社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－ARC－167（23）
2006－HPC－105（23）
2006／2／28

動的なデータ発生・移動に対応した一意にアクセス可能なデータグリッドシステムの提案

池部 実[†]　増田 慎吾[†]　藤川 和利[†]　砂原 秀樹[†]

科学技術計算では，グリッドシステムを利用して多数のジョブが並列に実行され多くのデータが発生する．最終的な結果の精度を高めるため，これらのデータは最終結果の状態だけでなく，ジョブ実行の途中で発生したものも利用される．グリッドシステムでは，ジョブが終了した場合，その所在がユーザに通知されるが，ユーザがジョブ実行途中のデータがどのノードで発生したかを知ることは難しい．本研究報告では，ジョブ実行の状態に関わらず常に同じ方法でデータにアクセス可能なデータグリッドシステムを提案する．本研究で提案するデータグリッドシステムでは，メタデータを用いデータの属性を表現し，メタデータにデータを特定することで一意にアクセスする機構を用いる．また，生成されるデータにメタデータを付加するためのデータ管理機構を提供している．

Proposal of the Data Grid System with an uniquely accessible method to data which are dynamically generated and migrated.

Minoru Ikebe[†]　Shingo Masuda[†]　Kazutoshi Fujikawa[†]　Hideki Sunahara[†]

In scientific simulations using Grid environments, huge data has been generated distributed. To get better final results, users want to analyze not only results produced when job execution is completed but also the intermediate result produced during jobs are still executing. However, since most grid systems provide the results for users only when job execution is completed, users cannot recognize the location of data during jobs are executing. In this paper, we propose a new Data Grid system where users can access data in a unique manner. In the proposed Data Grid system, data have meta-data to represent their own attribute, and users can access the data by specifying meta-data values. Moreover, our Data Grid system provides a mechanism to automatically attach meta-data values to the data which are generated in Grid environments.

1　はじめに

近年，計算機とネットワークの性能向上とともに，組織を越えて様々な種類の計算機を共有し，通常では得ることの難しい多くの計算機資源を獲得するためのグリッド [1] と呼ばれる分散処理技術が注目を集めている．グリッドの中でも，データ処理のためのグリッドをデータグリッドと呼ぶ．データグリッドは，大きく分けて 2 つが存在する．

(1) 膨大なデータを管理するためのデータグリッド

(2) データベース連携のためのデータグリッド

(1) は，高エネルギー物理学に代表されるような，実験や計算処理によって，データが大量に発生し，発生したデータを共有・管理するものであり，(2) は，バイオサイエンス分野などに代表されるような，異なる組織が管理し，データ形式などが異なるデータベースにおいて，メタデータを用いて横断的な検索を行い，データベース連携を行うものである．

グリッドで実行されるアプリケーションの特徴は，出力されるデータ量が膨大であること，並列性が高いことである．例えば，スイス CERN の LHC (Large Hadron Collider) 実験 [2] では，観測器から 1 秒間に約 100 メガバイト，年間ペタバイトの観測データが発生する．高エネルギー物理実験のデータ解析処理は，陽子同士の衝突反応が 1 事象ごとに独立であるため，独立して解析することができる．データを CPU ごとに割り当てることで，並列に処理することができる [3]．LHC 実験のようなシステムでは，多くの計算資源およびストレージ資源があらかじめ用意されていることが多く，グリッドシステムの特徴である Virtual organization(以下，VO)[4] を動的に構成するのではなく，決まった組織での運用が主である．一方，バイオサイエンス分野の研究プロジェクトでは，グリッドシステムによる多くの計算資源の提供を期待しており，VO を動的に形成しデータの解析やシミュレーション等を行っている．

グリッドシステムを利用して VO を動的に形成しジョブを投入する場合には，ユーザは実際にどの計算機を使うのかを意識する必要がなく，ジョブはアイドル状態もしくは低負荷状態の複数の計算ノードへ割り当てられ，並列にジョブが実行される．このとき，

[†]奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, NAIST

データは処理が行われた個々の計算ノード上に (一時的に) 分散して保存される．このため，従来のグリッドシステムでは，データを NFS で共有したり，ftp で他の計算機から取得する方法により，ジョブが出力したデータをユーザに提示されていた．

