか、`cmgr`(1M) の `admin fence query` コマンドを使用するか、または以下のように `hafence`(1M) を使用します。

```
/usr/cluster/bin/hafence -q
```

たとえば、次の出力は、すべてのノードが有効であることを示しています。

```
# /usr/cluster/bin/hafence -q
  Switch[0] "ptg-brocade" has 8 ports
    Port 1 type=FABRIC status=enabled  hba=210000e08b0102c6 on host thunderbox
    Port 2 type=FABRIC status=enabled  hba=210000e08b01fec5 on host whack
    Port 5 type=FABRIC status=enabled  hba=210000e08b027795 on host thump
    Port 6 type=FABRIC status=enabled  hba=210000e08b019ef0 on host thud
```

制限されているポートは、status=disabledと表示されます。例は、次のとおりです。

```
# /usr/cluster/bin/hafence -q
  Switch[0] "brocade04" has 16 ports
    Port 4 type=FABRIC status=enabled  hba=210000e08b0042d8 on host o200c
    Port 5 type=FABRIC status=enabled  hba=210000e08b00908e on host cxfs30
    Port 9 type=FABRIC status=enabled  hba=2000000173002d3e on host cxfssun3
```

詳細な (-v) 出力は、以下のようになります。

```
# /usr/cluster/bin/hafence -v
  Switch[0] "brocade04" has 16 ports
    Port 0 type=FABRIC status=enabled  hba=2000000173003b5f on host UNKNOWN
    Port 1 type=FABRIC status=enabled  hba=2000000173003adf on host UNKNOWN
    Port 2 type=FABRIC status=enabled  hba=210000e08b023649 on host UNKNOWN
    Port 3 type=FABRIC status=enabled  hba=210000e08b021249 on host UNKNOWN
    Port 4 type=FABRIC status=enabled  hba=210000e08b0042d8 on host o200c
    Port 5 type=FABRIC status=enabled  hba=210000e08b00908e on host cxfs30
    Port 6 type=FABRIC status=enabled  hba=2000000173002d2a on host UNKNOWN
    Port 7 type=FABRIC status=enabled  hba=2000000173003376 on host UNKNOWN
    Port 8 type=FABRIC status=enabled  hba=2000000173002c0b on host UNKNOWN
    Port 9 type=FABRIC status=enabled  hba=2000000173002d3e on host cxfssun3
    Port 10 type=FABRIC status=enabled  hba=2000000173003430 on host UNKNOWN
    Port 11 type=FABRIC status=enabled  hba=200900a0b80c13c9 on host UNKNOWN
    Port 12 type=FABRIC status=disabled hba=0000000000000000 on host UNKNOWN
```

```
Port 13 type=FABRIC status=enabled  hba=200d00a0b80c2476 on host UNKNOWN
Port 14 type=FABRIC status=enabled  hba=1000006069201e5b on host UNKNOWN
Port 15 type=FABRIC status=enabled  hba=1000006069201e5b on host UNKNOWN
```

現在の異常時のアクション設定を確認するには、cmgr の show node *nodename* コマンドを使用するか、以下のように cms_failconf コマンドを使用します。

```
/usr/cluster/bin/cms_failconf -q
```

たとえば、次の出力は、thud 以外のすべてのノードで、システムのデフォルトの異常時のアクション設定が使用されていることを示しています。ノード thud は、制限とリセットが実行されるよう設定されています。

```
# cms_failconf -q
CMS failure configuration:
        cell[0] whack    Reset Shutdown
        cell[1] thunderbox       Reset Shutdown
        cell[2] thud     Fence Reset
        cell[3] thump    Reset Shutdown
        cell[4] terry    Reset Shutdown
        cell[5] leesa    Reset Shutdown
```

トラブルシューティング

CXFS クラスタの設定と管理は複雑なタスクになる場合があります。一般に、ほとんどの問題は、ノードを再起動することによって解決できますが、この章のトピックでは、再起動せずに解決する方法を説明します。

- 「トラブルシューティング方法」
- 261 ページの「問題の回避」
- 270 ページの「一般的な問題」
- 273 ページの「エラー・メッセージの理解」
- 291 ページの「修正処置」

管理タスクは、`cluster_admin.sw.base` ソフトウェア・パッケージがインストールされているノードから実行してください。その他のトラブルシューティング情報については、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

トラブルシューティング方法

CXFS 問題をトラブルシューティングするには、以下を実行します。

- 「ツールについての理解」
- 258 ページの「クラスタのステータスの識別」
- 259 ページの「問題の検出」

ツールについての理解

この節では、CXFS のトラブルシューティングに必要なツールの概要について説明します。

注意: ここでリストされているコマンドの多くはこのマニュアルの対象外であり、簡単に参照できるように紹介されているだけです。これらのコマンドを使用する前に、参照されているその他のガイドやマン・ページで詳細を参照してください。

- 250 ページの「物理ストレージ・ツール」

- 250 ページの「クラスタ設定ツール」
- 251 ページの「クラスタ制御ツール」
- 253 ページの「ネットワーク・ツール」
- 253 ページの「クラスタまたはノードのステータス・ツール」
- 254 ページの「パフォーマンス・モニタ・ツール」
- 255 ページの「カーネルのステータス・ツール」
- 256 ページの「ログ・ファイル」

物理ストレージ・ツール

以下の物理ストレージ・ツールについて理解します。

- ハードウェア・インベントリを表示するには、hinv(1M) コマンドを使用します。

  ```
  # /sbin/hinv
  ```

  出力が予想したものとは異なる場合、scsiha(1M) コマンドを使用してデバイスの検査を行い、SCSI バス・リセットを実行します。

  ```
  # /usr/sbin/scsiha -pr bus_number
  ```

- I/O デバイスを設定するには、ioconfig(1M) コマンドを使用します。

  ```
  # /sbin/ioconfig -f /hw
  ```

- 物理ボリュームを表示するには、xvm(1M) コマンドを使用します。

  ```
  # /sbin/xvm show -v phys/
  ```

  『XVM Volume Manager Administrator's Guide』を参照してください。

クラスタ設定ツール

以下のクラスタ設定ツールについて理解します。

- XVM ボリュームを設定するには、xvm(1M) コマンドを使用します。

  ```
  # /sbin/xvm
  ```

  『XVM Volume Manager Administrator's Guide』を参照してください。

- CXFS ノードおよび CXFS クラスタを設定するには、GUI または cmgr(1M) コマンドのいずれかを使用します。
  - GUI:

    ```
    # /usr/sbin/cxfsmgr
    ```

    94 ページの「GUI の機能」および 89 ページの 第4章「CXFS の GUI タスク・リファレンス」を参照してください。
  - プロンプト表示ありの cmgr(1M) コマンド・ライン:

    ```
    # /var/cluster/bin/cmgr -p
    ```

    142 ページの「cmgr(1M) 概要」および 141 ページの 第5章「CXFS の cmgr タスク・リファレンス」を参照してください。
- データベースを再度初期化するには、cdbreinit コマンドを使用します。

  ```
  # /usr/cluster/bin/cdbreinit
  ```

  295 ページの「クラスタ・データベースの再作成」を参照してください。

クラスタ制御ツール

以下のクラスタ制御ツールについて理解します。

- クラスタ・サービス・デーモンを開始および停止するには、以下のコマンドを使用します。

  ```
  # /etc/init.d/cluster start
  # /etc/init.d/cluster stop
  ```

  これらのコマンドは、ファイルシステムが使用可能でもクラスタのステータスからはそれが判断できなかったり、クラスタ定足数が失われた場合に便利です。

  以下を参照してください。
  - 263 ページの「クラスタ・データベース・メンバーシップ定足数の安定性」
  - 291 ページの「CXFS サービスの再開」
  - 291 ページの「クラスタ・データベースのクリア」
  - 295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」

- CXFS サービスを開始および停止するには、GUI または以下の cmgr(1M) コマンドを使用します。

  ```
  cmgr> start cx_services on node hostname for cluster clustername
  cmgr> stop cx_services on node hostname for cluster clustername
  ```

  メタデータ・サーバでは、このコマンドを実行すると、メタデータ・サーバとして使用可能なほかのノードによってファイルシステムが回復されてしまいます。172 ページの「cmgr を使ったクラスタ・サービス・タスク」および 125 ページの「GUI を使ったクラスタ・サービス・タスク」を参照してください。

  メモ: このリリースでは、再設定の機能はデフォルトで無効になっており、回復の機能はスタンバイ・ノードを使用している場合のみサポートされています。

  再設定および回復の機能は完全に実装されていますが、関連する問題が多数存在するため、現在のリリースでは完全にサポートされていません。これらの機能を使用してもデータの整合性が損なわれることはありませんが、クラスタ・ノードのパニックやハングが発生する可能性があります。再設定および回復の機能は、このような問題が解決された時点で、今後のリリースで完全にサポートされる予定です。

- ローカル・ノードのメタデータ・サーバを強制的に回復して、そのローカル・ノードの CXFS カーネル・メンバーシップを許可および取消すには、GUI または以下の cmgr(1M) コマンドを使用します。

  ```
  cmgr> admin cxfs_start
  cmgr> admin cxfs_stop
  ```

  回復が完了するまでは、以降の admin cxfs_start を発行しないでください。ローカル・ノードは、回復が完了するまで CXFS カーネル・メンバーシップに再設定できません。

  以下を参照してください。

  - 128 ページの「GUI を使ったローカル・ノードのメンバーシップ取消し」
  - 128 ページの「GUI を使ったローカル・ノードのメンバーシップ許可」
  - 176 ページの「cmgr を使ったローカル・ノードのメンバーシップ取消し」
  - 177 ページの「cmgr を使ったローカル・ノードのメンバーシップ許可」

ネットワーク・ツール

以下のネットワーク・ツールについて理解します。

- ネットワーク・ホストにパケットを送信するには、ping(1) コマンドを使用します。

  # **/usr/etc/ping**

- ネットワークのステータスを表示するには、netstat(1) コマンドを使用します。

  # **/usr/etc/netstat**

クラスタまたはノードのステータス・ツール

以下のクラスタまたはノードのステータス・ツールについて理解します。

- 実行されているクラスタ・デーモンを表示するには、ps(1) コマンドを使用します。

  # **/sbin/ps -ef | grep cluster**

  71 ページの「クラスタ・デーモンが実行されていることの確認」を参照してください。

- クラスタおよびファイルシステムのステータスを表示するには、以下のいずれかを使用します。
  - GUI:

    # **/usr/sbin/cxfsmgr**

    124 ページの「GUI を使ったクラスタの表示」を参照してください。

  - cluster_status(1M) コマンド:

    # **/var/cluster/cmgr-scripts/cluster_status**

    239 ページの「cluster_status を使ったクラスタのステータスの確認」を参照してください。

  - clconf_info コマンド:

    # **/var/cluster/bin/clconf_info**

  - cxfs_info(1M) コマンド (クライアントのみのノード):
    - IRIX:

      irix# **/usr/cluster/bin/cxfs_info**

    - Solaris:

      solaris# **/usr/cxfs_cluster/bin/cxfs_info**

- Windows:

  C: **\Program Files\CXFS\cxfs_info**

- マウントされているファイルシステムを表示するには、mount(1M) コマンドまたは df(1) コマンドを使用します。

  # **/sbin/mount**
  # **/usr/sbin/df**

- 空きディスク・ブロックの数を報告するには、df(1) コマンドを使用します。

  # **/usr/sbin/df**

- ボリュームを表示するには、xvm(1M) コマンドを使用します。

  # **/sbin/xvm show vol/**

  『XVM Volume Manager Administrator's Guide』を参照してください。

パフォーマンス・モニタ・ツール

以下のパフォーマンス・モニタ・ツールについて理解します。

- システム・アクティビティをモニタするには、sar(1) コマンドを使用します。

  # **/usr/bin/sar**

- ファイルシステム・バッファ・キャッシュ・アクティビティをモニタするには、bufview(1) コマンドを使用します。

  # **/usr/sbin/bufview**

  メモ: bufview は、使用中のノードではインタラクティブに使用せずに、バッチ・モードで実行してください。

- オペレーティング・システム・アクティビティ・データをモニタするには、osview(1) コマンドを使用します。

  # **/usr/sbin/osview**

- XVM ボリュームの統計をモニタするには、xvm(1M) コマンドを使用します。

  # **/sbin/xvm change stat on** {*concatname*|*stripename*|*physname*}

  『XVM Volume Manager Administrator's Guide』を参照してください。

- システム・パフォーマンスをモニタするには、PCP (Performance Co-Pilot) を使用します。『Performance Co-Pilot User's and Administrator's Guide』、『Performance Co-Pilot Programmer's Guide』、および pmie(1) と pmieconf(1) のマン・ページを参照してください。

カーネルのステータス・ツール

以下のカーネルのステータス・ツールについて理解します (これについては、日本 SGI の営業担当員の支援が必要な場合があります)。

- カーネルのステータスを判断するには、icrash(1M) コマンドを使用します。

  ```
  # /usr/bin/icrash
  ```

  - cfs: CXFS コマンドのリストを表示します。
  - dcvn: クライアント v ノードのリストを表示します。
  - dsvn: サーバ v ノードのリストを表示します。
  - mesglist: メッセージを受信者までトレースします。
  - sinfo: クライアント/サーバおよびファイルシステムを表示します。
  - sthread | grep cmsd: CXFSカーネル・メンバーシップの状態を判断します。出力には、以下が表示される場合があります。
    - cms_dead(): ノードがダウンしていることを示します。
    - cms_follower(): ほかのノードによって CXFS カーネル・メンバーシップ (リーダー) が作成されるまでノードが待機していることを示します。
    - cms_leader(): ノードが CXFS カーネル・メンバーシップ作成をリードしていることを示します。
    - cms_declare_membership(): ノードは CXFS カーネル・メンバーシップを宣言する準備はできているが、リセットを待っていることを示します。
    - cms_nascent(): ノードが開始されてからクラスタに設定されていないことを示します。
    - cms_shutdown(): ノードがシャットダウンされており、CXFS カーネル・メンバーシップ内にないことを示します。
    - cms_stable(): CXFS カーネル・メンバーシップが形成されており、安定していることを示します。
  - tcp_channels: ほかのノードとの接続のステータスを判断します。

  - `-t -a -w` *filename*: CXFS に対してトレースします。
  - `-t` *cms_thread*: 上記のいずれかのスレッドをトレースします。
- 役立つデバッグ情報を提供する内部カーネル・ルーチンを呼出すには、`idbg`(1M) コマンドを使用します。

  # **/usr/sbin/idbg**

ログ・ファイル

以下のログ・ファイルについて理解します。

- `/var/adm/SYSLOG`(`Membership delivered` を検索)
- `/var/cluster/ha/log/cad_log`
- `/var/cluster/ha/log/clconfd_`*hostname*
- `/var/cluster/ha/log/cli_`*hostname*
- `/var/cluster/ha/log/cmond_log`
- `/var/cluster/ha/log/crsd_`*hostname*
- `/var/cluster/ha/log/diags_`*hostname*
- `/var/cluster/ha/log/fs2d_log`
- `/var/sysadm/salog`

237 ページの「ログ・ファイル」を参照してください。

cxfsdump を使用したクラスタ設定の収集

SGI に問題を報告する前に、`cxfsdump`(1M) コマンドを使用して、ネットワーク・インタフェース、CXFS の登録情報、I/O、クラスタ・データベースの内容など、CXFS クラスタに関する設定情報を収集してください。これにより、SGI サポートは問題をより迅速に解決できます。

メモ: `cxfsdump` コマンドでは、`rcp`(1M) および `rsh`(1M) コマンドを通じてクラスタ内の IRIX および Solaris ノード全体にアクセスできる必要があります。Windows ノードではこれらのコマンドが提供されていないため、各 Windows ノードで `cxfsdump` コマンドを手動で実行してください。

`cxfsdump` は、クラスタ内の CXFS 管理ノードから実行する必要があります。

/usr/cluster/bin/cxfsdump

IRIX および Solaris システムに対する出力は、cxfsdump コマンドを実行した CXFS 管理ノードの /var/cluster/cxfsdump-data ディレクトリ内のファイルに配置されます。完了すると、cxfsdump コマンドによってファイルの名前と場所が報告されます。

クラスタに Windows ノードが含まれる場合は、各 Windows ノードでコマンドを手動で実行してください。

例は、次のとおりです。

```
adminnode# cxfsdump

No cluster name supplied!! Detecting cluster configuration

 Executing CXFSDUMP on CLUSTER testcluster NODE o200a
Gathering cluster information...
Determining OS level......
Getting versions info....
Obtaining CXFS database...
Checking for tie-breakers etc...
Obtaining harware inventory...
Grabbing /etc/hosts.....
Grabbing /etc/resolv.conf...
Grabbing /ets/nsswitch.conf...
Obtaining phyVol information using XVM...
ioctl() to xvm api node failed: Invalid argument
Could not get xvm subsystem info:  xvmlib_execute_ioctl: system call failed.
Obtaining Volume topology information using XVM...
ioctl() to xvm api node failed: Invalid argument
Could not get xvm subsystem info:  xvmlib_execute_ioctl: system call failed.
Copying failover configuration and scsifo paths ...
Gathering network information...
Checking for any installed Patches..
Monitoring file system buffer cache for 3 minutes...
Running Systune ...
Obtaining modified system tunable parameters...
Creating ICRASH CMD file...
Executing ICRASH commands...
Copying CXFS logs...
Copying /var/cluster/ha/log/cad_log...
Copying /var/cluster/ha/log/clconfd_o200a...
Copying /var/cluster/ha/log/cli_o200a...
Copying /var/cluster/ha/log/cmond_log...
Copying /var/cluster/ha/log/crsd_o200a...
Copying /var/cluster/ha/log/fs2d_log...
```

```
Copying /var/cluster/ha/log/fs2d_log.old...
Copying SYSLOG...
Distributing /usr/cluster/bin/cxfsdump.pl to node o200c ...
Distributing /usr/cluster/bin/cxfsdump.pl to node o200b ...
Creating the output directory : /var/cluster/cxfsdump-data
Gathering node information for the cluster testcluster ...
Running RSH to node o200c...
Running RSH to node o200b...
Waiting for other cluster nodes to gather data...
FINAL CXFSDUMP OUTPUT IN /var/cluster/cxfsdump-data/testcluster_cxfsdump20020903.tar.gz
```

Windows システムでは、cxfsdump により、passwd ファイルが保持されているディレクトリと同じディレクトリ内に CXFSDump という名前のディレクトリが作成されます。完了すると、cxfsdump コマンドによって、データが格納された場所が報告されます。例は、次のとおりです。

```
FINAL CXFSDUMP output in output_filename
```

クラスタのステータスの識別

問題が発生した場合、以下の質問に回答することによって、クラスタのステータスを識別します。

- クラスタ・デーモンが実行されているかどうか。71 ページの「クラスタ・デーモンが実行されていることの確認」を参照してください。
- クラスタの状態が各ノードで一致しているか。各 CXFS 管理ノードで clconf_info コマンドを実行して、結果を比較します。
- どのノードが CXFS カーネル・メンバーシップ内にあるか。239 ページの「cluster_status を使ったクラスタのステータスの確認」、240 ページの「cmgr を使ったクラスタのステータスの確認」、および/var/adm/SYSLOG ファイルを参照してください。
- どのノードがクラスタ・データベース (fs2d) メンバーシップ内にあるか。各 CXFS 管理ノードの /var/cluster/ha/log/fs2d_log ファイルを参照してください。
- すべてのノードでデータベースが一致しているか。各 CXFS 管理ノードへのこのログインを判別し、/var/cluster/ha/log/fs2d_log ファイルおよびデータベース・チェックサムを調査します。
- さまざまな CXFS クライアント・ノードにログオンするか、詳細の表示されている GUI の表示領域を使用して、以下に回答します。
  - すべてのノードでデバイスが使用可能であるか。以下を使用します。
    - xvm(1M) コマンド: 物理ボリュームを表示します。

      ```
      xvm:cluster> show -v phys/
      ```

- ls(1) コマンド: /dev/cxvm ディレクトリの内容をリストします。

  # **ls /dev/cxvm**

- hinv(1M) コマンド: ハードウェア・インベントリを表示します。

  # **hinv**

– すべてのノードにファイルシステムがマウントされているか。mount(1M) コマンドおよび clconf_info コマンドを使用します。

– どのノードが各ファイルシステムのメタデータ・サーバであるか。Use the cluster_status(1M) or clconf_info commands.

