

Gfarm Workshop 2015
2015年9月4日神戸

Gfarmファイルシステムの 概要と実装

建部修見
筑波大学

Gfarmファイルシステム

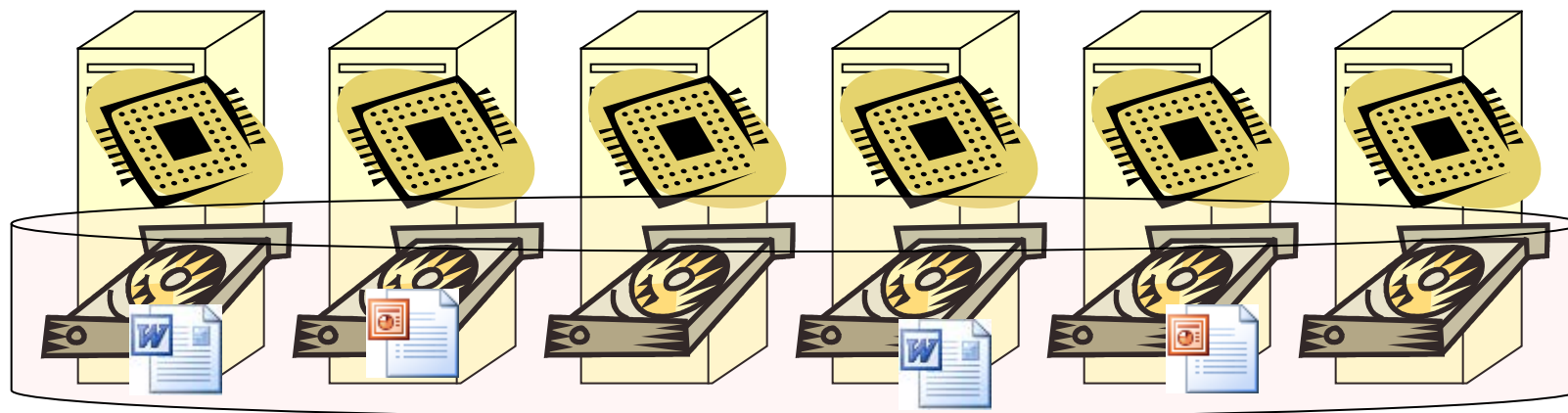


- 2000年より研究開発
 - 国際会議SC03、SC05、SC06で受賞
- オープンソース広域分散ファイルシステム
 - <http://oss-tsukuba.org/software/gfarm/>
 - 16,461 downloads since March, 2007
- NPO法人つくばOSS技術支援センターによるサポート
- 特徴
 - 広域で性能がスケールアウト
 - データアクセス局所性、ファイル複製
 - 単一障害点なし
 - 複製数維持機能
 - ホットスタンバイMDSサーバ
 - 無停止で拡張、更新可能
 - データ完全性を保証しサイレントデータ損傷も対応可

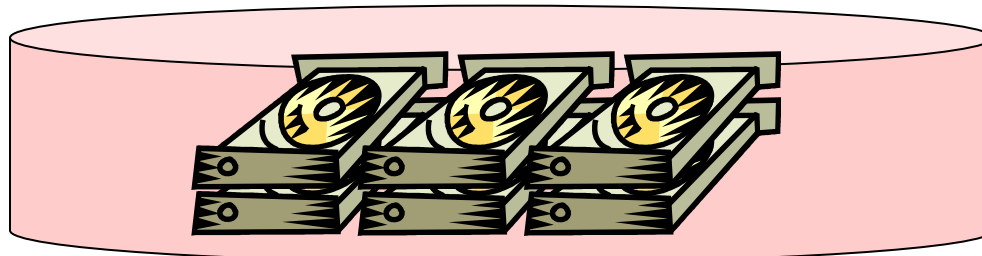


Gfarmファイルシステムの構成

- 一般的なPCのローカルディスクを束ねる
- ユーザには、共有ファイルシステムとして見える
- 複数のディスクに分散してデータを保持

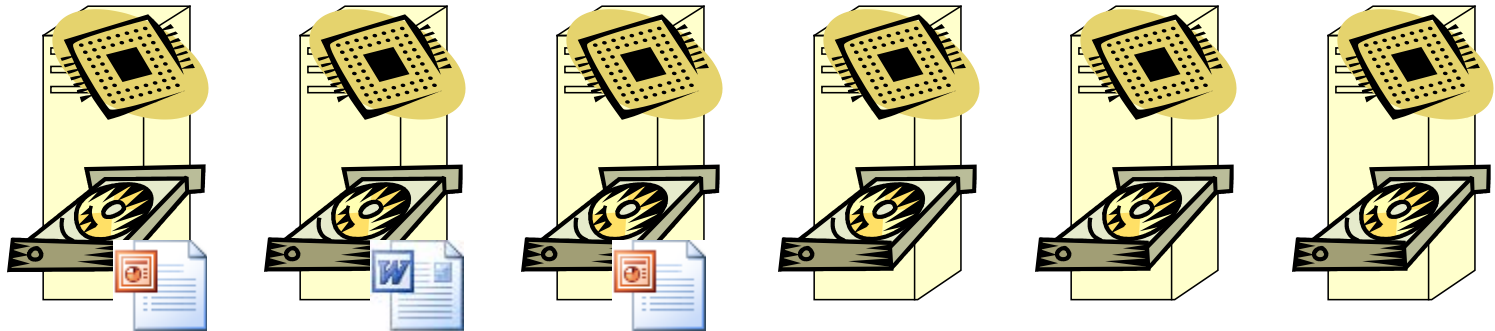
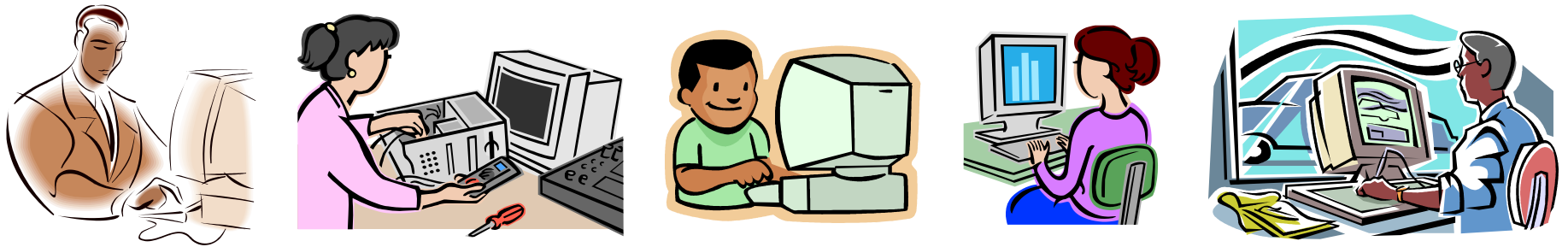