このほか，バイオサイエンス分野では，ゲノム，タンパク質に関する大量のデータが既に存在し，データ解析やシミュレーションによっては，これらのデータを保持するデータベースの中から，必要なデータを取り出して，処理を進める場合もある [5]．また，実行されたジョブの最終的な結果や実行途中に出力される中間データを動的に利用して処理を進めることもある．しかしながら，従来のグリッドシステムでは，中間データを逐次ユーザに提示することは想定していない．データベースによって管理されるデータや実行中のジョブの中間データ，実行が終了したジョブのデータなど，必要に応じてこれらのデータをユーザに提示する必要があり，これらの性質が異なり分散して存在するデータに対して，その所在を意識することなく容易にアクセスできる環境を提供しなければならない．そのため，本研究では，どのような種類のデータに対しても，ユーザがそのデータの所在を意識せずに統一されたインタフェースでアクセスできるデータグリッドシステムを構築を目指す．

本研究で構築するデータグリッドシステムに対する要求は以下の通りである．

- 膨大なデータの中から必要なデータを取り出すこと
- どのようなデータに対しても，同じインタフェースでアクセスできる

本研究では，グリッドシステムによりデータ解析が加速することを期待されているタンパク質立体構造予測問題をサンプルアプリケーションとして取り上げる．具体的には，神戸大学で開発されているタンパク質立体構造予測システム Rokky[6][7] を，本研究で開発するデータグリッドシステムを通して利用する．Rokky では，FASA (Fragment Assembly Simulated Annealing) を用いた SimFold[8][9] を利用して処理を行っている．Simulated Annealing とは，組み合わせ最適化問題のひとつの解法である．ひとつのジョブに対して，複数の乱数を発生させ，それぞれが独立に処理されるアプリケーションである．また，Rokky が生成するデータは，テキストファイルである．

本研究報告では，メタデータによるデータ管理機構を用い，データの所在を意識することなく，常に同じインタフェースでデータへアクセスできるデータグリッドシステムを提案する．2 節では，科学技術計算などのアプリケーションを利用する上でのデータグリッドにおける要求を述べる．3 節では，データグリッドの関連研究について述べる．4 節では，メタデータを用いた，一意にアクセス可能なデータグリッドシステムについての提案を行う．5 節では，現在実装中のシステム構成を述べる．6 節では，今後の予定や課題について述べる．

2 対象とするデータグリッドシステム

本節では，対象とするデータグリッドシステムにおいて必要な要件や必要となる機能を述べる．まず，データグリッドに必要な機能は次の 2 点である．

- ストレージ管理
- データ管理

ストレージ管理は，グリッドに参加する各ホストのストレージ資源，すなわち，ハードディスクの容量を把握し，どのホストにどれぐらい保存できるという情報を，グリッドのジョブスケジューラに提供する．この情報や，他の資源情報，例えば，CPU の負荷率などをふまえて，どのホストにどのぐらいのジョブを割り当てるかをジョブスケジューラが決定する．

データ管理は，ユーザに対して，データの所在を意識せずにデータへアクセス (以下，シームレスなアクセス) する機能を提供するため，データがどこにあるかを管理する機能である．通常，データ資源を管理するために，メタデータを用いる．メタデータとは，データ属性を記述したデータである．また，メタデータを用いることにより，データがもつ属性をもとに検索することも可能となる．

本研究では，データ管理に注目し，従来のデータアクセスと比較して，シームレスなアクセスの定義を行い，シームレスなアクセスをユーザに提供するためのシステムについて述べる．従来のデータアクセスでは，ftp などの転送プロトコルを用いてアクセスすることが多く，ftp を利用する場合には，ID/Password が必要である．しかしながら，ユーザがアクセスするために，利用するすべての計算機に対して，アカウント発行することは管理コストが高くなる．また，該当するデータがどこに存在するかを認識している必要がある．目標とするシームレスなアクセスとは，ユーザがアクセスするためにアカウントを発行することなく，かつ，どのようなデータに対しても，ユーザがデータの所在を知ることなく，統一したユーザインタフェースでデータアクセス (以下，一意にアクセス) することである．

