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

007–4016–011JP

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 François Barbou des Places、Ken Beck、Felix Blyakher、Laurie Costello、Dave Ellis、Brian Gaffey、Dean Jansa、Erik Jacobson、Dennis Kender、Chris Kirby、Ted Kline、Dan Knappe、Kent Koeninger、Linda Lait、Bob LaPreze、Steve Lord、Troy McCorkell、LaNet Merrill、Terry Merth、Nate Pearlstein、Alain Renaud、John Relph、Elaine Robinson、Dean Roehrich、Wesley Smith、Kerm Steffenhagen、Paddy Sreenivasan、Andy Tran、Connie Waring、Geoffrey Wehrman

COPYRIGHT

© 1999, 2001 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、IRIS、IRIX、O2、Octane、および Onyx2 は登録商標です。SGI、CXFS、FailSafe、FDDIXPress、IRIXconsole、IRIS FailSafe、Origin、Trusted IRIX、および XFS は Silicon Graphics, Inc. の商標です。

Adaptec は Adaptec, 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 は Sun Microsystems, Inc. の登録商標です。Linux は Linus Torvalds の登録商標です。Legato NetWorker は Legato Systems, Inc. の商標です。Netscape は Netscape Communications Corporation の商標です。QLogic は QLogic Corporation の商標です。Sun は Sun Microsystems, Inc. の商標です。VERITAS は VERITAS Software Corporation の商標です。Windows は Microsoft Corporation の商標です。

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

このガイドでの新機能

メモ: 再設定と回復の実装は延期されています。

この更新版には、以下の新しい情報が含まれています。

- グラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) が強化され、これまで分離されていたクラスタ表示（`cxdetail` コマンド）とタスク・マネージャ（`cxtask` コマンド）が、「CXFS マネージャ(CXFS Manager)」と呼ばれる 1 つのウィンドウに簡素化されました。`cxtask` コマンドと `cxdetail` コマンドは、従来の操作性を維持するために両方とも保持されています。このドキュメントでは、簡潔にするために `cxtask` だけを使用しています。

  新しい GUI には、以下の機能が用意されています。

  – メニュー・バーからタスク名を選択するか、ツリー表示内でマウスの右ボタンをクリックして、タスクにアクセスする

  – ファイルシステムとクラスタのステータスを高速に更新する

  – GUI から実行された各コマンドを表示する `salog`(4) ファイルにアクセスする

  – 「検索(Find)」テキスト・フィールドを使用して、表示されているツリー表示内でコンポーネントを検索する

  45 ページの「GUI の概要」を参照してください。

- `xfs_repair` および CXFS ファイルシステムの使用についての詳細は、209 ページの「`xfs_repair` の適切な使用」を参照してください。

注意: 問題があることが確実でないかぎり、CXFS ファイルシステムに対して `xfs_repair` を使用しないでください。

- `cmgr`(1M) の使用についての詳細:

  – `-c` オプションを使用した、コマンド・ラインでのサブコマンドの直接呼出し

  – `/var/cluster/cmgr-templates` ディレクトリ内にあるテンプレート・スクリプトの使用

  48 ページの「`cmgr`(1M) の概要」を参照してください。

- Trusted-IRIX (Trix) と IRIX が混在するクラスタにおける MAC ラベルについての詳細は、33 ページの「ハードウェアおよびソフトウェアの必要条件とサポート」を参照してください。
- IRIX 6.5.13 のリリースから CXFS ファイルシステム設定の構造が変更されています。IRIX 6.5.14 では、以前のバージョンとの逆互換性が維持されなくなりました。これは、クラスタ内のすべてのノードで同バージョンまたは直前のバージョンが実行されている必要があるためです。

  6.5.13f から 6.5.14f に完全にアップグレードする場合、影響はありません。

  6.5.13f ノードと 6.5.14f ノードを組合わせて実行する予定の場合は、逆互換性をオフにしてください。

  最初に 6.5.13f をインストールして実行せずに 6.5.12f 以前のバージョンからアップグレードする場合は、CXFS ファイルシステム定義の手動による切替えを一度だけ行う必要があります。

  73 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。

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
010	2001 年 9 月 IRIX 6.5.14f リリースで CXFS 製品をサポート

目次

このガイドについて xxiii
関連ドキュメント xxiii
出版物の入手方法 xxiv
表記規則 xxiv
ご意見とお問い合わせ先 xxv

1. CXFS とは 1
XFS と CXFS の比較 1
サポートされている XFS の機能 2
CXFS を使用すべきケース 3
XFS を使用すべきケース 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 4
ネットワーク・ファイルシステム 4
CXFS ファイルシステム 5
機能 5
制約 6
クラスタ環境 6
用語 7
ノード 7
クラスタ・データベース 7
プール 7
クラスタ 7
メンバーシップ 8
上限量 9
プライベート・ネットワーク 9
メタデータ 10
メタデータ・サーバとメタデータ・クライアント 10
再設定 13
回復 14
ハードウェア・コンポーネント 15

メンバーシップ上限量 17
CXFS メンバーシップ上限量 17
上限量の計算とノードの重み 17
上限量の変更の例 18
ネットワークの分割の例 19
fs2d データベース・メンバーシップ上限量 21
ハードウェア・リセット 22
メタデータ・クライアント/サーバ・モデル 23
システム表示 24
デーモン 24
通信パス 27
読取りおよび書込み時のメタデータの流れ 31
ハードウェアおよびソフトウェアの必要条件とサポート 33
CXFS と IRIS FailSafe の同時実行 35
FailSafe の CXFS リソース・タイプ 35
クラスタのタイプ 37
クラスタのサイズ 38
ノードのタイプ 38
GUI の違い 38
変換 38
ネットワーク・インタフェース 39
CXFS タイブレーカー・ノード 39
メタデータ・サーバとフェイルオーバー・ドメイン 39
同時実行クラスタ内の通信パス 40
重み付けされた 2 つのノードだけのクラスタでの回復と再設定の問題 41
クラスタ・マネージャのツール 42
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) 43
GUI の起動 43
GUI の概要 45

クラスタ・コンポーネントの表示 . . . 46
コンポーネントの詳細の表示 . . . 46
タスクの実行 . . . 46
画面 . . . 47
cmgr(1M) の概要 . . . 48
ヘルプの利用 . . . 49
プロンプト・モードの使用 . . . 49
アクションの完了とキャンセル . . . 50
スクリプト・ファイルの使用 . . . 50
cmgr 内部からのシェルの呼出し . . . 52
コマンド・ラインでのサブコマンドの入力 . . . 52
テンプレート・スクリプト . . . 53

2. CXFS ソフトウェアのインストールとシステムの準備 . . . 55

ソフトウェアのインストール . . . 56
システム・ファイルの設定 . . . 60
ホスト名解決: /etc/sys_id、/etc/hosts、/etc/nsswitch.conf . . . 60
/etc/services . . . 62
/etc/config/cad.options . . . 63
/etc/config/fs2d.options . . . 63
例 1 . . . 66
例 2 . . . 66
(オプション) /.rhosts . . . 66
(オプション) /etc/exports . . . 67
(オプション) 自動再起動用の設定 . . . 67
ネットワーク・インタフェースの設定 . . . 68
シリアル・ポートの設定 . . . 70
システムの再起動 . . . 71
システムのテスト . . . 71
プライベート・ネットワーク・インタフェース . . . 71

シリアル・リセット接続 72
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 73
6.5.12f 以前からのアップグレード 74
6.5.13f ノードと 6.5.14f ノードが混在する状態での実行 74
すべてのノードが 6.5.14f にアップグレードされた状態での実行 74

3. クラスタの初期設定 75

準備手順 75
ライセンスの確認 76
クラスタ・デーモンの開始 76
クラスタ・デーモンが実行されていることの確認 76
ノードのホスト名の判断 77
`chkconfig` フラグがオンであることの確認 78
GUI を使った設定 78
GUI の起動 78
GUI を使った新規クラスタの設定 80
GUI を使った新規ファイルシステムの設定 81
`cmgr`(1M) コマンドを使った設定 82

4. GUI リファレンス 93

設定ガイド・タスクセット 93
GUI を使った既存の FailSafe クラスタを CXFS 用に再設定 94
既存のクラスタの変更 94
クラスタ・ノードの修正またはアップグレード 95
GUI を使ったノード・タスク 95
GUI を使ったノードの定義 95
GUI を使ったノードの定義例 100
GUI を使ったクラスタ内のノードの追加と削除 101
GUI を使ったノードのリセット 102
GUI を使ったノードの変更 103

GUI を使った FailSafe ノードの CXFS への切替え . . . 105
GUI を使ったノードの削除 . . . 106
GUI を使ったノードの表示 . . . 106
GUI を使ったクラスタ・タスク . . . 106
GUI を使ったクラスタの定義 . . . 106
GUI を使ったクラスタ定義の変更 . . . 108
GUI を使った FailSafe クラスタの CXFS への切替え . . . 108
GUI を使ったクラスタの削除 . . . 109
GUI を使ったクラスタの表示 . . . 109
GUI を使った CXFS サービス・タスク . . . 109
GUI を使った CXFS サービスの開始 . . . 109
GUI を使った CXFS サービスの停止（通常 CXFS シャットダウン） . . . 110
GUI を使ったタイブレーカー・ノードの設定 . . . 111
GUI を使ったログの設定 . . . 111
GUI を使ったログ・グループ定義の表示 . . . 112
GUI を使ったログ・グループの設定 . . . 112
GUI を使ったファイルシステム・タスク . . . 113
GUI を使ったファイルシステムの定義 . . . 113
GUI を使ったファイルシステムのマウント . . . 114
GUI を使ったファイルシステムのアンマウント . . . 115
GUI を使ったファイルシステムの変更 . . . 115
GUI 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 . . . 116
GUI を使ったファイルシステムの削除 . . . 116
GUI を使った診断タスクとエラー回復 . . . 117
GUI を使った接続性のテスト . . . 117
GUI を使ったローカル・ノードのメンバーシップ取消し . . . 117
GUI を使ったローカル・ノードのメンバーシップ許可 . . . 118

5. `cmgr` リファレンス . . . 119

`cmgr` を使ったデフォルトの設定 . . . 120

cmgr を使ったノード・タスク 120
cmgr を使ったノードの定義 121
cmgr を使ったノードの変更 126
cmgr を使ったノードの重みの変更例 126
パーティションの設定例 127
cmgr を使ったノードのリセット 129
cmgr を使ったノードの CXFS または FailSafe への切替え 129
cmgr を使ったノードの削除 130
cmgr を使ったノードの表示 132
cmgr を使ったクラスタ・タスク 134
cmgr を使ったクラスタの定義 134
cmgr を使ったクラスタの変更 136
cmgr を使ったクラスタの CXFS または FailSafe への切替え 137
cmgr を使ったクラスタの削除 138
cmgr を使ったクラスタの表示 139
cmgr を使った CXFS サービス・タスク 140
cmgr を使った CXFS サービスの開始 140
cmgr を使った CXFS サービスの停止 140
cmgr を使ったタイブレーカー・ノードの設定 141
cmgr を使ったログの設定 142
cmgr を使ったログ・グループ定義の表示 142
cmgr を使ったログ・グループの設定 142
cmgr を使ったログ・グループの変更 143
cmgr を使ったファイルシステム・タスク 144
cmgr を使ったファイルシステムの定義 144
cmgr を使ったファイルシステムのマウント 150
cmgr を使ったファイルシステムのアンマウント 151
cmgr を使ったファイルシステムの変更 151
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 155

cmgr を使ったファイルシステムの削除 . . . 155
cmgr を使った診断タスクとエラー回復タスク . . . 156
cmgr を使ったネットワークの接続性のテスト . . . 156
cmgr を使ったシリアル接続のテスト . . . 157
cmgr を使ったローカル・ノードのメンバーシップ取消し . . . 157
cmgr を使ったローカル・ノードのメンバーシップ許可 . . . 157
スクリプトの例 . . . 157

6. 管理とメインテナンス . . . 161
マウント操作前後のスクリプトの実行 . . . 162
例: cxfs-pre-mount . . . 163
例: cxfs-post-mount . . . 163
例: cxfs-pre-umount . . . 164
例: cxfs-post-umount . . . 164
fsr(1M) の使用 . . . 164
find(1) と crontab(1) の使用 . . . 164
階層ストレージ管理 (HSM: Hierarchical Storage Management) システムの使用 . . . 165
ファイルシステムのメタデータ・サーバの検出 . . . 165
GUI を使用したメタデータ・サーバの検出 . . . 165
cluster_status を使用したメタデータ・サーバの検出 . . . 167
clconf_info を使用したメタデータ・サーバの検出 . . . 167
メタデータ・サーバの回復 . . . 167
データベースと CXFS のシャットダウン . . . 168
クラスタ設定データベースのシャットダウン . . . 169
ノードのステータスとクラスタ設定データベースのシャットダウン . . . 169
ノードがメタデータ・サーバの場合 . . . 170
クラスタ設定データベースの再起動 . . . 170
通常 CXFS シャットダウン . . . 170
ノードのステータスと通常 CXFS シャットダウン . . . 171
通常 CXFS シャットダウンを実行しない場合 . . . 171

通常 CXFS シャットダウン後のクラスタへの再設定 . . . 171
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 . . . 172
ノードのステータスと強制 CXFS シャットダウン . . . 172
強制 CXFS シャットダウン後のクラスタへの再設定 . . . 173
リセット機能と強制 CXFS シャットダウン . . . 173
再起動時のデータベースの再起動の回避 . . . 173
ログ・ファイル管理 . . . 174
すべてのログ・ファイルのローテーション . . . 174
大容量のログ・ファイルのローテーション . . . 174
ボリューム管理 . . . 175
ディスク管理 . . . 175
バックアップ . . . 175
NFS . . . 176
Quota . . . 176
SAMBA . . . 176
ファイルシステムのメインテナンス . . . 176
ファイルシステムのマウント . . . 177
ファイルシステムのアンマウント . . . 177
ファイルシステムの拡張 . . . 178
ダンプと復元 . . . 178
クラスタ・データベースのバックアップと復元 . . . 180
chkconfig フラグ . . . 180
システム調整可能パラメータ . . . 181

7. ステータスのモニタ . . . 183

ログ・ファイル . . . 183
クラスタのステータス . . . 184
アイコンとカラーのキー . . . 184
GUI を使ったクラスタのステータスの確認 . . . 186
cluster_status を使ったクラスタのステータスの確認 . . . 186

clconf_info を使ったクラスタのステータスの確認 . . . 187
cmgr を使ったクラスタのステータスの確認 . . . 187
ノードのステータス . . . 188
GUI を使ったノードのステータスのモニタ . . . 188
cmgr を使ったクラスタのステータスの照会 . . . 189
cluster_status を使ったノードのステータスのモニタ . . . 190
cmgr を使ったシステム・コントローラへの ping の実行 . . . 190
cmgr を使ったリセット・シリアル回線のモニタ . . . 190
XVM 統計 . . . 191

8. トラブルシューティング . . . 193
トラブルシューティング方法 . . . 193
ツールについての理解 . . . 193
物理ストレージ・ツール . . . 194
クラスタ設定ツール . . . 194
クラスタ制御ツール . . . 195
ネットワーク・ツール . . . 196
クラスタ/ノードのステータス・ツール . . . 197
パフォーマンス・モニタ・ツール . . . 198
カーネルのステータス・ツール . . . 198
ログ・ファイル . . . 199
クラスタのステータスの識別 . . . 200
問題の検出 . . . 201
問題の回避 . . . 203
適切な開始 . . . 204
残っているクラスタの削除 . . . 204
fs2d メンバーシップ上限量の安定性 . . . 204
設定の統一性 . . . 205
ノードの適切な重み付け . . . 205
GUI の使用 . . . 205

ログ・ファイル名とサイズ 206
Netscape と Brocade スイッチの GUI 206
書込まれていない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 206
書込まれていないエクステントの追跡 206
排他書込みトークン 207
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 207
システム容量の効果的な使用 208
ノード ID またはクラスタ ID を変更する前の再起動 208
使用されていないノードの削除 209
強制シャットダウン後の CXFS の再開 209
リセット回線の取外し 209
xfs_repair の適切な使用 209
一般的な問題 211
ファイルシステムにアクセスできない 211
GUI が実行されない 211
ログ・ファイルがディスク容量を消費しすぎる 211
ノードを定義できない 212
システムがハングする 212
ノードは検出されるが、メンバーシップに設定されない 212
セル ID カウントと“ Membership Delivered” メッセージ 212
ログインできない 213
エラー・メッセージの理解 213
通常メッセージ 213
clconfd デーモンが異常終了する場合 215
論理スワップ・スペース不足 216
クラスタ名 ID 欠如エラー 216
CXFS メンバーシップの取消し 217
ライセンス・エラー 217
IP アドレス・エラー 217

SYSLOG エラー 218
cli エラー 219
clconfd エラー 220
crsd エラー 223
cmond エラー 223
fs2d エラー 224
一般メッセージ 226
ログ・ファイル・エラー 226
cad メッセージ 227
cli メッセージ 228
crsd エラー 229
fs2d エラー 230
修正処置 231
CXFS サービスの再開 231
クラスタ設定データベースのクリア 231
再起動 232
2 ノード・クラスタの回復 232
ノードをクラスタに再設定しない再起動 234
クラスタ・インフラストラクチャ・デーモンの停止と再開 235
クラスタ設定データベースの再作成 235

付録A. 初期設定チェックリスト 237

用語集 239

索引 247

図一覧

図1-1 プールとクラスタの概念 8
図1-2 均等に分散されているメタデータ・サーバ 11
図1-3 複数のメタデータ・サーバ 12
図1-4 1 つのメタデータ・サーバ 13
図1-5 再設定と回復 14
図1-6 リセットを使用した 4 ノード・クラスタの例 15
図1-7 リセットを使用した 2 ノード・クラスタの例 16
図1-8 ネットワークの分割による上限量とアクティブ・メタデータ・サーバの変化 20
図1-9 最小限の CXFS および `fs2d` データベース・メンバーシップの概念 21
図1-10 特定のノード内の管理通信 27
図1-11 特定のノード内のデーモン通信 28
図1-12 プール内のノード間の通信 29
図1-13 クラスタ内にないノードの通信 30
図1-14 書込み時のメタデータの流れ 31
図1-15 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 . . 32
図1-16 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 . . . 33
図1-17 同時実行時の特定のノード内の管理通信 40
図1-18 同時実行時の特定のノード内のデーモン通信 41
図1-19 CXFS マネージャの初期ウィンドウ 47
図1-20 GUI でのノードの詳細の表示 48
図3-1 CXFS マネージャ 79
図4-1 ノード定義の例 100
図4-2 システム・コントローラ設定の例 101
図6-1 メタデータ・サーバが表示されているウィンドウ 166
図7-1 ノードのステータス 189

図7-2 pmgxvm グラフ 192

表一覧

表1-1 CXFS のデーモンとスレッド 25
表2-1 fs2d.options ファイル・オプション 64
表5-1 システム・コントローラ・タイプ 123
表6-1 システム調整可能パラメータ 181
表7-1 アイコンのキー 184
表7-2 カラーのキー 185
表8-1 SYSLOG エラー・メッセージの形式 218
表8-2 ログ・エラー・メッセージの形式 227

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 193 ページの 第8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。

- 『EL Serial Port Server Installation Guide』(Digi International)
- 『EL Serial Port Server Installation Guide Errata』
- 『FDDIXPress Administration Guide』
- 『IRISconsole Administrator's Guide』
- 『IRIS FailSafe Version 2 Administrator's Guide』
- 『IRIX 6.5 Installation Instructions』
- 『IRIX Admin: Disks and Filesystems』
- 『NIS Administrator's Guide』
- 『Performance Co-Pilot User's and Administrator's Guide』
- 『Performance Co-Pilot Programmer's Guide』
- 『Personal System Administration Guide』
- 『SGI TP 9400 RAID Owner's Guide』
- 『SGI TP9400 RAID Administration Guide』
- 『SGI Total Performance 9100 Storage System Owner'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)
- `cluster_mgr`(1M)
- `cluster_status`(1M)
- `crsd`(1M)

出版物の入手方法

SGI のドキュメントは、下記の SGI Technical Publications Library サイトで入手可能です。

`http://techpubs.sgi.com.`

表記規則

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。
変数	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。

「GUI 要素」　ウィンドウ、画面、ダイアログ・ボックス、メニュー、ツールバー、アイコン、ボタン、ボックス、フィールドなどのグラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) の要素は、本文中ではかぎかっこ (「」) で囲まれて表示されます。

このガイドでは、*IRIS FailSafe* の簡略名として *FailSafe* を使用しています。

ご意見とお問い合わせ先

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、弊社までお問い合わせください。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、

  `techpubs@sgi.com`

  までお送りください。

- 次の Technical Publications Library の World Wide 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://techpubs.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 Pkwy., 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

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。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) ディスクで動作します。CXFS と IRIS FailSafe は、同じインフラストラクチャを使用します。

この章では、以下の内容について説明します。

- 「XFS と CXFS の比較」
- 4 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」
- 6 ページの「クラスタ環境」
- 31 ページの「読取りおよび書込み時のメタデータの流れ」
- 33 ページの「ハードウェアおよびソフトウェアの必要条件とサポート」
- 35 ページの「CXFS と IRIS FailSafe の同時実行」
- 41 ページの「重み付けされた 2 つのノードだけのクラスタでの回復と再設定の問題」
- 42 ページの「クラスタ・マネージャのツール」

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 193 ページの 第8章「トラブルシューティング」をよくお読みください。

XFS と CXFS の比較

CXFS は、高いパフォーマンスを持つ SGI の XFS ファイルシステムに基づいており、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 つのシステム上に 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。
- CXFS の場合
  - ファイルシステムは、cmgr(1M) コマンドまたはグラフィカル・ユーザ・インタフェース (GUI: Graphical User Interface) を使用してマウントします。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。
  - ファイルシステムは、クラスタ内のすべてのシステム (ノード) に存在します。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メタデータ・サーバが行います。
  - 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)。

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、高トランザクション・レート、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 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、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。

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、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
- 1 つのプロセスだけが開いているファイルに対して読取りや書込みを行う場合
- ファイルを開いているすべてのプロセスが同じホスト上に存在するときに、そのファイルに対して読取りや書込みを行う場合
- 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ために、ダイレクト・アクセス I/O を使用して、ファイルに対して読取りや書込みを行う場合
- 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。

XFS を使用すべきケース

操作によっては、ローカル XFS ファイルシステムよりも CXFS の方が低速になる場合があります。たとえば、メタデータ・サーバ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挙げます。

- 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、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き込みを行うと、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。