Gfarmファイルシステム



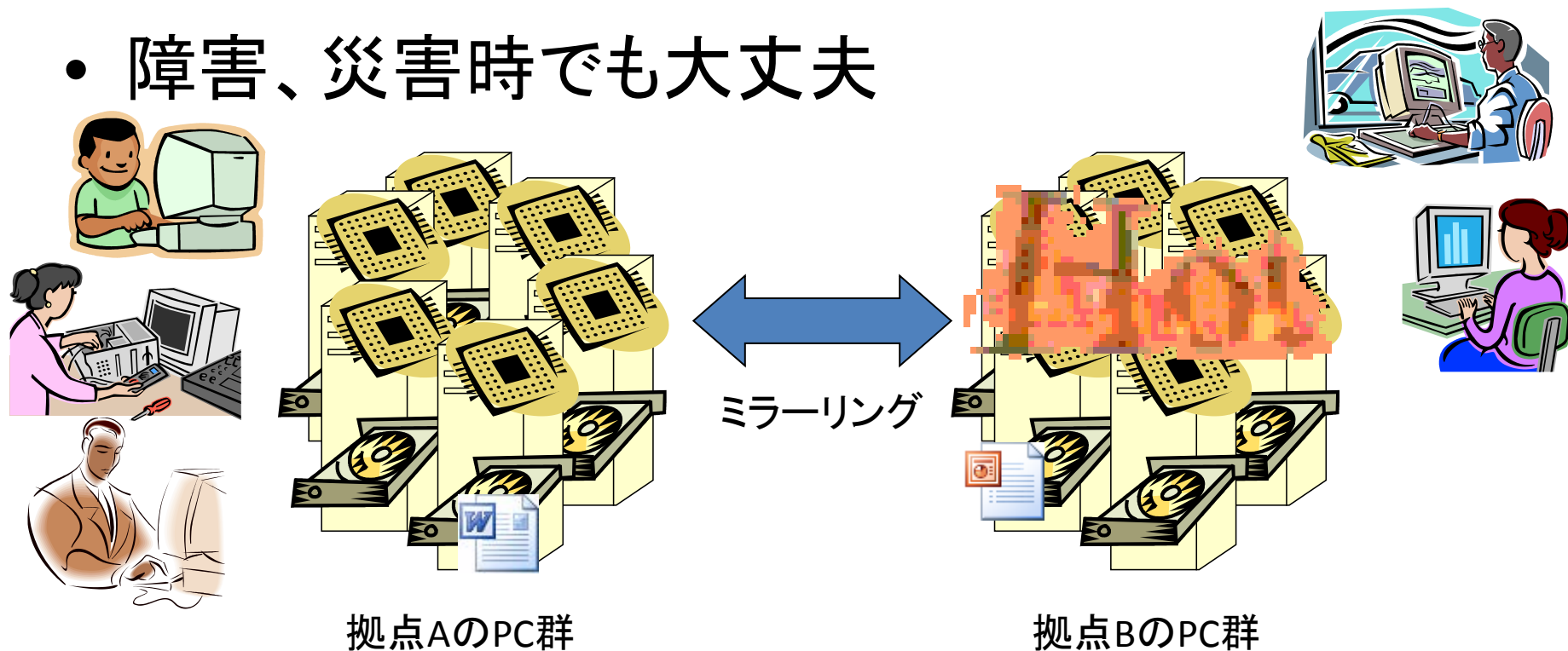
利用例(1): 組織内の共有ファイルシステム

- ファイルシステムの容量を運用中に増加
- ファイル複製の数を運用中に増やして、ホットスポットの回避と、信頼性の向上



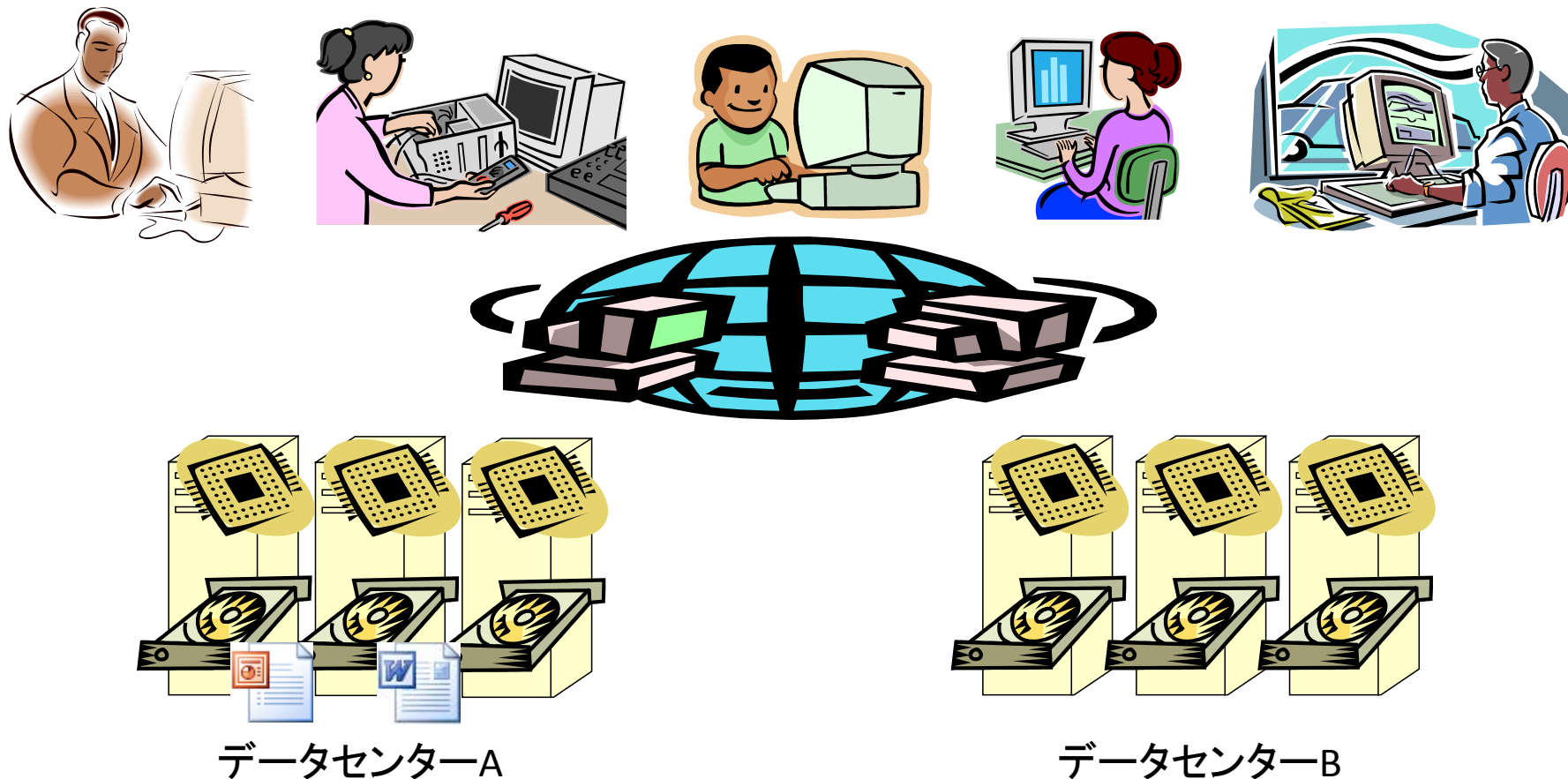
利用例(2): 拠点間でのデータ共有

- ミラーリングを行い、それぞれの拠点に保持されたデータをアクセス
 - データが近くにあるため高速なアクセス
- 障害、災害時でも大丈夫



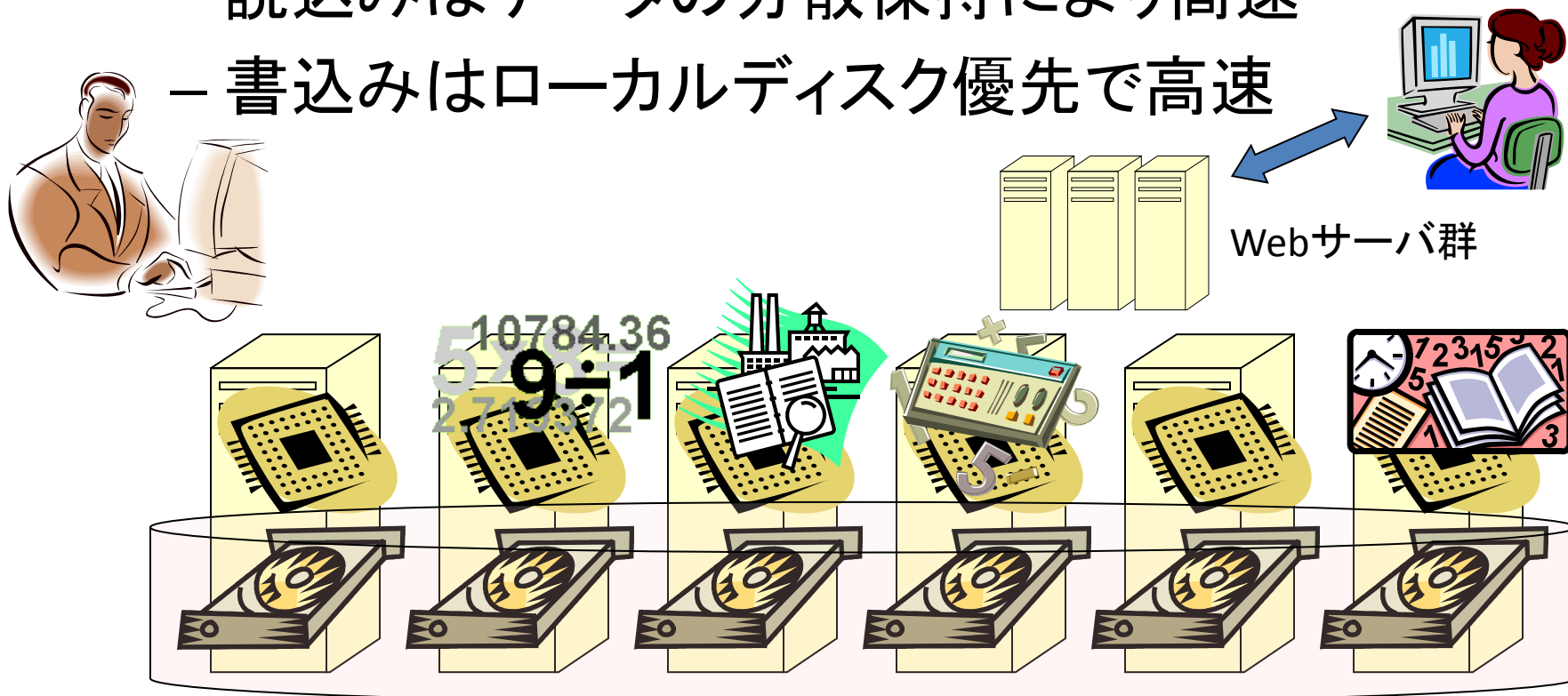
利用例(3): 遠隔のファイル格納サービス

- ファイルの複製を地理的に離れた場所に保持することにより、高信頼なサービスを実現



利用例(4): 大規模データ処理

- 高性能共有ファイルシステムとして、複数のPCで分散並列処理
 - 読込みはデータの分散保持により高速
 - 書込みはローカルディスク優先で高速



最新機能・状況紹介

ファイルシステムノードグループ

[Gfarm 2.6.0]

- ファイルシステムノードをグループ化
- グループに対して複製数を指定
 - 西拠点、東拠点に複製を一つずつ

% **gfhostgroup** *hostA groupA* // *hostA*を*groupA*に

% **gfncopy -S** *groupA:1,groupB:2 /home*

// *groupA*と*groupB*に複製を一つずつ作成

データ完全性 [Gfarm2.5.8.5]

- ファイル破損を検知
- digestを書込時に計算しメタデータに保存
- クライアントでもdigestを計算しend-to-endの完全性 [Gfarm 2.6.0]
- 読込時にユーザプログラムにchecksum errorを返す
 - 壊れたファイルは読み出せない
 - /lost+foundへ移動し自動修復 [Gfarm 2.6.1]
- 複製作成時にchecksum errorを返す [Gfarm 2.6.0]
 - /lost+foundへ移動し自動修復 [Gfarm 2.6.6]

% **gfcksum** *file*

- ファイルのdigestを表示

% **gfcksum** -c [-h *host*] *file*

- (hostに格納されている)ファイルのdigestを計算し確認

Gfarmファイルシステム運用監視

- Zabbixプラグインにより、各サーバを監視

The screenshot displays the Zabbix web interface in a Firefox browser window. The page title is "ZabbixServer01: ダッシュボード". The main navigation bar includes "ヘルプ", "サポート", "プリント", "プロフィール", and "ログアウト". The left sidebar shows navigation options like "監視データ", "インベントリ", "レポート", "設定", and "管理". The main content area is titled "Zabbixサーバの状態" and contains a table of server parameters.

