

4-2 先進的情報通信技術の活用

4-2 Applications of Advanced Information and Communications Technology

4-2-1 グリッドデータファームによる並列分散処理

4-2-1 Distributed Parallel Processing Based on Grid Datafarm Architecture

山本和憲 木村映善 村田健史 建部修見 松岡大祐 宮地英生
YAMAMOTO Kazunori, KIMURA Eizen, MURATA Ken T., TATEBE Osamu,
MATSUOKA Daisuke, and MIYACHI Hideo

要旨

太陽地球系物理学分野では、近年、衛星観測データの増大化及び計算機シミュレーションデータの大規模化が進み、効率的な大規模データ処理手法の開発が期待されている。本研究では、メタデータシステム(STARS)とグリッドデータファーム(Gfarm)による並列分散処理システムを提案し、8台のファイルシステムノードによる実装を行った。更に、構築したシステム上で、衛星観測データと計算機シミュレーションデータの並列分散処理を実行し、性能評価を行った。長期間観測データ処理では、約50KBのファイルサイズが小さい処理においてもメタデータのローカルディスクへの配置やGfarmファイルの階層化を行うことで、並列分散処理が有効であるという結果を示した。シミュレーションデータの並列3次元可視化では、FIFO型スケジューリングを適用し負荷分散を最適化することで、8並列処理時において97.6%という高い並列化効率を得られた。

In the Solar-Terrestrial Physics field, satellite observation data and computer simulation data have been tremendous increased. Since most of data files and computer resources are distributed over the Internet, analysis environments for data intensive processing are required. In this study, we propose a parallel distributed processing system with meta-data system and Grid Datafarm. A testing system is constructed with 8 filesystem nodes. As a result of small-data processing of observation data on the system, parallel processing is found effective using meta-data file at local disk and hierarchical Gfarm file. As for parallel visualizations of simulation data, it was achieved high parallelization efficiency of 97.6% when using FIFO-type scheduling.

[キーワード]

メタデータベース, グリッドデータファーム, 並列分散処理, 衛星観測データ, 計算機シミュレーションデータ

Meta-database, Grid datafarm, Distributed parallel processing, Satellite observation data, Computer simulation data

1 まえがき

太陽地球系物理学分野は、宇宙天気をはじめとする太陽活動が地球磁気圏・電離層や大気等に影響をおよぼす物理過程の解明を目的とする横断的研究分野である。観測データは、衛星機器の性能向上により高精度化・大規模化し、国際デジタル地球年 (eGY: electronic Geophysical Year)^[1] や太陽地球系物理学国際共同観測 (ISTP: International Solar-Terrestrial Physics) 計画^[2] などの国際プロジェクトにより、その蓄積量は増加の一方である。また、計算機シミュレーションは、地球シミュレータをはじめとするスーパーコンピュータの性能向上により計算モデルの高精度化や大規模化が進みつつある。今後は、観測データ解析では複数衛星による多地点長期観測データの統計解析処理^[3] など、計算機シミュレーションでは長時間ステップの3次元可視化処理^[4] などの大規模データ処理が期待されている。

太陽地球系物理学分野の大規模データ処理では、大量のデータファイルに同一処理を施すデータインテンシブ処理を行う場合が多い。そのため、データインテンシブ処理に有効なデータファイルの管理・共有方法やファイル I/O の負荷分散が可能な並列分散処理手法が必要とされている。

本研究では、グリッドデータファームアーキテクチャの参照実装である Gfarm(Grid Datafarm)^[5] を用いて、太陽地球系物理学分野の衛星観測データと計算機シミュレーションデータのデータインテンシブ処理を行う並列分散処理システムを構築し、その有効性・実用性の評価を行う。Gfarm はストレージと計算処理を兼用した複数ノードにより、ネットワーク間のファイル転送の制御とファイル I/O を分散するデータグリッドのためのミドルウェアである^[6]。これまでに天文データ解析^[7] や GEO Grid プロジェクト^[8] などで利活用されており、ファイル I/O のスケーラビリティや CPU 性能の負荷分散の有効性が検証されている。しかし、データファイルが分散管理されている太陽地球系物理学分野において、データインテンシブ処理環境を構築することは容易ではなく、データグリッドによる並列分散処理の有効性は検証されていない。また、データサイズやデータ処理量が解析対象データや解析期間により異なり、単純な

データファイル分割による負荷分散が有効であるとは限らないため、Gfarm が想定していないスケジューリング手法が必要となる。

2 太陽地球系物理学分野の統合的データ処理環境の現状・問題点とシステム提案

2.1 データ処理環境の現状と問題点

1 で述べたように太陽地球系物理学分野は横断的研究分野であり、観測データや計算機シミュレーションデータは国内外の大学や研究機関に分散して管理・公開されている。データ解析者は各自のデータ処理用計算機に必要な全てのデータファイルをダウンロードするため、データの種別とファイル数が多くなるとダウンロード処理と解析・可視化処理が煩雑になる。

また、データ解析や可視化を解析者の個人端末上で行う場合、データサイズやデータファイル数が膨大となる処理は、CPU 性能やディスク I/O の制約によりデータ処理の規模が制限される。このため、多くのデータセンターでは大規模データ処理のオンラインサービスを提供している。例えば、アメリカ航空宇宙局 (NASA) の SSCWeb^[9] や宇宙航空研究開発機構 (JAXA) の DARTS^[10] では、所有している衛星観測データの検索や簡易プロットサービスを提供している。また、情報通信研究機構 (NICT) のリアルタイム地球磁気圏シミュレーション^[11] では、シミュレーションデータの3次元可視化動画を公開している。これらの環境では、解析者はデータファイルをダウンロードせずに処理結果を取得することが可能である。しかし、処理内容はデータサイト側で定められるため、制約された条件下での単純な検索やデータプロットなどの原始的なデータ処理に留まっている。また、多くのデータセンターでは、他のデータセンターのデータも含めた統合的解析環境を提供するサービスを有しない。

2.2 本研究で提案する並列分散処理システムの概要

2.1 で述べた問題点を解決するため、本研究では太陽地球系観測データのメタデータ活用システム (STARS: Solar-Terrestrial data Analysis and

Reference System) [12][13] とグリッドデータファームアーキテクチャの参照実装である Gfarm を用いたデータ処理システムを提案する。STARS は分散管理された衛星観測データファイル及び計算機シミュレーションデータファイルに対して、アクセス透過性や位置透過性のあるデータ利用環境を提供する。本システムでは、データ検索・取得までを STARS を用いて行い、データ取得後の大規模並列分散処理を Gfarm 上で行う。システムの構築には Gfarm version 1.4.1 を用いた。