メタデータ・サーバでは、clconf_info コマンドを使用します。

- メタデータ・サーバが回復中であるか。icrash(1M) コマンドを使用してメッセージを検索し、/var/adm/SYSLOG をチェックします。255 ページの「カーネルのステータス・ツール」を参照してください。以下のようなメッセージは、回復ステータスを示します。

  – 回復が進行中の場合

```
Mar 13 11:31:02 1A:p2 unix: ALERT: CXFS Recovery: Cell 1: Client Cell 0 Died, Recovering </scratch/p9/local>
```

  – 回復が完了している場合

```
Mar 13 11:31:04 5A:p2 unix: NOTICE: Signaling end of recovery cell 1
```

- (20 秒を超える)長時間実行されているカーネル・メッセージがあるか。この状況を調べるには、icrash mesglist コマンドを使用します。例は、次のとおりです。

```
>> mesglist
Cell:7
THREAD ADDR          MSG ID TYPE CELL MESSAGE                              Time(Secs)
================== ======= ==== ==== ================================ ==========
0xa8000000d60a4800  5db537  Rcv    0                    I_dcvn_recall           0
0xa8000000d60a4800  5db541  Snt    0                 I_dsvn_notfound            0
0xa80000188fc51800  3b9b4f  Snt    0              I_dsxvn_inode_update  17:48:58
```

- ファイルシステムがマウントされていない場合、XVM にオンラインで表示されるか。次の xvm(1M) コマンドを使用できます。

  xvm:cluster> **show vol/***

問題の検出

問題を見つけるには、以下を実行します。

- ログ・ファイルを調べます(256 ページの「ログ・ファイル」を参照してください)。
  - すべてのログ・ファイル内でエラーを検索します。237 ページの「ログ・ファイル」を参照してください。問題の期間内のメッセージをすべて調べます。
  - エラーの原因を突止めます。エラーを引起こしたイベントを検索してみます。
- icrash コマンドを使用します。255 ページの「カーネルのステータス・ツール」を参照してください。
- GUI の表示領域の詳細情報を使用して、具体的な設定情報を絞込みます。
- GUI で「接続性のテスト(Test Connectivity)」タスクを実行します。121 ページの「GUI を使ったノードの接続性のテスト」を参照してください。
- 各 CXFS 管理ノードで次のコマンドを入力して、クラスタのノードで現在の CXFS カーネル・メンバーシップがどのように表示されているかを判別します。

  ```
  # /usr/cluster/bin/clconf_info
  ```

  このコマンドを実行すると、以下のフィールドが表示されます。
  - Node name(ノード名)
  - Node ID(ノード ID)
  - Status(ステータス。UP または DOWN)
  - Age(このフィールドは有益ではないので無視してください)
  - Incarnation(このフィールドは有益ではないので無視してください)
  - Cell ID(セル ID)。クラスタにノードを追加するときに CXFS ソフトウェアによって動的に割当てられる番号です(ユーザがセル ID 番号を定義することはありません)。セル ID を表示するには、clconf_info コマンドを使用します。

  例は、次のとおりです。

  ```
  # /usr/cluster/bin/clconf_info
  Membership since Fri Sep 10 08:57:36 1999
  Node         NodeId     Status    Age   Incarnation     CellId
  cxfs6          1001         UP      1             7          0
  cxfs7          1002         UP      0             0          1
  cxfs8          1003         UP      0             0          2
  2 CXFS FileSystems
  /dev/xvm/test1 on /mnts/test1  disabled  server   0 clients
  /dev/xvm/test2 on /mnts/test2  disabled  server   0 clients
  ```

- 各 CXFS 管理ノードで /var/adm/SYSLOG をチェックし、CXFS ファイルシステムが正しくマウントまたはアンマウントされていることを確認します。マウントまたはアンマウントできなかった場合、エラーがログに記録され、短い遅れの後で操作が再試行されます。
- アクティブなディスクを表示するには、sar(1) システム・アクティビティ・レポータを使用します。たとえば、以下の例では、アクティブなディスクを表示してそのディスク名を行の最後に配置し、10 秒の間毎秒ポーリングします。

  ```
  # sar -DF 1 10
  ```

  詳細については、sar(1) のマン・ページを参照してください。
- 使用中のバッファを表示するには、bufview(1) ファイルシステム・バッファ・キャッシュ・アクティビティ・モニタを使用します。bufview 内では、help サブコマンドを使用して、使用可能なサブコマンドを参照できます。たとえば、指定したフラグを持つものだけが表示されるように制限する f サブコマンドなどです。以下に、使用中のバッファを表示する場合の例を示します。

  ```
  # bufview
  f
  Buffer flags to display bsy
  ```

  詳細については、bufview(1) のマン・ページを参照してください。
- icrash(1M) IRIX システム・クラッシュ解析ユーティリティを使用します。詳細については、icrash(1M) のマン・ページを参照してください。
- クラスタ・データベースのダンプを取得します。このようなダンプは、次のコマンドを使用して抽出できます。

  ```
  # /usr/cluster/bin/cdbutil -c 'gettree #' > dumpfile
  ```

問題の回避

この節では、以下の内容について説明します。

- 262 ページの「適切な開始」
- 262 ページの「残っているクラスタの削除」
- 263 ページの「クラスタ・データベース・メンバーシップ定足数の安定性」
- 263 ページの「設定の統一性」
- 263 ページの「ノードの機能の適切な定義」
- 263 ページの「GUI の使用」
- 264 ページの「ログ・ファイル名とサイズ」

- 264 ページの「Netscape と Brocade スイッチの GUI」
- 264 ページの「未書込みエクステントの追跡と排他書込みトークンに関するパフォーマンスの問題」
- 265 ページの「crontab ファイルによる過剰なファイルシステム・アクティビティの回避」
- 266 ページの「システム容量の効果的な使用」
- 266 ページの「ノード ID またはクラスタ ID を変更する前の再起動」
- 267 ページの「使用されていないノードの削除」
- 267 ページの「強制シャットダウン後の CXFS の再開」
- 267 ページの「シリアル・ハードウェア・リセット回線の取外し」
- 268 ページの「xfs_repair の適切な使用」

適切な開始

クラスタを設定する前に、必ず 70 ページの「準備手順」の手順に従います。

残っているクラスタの削除

別の新しいクラスタの設定を始める前に、以前のクラスタのノードが CXFS メンバーシップに存在しないことを確認します。以下を入力します。

```
# icrash -e 'sthread | grep cmsd'
```

出力に cmsd カーネル・スレッドが表示される場合は、以下のコマンドを入力して強制的に CXFS をシャットダウンします。

```
# /usr/cluster/bin/cmgr -p
cmgr> admin cxfs_stop
```

その後、cmsd カーネル・スレッドをもう一度チェックします。

```
# icrash -e 'sthread | grep cmsd'
```

数秒待ってもまだ cmsd カーネルが存在している場合は、マシンを再起動するか、新しいクラスタ定義から除外してください。この状態では、マシンを新しいクラスタに設定することはできません。また、クラスタのその他の部分が新しい CXFS メンバーシップを形成できなくなる可能性があります。

クラスタ・データベース・メンバーシップ定足数の安定性

設定プロセス中は、クラスタ・データベース・メンバーシップ定足数が常に安定した状態になければなりません。可能であれば、設定タスクの実行中は、複数のウィンドウを使用して各 CXFS 管理ノードの `fs2d_log` ファイルを表示してください。以下を入力します。

```
# tail -f /var/cluster/ha/log/fs2d_log
```

新しい定足数が出力されたら、メンバー・カウントをチェックします。通常の状況では、ノードを追加または削除するとメッセージが 2〜3 出力されますが、新しい定足数の適用後数秒以内に停止します。

十分な数のマシンが応答していない場合、定足数はなくなります。この場合、データベースは伝えられません。

クラスタ・データベース・メンバーシップ定足数の問題を検出した場合は、データベースにほかの変更を加える前にその問題を解決してください。正しいクラスタ・データベース・メンバーシップ定足数がないノード、またはすべてのノードで同時にクラスタ・インフラストラクチャ・デーモンを再開します。以下を入力します。

```
# /etc/init.d/cluster stop
# /etc/init.d/cluster start
```

クラスタ・データベース・メンバーシップ定足数の問題を報告するときは、`fs2d` ログ・ファイルを提出してください。

設定の統一性

プール全体のノードの設定ファイルと、ネットワークの設定時には、すべて同じ設定を使用する必要があります。つまり、同じ名前を同じ順序で使用してください。51 ページの「IRIX システム・ファイルの設定」を参照してください。

ノードの機能の適切な定義

ノードの機能の適切な定義を使用します。

- 安定性のために、サーバとして使用可能なノードを奇数個と、CXFS 管理ノードを奇数個使用する
- 不安定なノードは CXFS クライアントのみのノードにする

GUI の使用

GUI を使用すると、クラスタとそのコンポーネントを表示領域で簡単に表示できます。表示領域を使用して実行状況を確認し、ノードを追加または削除するのが早すぎることがないようにしてください。ノードを

定義した後は、別のノードを追加する前に、定義したノードが表示領域に表示されるまで待つ必要があります。クラスタを定義した後は、ノードを追加する前に、定義したクラスタが表示されるまで待ってください。変更を加えるのが早すぎると、エラーが発生する可能性があります。

詳細については、90 ページの「GUI の起動」を参照してください。

IRIX で GUI を実行している場合は、GUI アクションが行われているときに別の IRIX デスクトップに移動しないでください。移動すると、GUI がクラッシュする可能性があります。

ログ・ファイル名とサイズ

ログ・ファイルの名前は変更しないでください。ログ・ファイルの名前を変更すると、エラーが発生します。

ログ・ファイルは定期的にローテーションし、ディスクがいっぱいにならないようにします。215 ページの「ログ・ファイル管理」を参照してください。ディスク容量に関する問題が発生した場合は、冗長性の低いログ・レベルを選択することをお薦めします。127 ページの「GUI を使ったログ・グループの設定」または 175 ページの「cmgr を使ったログ・グループの設定」を参照してください。

Netscape と Brocade スイッチの GUI

IRIX ノード上の Netscape を通じて Brocade Web Tools V2.0 にアクセスする場合、Netscape を開始する前にまず以下のいずれかを入力する必要があります。

- sh(1) シェルまたは ksh(1) シェルの場合

  ```
  $ NOJIT=1; export NOJIT
  ```

- csh(1) シェルの場合

  ```
  % setenv NOJIT 1
  ```

これが実行されていないと、Netscape はコア・ダンプを生成してクラッシュします。

未書込みエクステントの追跡と排他書込みトークンに関するパフォーマンスの問題

この節では、未書込みエクステントの追跡と排他書込みトークンに関するパフォーマンスの問題について説明します。

未書込みエクステントの追跡

ファイルシステムを定義するとき、未書込みエクステントの追跡をオン (unwritten=1) またはオフ (unwritten=0) にするかを指定できます。デフォルトではオンになります。

ほとんどの場合は、未書込みエクステントの追跡を使用してもパフォーマンスに影響が出ることはないので、セキュリティを高めるためにデフォルトを使用することをお薦めします。

ただし、以下の両方に該当する場合は、書き込まれていないエクステントの追跡を使用すると、パフォーマンスに影響が出る可能性があります。

- ファイルが事前に割当てられている場合
- これらの事前割当て済みエクステントが初めて 4 MB 未満のレコードで書込まれる場合

これらの両方の状況に該当するときに CXFS で最適なパフォーマンスを得るには、`unwritten=0`(オフ)を使用してファイルシステムをビルドしなければならない場合があります。

メモ: `unwritten=0` の使用に関しては、セキュリティ上の問題があります。詳細については、『IRIX Admin: Disks and Filesystems』を参照してください。

排他書込みトークン

適切なパフォーマンスを得るために、CXFS では排他書込みトークンを取得しないことをお薦めします。このため、以下のガイドラインを使用してください。

- ファイルを事前に割当てます。
- ファイルのサイズを最大サイズに設定し、切詰めなどによってサイズが変更されないようにします。
- ファイルには追加しないでください (つまり、`O_APPEND` は開かれているときに真ではありません)。
- `written` としてエクステントをマークしないでください。
- アプリケーションに連続 `preallocation` コールの実行を許可しないでください。

これらのガイドラインに従ってもパフォーマンスの問題が解決されない場合は、MPI ジョブの実行前後と実行中に `icrash stat` コマンドを実行すると、役立つ情報を見つけることができます。詳細については、`icrash`(1M) のマン・ページを参照してください。

crontab ファイルによる過剰なファイルシステム・アクティビティの回避

デフォルトの `root crontab` ファイルには、以下のエントリが含まれます(改行により読みやすくしてあります)。

```
0 5 * * *  find / -local -type f '(' -name core -o -name dead.letter ')' -atime +7
-mtime +7 -exec rm -f '{}' ';'

 0 3 * * 0 if test -x /usr/etc/fsr; then (cd /usr/tmp; /usr/etc/fsr) fi
```

最初のエントリは、過去 7 日間アクセスされていない core または dead.letter という名前のファイルをすべて検索して削除する find(1) コマンドを実行します。

2 番目のエントリは、マウントされているファイルシステムの編成を向上させる fsr(1M) コマンドを実行します。

find コマンドは、すべてのローカル・ファイルシステムで毎晩実行されます。CXFS ファイルシステムはクラスタ内のすべてのノードでローカルと見なされるので、ノードが同じファイルシステムに同時にアクセスしようとすると、ファイルシステム・アクティビティが過剰に生成される場合があります。このため、以下の順序を使用して、1 つを除くすべての CXFS 管理ノードで find crontab エントリを無効にするか、または変更します。

1. root としてログインします。
2. vi などの好みのエディタを定義します。

   # **setenv EDITOR vi**

3. crontab ファイルを編集します。

   # **crontab -e**

4. find 行をコメント・アウトまたは削除します。

fsr コマンドは、メタデータ・サーバでのみ実行できるため、CXFS クライアントの crontab ファイル内に残しておいても害はありませんが、実行されません。

システム容量の効果的な使用

メタデータ・サーバと CXFS クライアントの接続が切断されないようにするには、メタデータ・サーバや、クラスタ内のノードに接続されているプライベート・サーバを購読しすぎないでください。また、不要なメタデータ・トラフィックを避けます。

空きメモリの量が不十分な場合、ノードでハートビートに遅れが発生することがあり、その結果、ノードは CXFS メンバーシップから除外されます。システムでのメモリの空き容量を監視するには、osview(1) ツールを使用します。

276 ページの「論理スワップ・スペース不足」も参照してください。

ノード ID またはクラスタ ID を変更する前の再起動

ノード ID またはクラスタ ID を再定義する場合は、まず再起動しなければなりません。ここでの問題は、カーネルがまだ古い値を持っているために、CXFS メンバーシップの形成が妨げられる点です。ただし、最初に再起動すると元の値がクリアされるので、ノード ID またはクラスタ ID を再定義できます。

したがって、ノードで cdbreinit を使用してクラスタ・データベースを再作成する場合は、ノード ID またはクラスタ ID を変更する前に再起動しなければなりません。295 ページの「クラスタ・データベースの再作成」を参照してください。

使用されていないノードの削除

しばらくの間ノードを停止する場合は、クラスタおよびプールからそのノードを削除して、クラスタ・データベース・メンバーシップおよび CXFS メンバーシップの定足数の問題を回避します。以下の節を参照してください。

- 123 ページの「GUI を使ったクラスタ定義の変更」
- 168 ページの「cmgr を使ったクラスタの変更」
- 162 ページの「cmgr を使ったノードの削除」

強制シャットダウン後の CXFS の再開

強制シャットダウンをノードで実行した場合は、そのノードをクラスタに戻す前にノードで CXFS を再開する必要があります。ノードがクラスタ内にあってアクティブなことがまだデータベースに示されているときにこれを行うと、CXFS メンバーシップ・デーモンがノードによって再開されます。したがって、この操作は、データベースをリセットした後や、CXFS サービスを停止した後に行うことができます。

たとえば、開始するノードに以下を入力します。

```
# /usr/cluster/bin/cmgr -p
cmgr> stop cx_services on node localnode
cmgr> admin cxfs_start
```

213 ページの「強制 CXFS シャットダウン: ローカル・ノードのメンバーシップ取消し」も参照してください。

シリアル・ハードウェア・リセット回線の取外し

シリアル・ハードウェア・リセットが有効な場合、CXFS がメタデータ・サーバを移動するには reset successful メッセージが必要です。したがって、シリアル・ハードウェア・リセット機能が有効なときに何らかの理由でリセット回線を取外さなければならない場合は、リセット機能を無効にする必要があります。118 ページの「GUI を使ったノード定義の変更」または 156 ページの「cmgr を使ったノードの変更」を参照してください。

メモ: データの整合性を保証するため、サーバとして使用可能なノードが 2 つだけのクラスタでは、リセット機能の使用が必須です。また、サーバとして使用可能なすべてのノードに対しては、リセット機能の使用を強くお薦めします。より大きなクラスタでは、サーバとして使用可能なノードと CXFS 管理ノードを奇数個使用することをお薦めします。8 ページの「クラスタ環境」を参照してください。

xfs_repair の適切な使用

CXFS ファイルシステムは、実際にはクラスタ化された XFS ファイルシステムです。したがって、ファイルシステムが破壊された場合は、`xfs_check`(1M) および `xfs_repair`(1M) コマンドを使用できます。ただし、`xfs_repair`を実行する前に、まずデータが実際に破壊されていることを確認し、XFS ログを再生して重要なメタデータ情報が保持されるようにする必要があります。

注意: 最初に XFS ログを再生せずに `xfs_repair` を実行すると、データの破壊を招く可能性があります。

`xfs_repair` は、ファイルシステムが実際に破壊されている場合にのみ実行してください。ファイルシステムの強制シャットダウンを示すメッセージが表示されても、必ずしも `xfs_repair` の実行が必要なわけではありません。以下に、XFS ファイルシステムの破壊を示しているメッセージの例を示します。

```
XFS read error in file system meta-data block 106412416
```

ファイルシステムが強制的にシャットダウンされた場合、ログは空ではなく、重要なメタデータが含まれています。ファイルシステムをマウントして、このログを再生する必要があります。ログが空になるのは、ファイルシステムが正常にアンマウントされた場合(つまり、強制シャットダウンやクラッシュ以外の場合)だけです。以下のコマンド・ラインを使用して、ログ・ファイルに記録されるトランザクションの例を表示できます。

```
# xfs_logprint -t device
```

ファイルシステムをマウントする前に `xfs_repair` を実行すると、`xfs_repair` によって重要なメタデータがすべて削除されてしまいます。

実際にファイルシステムが破壊されていると思われる場合は、以下の手順に従ってください。

1. デバイスをマウントしてログを再生します。

   ```
   # mount device  any_mount_point
   ```

2. ファイルシステムをアンマウントします。

   ```
   # unmount device
   ```

3. ファイルシステムをチェックします。

   # **xfs_check *device***

4. xfs_repair を変更なしモードで使用して、実行可能な修復を表示します。

   # **xfs_repair -n *device***

5. 修復内容が適切であることが確かな場合は、修復を完了します。

   # **xfs_repair *device***

詳細については、『IRIX Admin: Disks and Filesystems』を参照してください。

一般的な問題

以下は、一般的な問題とその解決方法です。

ファイルシステムにアクセスできない

ファイルシステムにアクセスできない場合は、以下をチェックします。

- ファイルシステムが有効になっているか。GUI、`clconf_info` コマンド、および `cluster_status` コマンドをチェックします。
- マウント・エラーが発生したか。

GUI が実行されない

GUI が実行されない場合は、以下をチェックします。

- ライセンスが正しくインストールされているか。以下を参照してください。
  - 70 ページの「ライセンスの確認」
  - 277 ページの「ライセンス・エラー」
  - 43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」
- クラスタ・デーモンが実行されているかどうか。71 ページの「クラスタ・デーモンが実行されていることの確認」を参照してください。
- `tcpmux` サービスおよび `tcpmux/sgi_sysadm` サービスが `/etc/inetd.conf` ファイル内で有効になっているかどうか。
- `inetd` ラッパまたは `tcp` ラッパが干渉しているかどうか。これは、`connection refused` メッセージまたは `login failed` メッセージによって示される場合があります。
- CXFS 管理ノードに接続しているかどうか。`cxfsmgr`(1) コマンドは、CXFS 管理ノードでのみ実行できます。GUI を CXFS 管理ノードに接続している場合は、Web を通じてほかのシステムから GUI を実行できます。

ログ・ファイルがディスク容量を消費しすぎる

ログ・ファイルがディスク容量を消費しすぎる場合は、ログ・ファイルをローテーションします。215 ページの「ログ・ファイル管理」を参照してください。また、冗長性の低いログ・レベルを選択することもできます。以下を参照してください。

- 54 ページの「CXFS 管理ノードの /etc/config/cad.options」
- 55 ページの「CXFS 管理ノードの /etc/config/fs2d.options」
- 127 ページの「GUI を使ったログ・グループの設定」

ノードを定義できない

ノードを定義できない場合は、ホスト名解決問題が存在している可能性があります。51 ページの「ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf」を参照してください。

システムがハングする

以下に、システムがハングする原因を示します。

- ディスク・ドライブがオーバーランした。
- ハートビートが切断された。この場合、withdrawl of node を示すメッセージが表示されます。
- 最後の手段として、システムのノンマスカブル割込み (NMI: non-maskable interrupt) を行った後、日本 SGI まで連絡してください。メモリのイメージを保存して後で解析できるよう、NMI はカーネルにノードをパニックさせるよう命令します。詳細については、ノードに対するオーナーのガイドを参照してください。

  vmcore.#.comp、unix.#、/var/adm/SYSLOG、およびクラスタ・ログ・ファイルを使用可能にします。

ノードは検出されるが、メンバーシップに設定されない

ノードが /var/adm/SYSLOG で検出されるのに Membership delivered メッセージを受取らない場合は、ネットワークの問題が存在する可能性があります。51 ページの「IRIX システム・ファイルの設定」を参照してください。

セル ID カウントと“ Membership Delivered” メッセージ

/var/adm/SYSLOG ファイルの Membership delivered メッセージには、新しい CXFS メンバーシップのメンバーであるノードのセル ID のリストが含まれます。

セル ID の後にはメンバーシップ・バージョンを表す数字が続き、ノードがメンバーシップに設定されてからメンバーシップが変更された回数を示します。

/var/adm/SYSLOG に Membership delivered メッセージが頻繁に記録される場合は、ネットワークの問題を示している可能性があります。

- 安定していてメンバーシップに残っているノードのメンバーシップ・バージョン番号は大きくなります。
- 問題のあるノードでは、メッセージがないか、またはメンバーシップ・バージョン番号が小さくなります。