パラメータ	値	詳細
Zabbixサーバの起動	はい	esci-wgfarmc2.aics.riken.jp:10051
ホスト数 (有効/無効/テンプレート)	54	5 / 2 / 47
アイテム数 (有効/無効/取得不可)	146	129 / 17 / 0
トリガー数 (有効/無効)[障害/不明/正常]	59	54 / 5 [0 / 0 / 54]
ユーザ数 (オンライン)	2	1
1秒あたりの監視項目数 (Zabbixサーバの要求パフォーマンス)	0.33	-

更新: 15:33:29

Below this, the "システムステータス" section shows a table of system health metrics:

ホストグループ	致命的な障害	重度の障害	軽度の障害	警告	情報	未分類
Gfarm v2 FileSystem	0	0	0	0	0	0

更新: 15:33:29

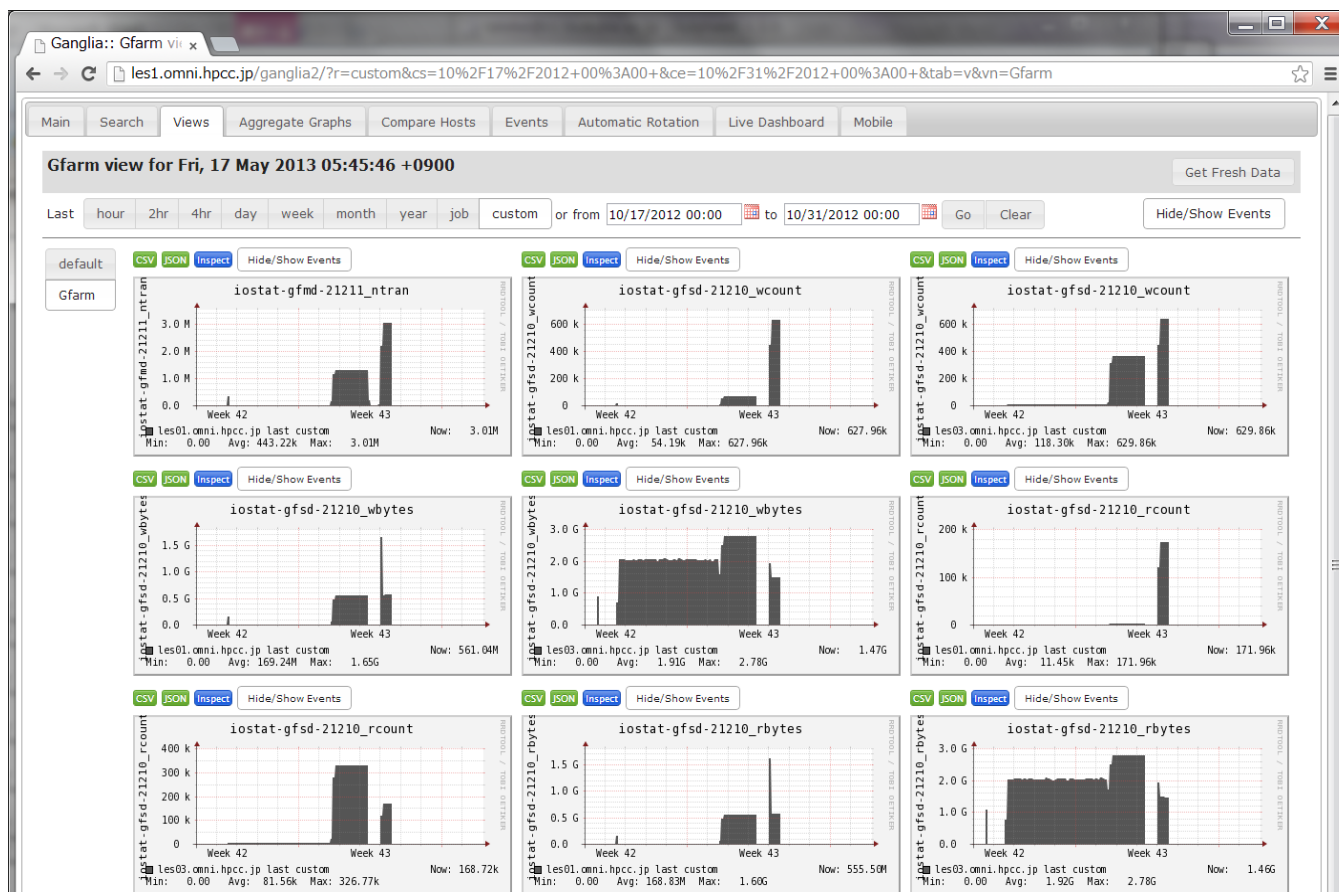
The "ホストステータス" section shows a table of host status:

ホストグループ	障害なし	障害あり	合計
Gfarm v2 FileSystem	5	0	5

更新: 15:33:29

性能モニタリング [Gfarm 2.5.8]

- GangliaプラグインによるIOPS、バンド幅のリアルタイム性能モニタリング



Automount [Gfarm2fs 1.2.9.2]

- ディレクトリをアクセスするとGfarmファイルシステムをautomount

```
% ls -a /usr/users
```

```
...
```

```
% ls -a /usr/users/tatebe
```

```
... .bash_logout .bash_profile .bashrc .emacs
```