8 台のファイルシステムノードで構成した本研究のシステムを図 1 に、計算機スペックを表 1 に示す。ユーザはまず、STARS 経由でデータサイトからデータファイルを取得し(図 1- (1))、Gfarm ファイルシステムに登録する(図 1- (2))。4.4 で後述するが、処理内容によっては効率的な負荷分散のために、登録と同時に各ファイルシステムノードにファイルを複製することもある。続いて、並列分散処理のジョブを投入すると(図 1- (3))、クライアントマシンから実行プログラムが各ファイルシステムノードにコピーされ(図 1- (4))、並列分散処理が行われる(図 1- (5))。最後に、処理

結果の表示の整合性が取られ、ユーザ端末に結果が返される(図 1- (6))。

3 長期間衛星観測データの並列分散処理

3.1 逐次処理と並列分散処理の比較

本節では、図 1 のシステムを用いて逐次処理と Gfarm による並列分散処理の比較実験を行う。逐次処理はファイルシステムノードを 1 台使用し、Gfarm によるオーバーヘッドは生じないものとする。並列分散処理は並列数 2~8 で行う。

実験ではデータインテンシブ処理を行い、1

表 1 システム構築に用いた計算機スペック

	Filesystem Node × 8	Metadata Server × 1 Client Machine × 1
CPU	Athlon 64 × 2 Dual Core 2GHz	Dual Core AMD Opteron 1.81GHz
Memory	2GB	1GB
Disk	1.2TB(using RAID0)	232GB
OS	Fedora Core 5	Fedora Core 5

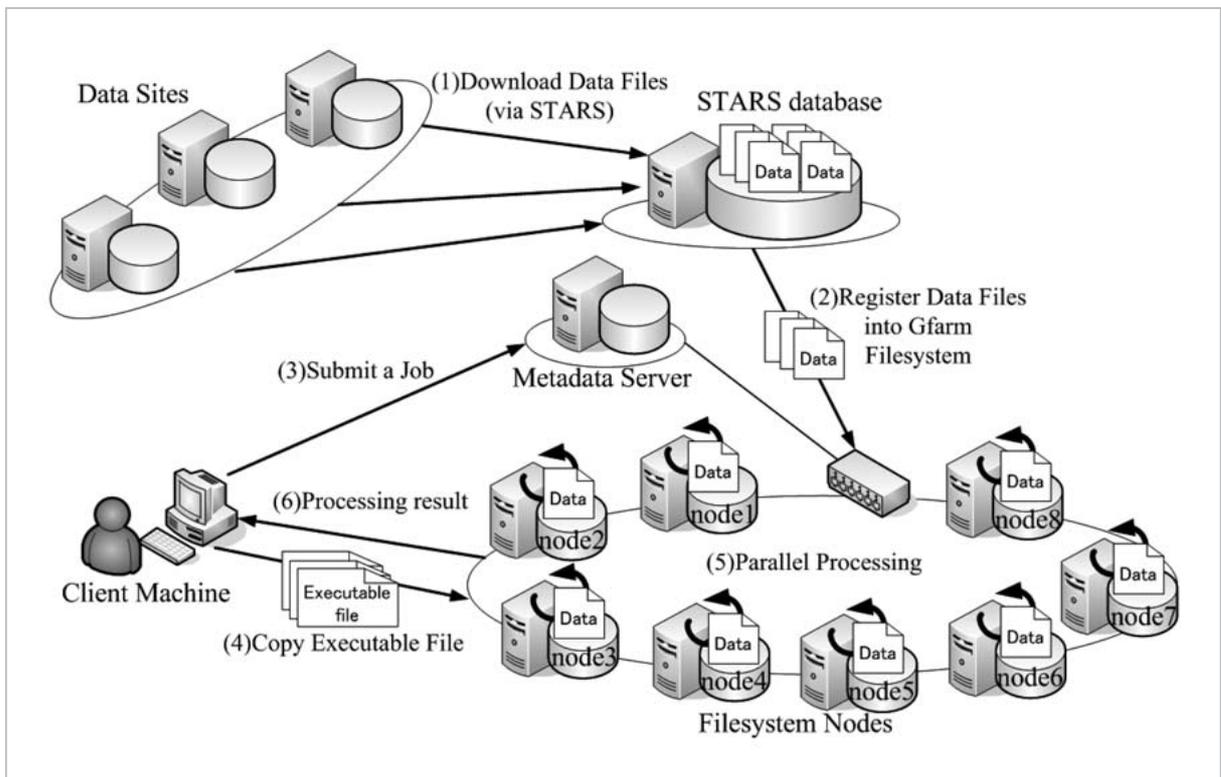


図 1 STARS と Gfarm による並列分散処理システム

ファイルあたり 1 プロセスでデータを処理する。データ処理は表 2 に示すファイルサイズの異なる 2 種類の衛星観測データファイルから全時系列データをシーケンシャルに読み込み、全データレコードを標準出力する。ジョブ割り当ては、データファイル数を M、並列数を N とした場合、逐次処理では 1 台で M 個のデータファイルを処理し(図 2(a))、並列分散処理では各ファイルシステムノードが割り振られた M/N 個のデータファイルを処理する(図 2(b))。なお、実験の前処理として全てのデータファイルを Gfarm ファイルシステムに登録した。

表2 使用した衛星観測データ

Mission / Data	File Size[KB/file]
(a)GEOTAIL/LEP(Key parameter)	450 - 995
(b)GEOTAIL/Orbit(Level 2)	44 - 58