51 ページの「IRIX システム・ファイルの設定」を参照してください。

ログインできない

CXFS 管理ノードにログインできない場合は、現在のノードがその他のノードの .rhosts ファイルにリストされていれば、以下のいずれかのコマンドを使用できます。

```
# rsh hostname ksh -i
# rsh hostname csh -i
```

ファイルシステムの I/O エラー

以下のメッセージは、問題が存在することを示しています(ここでは、読みやすくするために出力行を改行してあります)。

```
ALERT: I/O error in filesystem ("/mnt") meta-data dev 0xbd block 0x41df03 ("xlog_iodone")
ALERT:     b_error 0 b_bcount 32768 b_resid 0
NOTICE: xfs_force_shutdown(/mnt,0x2) called from line 966 of file ../fs/xfs/xfs_log.c.
  Return address = 0xc0000000008626e8
ALERT: I/O Error Detected.  Shutting down filesystem: /mnt
ALERT: Please umount the filesystem, and rectify the problem(s)
```

XFS ログにメタデータがない場合に限り、xfs_repair を使用してこの問題を解決できます。適切な手順については、268 ページの「xfs_repair の適切な使用」を参照してください。

クラスタの名前を変更後、ファイルシステムをマウントできない

ファイルシステムの定義後に、クラスタの名前を変更する(古いクラスタを削除して新しいクラスタを定義する)と、CXFS は、既存のファイルシステムをマウントできなくなります。これは、新しいクラスタが CXFS ファイルシステムの存在するクラスタ XVM ボリュームにアクセスできないため、それらのボリュームが外部ボリュームと見なされることが原因で発生します。

ファイルシステムを新しいクラスタにマウントするには、XVM steal コマンドを使用して、クラスタ XVM ボリュームを新しいクラスタのドメインに設定する必要があります。詳細については、『XVM Volume Manager Administrator's Guide』を参照してください。

GUI に無効なファイルシステムが表示される

以前にスライスしたディスク上に、ディスク上にすでに存在するスライスと同じ開始ブロックを持つ新しいスライスを作成していて、古いスライスにファイルシステムがある場合は、有効ではなくても古いファイルシステム が GUI に表示されます。

エラー・メッセージの理解

この節では、表示されることがあるいくつかのエラー・メッセージについて説明します。通常、メッセージの例は、まずタイプ順にリストされた後、メッセージ ID またはテキストのアルファベット順でリストされています。

以下の節が含まれます。

- 「通常メッセージ」
- 275 ページの「clconfd デーモンが異常終了する場合」
- 276 ページの「論理スワップ・スペース不足」
- 276 ページの「クラスタ名 ID 欠如エラー」
- 277 ページの「CXFS メンバーシップの取消し」
- 277 ページの「ライセンス・エラー」
- 278 ページの「IP アドレス・エラー」
- 278 ページの「SYSLOG エラー」
- 286 ページの「ログ・ファイル・エラー」

通常メッセージ

以下のメッセージが表示される場合があります。これらは通常のメッセージで、問題を示すものではありません。

```
NOTICE: Error reading mesg header 4 channel 1 cell 2
```

ノード 2 との接続用の MEMBERSHIP メッセージ・チャネル (チャネル 1。チャネル 0 は CXFS および XVM データのメイン・チャネルです) のエラー番号 4 (EINTR)。EINTR は、このメッセージ・チャネルが意図的に切断されていることを示すもので、それ自体にエラーがあることを示すわけではありません。その他のエラー・番号は実際のエラーで、ローカル・ノードはその他のノードに異常が発生したことを宣言します。これは情報メッセージです。修正処置は必要ありません。

```
NOTICE: Membership delivered.  Membership contains 0(21) 1(12) cells
```

ノード 0 およびノード 1 が CXFS メンバーシップに含まれています。ノード 0 は最後の 21 個の CXFS メンバーシップに含まれ、ノード 1 は最後の 12 個のメンバーシップに含まれます。ノードが定足数に追加されたり定足数から削除されると、そのたびにメンバーシップが形成されます。この数字は、定足数変更プロセスの間にメンバーシップが確認されたときにも増やされるので、連続する数字にはなりません。これは情報メッセージです。修正処置は必要ありません。

```
NOTICE: Resetting cells 0x4
```

この数字は、リセットが要求されたノード番号のビットマスクです。この場合、0x4 はノード 2 と同じです。これは情報メッセージです。修正処置は必要ありません。

```
CI_FAILURE, Cell 1 Machine cxfs1:  server has no information about a
machine that has reset capabilities for this machine
```

このノードにはリセット・メカニズムが提供されていないので、異常が発生しても自動的にリセットされません。リセット機能がない場合、このメッセージは無視してかまいません。サーバとして使用可能なノードが 2 つだけのクラスタでは、シリアル・ハードウェア・リセット回線の使用が必須です。また、サーバとして使用可能なすべてのノードに対しては、シリアル・ハードウェア・リセット回線の使用を強くお薦めします。より大きなクラスタでは、サーバとして使用可能なノードを奇数個使用する必要があります。

```
NOTICE: Error reading mesg header 4 channel 1 cell 2
```

mesg header 4 というテキストは、これが単なる情報メッセージであることを示します。

```
clconfd[16574]:  <<CI> E config 2> CI_ERR_NOTFOUND, Error reading CMS
status for machine tango, assuming machine is FailSafe-disabled in
cluster twango.
```

これは、クラスタが CXFS だけであり、IRIS FailSafe は使用されていないことを示します。

```
CI_CLCONFERR_INIT in ep_name() not binding socket
```

このメッセージはデーモンの開始前に表示されます。

```
clconfd[16574]:  <<CI> E clconf 0> CI_CLCONFERR_INIT, in ep_name():  not
binding socket
```

clconfd メッセージはデーモンの開始中に表示されます。

```
date <I0 clconfd clconf 610:0 clconfd_client.c:84> client registration:
clconfinfo, id 9119
date<I0 clconfd clconf 610:0 clconfd_service.c:781> sending reply
configuration and membership msg to client:  clconfinfo, id 9119
date <I0 clconfd clconf 610:0 clconfd_client.c:96> client
un-registration:  clconfinfo, id 9119
```

これらのメッセージは、clcon_info コマンドを実行する場合に発行されます。clconf_info コマンドは、まず clconfd に CXFS クライアントとして登録されます。続いて、その設定とメンバーシップのステータスの要求に対する応答メッセージを受信し、完了すると登録を解除します。

```
date <I0 clconfd clconf 610:0 clconfd_service.c:781 sending reply
configuration and membership msg to client:  cad, id 602
```

このメッセージは、cad デーモンがステータスについて定期的に clconfd をポーリングしていることを示します。clconf_info と異なり、cad は毎回登録と登録解除は行いません。これは、cad はデーモンであり、各要求の後に終了しないからです。cad または clconfd が再開されたときは、cad の登録または登録解除メッセージだけが表示されます。

```
dcvn_import_force:  error 1502 from invk_dsvn_obtain_exist
```

これは、回復プロセス中に送信される正常メッセージです。

clconfd デーモンが異常終了する場合

clconfd デーモンが開始直後に終了する場合は、CXFS ライセンスが適切にインストールされていないことを意味します。関連するエラー・メッセージについての詳細は、277 ページの「ライセンス・エラー」を参照してください。

CXFS を使用するには、各ノードにライセンスをインストールしなければなりません。システムの CPU の数を増やすと、新しいライセンスが必要になることがあります。43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。

論理スワップ・スペース不足

以下の /var/adm/SYSLOG メッセージの例では、過剰に購読されているシステムを示します。

```
ALERT: inetd [164] - out of logical swap space during fork while
allocating uarea - see swap(1M)
Availsmem 8207 availrmem 427 rlx freemem 10, real freemem 9
```

266 ページの「システム容量の効果的な使用」を参照してください。

この場合は、クラスタ・デーモンでメモリ・リークも発生している可能性があり、クラスタ・デーモンの再開が必要なことがあります。

```
# /etc/init.d/cluster restart
```

クラスタ名 ID 欠如エラー

例は、次のとおりです。

```
Mar  1 15:06:18 5A:nt-test-07 unix: NOTICE: Physvol (name cip4) has no
CLUSTER name id: set to ""
```

このメッセージには以下の意味があります。

- XVM physvol というラベル付きのディスクは、IRIX 6.5.6f でラベルが付けられた可能性があります。その後、システムは、新しいバージョンの VXM ラベル形式を使用する新しいバージョンにアップグレードされています。これは問題を示すわけではありません。
- XVM が、XVM クラスタの physvol というラベル付きのディスクを検出したときに、クラスタ名がまだ設定されていませんでした。これは、このホストでノードまたはクラスタの初期化が完了する前に XVM がディスク・インベントリの初期スキャンを実行したときの通常の出力です。

  このメッセージは、XVM クラスタの physvol というラベル付きのディスクは XVM によって認識されているが、このノードがまだ CXFS メンバーシップに設定されていないことを示します。したがって、クラスタ名は空 ("") になっています。

  ノードまたはクラスタの初期化時に、XVM はディスク・インベントリを再スキャンして、XVM クラスタの physvol ラベルを検索します。この時点では、このホストに対してクラスタ名が設定されている必要があります。ノードまたはクラスタの初期化後にクラスタ名が空の場合は、クラスタの初期化に問題があることを示します。

  このディスクの XVM 要素の設定を初めて変更すると、ラベルが更新されて新しいラベル形式に切替えられ、これらの通知は表示されなくなります。

  XVM についての詳細は、『XVM Volume Manager Administrator's Guide』を参照してください。

CXFS メンバーシップの取消し

/var/adm/SYSLOG の以下のメッセージは、カーネルによって実行された CXFS メンバーシップの取消しを示します。

```
Membership lost - withdrawing from cluster
```

この場合は、ローカル・ノードの CXFS メンバーシップをアクティブに許可する必要があります。128 ページの「GUI を使ったローカル・ノードのメンバーシップ許可」または 177 ページの「cmgr を使ったローカル・ノードのメンバーシップ許可」を参照してください。

ライセンス・エラー

/var/cluster/ha/log/clconf_hostname ログ・ファイルに以下のメッセージが記録されている場合は、CXFS ライセンスが適切にインストールされていなかったことを示します。

```
CXFS not properly licensed for this host.  Run
               '/usr/cluster/bin/cxfslicense -d'
       for detailed failure information.
```

CXFS ライセンスが適切にインストールされていない場合は、CXFS を実行すると、コンソールに次のエラーが表示されます。

```
Cluster services:CXFS not properly licensed for this host.  Run
        '/usr/cluster/bin/cxfslicense -d'
for detailed failure information.  After fixing the
license, please run '/etc/init.d/cluster restart'.
```

次の例のようなエラーが SYSLOG ファイルに表示されます。

```
Mar  4 12:58:05 6X:typhoon-q32 crsd[533]: <<CI> N crs 0> Crsd restarted.
Mar  4 12:58:05 6X:typhoon-q32 clconfd[537]: <<CI> N clconf 0>
Mar  4 12:58:05 5B:typhoon-q32 CLCONFD failed the CXFS license check.Use the
Mar  4 12:58:05 5B:typhoon-q32    '/usr/cluster/bin/cxfslicense -d'
Mar  4 12:58:05 5B:typhoon-q32 command to diagnose the license problem.
```

clconfd デーモンが開始直後に異常終了する場合は、このエラーが存在します。

CXFS を使用するには、各ノードにライセンスをインストールしなければなりません。43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。

IP アドレス・エラー

サイトにクラスタ ID 番号の競合がある場合は、以下のようなエラーが表示されます。

```
WARNING: mtcp ignoring  alive message from 1 with wrong ip addr 128.162.89.34
WARNING: mtcp ignoring  alive message from 0 with wrong ip addr 128.162.89.33
```

クラスタ ID 番号は一意でなければなりません。この問題を解決するには、クラスタ ID 番号を一意にします。

特定のノードに以前の設定の古い情報があるときに、クラスタ設定を再定義して CXFS サービスを開始すると、このエラーが発生する可能性があります。

問題を解決するには、まず 262 ページの「残っているクラスタの削除」の手順を試してみてください。これらの手順で問題が解決されない場合は、古い情報を持つノードを再起動します。古い情報を持っているノードの判別方法として、古いノードではすべてのノードに問題があることが示されますが、最新のノードでは古いノードに問題があることしか示されないことを利用できます。また、古いノードの /var/cluster/ha/log/clconfd_ ログ・ファイルには、SGI_CMS_CONFIG_ID 異常に関するエラー・メッセージも記録されます。

エラー・メッセージが多すぎて古いノードを認識できない場合は、すべてのノードを再起動してください。

SYSLOG エラー

CXFS は、通常の処理と重大なエラーの両方を、/var/adm/SYSLOG のほかに各ログ・グループの個々のログにも記録します。

一般的に、/var/adm/SYSLOG ファイルのエラーは以下の形式になります。

timestamp priority_&_facility : hostname process[ID]: <internal_info> CODE message_text

例は、次のとおりです。

```
Sep  7 11:12:59 6X:cxfs0 cli[5830]: < E clconf 0> CI_IPCERR_NOSERVER, clconf
ipc: ipcclnt_connect() failed, file /var/cluster/ha/comm/clconfd-ipc_cxfs0
```

表10-1 に、この SYSLOG メッセージの各部分を示します。

表10-1 SYSLOG エラー・メッセージの形式

内容	部分	意味
Sep 7 11:12:59	タイムスタンプ	9 月 7 日午前 11 時 12 分
6X	機能とラベル	6X は情報メッセージを示します。syslogd(1M) およびファイル /usr/include/sys/syslog.h を参照してください。
cxfs0	ノード名	論理名が cxfs0 のノードは、プロセスが実行されているノードです。
cli[5830]	プロセス[ID]	メッセージを送信するプロセスは cli で、プロセス ID 番号は 5830 です。
<CI>E clconf 0	内部情報: メッセージ・ソース、ログ・サブシステム、およびスレッド ID	このメッセージはクラスタ・インフラストラクチャ (CI: Cluster Infrastructure) からのものです。E は、エラーであることを示します。clconf コマンドはログ・サブシステムです。0 は、マルチスレッド化されていないことを示します。
CI_IPCERR_NOSERVER, clconf ipc	内部エラー・コード	メッセージのタイプに関する情報。この場合、サーバを示すメッセージはありません。通常メッセージの場合、エラー・コードは出力されません。
ipcclnt_connect() failed, file /var/cluster/ha/comm/clconfd-ipc_cxfs0	メッセージ・テキスト	clconfd-ipc_cxfs0 ファイルの接続に失敗しました。

以下の節では、メッセージ ID とテキストだけを示します。

cli エラー

cli メッセージについては、コマンドからの最後のメッセージ(CLI private command failed で始まります)だけに意味があります。その他の cli メッセージはすべて無視してかまいません。

以下に、cli デーモンのエラーの例を示します。

```
CI_ERR_INVAL, CLI private command:  failed (Machine (cxfs0) exists.)
```

論理名 cxfs0 を使用して新しいノード定義を作成しようとしましたが、そのノード名はすでにクラスタ・データベースに存在します。別の名前を選択してください。

```
CI_ERR_INVAL, CLI private command:  failed (IP address (128.162.89.33)
specified for control network is cxfs0 is assigned to control network of
machine (cxfs0).)
```

ノード cxfs0 の 2 つの異なるコントロール・ネットワークに対して同じ IP アドレスを指定しました。別の IP アドレスを使用してください。

```
CI_FAILURE, CLI private command:  failed (Unable to validate hostname of
machine (cxfs0) being modified.)
```

cxfs0 名の DNS 解決に失敗しました。この問題を解決するには、すべてのノードの /etc/hosts に cxfs0 のエントリを追加します。

```
CI_IPCERR_NOPULSE, CLI private command:  failed (Cluster state is
UNKNOWN.)
```

クラスタの状態が「不明(UNKNOWN)」で、コマンドを完了できませんでした。これは一時的なエラーです。ただし、エラーが解決されない場合は、クラスタ・デーモンを停止して再開してください。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

clconfd エラー

以下に、clconfd デーモンから送信されるエラーを示します。

```
CI_CONFERR_NOTFOUND, Could not access root node.
```

クラスタ・データベースが存在しないかまたは壊れています。あるいはデータベース・デーモンが応答していません。データベースが存在することを確認してください。

エラーが表示される場合や、ダンプが空の場合は、データベースを再作成してください。詳細については、291 ページの「クラスタ・データベースのクリア」を参照してください。

データベースが存在する場合は、クラスタ・デーモンを再開します。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

```
CI_ERR_NOTFOUND, Could not get Cellular status for local machine (cxfs1)
```

データベースが壊れているか、またはデータベースにアクセスできません。上記と同じアクションを実行してください。

```
CI_FAILURE, Call to open cdb for logging configuration when it is
already open.
```

これはソフトウェアの問題を示しており、デーモンを再開する必要があります。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

```
CI_FAILURE, Cell 1 Machine cxfs1:  server has no information about a
machine that has reset capabilities for this machine
```

このノードにはリセット・メカニズムが提供されていないので、異常が発生しても自動的にリセットされません。異常処理が適切に行われるようにするには、GUI または cmgr(1M) を使用してノードの定義を変更し、リセット情報を追加します。109 ページの「GUI を使ったノードの定義」または 156 ページの「cmgr を使ったノードの変更」を参照してください。

```
CI_FAILURE, CMD(/sbin/umount -k /dev/xvm/bob1):  exited with status 1
(0x1)
```

/dev/xvm/bob1 ファイルシステムをアンマウントするときにエラーが発生しました。通常は、このメッセージの直前に umount(1M) コマンドからのメッセージが発行され、異常の理由についての詳細が提供されます。

```
CI_FAILURE, CMD(/sbin/clmount -o 'server_list=(cxfs0,cxfs1)'
/dev/xvm/bob2 /bob2):  exited with status 1 (0x1)
```

/dev/xvm/bob2 ファイルシステムをマウントするときにエラーが発生しました。通常は、このメッセージの直前に mount(1M) コマンドからのメッセージが発行され、異常の理由についての詳細が提供されます。

```
CI_FAILURE, CMD(/sbin/clmount -o 'server_list=(cxfs2,cxfs0)'
/dev/xvm/stripe4 /xvm/stripe4):  exited with status 1 (0x1)
```

最初に mkfs(1M) を実行せずにファイルシステムをマウントしようとしました。ファイルシステムをマウントする前に、mkfs(1M) を使用してファイルシステムを構築する必要があります。詳細については、mkfs(1M) のマン・ページを参照してください。

```
CI_FAILURE, Could not write newincarnation number to CDB, error = 9.
```

クラスタ・データベースにアクセスできませんでした。操作を再試行してください。エラーが解決されない場合は、クラスタ・デーモンを停止して再開してください。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

問題が解決されない場合は、データベースをクリアして再起動し、データベースを再作成してください。291 ページの「クラスタ・データベースのクリア」を参照してください。

```
CI_FAILURE, Exiting, monitoring agent should revive me.
```

このデーモンには新しいデータが必要です。自動的に再開されます。

```
CI_FAILURE, No node for client (3) of filesystem (/dev/xvm/bob1) on
(/bob1).
```

(このメッセージは何度も表示される場合があります。) クラスタ・データベースに存在していない CXFS クライアント・ノードにファイルシステムがマウントされたままになっている可能性があります。以前にクラスタで使用されていて現在もファイルシステムをマウントしている CXFS クライアント・ノードを識別できる場合は、そのノードを再起動します。それ以外の場合は、クラスタ全体を再起動します。

```
CI_FAILURE, No node for server (-1) of filesystem (/dev/xvm/bob1) on
(/bob1).
```

(このメッセージは何度も表示される場合があります。) クラスタ・データベースに存在していないサーバ・ノードにファイルシステムがマウントされたままになっている可能性があります。以前にクラスタで使用されていて現在もファイルシステムをマウントしているサーバを識別できる場合は、そのノードを再起動します。それ以外の場合は、クラスタ全体を再起動します。

```
CI_ FAILURE, Node cxfs0:  SGI_CMS_HOST_ID(tcp,128.162.8 >9.33) error 149
(Operation already in progress)
```

カーネルはすでにこの情報を持っています。このメッセージは無視してかまいません。

```
CI_FAILURE, Unregistered from crs.
```

clconfd デーモンがリセット・デーモンに接続されていないので、異常ノードをリセットを処理できません。修正処置はありません。

```
CI_IPCERR_NOSERVER, Crs_register failed,will retry later.  Resetting not
possible yet.
```

clconfd デーモンがリセット・デーモンに接続できません。異常ノードのリセットを処理できません。リセット・デーモンのログ・ファイル (/var/cluster/ha/log/crsd_) でその他のエラー・メッセージがないかどうかをチェックしてください。

```
Clconfd is out of membership, will restart after notifying clients.
```

clconfd デーモンに、クラスタの現在の状態に関する十分な情報がありません。このデーモンは終了して、自動的に新しいデータで再開されます。

```
CMD(/sbin/clmount -o 'server_list=(cxfs2,cxfs0)' /dev/xvm/stripe4
/xvm/stripe4):  /dev/xvm/stripe4:  Invalid argument
```