- Gfarmのディレクトリをホームディレクトリとしてautomount可能！

Gfarm-Samba VFS plugin version 1.0.0

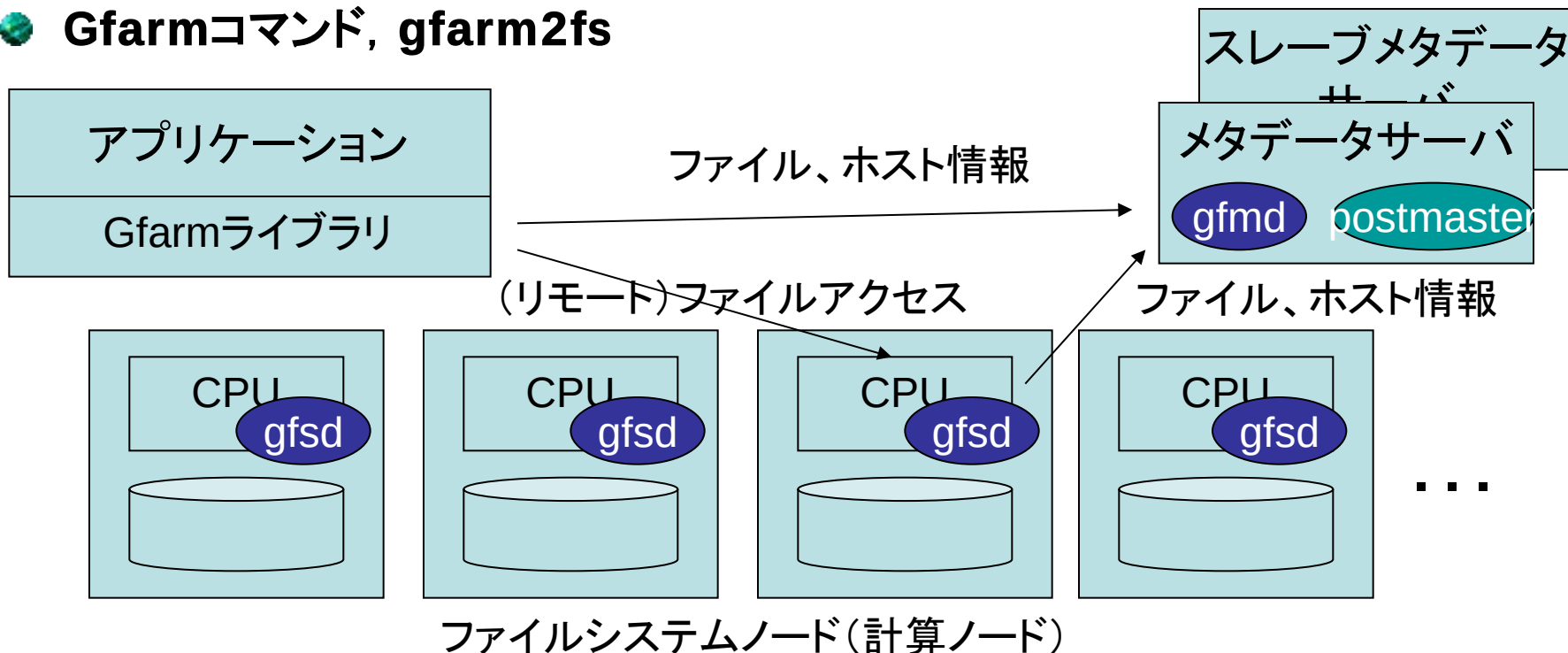
- Gfarm2fsを利用しなくてもSambaからGfarmを利用するためのモジュール
- Samba 3.6系に対応

```
[gfarm]
comment = GfarmFS
path = /
vfs objects = gfarm
writeable = yes
browseable = yes
```

Gfarmの実装

Gfarmの実装の概要(1)

- **gfmd** – メタデータサーバ(MDS)
 - ▶ ディレクトリ情報、複製カタログ、ホスト情報、プロセス情報
- **gfsd** – I/Oサーバ
 - ▶ (リモート)ファイルアクセス
- **libgfarm** – Gfarmライブラリ
 - ▶ Gfarm API
- **Gfarmコマンド, gfarm2fs**



Gfarmの実装の概要(2)

● gfmd – メタデータサーバ(MDS)

▶ メタデータをメモリにキャッシュ

- ⌚ 更新はジャーナル書込完了を待って応答を返す
- ⌚ バックエンドDBへの書込み、スレーブgfmdへの転送は別スレッドで

▶ プロセスのオープン状態を管理(CTO一貫性)

- ⌚ ファイルディスクリプタベースのアクセス
- ⌚ 複数プロセスでのファイルオープンに対する一貫性制御

▶ ファイルシステムノードの状態のモニタと操作

- ⌚ CPU負荷, ファイルシステム容量, . . .
- ⌚ 実ファイルの消去、ファイル複製作成、複製数維持

▶ コンパウンドRPCの導入によるRPC数削減

▶ ホットスタンバイgfmd

- ⌚ 同期スレーブと非同期スレーブ

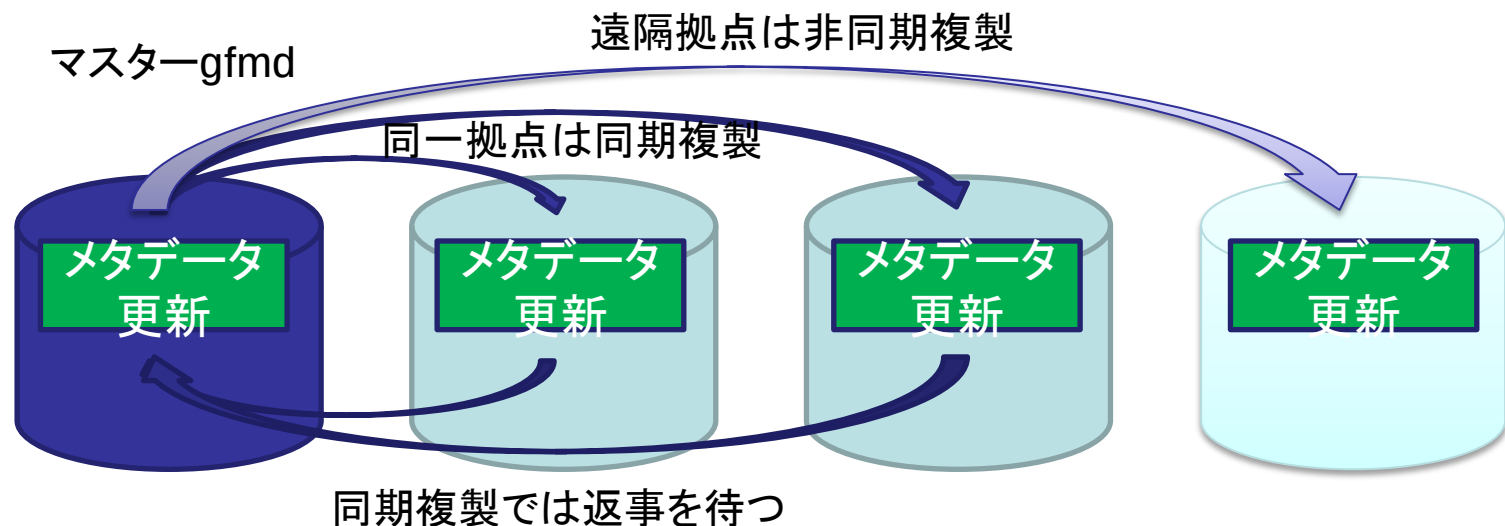
Gfarmの実装の概要(3)

● gfsd – I/Oサーバ

- ▶ ファイルディスクリプタベースのファイルアクセス
 - ◎ パス名やiノード番号ベースではない
 - ◎ iノード番号はオープン時にgfmdから取得し、アクセス権限、モードなどチェック
- ▶ メタデータをファイルクローズ時に更新
- ▶ 不必要にメタデータを保持しない
 - ◎ ファイルのrename, chmodなどで、gfsdへのアクセスを不要に
- ▶ チェックサム計算
- ▶ I/Oサーバ間転送によるファイル複製作成
- ▶ I/Oエラー、ストレージ障害、ディスクフルの対応