パフォーマンスの高い分散型の共有読取りや書込みが頻繁に必要なアプリケーションでは、バッファ I/O の代わりにダイレクト・アクセス I/O を使用して高速なローカル・ファイル・パフォーマンスを実現できます。たとえば、分散型の並列データベースでは、ダイレクト・アクセス I/O が使用される傾向があります。

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(1 秒あたりの I/O)および帯域幅(1 秒あたりの MB)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。10～1000 ビット Ethernet の上限は、それぞれ 1～100 MB/秒です。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・チャンネル全体で得られる帯域幅よりも数段遅いことになります。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、ネットワーク・ファイルシステム (NFS: Network File System) 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・システム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、クラスタ内の各コンピュータ・システムは、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・チャン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コンピュータ・システムでローカル・ファイルとして扱われます。各システムは、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 を要求したシステムに直接渡され、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、各システムは 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・チャン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 IRIX 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・システムは、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 つの対称マルチプロセッサ (SMP: Symmetric Multiprocessor) 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 IRIX 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。

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(CXFS ファイルシステム上に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。

- 「用語」
- 15 ページの「ハードウェア・コンポーネント」
- 17 ページの「メンバーシップ上限量」

- 22 ページの「ハードウェア・リセット」
- 23 ページの「メタデータ・クライアント/サーバ・モデル」
- 24 ページの「システム表示」
- 24 ページの「デーモン」
- 27 ページの「通信パス」

用語

この節では、CXFS の理解に必要な用語を定義します。239 ページの用語集も参照してください。

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。すべてのノードで、同バージョンまたは直前のバージョンの IRIX OS が実行されている必要があります (6.5.13f と 6.5.14f など)。ノードは、ストレージ・システムとクラスタ内のノードを接続する SAN に接続されています。ノードが属することのできるクラスタは 1 つだけです。

この意味でのノードという用語は、Origin システムのノードとは異なる意味を持ちます。

クラスタ・データベース

クラスタ・データベースには、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。`fs2d` デーモンは、プール内のすべてのノードへのクラスタ・データベース (CDB) の配布を管理します。

プール

プールは、ネットワークによって相互に結合され、クラスタ・データベースでノードとして定義されているノードのセット全体です。通常、これらのノードは互いに近い位置にあり、同じ目的に使用されます。プール内の各ノードには、複製されたクラスタ・データベースが格納されています。

クラスタに追加できるノードはすべてプールの一部ですが、プール内のすべてのノードをクラスタの一部にする必要はありません。プールは 1 つだけです。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。

クラスタ

クラスタは、クラスタとして定義されているプール内のノードのセットです。クラスタは単純な名前で識別されます。この名前は、プール内で一意でなければなりません。たとえば、クラスタとノードで同じ名前を使用することはできません。

クラスタ内のすべてのノードはプールにも存在しますが、プール内のすべてのノードがクラスタに存在するとはかぎりません。つまり、クラスタは、プール内のノードのサブセットで構成されている場合があります。各プールのクラスタは 1 つだけです。

図1-1 に、プールとクラスタの概念を示します。

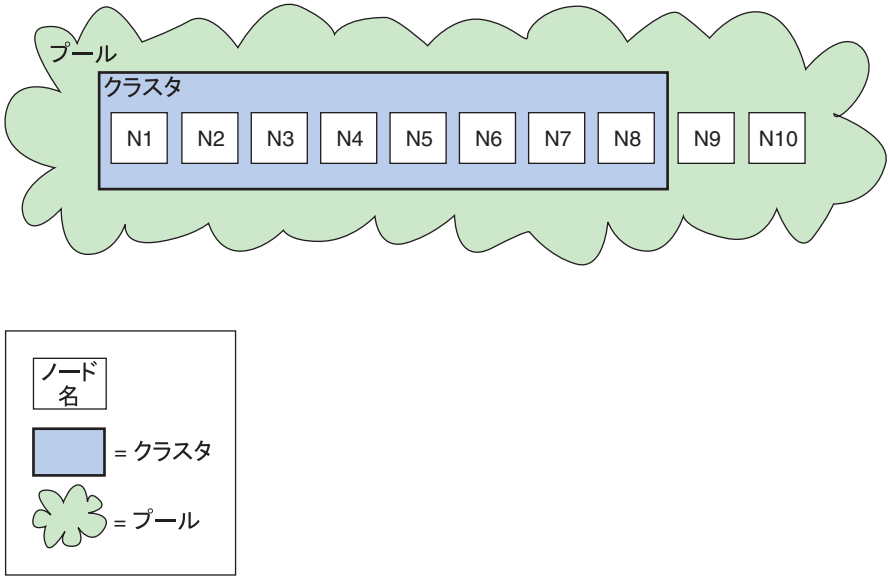

図1-1 プールとクラスタの概念

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 メンバーシップ。これは、カーネル空間メンバーシップとも呼ばれ、クラスタの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す。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。ノードは起動処理中に CXFS 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。
- `fs2d` データベース・メンバーシップ 。これは、ユーザ空間メンバーシップとも呼ばれ、`fs2d` にアクセス可能な、**プール**内のノードのグループです。`fs2d` デーモンは、プール内のすべてのノードへのクラスタ・データベースの配布を管理します。`fs2d` で利用可能なノードは、クラスタ・データベースの更新を受信できるため、`fs2d` データベース・メンバーシップの一部となります。プール内で定義されたノードのサブセットである場合があります。

CXFS メンバーシップおよび fs2d 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。

上限量

上限量とは、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 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、クラスタでノードの過半数 **(>50%)** が重み付けされている必要があります。
  - 既存のメンバーシップを維持するには、クラスタでノードの半数 **(50%)** が重み付けされている必要があります。
- fs2d 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 重み付けは使用されません。

詳細については、17 ページの「メンバーシップ上限量」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。

- 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・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 メンバーシップを取消し、設定によってはノードをリセットまたは切断することがあります。
- 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やなりすまし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(わずかな変動でも問題が発生する可能性があります)という理由から、プライベート・ネットワークは必須です。

さらに、すべてのノードを同じローカル・ネットワーク・セグメント上に配置することが推奨されています。

メモ: 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・セグメントとパーティション設定の詳細については、19 ページの「ネットワークの分割の例」を参照してください。

メタデータ

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。

メタデータ・サーバとメタデータ・クライアント

メタデータ・サーバは、クラスタ内のその他のノードに代わって、ファイルシステムのためにメタデータを調整します。その他のノードは、メタデータ・クライアントと呼びます。データは、メタデータと異なり、ホストとディスクの間を直接移動します(ピアツーディスク)。

メモ: メタデータ・サーバおよびメタデータ・クライアントを、ネットワーク・ファイス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メタデータ・クライアントのディスクに対して直接送受信されます。

各ファイルシステムには、使用可能な複数のメタデータ・サーバのリストがある場合がありますが、各ファイルシステムにアクティブなメタデータ・サーバは 1 つしかありません。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(ファイルシステムにつき 1 つ)。以下の図に、考えられるさまざまな場合を示します。

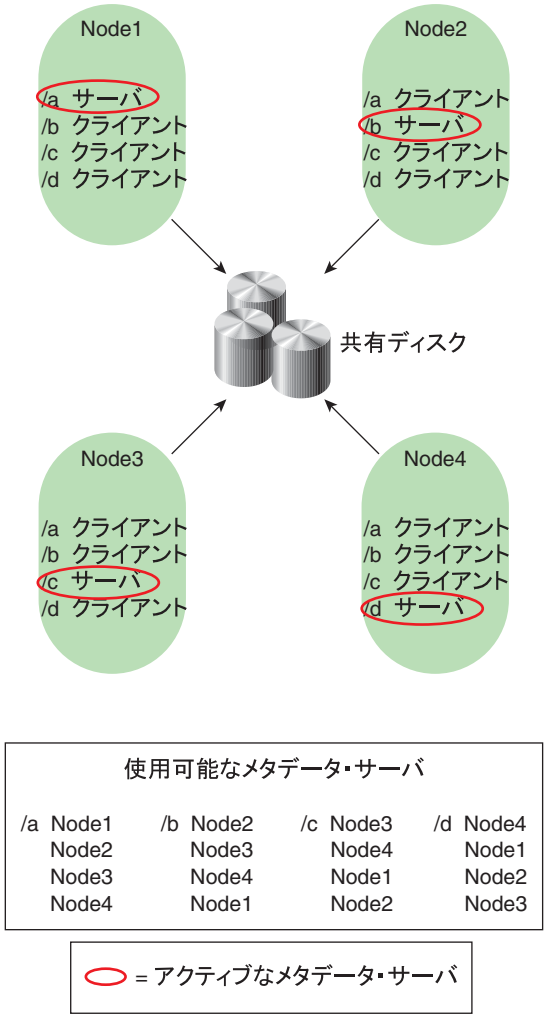

図1-2 均等に分散されているメタデータ・サーバ

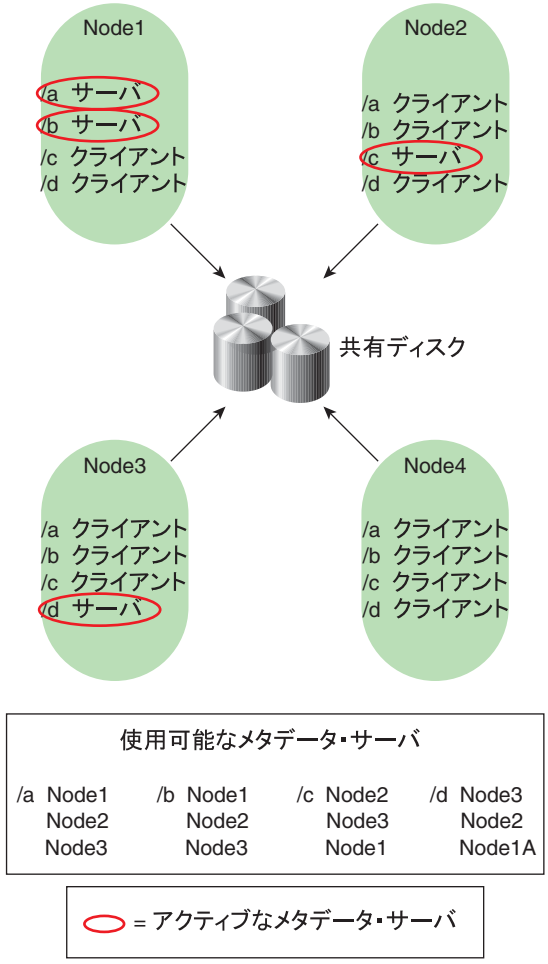

図1-3 複数のメタデータ・サーバ

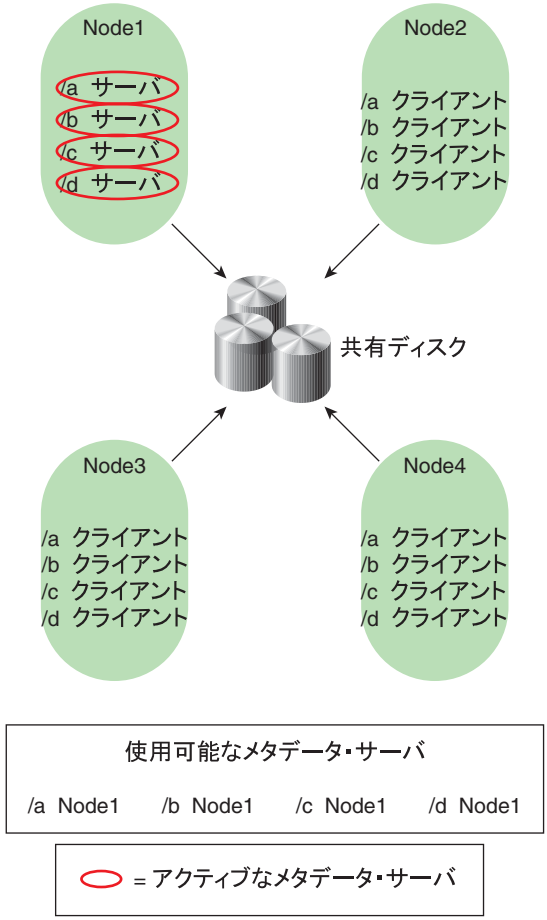

図1-4 1 つのメタデータ・サーバ

再設定

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。

ノード・メンバーシップは再設定の影響は受けません。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場合があります。

回復

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。

図1-5 に、再設定と回復の違いを示します。

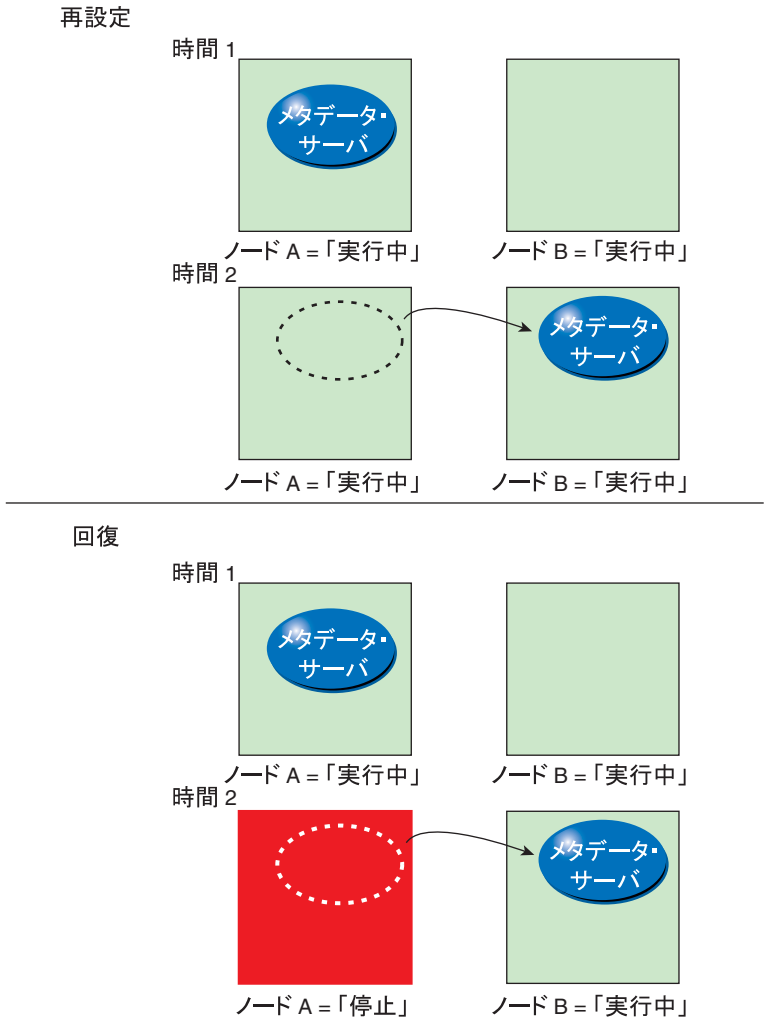

図1-5 再設定と回復

ハードウェア・コンポーネント

図1-6 に、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 4 ノード・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。パフォーマンスおよび制御の点から、ハブではなくスイッチが推奨されます（この図にはユーザ・ネットワークは示されていません）。

メモ: データの整合性を保証するため、特に重み付けされた 2 つのノードだけのクラスタでは、リセット機能の使用が強く推奨されています。IRIS FailSafe ではリセット機能は必須です（41 ページの「重み付けされた 2 つのノードだけのクラスタでの回復と再設定の問題」を参照してください）。

同様の設定では、Ethernet マルチプレクサの代わりに SCSI マルチプレクサを使用できます。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い合わせください。

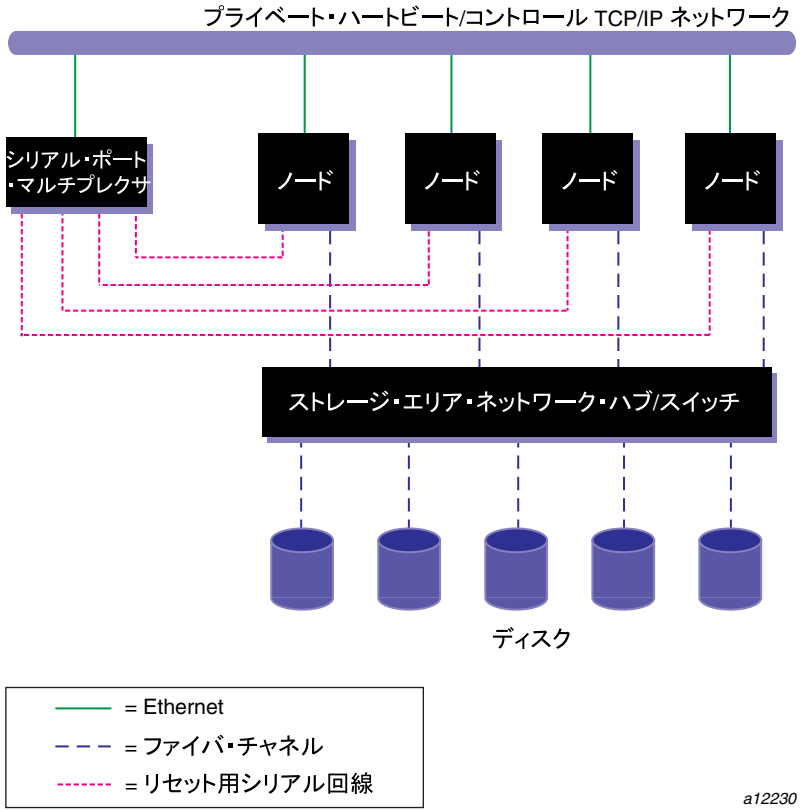

図1-6 リセットを使用した 4 ノード・クラスタの例

図1-7 に、リセット用の直接シリアル回線を使用する、重み付けされた 2 ノード・クラスタを示します（この図にはユーザ・ネットワークは示されていません）。

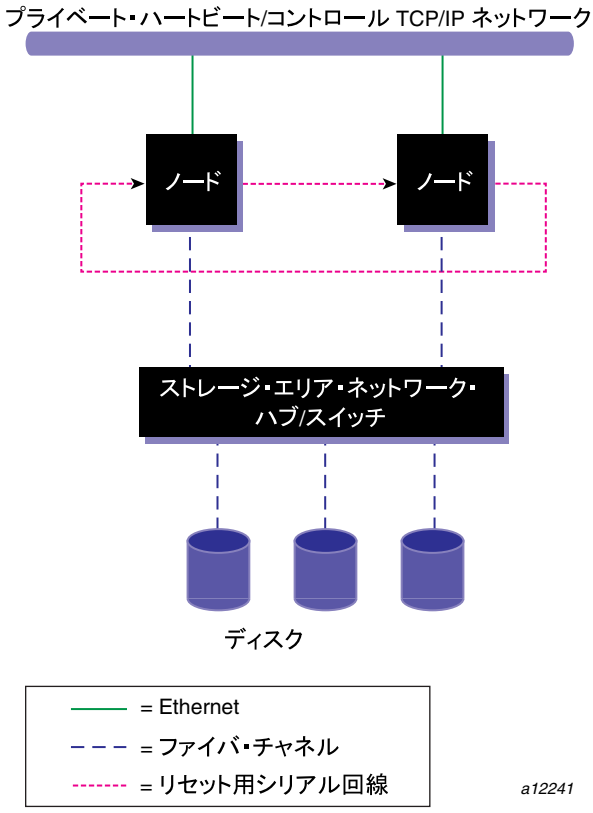

図1-7 リセットを使用した 2 ノード・クラスタの例

CXFS では、各ノードが SAN 上の論理ボリュームに直接アクセスできるので、パフォーマンスが向上します。各ノードは、NFS を使用したソリューションよりも密接に論理ボリュームに結合されています。データの整合性を保証するため、各ファイルシステムにメタデータ・サーバがあります。23 ページの「メタデータ・クライアント/サーバ・モデル」を参照してください。CXFS ソフトウェアにより、メタデータ・サーバが CXFS メンバーシップとアクセス上の問題を判断できるインフラストラクチャが提供されます。

メンバーシップ上限量

以下のように、メンバーシップの各タイプには別々の上限量のメカニズムがあります。

- 「CXFS メンバーシップ上限量」
- 21 ページの「fs2d データベース・メンバーシップ上限量」

CXFS メンバーシップ上限量

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/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/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 メンバーシップ上限量によって、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。

上限量の計算とノードの重み

CXFS メンバーシップ上限量は、CXFS メンバーシップに参加するノードの重みの合計を、クラスタ内で定義されているすべてのノードの合計の重みと比較することによって計算されます。

ノードのデフォルトの重みは 1 です。すべてのノードの重みが 1 の場合、初期 CXFS メンバーシップ上限量の計算結果は、原則的にノードの数の過半数 (>50%) になります。

クラスタが、1 つまたは複数の CXFS メタデータ・サーバと CXFS クライアントだけの複数のノードで構成されている場合は、クライアントの状態とは無関係に、メタデータ・サーバとなる可能性のあるノードだけで上限量が構成されるようにノードの重みを設定できます。この場合、メタデータ・サーバの重みは 1 に、クライアントの重みは 0 に定義する必要があります。重みが 0 のクライアントは、単体では CXFS メンバーシップを形成できません。この場合、CXFS メンバーシップには、メタデータ・サーバとなる可能性のあるノードの上限量が必ず必要です。

メモ: 少なくとも 1 つのノードのメンバーシップの重みが 0 より大きくなければなりません。また、メタデータ・サーバとなる可能性のあるノードのメンバーシップの重みも、0 より大きくなければなりません。

0 または 1 以外のメンバーシップの重みの値は推奨されません。

初期 CXFS メンバーシップ上限量では、クラスタを形成するには、重み付けされたノードの過半数 (>50%) が使用可能でなければなりません。すべてのノードの重みが 1 の大規模なクラスタの場合、クラスタを形成するには、大部分のノードの重みが 0 の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 メンバーシップ上限量を維持するには、クラスタで定義されている重み付けされたノードの半数 (50%) が必要です。

ハードウェア・リセットを使用しない場合は、ネットワーク分割の発生時に複数のクラスタが生じること(スプリットブレイン状態)を避けるために、CXFS タイブレーカー・ノードを設定してください。

メモ: データの整合性を保証するため、特に重み付けされた 2 つのノードだけのクラスタでは、リセット機能の使用が強く推奨されています。IRIS FailSafe ではリセット機能は必須です。

CXFS タイブレーカー・ノードは、CXFS で特に識別されるノードで、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 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、重み付けされた 2 つのノード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。

CXFS タイブレーカー・ノードがメタデータ・サーバとして動作できるよう、CXFS タイブレーカー・ノードの重みは 0 より大きくなければなりません。

ハートビート/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、メタデータ・サーバの上限量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。上限量の一部ではないハートビート/コントロール・ネットワークの部分にあるノードは、クラスタからシャットダウンされます。したがって、ハートビート/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 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 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 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 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にクラスタを拡大したり、CXFS 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不能なノードを削除すると、ノードの重みが 1 の場合は、上限量の形成に必要な重みが減ります。

