Gfarm: 広域大容量データ解析システム*

建部 修見　関口 智嗣
(産業技術総合研究所 情報処理研究部門)

1　はじめに

高エネルギー物理学，天文学，地球惑星物理学，人ゲノムなどの大規模データ解析を必要とする研究分野では，ハイパフォーマンスコンピューティング，データインテンシブコンピューティング，ネットワーク技術が不可欠となってきた．一つの例は，2006 年より開始される予定になっているスイス CERN の LHC(Large Hadron Collider) 実験プロジェクトである．LHC 実験には 4 つの観測器，実験グループがあり，それらの観測器は毎年ペタバイトオーダの観測データを生成する．それぞれの実験には数十ヶ国規模，数千人規模の素粒子物理学者が参加し，実験データの解析において協力および競争することになる．MONARC プロジェクト [1] では，世界規模の階層的な地域センタの計算モデルについての研究が行われた．この地域センタモデルでは，0 層センタは CERN におかれ，1 層センタはヨーロッパ，アメリカ，アジアなど，2 層センタは各国，3 層センタはそれぞれの大学，研究所におかれる．広域に分散するため，グリッド技術はこれらの実装のための鍵となっている．

Gfarm(Grid Data Farm) はペタバイトスケールのデータインテンシブコンピューティング環境の構築のため，産業技術総合研究所 (AIST)，高エネルギー加速器研究機構 (KEK)，東京大学素粒子物理国際研究センター (ICEPP)，東京工業大学の共同研究で始まった．目標は，LHC の ATLAS 実験による数百テラバイトから数ペタバイト規模の実験データ解析環境の構築である．ICEPP と KEK は共同で日本に ATLAS 実験の 1 層地域センタを構築することになっている．想定しているハードウェアは，それぞれのノードがテラバイト級のローカルディスクを持つ数千台規模の PC クラスタである．CERN からやってくる 600Mbps ほどの実験データは並列に系統的にそれらのディスクに格納され，それぞれの PC で処理される．Gfarm では，LHC 実験に加え，広くデータインテンシブコンピューティングのために以下の機能を提供する．

- ペタバイトスケールのファイルを扱うためのグローバル分散ファイルシステム
- 並列 I/O と並列処理
- 世界規模の認証とアクセス制御
- 数千ノード，広域の資源管理とスケジューリング
- 階層的データ共有と効率的アクセス
- プログラム共有と管理
- システムモニタリングと管理
- 耐故障性 / 動的再配置 / 動的データ復元，再計算

2　Gfarm コアアーキテクチャ

図 1 に Gfarm アーキテクチャを示す．Gfarm は大きく Gfarm クライアント，Gfarm サーバ，Gfarm プールからなり，それらを合わせて Gfarm システムと呼ぶ．Gfarm プールは広域の数千から数万ノードの PC クラスタで構成され，そのローカルディスクを用い Gfarm ファイルシステムが構成される．ペタバイトスケールの大規模ファイルは，物理的には断片に分割され，高速ネットワーク上に分散するディスクに分散配置される．それらファイルの断片の情

*http://datafarm.apgrid.org/

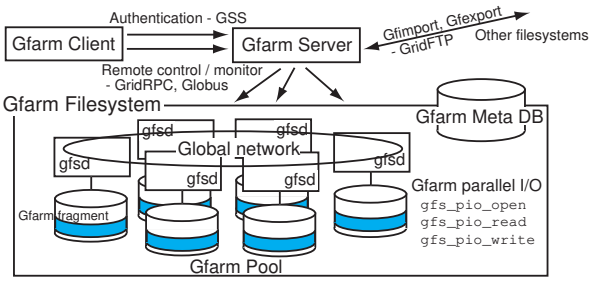

図 1: Gfarm アーキテクチャ

報および論理パス名からの変換等のファイルシステムのメタ情報は，ファイルシステムメタデータベースが管理する．ファイル断片に対する I/O は Gfarm 並列 I/O API を用い，論理パス名によりアクセスされる．この論理パス名は Gfarm ファイル名あるいは Gfarm URL と呼ばれ，`gfarm:/path/name` のように表される．

Gfarm ファイルシステムは，ただのファイルシステムではなく，ペタバイトスケールのデータインテンシブ処理のために，並列 I/O，並列処理を前提として設計されている．基本的な実行イメージでは，プログラムはそれらファイル断片の割り当てられている PC で実行され，ファイルアクセスはローカルアクセスとなる．プログラムは，Gfarm 並列 I/O API を用い論理名である Gfarm ファイル名でファイルを参照，生成するが，実際はローカルなファイル断片を参照し，新しいローカルなファイル断片を作成し，新しい Gfarm ファイル名をメタデータとして登録する．この場合，ファイル I/O のバンド幅は，分散している PC の数にスケーラブルに増加するため，バンド幅的にはペタバイトスケールの大規模データ処理に対応することができると考えられる．