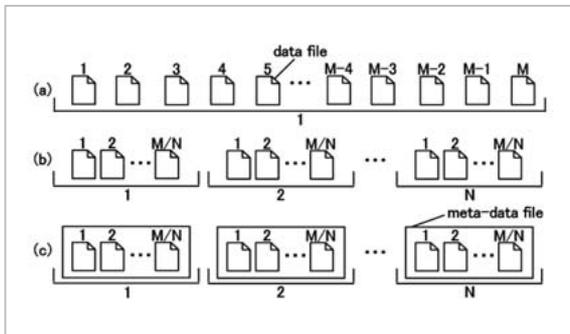


図2 衛星観測データの処理方法

(a)逐次処理、(b)並列分散処理、(c)Gfarmファイルを階層化した並列分散処理。

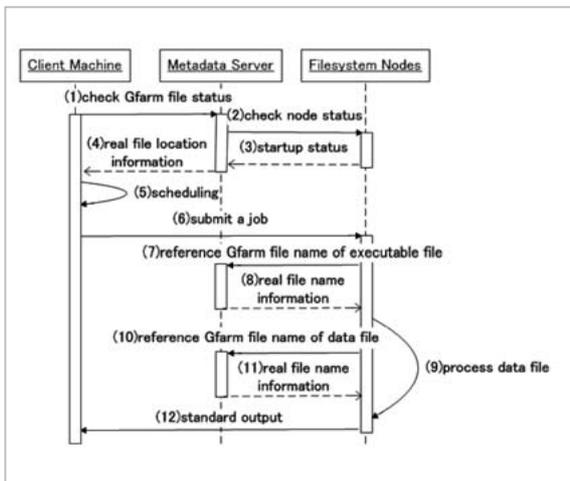


図3 Gfarm による並列分散処理の流れ

並列分散処理時のクライアントマシン、メタデータサーバ、ファイルシステムノード間の処理手順を図 3 に示す。処理の主な内訳は、図 3 - (1) ~ (8) に示すスケジューリング(ファイルシステムノードの選択とジョブの投入)、図 3 - (9) ~ (11) に示すファイル読み込み(Gfarm ファイルの実体ファイル参照のためのメタデータベースアクセスとデータ処理)、図 3 - (12) に示す標準出力(クライアントマシンへのアクセス)である。

3.2 結果

表 2 に示した 2 種類のデータの処理結果(データファイル数と実行時間の関係)を図 4 と図 5(a) に示す。また、計算機 1 台の処理時間を T_1 、計算機 n 台の処理時間を T_n とした場合の、図中の 1,000 ファイル処理時の並列化効率 $\eta(n) = (T_1/T_n) / n$ を表 3 に示す。ファイルサイズが大きい GEOTAIL/LEP データでは、図 4 に示すように並列分散処理の方が逐次処理に比べて処理時間が短く、8 並列処理時の並列化効率 $\eta(8)$ は約 43.4%であった。

一方、ファイルサイズが小さい GEOTAIL/Orbit データでは図 5(a) に示すように、並列数にかかわらず逐次処理の方が並列分散処理に比べて処理時間が短く、表 3(b) に示すように低い並列化効率となった。これは、並列分散処理のオーバーヘッドが並列化による短縮時間を上回ったためである。

オーバーヘッドの要因としては、3.1 で述べた

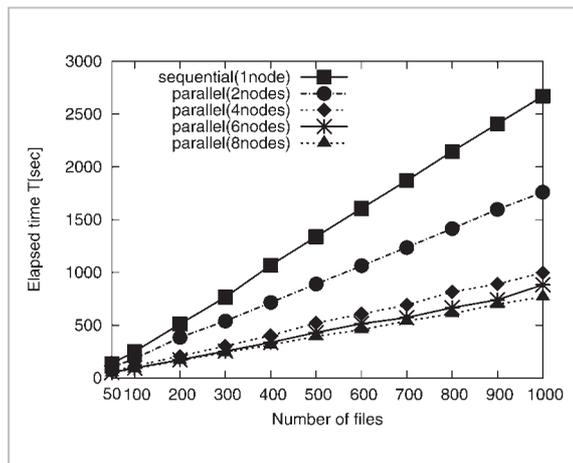


図4 GEOTAIL/LEP データの逐次処理と並列分散処理の処理時間の比較

(■)は逐次処理、●は2並列、◆は4並列、*は6並列、▲は8並列である。

表3 1,000 ファイルにおける並列化効率 η [%]

Number of nodes (n)	2	4	6	8
(a)GEOTAIL/LEP	75.90	66.85	50.43	43.39
(b)GEOTAIL/Orbit	4.05	2.30	1.53	1.15

表4 GEOTAIL/Orbit データの1ファイルの処理時間

(A)スケジューリング、(B)ファイル読み込み、(C)出力処理。

Process	Elapsed Time[sec]		Time Lag Average [sec]
	Sequential (Variance)	Parallel (Variance)	
(A)Scheduling Phase (Fig. 3-(1)~(8))	0.0122 (0.0000138)	5.3317 (0.0608658)	-5.3195
(B)Reading File Phase (Fig. 3-(9)~(11))	0.0615 (0.0000619)	0.9784 (0.0031524)	-0.9169
(C)Output Phase (Fig. 3-(12))	0.0002 (0.0000002)	0.0772 (0.0001900)	-0.0770
Total Time[sec]	0.0739 (0.0000861)	6.3873 (0.0724450)	-6.3134

(A)スケジューリング時間(図3-(1)~(8))、(B)ファイル読み込みにおけるメタデータベースアクセス時間(図3-(9)~(11))、及び(C)出力処理時間(図3-(12))が挙げられる。逐次処理と並列分散処理における GEOTAIL/Orbit データの1ファイル処理の(A)~(C)の各実行時間を表4に示す。表中のタイムラグが、逐次処理では生じないオーバーヘッドを表している。

Gfarm による並列分散処理では各プロセスが独立して処理されるが、表3では並列数が増えるほど並列化効率が低くなっている。これは並列数の増加に伴い、ジョブ投入時や Gfarm ファイルの実体ファイル参照時にメタデータサーバへのアクセスが集中し、サーバのレスポンスが低下したためである。

3.3 メタデータのキャッシングによるファイル名参照時間の改善

本節では、表4で2番目に影響の大きい(B)ファイル読み込みにおけるメタデータベースアクセス時間(図3-(9)~(11))の短縮を試みる。表4(B)のファイルの読み込み時間のオーバーヘッドは、ファイルシステムノードのローカルディスクに Gfarm ファイルの実体ファイルがあるにもかかわらずメタデータサーバとの通信が発生するこ

とにより生じる。ローカルディスクにファイルがあるときには、内部キャッシュで名前解決することによりパフォーマンス改善が期待できる。具体的には、各ファイルシステムノードで処理するファイルリストを事前にローカルディスク上のテキスト形式のメタデータファイルとして配置し、これを各ファイルシステムノード上で読み込む。この手法により GEOTAIL/Orbit データに 3.1 と同じ実験を行った結果を図5(b)に示す。図5(b)の sequential(1 node)と parallel(8 nodes)は図5(a)の逐次処理と8並列処理の結果を再掲したものである。図5(b)を見ると、改良した手法ではファイル読み込みにおけるメタデータベースアクセス時間が短縮されているが、parallel(8 nodes)からの改善の度合いは小さい。