上限量の変更の例

以下に、CXFS メンバーシップ上限量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 6 ノード(A、B、C、D、E、および F)のプールがあり、各ノードは重みが 1 で以下の状態だとします。

- 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 メンバーシップ上限量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 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の数は A、C、D、E の 4 つになります。この状態では、CXFS メンバーシップ上限量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

前の例の続きとして、図1-8 に、ルータがダウンしてハートビート/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上限量がありません。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、その他のアクティブでない 2 つのメタデータ・サーバ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。

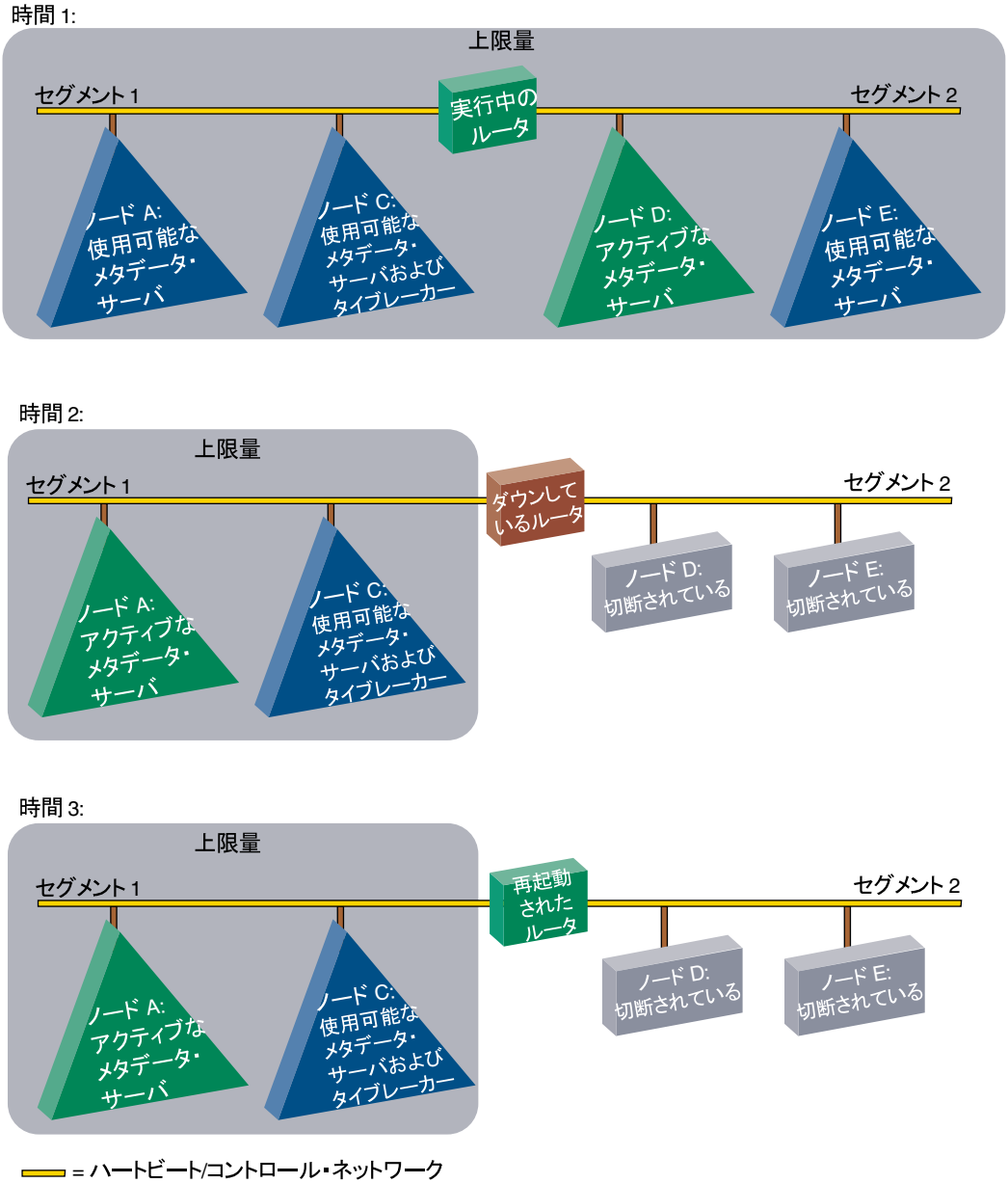

図1-8 ネットワークの分割による上限量とアクティブ・メタデータ・サーバの変化

fs2d データベース・メンバーシップ上限量

fs2d データベース・メンバーシップ上限量では、プール内のノードの半数 (50%) が fs2d デーモンで使用可能な場合に(したがって、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。fs2d データベース・メンバーシップ上限量を計算するときは、重み付けは使用されません。

メモ:これは CXFS メンバーシップとは異なります。CXFS メンバーシップの場合、初期クラスタを開始するには、クラスタ内の重み付けされたノードの過半数 (>50%) が使用可能である必要があります。CXFS メンバーシップは、重み付けされたノードの 50% だけで維持できます。

図1-9 に、最小限の CXFS および fs2d のデータベース・メンバーシップの概念を示します。

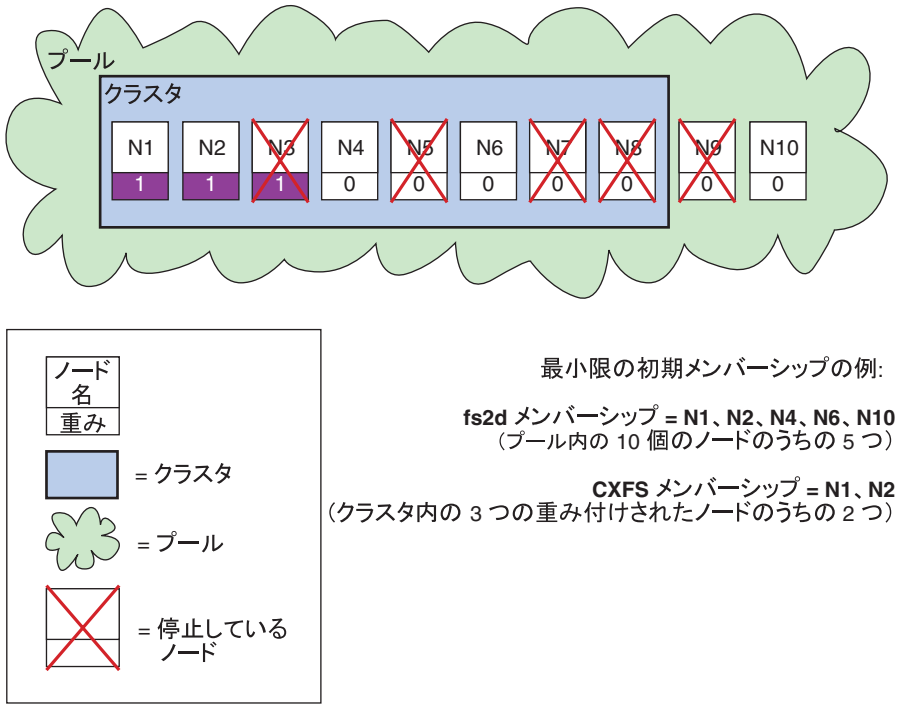

図1-9 最小限の CXFS および fs2d データベース・メンバーシップの概念

ハードウェア・リセット

クラスタとの接続が切断されたことを検出したノードは、強制的に共有ディスクへのアクセスを中止します。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、特に重み付けされた偶数のノードのクラスタの場合、中でも重み付けされた 2 つのノードだけのクラスタの場合は、リモート・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。メタデータ・サーバとなる可能性のあるすべてのノードは重み付けされていなければならず、重み付けされているノードにはリセット回線が必要です。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、リセット・ハードウェアがあれば、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 メンバーシップが 2 つの半数（それぞれが総重みの半分を持ちます）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・ノードとリセット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・レースに負けて再起動されます。
- CXFS タイブレーカー・ノードがなく、リセット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・ノードまたはリセット機能のいずれかを設定してください。ただし、重み付けされた 2 つのノード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、その重みが半分になって、初期CXFS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メンバーシップができるのを防止します。

リモート・リセット接続には以下の方法があります。

- 2つのノードのクラスタは、シリアル回線で直接接続できます。
- 3つ以上のノードのクラスタは、Ethernetマルチプレクサまたは SCSIマルチプレクサで接続する必要があります。各ノードは、ノードのリセット機能を持つノードであるオーナー・ホストを持つように定義されます。

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い合わせください。

メタデータ・クライアント/サーバ・モデル

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

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 LAN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。

- 重みが指定されているかぎり、クラスタ内のどのノードも、メタデータ・サーバとして使用可能なノードとして定義できます。
- 複数のファイルシステムのメタデータ・サーバとして同時に動作できるのは、クラスタ内の1つのノードだけです。

- それぞれが別のファイルシステムのセットを持つ複数のノードをメタデータ・サーバとして動作させることができます。ただし、特定のファイルシステムは、1 つのノードの 1 つアクティブなメタデータ・サーバを使用します。
- 複数のノード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。
- アクティブなクライアントがいる場合に、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、167 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している場合、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、162 ページの「マウント操作前後のスクリプトの実行」を参照してください。

詳細については、31 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。

デーモン

表1-1 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。

（タイプが `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1-1 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 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 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 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。
		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。

層	サブシステム	プロセス	説明
		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、書込まれていない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。

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、その他のデーモンの動作を継続させることです。

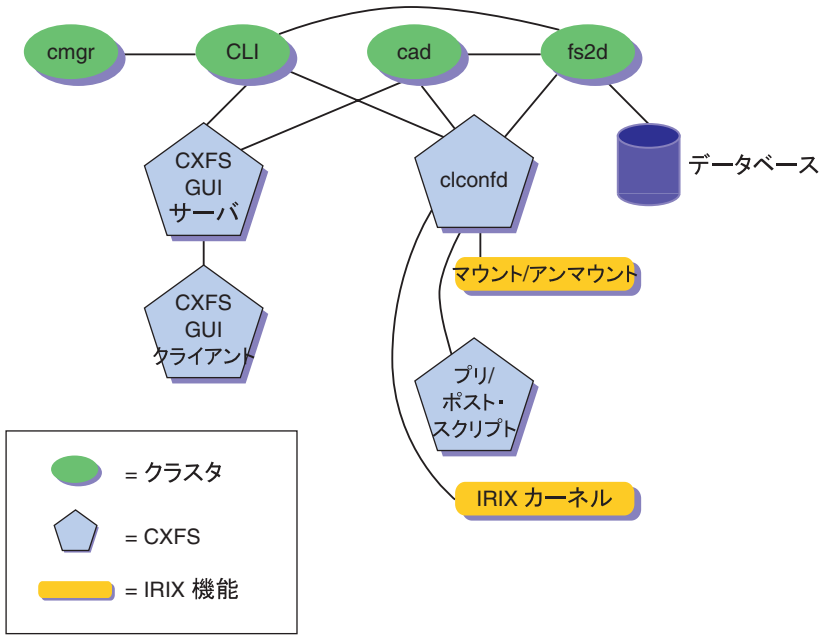

図1-10 特定のノード内の管理通信

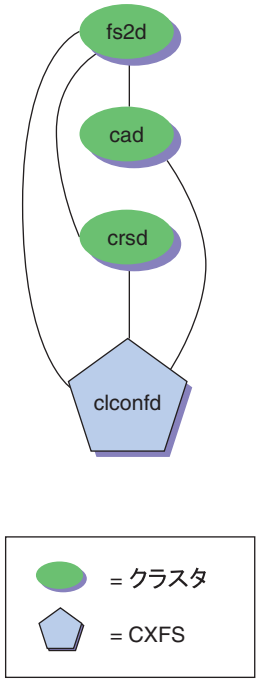

図1-11 特定のノード内のデーモン通信

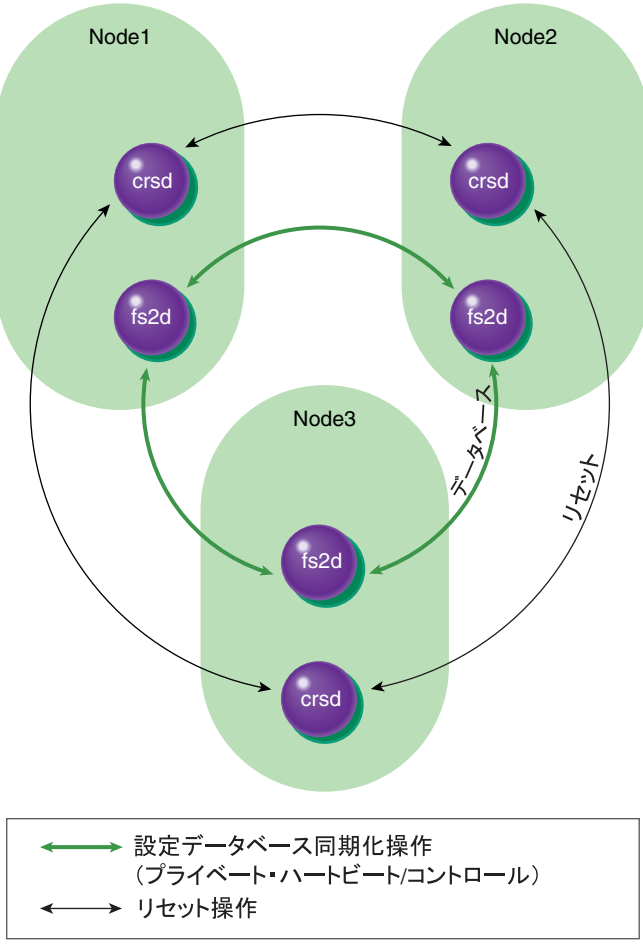

図1-12 プール内のノード間の通信

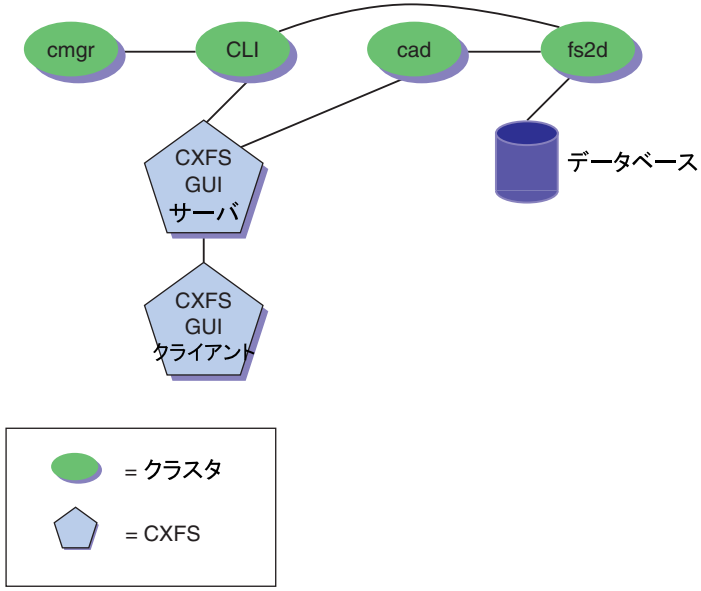

図1-13 クラスタ内にないノードの通信

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。

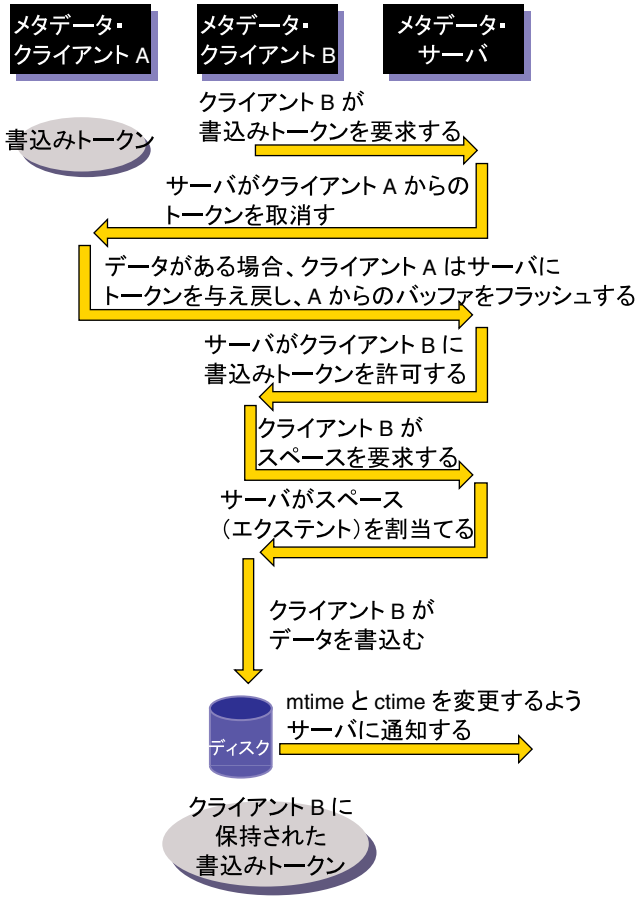

図1-14 書込み時のメタデータの流れ

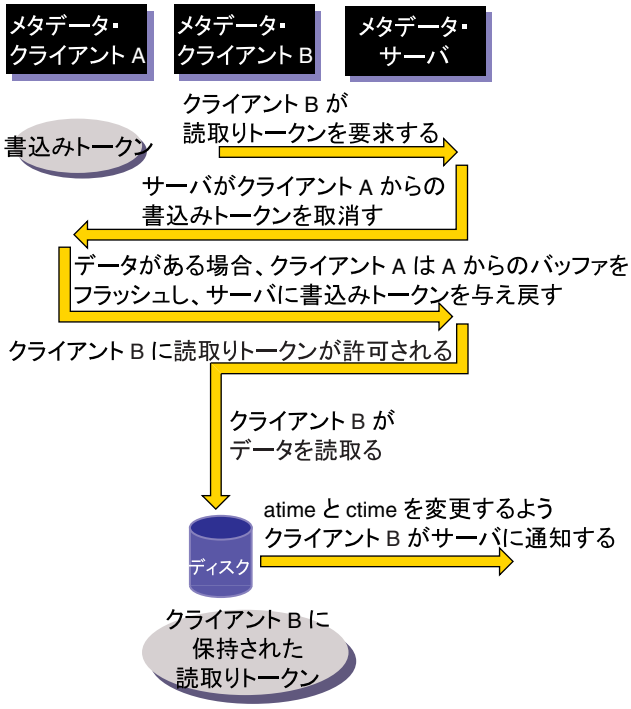

図1-15 クライアント A での書込み後のクライアント B での読取り時のメタデータの流れ

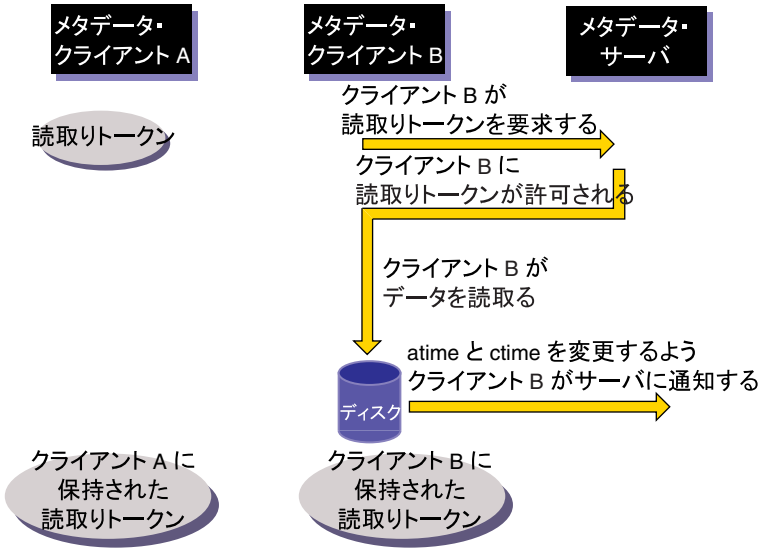

図1-16 クライアント A での読取り後のクライアント B での読取り時のメタデータの流れ

ハードウェアおよびソフトウェアの必要条件とサポート

CXFS では以下が必要です。

- CXFS だけのクラスタでは、最大 16 個のノードがサポートされています。CXFS のアーキテクチャは、最大 64 個のノードをサポートできるように設計されています。64 個のノードは、今後のリリースでサポートされる予定です。

  CXFS と IRIS FailSafe 2.1 以降の両方が実行されている(同時実行)クラスタでは最大 16 個のノードがサポートされており、これらのノードの 8 個までを FailSafe に使用できます。設定についての詳細は、35 ページの「CXFS と IRIS FailSafe の同時実行」を参照してください。

- プールとクラスタはそれぞれ 1 つだけです。
- すべてのノードで、同バージョンまたは直前のバージョンの IRIX OS が実行されている必要があります(6.5.12f と 6.5.13f など)。
- CXFS 用の FLEXlm ライセンス・キーが必要です。
- 各 CXFS メタデータ・サーバとクライアントに接続されたプライベート TCP/IP ネットワークが必要です。
- Origin 3000、Origin 2000、Origin 200、Onyx2、または Onyx3 プラットフォームが必要です。

- Silicon Graphics O2 ワークステーション以外のサポートされている SAN ハードウェア設定。これには以下のプラットフォームが含まれます。
  - Origin 2000
  - Origin 200
  - Octane
- 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。

CXFS は以下と共に動作します。

- Trusted IRIX/CMW。DMF (Data Migration Facility) および TMF (Tape Management Facility) を使用して、CXFS を SGI Trix 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。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) が同じになるようにシステムを設定する必要があります。

  Trusted-IRIX と IRIX が混在するクラスタの場合、IRIX クライアントは、Trix 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 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、`msen low, mint high` というラベルが使用されます。
- IRIS FailSafe。35 ページの「CXFS と IRIS FailSafe の同時実行」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。
- Ethernet マルチプレクサまたは SCSI マルチプレクサ (3 つ以上のノードのクラスタでリセット機能用に使用します)

CXFS と IRIS FailSafe の同時実行

CXFS 6.5.10 以降と IRIS FailSafe 2.1 以降（および関連するパッチ）を同じシステムにインストールして実行できます。これを同時実行と呼びます。これにより、アプリケーション・レベルの高可用性とクラスタ・ファイルシステムを利用できるようになります。

FailSafe の CXFS 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、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。

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。
- `CXFS`。この場合、すべてのノードのタイプが `CXFS` になります。
- `CXFS and FailSafe`(同時実行)。この場合、すべてのノードは、`CXFS` と `CXFS and FailSafe` 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。