ホットスタンバイgfmd

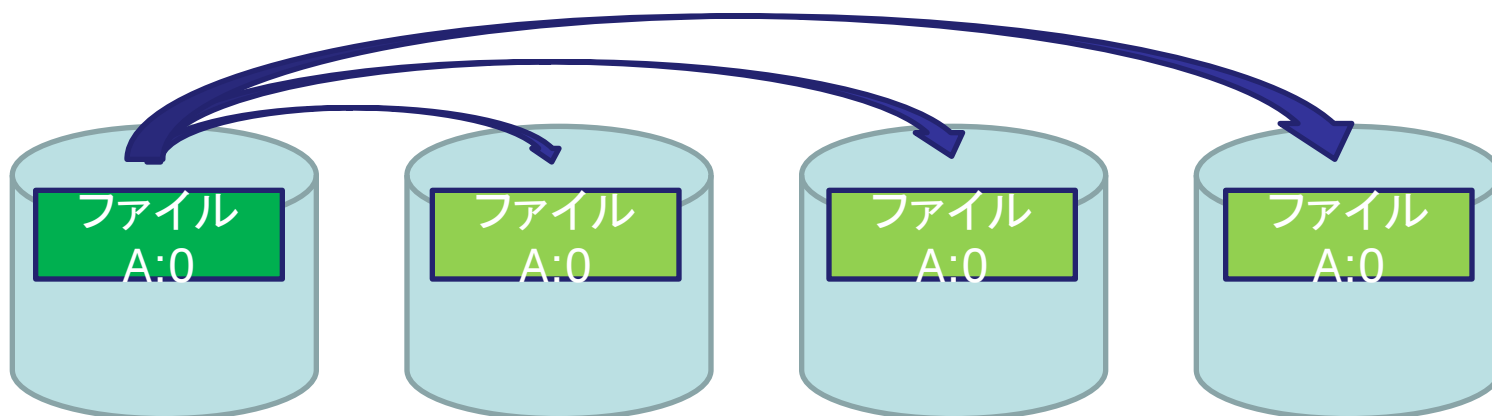
- マスターgfmdにおいてメタデータ更新
 - ▶ ジャーナルファイルに保持
 - ▶ スレーブgfmdに転送
- 同期スレーブの場合は、スレーブからの返事を待つ
- 非同期スレーブはディザスタリカバリのため



ファイル複製の自動作成(1)

- ファイルクローズ時に指定された数の複製を作成
 - ▶ 拡張属性(gfncopy)でディレクトリ単位に指定

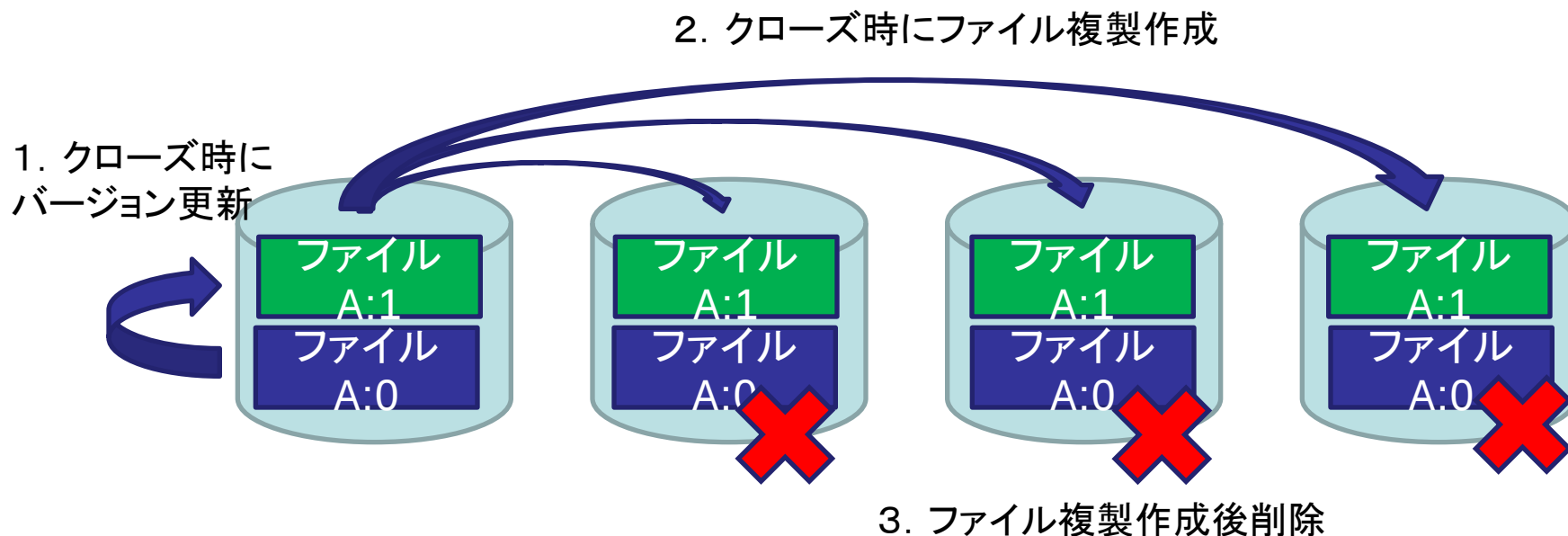
1. クローズ時にファイル複製作成



ファイル複製の自動作成(2)

● ファイル複製数を減らさない仕組み

- ▶ 無効化型一貫性制御では更新時複製数が1となる
- ▶ 更新型一貫性制御を安全に行うため、クローズ時にバージョンニングと複製作成



- 複製数変更の可能性があるとき、gfmdの専用スレッドで全ファイルの複製数のチェック
 - ▶ 足りなければ作成、多すぎれば削除
- チェックのタイミング
 - ▶ gfmd、gfsd起動時
 - ▶ gfsdが停止して指定時間後
 - ▶ 複製作成数変更時
 - ▶ ファイル、ディレクトリその他ディレクトリへの移動時
 - ▶ 複製作成失敗時
- チェックのon/offを動的に指定可能