3.4 Gfarm ファイルの階層化によるスケジューリング時間の改善

本節では、表4で特に影響の大きい(A)スケジューリング時間(図3-(1)~(8))の短縮を試みる。Gfarm では、各ファイルシステムノードでのスケジューリングの際に立ち上がるプロセス数は、Gfarm ファイルを構成するセグメントファイル数に一致する。したがって、図6に示すようにセグメントファイルを階層化し、複数ファイルの処理プロセスをまとめることでデータファイル数分のプロセスを起動する必要がなくなり、表4(A)のスケジューリングに要する時間の短縮が期待できる。Gfarm ファイルのセグメントファイルに Gfarm ファイルを持つ機能は Gfarm v2 で実装される計画であるが[14]、本稿執筆現在では Gfarm にセグメントファイルを階層化する機能が実装されていないため、1つのメタデータファイルに複数の観測データファイルをまとめ、1プロセスで複数ファイルを処理することで実現した(図2(c))。

この手法により、GEOTAIL/Orbit データに 3.1 と同じ実験を行った結果を図5(c)に示す。図中の sequential(1 node)は図5(a)の逐次処理結果を再掲したものである。改良した手法ではクライアントマシンからのジョブの投入が行われるため、sequential(1 node)に比べて切片(立ち上がりのオーバーヘッド)が大きい、負荷が分散されるためグラフの傾きは小さくなり、ファイル数の増加に伴い逐次処理よりも効率的となる。

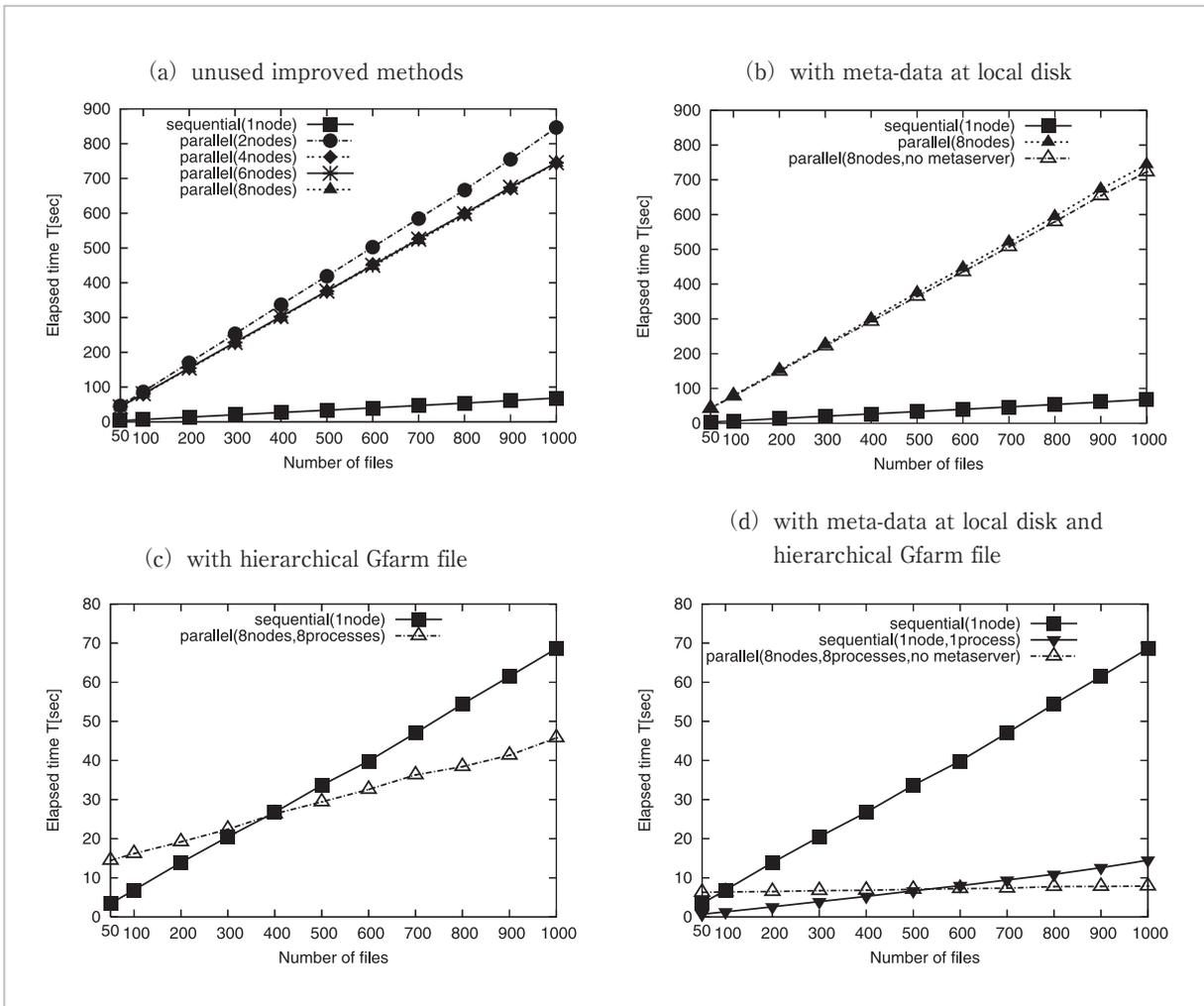


図5 GEOTAIL/Orbit データの逐次処理と並列分散処理の処理時間の比較

(■は逐次処理、●は2並列、◆は4並列、*は6並列、▲は8並列、△は8並列 (改良手法を適用)、▼は逐次処理(1プロセスで複数ファイル进行处理)である) : (a)改良手法を用いない場合、(b)メタデータのローカルキャッシュを使用した場合、(c)セグメントファイルを階層化した場合、(d)メタデータのローカルキャッシュと階層化したセグメントファイルを使用した場合。

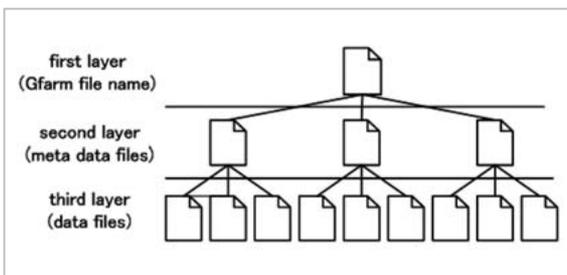


図6 階層化した Gfarm ファイルの構造

3.5 メタデータのキャッシングと Gfarm ファイルの階層化を併用した改善

図 5 (b) と (c) の両方の改善手法を適用した結果を図 5 (d) の parallel (8 nodes, 8 processes, no metaserver) に示す。図中の sequential (1node) は