クラスタのサイズ

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、重み付けされた 3 つ以上のノード（CXFS の重み）と 16 個以下のノードで設定することが推奨されています（リセット・ケーブルが接続されていて重み付けされた 2 つのノード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。41 ページの「重み付けされた 2 つのノードだけのクラスタでの回復と再設定の問題」を参照してください）。クラスタのすべてのノードでは、CXFS が実行されていなければなりません。最大 8 つのノードでは、IRIS FailSafe も実行できます。

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。

変換

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` に変換できます。FailSafe GUI を使用すると、並行して操作を実行できます。変換後のノードは、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。94 ページの「GUI を使った既存の FailSafe クラスタを CXFS 用に再設定」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。

129 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137 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に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

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）。

同時実行クラスタ内の通信パス

以下の図に、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。

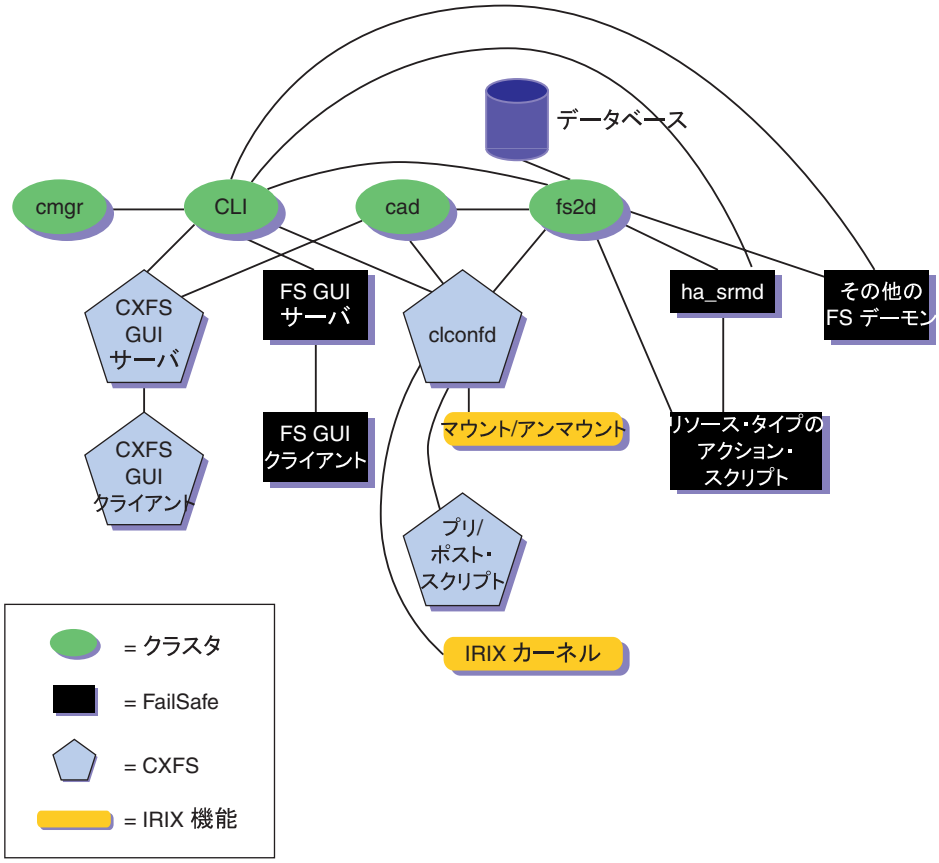

図1-17 同時実行時の特定のノード内の管理通信

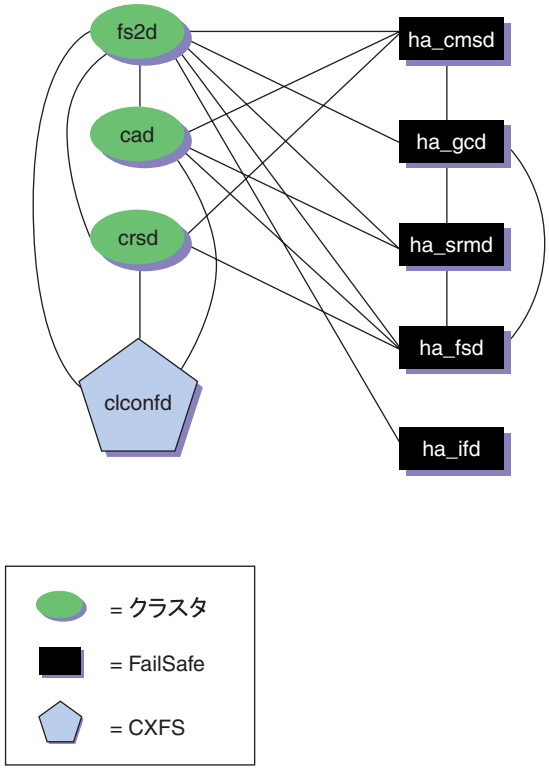

図1-18 同時実行時の特定のノード内のデーモン通信

重み付けされた 2 つのノードだけのクラスタでの回復と再設定の問題

実作業用の環境（つまり、メタデータ・サーバの再設定と回復が必要な環境）では、少なくとも 3 つの重み付けされたノードのクラスタが推奨されています。

重み付けされた偶数のノード（特に重み付けされた 2 つのノードだけ）の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。

- 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（リセット回線は、すべての CXFS クラスタに対して推奨されており、IRIS FailSafe と共に使用する場合は必須です）。
- 1 つのノードの重みを 1 に設定し、その他は 0 に設定します。
- CXFS タイブレーカー・ノードを設定します。

ただし、これらの方法を使用しても、重み付けされた 2 つのノード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。

リセット回線を持つ重み付けされた 2 つのノード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。

- 両方のノードの重みが 1 で、CXFS タイブレーカー・ノードがない場合

  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ます。

- 両方のノードの重みが 1 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 メンバーシップ上限量は重みの 50% があれば維持できるので、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ます。

- 両方のノードの重みが 1 で、クライアントが 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、クライアントは切抜けます。このクライアントがリストの 2 番目のメタデータ・サーバの場合は、アクティブなメタデータ・サーバになります。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。

  メモ: 通常は、プライマリ・メタデータ・サーバを CXFS タイブレーカー・ノードにします。

詳細については、15 ページの「ハードウェア・コンポーネント」および17 ページの「上限量の計算とノードの重み」を参照してください。

クラスタ・マネージャのツール

CXFS の設定タスクは、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) または `cmgr`(1M) クラスタ・マネージャ・コマンド (`cluster_mgr`) のいずれかを使用して実行できます。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。
- タスクおよびタスクセットにより、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 マシンか、または Windows や Linux 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 IRIX システム上だけでクラスタ・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。

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)

この節では、CXFS マネージャ GUI の概要について説明します。

GUI の起動

CXFS デーモンが開始されている場合、クラスタの正確なステータスを参照するには、すべての CXFS デーモンが実行されているノードに接続するようにしてください。クラスタ内で CXFS デーモンが開始されていない場合は、プール内の任意のノードに接続できます。

クラスタの表示は、どのノードから GUI を実行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、ツリー表示に変更内容が表示されるまで待ってください。ツリー表示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。クラスタ・データベースに変更を加えると、その都度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、設定が大きくなるほど送信に時間がかかります。

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 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（cxtask 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ウィンドウは、すべて単一の GUI インスタンスの一部なので、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 を起動するには、以下のいずれかの方法を使用します。

- 以下のコマンド・ラインを入力します。

  ```
  # /usr/sbin/cxtask
  ```

  クラスタを定義する前に cxtask 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は、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・ラインは cxtask と同じ機能を実行します。従来の操作性を維持するために両方のコマンドが保持されています。

- Toolchest から以下を選択します。

  「CXFS」
  　>「CXFS マネージャ(CXFS Manager)」

  メモ: 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 &
  ```

  これを反映させるには、クライアント・システムに sysadm_cxfs.sw.desktop がインストールされていなければなりません。

  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。

- Web ブラウザで http://*server*/CXFSManager/（*server* は、管理するプールまたはクラスタ内のノードの名前）と入力して、<Enter> キーを押します。表示される Web ページで「シールド」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。「シールド」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 `/usr/sbin/cxtask` コマンドを使用する方法が推奨されています。

GUI の概要

CXFS マネージャを使用すると、単一のポイントから CXFS クラスタ全体を管理できます。CXFS マネージャは、CXFS クラスタの設定と管理を行いやすくするタスクとタスクセット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・タスクセットは、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。

「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ます)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(Edit)」メニューでは、ツリー表示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に選択することもできます。

「タスク(Tasks)」メニューには以下が含まれます。

- 「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。
- 「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、既存のクラスタの変更、およびステータスの確認を行うためのタスクセットが含まれます。
- 「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、およびログ設定を行うことができます。
- 「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・サーバの再設定のためタスクが含まれます。

メモ: 再設定については、このリリースでは未実装であるため使用できません。

- 「診断(Diagnostics)」設定に問題がないかどうか、クラスタとノードをテストするためのタスクが含まれます。

- 「エラー回復(Error Recovery)」CXFS メンバーシップ・エラーから回復できるタスクが含まれます。

デフォルトでは、ウィンドウはツリー表示とアイテム表示の 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。現在のツリーでコンポーネントを検索するには、コンポーネントの名前を入力して、「検索(Find)」ボタンをクリックします。

ツリー表示でアイテムの選択を解除するには、ツリー表示でアイテムの名前以外の場所をクリックします。

クラスタ・コンポーネントの表示

クラスタ内のノード、プール内のノード(定義されているすべてのノード)、またはクラスタ内のファイルシステムのうち、表示したいものを「表示(View)」の選択肢から選択します。

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、ツリー表示でそのアイコン名をクリックします。設定とステータスの詳細が右側のアイテム表示に表示されます。アイテム表示でアイテムの詳細を参照するには、名前を選択すると(選択した名前は青色で表示されます)、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(Task)」メニューからタスク名を選択するか、またはツリー表示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(Task)」
   >「設定ガイド(Guided Configuration)」
   >「新規クラスタの設定(Set Up a New Cluster)」

   タスク・ウィンドウまたはタスクセット・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ウィンドウに移動し、そのウィンドウで情報を入力してから「OK」をクリックします。

   メモ: 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。

画面

図1-19 に、「CXFS マネージャ(CXFS Manager)」ウィンドウを示します。図1-20 には、アイテム表示の詳細を示します。

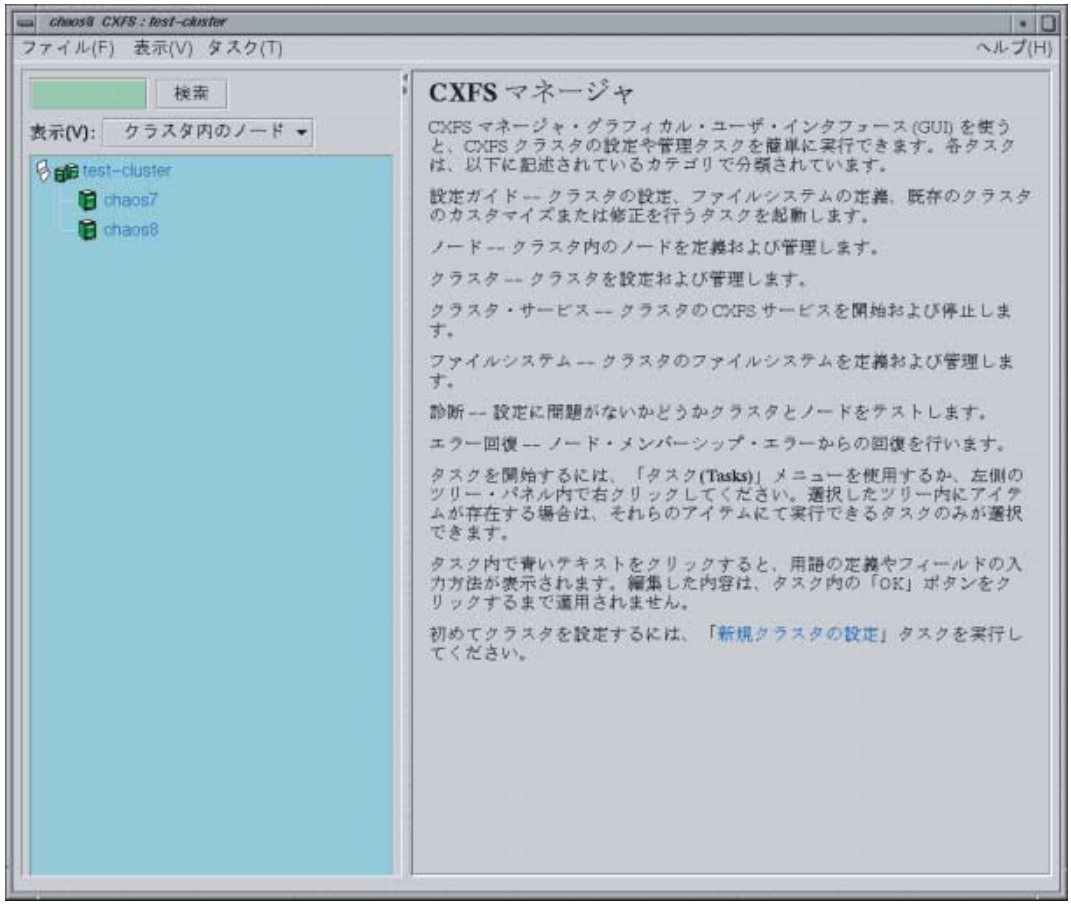

図1-19 CXFS マネージャの初期ウィンドウ

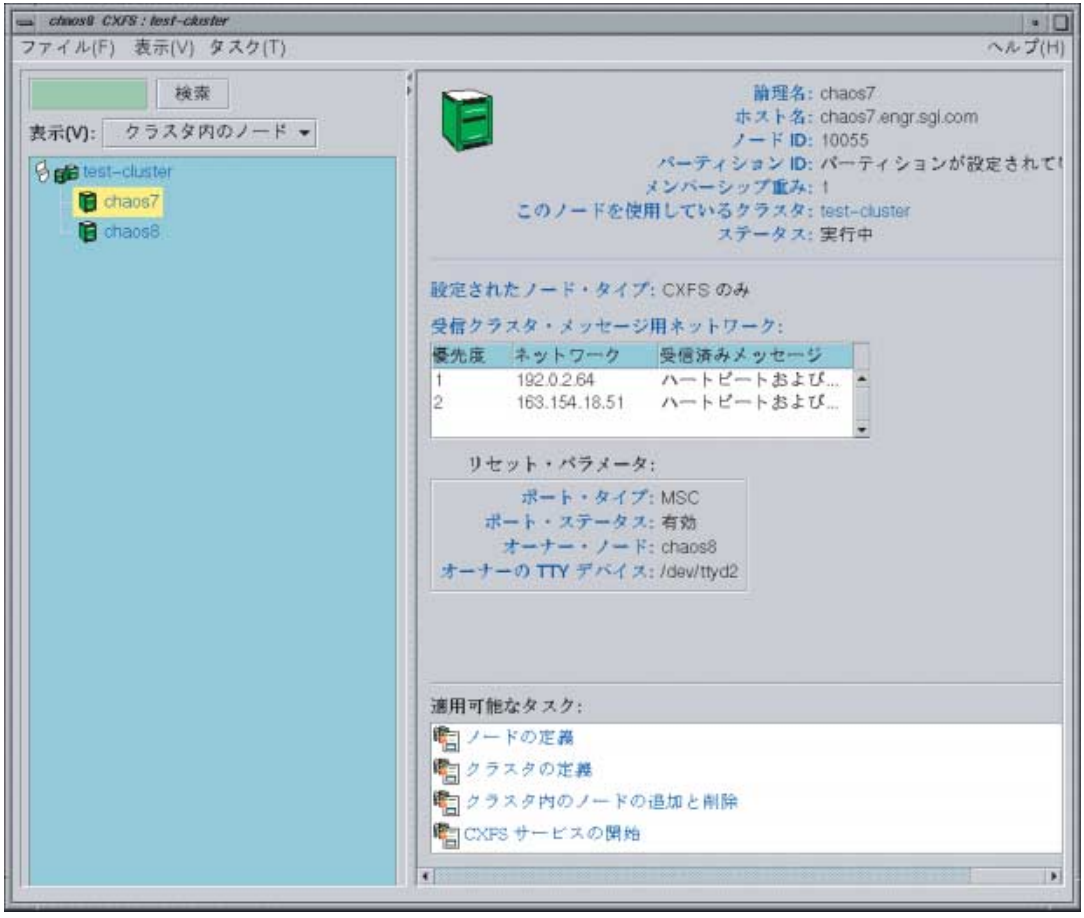

図1-20 GUI でのノードの詳細の表示

cmgr(1M) の概要

cmgr(1M) を使用するには、root としてログインする必要があります。続いて以下を入力します。

```
# /usr/cluster/bin/cluster_mgr
```

または

```
# /usr/cluster/bin/cmgr
```

このコマンドを入力すると、以下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>
```

使用可能な cmgr コマンドについての詳細は、119 ページの 第5章「cmgr リファレンス」および cmgr(1m) のマン・ページを参照してください。

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。以下のいずれかの方法で、プロンプト・モードで実行できます。

- 以下の例のように、コマンド・ラインで -p オプションを指定します。

  ```
  # cmgr -p
  ```

- 以下の例のように、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、以下のコマンドを入力します。

  ```
  cmgr> set prompting off
  ```

以下に、プロンプト・モードで尋ねられる質問の例を示します。

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

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> ?
Node ID ? [optional]
Partition ID ? [optional] (0)
Do you wish to define system controller info[y/n]:
Reset type <powerCycle> ? (powerCycle)
```

```
Do you wish to define system controller info[y/n]:
Sysctrl Type <msc|mmsc|l2>? (msc)
Sysctrl Password[optional] ? ( )
Sysctrl Status <enabled|disabled> ?
Sysctrl Owner ?
Sysctrl Device ?
Sysctrl Owner Type <tty> ? (tty)
Number of Network Interfaces ? (1)
NIC 1 - IP Address ?
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false> ?
NIC 1 - (use network for control messages) <true|false> ?
NIC 1 - Priority <1,2,...> ?
Node Weight ? (1)
```

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確定して、`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 行目に以下を含めておき、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（`#` で始まります）、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に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 に以下のコマンドが含まれるとします。

```
cxfs6# more /tmp/showme
show clusters
show nodes in cluster cxfs6-8
quit
```

この場合は、以下のコマンドを実行でき、以下の出力が得られます。

```
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、157 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、以下を入力します。

```
cmgr> sh
cxfs6#
```

シェルを終了して cmgr> プロンプトに戻るには、以下を入力します。

```
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。

```
cluster_mgr -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、次のコマンドを入力します。

```
# cmgr -c "show clusters"
1 Cluster(s) defined
      eagan
```

詳細については、119 ページの 第5章「cmgr リファレンス」および 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 つのファイルに連結して、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連結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 ファイルを連結します。

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。

この章は、以下の手順で構成されています。

1. 56 ページの「ソフトウェアのインストール」
2. 60 ページの「システム・ファイルの設定」
3. 67 ページの「(オプション) 自動再起動用の設定」
4. 68 ページの「ネットワーク・インタフェースの設定」
5. 70 ページの「シリアル・ポートの設定」
6. 71 ページの「システムの再起動」
7. 73 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、75 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。

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 193 ページの 第8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。

ソフトウェアのインストール

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。必要な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. プール内の各ノードで、『IRIX 6.5 Installation Instructions』に従って IRIX 6.5.14f 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、以下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。シリアル・ポート・サーバに付属の CD を使用してください。インストールの後、システムを再起動してください。

   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. プール内の各ノードで、以下の作業を行います。

   a. CXFS license key 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の注意事項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・プロバイダにお問い合せください。

      ライセンスが適切にインストールされている場合は、cxfslicense コマンドから以下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、以下のエラーが表示されます。

      ```
      CXFS not properly licensed for this host.  Run
                      '/usr/cluster/bin/cxfslicense -d'
             for detailed failure information.
      ```

      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。ただし、既存の Origin 3000 システムを再パーティション設定した場合は、新しいライセンスは必要ありません。

      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。

   b. CD-ROM #2 を CD-ROM ドライブに挿入します。

c. 以下を入力して、挿入されている CD-ROM を読取るように 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 (exitops) 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. 以下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> install cluster*
Inst> install cxfs*
Inst> install sysadm_base*
Inst> install sysadm_cxfs*
```

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
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_base がインストールされると、tcpmux サービスが有効になります。

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、接続先のプール・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てください。

5. CXFS の管理を実行するワークステーション（管理ワークステーションと呼び、クラスタ内またはクラスタ外のノードを使用できます）で IRIX デスクトップから 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_cxfs.man
Inst> install sysadm_cxfs.sw.client
Inst> install sysadm_cxfs.sw.desktop
Inst> go
```

注意: GUI は Java 1.1.8 でのみ動作します。これは、IRIX 6.5.*xf* 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 が失敗します。

6. 管理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 を起動する IRIX マシンの場合は、CXFS CD から `java_plugin` をインストールします(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、ブラウザを再起動してください。

   IRIX 以外のワークステーションに対しては、上記の URL から Java Plug-in をダウンロードします。Java Plug-in がインストールされていない場合にブラウザから GUI を実行すると、ブラウザはこのサイトにリダイレクトされます。

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` サブシステムとともに、`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 71 ページの「システムの再起動」で説明する手順で行います。

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。静的ルーティングが推奨されています。

この節では、必要な必須の変更とオプションの変更について説明します。

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。

CXFS クラスタには、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 ファイルの形式は以下のとおりです。*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

  2 つのネットワークを用意することが推奨されています。CXFS が使用するのはハートビート/コントロール用のプライマリ・ネットワークだけですが、2 つのネットワークを用意しておくと、必要な場合にリセット・デーモンが 2 番目のネットワークにフェイルオーバーできます。リセット・デーモンは、リセット要求を異常ノードの適切なデーモンに転送し、ネットワークを通じてリセット・ハードウェアにアクセスできる場合は、そのハードウェアにアクセスします。

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 ファイルは、ホスト名 color-green または完全修飾ドメイン名 color-green.sgi.com で指定できます。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」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」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:

  ```
  hosts:      files nis dns
  ```

  (nis と dns の順序は CXFS では重要ではありませんが、files を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については、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 ファイルにそのホストの IP/ホスト名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。235 ページの「クラスタ設定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は、/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。

