

広域分散ネットワーク環境におけるデータのセマンティクスを用いたデータ管理・アクセス手法の提案と評価

池部 実[†] 藤川 和利[†] 砂原 秀樹[†]

ユビキタスコンピューティング環境では、大量のデータが創出される。グリッドコンピューティングを用いた科学技術計算や、インターネット上に散りばめられたセンサから出力されるデータを取り扱うアプリケーションは、定常的にデータを出力し続けている。ユーザが目的のデータへアクセスする際に、データの位置を意識してアクセスしなければならない。また、ユーザがデータへアクセスするためには、データのファイル名やアクセス方法を知らなければならない。常にデータが発生し続ける環境においては、目的のデータ名を知ることは困難である。しかしながら、ユーザはデータを出力しているアプリケーション名や、出力された時刻などの様々なデータ属性を知っている。本論文では、ユーザが既知のデータ属性をもとにしたデータ管理およびアクセス手法の提案を行う。現在、提案手法をもとにして行っているプロトタイプ実装について述べる。

A data management/access mechanism using data semantics for distributed data-centric network environments

Minoru IKEBE[†] Kazutoshi Fujikawa[†] Hideki Sunahara[†]

Distributed data-centric applications constantly generate a huge amount of data on the Internet, especially in ubiquitous computing environments. As such data may not stay the same storage space due to the limitation of the storage resource, users have to be aware of the location of data to access them. However, users will not know the exact data name or the access method. In some case, users know the attributes of target data such as purpose, date, and/or application of data generation. In another case, users know only the original location of target data. In this paper, we propose a new data management/access mechanism using data attributes. We have been developing a prototype system to verify the efficiency of our proposed mechanism.

1 はじめに

グリッドコンピューティング [1] を用いた科学技術計算では、シミュレーションプログラムや観測機器から出力、生成される大量のデータが広域ネットワーク上に分散して蓄積されていく。

ユーザは、大量のデータの中から目的のデータへ容易にアクセスできる仕組みを求めている。例えば、グリッドコンピューティング環境の場合、様々な組織の計算機資源を用いるため、異機種環境となる。ユーザがデータアクセスを行うと、各組織では資源の提供方法が異なるためアクセスの不均一がおきる。また、従来のデータアクセスではデータの名前であるファイル名をキーとしたアクセスであった。目的のデータへアクセスする際は、データの名前を知ることが前提条件となる。しかし、次々と発生するデータの名前をすべて把握することは困難である。World Wide Web においては Google[2] のような検索エンジンで、サイト名や URI を知らなくてもキーワードを用いてアクセスすることが可能である。広域分散ネットワーク環境においても、目的のデータに対して、そのデータの属性値を用いてアクセスできるサービスが求められる。

科学者などは、グループ内で実験データを簡単に管理・共有する仕組みを必要としている。これは、

個人レベルでストレージの確保が難しいことや、他人と共有するためにデータを分類するということは行われていないことが多いためである [3][4]。ユーザは、直感的にデータ分類できるインタフェースを必要としている。

グリッドコンピューティング環境では、複数の組織の計算機資源を仮想化して利用することを、Virtual Organization(以下、VO)と呼ぶ。本論文では、データに注目した VO を構築する。

以下、第 2 節では、広域分散環境においてデータを扱う際のユーザからの要求によって、データ管理システムに求められる機能要件を述べる。第 3 節では、既存の広域分散環境におけるデータ管理、アクセスシステムについて述べる。第 4 節では、データ属性を用いてデータ管理およびデータアクセス手法についての提案を行う。第 5 節では、第 4 節で述べた提案を実現するためのプロトタイプ実装について述べる。第 6 節では、提案システムと関連研究の機能要件の比較した。第 7 節では、現在取り組んでいる課題や、今後の予定について述べる。第 8 節では、本論文をまとめる。

2 データ管理における要求事項

本論文では、インターネット上に広がるデータに対しての管理、アクセス手法について注目する。イ

[†]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, Nara Institute of Science and Technology