最初に mkfs(1M) を実行せずにファイルシステムをマウントしようとしました。ファイルシステムをマウントする前に、mkfs(1M) を使用してファイルシステムを構築する必要があります。詳細については、mkfs(1M) のマン・ページを参照してください。

```
CMD(/sbin/clmount -o 'server_list=(cxfs0,cxfs1)' /dev/xvm/bob2 /bob2):
/dev/xvm/bob2:  Invalid argumentSep 9 14:12:43 6X:cxfs0 clconfd[345]:  <
E clconf 3> CI_FAILURE, CMD(/sbin/clmount -o 'server_list=(cxfs0,cxfs1)'
/dev/xvm/bob2 /bob2):  exited with status 1 (0x1)
```

最初のメッセージは clmount コマンド(内部 CXFS マウント・コマンド)からのもので、エラーを説明しています(無効な引数が発行されました)。2 番目のメッセージは、マウントに失敗したことを示します。

crsd エラー

以下に、crsd デーモンから送信されるエラーを示します。

```
CI_ERR_NOTFOUND, No logging entries found for group crsd, no logging
will take place - Database entry #global#logging#crsd not found.
```

クラスタ・データベースに crsd ログ定義がありませんでした。これは、データベースを作成せずにクラスタ・プロセスを開始した場合に発生することがあります。295 ページの「クラスタ・データベースの再作成」を参照してください。

```
CI_ERR_RETRY, Could not find machine listing.
```

crsd デーモンは、クラスタ・データベースでローカル・ノードを見つけることができませんでした。ローカル・ノード定義をまだ作成していない場合、このメッセージは無視してかまいません。

```
CI_ERR_SYS:125, bind() failed.
```

/etc/services ファイルの sgi-crsd のポート番号が一意ではないか、またはこのファイルに sgi-crsd エントリがありません。このエントリの追加についての詳細は、54 ページの「CXFS 管理ノードの /etc/services」を参照してください。

```
CI_FAILURE, Entry for sgi-crsd is missing in /etc/services.
```

/etc/services ファイルに sgi-crsd エントリがありません。このエントリの追加についての詳細は、54 ページの「CXFS 管理ノードの /etc/services」を参照してください。

```
CI_FAILURE, Initialization failed, exiting.
```

このメッセージは、連続する複数のメッセージの最後に示されます。異常の原因を判別するには、このメッセージの前のメッセージを参照してください。

cmond エラー

以下に、cmond デーモンから送信されるエラーを示します。

```
Could not register for notification.cdb_error = 7
```

エラー番号 7 は、クラスタ・プロセスが開始されたときにクラスタ・データベースが初期化されなかったことを示します。

これは、特定の CXFS 管理ノードで cdbreinit を実行する場合に、プール内のほかの複数の CXFS 管理ノードでまだ fs2d が実行中で、該当するノードがすでにデータベースにリストされているときに発生することがあります。

以下を実行します。

1. エラーが表示されるノードに対して以下のコマンドを実行します。

   # **/usr/cluster/bin//cdb-init-std-nodes**

   このコマンドは、データベースのほかのノードに影響を与えずに、見つからないノードを再作成します。

2. エラーが解決されない場合は、以下のコマンドを実行してデーモンを強制的に再開します。

   # **/etc/init.d/cluster restart**

   cmond が再開されたことを確認します。

3. エラーが解決されない場合は、問題のあるノードだけでデータベースを再初期化します。

4. それでもエラーが解決されない場合は、クラスタ内のすべてのノードを再初期化します。

295 ページの「クラスタ・データベースの再作成」を参照してください。

```
Process clconfd:343 of group cluster_cx exited, status = 3.
```

clconfd プロセスがステータス 3 で終了しました。つまり、このプロセスは cmond によって再開されません。修正処置は必要ありません。

```
Process crsd:1790 of group cluster_control exited, status = 127
```

crsd プロセスがエラー(ゼロ以外)のステータスで終了しました。対応するデーモンのログでエラー・メッセージを検索してください。

fs2d エラー

以下に、fs2d デーモンから送信されるエラーを示します。

```
Error 9 writing CDB info attribute for node
#cluster#elaine#machines#cxfs2#Cellular#status
```

クラスタ・データベースへの書込みの際に内部エラーが発生しました。操作を再試行してください。エラーが解決されない場合は、クラスタ・デーモンを停止して再開してください。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

問題が解決されない場合は、データベースをクリアして再起動し、データベースを再作成してください。291 ページの「クラスタ・データベースのクリア」を参照してください。

```
Error 9 writing CDB string value for node
#cluster#elaine#machines#cxfs2#Cellular#status
```

クラスタ・データベースへの書込みの際に内部エラーが発生しました。操作を再試行してください。エラーが解決されない場合は、クラスタ・デーモンを停止して再開してください。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

問題が解決されない場合は、データベースをクリアして再起動し、データベースを再作成してください。291 ページの「クラスタ・データベースのクリア」を参照してください。

```
Failed to update CDB for node
#cluster#elaine#Cellular#FileSystems#fs1#FSStatus
```

クラスタ・データベースへの書込みの際に内部エラーが発生しました。操作を再試行してください。エラーが解決されない場合は、クラスタ・デーモンを停止して再開してください。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

問題が解決されない場合は、データベースをクリアして再起動し、データベースを再作成してください。291 ページの「クラスタ・データベースのクリア」を参照してください。

```
Failed to update CDB for node
#cluster#elaine#machines#cxfs2#Cellular#status
```

クラスタ・データベースへの書込みの際に内部エラーが発生しました。操作を再試行してください。エラーが解決されない場合は、クラスタ・デーモンを停止して再開してください。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

問題が解決されない場合は、データベースをクリアして再起動し、データベースを再作成してください。291 ページの「クラスタ・データベースのクリア」を参照してください。

```
Machine 101 machine_sync failed with lock_timeout error
```

fs2d デーモンがクラスタ・データベースを同期できず、sync プロセスがタイムアウトしました。この操作は、fs2d によって自動的に再試行されます。

```
ALERT: CXFS Recovery:  Cell 0:  Server Cell 2 Died, Recovering
```

サーバ (cell 2) がダウンしたため、現在、システムがファイルシステムを回復しています。

一般メッセージ

```
CI_CONFERR_NOTFOUND, Logging configuration error:  could not read
cluster database /var/cluster/cdb/cdb.db, cdb error = 3.
```

クラスタ・データベースが初期化されていません。295 ページの「クラスタ・データベースの再作成」を参照してください。

```
WARNING: Error receiving messages from cell 2 tcpchannel 1
```

CXFS メンバーシップ・チャネル (チャネル 1。チャネル 0 は CXFS および XVM データのメイン・メッセージ・チャネルです) にエラーがあります。これは、チャネルを切断した結果か、またはノード (この場合は ID が 2 のノード) のエラーの可能性があります。修正処置はありません。

ログ・ファイル・エラー

CXFS では、各 CXFS デーモンのログを維持します。これらのログのカスタマイズについての詳細は、126 ページの「GUI を使ったログの設定」を参照してください。

以下に、ログ・メッセージの形式を示します。

*daemon*_log *timestamp internal_process: message_text*

例は、次のとおりです。

```
cad_log:Thu Sep  2 17:25:06.092  cclconf_poll_clconfd: clconf_poll failed with error CI_IPCERR_NOPULSE
```

287 ページの表10-2 に、このメッセージの各部分を示します。

表10-2 ログ・エラー・メッセージの形式

内容	部分	意味
cad_log	タイムスタンプ	このメッセージは cad デーモンに関係しています。
Sep 2 17:25:06.092	タイムスタンプおよびプロセス ID	9 月 2 日午後 5 時 25 分、プロセス ID 92
cclconf_poll_clconfd	内部プロセス情報	内部プロセス情報
clconf_poll failed with error CI_IPCERR_NOPULSE	メッセージ・テキスト	クラスタのステータスの更新を取得するために clconfd デーモンと通信できませんでした。

cad メッセージ

以下に、/var/cluster/ha/log/cad_log メッセージの例を示します。

ccacdb_cam_open: failed to open connection to CAM server error 4

内部メッセージ。cad の動作は自動的に再試行されるので無視してかまいません。

ccamail_cam_open: failed to open connection to CAM server error 4

内部メッセージ。cad の動作は自動的に再試行されるので無視してかまいません。

ccicdb_cam_open: failed to open connection to CAM server error 4

内部メッセージ。cad の動作は自動的に再試行されるので無視してかまいません。

cclconf_cam_open: failed to open connection to CAM server error 4

内部メッセージ。cad の動作は自動的に再試行されるので無視してかまいません。

cclconf_poll_clconfd: clconf_poll failed with error CI_IPCERR_NOCONN

clconfd デーモンが実行されていないか、または外部の要求に応答していません。エラーが解決されない場合は、クラスタ・デーモンを停止して再開してください。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

```
cclconf_poll_clconfd:  clconf_poll failed with error CI_IPCERR_NOPULSE
```

クラスタのステータスの更新を取得するために clconfd デーモンと通信できませんでした。エラーが解決されない場合は、クラスタ・デーモンを停止して再開してください。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

```
cclconf_poll_clconfd:  clconf_poll failed with error CI_CLCONFERR_LONELY
```

clconfd デーモンは、クラスタの正確な状態を提供するための十分な情報を持っていません。このデーモンは新しいデータを使用して自動的に再開され、サービスを再開します。

```
csrm_cam_open:  failed to open connection to CAM server error 4
```

内部メッセージ。cad の動作は自動的に再試行されるので無視してかまいません。

```
Could not execute notification cmd.  system() failed.  Error:  No child
processes
```

cad がプロセスをフォークできなかったため、メール・メッセージは送信されませんでした。クラスタ・デーモンを停止して再開します。295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」を参照してください。

```
error 3 sending event notification to client 0x000000021010f078
```

GUI プロセスがクリーンアップせずに終了しました。

```
error 8 sending event notification to client 0x000000031010f138
```

GUI プロセスがクリーンアップせずに終了しました。

cli メッセージ

以下に、/var/cluster/ha/log/cli_*hostname* メッセージの例を示します。

```
CI_CONFERR_NOTFOUND, No machines found in the CDB.
```

ローカル・ノードはクラスタ・データベースに定義されていません。

```
CI_ERR_INVAL, Cluster (bob) not defined
```

bob というクラスタはクラスタ・データベースに存在しません。

```
CI_ERR_INVAL, CLI private command:  failed (Cluster (bob) not defined)
```

bob というクラスタはクラスタ・データベースに存在しません。

```
CI_IPCERR_AGAIN, ipcclnt_connect():  file
/var/cluster/ha/comm/clconfd-ipc_cxfs0 lock failed - Permission denied
```

CLI が root 以外のログインによって呼出されました。root としてログインしたときは、cmgr(1M) のみ使用してください。

```
CI_IPCERR_NOPULSE, CLI private command:  failed (Cluster state is
UNKNOWN.)
```

クラスタの状態を判別できませんでした。clconfd(1M) デーモンが実行されているかどうかをチェックしてください。

```
CI_IPCERR_NOPULSE, ipcclnt_pulse_internal():  server failed to pulse
```

クラスタの状態を判別できませんでした。clconfd(1M) デーモンが実行されているかどうかをチェックしてください。

```
CI_IPCERR_NOSERVER, clconf ipc:  ipcclnt_connect() failed, file
/var/cluster/ha/comm/clconfd-ipc_cxfs0
```

ローカル・ノード (cxfs0) はクラスタ・データベースに定義されていません。

```
CI_IPCERR_NOSERVER, Connection file
/var/cluster/ha/comm/clconfd-ipc_cxfs0 not present.
```

ローカル・ノード (cxfs0) はクラスタ・データベースに定義されていません。

crsd エラー

以下に、/var/cluster/ha/log/crsd_*hostname* メッセージの例を示します。

```
CI_CONFERR_INVAL, Nodeid -1 is invalid.
I_CONFERR_INVAL, Error from ci_security_init().
CI_ERR_SYS:125, bind() failed.
CI_ERR_SYS:125, Initialization failed, exiting.
CI_ERR_NOTFOUND, Nodeid does not have a value.
CI_CONFERR_INVAL, Nodeid -1 is invalid.
```

これらの各メッセージが表示された場合は、ノード定義にノード ID が指定されていなかったか、またはクラスタ・データベースにノード定義を作成したときにそのノードでクラスタ・プロセスが実行されていませんでした。これは、必要なときにオプションの情報を使用できないことを示す警告です。

```
CI_ERR_NOTFOUND, SystemController information for node cxfs2 not found,
requests will be ignored.
```

ノード cxfs2 に対してシステム・コントローラ情報（オプションの情報）が指定されていませんでした。ノード定義を変更して、ノード cxfs2 に対してシステム・コントローラ情報を指定します。これは、必要なときにオプションの情報を使用できないことを示す警告です。この情報がないと、異常が発生してもノードはリセットされません。このため、クラスタが異常から正常に回復できなくなる可能性があります。

```
CI_ERR_NOTFOUND, SystemController information for node cxfs0 not found,
requests will be ignored.
```

ノード ID が 101 のノードに対してノード定義で指定されているオーナー・ノードが定義されていません。オーナー・ノードを定義してください。

```
CI_CRSERR_NOTFOUND, Reset request 0x10087d48 received for node 101, but
its owner node does not exist.
```

ノード ID が 101 のノードに対してノード定義で指定されているオーナー・ノードが定義されていません。オーナー・ノードを定義してください。

fs2d エラー

以下に、/var/cluster/ha/log/fs2d_*hostname* メッセージの例を示します。

```
Failed to copy global CDB to node cxfs1 (1), error 4
```

ローカル・ノードとノード cxfs2 の間に通信上の問題があります。これらの 2 つのノードのコントロール・ネットワークをチェックしてください。

```
Communication failure send new quorum to machine cxfs2 (102) (error
6003)
```

ローカル・ノードとノード cxfs2 の間に通信上の問題があります。これらの 2 つのノードのコントロール・ネットワークをチェックしてください。

```
Failed to copy CDB transaction to node cxfs2 (1)
```

ローカル・ノードとノード cxfs2 の間に通信上の問題があります。これらの 2 つのノードのコントロール・ネットワークをチェックしてください。

```
Outgoing RPC to hostname :  NULL
```

このメッセージが表示される場合は、RPC の設定をチェックします。詳細については、rpcinfo(1M)、rpcinfo(1M)、および portmap(1M) のマン・ページを参照してください。

修正処置

この節では、以下の修正処置について説明します。

- 291 ページの「CXFS サービスの再開」
- 291 ページの「クラスタ・データベースのクリア」
- 292 ページの「再起動」
- 293 ページの「2 ノード・クラスタの回復」
- 294 ページの「ノードをクラスタに再設定しない再起動」
- 295 ページの「クラスタ・インフラストラクチャ・デーモンの停止と再開」
- 295 ページの「クラスタ・データベースの再作成」

CXFS サービスの再開

再起動後に CXFS サービスが再開されない場合は、GUI または `cmgr`(1M) コマンドの停止機能を使用して開始フラグがオフにされている可能性があります。この場合、`/etc/init.d/cluster start` を発行してもサービスは再開されません。CXFS サービスを開始する必要があります。GUI または `cmgr` を使用してサービスを再開すると、以降の再起動時に CXFS サービスが再開されるよう設定されます。

詳細については、125 ページの「GUI を使った CXFS サービスの開始」または 172 ページの「`cmgr` を使った CXFS サービスの開始」を参照してください。

クラスタ・データベースのクリア

クラスタのすべてのノードのクラスタ・データベースをクリアするには、以下の手順に従います。次の手順に進む前に、各ノードで各手順を完了してください。

注意: この手順を実行すると、すべての設定情報が削除されます。

1. すべてのノードで以下のコマンドを入力します。

   # **/usr/cluster/bin/cmgr -c 'admin cxfs_stop'**

2. すべてのノードで以下のコマンドを入力します。

   # **/etc/init.d/cluster stop**

注意: 任意のノードに対して手順 3 を実行する前に、各ノードで手順 1 と 2 を完了してください。

3. すべてのノードで以下のコマンドを入力します。

   ```
   # /usr/cluster/bin/cdbreinit
   ```

   266 ページの「ノード ID またはクラスタ ID を変更する前の再起動」も参照してください。

4. すべてのノードで以下のコマンドを入力します。

   ```
   # /etc/init.d/cluster start
   ```

5. すべてのノードで以下のコマンドを入力します。

   ```
   # /usr/cluster/bin/cmgr -c 'admin cxfs_start'
   ```

カーネル内の古いと思われるクラスタ設定を取除くには、262 ページの「残っているクラスタの削除」を参照してください。必要に応じてノードを再起動します。

再起動

すべてのノードで個別に以下のコマンドを入力して、クラスタを再起動します。

```
# reboot
```

IRIX 以外のオペレーティング・システムが実行されているノードについての詳細は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』を参照してください。

ノードの再起動時に常に CXFS サービスを再開したい場合は、GUI または `cmgr`(1M) を使用して CXFS サービスを開始します。詳細については、125 ページの「GUI を使った CXFS サービスの開始」および 172 ページの「`cmgr` を使った CXFS サービスの開始」を参照してください。

以下に、再起動が必要になることがある状況を示します。

- v ノードが使用中であるために一部の CXFS クライアントがファイルシステムをマウントできない場合や、ノードをリセットしても問題が解決されない場合は、クラスタ内のすべてのノードの再起動が必要になる場合があります。
- 10 分以内に回復アクティビティが行われない場合は、該当するノードの再起動が必要になる場合があります。

2 ノード・クラスタの回復

以下のような状況であるとします。

1. サーバとして使用可能な以下の 2 つのノードを持つ clusterA という名前のクラスタがあり、CXFS タイブレーカー・ノードはありません。
   - node1
   - node2
2. node1 が停止し、しばらく停止したままになります。
3. node2 が回復し、clusterA は引続き動作しています。

メモ: 初期 CXFS カーネル・メンバーシップの形成後、既存のクラスタは、サーバとして使用可能な残りのノードの 50% まで低下する場合があります。詳細については、28 ページの「メンバーシップ定足数」を参照してください。

---

4. node2 が停止したために、clusterA に異常が発生します。
5. node2 が復帰します。ただし、クラスタの初期化には以下のいずれかが必要なので、clusterA は形成できません。
   - **50%** を超えるサーバとして使用可能なノード
   - サーバとして使用可能なノードの 50%。これらの 1 つは CXFS タイブレーカーです。

node2 が単独でクラスタを形成できるようにするには、以下を実行する必要があります。

1. GUI または cmgr を使用して、node2 を CXFS タイブレーカー・ノードに設定します。
   - 126 ページの「GUI を使ったタイブレーカー・ノードの設定」を参照してください。
   - 173 ページの「cmgr を使ったタイブレーカー・ノードの設定」を参照してください。
2. node2 の CXFS カーネル・メンバーシップを取消します。
   - 128 ページの「GUI を使ったローカル・ノードのメンバーシップ取消し」を参照してください。
   - cmgr で、以下のコマンドを入力します。

     ```
     cmgr> admin cxfs_stop
     ```

     176 ページの「cmgr を使ったローカル・ノードのメンバーシップ取消し」を参照してください。
3. node2 の CXFS カーネル・メンバーシップを許可します。

- 128 ページの「GUI を使ったローカル・ノードのメンバーシップ許可」を参照してください。
- cmgr で、以下のコマンドを入力します。

  ```
  cmgr> admin cxfs_start
  ```

  177 ページの「cmgr を使ったローカル・ノードのメンバーシップ許可」を参照してください。

4. CXFS タイブレーカー・ノードの機能の設定を解除します。

注意: サーバとして使用可能な 2 つのノードで構成されるクラスタ内の CXFS タイブレーカー・ノードに異常が発生したり、管理者が CXFS サービスを停止した場合は、その他のノードによって強制シャットダウンが実行され、CXFS ファイルシステムはすべてアンマウントされます。データの整合性を保証するため、サーバとして使用可能なノードが 2 つだけのクラスタでは、シリアル・ハードウェア・リセット機能の使用が必須です。また、サーバとして使用可能なすべてのノードに対しては、シリアル・ハードウェア・リセット機能の使用を強くお薦めします。より大きなクラスタでは、サーバとして使用可能なノードを奇数個使用し、ノードの 2 つのみがサーバとして使用可能なノードの場合は、シリアル・ハードウェア・リセット回線を使用するか、I/O 制限とスイッチを使用する必要があります。

GUI または cmgr を使用します。

- 126 ページの「GUI を使ったタイブレーカー・ノードの設定」
- 173 ページの「cmgr を使ったタイブレーカー・ノードの設定」

node1 は現在もクラスタに設定されているので、クラスタはノードが停止していても通信を試みます。したがって、CXFS カーネル・メンバーシップの形成とファイルシステムのマウントに多少時間がかかる場合があります。

ノードをクラスタに再設定しない再起動

chkconfig(1M) の cluster フラグは、ほかのクラスタ管理デーモンおよび複製されたクラスタ・データベースを制御します。これがオフの場合、データベース・デーモンは次回再起動されるときに開始されず、ほかのノードでクラスタ設定を変更してもデータベースのローカル・コピーは更新されません。これにより、後で問題が発生することがあります。特に、ノードの過半数でデータベース・デーモンが実行されていない場合は、問題が発生する可能性が高くなります。

クラスタ・デーモンが深刻な問題を引起していてマシンを再起動できない場合は、シングルユーザ・モードで起動して cluster フラグをオフにし、マルチユーザ・モードで再起動することで、ノードを回復できます。

```
# init 1
# /etc/chkconfig cluster off
```

```
# init 2
```

クラスタ・インフラストラクチャ・デーモンの停止と再開

クラスタ・インフラストラクチャ・デーモンを停止して再開するには、以下のコマンドを入力します。

```
# /etc/init.d/cluster stop
# /etc/init.d/cluster start
```