図 5 (a) の逐次処理結果を再掲したものであり、sequential (1 node, 1 process) は parallel (8 nodes, 8 processes, no metaserver) と比較条件が同等になるように sequential (1 node) を 1 プロセスで処理したものである。

改良した両手法を適用した結果は、図 5 (c) と同様にクライアントマシンからのジョブの投入が行われるため、sequential (1 node, 1 process) に比べて切片 (立ち上がりのオーバーヘッド) が大きいですが、負荷が分散されるためファイル数の増加に伴い sequential (1 node, 1 process) よりも効率的となる。

図 7 (a) は図 5 (d) と同様の実験を、5,000 ~ 50,000 ファイルに対して行った結果である。図 7

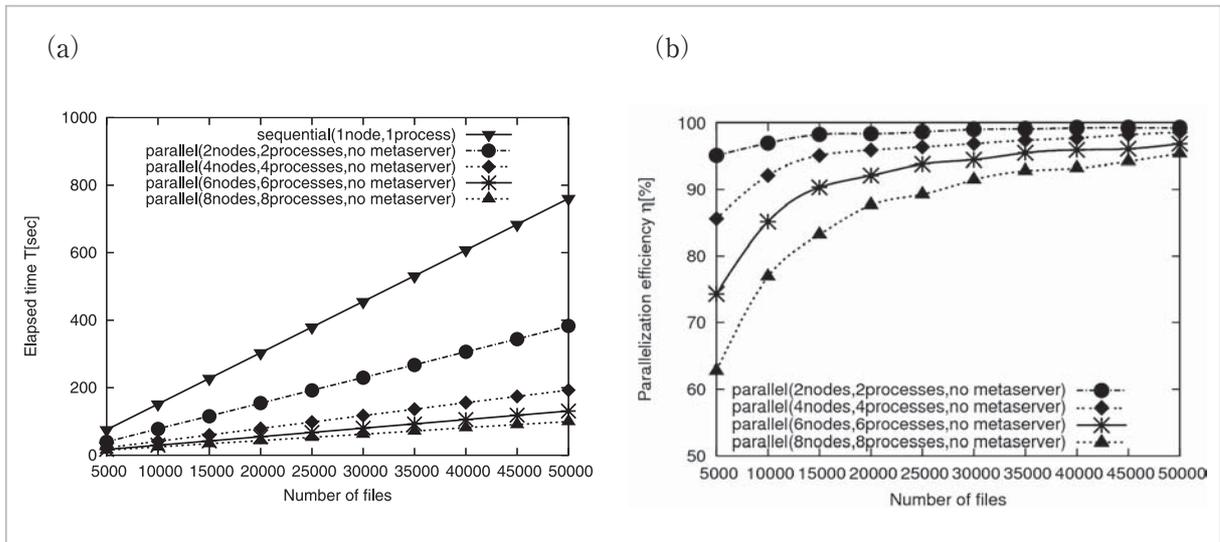


図7 GEOTAIL/Orbitデータの逐次処理と並列分散処理(メタデータのローカルキャッシュと階層化したセグメントファイルを使用)の比較

(▼は逐次処理(1プロセスで複数ファイルを処理)、●は2並列、◆は4並列、*は6並列、▲は8並列である) : (a)ファイル数と実行時間の関係、(b)並列化効率。

(a)の sequential (1 node, 1 process) に対する並列数ごとの並列化効率を図 7(b) に示す。図 7(b) より、並列数に関係なくファイル数の増加に伴い並列化効率が増加していることが分かる。これはデータ処理時間の増加に伴い、図 5(d) に示すように切片(立ち上がりのオーバーヘッド)の割合が小さくなるためである。50,000 ファイルの処理においては各並列数で 95% を超える高い並列化効率を得られていることが分かる。

3.6 考察

Gfarm の並列分散処理はデータインテンシブ処理で各プロセスが独立しており、プロセス間通信が行われなため、並列数に依存しない高い並列化効率が期待される。しかし、並列分散処理時に 1 台のメタデータサーバを共有するため、図 5(a) に示すようにファイルサイズが小さい処理においてはオーバーヘッドの占める割合が大きくなり逐次処理の方が効率的となる。並列分散処理の主なオーバーヘッドであるスケジューリング時間とファイル読み込みにおけるメタデータベースアクセス時間を短縮した場合においても、ジョブ投入時のオーバーヘッドがあるため、図 5(d) に示すようにファイル数が一定数よりも小さいときには 1 プロセスで複数ファイルを逐次処理の方が効率的となる。これは、Gfarm がライトアットワンスなデータサイズが大きいデータを対象としてい

る理由の 1 つである。

衛星ごとのデータを 1 日 1 ファイルとした場合、図 7 に示すデータファイル数 50,000 は 14 衛星の 10 年間のデータ量となる。本実験は、メタデータ取得のパフォーマンスの改善により、Gfarm の並列分散処理が太陽地球系物理学分野の多地点長期衛星観測データ処理に適用可能であることを示唆している。なお、Gfarm v1 を用いた本研究結果をもとに、Gfarm v2 ではデータベース処理をオンメモリで行ったり、メタデータ参照回数を最小化するなどの改良が行われている [15]。

4 計算機シミュレーションデータの並列3次元可視化処理

4.1 大規模並列可視化の現状

大規模並列可視化では、空間領域の分割や可視化手法ごとに分散処理する手法などが提案されている [16] - [18]。これらの手法は、1 ノードの計算機性能以上の格子サイズの可視化に対しては有効である。しかし、1 タイムステップの可視化を各ノードが協調して行う必要があり、計算機構成と可視化領域によってはノードごとの可視化処理時間にばらつきが生じるため、タイムステップが多い長期間可視化などのデータインテンシブ処理には不向きである。本節では、図 1 のシステムを STARS が管理する情報通信研究機構のリアルタ

イム地球磁気圏シミュレーション(以下、リアルタイムシミュレーション)[11]データの3次元可視化に応用し、並列可視化の有効性を調べる。

4.2 時系列方向の並列可視化

リアルタイムシミュレーションでは、タイムステップ毎に1つの数値データファイルが出力される。本実験では、150 タイムステップのシミュレーションデータを図1のシステムで並列可視化する。可視化に用いたシミュレーションデータは、リアルタイムシミュレーションにより計算された磁場3成分データで、1タイムステップ(1ファイル)のサイズは約80MB(合計12GB)である。また、可視化により生成される3Dオブジェクトファイルは1タイムステップあたり約1.2MB(合計180MB)である。可視化には汎用可視化アプリケーション AVS/Express 7.1.1[19]を用い、磁力線の3次元可視化を行った。