インターネット上に存在する計算機を仮想化するグリッドコンピューティング環境で発生するデータの特徴は、Write Once Read Many (WORM) であることがあげられる。WORM 型のデータとは、一度だけ書き込みが行われ後に大量の読み込みが行われるデータである。グリッドコンピューティングでは一度出力されたデータを一次データとして利用する。ユーザ・アプリケーションは一次データを利用して新たなデータを出力するため一次データは修正されることはない。

2.1 データアクセス・共有におけるユーザからの要求

広域分散ネットワーク環境におけるデータアクセス、データ共有におけるユーザからの要求事項を以下にあげる。

- シームレスなデータアクセス
- 複数属性を用いたデータアクセス
- 容易なデータ分類
- アプリケーション間の連携

シームレスにデータアクセスするとは、ユーザはデータが広域分散環境において、どこに位置し、どのようにアクセスしているかどうかを意識せずにアクセス可能なことである。

複数のデータ属性によるデータアクセスとは、ユーザが指定したデータ属性によって、目的のデータへアクセス可能なことである。しかし、広域分散ネットワーク環境ではユーザが大量のデータの中から目的データのファイル名を知ることは難しい。このようなことから、ユーザはセンサがいつデータを出力したかなどのデータ属性を用いたデータアクセスを望んでいる [5]。

ユーザは、データを共有する際に、容易にデータ分類できる仕組みを必要としている。ユーザ自身がデータを利用する場合、自分自身がわかる形で整理すれば十分である。他人とデータを共有するためには、他のユーザと共通の整理方法が求められる。しかしながら、ユーザ間で共通の整理方法を取り決めて、データを分類が行われていない。データ分類を複数のユーザで行うことができる機能が求められる。

アプリケーション間の連携とは、あるアプリケーションから出力されたデータを、別のアプリケーションの入力データとすることができることである。アプリケーション間の連携を行うためには、データ操作コマンドや API が必要である。

2.2 データ管理システムに求められる機能要件

第 2.1 節では、ユーザからの要求事項を述べた。ユーザからの要求事項に基づき、広域分散ネット

ワーク環境におけるデータ管理システムに求められる機能要件について述べる。データ管理システムの機能要件を以下にあげる。

- (1) シームレスデータアクセスのための透過性
- (2) データ位置情報管理
- (3) アプリケーション間連携のためのデータ操作コマンド
- (4) データ分類・アクセスのためのデータ属性記述
- (5) データ属性の自動取得
- (6) データ属性を用いたデータアクセス

シームレスアクセスとは、分散システムに必要な透過性を満たし、ユーザへデータアクセスを提供することである (図 1)。本論文が想定する透過性は、位置透過性、アクセス透過性、プロトコル透過性、移動透過性である。

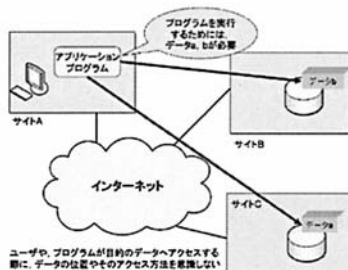


図 1: 分散するデータ資源に対してシームレスにアクセス

データ位置情報管理とは、ユーザがデータアクセスを行った際、実際の位置を提供する機能である。データが出力された際に、データの位置を取得する。また、データの移動が発生した際に、ユーザが同じ場所へアクセスしてもデータが存在しないためアクセスできなくなる。データが移動したことを検知し、移動した先の位置情報を管理する機能が必要となる。

データ属性は、従来のシステムではユーザが手動で入力されていた。しかしながら、大量に発生するデータに対してユーザが手動で記述することは難しい。そのため、発生するデータに対して自動的にデータ属性を取得する機能が必要となる。