Gfarm ファイルシステムデーモン (gfsd) は Gfarm プールのそれぞれのノードで実行され，主にアクセス制御，リモートディスクのアクセスに利用される．gfsd では，軽量な GFS RPC を用い，低オーバヘッドのアクセスを可能にしている．gfsd は，ファイルアクセス以外にも，Gfarm サーバに登録されている実行形式の動的ローディング，ノードの資源のモニタ，制御も行う．

Gfarm ファイルシステムのメタデータは，論理 Gfarm ファイル名からそれぞれの断片の物理ファイル名へのマッピング，ファイルサイズ，プロテクション，アクセス/ 修正/ 変更時刻，チェックサムなどのファイル状態情報，ファイル複製カタログ，生成ヒストリなどで構成され，Gfarm メタデータベースで管理される．生成ヒストリは，ノードやディスクの障害時におけるデータの再計算，データ正当性の検証のためのデータ生成プロセス解析などに利用される．メタデータは，それぞれのファイル断片のクローズ時に登録され，すべての並列プロセスが終了したとき，その論理 Gfarm ファイル名のメタデータがチェックされる．もしどれかのプロセスがメタデータを正しく登録しないで，エラーなどで終了してしまった場合，そのメタデータは一貫性がとれていないため，チェック時にそのメタデータは消去される．

Gfarm サーバは，Gfarm プール，Gfarm ファイルシステムを効率的に利用するよう拡張したネットワーク可用サーバ (network-enabled server)[4] である．Gfarm サーバはまず，GSS(Generic Security Service)[3] あるいは IPsec を用い，Gfarm クライアントを相互に認証する．認証は Gfarm システムに対して一度だけのシングルサインオンである．認証後，Gfarm サーバは Gfarm クライアントの指示に従い，登録されている並列プログラムを実行する．並列プログラムは，世界中に存在する Gfarm プールで実行される．Gfarm サーバは入力，出力の Gfarm ファイル名をもとに Gfarm メタデータベースに問い合わせ，実行する Gfarm プールのノードスケジューリングを行う．このスケジューリングは，Gfarm ファイルの物理的な Gfarm ファイル断片の格納場所，その複製の格納場所，およびノードの状態などを元に行われる．

Gfarm クライアントは，GUI，*Gfarm* シェルと呼ばれるシェル，そして Grid-based RPC(GridRPC) を用い，Gfarm サーバおよび Gfarm システムにアクセスする．GridRPC は，グリッド環境における容易な遠隔手続き呼び出し方式であり，インターフェー

ス記述言語 (IDL) の動的な読み込みによるクライアント側の簡素化と一貫性の保証，および拡張された多次元整合配列が記述可能な IDL などを特徴としている．ユーザは，大規模ファイルの Gfarm ファイルシステムへの読み込み，ユーザプログラムの登録，実行，および Gfarm ファイルシステムからの書き出しなどをすることができる．また，実行のモニタ，管理もすることができる．

3 Gfarm 並列 I/O API

Gfarm 並列 I/O API は Gfarm ファイルシステムのファイルアクセスを提供する．Gfarm ファイルは，複数のインデックス付けされたファイル断片に分割され，複数のディスクに格納される．Gfarm 並列 I/O API は，それぞれのファイル断片を明示的に並列にアクセスする．並列 I/O API は，Gfarm プールのノードだけではなく，Gfarm サーバでも実行される．

3.1 定義

Gfarm ファイルと Gfarm ファイル断片 *Gfarm* ファイルは Gfarm URL，Gfarm ファイル名で指定される論理的なファイルである．物理的には，複数のインデックス付けされた *Gfarm* ファイル断片に分割され，複数のディスクに格納される．それぞれの Gfarm ファイル断片は Gfarm URL とインデックスで指定される．

Gfarm URL と Gfarm ファイル名 *Gfarm URL* あるいは *Gfarm* ファイル名は Gfarm ファイルシステムにおける Gfarm ファイルのパス名である．Gfarm URL は “gfarm:” で始まる．

Gfarm ファイルハンドル *Gfarm* ファイルハンドルは不透明オブジェクトであり，gfs_pio_close により解放される．すべてのファイル操作は Gfarm ファイルハンドルを用いて行われる．

3.2 ファイル操作

3.2.1 ファイルのオープン，作成

```
char* gfs_pio_open(char *url, int index,
  char *host, int flags, GFS_FILE *gf);
char* gfs_pio_create(char *url, int index,
  char *host, mode_t mode, GFS_FILE *gf);
```

gfs_pio_open は Gfarm URL url とインデックス index で指定された Gfarm ファイル断片をオープンし，新たな Gfarm ファイルハンドル gf を返す．host が指定されない場合は，Gfarm メタデータベースから得られる．flags は GFARM_FILE_RDONLY か GFARM_FILE_RDWR のどちらかで，読み込みモードか読み書きモードかを指定する．gfs_pio_create は Gfarm URL url とインデックス index で指定される新たな Gfarm ファイル断片を Gfarm プールのノード host 上にアクセスモード mode で作成し，新たな Gfarm ファイルハンドル gf を返す．mode はアクセス許可を指定するが，アクセス許可はプロセスの umask でマスクされる．