本研究の並列可視化手法を図8に示す。まず、

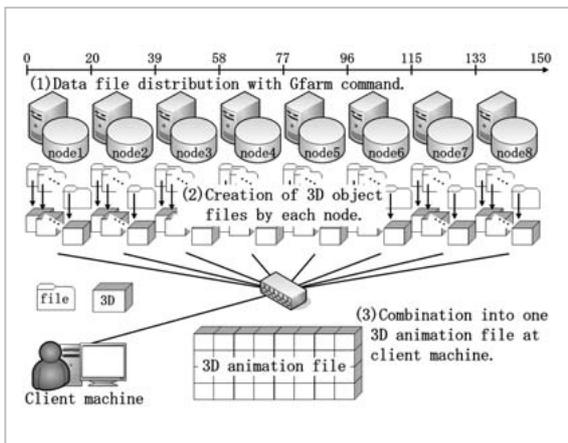


図8 可視化の並列分散処理

Gfarm コマンドによりデータファイルを各ファイルシステムノードに分配する(図8-(1))。実験では150ファイルを8分割し、各ノードに均等に分配した。次に、各ファイルシステムノードが分配されたローカルディスク上のデータファイルに対して可視化を行い、3DオブジェクトをGfarmファイルとして出力する(図8-(2))。これらの処理を各ファイルシステムノードが分配されたファイル数分繰り返す。最後に、生成された3Dオブジェクトをクライアントマシンで1つのアニメーションファイルとして結合する(図8-(3))。

4.3 結果

4.2による可視化結果を図9に示す。また、各タイムステップの可視化処理時間を図10(a)に、各ファイルシステムノードの可視化処理時間の合計を図11(a)に示す。図11(a)の最長時間は並列可視化処理全体の実行時間と等しく、10,962秒(約183分)であった。同様の可視化処理をファイルシステムノード1台で逐次処理した場合には約4.5倍の49,726秒(約829分)を要した。これより並列可視化処理の有効性は示されたが、図11(a)では最短と最長の可視化処理時間には8,160秒(約136分)の差があり、並列化効率が56.7%と低いため、改善の余地がある。

4.4 負荷分散の最適化の検討

磁力線の本数が多い場合には、図9(b)に示すように磁力線描画の可視化処理時間が長くなる。その結果、図11(a)のnode4やnode5のように処理時間が長くなり、負荷バランスが崩れる。これは、Gfarmのスケジューラが全セグメントファ