3 関連研究

本節では、広域分散ネットワーク環境におけるデータ管理・アクセス手法についての関連研究を述べる。

3.1 SRB

Storage Resource Broker(SRB)[6]は、San Diego Supercomputer Center(SDSC)が開発しているデータグリッドシステムである。SRBは、資源へのアクセスの仮想化を行い、分散する異機種環境にある資源への仲介を行う。SRBは、メタデータを用いて資源への仲介を容易にしている。

SRBでは、ユーザが識別しやすいデータ属性であるファイル名と、SRBシステム内のユニークな識別子をマッピングする。マッピング情報とデータの物理的な位置情報は、MCAT(Metadata CATalog)サーバと呼ばれるメタデータ管理サーバ上に保存され、管理される。

SRBのメタデータは、Dublin Coreメタデータと科学技術の専門分野のメタデータから構成されている。SRBのメタデータは、生成されたすべてのデータに対して、ユーザが記述する必要がある。SRBでのメタデータの利用方法は、データ属性を記述して、その値に該当するデータを表示するために用いている。SRBは、データの移動は発生しない前提でシステムを構築している。しかし、複製管理機能で、データ移動管理機能を代用可能である。

3.2 Gfarm

Gfarm[7]は、産業総合研究所グリッド研究センターが開発している広域ネットワーク環境におけるグリッドファイルシステムである。Gfarmは、高信頼、高性能なグローバルファイルシステムを低コストで提供する。Gfarmは、"/gfarm"という仮想ディレクトリをそれぞれの計算機上にマウントする。Gfarmでは、ひとつのファイルを分割し、分散して保存する。ユーザ側から見るとひとつのファイルとして見える。Gfarmは、ファイル名を用いたデータの提供を行っている。ユーザは、データの目的や日付、どのアプリケーションのデータなどといったデータの属性によってデータへアクセスはできない。

4 広域ネットワーク環境におけるデータ属性を用いたデータ管理・データアクセス手法の提案

本論文では、データ属性を用いてデータ管理およびデータアクセスする手法を提案する。データ属性を用いることで、ユーザの直感をデータアクセスに反映させることが可能となる。

4.1 データ属性とメタデータ

データ属性とは、データに関する様々な属性を示すものである。

従来のデータアクセス方法は、ファイルの名前をベースとしたアクセスであった。ユーザは、ファイ

ルの名前のなかに、知っている属性情報を記述していた。例えば、データを作成した日時、データの目的などの属性情報をファイル名に記述していた。本論文では、ユーザが無意識に利用していたデータ属性に注目する。

メタデータは、データについてのデータ属性と属性値を記述したデータである。本論文では、以下の2種類をとってメタデータとして定義する。

- 基本メタデータ
- アプリケーションメタデータ

基本メタデータとは、どのようなデータに対しても共通なデータ属性を記述したメタデータである。基本メタデータは、Dublin Core[8]の要素に基づく。

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徴的なデータ属性を記述したメタデータである。アプリケーションメタデータは、ユーザが自由に定義することが可能である。

ユーザは、データの目的、データが出力された日付、データを出力したアプリケーションといった様々なデータ属性を知っている。ユーザが既知のデータ属性を用いて目的のデータへアクセスすることを、本論文では、データのセマンティクスを用いたデータアクセスと呼ぶ。

メタデータを用いた理由は、データ属性を的確に記述できるためである。Dublin Coreなどの標準化されたメタデータを用いることで、記述性を高めることが可能である。

また、複数組織間でのデータ管理となるため、異なる資源に対して共通の記述をすることができる。分散するデータと属性情報を含むメタデータを別々に管理することで、相互運用性の向上が見込める。