/etc/services

プール内の各ノードに cluster_admin 製品をインストールする前に、各ノードの /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 つを両方ともインストール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は、sgi-cmsd は必要ありません。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
```

/etc/config/cad.options

/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	詳細レベル。v という文字の数は、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は、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。

cmgr(1M) コマンドまたは GUI を使用して /etc/config/cad.options ファイルの内容を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、56 ページの「ソフトウェアのインストール」を参照してください。

/etc/config/fs2d.options

各ノードの /etc/config/fs2d.options ファイルには、cad プロセスの開始時にこの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ノードへのクラスタ設定データベース (CDB) 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。
-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。

オプション	説明
-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。
-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、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、新しいファイル /var/cluster/ha/log/fs2d_ops1 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、各ノードの /.rhosts ファイルでクラスタ内のすべてのホスト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・ノードに対して 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、以下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も、ノードのすべての IP アドレスとホスト名をすべてのノードの `/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。

   `hosts`(4)、`named`(1M)、`dns`(7P)、および `nis`(7P) のマン・ページを参照してください。『IRIX Admin Networking and Mail』および『NIS Administrator's Guide』も参照してください。

   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。

   ---

   60 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` に追加します。

例:

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

$HOSTNAME は、/etc/hosts で設定されている IP アドレスのエイリアスです。60 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は、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 で設定されている 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 個未満の場合、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に、各ノードの /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:

   ```
   -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 ファイルに以下の行を追加します。

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

   cxfs2 で、/usr/lib/aliases ファイルに以下の行を追加します。

   ```
   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 が各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) コマンドを実行してください。

   ---

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。

シリアル・ポートの設定

ノードが別のマシンをリセットするように設定されている場合は、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ノードで実行してください。これを行うには、各ノードで以下の手順を実行します。

1. リセット・シリアル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マルチノード・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。

システムの再起動

各ノードで以下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。

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」
- 72 ページの「シリアル・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

プール内の各ノードの各プライベート・ネットワークに対して、以下のコマンドを入力します。*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` が失敗する場合は、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。

各ノードでこの手順を繰返します。

シリアル・リセット接続

シリアル・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. ノードとシリアル・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. プール内のノードの 1 つで cmgr(1M) コマンドを開始します。

   # **cmgr**

3. 各ノードで HA サービスを停止します。

   stop cx_services for cluster *clustername*

   例:

   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、以下を入力します。

     test serial in cluster *clustername*

     例:

```
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、以下を入力します。

  test serial in cluster *clustername* node *machinename*

  例:

```
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、以下を入力します。

  admin ping node *nodename*

  例:

```
cmgr> admin ping node cxfs7

ping operation successful
```

5. コマンドが失敗する場合は、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

IRIX 6.5.13f のリリースから、CXFS ファイルシステム設定の構造が変更されています。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。6.5.13f と 6.5.14f を混在させて実行する予定の場合は、逆互換性を使用しないようにしてください。

6.5.12f 以前からのアップグレード

メモ: 6.5.13f 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くなります。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 ノードの 1 つで以下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、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。

6.5.13f ノードと 6.5.14f ノードが混在する状態での実行

クラスタ内で 6.5.13f ノードと 6.5.14f ノードを混在させて実行する予定の場合は、6.5.13f を実行するノードが以前の CXFS ファイルシステム定義との逆互換性を持たないようにする必要があります（逆互換性を持つと、クラスタ内の一部のノードだけが逆互換性を強制的に適用することになり、ファイルシステム定義が失われたり、破壊されたりする可能性があります）。

6.5.13f ノードの逆互換性をオフにするには、クラスタ内の任意のノードから以下のコマンドを一度だけ実行します。

```
# /var/cluster/cmgr-scripts/cxfs_backwards_fs_compat off
```

すべてのノードが 6.5.14f 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ノードを 6.5.14f 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、以下のコマンドを入力します。

```
# /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に必要な手順の概要を説明します。237 ページの付録 A「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できます。既存の IRIS FailSafe クラスタから切替える場合は、94 ページの「GUI を使った既存の FailSafe クラスタを CXFS 用に再設定」を参照してください。

この章は、55 ページの 第2章「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・タスクセットを使用することが強く推奨されています。78 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、193 ページの 第8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。56 ページの「ソフトウェアのインストール」を参照してください。

準備手順

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。

- 76 ページの「ライセンスの確認」
- 76 ページの「クラスタ・デーモンの開始」
- 76 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」
- 77 ページの「ノードのホスト名の判断」
- 78 ページの「`chkconfig` フラグがオンであることの確認」

設定作業中は、ログ・ファイルにさまざまな情報専用メッセージが示されます。213 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.
```

適切にインストールされたライセンスがある場合は、各ノードの /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

以下のコマンド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

ソフトウェアをはじ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、以下のコマンドを入力します。

```
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てください。デーモンを再開する必要がある場合は、以下のコマンドを入力します。

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

詳細については、235 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 24 ページの「デーモン」を参照してください。

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りません)を使用する必要があります。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。60 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、およびプールの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。56 ページの「ソフトウェアのインストール」を参照してください。

chkconfig フラグがオンであることの確認

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:

```
# 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。

- 75 ページの「準備手順」
- 78 ページの「GUI の起動」
- 80 ページの「GUI を使った新規クラスタの設定」
- 81 ページの「GUI を使った新規ファイルシステムの設定」

クラスタの表示は、どのノードから GUI を実行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、ツリー表示に変更内容が表示されるまで待ってください。ツリー表示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

以下を入力して、CXFS マネージャを起動します。

```
# /usr/sbin/cxtask
```

接続先のノードの名前と root パスワードを入力します。

注意: プールの最初のノードでは、「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。60 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。

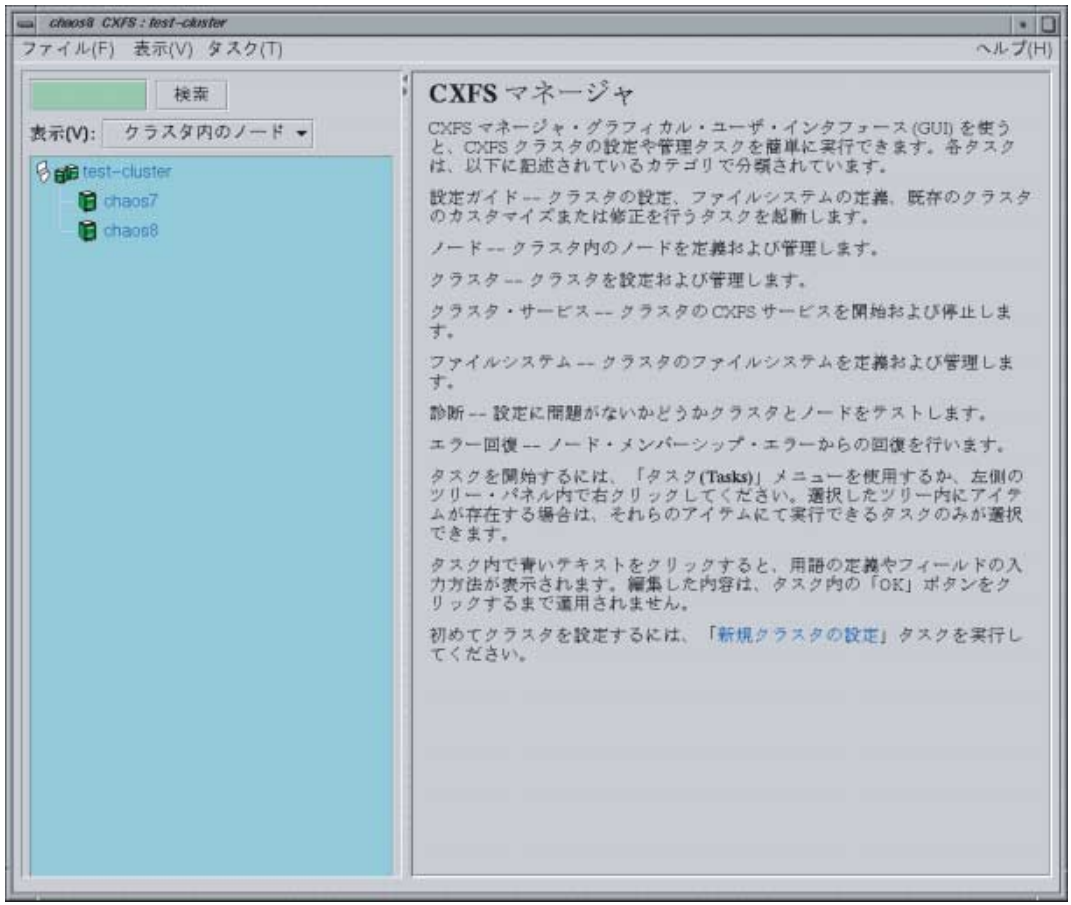

図3-1 CXFS マネージャ

GUI の起動方法はほかにもあります。詳細については、42 ページの「クラスタ・マネージャのツール」を参照してください。

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。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。

「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)」タスクセットを使用すると、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。このタスクセットには、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。77 ページの「ノードのホスト名の判断」を参照してください。95 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)左側のツリー表示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」をクリックして、クラスタのその他のノードを定義します。ホスト名 /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。95 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: ツリー表示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、重み付けされた最初の 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ます。

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。106 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。ツリー表示にクラスタが表示されていることを確認します。ツリー表示で「表示(View)」->「クラスタ内のノード(Nodes in Cluster)」を選択します。

4. ツリー表示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。101 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。117 ページの「GUI を使った接続性のテスト」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。66 ページの「（オプション）/.rhosts」を参照してください。

6. 「CXFS サービスの開始(Start CXFS Services)」をクリックします。109 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。クラスタとノードがツリー表示に緑色で表示されたら、クラスタは動作可能になっています。

7. 「閉じる(Close)」をクリックします。「閉じる」をクリックすると、タスクセットは終了します。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。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。

「新規ファイルシステムの設定(Set Up a New Filesystem)」タスクセットを使用すると、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。このタスクセットには、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）。109 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. クラスタ内のいずれかのノードのシェル・ウィンドウを開き、fx(1M) コマンドを使用して、ボリュームで使用するディスク・ドライブのボリューム・ヘッダを作成します（この手順に対しては GUI は使用しません）。詳細については、fx(1M) のマン・ページと、『IRIX Admin: Disks and Filesystems』のディスク管理手順の実行に関する章を参照してください。

3. シェル・ウィンドウで XVM 論理ボリュームを作成します。xvm(1M) probe コマンドを使用して、ラベルの付いていないボリュームのリストを表示できます。詳細については、『XVM Volume Manager Administrator's Guide』を参照してください。

4. シェル・ウィンドウで、mkfs(1M) コマンドを使用して CXFS ファイルシステムを作成します。『IRIX Admin: Disks and Filesystems』を参照してください。

5. GUI に戻り、「ファイルシステムの定義(Define a Filesystem)」をクリックします。113 ページの「GUI を使ったファイルシステムの定義」を参照してください。

6. 「ファイルシステムのマウント(Mount a Filesystem)」をクリックします。114 ページの「GUI を使った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・タスクセットを使用することが強く推奨されています。78 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. 75 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 /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。121 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 ID[optional]?
Partition ID[optional] ? (0)
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)
Number of Network Interfaces ? (1)
NIC 1 - IP Address ? cxfs6
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
Node Weight ? (1)

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 ID[optional] ?
Partition ID[optional] ? (0)
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
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
Node Weight ? (1)

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 ID[optional] ?
Partition ID[optional] ? (0)
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
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
Node Weight ? (1)

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、以下のコマンドを入力します。

```
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、以下のコマンドを入力します。

```
cmgr> show node cxfs6

Logical Machine Name: cxfs6
Hostname: cxfs6.americas.sgi.com
Node Is FailSafe: false
```

```
Node Is CXFS: true
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。134 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
```

詳細については、134 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は、157 ページの「スクリプトの例」を参照してください。

4. 以下を入力して、クラスタの CXFS サービスを開始します。

   `start cx_services for cluster` *clustername*

   例:

   ```
   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に設定されます。

   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、以下の `cmgr`(1M) コマンドを使用できます。

   `show status of cluster` *clustername*

例:

```
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、139 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. シェル・ウィンドウで XVM 論理ボリュームを作成します。詳細については、『XVM Volume Manager Administrator's Guide』を参照してください。
7. シェル・ウィンドウで、mkfs(1M) コマンドを使用して ファイルシステムを作成します。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。144 ページの「cmgr を使ったファイルシステム・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にもかかわらず、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/dks2d76s0
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/dks2d77s0
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、以下のコマンドを入力します。

```
cmgr> quit
```

GUI リファレンス

この章では、以下のタスクを実行するために 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) を使用する方法について説明します。

- 「設定ガイド・タスクセット」
- 95 ページの「GUI を使ったノード・タスク」
- 106 ページの「GUI を使ったクラスタ・タスク」
- 109 ページの「GUI を使った CXFS サービス・タスク」
- 113 ページの「GUI を使ったファイルシステム・タスク」
- 117 ページの「GUI を使った診断タスクとエラー回復」

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、56 ページの「ソフトウェアのインストール」を参照してください。

設定ガイド・タスクセット

設定ガイド・タスクセットには、以下のタイプがあります。

- 80 ページの「GUI を使った新規クラスタの設定」
- 81 ページの「GUI を使った新規ファイルシステムの設定」
- 94 ページの「GUI を使った既存の FailSafe クラスタを CXFS 用に再設定」
- 186 ページの「GUI を使ったクラスタのステータスの確認」
- 94 ページの「既存のクラスタの変更」
- 95 ページの「クラスタ・ノードの修正またはアップグレード」

GUI を使った既存の FailSafe クラスタを CXFS 用に再設定

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。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。

「既存の FailSafe クラスタを CXFS 用に再設定(Set Up an Existing FailSafe Cluster for CXFS)」タスクセットを使用すると、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。このタスクセットには、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、このタスクセットは 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 への切り替え(Convert a FailSafe Cluster to 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。108 ページの「GUI を使った FailSafe クラスタの CXFS への切替え」を参照してください。
2. FailSafe GUI を使用して、切替える高可用性 (HA) サービスをノードで停止します。『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。103 ページの「GUI を使ったノードの変更」を参照してください。
4. 「FailSafe ノードの CXFS への切り替え(Convert a FailSafe Node to 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。105 ページの「GUI を使った FailSafe ノードの CXFS への切替え」を参照してください。
5. 「FailSafe ノードの CXFS への切り替え(Convert a FailSafe Node to 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

このタスクセットでは、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。クラスタ管理者にイベントを通知する方法の変更など、ほとんどの変更は、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。

- 108 ページの「GUI を使ったクラスタ定義の変更」
- 81 ページの「GUI を使った新規ファイルシステムの設定」
- 115 ページの「GUI を使ったファイルシステムの変更」
- 95 ページの「GUI を使ったノードの定義」
- 117 ページの「GUI を使った接続性のテスト」
- 101 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

このタスクセットを使用すると、メインテナンスを行うためにクラスタからノードを削除する手順を実行できます。

このタスクセットは、以下の手順で構成されています。

- 110 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。102 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。
- 109 ページの「GUI を使った CXFS サービスの開始」
- ツリー表示で、クラスタのコンポーネントの状態をモニタします。186 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・タスクセットを使用すると、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。80 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ログインしたノードでなければなりません。最初のノードを定義するには、`/etc/sys_id` に表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。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。

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。プール内で定義されている最初のノードに対しては、`/etc/sys_id` に表示されているホスト名を使用しなければなりません。これは、最初のノードに対してはデフォルトで入力されています。`/etc/sys_id` ファイルに表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。60 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。

3. 「メンバーシップ重み(Membership Weight)」: 0 または 1 で、CXFS メンバーシップ定義でのノードの重みを指定します。CXFS メンバーシップを決定するときにノードが考慮に入れられないようにする場合は、0 を使用します。デフォルトは 1 です。少なくとも 1 つのノードの重みに 1 を指定しなければなりません。0 または 1 以外の値は推奨されません。

   CXFS メンバーシップ上限量を形成するためには、重み付けされたノードの **50%** 以上が初期クラスタに必要です。クラスタを維持するには、重み付けされたノードの **50%** が実行中でなければなりません。詳細については、17 ページの「上限量の計算とノードの重み」を参照してください。

4. 「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、60 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、9 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。ただし、優先度が 1 のネットワークに

異常が発生した場合でもクラスタ・インフラストラクチャがリセット用として使用できる追加のネットワークを定義できます。

- 「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: GUI では「ハートビート(Heartbeat)」または「コントロール(Control)」だけを指定できますが、これらの設定は CXFS では使用できないので選択しないでください。CXFS は、ハートビートとコントロールの両方に対して単一のネットワークを使用します。

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、6 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、「削除(Delete)」をクリックします。

5. 「ノード ID(Node ID)」: プール内のノード全体で一意な 1〜32767 の範囲の整数(オプション)。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。ノードを定義したら、ノード ID 番号は変更しないでください(デフォルトの CXFS タイブレーカーがなくなります。詳細については、17 ページの「メンバーシップ上限量」を参照してください)。

6. 「パーティション ID(Partition ID)」: 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(オプション)。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てください。パーティション ID の値を決定するには、`mkpart`(1M) コマンドを使用してください(以下を参照してください)。

   次の画面に移動するには、「次へ(Next)」をクリックします。

7. ノードをリセットするためにシステム・コントローラ・ポートを使用するかどうかを選択できます。ノードをリセットするために CXFS でシステム・コントローラを使用できるようにする場合は、「リセット・パラメータの設定(Set Reset Parameters)」チェックボックスをオンにし、次の情報を指定します。

   - このノード:

     – 「ポート・タイプ(Port Type)」: プルダウン・リストから、「L2」、「MSC」(モジュール・システム・コントローラ)、または「MMSC」(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・タイプは、ノード・タイプによって異なります。ノードに使用されるポート・タイプについての詳細は、「ポート・タイプ(Port Type)」という青いテキストをクリックしてください。

  - 「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。

  - 「ポートを一時的に無効にする(Temporarily Disable Port)」: 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)」: リモート・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。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。

リセット機能をまったく使用しない場合は、「リセット・パラメータの設定(Set Reset 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)。

8. 「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、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ツリー表示で表示されるまで、2 番目のノードは追加しないでください。設定データベースに変更を加えると、その都度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、設定が大き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。

図4-1 ノード定義の例

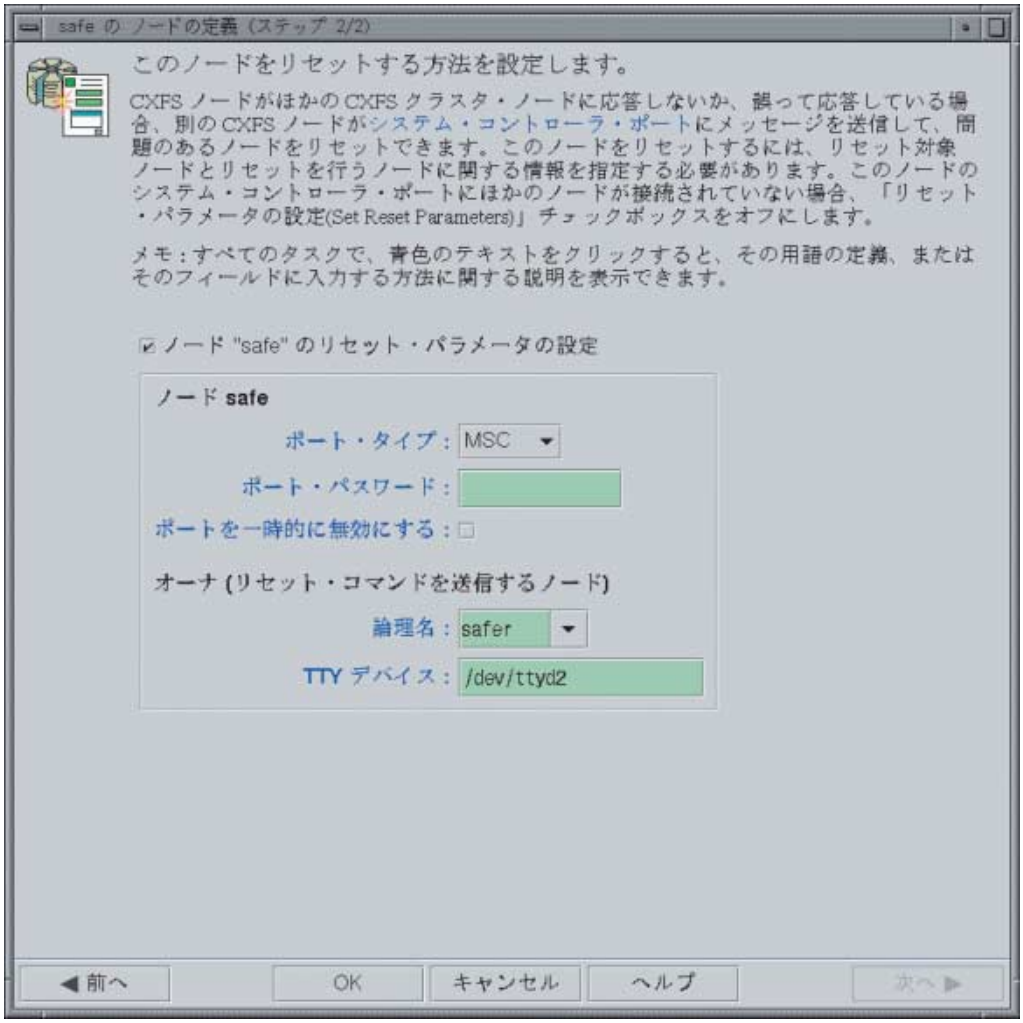

図4-2 システム・コントローラ設定の例

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・アイコンがツリー表示に表示されるまで、ノードを追加したり削除しないでください。ツリー表示は、「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(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ノードをリセットできます。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 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 を使ったノードの変更

ノードを定義したら、そのノードを変更できます。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。

ノードを変更するには、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. 「メンバーシップ重み(Membership Weight)」: CXFS メンバーシップ重みを変更する場合は、テキスト入力領域をクリックし、<Backspace> キーを使用して現在の値を削除します。次に、正しい値を追加します。0 または 1 の値のみを使用してください。詳細については、17 ページの「上限量の計算とノードの重み」を参照してください。

   メモ: CXFS では、ノードが現在クラスタでアクティブな場合、CXFS メンバーシップ重みまたはネットワークの変更は直ちに反映されません。変更は、次回ノードをクラスタに設定する際(つまり、再起動の後、または CXFS サービスを停止して再開した後)に考慮されます。

   ---

3. 「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。ただし、優先度が 1 のネットワークに異常が発生した場合でもクラスタ・インフラストラクチャがリセット用として使用できる追加のネットワークを定義できます。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場合は、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」をクリックします。

   - ネットワークを削除する場合は、優先度リストでそのネットワークを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場合は、優先度リストでそのネットワークを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: GUI では「ハートビート(Heartbeat)」または「コントロール(Control)」だけを指定できますが、これらの設定は CXFS では使用できないので選択しないでください。CXFS は、ハートビートとコントロールの両方に対して単一のネットワークを使用します。