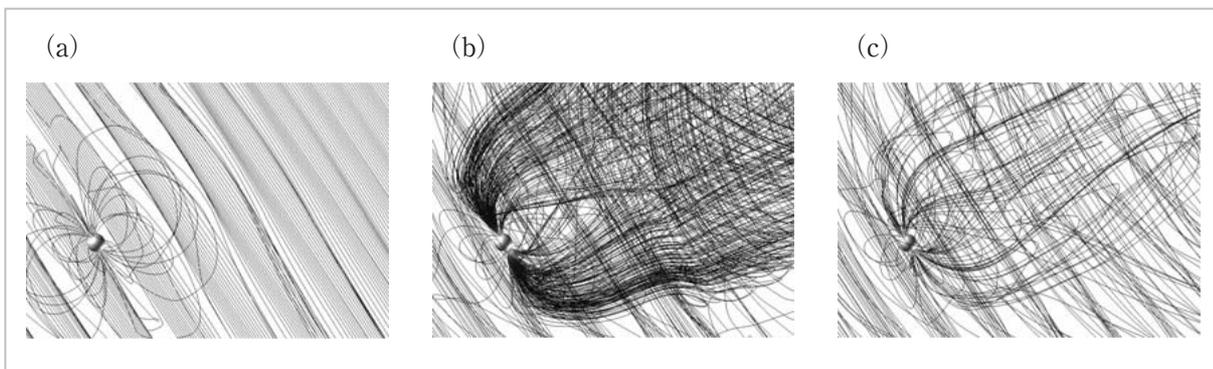


図9 地球磁気圏シミュレーションの3次元可視化

(a)1ステップ目、(b)82ステップ目、(c)100ステップ目。

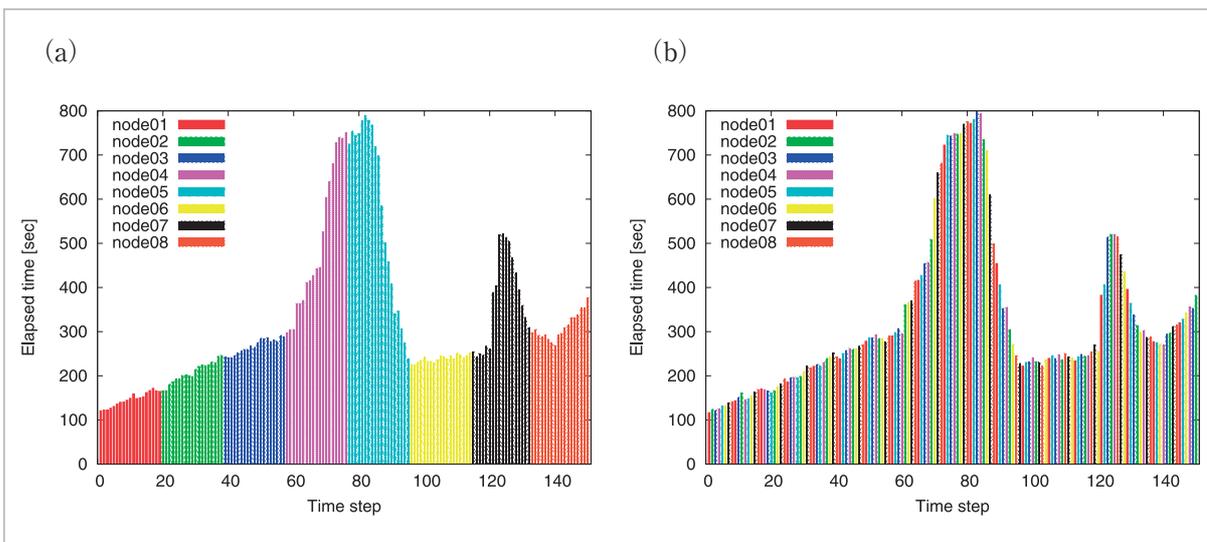


図10 タイムステップごとの実行時間

(a) ファイルは各ノードに均等に分配、(b) FIFO型スケジューリングによるファイル分配。

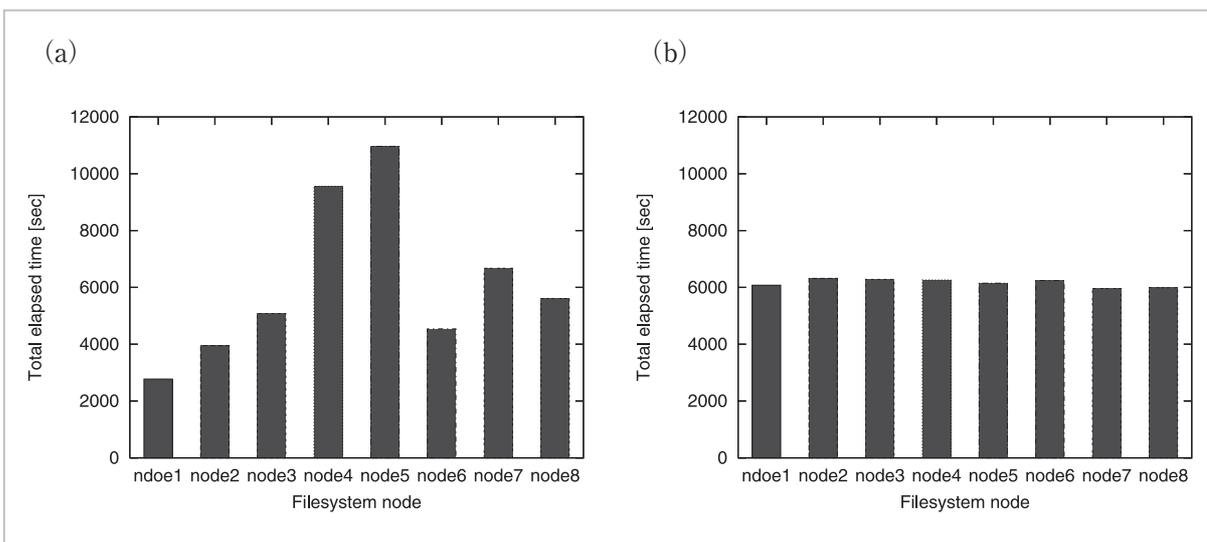


図11 各ファイルシステムノードの可視化処理時間の合計

(a) ファイルは各ノードに均等に分配、(b) FIFO型スケジューリングによるファイル分配。

イルへのジョブ割り当てを最初に一括して行うためである(図12(a))。しかし、この手法ではファイルの処理時間が均一でない場合に負荷分散が最適化されない。そこで、スケジューリングをジョブ投入時に一括して行うのではなく、ジョブをキューに格納し、ジョブ終了ノードから順に次のジョブを割り当てるFIFO型スケジューリング(図12(b))に改良した。なお、この手法では各ノードに割り当てられるファイルを予想できないため、割り当てられたファイルがローカルディスクに存在しない場合には、データ転送時間が追加される。本実験では、あらかじめ全データファイ

ルをGfarmのgfredコマンドを用いて各ノードに複製した。

改良後のタイムステップ毎の可視化処理時間を図10(b)に、各ファイルシステムノードの合計可視化処理時間を図11(b)に示す。図11(b)では最短と最長の可視化処理時間の差が360秒(6分)に短縮され、負荷分散が図11(a)よりも最適化された。並列可視化処理全体の実行時間も6,360秒(106分)に短縮され、並列化効率は97.6%に向上した。これより、各データファイルのデータ処理粒度が異なる並列分散処理においても、高い並列化効率が実現可能であることが確認できた。

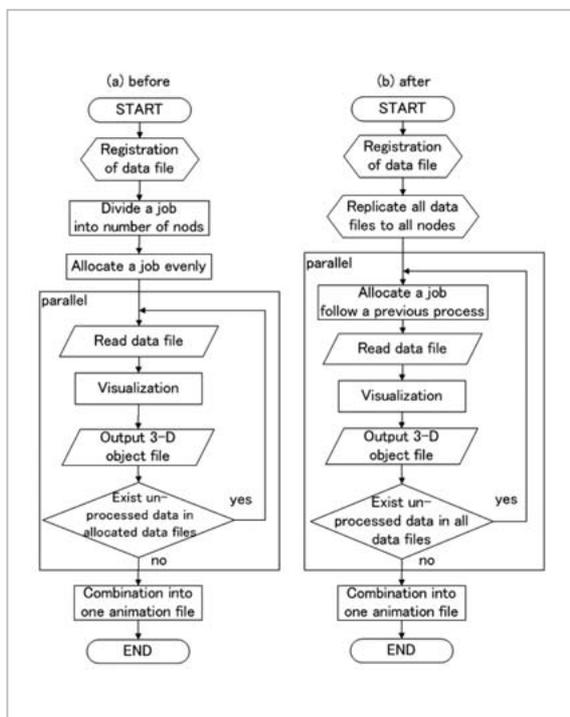


図12 可視化スケジューリングのフローチャート

(a)ファイルは各ノードに均等に分配、(b)FIFO型スケジューリングによるファイル分配。

4.5 考察

本研究の提案手法では並列数に関係なく負荷分散が最適化されるため、スケーラブルな並列分散処理が可能である。ただし、全データファイルを各ノードに複製する必要があるため、並列化効率とデータファイル複製時間がトレードオフの関係になる。本実験では、12 GB の全データファイルを全ノードに複製するのに要した時間は約 77 分であり、1 回目の可視化では両スケジューリング手法に処理時間の差が見られない。なお、2 回目以降の可視化では複製が不要なため、可視化パラメータを変えて同じデータの可視化を行う場合には提案した手法が効率的となる。今後は、可視化処理とファイルシステムノードへのデータファイル転送のパイプライン処理により、問題点を改善する必要がある。