4.2 システムアーキテクチャ

我々が考える広域分散ネットワーク環境におけるデータ管理・アクセスシステムのアーキテクチャを図2に示す。

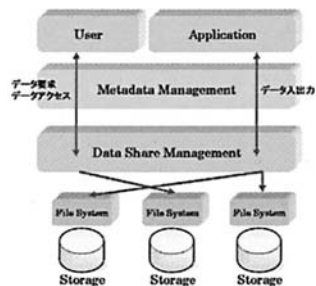


図2: システムアーキテクチャ

ファイルシステム層は、OSが提供するファイルシステムやNFSなどのミドルウェアが提供するファイルシステムなど実際のストレージ上で動作する機能を提供する。

データ共有層は、広域に分散するファイルシステムを統合し、巨大なファイルシステムを構築するための機能を提供する。

メタデータ管理層は、データインデックス機能、メタデータ管理機能、広域分散ネットワーク上での名前空間を提供する。

ユーザ、アプリケーション層は、ユーザやアプリケーションがデータへアクセスするための機能を提供する。

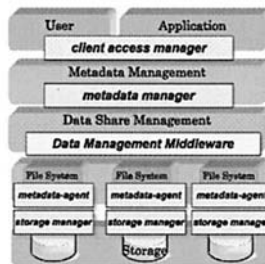


図 4: システム設計

4.3 仮想名前空間

広域ネットワーク環境上でのデータ管理システムでは、グローバルな名前空間を提供することが必要となる。本研究では、データに対するグローバルなPATHを提供するとともに、ユーザが使いやすい形でデータの分類を行うために、ユーザごとの仮想名前空間を構築する。ユーザごとの名前空間を”Personal View”と呼ぶ(図3)。Personal View上のディレクトリに、あるデータ属性と属性値を記述すると、その属性値に該当するデータを表示する機能を提供する。さらに、データ属性と属性値を記述したディレクトリの子ディレクトリにさらに属性値を記述することで、親ディレクトリの属性値を継承した形でデータの表示を行う。この機能を提供できることで、ユーザはそれぞれのディレクトリにデータ属性と属性値を記述するだけで、これまで煩雑であったデータの分類を行うことが可能になる。

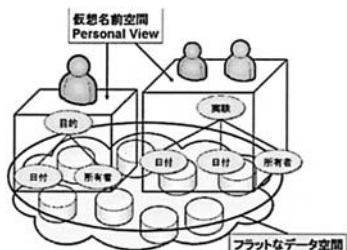


図 3: 仮想名前空間 (Personal View) とフラットなデータ空間

5 システム設計, プロトタイプ実装

5.1 システム設計

提案システムの設計を図4に示す。

ユーザ・アプリケーション層で動作する”クライアントアクセスマネージャ”, メタデータ管理層で動作する”メタデータマネージャ”, ファイルシステム層およびストレージ層で動作する”メタデータエージェント”, ”ストレージマネージャ”から構成

される。データ共有層は、既存のデータグリッドシステムを利用することを想定する。

- クライアントアクセスマネージャユーザがメタデータを記述し、データアクセスする。
- メタデータマネージャストレージ側から送られてきたメタデータを管理する。ユーザからの要求に応じてデータの位置などを提供する。
- メタデータエージェントデータが出力された際に、メタデータを取得しメタデータマネージャにメタデータを送信する。
- ストレージマネージャデータの移動を検知を行う。また、ユーザが1次データを基に新たにデータを出力した際に、メタデータエージェントを呼び出す。

5.2 プロトタイプ実装

本提案を、システムとして実現するためプロトタイプ実装を行った。本実装では、C言語を用いて行った。各サーバ間の通信プロトコルには、XML-RPC[9][10]を用いた。メタデータを管理するためのデータベースとして、PostgreSQLを用いた。以降、プロトタイプシステムの詳細について述べる。

5.3 メタデータ管理システム

本節では、メタデータ管理システムの実装について述べる。

5.3.1 メタデータエージェント

メタデータエージェントとは、アプリケーションがデータを出力する先のストレージシステムで動作し、データが発生した際に自動的にメタデータを取得するプログラムである。図5にメタデータエージェントのモジュール構成を示す。メタデータエージェントは、*open_hook*, *get-metadata*, *send-metadata*から構成される。