これらのコマンドは、クラスタ・インフラストラクチャ・デーモンだけを制御します。

注意: クラスタ・インフラストラクチャ・デーモンを停止すると、ノードはデータベースの更新を受付けず、カーネル設定を更新しなくなります。これにより、好ましくない影響が出る場合があります。ほとんどの場合、インフラストラクチャ・デーモンは常に実行したままにすることが推奨されています。これらのコマンドは、指示を受けた場合にのみ使用してください。

291 ページの「CXFS サービスの再開」も参照してください。デーモンについての一般情報は、297 ページの「デーモン」を参照してください。

クラスタ・データベースの再作成

初期クラスタ・データベースを再作成するには、以下の手順に従います。

1. データベース・メンバーシップ定足数が正常なデータベースを持つノードによって保持されるようにして、不正なデータベースが伝えられるのを避けます。
2. 以下を入力します。

   ```
   # /usr/cluster/bin/cdbreinit
   ```

メモ: 266 ページの「ノード ID またはクラスタ ID を変更する前の再起動」も参照してください。

CXFS ソフトウェアのアーキテクチャ

この付録では、管理ノードの以下の部分について説明します。

- 「デーモン」
- 300 ページの「通信パス」
- 305 ページの「同時実行クラスタ内の通信パス」
- 306 ページの「読取りおよび書込み時のメタデータの流れ」

『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』も参照してください。

デーモン

以下の表に、CXFS のデーモンとスレッドを示します。

メモ: CXFS は、バッファされている書込みをディスクに書込むために `xfsd` カーネル・スレッドを XFS と共有します。

(種類が `CXFS and FailSafe` の) 同時実行クラスタを使用している場合、FailSafe デーモンについての詳細は、『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。

表A-1 CXFS のデーモンとスレッド

層	サブシステム	プロセス	説明
クラスタ・サービス (CXFS)	cluster_services	clconfd	CXFS 管理クラスタ設定デーモン。CDB データベースからクラスタ設定を読取り、それに応じてローカル・カーネルの CXFS カーネル・メンバーシップ・サービスを管理します。
	cxfs_client	cxfs_client	CXFS クライアントのみのクラスタ設定デーモン。それに応じてローカル・カーネルの CXFS カーネル・メンバーシップ・サービスを管理します。
クラスタ・ソフトウェア・インフラストラクチャ(クラスタ管理プロセス)	cluster_admin	cad	クラスタ管理デーモン。管理サービスを提供します。
	cluster_control	crsd	ノード・コントロール・デーモン。その他のノードとのシリアル接続をモニタします。その他のノードをリセットする機能があります。
		cmond	その他のすべてのデーモンを管理するデーモン。このプロセスは、クラスタ内のすべてのノードでほかのプロセスを開始し、異常時にはこれらのプロセスを再開します。
		fs2d	データベースを管理し、プール内のすべてのノードで各コピーの同期を保ちます。
カーネル・スレッド	sthreads	cmsd	CXFS カーネル・メンバーシップとハートビートを管理します（CXFS cmsd はカーネルに常駐し、ユーザ空間に常駐する IRIS FailSafe cmsd とは異なります）。
		Recovery	ノードの回復プロトコルを管理します。
		corpseleader	ノード間の回復を調整します。

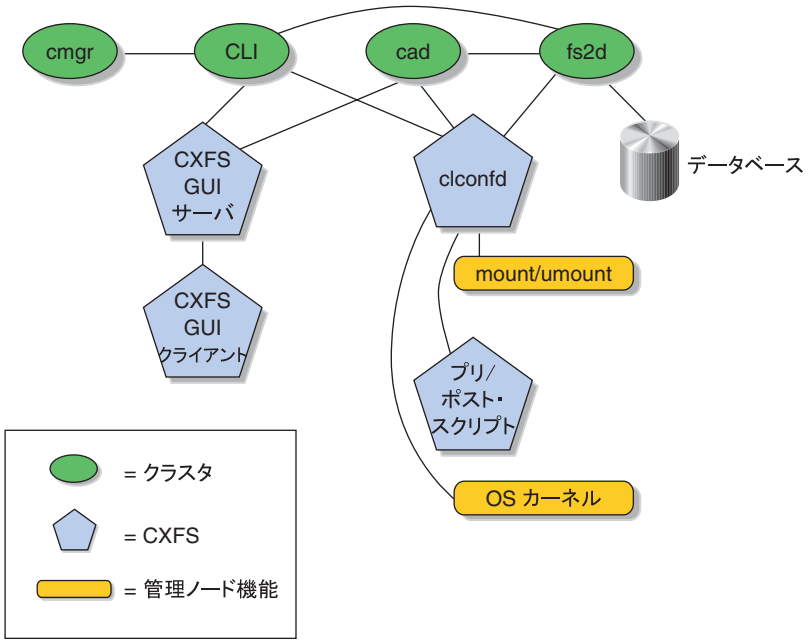
層	サブシステム	プロセス	説明
		dcshake	CXFS クライアント上の動作していない CXFS v ノードをパージします。
		cxfsd	クライアントからサーバへのエクステントとサイズの更新の送信を管理します。CXFS クライアント上で動作するこのデーモンは、クライアント上の変更された i ノードを取得して、サイズと、未書込みエクステントの変更内容をサーバに戻します。
	xthreads	mesgtcprcv	開いているメッセージ・チャネルごとに 1 つのメッセージを読取ります。
		mesgtcpaccept	新しい接続を受付けます。
		mesgtcpdiscovery	ほかのノードのモニタと検出を行います。
		mesgtcpmulticast	ハートビートを提供します。

fs2d、clconfd、および crsd デーモンは、リアルタイムの優先度で動作します。ただし、mount および umount コマンドとスクリプトは、clconfd により、通常の時分割方式の優先度で実行されます。

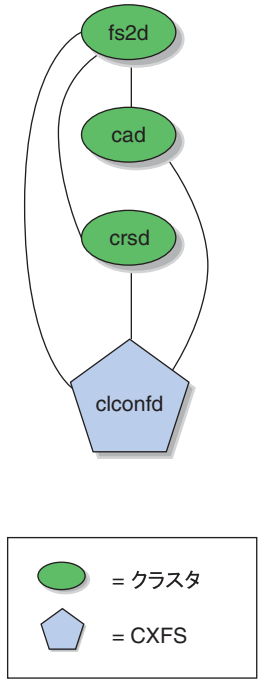
通信パス

以下の図に、CXFS の通信パスを示します。

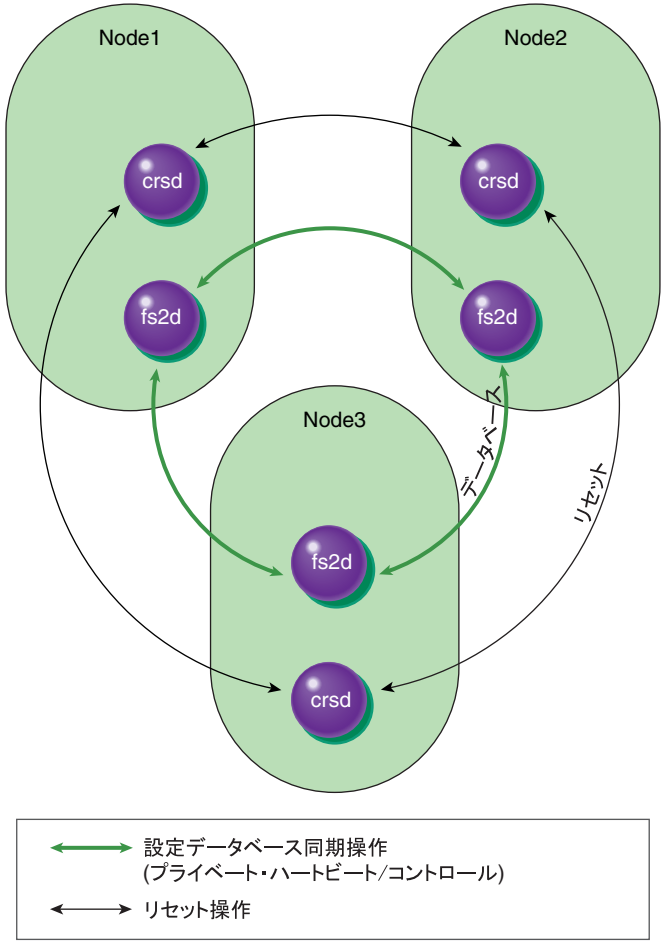
メモ: 以下の図には、cmond クラスタ・マネージャ・デーモンは表されていません。このデーモンの目的は、その他のデーモンを実行し続けることです。



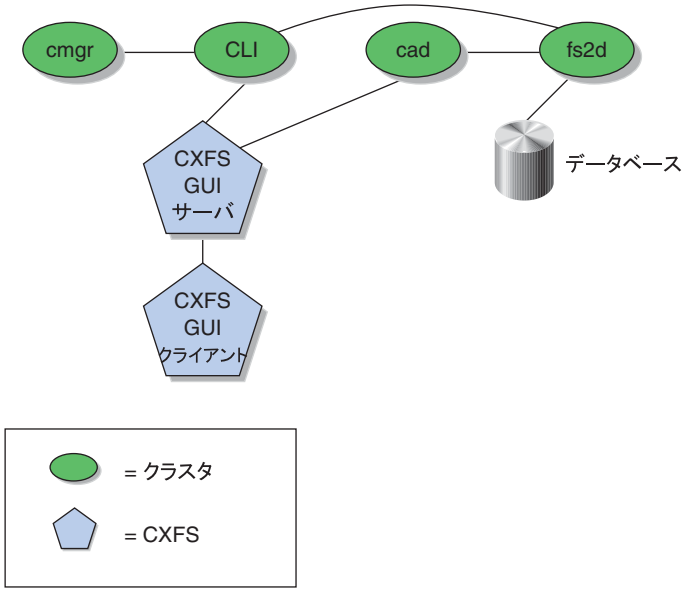
図A-1 特定の管理ノード内の通信



図A-2 特定の管理ノード内のデーモン通信

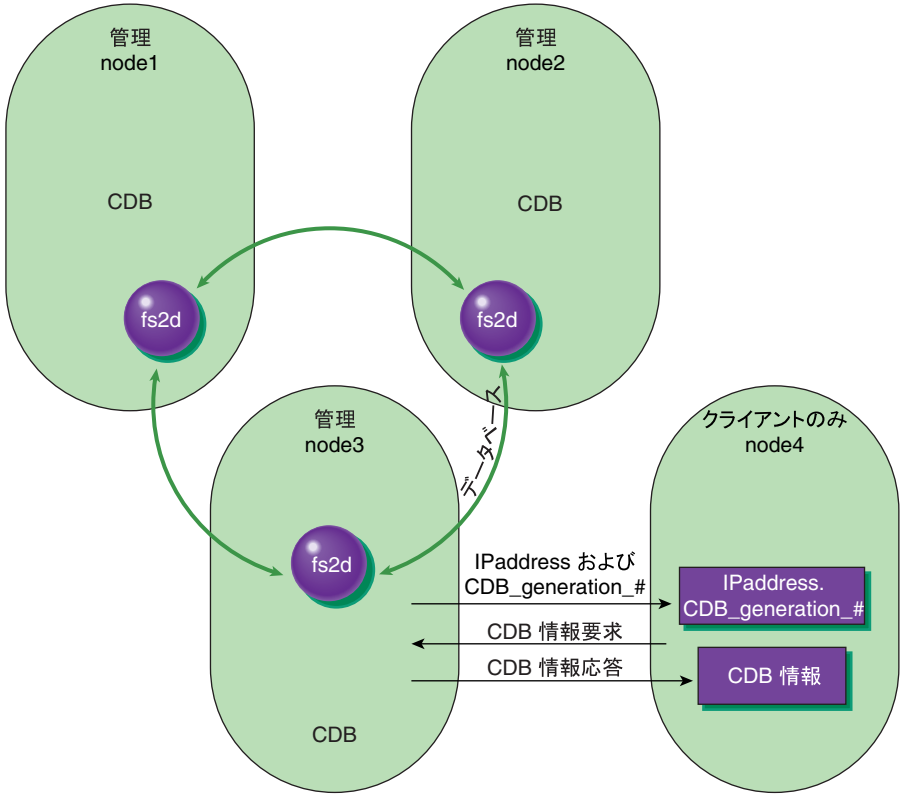


図A-3 プール内のノード間の通信



図A-4 クラスタ内にない管理ノードの通信

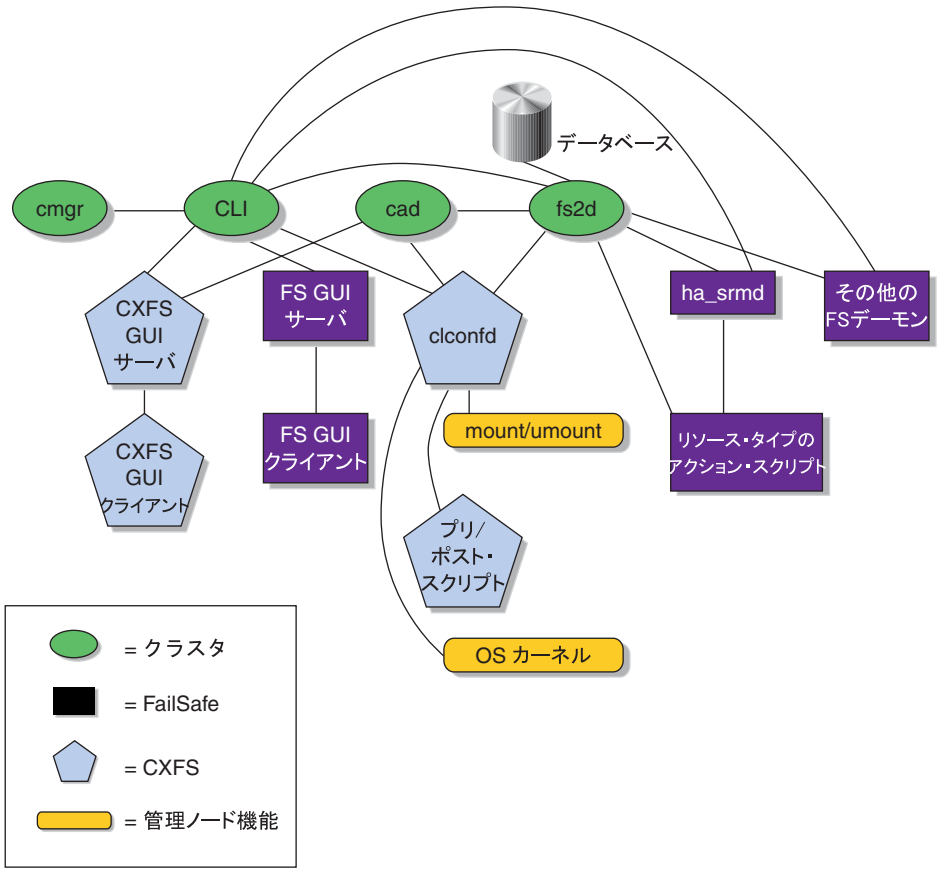
`fs2d` デーモンが実行されている管理ノードの 1 つが、その IP アドレスおよびクラスタ・データベースの生成番号を各クライアントのみのノードに定期的にマルチキャストするよう選択されています。データベースが変更されるたびに、新しい生成番号が作成され、マルチキャストされます。以下の図に、ノード間の通信について Solaris のクライアントのみのノードを例に説明します。



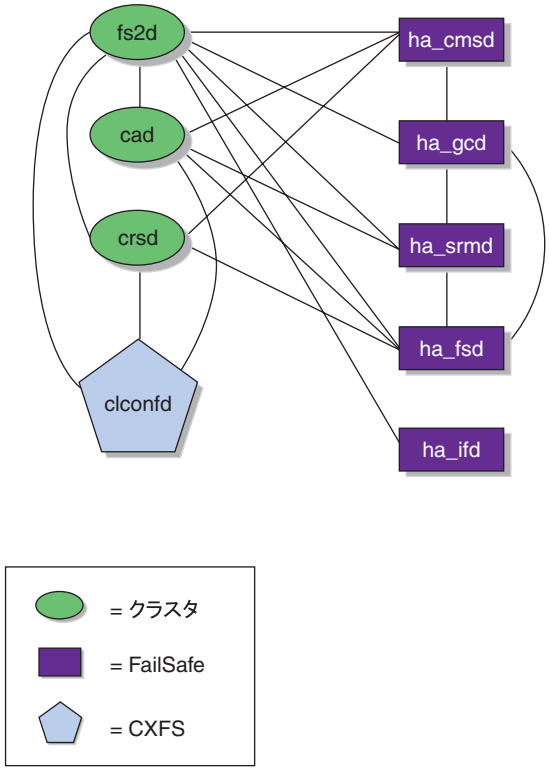
図A-5 管理ノードとクライアントのみのノード間の通信

同時実行クラスタ内の通信パス

以下の図に、CXFS と IRIS FailSafe が実行されている同時実行クラスタ内の 1 つのノードの通信パスを示します。



図A-6 同時実行時の特定の IRIX ノード内の管理通信

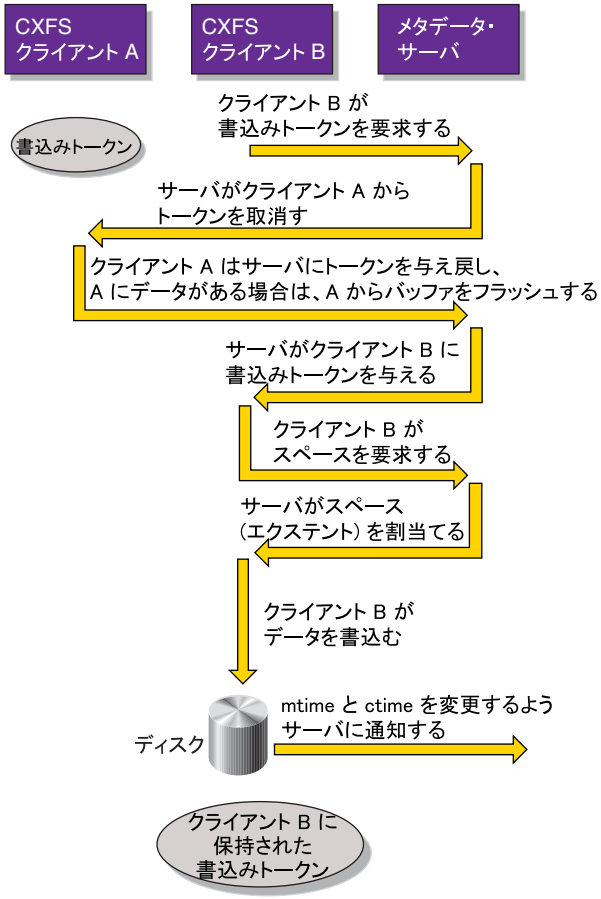


図A-7 同時実行時の特定の IRIX ノード内のデーモン通信

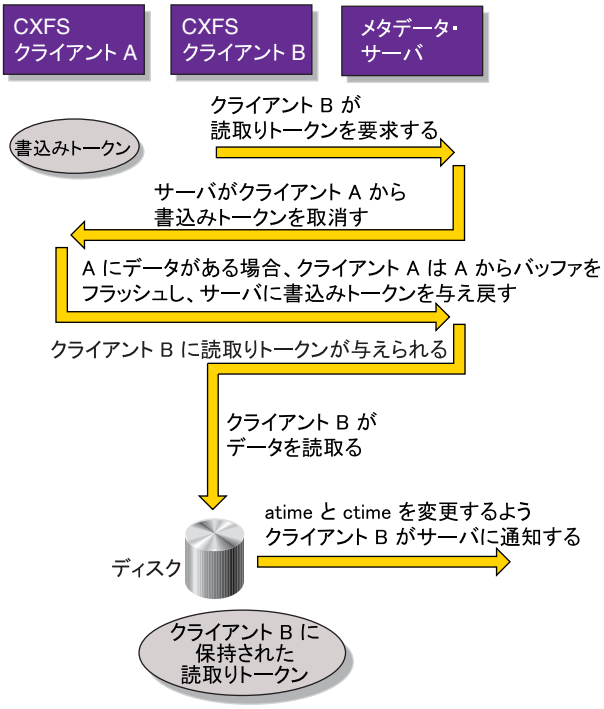
読取りおよび書込み時のメタデータの流れ

以下の図に、メタデータの流れの例を示します。

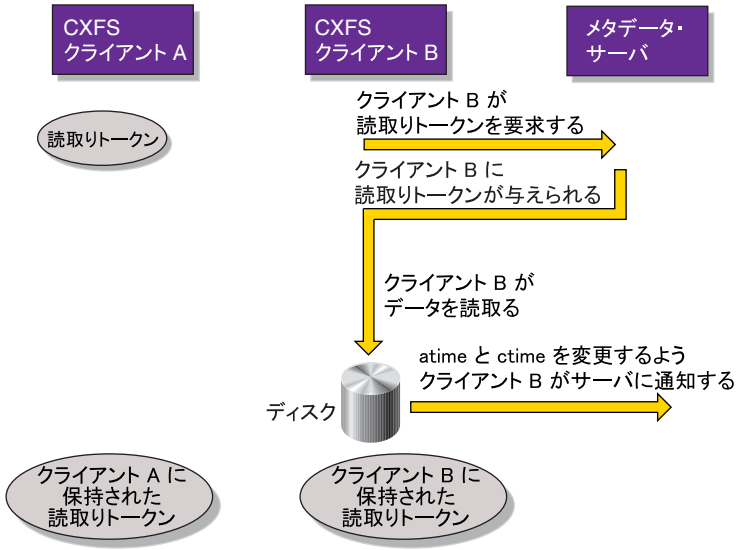
メモ:トークンはファイルを保護します。特定の時点で 1 つのファイルに対して複数の読取りトークンが存在する場合がありますが、書込みトークンは 1 つしか存在しません。