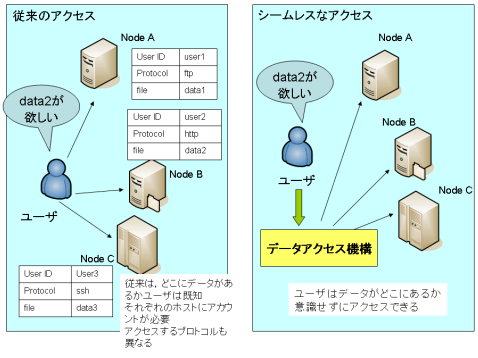

図 1: 従来のアクセスとシームレスなアクセス

3 関連研究

ここでは，データグリッドシステムについて関連研究を述べる．各システムのデータ管理について注目する．特に，1 節で述べたデータグリッドシステムへの要求について比較する．

- 膨大なデータの中から必要なデータを取り出す
  - **(1)** データ検索
  - **(2)** メタデータの利用目的
  - **(3)** データ生成時のユーザが行う作業
- どのようなデータに対しても，同じインタフェースでアクセスできる
  - **(4)** 一意にアクセスするためのユーザインタフェースの提供
  - **(5)** 管理するデータの対象

3.1 SRB

SRB (Storage Resource Broker)[10] は，SDSC(San Diego Supercomputing Center) で開発された様々な種類のデータ資源へネットワークを越えて接続するものであり，データセットの複製に対してアクセスするためのクライアント・サーバ型ミドルウェアである．SRB は，MCAT (Metadata CATalog) と呼ばれる，データの属性と物理的な位置を基にしてデータへのアクセス方法を提供する．SRB では，データやデータベースに対して，同一のユーザインタフェースでアクセスすることは可能である．しかし，データの登録とメタデータの記述は，ユーザによって行われるため，ユーザの作業コストは高くなる．従って，SRB が管理する対象データは，ユーザが登録したものに限られる．

3.2 Globus Toolkit

Globus Toolkit[11] は，米国アルゴンヌ国立研究所と南カリフォルニア大学によって共同開発されたグリッド構築のためのツールキットであり，世界中の多くのグリッドプロジェクトで採用されている．Globus Toolkit のデータ管理は，データ転送機能とデータ複製機能を提供している．データ複製機能では，データの複製を作成した際，ユーザが複製されたデータの中からアクセスが最適になるように元のデータと複製されたデータの位置管理している．データの登録やメタデータの記述は，ユーザによって行う必要がある．従って，管理する対象データは，ユーザが登録したものに限られる．

3.3 Gfarm

Gfarm (Grid Datafarm)[12] は，産業技術総合研究所グリッド研究センターが開発した大容量，高信頼，高性能低コストを目的としたファイルシステムを実現するためのミドルウェアである．Gfarm グリッドファイルシステムは，PC のディスクを仮想ディレクトリで統合し，クラスタ，グリッドにおけるネットワーク共有ファイルシステムとして開発されている．データは，一定のサイズに分割して，分散して保存される．データの位置などを記述したメタデータを用いて，データ格納場所を気にせずアクセス可能である．データ検索は，一度分割したファイルを復元する必要があるため，現在は行われてない．Gfarm は，ファイルシステムとして提供されているため，発生したデータすべてを管理することができる．

3.4 関連研究の問題点のまとめ

関連研究についての問題点を表 1 にまとめる．

データ検索が行えるシステムは，ユーザによるデータ登録，メタデータ記述作業が発生している．これを発生するすべてのデータに対して行うことは，非常に困難である。データ検索を行うシステムでは，データ登録，メタデータ記述作業のコストを下げる必要がある。SRB や Globus Toolkit では，データへ一意にアクセスするために，ユーザがあらかじめシステムへデータを登録しておく必要がある。

4 シームレスなアクセスを提供するデータグリッドシステムの提案

本研究では，3.4 節の問題点を解決し，発生するデータに対してメタデータを付与しデータ管理を行い，メタデータによるデータ検索を可能にし，どのようなデータに対しても，一意にアクセスできるユーザインタフェースを提供するシステムを提案する．