`open_hook` は、データが出力される際に、UNIX の `open` システムコールをフックする。新規作成 (`O_CREAT`) を検知すると、`get-metadata` モジュールを呼び出す。UNIX の `open` システムコールをフックするため、あらかじめアプリケーションを利用する前に、`open_hook` を preload する。`get-metadata` では、該当するファイルから UNIX システムコールなどを用いて基本メタデータを取得する。`send-metadata` では、インデックスサーバへ XML-RPC プロトコルを用いて送信する。

metadata-agent

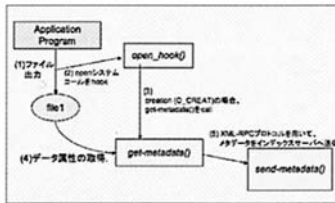


図 5: metadata-agent プログラム

5.3.2 メタデータ管理サーバ

メタデータ管理サーバ (Metadata Management Server, MMS) プログラムは、メタデータエージェントが取得したメタデータを管理し、ユーザからの要求をもとにして、データを発見する機能を提供する。図 6 に、MMS のモジュール構成を示す。MMS プログラムは、`mms-daemon` モジュール、`mms-access` モジュール、`metadata2sql` モジュール、`db-access` モジュールおよび、Metadata-DB から構成される。メタデータ管理サーバプログラムが動作しているサーバを、インデックスサーバと呼ぶ。`mms-daemon` モジュールは、メタデータエージェント側および、ユーザのアクセスクライアント側からのリクエストを受け付けるプログラムである。`metadata2sql` モジュールは、`mms-daemon` モジュールで受け取ったメタデータを RDBMS である PostgreSQL に格納するため、もしくは、検索するために SQL 形式に変換するためのプログラムである。`db-access` モジュールは、`metadata2sql` モジュールで生成された SQL を Metadata-DB にクエリとして発行するプログラムである。`db-access` モジュールは、C 言語用の PostgreSQL インタフェースである、`libpq` を用いて実装した。

5.3.3 基本メタデータ

基本メタデータは、第 4.1 節で説明したが、Dublin Core の要素に基づいている。表 1 に、プロトタイプ実装で用いている基本メタデータの要素を示す。

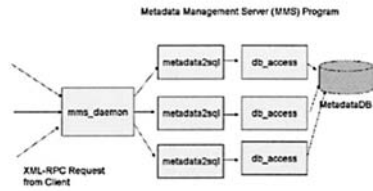


図 6: Metadata Management Server (MMS) プログラム

表 1: 基本メタデータ

属性名	意味
dataid	提案システム内でのデータ識別子
appid	提案システム内でのアプリケーション識別子
dc:title	データのタイトル
dc:creator	データの所有者
dc:date	データの作成日時, 編集日時, アクセス日時
dc:identifier	データの識別子 (URI など)
dc:language	データ記述言語

5.3.4 アプリケーションメタデータ

グリッドコンピューティングでは、アプリケーションの実行は統一されたインタフェース (ウェブインタフェースなど) から実行される。統一されたインタフェースからアプリケーションを実行すると、ジョブディスパッチャから VO 上の計算機資源を有効活用される。ユーザがアプリケーションを実行する際に、ウェブインタフェース上でアプリケーションメタデータを記述する。

6 評価

本節では、システムに求められる機能要件について関連研究と比較する (表 2)。

(1) シームレスなデータアクセスのための透過性: 提案システムでは、ユーザがデータアクセスする際に、データ属性とデータ属性値を基にしてデータのリストを取得する。データリストの中には、位置情報、プロトコル情報を保有している。ユーザが該当するデータを選ぶと、提案システム側で位置情報などを用いてデータを取得する。ユーザがデータを取得する際には、位置やプロトコルを意識しなくてよい。

(2) データ位置情報管理: Gfarm では、データを分割して保存する。分割したデータの位置情報管理は行っているが、データが移動したなどのことには対応が難しい。