以下の API は，それぞれのノードが少なくとも一つの Gfarm ファイル断片を持っているという特殊ケースであるが，Gfarm においては典型的な場合の API である．

```
char* gfs_pio_set_local(
          int index, int size);
char* gfs_pio_open_local(char *url,
          int flags, GFS_FILE *gf);
char* gfs_pio_create_local(char *url,
          mode_t mode, GFS_FILE *gf);
char* gfs_pio_local_paths_get(char *url,
          int *npaths, char ***paths);
```

gfs_pio_open_local は Gfarm URL url で指定されるローカルな Gfarm ファイル断片をオープンし，新たな Gfarm ファイルハンドル gf を返す．flags は GFARM_FILE_RDONLY か GFARM_FILE_RDWR のどちらかである．gfs_pio_create_local は Gfarm URL url で指定されるローカルな Gfarm ファイル断片をアクセスモード mode で作成し，新たな Gfarm ファイルハンドル gf を返す．gfs_pio_set_local はこれらに先立ち呼び出され，ローカルな Gfarm ファイル断片のインデックス index とファイル断片の総数 size を指定

する．gfs_pio_local_paths_get は Gfarm URL url で指定されたローカルな Gfarm ファイル断片のパス名のリストと，総数を返す．

3.2.2 ファイルのクローズ

```
char* gfs_pio_close(GFS_FILE gf)
```

gfs_pio_close は Gfarm ファイルハンドル gf をクローズし，Gfarm メタデータベースのファイルサイズ，チェックサムを更新，チェックする．チェックサムはマスターファイルと複製ファイルの同一性を検証するために使われる．

3.3 ファイルアクセス

Gfarm 並列 I/O API では，ファイルアクセスに対しブロッキング，非集合的，個別ファイルポインタの操作を提供する．本節では，紙面の関係上 API のみ紹介する．

```
char* gfs_pio_read(GFS_FILE gf,
    void *buf, int size, int *nread);
char* gfs_pio_write(GFS_FILE gf,
    void *buf, int size, int *nwrite);
char* gfs_pio_seek(GFS_FILE gf,
    file_offset_t offset, int whence);
char* gfs_pio_flush(GFS_FILE gf);
int gfs_pio_getc(GFS_FILE gf);
int gfs_pio_ungetc(GFS_FILE gf, int c);
char* gfs_pio_putc(GFS_FILE gf, int c);
char* gfs_pio_getline(GFS_FILE gf,
    char *s, size_t size, int *eofp);
char* gfs_pio_puts(GFS_FILE gf, char *s);
char* gfs_pio_putline(GFS_FILE gf,
    char *s);
```

4 Gfarm コマンド

Gfarm ファイルシステムにおける ls，mkdir，rm などのファイル操作コマンドも提供される．詳細は紙面の都合上割愛する．

5 まとめ

ペタバイトスケールのデータインテンシブコンピューティングには，ファイルアクセスのバンド幅を向上させるため，多数の PC クラスタのローカルディスクを利用することが有効であると考えられる．Gfarm では，それら多数の PC のローカルディスクを一つのファイルシステムとしてみせることにより，プログラミング，管理の複雑さを回避し，効率的な利用を目指している．これらシステムの構築には，Ninf[6, 5] などの Grid-based RPC および Globus[2] などのより低レベルのグリッドサービスなどのグリッド技術，クラスタ技術が必須である．Gfarm は，CERN LHC 実験プロジェクトに同期し，2005 年までにはペタスケールのオンラインディスクシステムの構築を目指しているが，それ以外のバイオインフォマティックス，天文学，地球惑星物理学などのデータインテンシブコンピューティングへの適応も検討している．

参考文献

[1] MONARC Collaboration. Models of Networkd Analysis at Regional Centres for LHC experiments: Phase 2 report. Technical Report 001, CERN/LCB, 2000. http://www.cern.ch/MONARC/.

[2] Ian Foster and Carl Kesselman. Globus: A metacomputing infrastructure toolkit. *Intl J. Supercomputer Applications*, 11(2):115–128, 1997.

[3] John Linn. *Generic Security Service Application Program Interface Version 2, Update 1*, January 2000. RFC-2743.

[4] Satoshi Matsuoka, Hidemoto Nakada, Mitsuhisa Sato, and Satoshi Sekiguchi. Design issues of network enabled server systems for the Grid. *Lecture Notes in Computer Science*, 1971:4–17, 2000. Proceedings of Grid Computing – GRID 2000.

[5] Hidemoto Nakada, Mitsuhisa Sato, and Satoshi Sekiguchi. Design and implementations of Ninf: towards a global computing infrastructure. *Future Generation Computing Systems*, Metacomputing Issue, 1999.

[6] Satoshi Sekiguchi and Mitsuhisa Sato. World-wide computing infrastructure: Global and local partnership. *Proceedings of the second AIZU International Symposium on Parallel Algorithms / Architecture Synthesis*, pages 25–30, 1997.