4.1 メタデータ管理機構

4.1.1 メタデータの定義

メタデータとは，データについてのデータ属性を記述したデータである．本研究では，基本メタデータ，アプリケーションメタデータの 2 種類のメタデータを定義する．基本メタデータとは，Dublin Core[13] を拡張したメタデータである．Dublin Core とは，WWW 上におけるリソースに関する情報を記述して有用な情報の探索・発見に役立てる目的で作られたものである．

アプリケーションメタデータとは，実行するアプリケーションに特有な情報として定義する．サンプルアプリケーションとして取り上げたタンパク質立体構造予測システム Rokky では，入力データとなるアミ

表 1: 関連研究の比較

データグリッドシステム	(1)	(2)	(3)	(4)	(5)
SRB	○	位置管理/データ検索	ユーザによるデータ登録	△	任意
Globus Toolkit	○	位置管理/データ検索	ユーザによるデータ登録	△	任意
Gfarm	×	位置管理	なし	○	すべて

ノ酸配列，実行時間の制約を表すジョブの開始時刻，ジョブの終了時刻などがある．また，Rokky で実行される SimFold では，各計算機で処理されたデータを用いて，クラスタリング分析を定期的に行う．クラスタリング分析のために，データを一カ所に集める必要がある．実行しているジョブで，クラスタリング分析に必要なデータをどの計算機から取得すればよいかを，システムで管理するためにジョブを識別する識別子 groupid を用いて管理する．

それぞれのメタデータの要素について，表 2 と表 3 に示す．

表 2: 基本メタデータ

要素名		定義
groupid		ジョブスケジューラによって 割り当てられる一意な名前
title		リソースに与えられた名前
creator		リソースの内容に主たる 責任を持つ人や組織
subject		リソースの内容に含まれるトピック
description		リソースの内容の説明
publisher		このリソースを提供している母体
contributor		リソースの内容に協力, 貢献している人や組織
date	ctime	リソースの生成された日時
	atime	リソースにアクセスした日時
	mtime	リソースが編集された日時
type		リソースの内容の性質
format		リソースの形式
identifier		リソースへの曖昧さのない参照
source	host	リソースが保存されているノード
	path	リソースが存在する絶対パス
language		リソースの内容を記述している言語
relation		関連するリソースへの参照
coverage		リソースの範囲もしくは対象
rights		リソースに適用される権利に関する情報

表 3: アプリケーションメタデータ

要素名	定義
groupid	グリッドシステムに投入されたジョブを 区別するための識別子
Name	ジョブを投入したユーザ名, またはそのユーザの名前
Mail	ジョブを投入したユーザのメールアドレス
TargetName	Jobid とは別の名前 ジョブ投入時にユーザが決定
AminoAcid	実行するアミノ酸配列
JobStart	ジョブの開始時刻
JobEnd	ジョブの終了時刻

4.1.2 メタデータ管理

基本メタデータの取得には，一部の要素が発生したデータから取得可能である．主に，位置管理のためのメタデータを取得する。残りのメタデータに関しては，ユーザが自由に記述することが可能である．アプリケーションメタデータの取得は，アプリケーション自体の改変が必要となる．このため，アプリケーションメタデータの取得については，ユーザがアプリケーションを実行後に，入力することとした．一度，入力すれば発生するデータとの関連付けは，システム側で行う．取得したメタデータは，サーバ側に送られ，データベースで管理する．

4.2 データアクセス機構

4.2.1 データ空間

ユーザがアクセスするデータ空間について述べる．データ空間は，フラットな空間である．ひとつのディレクトリの下にすべてのデータが仮想的に存在している (図 2)．メタデータの要素を条件指定することで，フラットなデータ空間の中に，条件に合致する仮想的なディレクトリが作られる．例えば，メタデータの条件として，ジョブの識別子 groupid に実験 A, B, C と指定することにより，仮想的なディレクトリが作られる (図 3)．