クライアントライブラリの初期化処理

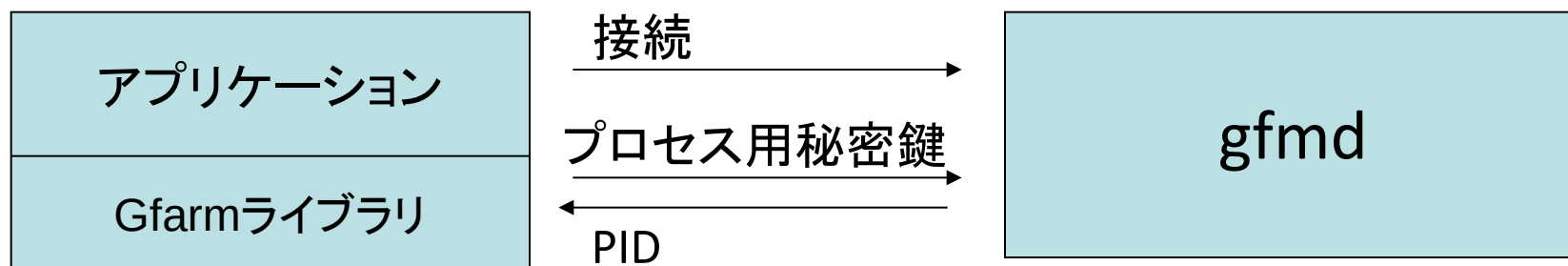
● 設定ファイルの読み込み

▶ gfarm2.conf

● メタデータサーバへ接続

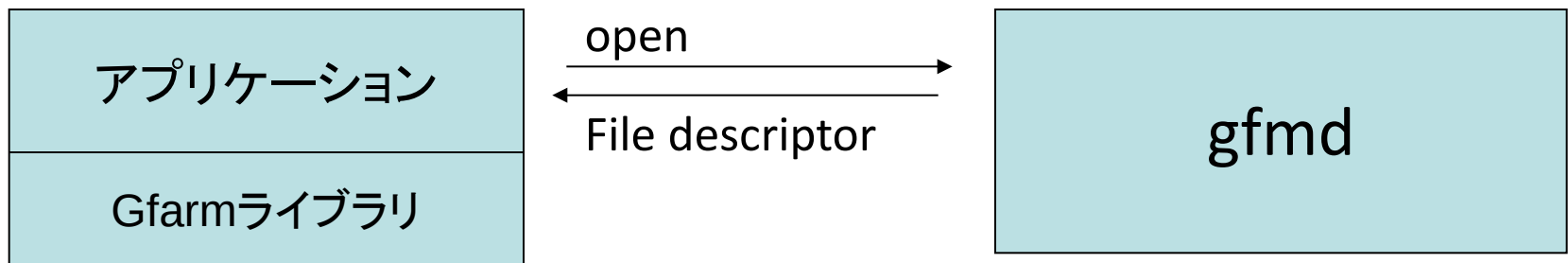
● 自プロセス情報の登録

▶ プロセス用秘密鍵を送付し, プロセスIDの割当



ファイルオープン

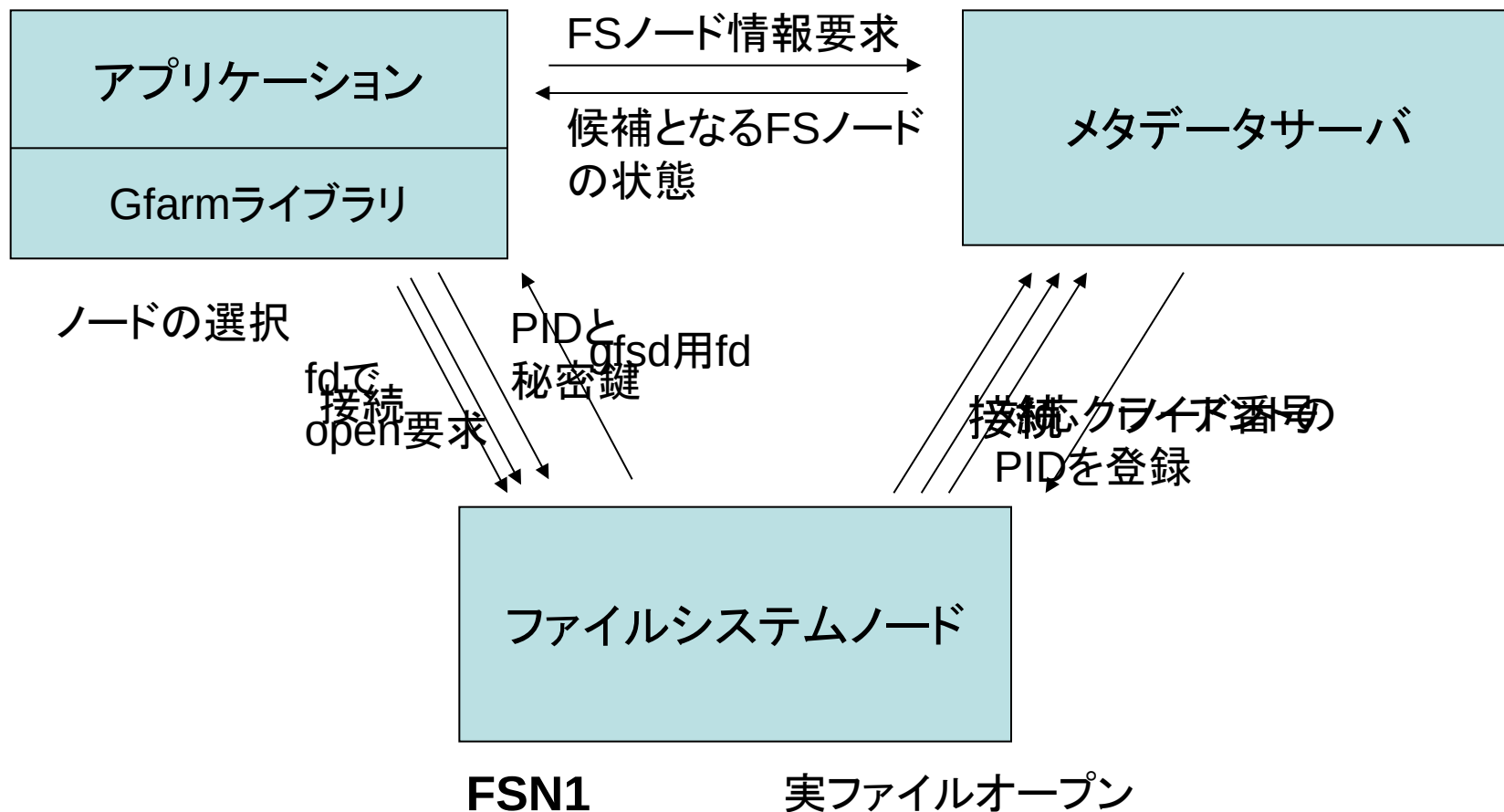
- **gfmdにファイルオープン要求**
 - ▶ ファイルディスクリプタ割当
- **GFS_File構造体の割当**
 - ▶ ファイルバッファの割当 (256KB)
- **(gfsdへのアクセスは, 初めてファイルアクセスがあったとき)**