5 むすび

地球科学の多くの分野で観測データのデジタル化が進み、計算機シミュレーションデータを含めたあらゆるデータの蓄積量が増大している。本

研究では、太陽地球系物理学分野においてデータ管理やデータ処理環境が分散化している現状を背景に、大規模データ処理を実現する手法として、STARS と Gfarm による分散データ型データインテンシブ処理システムを提案した。8 台のファイルシステムノードによりシステムを構築し、衛星観測データと計算機シミュレーションデータを用いてシステムの有効性を検証した。その結果、長期間衛星観測データ処理では、メタデータのローカルディスク配置と Gfarm ファイルの階層化を併用することで、ファイルサイズが小さなデータファイルに対しても並列分散処理が有効であることが分かった。計算機シミュレーションデータの並列 3 次元可視化では、データ処理粒度が異なる場合でも FIFO 型スケジューリングを行うことで負荷分散が最適化され、8 並列処理で 97.6 % の高い並列化効率が得られた。

太陽地球系物理学分野では、衛星観測データと計算機シミュレーションデータは相補的な関係にあり、これまでの個別データ解析環境から横断的・統合的データ解析環境の実現が求められている。筆者らの研究グループでは両データの融合表示・解析を試みてきた[20]。今後は、本研究結果を発展させ、データファイルサイズやデータ処理粒度がヘテロなデータセットを組み合わせた融合型データインテンシブ処理により、統合的な多目的データ処理環境の実現が期待される。さらに、STARS にメタデータを提供している各データサイトにファイルシステムノードを導入し、STARS のメタデータ管理を仮想組織 (VO: Virtual Organization) 間で共通な太陽地球系物理学分野の仮想ディレクトリで統一し、データインテンシブな処理環境を構築することが望まれる。

本研究にご協力して下さいました宇宙航空研究開発機構・篠原育准教授に感謝致します。本研究は文部科学省の科学研究費補助金・学術創成研究費「宇宙天気予報の基礎研究」(17GS0208、代表者：柴田一成) の助成を受けて行いました。本研究では、宇宙航空研究開発機構科学衛星運用・データ利用センター及び京大生生存圏研究所により公開されている衛星観測データを利用致しました。また、NICT リアルタイム地球磁気圏シミュレーションデータは、情報通信研究機構の SX-8R で計算致しました。

参考文献

- 1 eGY: The electronic Geophysical Year, <http://www.egy.org/>, 2010.
- 2 NASA Polar, Wind, and Geotail Projects, <http://www-istp.gsfc.nasa.gov/>, 2010.
- 3 SCOPEWG, <http://www.stp.isas.ac.jp/scope/index.html>, 2010.
- 4 松岡大祐, 村田健史, 藤田茂, 田中高史, 山本和憲, 木村映善, “Global MHD シミュレーションによる磁気フラックスロープの3次元構造解析”, 可視化情報学会論文集, Vol.28, No.6, pp.38-46, 2008.
- 5 Grid Datafarm- Gfarm file system, <http://datafarm.apgrid.org/>, 2010.
- 6 建部修見, 森田洋平, 松岡聡, 関口智嗣, 曾田哲之, “ベタスケール広域分散データ解析のための Grid Datafarmアーキテクチャ”, ハイパフォーマンスコンピューティングと計算科学シンポジウム HPCS2002 論文集, pp.89-96, 2002.
- 7 山本直孝, 建部修見, 関口智嗣, “グリッドデータファームにおける天文学データ解析環境の構築と評価”, 先進的計算基盤システムシンポジウム SACSIS2004 論文集, pp.233-240, 2004.
- 8 GEO Grid, <http://www.geogrid.org/>, 2010.
- 9 SPDF- Satellite Situation Center Web(SSCWeb), <http://sscweb.gsfc.nasa.gov/>, 2010.
- 10 DARTS at ISAS/JAXA, <http://darts.isas.jaxa.jp/>, 2010.
- 11 NICT Real-Time Magnetosphere Simulation, <http://www2.nict.go.jp/y/y223/simulation/realtime/home.html>, 2010.
- 12 村田健史, 岡田雅樹, 阿部文雄, 荒木徹, 松本紘, “太陽地球系物理観測の分散メタデータベースの設計と評価”, 情報処理学会論文誌, Vol.43, No.SIG12 (TOD16), pp.115-130, 2002.
- 13 村田健史, “国際太陽地球系物理観測の広域分散メタデータベース”, 電子情報通信学会論文誌 (B), Vol.J86-B, No.7, pp.1331-1343, 2003.
- 14 建部修見, 曾田哲之, 関口智嗣, “広域仮想ファイルシステム Gfarm v2 の設計と実装”, 情報処理学会研究報告, 2004-HPC-99, SWoPP2004, pp.145-150, 2004.
- 15 建部修見, 曾田哲之, “広域分散ファイルシステム Gfarm v2 の実装と評価”, 情報処理学会研究報告, 2007-HPC-113, pp.7-12, 2007.
- 16 鈴木喜雄, “グリッド環境の大規模可視化システム”, 第56回理論応用力学講演会論文集, pp.33-34, 2007.
- 17 中島研吾, 陳莉, “ボクセル型背景格子を使用した大規模並列可視化手法”, 情報処理学会研究報告, Vol.2006-HPC-107, No.87, pp.91-96, 2006.
- 18 T. W. Crockett, "An introduction to parallel rendering", In Parallel Computing, p.23(7): 819L843, 1997.
- 19 AVS/Express, http://www.avs.com/software/soft_t/avsxps.html, 2010.
- 20 Ken T. Murata, K. Yamamoto, D. Matsuoka, E. Kimura, H. Matsumoto, M. Okada, T. Mukai, J. B. Sigwarth, S. Fujita, T. Tanaka, K. Yumoto, T. Ogino, K. Shiokawa, N. A. Tsyganenko, J. L. Green and T. Nagai, "Development of the Virtual Earth's Magnetosphere System(VEMS)", Advances in Polar Upper Atmosphere Research, Vol.19, pp.135-151, 2005.



やまもと かずひろ
山本和憲

愛媛大学工学部研究生
Virtual observatory、セマンティックWeb

きむら せいぜん
木村映善

愛媛大学大学院医学系研究科准教授
博士(医学)
医療情報学、医療経済学



むらた けんじ
村田健史

電磁波計測研究センター宇宙環境計測
グループグループリーダー 博士(工
学) 宇宙情報工学、福祉情報工学



たにやま せいのみ
建部修見

筑波大学大学院システム情報工学研究
科准教授 博士(理学)
超高速計算システム、グリッドコ
ンピューティング



まつおか たいすけ
松岡大祐

海洋研究開発機構地球シミュレータ
センター研究員 博士(工学)
太陽地球系物理学、科学可視化



みやち ひでお
宮地英生

(株)ケイ・ジー・ティー ビジューア
リゼーション事業部技術部長
博士(工学)
可視化ソフトウェアの開発と利用支援