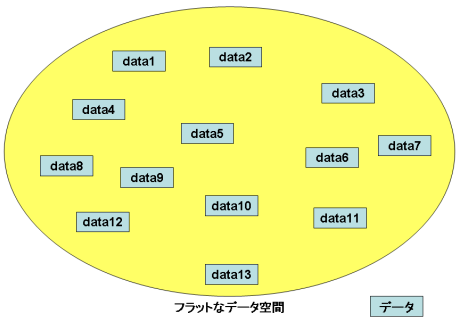

図 2: フラットなデータ空間

4.2.2 データ検索

ユーザは，メタデータの要素についての条件を指定してデータを検索する．データ表示のためのコマンドの引数に，入力されたアミノ酸配列，ジョブの開始時刻から終了時刻の範囲指定を行うことで該当するデータが表示される (図 4)．

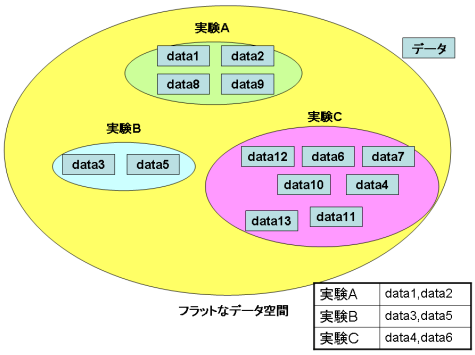

図 3: メタデータによる仮想ディレクトリ

```
gls -s "AMINOACID=ACGTRFHIK \
LYIPFGHTREWETSKELKITREYPIKL \
GCVAQREWDFHGKLMNCVASPIILTEQ \
LKMNHESAAQFDKPQER, \
TIMEFROM=2006-1-13 0:0:0, \
TIMETO=2006-1-15 0:0:0"
```

図 4: データ検索例

4.2.3 データへ一意にアクセス

データへ一意にアクセスするとは，どのようなデータ (ここでは，最終的な計算処理結果，計算途中に出力される中間データなどを指す) を同一のインタフェースからアクセス可能にすることである．発生するデータからメタデータを取得し，管理することで，メタデータを用いてデータを検索とデータアクセスのための位置管理を行っている．ユーザがアクセスするためのユーザインタフェースは，データ表示，提案システム上に存在するデータを既存のアプリケーションから実行するためのプログラムを提供している．

5 システム構成

本研究での，提案システムの構成は，図 5 のようになる．

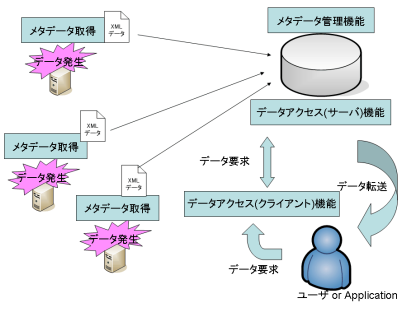

図 5: システム構成と実装

システムは，ノード側でデータ発生時にメタデータを取得するメタデータ取得機能，サーバ側でメタデータを管理するメタデータ管理機能，実際に，データへアクセスするためのユーザインタフェースと，データアクセスのリクエストを処理して，ユーザへ結果を返すデータアクセスサーバ機能からなる．本システムでは，Globus Toolkit を用いて，システム構築を行い，提案システムの実装には，C 言語を用いた．

1. メタデータ取得機能
   メタデータを取得するために，Gamin というファイルとディレクトリ監視用のライブラリを用いて，ファイルがディレクトリに出力されたことを検知する．出力されたファイルを lstat() システムコールにより，データ属性をメタデータとして取得する．取得したメタデータは，XML データ形式でサーバへ送信する．

2. メタデータ管理機能
   ノード側から送られてきた XML データの構文解析を行う．構文解析には，libxml2 を使用した．libxml2 は，C/C++用の XML 構文解析用ライブラリである．XML データからメタデータの要素を取り出し，データベースへと格納する．データベースには，PostgreSQL を用いた．

3. データアクセス (クライアント側)
   データアクセスを行うユーザに対して，ターミナルからデータへ一意にアクセスすることのできるコマンドを提供している．現在，実装したコマンドは，”gls” ，”gexec”である．gls は，図 4 のように，オプションに検索条件を指定することによって，条件にマッチするデータを表示する．gexec は，”gexec コマンド名 検索条件”という形式で，オプションに，コマンドを指定することによって，コマンドに検索条件にマッチするデータを引数として実行することが可能である．