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、6 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)」: 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(オプション)。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てください。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)。

4. 次のページに移動するには、「次へ」をクリックします。必要に応じて、値を変更します。
5. 「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 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 ノード(タイプ `FailSafe`)を `CXFS and FailSafe` または `CXFS` のいずれかのタイプに切替えるには、以下を実行します。

1. FailSafe GUI を使用して、切替える HA サービスをノードで停止します。『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。103 ページの「GUI を使ったノードの変更」を参照してください。
3. 次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。
   - 「メンバーシップ重み(Membership Weight)」: CXFS メンバーシップ重みを変更する場合は、テキスト入力領域をクリックし、<Backspace> キーを使用して現在の値を削除します。次に、正しい値を追加します。0 または 1 の値のみを使用してください。詳細については、17 ページの「上限量の計算とノードの重み」を参照してください。
   - 「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、103 ページの「GUI を使ったノードの変更」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を確認してください。

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、108 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 を使ったノードの表示

ノードを定義したら、ツリー表示の「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の表示
- 「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の表示

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

この節では、クラスタを定義、変更、削除、および表示するために個々の GUI タスクを実行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・タスクセットを使用すると、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。80 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連結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. 次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。

- 「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、IRIX 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。

- 「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 の CXFS 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. 次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、106 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、106 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、101 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 番号を流用してください。

GUI を使った FailSafe クラスタの CXFS 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. 次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、IRIX 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、101 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、108 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。

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、要素を選択してチェックできます。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。ステータスと設定の情報が右側のアイテム表示に表示されます。

また、`cluster_status` コマンドを使用することもできます。183 ページの 第7章「ステータスのモニタ」を参照してください。

GUI を使った CXFS サービス・タスク

以下のタスクでは、CXFS クラスタ・サービスを開始および停止する方法と、ログを設定する方法について説明します。

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. 次の情報を入力します。

   - 「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、使用可能なメタデータ・サーバ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。

   - 「強制(Force)」: エラーが発生しても（通常はエラーが発生して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。

2. 「OK」をクリックします。プロセスを完了するには、数分かかります。

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、(CXFS メンバーシップ重みが等しい)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 メンバーシップ上限量が維持されるかどうかを決定します。CXFS タイブレーカー・ノードは、必ず重み付けされている必要があります。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、17 ページの「メンバーシップ上限量」を参照してください。

注意: 重み付けされた 2 ノード・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、SGI では、すべての CXFS システム(特に、重み付けされた 2 ノード・クラスタ)に対してリセット・ハードウェア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、以下を実行してください。

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、ログをカスタマイズできます。

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。

183 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。

   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。174 ページの「ログ・ファイル管理」、63 ページの「`/etc/config/cad.options`」、および 63 ページの「`/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。142 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 を使ったファイルシステム・タスク

以下のタスクでは、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:「新規ファイルシステムの設定(Set Up a New Filesystem)」設定ガイド・タスクセットを使用すると、新しい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。81 ページの「GUI を使った新規ファイルシステムの設定」を参照してください。

GUI を使った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、78 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(144 ページの「cmgr を使った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・サーバとして動作するノードのリスト。サーバのリストに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。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、リストでその名前を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、または次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。

2. 「選択されたノード(Selected Nodes)」: 上で「選択されたノードのみ(Only Selected Nodes)」を選択した場合、希望のノードを「ノード(Node)」リストから選択できます。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。
3. 「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)」: 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がマウントされるようにする場合、このオプションを選択します。これはデフォルトで選択されます。
4. 「OK」をクリックします。

GUI を使った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、109 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)」: 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。
   - 「強制(Force)」: `umount -k` オプションを使用して強制的にアンマウントするには、「強制」切替えボタンをクリックします。`-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。

GUI を使ったファイルシステムの変更

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。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、または下で指定して選択されたノードのリストのいずれか。
- 「選択されたノード(Selected Nodes)」: 上で「選択されたノードのみ(Only Selected Nodes)」を選択した場合、希望のノードを「ノード(Node)」リストから選択できます。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。
- 「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)」: 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がマウントされるようにする場合、このオプションを選択します。これはデフォルトで選択されます。

2. 「OK」をクリックします。

GUI 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: 実装は延期されています。

ファイルシステムのメタデータ・サーバは、リスト内のその他の使用可能なメタデータ・サーバに再設定できます。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。

- 次の情報を入力します。
  - 「CXFS ファイルシステム(CXFS 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。

    選択されたサーバは、「CXFS サービスの開始(Start CXFS Services)」タスクを実行した後は、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。109 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。
- 「OK」をクリックして、タスクを完了します。

ノード・メンバーシップでは、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。

GUI を使った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、115 ページの「GUI を使った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。

GUI を使った診断タスクとエラー回復

この節では、GUI を使用して接続性をテストする方法を説明します。ネットワークおよびクラスタ内のノードのシリアル接続は、要求された入力値を入力することによってテストできます。クラスタ内のノードをすべて同時にテストしたり、個々のノードを指定してテストできます。

GUI を使った接続性のテスト

「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 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 (168 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 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 メンバーシップ上限量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 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、168 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 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。

- 117 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 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。

cmgr リファレンス

以下のいずれかを入力できます。

```
# /usr/cluster/bin/cmgr
# /usr/cluster/bin/cluster_mgr
```

-p オプションを使用して、ほかのプロンプトを表示することもできます。cmgr の使用についての一般情報は、48 ページの「cmgr(1M) の概要」を参照してください。

この章では、以下のタスクを実行するために cmgr(1M) コマンドを使用する方法について説明します。

- 120 ページの「cmgr を使ったデフォルトの設定」
- 120 ページの「cmgr を使ったノード・タスク」
- 134 ページの「cmgr を使ったクラスタ・タスク」
- 140 ページの「cmgr を使った CXFS サービス・タスク」
- 144 ページの「cmgr を使ったファイルシステム・タスク」
- 156 ページの「cmgr を使った診断タスクとエラー回復タスク」
- 157 ページの「スクリプトの例」

この章では、スクリプト内で cmgr(1M) コマンドを使用する例も示します。157 ページの「スクリプトの例」を参照してください。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、93 ページの 第4章「GUI リファレンス」を参照してください。

タスクは、特定の階層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、文字端末モードでステータスを表示することもできます。183 ページの 第7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、56 ページの「ソフトウェアのインストール」を参照してください。

---

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:

```
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、設定が大き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 partition_id to partitionID
    set reset_type to powerCycle
    set sysctrl_type to msc|mmsc|l2_(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)
```

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。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。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。

- *partitionID* 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、このコマンドを使用する必要はありません。

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
```

システムにパーティションが設定されていない場合は、このコマンドを省略するか、または 0 という値を使用できます。

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。

- powerCycle は、唯一使用可能なリセット・タイプです。
- システム・コントローラ・タイプは、表5-1 に示すように、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

msc	mmsc	l2
Origin 2000 Deskside	Origin 2000 Rackmount	SGI Origin 3400
Origin 200	SGI 2400	SGI Origin 3800
SGI 2100	SGI 2800	SGI Origin 3200（オプション L2 装備）
SGI 2200		

- *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。
- *enabled*|*disabled* 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 でノードをリセットできない）に設定することにより一時的に無効にできます。CXFS がノードをリセットできるようにするには、`disabled` と入力します。
- システム・コントローラ・デバイスに対しては、`/dev/ttyd2` だけが正しい値です。
- *node_sending_reset_command* 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・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。
- *tty_device* 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。
- ノードの重みは 0 または 1 にします。デフォルトは 1 です。少なくとも 1 つのノードの CXFS メンバーシップ重みに、0 より大きい値を設定する必要があります。

- *IP_address_or_hostname_(if_DNS)* 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。9 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、60 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、9 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。

詳細については、17 ページの「上限量の計算とノードの重み」および 95 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、95 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 ID ? [optional] node_ID
Partition ID ? [optional] (0)partition_ID
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>? (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
Number of Network Interfaces ? (1) 1|2|3|4|5|6|7|8
NIC 1 - IP Address ? IP_address_or_hostname_(if_DNS)
NIC 1 - Heartbeat HB (use network for heartbeats) <true|false> ? true|false
NIC 1 - (use network for control messages) <true|false> ? true|false
NIC 1 - Priority <1,2,...> ? priority_number
Node Weight ? (1) 0|1
```

通常モードの場合:

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

cmgr> define node cxfs6
Enter commands, when finished enter either "done" or "cancel"

cxfs6 ? set is_failsafe to false
cxfs6 ? set is_cxfs to true
cxfs6 ? set weight to 1
cxfs6 ? add nic cxfs6
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
cxfs6 ? done
```

プロンプト・モードの場合:

```
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 ID[optional]?
Partition ID ? [optional] (0)
Reset type <powerCycle>  ? (powerCycle)
Do you wish to define system controller info[y/n]:n
Number of Network Interfaces ? (1)
NIC 1 - IP Address ? cxfs6
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
Node Weight ? (1)

Successfully defined node cxfs6
```

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_(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)
```

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、121 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。

cmgr を使ったノードの重みの変更例

0 というノードの重みを変更するには、以下のコマンドを入力します。現在の設定を維持するには、<Enter> キーを押します（例では、<Enter> キーは示されていません）。

```
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)
Is this a CXFS node <true|false> ? (true)
Node ID[optional] ? (13203)
```

```
Partition ID[optional] ? (0)
Reset type <powercycle> ? (powerCycle)
Do you wish to modify system controller info[y/n]:n
Number of Network Interfaces ? (1)
NIC 1 - IP Address ? (cxfs6)
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)
Node Weight ? (1) 0

Successfully modified node cxfs6
```

以下に結果を示します。

```
cmgr> show node cxfs6
Logical Machine Name: cxfs6
Hostname: cxfs6.americas.sgi.com
Node Is FailSafe: false
Node Is CXFS: true
Nodeid: 13203
Reset type: powerCycle
ControlNet Ipaddr: cxfs6
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 1
Node Weight: 0
```

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
Reset type <powerCycle> ? (powerCycle)
Do you wish to modify system controller info[y/n]:n
Number of Network Interfaces ? (2)
NIC 1 - IP Address ? (preston)
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)
NIC 2 - IP Address ? (192.168.168.1)
NIC 2 - Heartbeat HB (use network for heartbeats) <true|false> ? (true)
NIC 2 - (use network for control messages) <true|false> ? (true)
NIC 2 - Priority <1,2,...> ? (2)
Node Weight ? (1)

Successfully modified node n_preston

cmgr> show node n_preston
Logical Machine Name: n_preston
Hostname: preston.engr.sgi.com
Node Is FailSafe: true
Node Is CXFS: true
Nodeid: 606
Partition id: 1
Reset type: powerCycle
ControlNet Ipaddr: preston
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 1
ControlNet Ipaddr: 192.168.168.1
ControlNet HB: true
```

```
ControlNet Control: true
ControlNet Priority: 2
Node Weight: 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。

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
```

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) 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)
Reset type <powerCycle> ? (powerCycle)
Do you wish to modify system controller info[y/n]:n
Number of Network Interfaces ? (1)
NIC 1 - IP Address ? (cxfs6)
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)
Node Weight ? (0)

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
FileSystem Device Name: /dev/cxvm/dks2d72s0
FileSystem Mount Point: /mnts/fs1
FileSystem Mount Options:
FileSystem Status: enabled
FileSystem Force flag: false
        FileSystem Server Node: cxfs6
        FileSystem Server Rank: 0
```

```
        FileSystem Server Node: cxfs7
        FileSystem Server Rank: 1
FileSystem Device Name: /dev/cxvm/dks2d73s0
FileSystem Mount Point: /mnts/fs2
FileSystem Mount Options: enabled
FileSystem Status: enabled
FileSystem Force flag: false
        FileSystem Server Node: cxfs8
        FileSystem Server Rank: 0


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)
Number of Cluster FileSystems ? (0)

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:

```
cmgr> show node cxfs6
Logical Machine Name: cxfs6
Hostname: cxfs6.americas.sgi.com
Node Is FailSafe: false
Node Is CXFS: true
Nodeid: 13203
Reset type: powerCycle
ControlNet Ipaddr: cxfs6
ControlNet HB: true
ControlNet Control: true
ControlNet Priority: 1
Node Weight: 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:

```
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:

```
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:

```
cmgr> set cluster cxfs6-8
cmgr> show nodes

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

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、106 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。

- *clustername* 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 に設定してください。
- *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、IRIX 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。
- *notify_command* 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。
- *email_address* 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、常に 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 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:

```
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
```

クラスタ ID 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、IRIX 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。

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、134 ページの「cmgr を使ったクラスタの定義」および 106 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。

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 cluserid to clusterID
```

clusterID の値は、既存の FailSafe クラスタを CXFS に切替える際に指定する必要があります。ネットワーク内で一意な 1～128 の範囲の数字を指定してください。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、IRIX 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、以下を入力します。

```
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)
Number of Cluster FileSystems ? (0) 0

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:

```
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 CXFS 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:

```
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。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。

```
stop cx_services [on node hostname]for cluster clustername[force]
```

例:

```
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、(重みが等しい)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 メンバーシップ上限量が維持されるかどうかを決定します。CXFS タイブレーカー・ノードは、必ず重み付けされている必要があります。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: 重み付けされた 2 ノード・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、SGI では、すべての CXFS システム(特に、重み付けされた 2 ノード・クラスタ)に対してハードウェア・リセット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。

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 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 ""
```

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、111 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 hostname [in cluster clustername]
```

log_group 変数は、以下のいずれかになります。

```
clconfd
cli
crsd
diags
```

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。

log_level 変数には、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。

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、10 になるようログ・レベル cli を変更するには、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

  144 ページの「cmgr を使った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。150 ページの「cmgr を使った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 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
```

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_filesystem_name* は、どんな論理名でもかまいません。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/dks2d1l5s10 の場合、GUI によって、dks2d1l5s10 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。

- *devicename*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。

- *mountpoint* 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。

- set dflt_local_status コマンドを使用すると、指定していないノードにファイルシステムをマウントできるかどうかを指定できます。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、150 ページの「cmgr を使ったファイルシステムのマウント」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、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/dks2d76s0
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/dks2d76s1
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/dks2d76s0
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/dks2d77s1
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、以下を入力します。

```
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。*nodename* を指定した場合は、ローカル・ステータスを有効にします。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。144 ページの「cmgr を使ったファイルシステム・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。144 ページの「cmgr を使ったファイルシステムの定義」および 151 ページの「cmgr を使った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
```

このファイルシステムに対するローカル・ステータスが enabled のすべてのノードに、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、140 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**

このファイルシステムに対するローカル・ステータスが enabled の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 を使ったファイルシステムの変更

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、144 ページの「cmgr を使った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/dks2d76s0
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/dks2d76s0
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/dks2d76s0
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/dks2d76s0)
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:実装は延期されています。

再設定を使用するには、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ている必要があります。メタデータ・サーバを別のノードに再設定するには、次のコマンドを使用します。

```
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
```

ノード・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、151 ページの「cmgr を使った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:

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

cmgr を使った診断タスクとエラー回復タスク

この節では、テストおよびエラー・タスクを実行する方法について説明します。

cmgr を使ったネットワークの接続性のテスト

クラスタでのネットワークの接続性をテストするには、cmgr を使用できます。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。66 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 hostname1 node hostname2 ...]
```

例:

```
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

72 ページの「シリアル・リセット接続」を参照してください。

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 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 メンバーシップ上限量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 メンバーシップを許可すると、そのノードに CXFS 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 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。

- 「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 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
```

168 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。

スクリプトの例

次のスクリプトによって、タイプ CXFS and FailSafe 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、60 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。

この例では、システム・コントローラ、FailSafe リソース、およびフェイルオーバー・ポリシー定義は取上げていません。これらのトピックについての詳細は、『IRIS FailSafe Version 2 Administrator's Guide』を参照してください。

```
#!/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 weight to 1
        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 weight to 1
        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 weight to 1
```

```
        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/dks2d76s0
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/dks2d76s1
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
```

管理とメインテナンス

オフライン管理タスクは、クラスタ内のすべてのノードから実行できます。ただし、ファイルシステムがマウントされている場合は、サーバから管理を実行しなければなりません。

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。

- 162 ページの「マウント操作前後のスクリプトの実行」
- 164 ページの「`fsr`(1M) の使用」
- 164 ページの「`find`(1) と `crontab`(1) の使用」
- 165 ページの「階層ストレージ管理 (HSM: Hierarchical Storage Management) システムの使用」
- 165 ページの「ファイルシステムのメタデータ・サーバの検出」
- 167 ページの「メタデータ・サーバの回復」
- 168 ページの「データベースと CXFS のシャットダウン」
- 173 ページの「再起動時のデータベースの再起動の回避」
- 174 ページの「ログ・ファイル管理」
- 175 ページの「ボリューム管理」
- 175 ページの「ディスク管理」
- 176 ページの「ファイルシステムのメインテナンス」
- 178 ページの「ダンプと復元」

- 180 ページの「クラスタ・データベースのバックアップと復元」
- 180 ページの「chkconfig フラグ」
- 181 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。73 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。

マウント操作前後のスクリプトの実行

clconfd デーモンは、以下のファイルが /var/cluster/clconfd-scripts ディレクトリに存在しているかどうかをチェックし、存在する場合は、指定した 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これらのファイルを実行します。

- cxfs-pre-mount
- cxfs-post-mount
- cxfs-pre-umount
- cxfs-post-umount

このような名前のスクリプトを作成し、それらのスクリプトを使用して、CXFS ファイルシステムが正常にマウントされた場合のファイルシステムの NFS エクスポートなど、必要なアクションに適用できます。実行可能ファイルは、スクリプトまたはバイナリの場合があります。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。

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。

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。

例: cxfs-pre-mount

```
#!/bin/sh
#/var/cluster/clconfd-scripts/cxfs-pre-mount

echo "Preparing to mount CXFS file system \"$1\"" >> /dev/console
MNTPNT=`mount | grep $1 | cut -f 3 -d" "`
if [ -n "${MNTPNT}" ] ; then
    /usr/etc/exportfs -u $MNTPNT
fi
```

例: cxfs-post-mount

```
#!/bin/sh
#/var/cluster/clconfd-scripts/cxfs-post-mount

if [ $2 -eq 0 ] ; then
    echo "Successfully mounted CXFS file system \"$1\"" >> /dev/console
    /usr/etc/exportfs -a
else
    echo "FAILED to mount CXFS file system \"$1\" (error=$2)" >> /dev/console
fi
exit $2
```

例: cxfs-pre-umount

```
#!/bin/sh
#/var/cluster/clconfd-scripts/cxfs-pre-umount

echo "Preparing to unmount CXFS file system \"$1\"" >> /dev/console
MNTPNT=`mount | grep $1 | cut -f 3 -d" "`
if [ -n "${MNTPNT}" ] ; then
    /usr/etc/exportfs -u $MNTPNT
fi
```

例: cxfs-post-umount

```
#!/bin/sh
#/var/cluster/clconfd-scripts/cxfs-post-umount

if [ $2 -eq 0 ] ; then
    echo "Successfully unmounted CXFS file system \"$1\"" >> /dev/console
else
    echo "FAILED to unmount CXFS file system \"$1\" (error=$2)" >> /dev/console
    /usr/etc/exportfs -a
fi
exit $2
```

fsr(1M) の使用

fsr(1M) コマンドは、操作対象の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現在のメタデータ・サーバだけに対して手動で使用してください。

find(1) と crontab(1) の使用

find(1) コマンドは、crontab(1) ファイル内に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。

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。

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。

メモ: 現在のリリースは、DMF などの HSM で使用される DMAPI ファイルシステムの再設定をサポートしていません。この機能は、今後のリリースで提供されます。

HSM を CXFS で使用するには、以下を実行します。

- クラスタ内の各ノードに、`eoe.sw.dmi` サブシステムをインストールします。
- 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、113 ページの「GUI を使ったファイルシステムの定義」または 136 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、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メタデータ・サーバの検出

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. ツリー表示で、表示するファイルシステムの名前をクリックします。メタデータ・サーバの名前が、右側のアイテム表示に表示されます。

図6-1 に、例を示します。

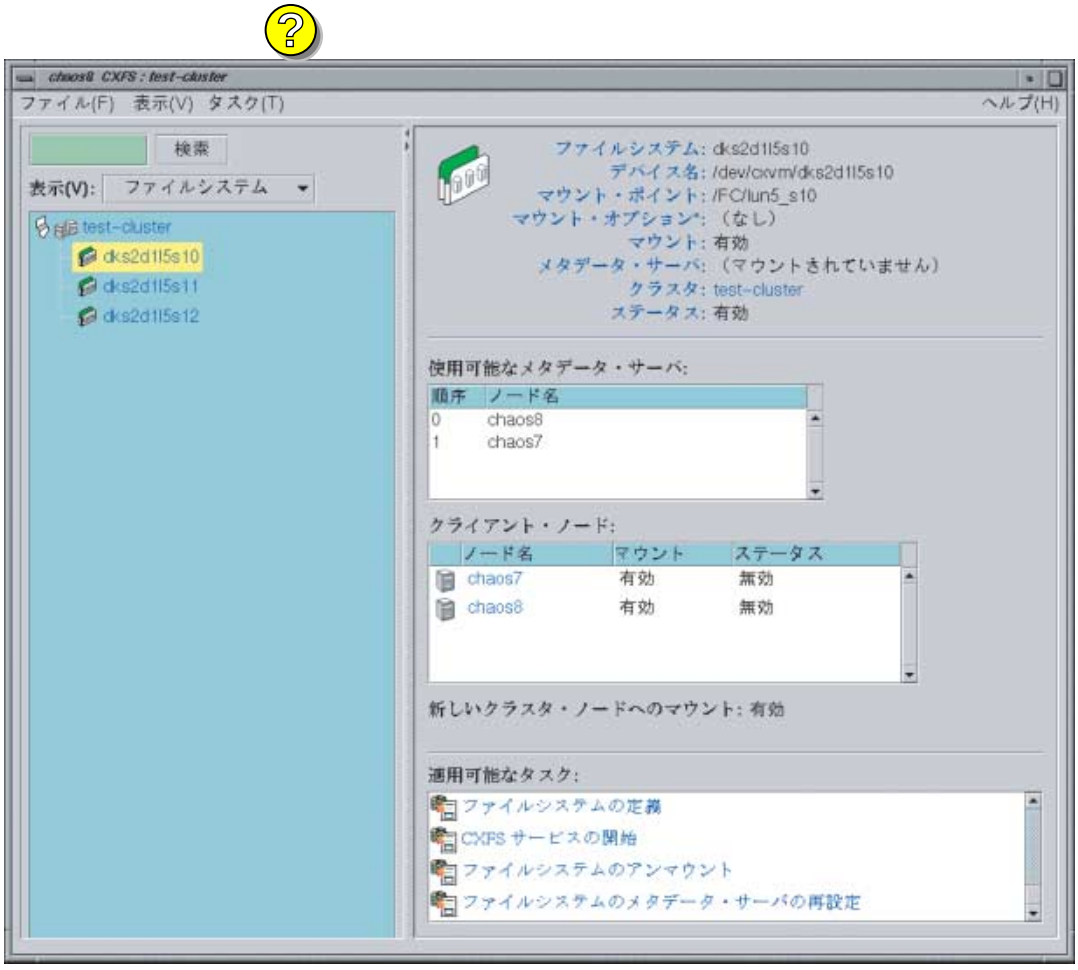

図6-1 メタデータ・サーバが表示されているウィンドウ

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、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、186 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現在のメタデータ・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: 実装は延期されています。