図A-8 書込み時のメタデータの流れ



図A-9 クライアント A での書込み後のクライアント B での読取り時のメタデータの流れ



図A-10 クライアント A での読取り後のクライアント B での読取り時のメタデータの流れ

CXFS プライベート・ネットワークの IP フィルタリングの例

この付録は、CXFS プライベート・ネットワークに IP フィルタリングを提供する場合に使用できる /etc/ipfilterd.conf ファイルの例で構成されます。

以下の点に注意してください。

- IP トラフィックをフィルタリングする各ノードに、設定済みの /etc/ipfilterd.conf ファイルが存在する必要があります。各ファイルは、ノードに依存する最初のグループを除いて同様になります。つまり、ファイル内の NodeA に対応する行は、NodeA に対するネットワーク・トラフィックが通過する可能性があるネットワーク・インタフェースに一致する必要があります。
- systune(1M) コマンドの変数 ipfilterd_inactive_behavior は、0 に設定する必要があります。これによって、killall(1M) コマンドを使用して ipfilterd を終了すると同時に、フィルタが無効になります。
- ipfilterd を実行する各ノードに対して、chkconfig(1M) の ipfilterd フラグをオンにする必要があります。例は、次のとおりです。

  ```
  nodeA# chkconfig ipfilterd on
  ```

- システムのいずれかのネットワーク・インタフェース名を変更した場合は、該当する accept の行に変更が含まれるよう、/etc/ipfilterd.conf ファイルを更新する必要があります。例は、次のとおりです。

  ```
  accept -i changed_or_new_interface
  ```

- デバッグのため、破棄された各パケットは、syslog ファイルに以下のようなメッセージを記録します。

```
May 24 16:44:44 5A:rodin unix: NOTICE: ipfilter(cache) - packet dropped:
10.1.1.5 SPT=137 DPT=137 UDP
```

必要なトラフィックがブロックされてしまう場合などにフィルタリングを無効にするときは、以下を実行します。

1. ipfilterd デーモンを強制終了します。

   ```
   nodeA# killall ipfilterd
   ```

2. ipfilterflag フラグをオフにします。

   ```
   nodeA# chkconfig ipfilterd off
   ```

以下に、NodeA 用のサンプル・ファイルを示します。

```
nodeA# cat ipfilterd.conf
#
# ipfilterd.conf for NodeA
#
#
# Filters follow:
#
# Do not restrict traffic on any of the interfaces for NodeA,
# except from ef1 (CXFS heartbeat)
#
accept -i lo0
accept -i ef0
accept -i eg0
accept -i eg1
accept -i lb0

#
# Restrict access over the CXFS heartbeat network
# Interface ef1
#

# Accept any fragment, reassembly won't work if first fragment filtered out.
accept -i ef1 ip.off>0

# CXFS is using RPC, need portmapper.
accept -i ef1 udp.port 111
accept -i ef1 tcp.port 111


# fs2d daemon is dynamically assigning ports in range 600-1023.
# We need port definition (sport + dport for both directions).
accept -i ef1 tcp.sport>=600 and tcp.sport<=1023
accept -i ef1 tcp.dport>=600 and tcp.dport<=1023


# sgi-cad defaults to 5435/tcp
accept -i ef1 tcp.port 5435

# sgi-crsd
# Each node opens 7500/udp, both directions needed
accept -i ef1 udp.port 7500
```

```
# Uncomment the line below for CXFS client-only node.
# accept -i ef1 udp.port 5449


# CXFS kernel ports 5450-5453
# Connections in both directions so open dport and sport.
accept -i ef1 tcp.port 5450
accept -i ef1 tcp.port 5451
accept -i ef1 udp.port 5452
accept -i ef1 udp.port 5453

# fs2d client are using ports in range 7000-8500
accept -i ef1 tcp.dport>7000
accept -i ef1 udp.dport>7000

# Uncomment the line below for IO fencing only if switches are on CXFS private network
#  (ip.src is the switch address)
# accept -i ef1 tcp.sport=23 and ip.src=10.1.1.6

# Let icmp traffic pass, especially 'PORT UNREACHABLE ICMP packet'
accept -i ef1 icmp

# Reject the rest (-l will log any rejected packet to the SYSLOG)
reject -i ef1 -l
```

初期設定チェックリスト

以下は、CXFS システムのインストールと設定の際に実行しなければならない手順のチェックリストです。

注意: CXFS は複雑な製品です。CXFS が最適にインストールおよび設定されるよう、必ず日本 SGI から初期セットアップ・サービスをご購入ください。

このチェックリストは、カスタマが直接使用することを目的としておらず、リファレンスとして提供されています。つまり、補足資料としての使用のみを目的としています。これらの手順を完了する前に、このマニュアル全体、特に 249 ページの 第10章「トラブルシューティング」をよくお読みください。

[ ] サイトでのアプリケーションの用途、ストレージの設定、および I/O パターンを理解します（一部のアプリケーションでは CXFS の効果はありません。2 ページの「XFS と CXFS の比較」を参照してください）。

[ ] SAN ハードウェアを接続します。以下のドキュメントを参照してください（日本 SGI の営業担当員のみ入手可能です）。

- 『SGI Storage Area Network Installation Instructions』
- 『SGI TP9400 RAID Installation and Upgrade Guide』
- 『SGI Total Performance 9100 Storage System Installation Instructions』
- 『IRIS FailSafe Version 2 Installation and Maintenance Instructions』（主にハードウェア設定情報が適用されます）

[ ] プライベート・ネットワークがあるかどうか。これは必要条件です。パフォーマンスおよび制御の点から、ハブではなくネットワーク・スイッチが推奨されます

[ ] シリアル・ハードウェア・リセット回線があるかどうか。データの整合性を保証するため、サーバとして使用可能なノードが 2 つだけのクラスタでは、リセット機能の使用が必須です。また、サーバとして使用可能なすべてのノードに対しては、リセット機能の使用を強くお薦めします。より大きなクラスタでは、サーバとして使用可能なノードを奇数個使用し、ノードの 2 つのみがサーバとして使用可能な場合は、シリアル・ハードウェア・リセット回線を使用するか、I/O 制限とスイッチを使用する必要があります。20 ページの「異常ノードの分離」および 24 ページの「シリアル・ハードウェア・リセット」を参照してください。

[ ] このマニュアルを読みます。IRIX 以外のオペレーティング・システムが実行されているクライアントがある場合は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』も読みます。

[ ] ネットワークが使用可能であることを確認します。43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。

[ ] CXFS ソフトウェアをインストールします。43 ページの 第2章「IRIX システム: CXFS ソフトウェアのインストールとシステムの準備」を参照してください。IRIX 以外のオペレーティング・システムが実行されているクライアントがある場合は、『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』も読みます。

[ ] 必要に応じて設定ファイルを変更します。 `/etc/sys_id` ファイルの内容が `/etc/hosts` ファイルのプライマリ名と一致することを確認します。51 ページの「IRIX システム・ファイルの設定」を参照してください。

[ ] その他のシステム設定手順を実行し、再起動します。以下を参照してください。

  [ ] 51 ページの「IRIX システム・ファイルの設定」

  [ ] 60 ページの「(オプション) 自動再起動用の設定」

  [ ] 60 ページの「ネットワーク・インタフェースの設定」

  [ ] 62 ページの「IRIX ノードのシリアル・ポートの設定」

  [ ] 63 ページの「システムの再起動」

[ ] 小さいクラスタ(3 ノード)を完全に設定します。67 ページの 第3章「クラスタの初期設定」を参照してください。

[ ] /var/cluster/ha/logs ディレクトリのデーモン・ログ・ファイル内でエラーを検索します。

[ ] すべてが正常な場合は、残りのノードを追加します。問題がある場合は、249 ページの 第10章「トラブルシューティング」を参照してください。

[ ] ファイルシステムを設定します。67 ページの 第3章「クラスタの初期設定」を参照してください。

用語集

CLI

CXFS マネージャ・グラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) および `cmgr`(1M) コマンドで使用される下層のコマンド・ライン・インタフェース・コマンド。

CXFS カーネル・メンバーシップ

クラスタでファイルシステムを共有できる CXFS ノードのグループ。クラスタ内で定義されているノードのサブセットの場合があります。ノードは起動処理中に CXFS カーネル・メンバーシップを要求し、要求が受付けられると、クラスタのファイルシステムを共有できます。カーネル空間メンバーシップとも呼びます。CXFS カーネル・メンバーシップは、クラスタ・データベース・メンバーシップおよび FailSafe メンバーシップとは異なります。FailSafe についての詳細は、『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。

CXFS 管理ノード

`cluster_admin.sw.base` ソフトウェア製品がインストールされるプール内のノードで、このノードでは、クラスタ管理タスクを実行したり、クラスタ・データベースのコピーを含むことができます。管理ノードには、サーバとして使用可能な管理ノードとクライアント管理ノードの 2 つのタイプがあります。

CXFS シャットダウン

CXFS ファイルシステムのメンバーシップが失われた場合に、ノードの CXFS カーネル・ベースのサービスを停止する異常時のアクション。正常なクラスタは、ノードにシャットダウンする余裕を与えるために回復の開始を遅らせます。

CXFS タイブレーカー・ノード

CXFS のタイブレーカーとして識別されるノード。クラスタ内のちょうど半数のノードが動作していて相互に通信できる場合にクラスタの CXFS カーネル・メンバーシップを計算する処理で使用されます。デフォルトの CXFS タイブレーカー・ノードはありません。CXFS タイブレーカーは、FailSafe タイブレーカーとは異なります。『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。

CXFS データベース

クラスタ・データベースを参照してください。

FailSafe メンバーシップ

クラスタでアクティブにリソースを共有しているノードのグループ。クラスタで定義されているノードのサブセットの場合があります。FailSafe メンバーシップは、CXFS カーネル・メンバーシップおよび クラスタ・データベース・メンバーシップとは異なります。FailSafe についての詳細は、『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。

fs2d データベース・メンバーシップ

クラスタ・データベース・メンバーシップを参照してください。

I/O 制限

問題のあるノードを分離する異常時のアクション。これにより、問題のあるノードが I/O デバイスにアクセスできなくなるため、共有 CXFS ファイルシステムのデータが破壊されることがなくなります。I/O 制限は、クラスタ内の任意のノード (CXFS クライアントおよびメタデータ・サーバ) に適用できます。クラスタの残りの部分は、直ちに回復を開始することができます。

SAN

ストレージ・エリア・ネットワーク (Storage Area Network)。サーバとストレージ・デバイスで構成される非常に高速でスケーラブルなネットワークで、ストレージ・リソースの統合、データ・アクセスと可用性の向上、およびストレージ管理の一元化を実現できます。

アクティブなメタデータ・サーバ

メタデータ・サーバとして使用可能なノードのリストから選択される、サーバとして使用可能な管理ノード。1 つのファイルシステムに対して 1 つのアクティブなメタデータ・サーバのみ使用できます。

異常時のアクション階層

異常ノードに対する処理を決定する命令の集合。2 番目の命令は最初の命令が失敗した場合にのみ実行され、3 番目の命令は、最初の命令と 2 番目の命令が失敗した場合にのみ実行されます。使用可能なアクションは、I/O 制限、リセット、およびシャットダウンです。

オーナー TTY 名

システム・コントローラが接続されている**オーナー・ホスト**の端末ポート (TTY) のデバイス・ファイル名。オーナー・ホストからノードをリモートで制御できるよう、このケーブルのもう一方はシステム・コントローラ・ポートを持つノードに接続します。

オーナー・ホスト

ノードの電源サイクルなどのノードの制御をリモートで行うことができるシステム。ランタイム時には、オーナー・ホストがプール内のノードとして定義されていなければなりません。

重み

メンバーシップ重みを参照してください。

カーネル空間メンバーシップ

CXFS カーネル・メンバーシップを参照してください。

回復

あるノード上のサービスが中断した場合に、最初のノードから別のノードにメタデータ・サーバが移動するプロセス。このリリースでの回復の機能は、スタンバイ・ノードでのみサポートされています。

管理ノード

CXFS 管理ノードを参照してください。

クライアント

CXFS クライアント・ノード、CXFS クライアントのみのノード、および CXFS 管理ノードを参照してください。

クライアント管理ノード

`cluster_admin` ソフトウェア製品がインストールされるノードで、このノードでは、クラスタ管理タスクを実行したり、クラスタ・データベースのコピーを含むことができますが、クラスタのアクティビティとメタデータを調整することはできません。

クライアントのみのノード

`cxfs_client.sw.base` ソフトウェア製品がインストールされるノードで、このノードでは、クラスタ管理デーモンは実行されず、クラスタのアクティビティやメタデータを調整することはできません。IRIX、Solaris、および Windows ノードは、クライアントのみのノードとして使用できます。CXFS 管理ノードおよびクライアント管理ノードも参照してください。

クラスタ

クラスタは、単一のコンピュータ・リソースとして連携動作するよう設定されたシステム（ノード）の集合です。クラスタは、単純な名前とクラスタ ID で識別されます。複数のオペレーティング・システムが実行されているクラスタは、マルチ OS クラスタと呼ばれます。

ノードの特定のプールから形成されるのは、1 つのクラスタだけです。

ディスクまたは論理ユニット (LUN: Logical Unit) は、クラスタの名前をディスク（または LUN）に記録することによってクラスタに割当てられます。したがって、複数のクラスタ内のマシンから（ファイバ・チャネル接続を通じて）任意のディスクにアクセスできるようにするには、このようなクラスタに一意な名前が必要です。クラスタのメンバーからメッセージが相互に送信される際、クラスタ ID によってそのクラスタが識別されます。したがって、2 つのクラスタで通信用に同じネットワークを共有するようにする場合は、クラスタに一意なクラスタ ID が必要です。マルチ OS クラスタに対しては、クラスタがネットワークを共有する場合、名前と ID の両方が一意でなければなりません。

クラスタ名およびクラスタ ID の上記の制約と、クラスタが作成されたらクラスタ名とクラスタ ID を（そのクラスタを削除して再作成しないかぎり）変更できないという理由により、構成内の各クラスタに対しては一意の名前とクラスタ ID を選択することが推奨されます。

クラスタ ID

ネットワーク内で一意な 1～128 の範囲の数字。クラスタ ID は、ネットワーク上に存在する可能性のあるほかのクラスタからクラスタ情報を受付けないようにするために、オペレーティング・システム・カーネルによって使用されます。カーネルは通信にデータベースを使用しないため、クラスタ通信を確認するためにクラスタ ID が必要です。カーネルのこの情報は、いったん初期化したら変更できません。このため、クラスタを定義したらクラスタ ID は変更しないでください。ネットワークを共有するクラスタには、一意の名前と ID を付ける必要があります。

クラスタ管理者

クラスタの管理とメインテナンスの担当者。

クラスタ・データベース

すべてのノードとクラスタに関する設定情報が含まれます。このデータベースは、クラスタ管理デーモンによって管理されます。

クラスタ・データベース・メンバーシップ

クラスタ管理デーモンにアクセス可能で、クラスタ・データベースの更新を受信できるプール内の管理ノードのグループ。プール内で定義されたノードのサブセットの場合があります。クラスタ管理デーモンは、プール内のすべての管理ノードへのクラスタ・データベース (CDB: Cluster Database) の配布を管理します。ユーザ空間メンバーシップおよび fs2d データベース・メンバーシップとも呼びます。

クラスタ・ドメイン

ファイルシステムがローカル・ノードだけでなくクラスタ全体に適用されるという XVM での概念。ローカル・ドメインも参照してください。

クラスタ・ノード

クラスタの一部として定義されているノード。**ノード**も参照してください。

クラスタ・モード

CXFS クラスタを運用する 2 つの方法のうちの 1 つで、通常(`Normal`) または実験中 (`Experimental`) のいずれかです。通常 (`Normal`) モードでは、CXFS は、ハートビート異常を検出したノードをリセットします。実験中 (`Experimental`) モードでは、CXFS はハートビート異常を無視します。実験中 (`Experimental`) モードを使用すると、ノード異常を発生させずにカーネル・デバッガを使用できます（ほかのモードでは、カーネル・デバッガを使用するとハートビートが停止します）。実験中 (`Experimental`) モードは、デバッグする場合にのみ使用することをお薦めします。

コントロール・ネットワーク

ネットワーク・インタフェース（通常は Ethernet）を通じてノードを接続するネットワーク。CXFS が、接続されているノードにネットワークを通じてハートビート・メッセージとコントロール・メッセージを送信できるようにします。CXFS では、コントロール・ネットワーク上で最も優先度が高いネットワーク・インタフェースが使用されます。コントロール・ネットワーク上にある優先度の高いすべてのネットワーク・インタフェースに異常が発生した場合は、優先度の低いネットワーク・インタフェースが使用されます。

コントロール・メッセージ

クラスタ・ノードに対する操作を要求したり、クラスタ・ノードに関する情報を配布するためにクラスタ・ソフトウェアがクラスタ・ノード間で送信するメッセージ。コントロール・メッセージおよびハートビート・メッセージは、コントロール・ネットワークに接続されているノードのネットワーク・インタフェースを通じて送信されます。

ノードが専用の CXFS ノードではない場合、ノードのコントロール・ネットワークは、コントロール・メッセージを受付けないように設定してください。コントロール・メッセージを受付けるように設定すると、CXFS がノードをリセットしたときに、そのマシンでほかのジョブを実行するエンド・ユーザのジョブが予期せず強制終了される可能性があります。

サーバとして使用可能な管理ノード

`cluster_admin` 製品がインストールされ、クラスタのアクティビティとメタデータを調整することもできるノード。

再設定

管理操作のために、あるノードから別のノードにメタデータ・サーバを移動するプロセス。最初のノード上のその他のサービスが中断されることはありません。再設定は、このリリースでは無効になっています。

システム・コントローラ・ポート

リモートで IRIX ノードの電源サイクルを行う方法を提供するノード上のポート。クラスタ・データベースでシステム・コントローラ・ポートが有効または無効のどちらに設定されているかによって、システム・コントローラ・ポートで CXFS が処理を実行できるかどうかが決まります。システム・コントローラ・ポートの情報はプール内のノードの場合はオプションですが、そのノードがクラスタに追加される場合は必須です。システム・コントローラ・ポートの情報がないと、そのノードで実行されているリソースは高可用性になりません。

シャットダウン

CXFS シャットダウンを参照してください。

定足数

クラスタを形成するために必要なノードの数のことで、メンバーシップに応じて以下のように変わります。

- CXFS カーネル・メンバーシップの場合
  - 初期メンバーシップを形成するには、クラスタ内のサーバとして使用可能なノードの過半数 **(>50%)** が必要です。
  - 既存のメンバーシップを維持するには、クラスタ内のサーバとして使用可能なノードの半数 **(50%)** が必要です。
- クラスタ・データベース・メンバーシップの場合、クラスタの形成と維持には、プール内のノードの **50%** が必要です。

詳細領域

表示領域で選択したコンポーネントの詳細が表示される GUI ウィンドウの部分。表示領域も参照してください。

シリアル・ハードウェア・リセット

システム・コントローラに接続されているシリアル回線を通じてシステム・リセットを実行する異常時のアクション。この異常時のアクション階層の選択肢は、IRIX ノードにのみ適用されます。

スタンバイ・ノード

特定のファイルシステムのメタデータ・サーバとして使用可能なノードとして設定されているが、そのファイルシステムを使用するアプリケーションを現在は実行していない、サーバとして使用可能な管理ノード。

スプリットブレイン状態

ネットワーク分割が発生したり、シリアル・ハードウェア・リセットまたは CXFS タイブレーカー機能、あるいはその両方がないために、複数のクラスタが形成される状態。

制限

I/O 制限を参照してください。

制限の回復

制限からの回復の処理。影響を受けるノードが自動的に CXFS カーネル・メンバーシップから取消され、制限された HBA を経由する I/O パスを使用しているすべてのファイルシステムがアンマウントされた後、ノードがクラスタに再設定されます。

セル ID

ノードに関連付けられている数字。CXFS ソフトウェアによって使用され、メッセージに表示されます。

タイブレーカー・ノード

CXFS タイブレーカー・ノードを参照してください。

通知コマンド

クラスタおよびノードの変更や異常をクラスタ管理者に通知するために使用されるコマンド。このコマンドは、クラスタ内のすべてのノードに存在する必要があります。

データベース

クラスタ・データベースを参照してください。

データベース・メンバーシップ

クラスタ・データベース・メンバーシップを参照してください。

同時実行

CXFS と IRIS FailSafe を同時に実行する機能。詳細については、39 ページの「IRIS FailSafe 同時実行の概要」を参照してください。

ドメイン

クラスタ・ドメインおよびローカル・ドメインを参照してください。

なりすまし

ネットワーク上の特定のマシンを別のマシンに見せかけるセキュリティ侵害。

ノード

ノードはオペレーティング・システム (OS: Operating System) の 1 個のイメージで、通常は個々のコンピュータです。この意味での**ノード**という用語は、SGI Origin 3000 または SGI 2000 システムのノードとは異なる意味を持ちます。

特定のノードは、1 つだけのプール、つまり 1 つだけのクラスタのメンバーになることができます。

『CXFS MultiOS for CXFS Client-Only Nodes: Installation and Configuration Guide』で定義されているように、ノードでは、IRIX オペレーティング・システム、または Solaris などの別のオペレーティング・システムを実行できます。