(4) データ属性記述: Gfarm では、分割したデータの位置情報などの属性情報を取り扱っているが、データ分類のためには、データ属性を記述できない。

表 2: 機能要件の比較

機能要件	SRB	Gfarm	提案システム
(1)	○	○	○
(2)	△	△	○
(3)	○	○	○
(4)	○	×	○
(5)	×	△	○
(6)	○	×	○

- (1) シームレスデータアクセスのための透過性
- (2) データ位置情報管理
- (3) アプリケーション間連携のためのデータ操作コマンド, API の提供
- (4) データ分類・アクセスのためのデータ属性記述
- (5) データ属性の自動取得
- (6) データ属性を用いたデータアクセス

(5) データ属性の自動取得: SRB では, メタデータの記述, 操作はユーザが行う。大量に発生するデータに対して, メタデータをユーザが記述することは膨大な作業量となる。提案システムでは, 基本メタデータの自動取得を行う。アプリケーションメタデータに関しては, ユーザが1回のアプリケーションの実行に対して行う。アプリケーションの1回の実行で, 大量にデータが発生する場合非常に有効である。

7 今後の課題

7.1 アプリケーションメタデータ定義インタフェース

プロトタイプシステムでは, 基本メタデータのみに扱っている。プロトタイプでは, ひとつのアプリケーションメタデータをとりあげて実装を行っている。どのようなアプリケーションにも対応するためには, ユーザがアプリケーションごとに自由にアプリケーションメタデータを定義できる仕組みが必要となる。ユーザがアプリケーションメタデータを自由に記述できるためのインタフェースを現在, 実装している。

7.2 メタデータ管理機能の分散化

現在のプロトタイプの実装方法では, メタデータ管理機能がボトルネックになる。このボトルネックを解消するために, メタデータ管理機能の分散化を図る。このメタデータ管理機能を分散化するための技術として, DHT や, 分散データベースなどの技術がある。システムを構築する上で, どの技術がよいか十分に検証を行った上で実装を行っていく予定である。

7.3 広域ネットワーク上での実験

現在, プロトタイプシステムの実装を行っている。システムを運用するためには, 実ネットワークを用いた実験が必要である。今後は, 本提案のプロトタイプシステムをオーバーレイネットワークテストベッドである PlanetLab[11] 上での実験を行う予定である。

8 まとめ

本論文では, 広域分散ネットワーク環境における新たなデータ管理およびデータアクセス手法について提案した。提案する手法では, メタデータを用いてデータセマンティクスを記述する。メタデータは, どのようなデータにも共通であるデータ属性と, アプリケーション固有のデータ属性から構成されている。提案手法に基づき, システム設計・プロトタイプ実装を行った。

参考文献

- [1] I. Foster, C. Kesselman. *The GRID2 Blueprint for a New Computing Infrastructure*. Morgan Kaufman, 2003.
- [2] Google, Inc. <http://www.google.com/>.
- [3] 佐藤麻美, 渡辺知恵美. P2P を利用した地球物理データのネットワーク横断検索・共有システムの実現に向けて. 電子情報通信学会 第 18 回データ工学ワークショップ, 2007.
- [4] 柳平有美, 渡辺知恵美, 堀之内武. 地球流体物理科学者のためのデータアーカイブサーバ構築支援ツール: Gfdnavi におけるデータベース設計と検索インタフェースの実装. 電子情報通信学会 第 18 回データ工学ワークショップ, 2007.
- [5] 松田哲史. 複数属性を持つデータの属性値範囲指定検索用分散ネットワーク構成方法の提案. 情報処理学会 研究報告 第 44 回 分散システム / インターネット運用技術, 2007.
- [6] Storage Resource Broker. <http://www.sdsc.edu/srb/>.
- [7] Grid Datafarm. <http://datafarm.apgrid.org/>.
- [8] Dublin Core. <http://dublincore.org/>.
- [9] XML-RPC. <http://www.xmlrpc.com/>.
- [10] XML-RPC for c and c++. <http://xmlrpc-c.sourceforge.net/>.
- [11] PlanetLab. <http://www.planet-lab.org/>.