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、使用可能なメタデータ・サーバ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、使用可能なサーバが少なくとも 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、113 ページの「GUI を使ったファイルシステムの定義」または 136 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、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。

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 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 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。

- 169 ページの「クラスタ設定データベースのシャットダウン」
- 170 ページの「通常 CXFS シャットダウン」
- 172 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、193 ページの 第8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、183 ページの 第7章「ステータスのモニタ」を参照してください。

クラスタ設定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、クラスタ設定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設定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。

ノードがメタデータ・サーバの場合

ノードが CXFS ファイルシステムのメタデータ・サーバの場合、ノードを停止したままにするには、シャットダウンを完了する前にメタデータ・サーバを別のノードに移動します。

新しいメタデータ・サーバにファイルシステムを移動するには、以下を実行します。

1. メタデータ・サーバで `fuser`(1M) コマンドを実行し、ローカル・ユーザがいないことを確認します。
2. `umount`(1M) コマンドを使用して、メタデータ・サーバだけから CXFS ファイルシステムをアンマウントします(この場合以外は、`umount` を使用して CXFS ファイルシステムを手動でアンマウントしないでください)。

   ```
   # umount filesystem
   ```

   詳細については、115 ページの「GUI を使ったファイルシステムのアンマウント」、151 ページの「`cmgr` を使ったファイルシステムのアンマウント」、および 177 ページの「ファイルシステムのアンマウント」を参照してください。

ファイルシステムに定義されたメタデータ・サーバがほかにも存在すると想定して、再設定プロセスが自動的に開始されます。詳細については、113 ページの「GUI を使ったファイルシステムの定義」または 136 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、再設定プロセス中に遅れが発生します。

クラスタ設定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 メンバーシップ上限量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。

- すべてのファイルシステムがアンマウントされる
- クラスタ内のこのノードの CXFS 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。

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。110 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 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 メンバーシップの計算時に使用されなくなります。188 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 メンバーシップ上限量の計算に影響はありません。170 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 メンバーシップ上限量が失われた場合。

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。172 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（109 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 メンバーシップ上限量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)
- クラスタ・メンバーシップからこのノードが削除される
- 「停止(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 メンバーシップ上限量を下回った場合、クラスタに異常が発生する可能性があります。

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。この時点でセカンダリ・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、上限量が失われ、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。

上限量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。117 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 メンバーシップ上限量が失われた後、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、fs2d メンバーシップ上限量を計算するとき(これにより、クラスタのその他の部分の上限量が失われ、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。

- 101 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 136 ページの「cmgr を使ったクラスタの変更」
- 106 ページの「GUI を使ったノードの削除」または 130 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 メンバーシップを許可するまで、ローカル・ノードは CXFS 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。118 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 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:

```
cmgr> admin cxfs_start
```

リセット機能と強制 CXFS シャットダウン

注意: リセット機能を使用して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上限量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、22 ページの「ハードウェア・リセット」を参照してください。

再起動時のデータベースの再起動の回避

chkconfig(1M) に対する cxfs_cluster フラグは、clconfd デーモンを制御します。このフラグがオフの場合、clconfd 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。

以下を実行します。

```
# /etc/chkconfig cxfs_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、112 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。このスクリプトを定期的に実行するため、root crontab にエントリを加えることができます。

注意: 1 日にスクリプトを 2 回実行した場合、以前に保存されたコピーは上書きされます。

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
```

63 ページの「/etc/config/cad.options」および63 ページの「/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
- 連結
- 高度な回復機能

メモ: 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、クライアント・ノードまたはサーバ・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、NT マシンをクライアントに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 `/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設定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 113 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 134 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・タスクセットの一部)、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、81 ページの「GUI を使った新規ファイルシステムの設定」および 113 ページの「GUI を使った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、136 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、144 ページの「cmgr を使った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 modify を使用します。詳細については、115 ページ

の「GUI を使ったファイルシステムのアンマウント」または 151 ページの「cmgr を使った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。メタデータ・サーバでは、すべての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、115 ページの「GUI を使ったファイルシステムのアンマウント」または 151 ページの「cmgr を使った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。114 ページの「GUI を使ったファイルシステムのマウント」または 144 ページの「cmgr を使った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、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、CXFS メタデータ・サーバ・ノード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クライアントと 1 つのメタデータ・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する必要があります。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、クラスタ内のすべての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:

  – 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:

  – 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。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。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。

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。
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。ノードが上限量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、ノードはクラスタへの再設定を試みます。 • 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。

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。

- /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
- ツリー表示または 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』、『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。56 ページの「ソフトウェアのインストール」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(fs2d 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、174 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 または cluster_status、clconf_info、または cmgr コマンドを使用してモニタできます。

アイコンとカラーのキー

以下の表に、CXFS マネージャ GUI で使用されるアイコンとカラーを説明します。

表7-1 アイコンのキー

アイコン	実体
	ノード
	クラスタ

アイコン	実体
	ファイルシステム
	展開されているツリー
	縮小されているツリー

表7-2 カラーのキー

アイコン	カラー	状態
	灰色	非アクティブまたは不明(CXFS サービスがアクティブでない可能性があります)
	緑と白	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)
	緑	実行中またはマウント済み(エラーなし)
	赤	停止またはマウント済み(エラーあり)

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、ツリー表示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、188 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 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
```

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:

```
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、次のコマンドを使用します。

```
cmgr> show status of cluster cluster_name
```

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 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、17 ページの「メンバーシップ上限量」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 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。

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、ツリー表示を使用できます。「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。189 ページの図7-1に、タイプ `CXFS only` のノードの例を示します。

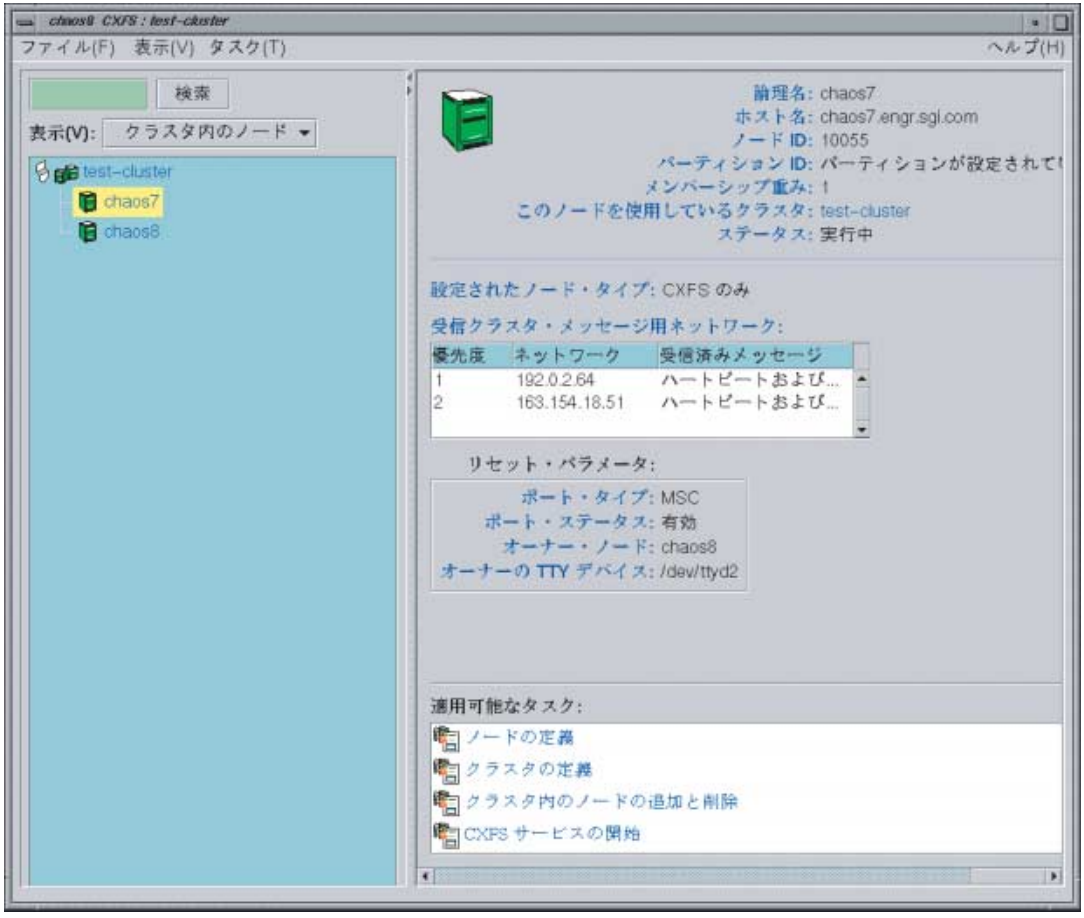

図7-1 ノードのステータス

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 を使ったリセット・シリアル回線のモニタ

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。56 ページの「ソフトウェアのインストール」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的なグラフに表示することもできます。図7-2 に、例を示します。

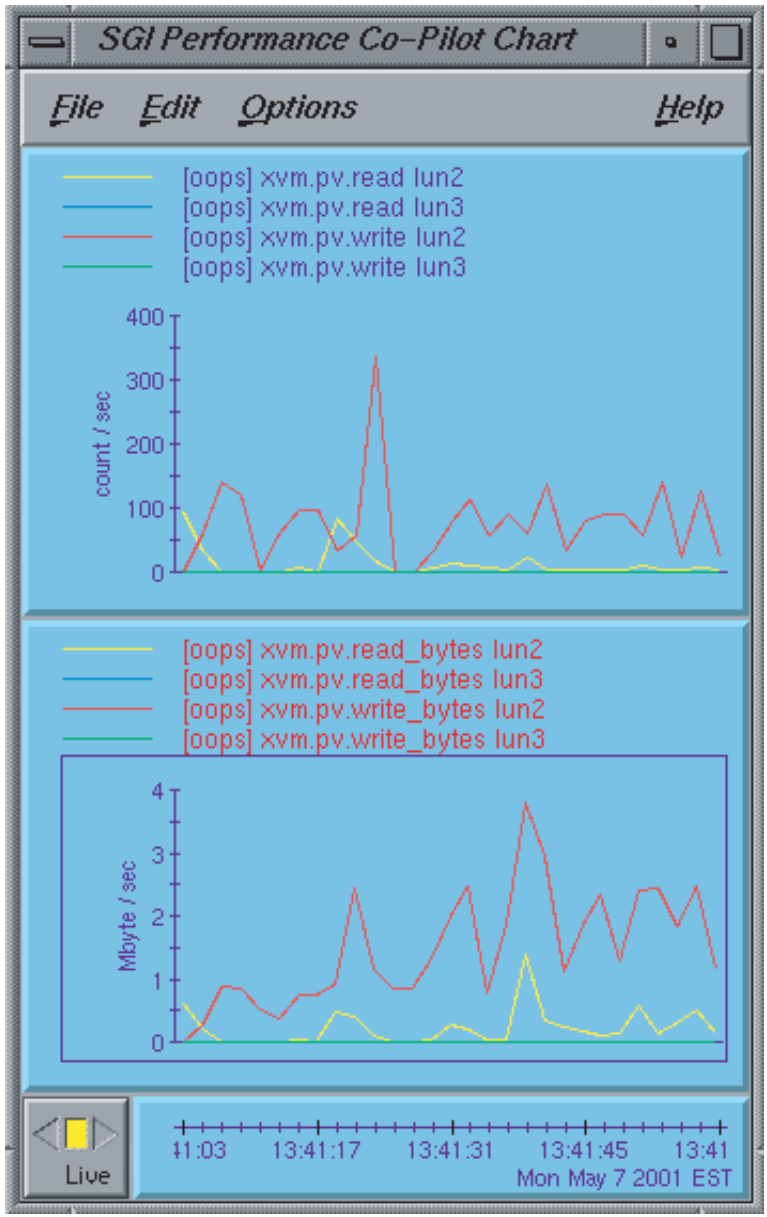

図7-2 pmgxvm グラフ

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」
- 203 ページの「問題の回避」
- 211 ページの「一般的な問題」
- 213 ページの「エラー・メッセージの理解」
- 231 ページの「修正処置」

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」
- 200 ページの「クラスタのステータスの識別」
- 201 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。

- 194 ページの「物理ストレージ・ツール」
- 194 ページの「クラスタ設定ツール」
- 195 ページの「クラスタ制御ツール」
- 196 ページの「ネットワーク・ツール」

- 197 ページの「クラスタ/ノードのステータス・ツール」
- 198 ページの「パフォーマンス・モニタ・ツール」
- 198 ページの「カーネルのステータス・ツール」
- 199 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/cxtask**

    45 ページの「GUI の概要」および 93 ページの 第4章「GUI 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**

    48 ページの「cmgr(1M) の概要」および 119 ページの 第5章「cmgr 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**

  235 ページの「クラスタ設定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、クラスタ上限量が失われた場合に便利です。

  以下を参照してください。
  - 204 ページの「fs2d メンバーシップ上限量の安定性」
  - 231 ページの「CXFS サービスの再開」
  - 231 ページの「クラスタ設定データベースのクリア」
  - 235 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しないようにしてください。実行すると、ファイルシステムが再設定されてしまいます。140 ページの「cmgr を使った CXFS サービス・タスク」および 109 ページの「GUI を使った CXFS サービス・タスク」を参照してください。

メモ: 再設定については、このリリースでは使用できません。

- 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。

  - 157 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」
  - 157 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」
  - 117 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」
  - 118 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」

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/ノードのステータス・ツール

以下のクラスタ/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**

  76 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/cxdetail**

    109 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**

    186 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**

- 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 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 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 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 メンバーシップ内にないことを示します。
    - cms_stable(): CXFS 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

183 ページの「ログ・ファイル」を参照してください。

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・デーモンは実行されているか。76 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 メンバーシップ内にあるか。186 ページの「cluster_status を使ったクラスタのステータスの確認」、187 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 fs2d メンバーシップ内にあるか。各ノードの /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 のツリー表示を使用して、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。cluster_status(1M) コマンドまたは clconf_info コマンドを使用します。

    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。198 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（199 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。183 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。198 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 のツリー表示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。117 ページの「GUI を使った接続性のテスト」を参照してください。
- 各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 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:

  ```
  # /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
  ```

- 各ノードで 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。

- 204 ページの「適切な開始」
- 204 ページの「残っているクラスタの削除」
- 204 ページの「fs2d メンバーシップ上限量の安定性」
- 205 ページの「設定の統一性」
- 205 ページの「ノードの適切な重み付け」
- 205 ページの「GUI の使用」
- 206 ページの「ログ・ファイル名とサイズ」
- 206 ページの「Netscape と Brocade スイッチの GUI」
- 206 ページの「書込まれていない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」
- 207 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」
- 208 ページの「システム容量の効果的な使用」
- 209 ページの「強制シャットダウン後の CXFS の再開」

- 209 ページの「使用されていないノードの削除」
- 208 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」
- 209 ページの「リセット回線の取外し」
- 209 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75 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。

fs2d メンバーシップ上限量の安定性

設定プロセス中は、fs2d メンバーシップ上限量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
```

新しい上限量が出力されたら、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、新しい上限量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、上限量はなくなります。この場合、データベースは伝えられません。

fs2d メンバーシップ上限量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 fs2d メンバーシップ上限量がないノード、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。以下のコマンドを入力します。

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

fs2d メンバーシップ上限量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。60 ページの「システム・ファイルの設定」を参照してください。

ノードの適切な重み付け

以下のように、ノードに適切な重みを使用します。

- 0 または 1 の重みのみを使用する。
- 安定性のために、奇数の重み付けされたノードを使用する。
- 不安定なクライアントに重みを設定しない。

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ツリー表示で簡単に表示できます。ツリー表示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、定義したノードがツリー表示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、43 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。174 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。112 ページの「GUI を使ったログ・グループの設定」または 142 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。

書込まれていない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、書込まれていない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。

書込まれていないエクステントの追跡

ファイルシステムを定義するとき、書込まれていない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、書込まれていない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の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 つを除くすべてのノードでこれらの 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 行および fsr 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、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。

216 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。235 ページの「クラスタ設定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、fs2d メンバーシップおよび CXFS メンバーシップの上限量の問題を回避します。以下の節を参照してください。

- 108 ページの「GUI を使ったクラスタ定義の変更」
- 136 ページの「cmgr を使ったクラスタの変更」
- 130 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
```

172 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。

リセット回線の取外し

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、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。103 ページの「GUI を使ったノードの変更」または 126 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、特に重み付けされた 2 ノード・クラスタでは、リセット機能の使用が強く推奨されています。IRIS FailSafe ではリセット機能は必須です。6 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。
  - 76 ページの「ライセンスの確認」
  - 217 ページの「ライセンス・エラー」
  - 56 ページの「ソフトウェアのインストール」
- クラスタ・デーモンは実行されているか。76 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。
- `inetd` ラッパまたは `tcp` ラッパが干渉しているか。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。

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。174 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。

- 63 ページの「`/etc/config/cad.options`」
- 63 ページの「`/etc/config/fs2d.options`」

- 112 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。60 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。60 ページの「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。

60 ページの「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

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
```

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」
- 215 ページの「`clconfd` デーモンが異常終了する場合」
- 216 ページの「論理スワップ・スペース不足」
- 216 ページの「クラスタ名 ID 欠如エラー」
- 217 ページの「CXFS メンバーシップの取消し」
- 217 ページの「ライセンス・エラー」
- 217 ページの「IP アドレス・エラー」
- 218 ページの「`SYSLOG` エラー」
- 226 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。ノードが上限量に追加されたり上限量から削除されると、そのたびにメンバーシップが形成されます。この数字は、上限量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。ただし、CXFS クラスタ、特に重み付けされた 2 ノード・クラスタでは、リセット回線が強く推奨されています。IRIS FailSafe ではリセット回線は必須です。

```
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はないことを示します。

```
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/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、217 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。56 ページの「ソフトウェアのインストール」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
Availsmem 8207 availrmem 427 rlx freemem 10, real freemem 9
```

208 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:

```
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 としてラベルが付けられているディスクは、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 ラベルが付いたこれら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/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 ラベルの付いたディスクは XVM によって認識されていますが、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/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。118 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 157 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.
```

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。56 ページの「ソフトウェアのインストール」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 204 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:

```
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8-1 に、この SYSLOG メッセージの各部分を示します。

表8-1 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 (Node name)	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。

内容	部分	意味
<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。235 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、231 ページの「クラスタ設定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。235 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。235 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。95 ページの「GUI を使ったノードの定義」または 126 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.
```

クラスタ設定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。235 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。231 ページの「クラスタ設定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。235 ページの「クラスタ設定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、62 ページの「/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、62 ページの「/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、プール内のほかの複数のノードでまだ fs2d が実行されているときに特定のノードで cdbreinit を実行し、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場合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。

235 ページの「クラスタ設定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。235 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。231 ページの「クラスタ設定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。235 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。231 ページの「クラスタ設定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。235 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。231 ページの「クラスタ設定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。235 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。231 ページの「クラスタ設定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。235 ページの「クラスタ設定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、111 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:

```
cad_log:Thu Sep  2 17:25:06.092  cclconf_poll_clconfd: clconf_poll failed with error CI_IPCERR_NOPULSE
```

227 ページの表8-2 に、このメッセージの各部分を示します。

表8-2 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開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開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開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開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。235 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。235 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開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。235 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。

- 231 ページの「CXFS サービスの再開」
- 231 ページの「クラスタ設定データベースのクリア」
- 232 ページの「2 ノード・クラスタの回復」
- 232 ページの「再起動」
- 234 ページの「ノードをクラスタに再設定しない再起動」
- 235 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」
- 235 ページの「クラスタ設定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に設定されます。

詳細については、109 ページの「GUI を使った CXFS サービスの開始」または 140 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。

クラスタ設定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**

   208 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、204 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

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、109 ページの「GUI を使った CXFS サービスの開始」および 140 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。

- vnode が使用中であるために一部の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. 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 メンバーシップの形成後、既存のクラスタは、重み付けされた残りのノードの 50% まで低下する場合があります。詳細については、17 ページの「メンバーシップ上限量」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%** を超える重み付けされたノード
   - 重み付けされたノードの 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。
   - 111 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。
   - 141 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 メンバーシップを取消します。
   - 117 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
     ```