CXFS 管理ノード、クライアント管理ノード、クライアントのみのノード、サーバとして使用可能な管理ノード、およびスタンバイ・ノードも参照してください。

ノード ID

プール内のノードで一意の 1～32767 の範囲の整数。数字を指定しないと、CXFS によって自動的に ID が計算されます。IRIX ノード ID のデフォルトは、マシンのシリアル・ナンバーなどのマシン固有の情報を基にした 5 桁の数字で、連続する数字ではありません。IRIX 以外のオペレーティング・システムが実行されているノードに対しては、ノード ID を指定する必要があります。ノードを定義した後にノード ID 番号を変更しないでください。

ノード・タイムアウト

この時間内にノードからハートビートが受信されない場合、ノードはダウンしていると見なされます。CXFS を正しく動作させるには、ノード・タイムアウト値をハートビート周期の少なくとも 10 倍に設定してください。

ノード・メンバーシップ

クラスタでアクティブな（CXFS カーネル・メンバーシップを持っている）ノードのリスト。

ハートビート周期

ハートビート・メッセージの周期。CXFS を正しく動作させるには、ノード・タイムアウト値をハートビート周期の少なくとも 10 倍に設定してください。ハートビートの数が多いほど（ハートビート周期が小さいほど）、ネットワークの速度が遅くなる可能性が高くなります。

ハートビート・メッセージ

クラスタ・ソフトウェアがノード間で送信する、ノードが動作中であることを示すメッセージ。コントロール・メッセージおよび**ハートビート・メッセージ**は、コントロール・ネットワークに接続されているノードのネットワーク・インタフェースを通じて送信されます。

パケットの覗き見

不正なデータ傍受を含むセキュリティ侵害。

表示領域

コンポーネントがグラフィカルに表示される GUI ウィンドウの部分。詳細領域も参照してください。

プール

プールは、特定のクラスタを形成できるノードの集合です。特定のプールから設定できるのは 1 つのクラスタのみです。使用可能なすべてのノードをクラスタに含める必要はありません。ほかのプールを存在させることはできますが、各プールはお互いから分離した状態になり、ノードやクラスタ定義は共有されません。

プールは、特定のノードに接続し、CXFS GUI または `cmgr`(1M) コマンドを使用してクラスタ・データベース内でそのノードを定義すると形成されます。その後、最初のノードに接続されたままの状態でその他のノードを定義することにより、それらのノードをプールまたはプール内にすでにあるその他のノードに追加できます。別のノードに接続して定義を行うと、別のプールが作成されてしまいます。

ポート・パスワード

システム・コントローラ・ポートのパスワード。通常は、ファームウェアまたはジャンパ・ワイヤで一度だけ設定します。ポート・パスワードは、ノードの root パスワードとは異なります。

マルチ OS

IRIX や Solaris などの複数のオペレーティング・システムが実行されているクラスタ。

メタデータ

ファイルの名前、サイズ、場所、アクセス権など、ファイルを記述する情報。

メタデータ・サーバ

クラスタ内のすべてのノードに代わってメタデータの更新を調整する管理ノード。メタデータ・サーバとして使用可能なノードが複数存在することは可能ですが、1 つのファイルシステムに対してアクティブなメタデータ・サーバとして選択されるのは 1 つだけです。

メタデータ・サーバとして使用可能なノード

ファイルシステムを定義するときにメタデータ・サーバのリストに表示される、サーバとして使用可能な管理ノード。このリスト内のノードのいずれか 1 つが、アクティブなメタデータ・サーバとして選択されます。

メンバーシップ

クラスタ・データベース・メンバーシップおよび CXFS カーネル・メンバーシップを参照してください。

メンバーシップ重み

以前のリリースにおいて、CXFS カーネル・メンバーシップ定足数を計算するためにノードに割当てられる数字(通常は 0 または 1)。1 は、そのノードが、メタデータ・サーバとして使用可能なノードになることができることを示します。ほとんどの場合、メンバーシップ重みは設定されなくなっており、ノードの機能の定義に置き換えられています。

メンバーシップ・バージョン

ノードのセル ID に関連付けられている数字。ノードがメンバーシップに設定されてから CXFS カーネル・メンバーシップが変更された回数を示します。

ユーザ空間メンバーシップ

クラスタ・データベース・メンバーシップを参照してください。

リセット

シリアル・ハードウェア・リセットを参照してください。

ローカル・ドメイン

ファイルシステムがクラスタではなくローカル・ノードだけに適用されるという XVM での概念。クラスタ・ドメインも参照してください。

ログ・グループ

同じログ設定を使用する 1 つまたは複数の CXFS プロセスのセット。通常、ログ・グループは `gcd` などの 1 つのデーモンに対応します。

ログ設定

ログ設定は、**ログ・レベル**と**ログ・ファイル**の 2 つの部分で構成され、どちらも**ログ・グループ**に関連付けられます。クラスタ管理者は、ログ出力の場所と量をカスタマイズしたり、すべてのノードまたは 1 つのノードだけを対象にしたログ設定を指定できます。たとえば、`crsd` ログ・グループを設定して、ノード `foo` だ

けについては詳細レベル 10 のメッセージを `crsd-foo` ログに記録し、その他のすべてのノードについては最小レベル 1 のメッセージだけを `crsd` ログに書込むことができます。

ログ・ファイル

特定の**ログ・グループ**に関する通知が含まれるファイル。ログ・ファイルは、ログ・グループの**ログ設定**の一部です。

ログ・レベル

関連付けられているログ・グループのログ・ファイルに CXFS が書込むログ・メッセージの数を制御する数字。ログ・レベルは、ログ・グループのログ設定の一部です。

索引

数字

100baseT 37
2 ノード・クラスタの回復 293
6.5.12f 以前からのアップグレード 66
6.5.12f 以前のファイルシステム定義の切替え 66
6.5.12f 以前のファイルシステムの切替え 66
6.5.12f と 6.5.13f 間でのファイルシステム形式の変更 66
64 ビットのスケーラビリティ 3

A

ACL 4
add nic 149
admin コマンド (cmgr) 146
age 260
aliases 62
Apache サーバ 47
atime 306
autoconfig 63
AutoLoad 起動パラメータ 60

B

B ツリー 4
Brocade Web Tools V2.0 264
Brocade スイッチの GUI 264
Brocade ファイバ・チャネル・スイッチ 20
  定義
    cmgr 190
    GUI 129
BSD インタフェース 7
`bufview`(1) 254, 261
build_cmgr_script コマンド 197
bulkstat 205

C

cad
  options ファイル 54
  実行されていることの確認 71
  プロセス 55, 71, 298
  メッセージ 287
cad_log 256
CDB 10
`cdbreinit` 251, 267, 292, 295
cdbutil 261
cfs 255
CFS ファイルシステムのエクスポート 35
chkconfig フラグ 222, 311
chkconfig フラグが on であることの確認 72
`chkconfig`(1M) 72
clconf_info 82, 208, 258–260
clconfd
  エラー 280
  プロセス 71, 298
clconfd_*hostname* 256
clconfd-scripts ディレクトリ 203
`cli`
  エラー 279
  メッセージ 288
cli_*hostname* 256
CLI（下層のコマンド・ライン・インタフェース） 40
cluster chkconfig フラグ 222
cluster_admin サブシステム 298
cluster_control サブシステム 298
cluster_mgr 40
cluster_services サブシステム 298
cluster_status 208, 239
`cluster_status`(1M) 259
cmgr
  設定タスクを参照
  -c オプション 146
  概要 40, 142

コマンド・ライン実行　146
シェル　146
自動スクリプト生成　196
終了　143
初期設定　77
スクリプト　144
スクリプト・モード　194
テンプレート・ファイル　147
ヘルプ　142
cmgr でのシェルの使用　146
cmgr の終了　87
cmgr のスクリプト生成　196
cmgr のプロンプト・モード　142
cmgr-create-cluster　147
cmgr-create-node　147
cmgr-templates ディレクトリ　147
cmond
エラー　283
実行されていることの確認　71
プロセス　71, 298
cmond_log　256
cms_dead()　255
cms_declare_membership()　255
cms_follower()　255
cms_leader()　255
cms_nascent()　255
cms_reset_timeout　223
cms_reset_timeout_action　223
cms_shutdown()　255
cms_stable()　255
cmsd　298
cmsd カーネル・スレッド　262
corpseleader プロセス　298
crontab　68, 70, 205, 265
crsd
実行されていることの確認　71
プロセス　71, 298
crsd エラー　283, 289
crsd_*hostname*　256
ctime　306
curses インタフェース　239
cxdetail (cxfsmgr)　92
CXFS カーネル・メンバーシップ
許可と取消し
cmgr　176
現在の情報　260
定足数　28
CXFS カーネル・メンバーシップ許可と取消し
GUI　128
CXFS カーネル・メンバーシップの許可
cmgr　177
GUI　129
CXFS カーネル・メンバーシップの取消し
cmgr　176
GUI　128
CXFS クライアントのみのノード　11
CXFS クライアント・ノード　10
CXFS サービス
開始
cmgr　172
GUI　125
停止　64
cmgr　172
GUI　125
CXFS サービスのアクティブ化
cmgr　172
GUI　125
CXFS サービスの開始　212
cmgr　172
GUI　125
CXFS サービスの再開　291
CXFS サービスの停止
cmgr　172
GUI　125
CXFS サービスの非アクティブ化
cmgr　172
GUI　125
CXFS シャットダウン、強制　176
cmgr　173
GUI　128
CXFS シャットダウン、通常
cmgr　211

GUI 125
CXFS タイブレーカー・ノード 29
cmgr 173
GUI 126
CXFS でサポートされている XFS のファイルシステム機能 3
CXFS とは 1
CXFS の概念 8
CXFS のサービス
停止 64
CXFS ファイルシステムの制約 7
CXFS マネージャ
GUIを参照
設定タスク、cmgrを参照
CXFS メンバーシップ
状態の判断 255
問題 262
用語 16
CXFS を使用すべきケース 4
cxfs_client
プロセス 298
cxfs_client サブシステム 298
cxfs_cluster 72
cxfs_cluster chkconfig フラグ 222
cxfs_info 240
cxfs*mount.sh スクリプト 203
cxfs_relocation_ok 225
cxfs*umount.sh スクリプト 203
cxfsd プロセス 299
cxfsd_max 225
cxfsd_min 224
cxfsdump 256
cxfsfilesystemUpgrade 66
cxfslicense 44, 49
cxfsmgr 92, 251
cxtask (cxfsmgr) 92

D

Data Migration Facility 38, 205
dcshake プロセス 299
dcvn 255
/dev/cxvm directory 259
df(1) 254
diags_*hostname* 256
DMAPI 205
DMF 38, 205
DNS 53, 60–61, 72
dsvn 255

E

eag:mac-ip=label 236
Entitlement Sheet 36
/etc/aliases 62
/etc/chkconfig 72
/etc/config/cad.options ファイル 54
/etc/config/fs2d.options ファイル 55
/etc/config/netif.options 53, 61
/etc/config/routed.options 61
/etc/exports 59
/etc/hosts 52, 72
/etc/inetd.conf 270
/etc/init.d 295
/etc/init.d/cluster 251
/etc/init.d/cxfs stop 125
/etc/inittab 62
/etc/ipfilterd.conf 311
/etc/mtab 3
/etc/nsswitch.conf 53
/etc/services ファイル 54
/etc/sys_id 52, 73
Ethernet ネットワーク 35
exitop と別の root 45, 50
exports ファイル 59

F

FailSafe クラスタの切替え
cmgr 169
GUI 107, 123
FailSafe タスクからの切替え 107
FailSafe 同時実行
概要 39, 228
用語 38
FailSafe ノードの切替え

cmgr 161
GUI 107, 120
FailSafe メンバーシップ 16
FDDI 62
Filesystem Reorganizer 3
find と crontab 68, 70, 205
find(1) 266
FLEXlm ライセンス・キー 37
fs2d
options ファイル 55
エラー 284, 290
クラスタ・データベース・メンバーシップ定足数 32
実行されていることの確認 71
データベース・メンバーシップ 16
プロセス 71, 298
fs2d オプションのログ 55
fs2d メンバーシップ
CXFS カーネル・メンバーシップを参照
fs2d_log 256, 258, 263
fsr 205
fsr(1M) 266
fsr_xfs 3
fx(1M) 82

G

GRIO 7
GUI
Web からの呼出し 91
Web ベースのバージョン 47
概要 40, 94
起動 91
設定 73
タスク
設定タスクを参照
複数のインスタンス 92
GUI が実行されない 270
GUI と xvm コマンドの相違点 102
GUI を使った設定 73

H

hinv(1M) 250, 259
hosts ファイル 52
hsm 205

I

I/O オーバーヘッド 37
I/O 制限とデータの整合性 20
I/O 操作 4
I/O デバイス設定 250
I/O パフォーマンス解析 101
I/O パフォーマンスの解析 101
I/O レート保証 7
icrash 255, 262
icrash(1M) 259, 265
idbg 256
ifconfig 64
incarnation 260
inittab 62
ioconfig(1M) 250
IP アドレスとコントロール・ネットワーク 110
IP アドレスとホスト名のペア 75
IP アドレス・エラー 278
IP トラフィックのフィルタリング 311
IP フィルタリング 311
ipfilterd chkconfig フラグ 311
ipfilterd_inactive_behavior 311
IRISconsole 39
IRISconsole とシリアル・ポート設定 63
is_* コマンド 149

J

Java が有効な Web ブラウザによる GUI へのアクセス 40
Java のバージョン 48
Java ファイルとリリース・レベル 47
java_eoe* 47
java_plugin 48
JBOD 36

L

L1 150
L2 150
LAN 5

Legato NetWorker 217
license.dat 70
lofs ファイルシステム 204
LOFS ファイルシステムのアンマウント 204

M

MAC ラベルと Trusted IRIX 236
Membership delivered 256, 271
mesglist 255, 259
mesgtcpaccept プロセス 299
mesgtcpdiscovery 299
mesgtcpmulticast 299
mesgtcprcv プロセス 299
mkfs 2, 76
mkfs(1M) 83
mkpart 114, 150
MMSC 150
mount
  コマンド 2, 259
MSC 150
msenlow, minthigh 236
mtime 306

N

NetBackup 217
netif.options 53, 61
Netscape サーバ 47
Netscape と Brocade スイッチの GUI 264
netstat(1) 253
Network Information Service 53
NetWorker 217
newaliases 62
NFS 6, 217
NFS エクスポート 35, 59, 203
NFS と CXFS 5
NIS 53, 60–61, 72
nsadmin 53
nsd 53, 72
nss_enterprise.sw.server 47
nsswitch.conf 53
NT ノード 218
NUMAlink モジュール 112
NVRAM 変数 60

O

O2 36
Octane 36
Octane2 36
Onyx 300 112
Onyx 3000 シリーズ 36
Onyx 3200C 112
Onyx2 シリーズ 36
Origin 300 112
Origin 3000 シリーズ 36
Origin 3200C 112
Origin 3400 112
Origin 3800 112
OS レベル 38
osview(1) 254, 266

P

PCP 101
PCP XVM I/O モニタ 98
pcp_eoe 245
pcp_eoe.sw.xvm 245
Performance Co-Pilot (PCP) 255
  XVM 統計 48, 51, 245
physvol 276
ping(1) 64, 253
pmgxvm 98, 245
pmie(1) 255
pmieconf(1) 255
POSIX パス名 7

Q

Quota 3, 218

R

remove nic 149
Reorganizer 3
rhost データベース・エントリと Trusted IRIX 236

/.rhosts ファイル 58
root ファイルシステムと CXFS 7
rotatelogs 215
routed 61
routed.options 61
Router Discovery Advertisement 62
rsh 58

S

salog 256
Samba 6, 218
SAN
  使用 1
  ドキュメント 315
  ハードウェア 36
sar(1) 254, 261
SCSI バス・リセット 250
scsiha(1M) 250
services ファイル 54
set コマンド 149
SGI 2000 シリーズ 36
SGI 2100 112
SGI 2200 112
SGI 2400 112
SGI 2800 112
SGI Origin 200 36
SGI Origin 3000 シリーズ 36
sgi_apache.sw.server 47
sgi-cad 54
sgi-cmsd 54
sgi-crsd 54
sgi-gcd 54
show コマンド (cmgr) 146
Silicon Graphics O2 36
Silicon Graphics Octane 36
Silicon Graphics Octane2 36
Silicon Graphics Onyx 3000 シリーズ 36
Silicon Graphics Onyx2 シリーズ 36
sinfo 255
start コマンド (cmgr) 146
sthreads 255, 298
stop コマンド (cmgr) 146
sys_id 52, 73
sysadm_base* 46, 51
sysadm_cluster* 46, 51
sysadm_cxfs* 46, 51
sysadm_cxfs.sw.desktop 90
sysadmdesktop 47, 91
sysctrl* コマンド 149
SYSLOG 95, 219, 256, 258–259, 261
SYSLOG エラー 278
System V インタフェース 7
systune 222

T

Tape Management Facility (TMF) 38
tcp_channels 255
TCP/IP ネットワーク 35
TCP/IP ネットワークの必要条件 37
tcpmux 270
tcpmux サービス 47
tcpmux/sgi_sysadm 270
telnet ポートと I/O 制限 21
test コマンド (cmgr) 146
TMF 38
/tmp/cmgr_create_cluster<cluster_PID> 197
Toolchest 90
TRIX (Trusted IRIX) 38
Trusted IRIX 38–39

U

uname 44, 49
Unified Name Service 53
UNS 53
/usr/bin/icrash 255
/usr/cluster/bin/clconf_info 82
/usr/cluster/bin/cxfslicense 44, 49
/usr/lib/aliases 62

V

v ノード 255
/var/adm/SYSLOG 95, 256, 258–259, 261

/var/cluster/clconfd-scripts ディレクトリ　203
/var/cluster/cmgr-scripts/rotatelogs　215
/var/cluster/cmgr-templates　147
/var/cluster/ha/log/cad_log　256
/var/cluster/ha/log/clconfd_*hostname*　256
/var/cluster/ha/log/cli_*hostname*　256
/var/cluster/ha/log/cmond_log　256
/var/cluster/ha/log/crsd_*hostname*　256
/var/cluster/ha/log/diags_*hostname*　256
/var/cluster/ha/log/fs2d_log　256, 258
/var/flexlm/license.dat　70
/var/sysadm/salog　256
/var/xfsdump/inventory　220
VERITAS NetBackup　217

W

Web ブラウザによる GUI へのアクセス　40
Web ベースのバージョンの GUI　47
Windows ノード　218
WWNN　21

X

X/Open Data Storage Management Specification　205
XFS
　CXFS に対する比較　2
　Quota　218
　サポートされている機能　3
XFS の /etc/fstab（CXFS ではない）　2
XFS の fstab（CXFS ではない）　2
xfs_fsr　3
xfs_repair　68
`xfs_repair` の適切な使用　268
xfsd　297
xfsdump と xfsrestore　220
XSDM　205
xthreads　299
XVM　216
　統計　245
　必要条件　38
　論理ボリューム作成　83
XVM I/O モニタ　98
XVM ショートカット　97
XVM の I/O モニタ　98
XVM ボリューム　250
XVM ボリュームの設定　250
XVM ボリュームの統計　254
`xvm`(1M)　250, 254, 258–259

あ

アイコンと状態　103
アイテム表示
　詳細領域を参照
青いテキスト　96
空きディスク・ブロック　254
アクセス・コントロール・リスト　4
「アクティブ」クラスタのステータス　238
アクティブなメタデータ・サーバ　10
アップグレード
　既存のノードのクライアントのみのノードへの切替え　202
アンマウント　219
アンマウント（強制）
　cmgr　179
　GUI　138
　推奨　69
アンマウント・スクリプト　203

い

一時名　97
インストール　43
インストールされるサブシステム　46, 51
インベントリ・ファイル　220

え

エクステントの追跡　264
「エラー」クラスタのステータス　239

お

オペレーティング・システム・アクティビティ・データ　254
重み　152
重み付けされたノード　10
オーナー・ホスト　27

か

開始　262
階層ストレージ管理　205
回復
　機能　216
　ファイルシステムと XFS　3
　メタデータ・サーバの　209
　問題　33
　用語　18
回復プロセス　298
書込みとメタデータの流れ　306
書込みトークン　265
カラーと状態　103
管理　201
管理者へのクラスタ変更の通知
　cmgr　166
　GUI　123
管理ツール　40
管理デーモン　54
管理ノード　10, 110
管理メンバーシップ　16
カーネル空間メンバーシップ　16
カーネル生成　63
カーネル調整可能パラメータ　222
カーネルのステータス・ツール　255
カーネル・スレッド　297
カーネル・メンバーシップの状態の判断　255

き

既存のノードのクライアントのみのノードへの切替え　202
既存のノードのクライアントのみのノードへの変更　202
強制 CXFS シャットダウン
　cmgr　213
　GUI　125
強制アクセス・コントロール・ラベルと Trusted IRIX　236
強制アンマウント
　cmgr　179
　GUI　138
　推奨　69
強制シャットダウン　27, 213
強制シャットダウンと再開　214, 267
強制レコード・ロック　7