4. データアクセス (サーバ側)
   メタデータ管理サーバは，クライアントからの検索条件をもとに，SQL 文を生成し，データベースを検索し，条件に該当するデータを探す．該当するデータがひとつの場合は，そのまま，クライアントがデータアクセスするために，データの位置を送る．該当するデータが複数の場合は，システム側がデータに優先順位をつけて，クライアント側へ提示し．ユーザは，優先順位のなかから，該当するデータを選択し，データアクセスする．データアクセスには，GridFTP を用いている．

6 現状と今後の課題

現在の実装においては，ジョブを受け付けるノード側で，ジョブが出力するディレクトリを監視し，データが発生すると，データからメタデータを取得する．取得したメタデータは，メタデータ管理サーバへ送信する．メタデータ管理サーバは，受信したメタデータ

をデータベースへ格納する．データアクセスは，ユーザ側で，メタデータを指定して，データを検索する機能を実装した．

本システムの評価は，要求事項である，膨大なデータの中から必要なデータを取り出すことについては，管理するデータの中からユーザがデータを取り出すのに，要する時間についての評価を行う．

現在は，LAN 内に存在する計算機での実験を行っている。今後は，動的に計算機資源を確保する VO 上での実験を行う．今回は，タンパク質立体構造予測を対象としたが，様々なアプリケーションへの対応を行っていく予定である．

7 おわりに

我々の目標は，グリッドコンピューティングにおいて，演算処理能力の向上を図ること，ユーザにとって使いやすい環境を構築することである．この目標のために，本研究報告では，動的に発生するデータや，移動するデータに対して，一意にアクセスすることのできるデータグリッドシステムを提案した．提案したシステムは，ユーザがデータの所在を意識することなくデータへアクセスできる設計と実装であることを述べた．

謝辞

タンパク質立体構造予測システム Rokky を開発している神戸大学高田研究室の皆様から様々な助言を頂き，本システムの開発にご協力して頂いたことに深く感謝いたします．

参考文献

[1] Ian Foster, Carl Kesselman. ”The GRID2 Blueprint for a New Computing Infrastruc ture”, Morgan Kaufman, May 2003.

[2] CERN LHC, http://lhc.web.cern.ch/lhc/

[3] 渡瀬　芳行，森田　洋平: グリッドコンピューティング:3. アカデミック分野での応用 b. 高エネルギー物理学におけるグリッド技術の応用　情報処理, Vol.44 No.6 pp.608-614 (Jun. 2003).

[4] Ian Foster. ”WHAT IS THE GRID? A THREE POINT CHECKLIST”, DAILY NEWS AND INFORMATION FOR THE GLOBAL GRID COMMUNITY /JULY 22, 2002: VOL.1 NO.6

[5] 伊達　進，奥村　利幸，秋山　豊和，下條　真司，松田　秀雄，中村　春木: グリッドコンピューティング:3. アカデミック分野での応用 a. バイオグリッドプロジェクト「スーパーコンピュータネットワークの構築」 情報処理, Vol.44 No.6 pp.601-607 (Jun. 2003).

[6] ROKKY Protein Structure Suite, http://www.proteinsilico.org/rokky/

[7] W. Jin, T. Furuta, S.J. Park, N. Koga, Y. Fujitsuka , G. Chikenji and S. Takada: ROKKY: structure prediction server that integrates PDB-BLAST, 3D-Jury, and the SimFold fragment assembly simulator, CASP6 Abstracts , pp. 128-129 (2004).

[8] S. Takada: Protein Folding Simulation With Solvent-Induced Force Field: Folding Pathway Ensenble of Three-Helix-Bundle Proteins, Proteins:Structure,Function and Genetics , No. 42, pp. 85-98 (2001).

[9] Y. Fujitsuka, S. Takada, Z. A.Luthey-Schulten, and P.G. Wolynes: Optimizing Phys-ical Energy Functions for Protein Folding, Pro-teins: Structure, Function and Bioinformatics , Vol. 54, pp. 88-103 (2004).

[10] SRB, http://www.sdsc.edu/srb/

[11] Globus Toolkit, http://www.globus.org/toolkit/

[12] Gfarm, http://datafarm.apgrid.org/

[13] Dublin Core, http://dublincore.org/