GFS_File構造体の割当

ファイルのアクセス時のgfsdとの接続(1)

● gfsdと接続していない場合、接続して実ファイルをオープンする

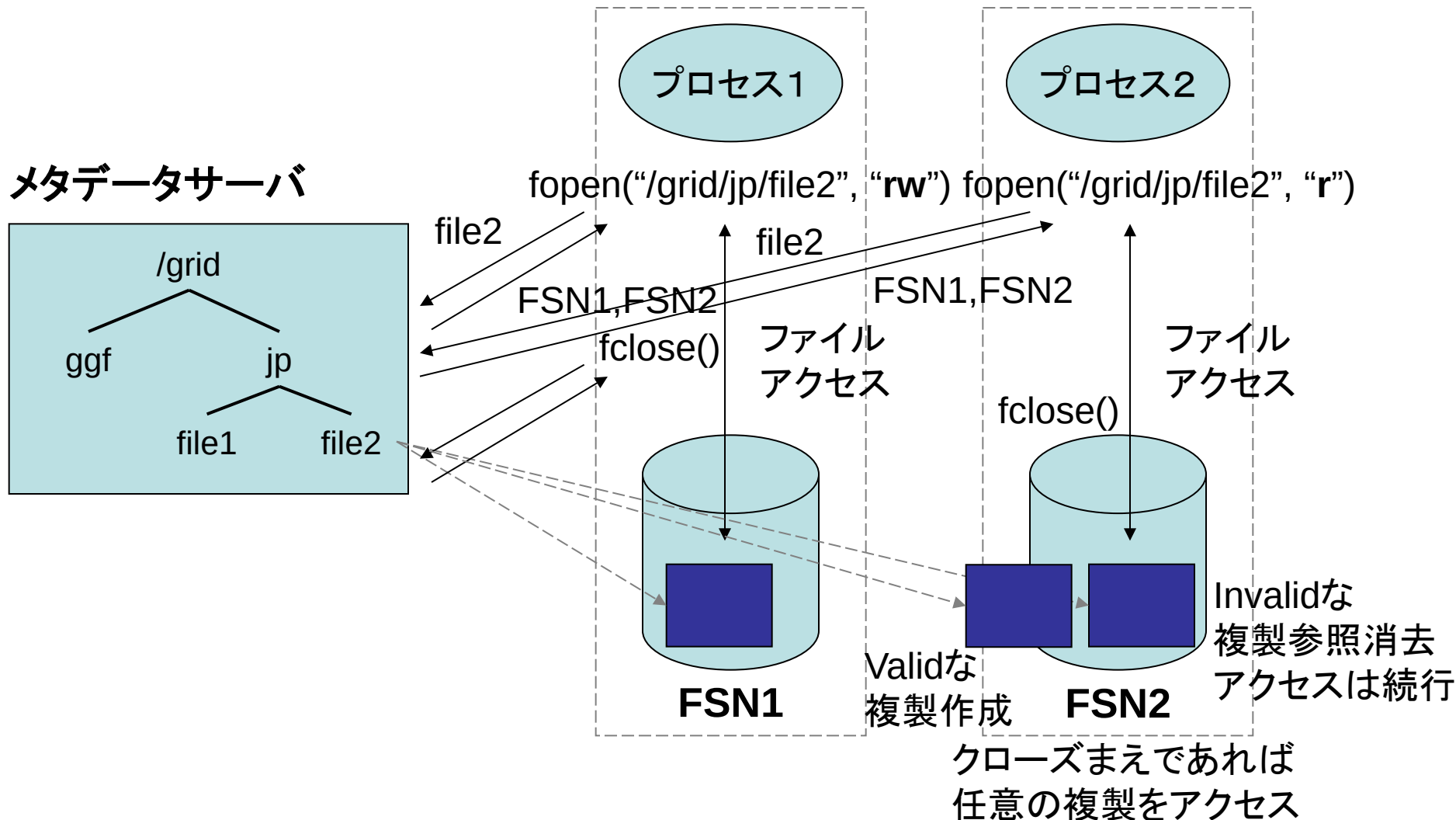


ファイルのアクセス時のgfsdとの接続(2)

- **gfmndから候補となるファイルシステムノードの情報を取得**
 - ▶ CPU負荷とファイルシステムの空き容量
 - ▶ 既存ファイルへのアクセスの場合はファイル複製を持つノード群
 - ④ 自ノードが持つ場合は自ノード
 - ▶ 新規ファイルの場合は自ノード, あるいは近くのノード群
- **ファイルシステムノードの選択**
 - ▶ 自ネットワークアドレスのノードから選択
 - ▶ ネットワークごとのRTTを計測し, 近いネットワーク順にノードを選択
- **gfsdに接続**
 - ▶ 自ホストの場合はUNIXドメインソケットで接続
- **実ファイルのファイルオープンを要求**
 - ▶ 自ホストの場合は, ファイルディスクリプタパッシングで実ファイルのファイルディスクリプタをクライアントに返す
 - ④ 以降は直接参照
 - ▶ リモートホストの場合は, 実ファイルのファイルディスクリプタを表に登録, 登録番号を返す

close-to-open一貫性の保持と複製数維持

● クローズ時に他のファイル複製を消去



ネットワーク切断時の処理

● クライアント-gfmd間

- ▶ 再接続し続行
- ▶ ファイルオープンのフェイルオーバー処理 [2.6から]
 - ◎ オープン中のファイルの再オープン

● クライアント-gfsd間

- ▶ 読込アクセスは別複製アクセスへ自動的に切替
- ▶ 書込アクセスはエラーを返す

● gfsd-gfsd間

- ▶ ファイル複製作成をスケジューリングからやり直し

● gfmd-gfmd間

- ▶ 再接続、ジャーナル転送

→アプリケーション透明にgfmdのフェイルオーバー、停止
gfsdの停止、起動が可能

障害対応と一貫性チェック

- gfmd起動時のメタデータの参照数チェック
- gfsd起動時のメタデータと実ファイルとの一貫性チェック(存在、サイズ)
- アクセス時、複製作成時のチェックサムによるファイル損傷チェック(サイレントデータ損傷対応)
- 複製数維持機能(障害発生時、複製数変更時)
- I/Oエラー発生時のgfsd自動停止
- ディスクフル、障害によるROFSへの移行時に空き容量0とみなす
- Zabbix pluginによる動作監視と自動gfmdフェイルオーバー

まとめ

- Gfarmファイルシステム
 - NPO法人つくばOSS技術支援センターによるサポート
- 大容量、高速なデータ共有
- 耐障害性に優れる
- データ完全性、サイレントデータ損傷対応
- HPCI共用ストレージ、JLDGなど実運用実績