く

偶数個のノードで構成されるクラスタの問題　33
クライアント v ノード　255
クライアント/サーバの表示　255
クライアント管理ノード　10, 110, 149
クライアントのみのノード　11, 110, 150
　ステータスの収集　240
クラスタ ID
　指定
　　cmgr　167
　　GUI　124
　変更　266
クラスタからのノードの削除
　cmgr　168
　GUI　117
クラスタからのノードの除去
　cmgr　168
　GUI　117
クラスタ管理デーモン
　options ファイル　54
クラスタ切替え
　cmgr　169
　GUI　123
クラスタ削除
　cmgr　170
　GUI　124
クラスタ設定情報の収集　256
クラスタ定義
　cmgr　166
　GUI　122
クラスタの FailSafe 切替え
　cmgr　169
　GUI　107, 123
クラスタのサイズ　38
クラスタの再設定　262
クラスタの削除
　cmgr　170
　GUI　124
クラスタの作成

cmgr 166
GUI 122
クラスタの除去
cmgr 170
GUI 124
クラスタのステータス
確認 238
トラブルシューティングのツール 253
クラスタの定義
cmgr 166
GUI 122
クラスタの表示
cmgr 171
GUI 124
クラスタのモード 123
クラスタ表示
cmgr 171
GUI 124
クラスタへの再設定
強制 CXFS シャットダウン 214
通常 CXFS シャットダウン 212
クラスタ変更
cmgr 168
GUI 123
クラスタ(用語) 8
クラスタ・コンポーネントの表示 99
クラスタ・サイズ 39
クラスタ・タスク
cmgr 166
GUI 122
クラスタ・データベース
再作成 196
自動 cmgr スクリプト生成 196
シャットダウン 210
定足数 263
バックアップと復元 221
メンバーシップ定足数の安定性 263
用語 10
用語の定義 2
クラスタ・データベースの再作成 295
クラスタ・データベース・メンバーシップ 16
クラスタ・デーモン開始 71
クラスタ・デーモンが実行されていることの確認 71
クラスタ・デーモンの開始 71
クラスタ・デーモンの再開 295
クラスタ・ドメイン 217
クラスタ・マネージャ・ツール 40
クラスタ・メンバーシップ 16
クラスタ・モード 123

け

結合 216
結合作成 97
「検索(Find)」テキスト・フィールド 99
検出、問題 259

こ

高速性 3
コマンド・ボタン 97
コマンド・ライン・インタフェース 40, 142
cmgrも参照
コントロール・ネットワーク 16, 110
コンポーネントの詳細の表示 99

さ

再開、クラスタ・デーモン 295
再起動 63, 292
再起動(回避) 215
再設定 17
サブネット 16
サブボリューム作成 98
サポートされているノードの数 38
サポートされているプラットフォーム 36
サーバ v ノード 255
サーバとして使用可能な管理ノード 110, 149
サーバとして使用可能なノード 10
サーバとして使用可能なノードを 2 つ含むクラスタの問題 33

し

システム
ステータス 237

ソフトウェアの通信パス　300
表示　36
ファイル　51
システム再起動　63
システム調整可能パラメータ　222
システムのテスト　63
システム容量　266
システム・アクティビティ　254
システム・アクティビティ・データ　254
システム・コントローラのステータス　244
システム・コントローラ・タイプ　150
システム・バッファ・キャッシュ・アクティビティ　254
システム・ハング　271
システム・ファイルの設定　51
システム・ログ・ファイル　95
「実行中」ノードの状態　241
自動再起動用の設定　60
自動的な cmgr スクリプトの構築　196
自動的なノードの再起動　60
シャットダウン
強制終了された CXFS クラスタ・データベース　213
クラスタ・データベース　210
再開　214, 267
通常 CXFS
cmgr　153
GUI　125
通常の CXFS データベース　211
終了操作と別の root　45, 50
準備手順　70
詳細領域　95
使用できる連続するバージョンの OS　38
初期クラスタ設定
cmgr　77
GUI　73
初期設定　67
初期設定チェックリスト　315
処理能力　37
ショートカット　97
シリアル接続
cmgr　165
GUI　121
シリアル・ハードウェア・リセット　20, 24, 37
取外し　267
シリアル・ハードウェア・リセット・サービス・デーモン　54
シリアル・ポート設定　62
シリアル・ポートの設定　62
シリアル・ポート・サーバ　44, 49
シリアル・ポート・マルチプレクサ　24, 39
シリアル・リセット接続テスト　64
新規クラスタの設定　74
診断タスク
cmgr　165

す

推奨事項　39, 67
スイッチ
Brocade の必須使用　20
照会　193
推奨事項　39
表示　192
スイッチ削除
cmgr　192
GUI　131
スイッチ定義
cmgr　190
GUI　129
スイッチの表示　192
スイッチ変更
cmgr　190
GUI　130
スイッチ・ポート情報
cmgr　192
GUI　131
スクリプト
cmgr　144
cmgr のテンプレート　147
pre/post-mount スクリプト　203
スクリプト・モードと cmgr　194
スケーラビリティ（64 ビット）　3
スタンバイ・ノード　11
ステータス
クラスタ　238

システム、概要　237
システム・コントローラ　244
ノード　241
ストライプ　216
ストライプ作成　98
ストレージ・エリア・ネットワーク　1
ストレージ・ツール　250
スプリットブレイン状態　29
スペース
必要条件　44, 49
スレッド　297

せ

制限
緩和（アクセスの有効化）
cmgr　191
GUI　131
強化（アクセスの無効化）
cmgr　191
GUI　131
制限とデータのセキュリティ　20
制限の強化
cmgr　191
GUI　131
接続性テスト
cmgr　165
GUI　121
接続性のテスト
cmgr　165
GUI　121
「接続性のテスト」タスク　260
設定ガイド・タスク　94
設定タスク
cmgr　141
CXFS サービス　172
cxfs タイブレーカー　173
FailSafe クラスタの切替え　169
異常時のアクション階層　152
管理者へのクラスタ変更の通知　166
クラスタ削除　170
クラスタ定義　166
クラスタ表示　171
クラスタ・タスク　166
スイッチの削除　192
スイッチの定義　190
スイッチ変更　190
スイッチ・ポート情報の更新　192
制限　152
制限緩和　191
制限強化　191
接続性テスト　165
デフォルト　147
ノード削除　162
ノード表示　163
ノード変更　156
ノード・タスク　148
ノード・リセット　160
ファイルシステム削除　189
ファイルシステム変更　185
ファイルシステム・マウントとアンマウント　183
メタデータ・サーバ再設定　189
メタデータ・サーバ定義　178
メンバーシップ許可と取消し　176
ログ設定　174
GUI
CXFS サービス開始と停止　125
cxfs タイブレーカー　126
FailSafe クラスタの切替え　123
FailSafe ノードの切替え　120
クラスタ削除　124
クラスタ定義　122
クラスタ表示　124
クラスタ変更　123
新規クラスタの設定　75
新規ファイルシステムの設定　76
スイッチ定義　129
スイッチの削除　131
スイッチ変更　130
スイッチ・ポート情報の更新　131
制限緩和　131, 148
制限強化　131
接続性テスト　121

ノード削除　121
ノード追加と削除　117
ノード定義　109
ノードの表示　121
ノード変更　118
ノード・リセット　117
ファイルシステム削除　139
ファイルシステム変更　136
ファイルシステム・マウントとアンマウント　137
メタデータ・サーバ再設定　140
メタデータ・サーバ定義　135
メンバーシップ許可と取消し　128
ログ設定　126
設定チェックリスト　315
設定の統一性　263
セル ID　260, 271

そ

ソフトウェアの必要条件　36
ゾーニング　20

た

帯域幅　3–4, 6
大規模なクラスタ　75
タイブレーカー・ノード　29
cmgr　173
GUI　126
ダイレクトツーディスク I/O　7
タスク　100
「タスク(Tasks)」メニュー　96
タスクの実行　100
単一のシステム表示　36
単一のファイルシステム表示　1
断片化解消プログラム　3

ち

チェックリスト　315
チャネル・トラフィック　274

つ

通常 CXFS シャットダウン　125
通信パス　300
ツリー表示
表示領域を参照
ツール　40

て

「停止」ノードの状態　242
ディスク管理　217
ディスク設定　102
ディスクのストライプ　216
ディスクのスライス　97
ディスクのバッファ　6
ディスクのラベル削除　98
ディスクのラベル付け　97
ディスク・バッファ　6
ディスク・ブロック、空き　254
定足数　263
適切な開始　262
デバイスの検査　250
電子メール・エイリアス　62
テンプレート・ファイル　147
データ管理 API　205
データベース
クリア　291
シャットダウン　210
ダンプ　261
メンバーシップ　16
データベースのクリア　291
データベースの再初期化　251
データベースのダンプ　261
データベースのリセット　291
データベース・メンバーシップ
クラスタ・データベース・メンバーシップを参照
データ・セキュリティ　20
データ・フロー　306
デーモン
開始　71
確認　71
再開　295

リスト　297

と

同時実行
　FailSafe 同時実行を参照
　通信パス　305
特権マネージャ　91
ドメイン　217
「ドメイン(Domain)」メニュー　97
ドメイン・ネーム・サービス　53
ドラッグアンドドロップ　100
トラブルシューティング　249
　2 ノード・クラスタの回復　293
　cad メッセージ　287
　clconfd エラー　280
　clconfd デーモンの異常終了　275
　cli エラー　279
　cli メッセージ　288
　cmond エラー　283
　crontab の使用　265
　crsd エラー　283, 289
　CXFS サービスの再開　291
　CXFS メンバーシップの取消し　277
　fs2d エラー　284, 290
　GUI が実行されない　270
　GUI の使用　264
　IP アドレス・エラー　278
　Netscape と Brocade スイッチの GUI　264
　SYSLOG エラー　278
　xfs_repair の適切な使用　268
　一般的な問題　270
　一般メッセージ　286
　過剰なファイルシステム・アクティビティの回避　265
　カーネルのステータス・ツール　255
　強制シャットダウン後の CXFS の再開　267
　クラスタ制御ツール　251
　クラスタ設定ツール　250
　クラスタのステータスの識別　258
　クラスタまたはノードのステータス・ツール　253
　クラスタ名 ID 欠如エラー　276
　クラスタ・データベースの再作成　295
　クラスタ・データベース・メンバーシップ定足数の安定性　263
　クラスタ・デーモンの再開　295
　再起動　292
　システムがハングする　271
　システム容量の効果的な使用　266
　修正処置　291
　使用されていないノードの削除　267
　シリアル・ハードウェア・リセット回線の取外し　267
　設定の統一性　263
　セル ID カウントと「Membership Delivered」メッセージ　271
　通常メッセージ　273
　ツール　249
　適切な開始　262
　データベースのクリア　291
　ネットワーク・ツール　253
　残っているクラスタの削除　262
　ノード ID またはクラスタ ID を変更する前の再起動　266
　ノードの機能の適切な定義　263
　ノードは検出されるが、メンバーシップに設定されない　271
　ノードをクラスタに再設定しない再起動　294
　ノードを定義できない　271
　排他書込みトークン　265
　パフォーマンス・モニタ・ツール　254
　ファイルシステムにアクセスできない　270
　物理ストレージ・ツール　250
　方法　249
　未書込みエクステントの追跡　264
　メッセージ　273
　問題の回避　261
　問題の検出　259
　ライセンス・エラー　277
　ログインできない　272
　ログ・ファイル　256
　ログ・ファイルがディスク容量を消費しすぎる　270
　ログ・ファイル名とサイズ　264
　ログ・ファイル・エラー　286
　論理スワップ・スペース不足　276
トランザクション・レート　3, 5
トレース、メッセージ　255
トークン　265, 306

な

名前付きパイプ　7

に

入力詳細　102
任意レコード・ロック　7

ね

ネットワーク　16
ネットワーク上の問題　67
ネットワークの接続性
　cmgr　165
　GUI　121
ネットワークの必要条件　37
ネットワークの分割とハードウェアのリセット　27
ネットワーク分割　26
ネットワーク・インタフェースの設定　60
ネットワーク・スイッチ　39
ネットワーク・ステータス　253
ネットワーク・セグメント　39
ネットワーク・ファイルシステムの比較　5
ネットワーク・ルーティング　61
ネーム・サービス・デーモン　53

の

残っているクラスタ　262
ノード
　状態　241
　ステータス　241
　タスク
　　設定タスクを参照
　分離　20
　用語　9
ノード ID、変更　266
ノード切替え
　cmgr　161
　GUI　120
ノード削除
　cmgr　162
　GUI　121
ノード修正
　GUI　118
ノード追加
　cmgr　148
　GUI　117
ノード定義
　cmgr　148
　GUI　109
ノードの FailSafe 切替え
　cmgr　161
　GUI　120
ノードの重み　152
ノードの機能　110, 149, 263
ノードの作成
　cmgr　148
　GUI　109
ノードの自動再起動　60
ノードの修正
　cmgr　156
　GUI　118
ノードのステータス
　強制 CXFS シャットダウン　213
　通常 CXFS シャットダウン　212
　データベースのシャットダウン　210
ノードのステータス・ツール　253
ノードの追加
　cmgr　148
　GUI　117
ノードの表示
　cmgr　163
　GUI　121
ノードの変更
　cmgr　156
　GUI　118
ノードのリセット
　cmgr　160
　GUI　117
ノード表示
　cmgr　163
　GUI　121
ノード変更

cmgr 156
ノードをクラスタに再設定しない再起動 294
ノード・タスク
cmgr 148
GUI 109
ノード・リセット
cmgr 160
GUI 117

は

排他書込みトークン 265
パイプ(名前付き) 7
バックアップ 217
バックアップ・ノード
スタンバイ・ノードを参照
バッファのコヒーレンシ 35
バッファ・キャッシュ・アクティビティ 254, 261
ハブ 39
パフォーマンス・モニタ・ツール 254
パラメータ(システム調整可能) 222
ハングしたシステム 271
パーティション 150
パーティションが設定されたシステムのライセンス 37
パーティション設定とライセンス 45, 50
ハードウェアの必要条件 36
ハードウェア・インベントリ 250, 259
ハードウェア・リセット
シリアル・ハードウェア・リセットを参照
ハートビートのタイミング 16
ハートビート・ネットワーク 16, 110

ひ

「非アクティブ」
クラスタのステータス 238
ノードの状態 242
ピアツーディスク・モデル 6
比較
XFS と CXFS 2
ネットワークと CXFS ファイルシステム 5
必要条件 36
必要なプライベート・ネットワーク 37
非同期バッファ技術 4
表示または変更するアイテムの選択 99
表示領域 95

ふ

ファイバ・チャネル・スイッチ 20
「ファイル(File)」メニュー 92, 95
ファイルシステムがアクセス不可能 270
ファイルシステム拡張 220
ファイルシステム削除
cmgr 189
GUI 139
ファイルシステムと IRIS FailSafe 233
ファイルシステムによるアクセスの制限 4
ファイルシステムによるローカルホスト・アクセス 4
ファイルシステムのアンマウント
cmgr 184
GUI 138–139
ファイルシステムの拡張 220
ファイルシステムの構造 2
ファイルシステムの削除
cmgr 189
GUI 139
ファイルシステムの作成 2
ファイルシステムの修正
cmgr 185
GUI 136
ファイルシステムの除去
cmgr 189
GUI 139
ファイルシステムの信頼性 3
ファイルシステムのネットワーク・アクセス 4
ファイルシステムの比較、ネットワークと CXFS 5
ファイルシステムの変更
cmgr 156, 185
GUI 136
ファイルシステムのマウント 83
ファイルシステムのメインテナンス 218
ファイルシステム表示 1
ファイルシステム変更
cmgr 185

GUI 136
ファイルシステム・アンマウント
cmgr 184
GUI 137
ファイルシステム・タスク
cmgr 177
GUI 132
設定ガイド 76
ファイルシステム・バッファ・キャッシュ・アクティビティ・モニタ 261
ファイルシステム・マウント
cmgr 168, 183
GUI 137
XFS の違い 2
概要 218
ファイルシステム・マウントとアンマウント
GUI 138–139
ファイルシステム・マネージャ 2
ファイルシステム・メタデータ・サーバ定義
cmgr 178
GUI 135
ファイルシステム・メタデータ・サーバの作成
cmgr 178
GUI 135
ファイルの応答時間 4
ファイルのロック 34
ファイルへのスワップ 7
復元 220
物理ストレージ・ツール 250
物理ネットワーク・セグメント 39
物理ボリューム、表示 250
「不明」クラスタのステータス 239
「不明」ノードの状態 242
プライベート・ネットワーク 16, 60
プライベート・ネットワーク・インタフェース・テスト 64
古いクラスタ 262
プール 9
プールからのノードの削除
cmgr 162
GUI 121
プールからのノードの除去
cmgr 162
GUI 121

へ

別の root 45, 50
別のログイン 47
ヘルプ
cmgr 142, 146
GUI 74
「ヘルプ(Help)」ボタン 40
「ヘルプ(Help)」メニュー 96
「編集(Edit)」メニュー 96

ほ

ホスト名 72
ホスト名解決 51
ホスト名コントロール・ネットワーク 110
ホスト名と IP アドレスのペア 75
ホスト名の判断 72
ホスト・バス・アダプタ 21
ボリューム管理 216
ボリューム作成 83, 98
ボリューム・エレメント削除 98
ボリューム・エレメントの分離 98
ボリューム・エレメント分離 98
ボリューム・トポロジー 101
ボリューム・ヘッダの作成 82
ボリューム・マネージャ 38
ポート設定 62

ま

マウント
オプション 135
ファイルシステム 218
cmgr 183
GUI 137
ポイント 135
マウントされているファイルシステムの表示 254
マウントされているファイルシステム、表示 254
マウント・スクリプト 203
マウント・ポイントのネスト 7

マウント・ポイント・ネスト　7
マスクとスイッチ定義　130
マルチ OS クラスタ　20
マルチプレクサ　24, 39

み

未書込みエクステントの追跡　264
見つからない XVM ボリューム　103
ミラー化　216
ミラー作成　97

め

命名の制約　52
メインテナンス　201
メタデータ
　トランザクションの例　5
　用語　2, 10
　ログ　34
メタデータの流れ　306
メタデータ・サーバ　28
　回復　209
　検出　206, 208
　再設定
　　cmgr　189
　　GUI　140
　順序　83
　使用可能なノードのリストからの選択　35
　定義
　　cmgr　178
　　GUI　135
　モデル　34
　用語　2, 10
メタデータ・サーバからのダンプ　220
メタデータ・サーバとして使用可能な管理ノード　10
メタデータ・サーバとして使用可能なノード　10
メタデータ・サーバの再設定
　cmgr　189
　GUI　140
メッセージ・トレース　255
メンバーシップ
　クラスタ・データベース・メンバーシップまたは CXFS カーネル・メンバーシップを参照
メンバーシップ重み　152

も

モニタ・ツール　254

ゆ

ユーザ空間メンバーシップ　16

よ

用語　8
用語定義　102
容量、システム　266
読取りとメタデータ
　流れ　306

ら

ライセンス
　FLEXlm キー　37
　GUI　270
　インストール　44, 49
　エラー　277
　確認　70
ライセンスの確認　70
ラベル削除、ディスク　98

り

リアルタイム・ファイルシステム　7
リセット
　シリアル・ハードウェア・リセットを参照
リセットのためのレース　27
リセット・レース　27
リモート・リセット
　接続　27
領域
　割当て　4
領域の連続的な割当て　4
領域の割当て　4

る

ルーティング　61

れ

例
admin ping　65
bufview　261
clconf_info　82
cmgr　146
cmgr スクリプト　144
cmgr の終了　87
CXFS サービスの開始　82
CXFS サービスの停止　64
CXFS マネージャ GUI　73
/etc/config/netif.options　61
/etc/config/routed.options　61
/etc/hosts　60
/etc/inittab　63
fs2d オプション　57
fs2d ログとトレース　57
GUI 画面　92
ifconfig　64
IP フィルタリング　311
LOFS ファイルシステムのアンマウント　204
ping 出力　64
.rhosts　58
sar　261
show clusters　146
show status　82
クラスタ定義　80
クラスタ内にないノードの通信　303
クラスタ表示　81
クラスタ・デーモンが実行されていることの確認　71
最小限の CXFS カーネルおよびクラスタ・データベース・メンバーシップの概念　32
再設定と回復　18
シリアル・ハードウェア・リセット　24
シリアル・ポート設定　63
シリアル・リセット接続のテスト　65
通信パス　300
定足数　30
適切にインストールされているライセンス　44, 49
同時実行時の特定の IRIX ノード内のデーモン通信　306
同時実行時の特定のノード内の管理通信　305
特定の管理ノード内のデーモン通信　301
特定のノード内の管理通信　300
ネットワーク分割　30
ネーム・サービス　60
ノード設定　77
ファイルシステムの定義　83
プライベート・ネットワーク・インタフェース・テスト　64
プールとクラスタの概念　9
プール内のノード間の通信　302
プール内のノードの表示　80
メタデータの流れ　307
メタデータ・サーバとして使用可能なノード　83
メタデータ・サーバの分散　12
レコード・ロック　7
レコード・ロック用ライブラリ　7
レコード・ロック・ライブラリ　7

ろ

ログインの問題　272
ログ設定
cmgr　174
GUI　126
ログベースのファイルシステム　3
ログ・ファイル　256
エラー　286
管理　215
ディスク容量を消費しすぎる　270
名前とサイズ　264
モニタ　237
リスト　237
ログ・ファイルのローテーション　215
ロック　7
論理ボリューム再設定　101
論理ボリューム作成　83
ローカル・エリア・ネットワーク (LAN: Local Area Network)　5
ローカル・ドメイン　217
ローカル・ノード　212

わ

ワールドワイド・ノード名　21