     157 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 メンバーシップを許可します。
   - 118 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
  ```

  157 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・ノード機能の設定を解除します。

注意: 重み付けされた 2 ノード・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。データを確実に保護するため、特に重み付けされた 2 ノード・クラスタでは、CXFS システムに対してハードウェア・リセットを使用することが推奨されています。IRIS FailSafe ではリセットは必須です。

GUI または cmgr を使用します。

- 111 ページの「GUI を使ったタイブレーカー・ノードの設定」
- 141 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 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。

231 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、24 ページの「デーモン」を参照してください。

クラスタ設定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. `fs2d` メンバーシップ上限量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. 以下のコマンドを入力します。

   ```
   # /usr/cluster/bin/cdbreinit
   ```

メモ: 208 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。

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193 ページの 第8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。1 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 Upgrades 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に適用されます)

[ ] 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、ハブではなくスイッチが推奨されます。

[ ] 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、特に重み付けされた 2 ノード・クラスタでは、リセット機能の使用を強くお薦めします。IRIS FailSafe ではリセット機能は必須です。15 ページの「ハードウェア・コンポーネント」および22 ページの「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。

[ ] ネットワークが使用可能であることを確認します。55 ページの 第2章「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。56 ページの「ソフトウェアのインストール」を参照してください。

[ ] 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。60 ページの「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。

  [ ] 60 ページの「システム・ファイルの設定」

  [ ] 67 ページの「(オプション) 自動再起動用の設定」

  [ ] 68 ページの「ネットワーク・インタフェースの設定」

  [ ] 70 ページの「シリアル・ポートの設定」

  [ ] 71 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。75 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、193 ページの 第8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。75 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) コマンドのクラスタ・マネージャ・ツールで使用されるコマンド・ライン・インタフェース・コマンド。

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 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。

CXFS 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 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 メンバーシップは、fs2d 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。

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 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

`fs2d` 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。`fs2d` デーモンは、プール内のすべてのノードへのクラスタ・データベース (CDB) の配布を管理します。ユーザ空間メンバーシップとも呼びます。

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 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。

クラスタ

クラスタとして定義されているプール内のノードのセット。クラスタは単純な名前で識別されます。この名前は、プール内で一意でなければなりません。たとえば、クラスタとノードで同じ名前を使用することはできません。

クラスタ内のすべてのノードはプールにも存在しますが、プール内のすべてのノードがクラスタに存在するとはかぎりません。つまり、クラスタは、プール内のノードのサブセットで構成されている場合があります。各プールのクラスタは 1 つだけです。

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、`fs2d` デーモンによって管理されます。

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

ノードを参照してください。

クラスタ・メンバーシップ

fs2d データベース・メンバーシップを参照してください。

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。

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。

システム・コントローラ・ポート

リモートで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 (CDB) 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。

上限量

クラスタの形成に必要なノードの数。CXFS メンバーシップの場合、初期メンバーシップ上限量の形成には、クラスタ内の重み付けされたノードの過半数 (>50%) が必要です。既存のメンバーシップ上限量を維持するには、クラスタ内の重み付けされたノードの半数 (50%) が必要です。`fs2d` データベース・メンバーシップの場合、メンバーシップ上限量の形成および維持には、プール内のノードの 50% が必要です。

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、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た状態。

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

fs2d 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、35 ページの「CXFS と IRIS FailSafe の同時実行」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。

ノード

オペレーティング・システム・イメージ。通常は、個々のコンピュータです。すべてのノードで、同バージョンまたは直前のバージョンの IRIX 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（6.5.13f と 6.5.14f など）。ノードは、ストレージ・システムとクラスタ内のノードを接続する SAN に接続されています。ノードが属することのできるクラスタは 1 つだけです。

このノードという用語は、Origin システムのノードとは異なる意味を持ちます。

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。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。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 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。

プール

ネットワークによって相互に結合され、クラスタ・データベースでノードとして定義されているノードのセット全体。通常、これらのノードは互いに近い位置にあり、同じ目的に使用されます。プール内の各ノードには、複製されたクラスタ・データベースが格納されています。

クラスタに追加できるノードはすべてプールの一部ですが、プール内のすべてのノードをクラスタの一部にする必要はありません。プールは 1 つだけです。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。

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。

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ノード。メタデータも参照してください。

メンバーシップ

fs2d データベース・メンバーシップおよび CXFS 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

CXFS メンバーシップ上限量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）。

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 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

fs2d データベース・メンバーシップ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

6.5.12f 以前からのアップグレード　73
6.5.12f 以前のファイルシステム定義の切替え
　73
6.5.12f 以前のファイルシステムの切替え
　73
6.5.12f から 6.5.13f でのファイルシステム形式の変更
　73

A

add nic　121
age　202
atime　31
AutoLoad 起動パラメータ　67

B

bulkstat　164

C

CAD オプション・ファイル　63
cad プロセス　25, 76
cdbreinit　208, 232, 235
cdbutil　203
chkconfig フラグ　180
clconf_info　167, 202
clconfd プロセス　25, 76
clconfd-scripts ディレクトリ　162
CLI（下層のコマンド・ライン・インタフェース）　42
cluster chkconfig フラグ　180
cluster_admin サブシステム　25
cluster_control サブシステム　25
cluster_mgr
　cmgrを参照
cluster_services サブシステム　25
cluster_status　167, 186
cmgr
　設定タスクを参照
　-c オプション　52
　概要　42, 48
　コマンド・ラインの実行　52
　シェルおよび　52
　終了　50
　スクリプトおよび　50
　スクリプト・モード　157
　テンプレート・ファイル　53
　ヘルプ　49
cmgr のプロンプト・モード　49
cmgr-templates ディレクトリ　53
cmond プロセス　25, 76
cms_dead()　199
cms_declare_membership()　199
cms_follower()　199
cms_leader()　199
cms_nascent()　199
cms_reset_timeout　181
cms_reset_timeout_action　181
cms_shutdown()　199
cms_stable()　199
cmsd　25
cmsd カーネル・スレッド　204
corpseleader プロセス　25
crontab の制約　164
crsd プロセス　25, 76
ctime　31
curses インタフェース　186
CXFS サービス開始と停止
　GUI　109
CXFS サービス停止と開始
　cmgr　140
CXFS サービスのアクティブ化

cmgr 140
CXFS サービスのアクティブ化と非アクティブ化
GUI 109
CXFS サービスの開始 171
cmgr 140
CXFS サービスの開始と停止
GUI 109
CXFS サービスの再開 231
CXFS サービスの停止
cmgr 140
CXFS サービスの非アクティブ化
cmgr 140
GUI 110
CXFS シャットダウン、強制 157
cmgr 172
GUI 117
CXFS シャットダウン、通常
cmgr 170
GUI 110
CXFS タイブレーカー・ノード
cmgr 141
GUI 111
CXFS で使用されない /etc/fstab 2
CXFS で使用されない fstab 2
CXFS マネージャ
設定タスク、cmgrを参照
概要 45
CXFS メンバーシップ 8
現在の情報 202
分割 22
CXFS メンバーシップ許可と取消し
cmgr 157
GUI 117
CXFS メンバーシップの許可
cmgr 157
GUI 118
CXFS メンバーシップの取消し
cmgr 157
GUI 117
CXFS メンバーシップの問題 204
cxfs_cluster chkconfig フラグ 180
cxfs*mount.sh スクリプト 162
cxfs*umount.sh スクリプト 162
cxfsd プロセス 26
cxfsfilesystemUpgrade
74

D

Data Migration Facility 34, 165
dcshake プロセス 25
DMAPI 165
DMF 34, 165
DNS 61, 68

E

/etc/config/cad.options ファイル 63
/etc/config/fs2d.options ファイル 63
/etc/config/netif.options 61
/etc/hosts 60
/etc/init.d 235
/etc/init.d/cxfs stop
110
/etc/mtab 2
/etc/nsswitch.conf 61
/etc/services ファイル 62
/etc/sys_id 60

F

FailSafe クラスタの切替え
cmgr 137
GUI 94, 108
FailSafe タスクセットからの切替え 94
FailSafe 同時実行 35
FailSafe ノードの切替え
cmgr 129
GUI 94, 105
FailSafe メンバーシップ 9
filesystem reorganizer 2
find コマンドおよび crontab 164
FLEXlm ライセンス・キー 33
fs2d
データベース・メンバーシップ 8

プロセス　25
fs2d オプション・ファイル　63
fs2d プロセス　76
fs2d_log　204
fsr　164
fsr_xfs(1M)　2
fx コマンド　81

G

GUI
　概要　42
　タスク
　　設定タスクを参照

H

hosts ファイル　60
hsm　165

I

I/O レート保証　6
icrash　204
incarnation　202
IP アドレスおよびコントロール・ネットワーク　96
IRIS FailSafe 同時実行　35
IRIS FailSafe との同時実行　35
IRISconsole　34
is_* コマンド　121

J

java ファイルとリリース・レベル　58
java_plugin　59

L

L2　122
LAN　4
Legato NetWorker　175

M

mesgtcpaccept プロセス　26
mesgtcpdiscovery　26
mesgtcpmulticast　26
mesgtcprcv プロセス　26
mkfs コマンド　81
mkpart　98, 122
MMSC　122
MSC　122
mtime　31

N

NetBackup　175
netif.options　61
network information service　61
NetWorker　175
NFS　176
NFS エクスポート　162
NIS　61, 68
nsadmin　61
nsd　61
nsswitch.conf　61
NT クライアント　176
NVRAM 変数　67

O

Origin 200　33
Origin 2000　33
OS レベル　33

P

pcp_eoe　191
pcp_eoe.sw.xvm　191
Performance Co-Pilot (PCP)
　XVM 統計　59, 191
physvol　216
pmgxvm　191

Q

quota　2, 176

R

remove nic　121

reorganizer 2
/.rhosts ファイル 66
rotatelogs 174
routed 69

S

SAMBA 176
SAN 1
SAN ハードウェア 34
set コマンド 121
sthreads 25
sys_id 60
sysctrl* コマンド 121
SYSLOG 178
systune 181

T

Tape Management Facility (TMF) 34
TMP 34
TRIX 34
Trusted IRIX 34

U

Unified Name Service 61
UNS 61

V

/var/cluster/clconfd-scripts ディレクトリ 162
/var/cluster/cmgr-scripts/rotatelogs 174
/var/xfsdump/inventory 179
VERITAS NetBackup 175

X

X/Open Data Storage Management Specification 165
xfs quota 176
xfs_fsr (fsr_xfs)(1) 2
xfsdump および xfsrestore 178
XSDM 165
xthreads 26
XVM 34, 175
xvm コマンド 81
XVM 統計 191

あ

アイコンとカラー 184
「アクティブ」クラスタのステータス 186
アンマウント 177
アンマウント・スクリプト 162

い

インストール 55
インストールされるサブシステム 57
インベントリ・ファイル 179

え

エクステントの追跡 206
「エラー」クラスタのステータス 186

お

重み 96
重み付け 17
重み付けされた 2 ノード・クラスタ 42

か

階層ストレージ管理 165
回復
  重み付けされた 2 ノード・クラスタ 42
  メタデータ・サーバの 168
  用語 14
回復機能 175
回復プロセス 25
書込まれていないエクステントの追跡 206
書込みとメタデータの流れ 31
カラーとアイコン 184
管理 161
管理シャットダウン 170
管理者へのクラスタ変更の通知
  cmgr 134
  GUI 108
カーネル空間メンバーシップ 8

カーネル調整可能パラメータ　181

き

強制 CXFS シャットダウン
　cmgr　172
強制アンマウント　115
強制シャットダウン　172
強制シャットダウンおよび再開　173, 209

く

クラスタ ID　108
クラスタからのノードの削除
　cmgr　136
　GUI　101
クラスタからのノードの除去
　cmgr　136
　GUI　101
クラスタ切替え
　cmgr　137
　GUI　108
クラスタ削除
　cmgr　138
　GUI　109
クラスタ定義
　cmgr　134
　GUI　106
クラスタの FailSafe 切替え
　cmgr　137
　GUI　94, 108
クラスタのサイズ　33
クラスタの再設定　204
クラスタの削除
　cmgr　138
　GUI　109
クラスタの作成
　cmgr　134
　GUI　106
クラスタのステータス　186
「不明」クラスタのステータス　186
クラスタの定義
　cmgr　134, 138
　GUI　106
クラスタの表示
　cmgr　139
　GUI　109
クラスタのモード
　108
クラスタ表示
　cmgr　139
　GUI　109
クラスタへの再設定
　強制 CXFS シャットダウン　173
　通常 CXFS シャットダウン　171
クラスタへのノードの追加
　GUI　101
クラスタ変更
　cmgr　136
　GUI　108
クラスタ（用語）　8
クラスタ・タスク
　cmgr　134
クラスタ・データベース
　シャットダウン　169
　上限量　204
　バックアップと復元　180
クラスタ・デーモンの再開　235
クラスタ・ドメイン　175
クラスタ・マネージャのツール　42
クラスタ・メンバーシップ　8
クラスタ・モード　108

こ

コマンド・ライン・インタフェース　42, 48
　cmgrも参照
コントロール・ネットワーク　9, 96

さ

再起動　71, 232
再起動（回避）　173
再設定　13
　重み付けされた 2 ノード・クラスタ　42
サポートされていないノードの数　33

し

システム
  ステータス　183
  ソフトウェアの通信パス　27
  表示　24
  ファイル　60
システム調整可能パラメータ　181
システム・コントローラのステータス　190
システム・コントローラ・タイプ　122
「実行中」ノードの状態　188
シャットダウン　170
  強制終了された CXFS クラスタ・データベース　172
  クラスタ・データベース　169
  通常 CXFS
    GUI　110
  通常の CXFS データベース　170
シャットダウンおよび再開　173, 209
上限量　17, 204
使用できる直前のバージョンの OS　33
初期クラスタ設定
  cmgr　82
  GUI　78
シリアル回線　10
シリアル接続
  cmgr　156
  GUI　117
シリアル・ポート設定　70
診断タスク
  cmgr　156
  GUI　117

す

スクリプト・モードと cmgr　157
ステータス
  クラスタ　186
  システム、概要　183
  システム・コントローラ　190
  ノード　188
ストライプ　175
ストレージ・エリア・ネットワーク (SAN: Storage Area Network)　1
スレッド　24

せ

接続性テスト
  cmgr　156
  GUI　117
接続性のテスト
  cmgr　156
  GUI　117
設定タスク
  cmgr　119
    CXFS サービス　140
    cxfs タイブレーカー　141
    FailSafe クラスタの切替え　137
    管理者へのクラスタ変更の通知　134
    クラスタ削除　138
    クラスタ定義　134
    クラスタ表示　139
    クラスタ・タスク　134
    接続性テスト　156
    デフォルト　120
    ノード削除　130
    ノード定義　121
    ノード表示　132
    ノード変更　126
    ノード・タスク　120
    ノード・リセット　129
    ファイルシステム削除　155
    ファイルシステムのマウントとアンマウント　150
    ファイルシステム変更　151
    メタデータ・サーバ再設定　155
    メタデータ・サーバ定義　144
    メンバーシップ許可と取消し　157
    ログ設定　142
  GUI
    CXFS サービス開始と停止　109
    cxfs タイブレーカー　111
    FailSafe クラスタの切替え　108
    FailSafe ノードの切替え　105
    クラスタ削除　109
    クラスタ定義　106

クラスタ表示　109
クラスタ変更　108
新規クラスタの設定　80
新規ファイルシステムの設定　81
接続性テスト　117
ノード削除　106
ノード追加と削除　102
ノード定義　95
ノードの表示　106
ノード変更　103
ノード・リセット　102
ファイルシステム削除　116
ファイルシステムのマウントとアンマウント　114
ファイルシステム変更　115
メタデータ・サーバ再設定　116
メタデータ・サーバ定義　113
メンバーシップ許可と取消し　117
ログ設定　111
セル ID　202

た

タイブレーカー・ノード
cmgr　141
GUI　111
タスク　46
ダンプ　178
断片化解消プログラム　2

ち

チャネル・トラフィック　214

つ

通常 CXFS シャットダウン　110
通信パス　27
ツール　42

て

「停止」ノードの状態　188
ディスク管理　175
ディスクのストライプ　175
ディスク・バッファ　5
テンプレート・ファイル　53
データ管理 API　165
データベース
クリア　231
シャットダウン　169
ダンプ　203
メンバーシップ　8
データベースのクリア　231
データベースのダンプ　203
データベースのリセット　231
データ・フロー　31
デーモン　24, 76
デーモンの再開　235

と

同時実行　33
ドメイン　175
ドメイン・ネーム・サービス　61
トラブルシューティング　193
トークン　31

な

名前付きパイプ　6

ね

ネットワーク　9
ネットワークの接続性
cmgr　156
GUI　117
ネットワークの分割とハードウェアのリセット　23
ネットワーク分割　23
ネットワーク・インタフェースの設定　68
ネットワーク・セグメント　9
ネットワーク・ファイルシステムの比較　4
ネットワーク・ルーティング　69
ネーム・サービス・デーモン　61

の

ノード

重み付け 17
状態 188
ステータス 188
タスク
設定タスクを参照
定義 7
ノード切替え
cmgr 129
GUI 105
ノード削除
cmgr 130
GUI
106
ノード追加と削除
GUI 101
ノード定義
cmgr 121
GUI 95
ノードの FailSafe 切替え
cmgr 129
GUI 105
ノードの重み 96
ノードの作成
GUI 95
ノードの自動再起動 67
ノードのステータス
強制 CXFS シャットダウン 172
ステータス
データベースのシャットダウン 169
通常 CXFS シャットダウン 171
ノードの表示
cmgr 132
GUI 106
ノードの変更
cmgr 126
GUI 103
ノードのリセット
cmgr 129
GUI 102
ノード表示
cmgr 132
GUI 106
ノード変更
cmgr 126
GUI 103
ノードを自動的に再起動する 67
ノード・タスク
cmgr 120
GUI 95
ノード・リセット
cmgr 129
GUI 102

は

パイプ（名前付き） 6
バックアップ 175
パラメータ（システム調整可能） 181
パーティション 121–122
ハートビート・ネットワーク 9, 96

ひ

「非アクティブ」
クラスタのステータス 186
ノードの状態 188
ピアツーディスク・モデル 5
必要なプライベート・ネットワーク 33

ふ

ファイルシステム拡張 178
ファイルシステム削除
cmgr 155
GUI 116
ファイルシステムと IRIS FailSafe 37
ファイルシステムのアンマウント
cmgr 151
GUI 115
ファイルシステムの拡張 178
ファイルシステムの削除
cmgr 155
GUI 116
ファイルシステムの除去
GUI 116

ファイルシステムの変更
cmgr 126, 151
GUI 115
ファイルシステムのマウント
cmgr 150
GUI 114
ファイルシステムのメインテナンス 176
ファイルシステム変更
cmgr 151
GUI 115
ファイルシステム・アンマウント
cmgr 151
ファイルシステム・タスク
cmgr 144
GUI 113
ファイルシステム・タスクセット 81
ファイルシステム・マウント
cmgr 136, 150
ファイルシステム・マウントとアンマウント 177
GUI 114–115
ファイルシステム・メタデータ・サーバ定義
cmgr 144
GUI 113
ファイルシステム・メタデータ・サーバの作成
cmgr 144
GUI 113
ファイルへのスワップ 6
復元 178
「不明」ノードの状態 188
プライベート・ネットワーク 9
プラットフォーム 33
プール 7
プールからのノードの削除
cmgr 130
GUI 106
プールからのノードの除去
cmgr 130
GUI 106

へ

ヘルプ
cmgr 49
GUI の 80

ほ

ホスト名コントロール・ネットワーク 96
ボリューム管理 175

ま

マウント 177
マウントポイントのネスト 6
マウント・オプション
113
マウント・オプションのネスト 6
マウント・スクリプト 162
マウント・ポイント
113

み

ミラー化 175

め

命名の制約 60
メインテナンス 161
メタデータの例 4
メタデータ・クライアント 23
メタデータ・サーバ
回復 168
検出 165, 167
定義されている用語 10
メタデータ・サーバからのファイルシステムの削除 170
メタデータ・サーバ再設定
cmgr 155
GUI 116
メタデータ・サーバ定義
cmgr 144
GUI 113
メタデータ・サーバの移動 170
メタデータ・サーバの再設定
cmgr 155
GUI 116

メンバーシップ
　クラスタ・メンバーシップまたは CXFS メンバーシップを参照
メンバーシップ重み　96

ゆ
ユーザ空間メンバーシップ　8

よ
読取りとメタデータの流れ　31

ら
ライセンス・エラー　217

り
リセットのためのレース　23
リセット・ハードウェア　22
リセット・レース　23
リモート・リセット接続　23

る
ルーティング　69

れ
連結　175

ろ
ログ設定
　cmgr　142
　GUI　111
ログ・ファイル
　管理　174
　モニタ　183
　リスト　183
ログ・ファイルのローテーション　174
ローカル・エリア・ネットワーク (LAN: Local Area Network)　4
ローカル・ドメイン　175
ローカル・ノード　171

訂正のお知らせ

IRIX CXFS 製品ドキュメント『CXFS Software Installation and Administration Guide (日本語版)』(007-4016-011JP, IRIX 6.5.14 対応) について、「図 6-1 メタデータ・サーバが表示されているウィンドウ (本文 p.170)」に誤りがありました。ドキュメント中の図には、メタデータ・サーバのパラメータ値が正しく表示されていません。正しい図を下に示します。

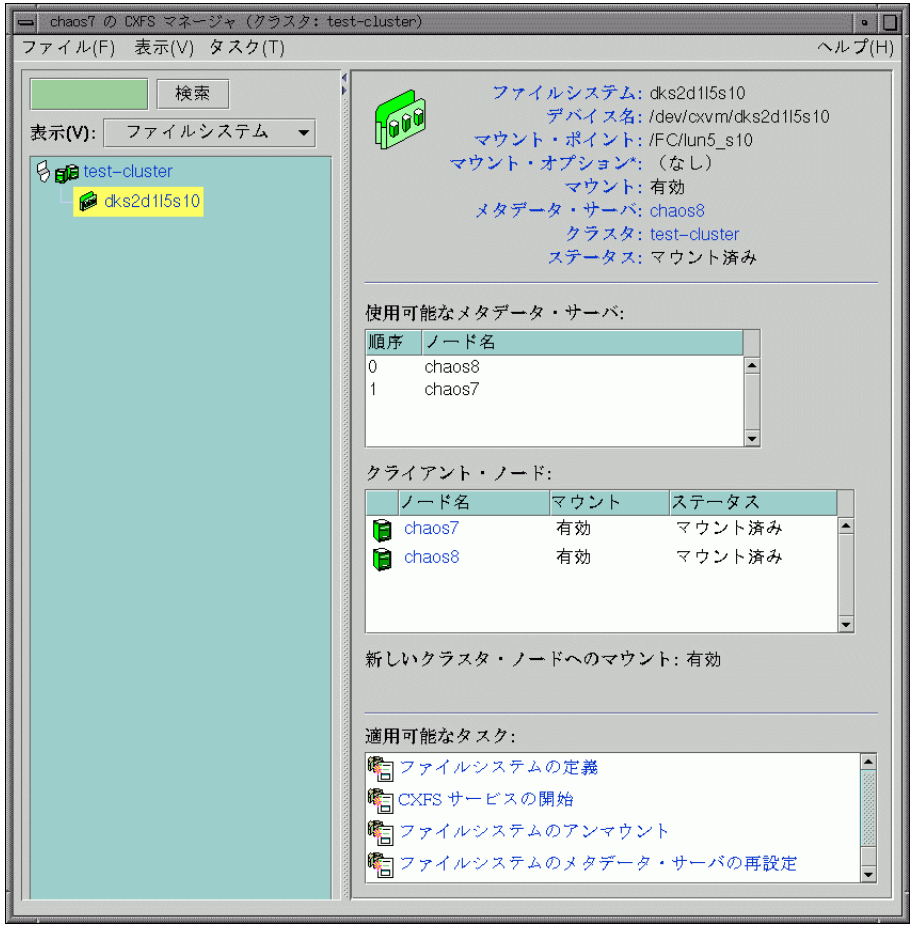

図 6-1 メタデータ・サーバが表示されているウィンドウ

以上、お詫びするとともに訂正をお知らせいたします。