Evaluation of Parallel Distributed Processing of NICT Science Cloud for Data Analysis of Waveform Obtained by Spacecraft

研究論文

科学衛星で観測された波形データ処理を用いた NICT サイエンスクラウド上での並列分散処理の評価

Evaluation of Parallel Distributed Processing of NICT Science Cloud for Data Analysis of Waveform Obtained by Spacecraft

矢木大介[1*]，村田健史[2] ，笠原禎也[1]
Daisuke YAGI[1*], Ken T. MURATA[2], Yoshiya KASAHARA[1]

1 金沢大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Kanazawa University
〒920-1192 石川県金沢市角間町
E-mail: yagi@cie.is.t.kanazawa-u.ac.jp
2 情報通信研究機構
National Institute of Information and Communications Technology
〒184-0015 東京都小金井市貫井北町4-2-1
E-mail: ken.murata@nict.go.jp
*連絡先著者 Corresponding Author

昨今，種々の目的で打ち上げられる科学衛星や地球観測衛星の観測データは蓄積され続け，膨大な量となっている．自然科学ビッグデータをプロセス間通信等の複雑なプログラミングや大規模な環境設定なしに解析するためのデータ処理手法と環境の需要が高まっている．本論文では，サイエンスクラウド上で容易に並列処理の実装が行えるタスクスケジューラを用いたヘテロタイプの並列分散処理の性能評価について議論する．評価を行うにあたって，月周回衛星かぐやに搭載された WFC-L が観測した波形データセットと既存のデータ解析プログラムを用いた．今回提案するタスクスケジューリング技術および並列分散処理技術がヘテロタイプの時系列データ処理に適していることを実証する．

A variety of satellites for space investigation and Earth observation has been launched and are yielding a large amount of data. Easy and effective parallel processing technique is required to analyze such scientific big data without heavy programming. In the study we analyze a set of waveform data measured by the WFC-L receiver onboard Japanese lunar orbiter “KAGUYA” for 9 months using our original program. The total data size is as small as 144GB, but it takes long time (230 hours) to survey

all data files to detect specific waveform patterns. The practical issue is that it is not easy for many space scientists to rewrite a program via parallelization library such as MPI (message passing interface). Herein we import our original program, without rewriting, on a science cloud system on which a task manager is ready for use for development and management of parallel data processing. We demonstrate that easy task scheduling and parallel processing is effective and practical for big data analysis even in case that the data set is heterogeneous.

キーワード: タスクスケジューラ, ヘテロ並列処理, サイエンスクラウド, 衛星観測, 波形捕捉器
Keywords: task scheduler, hetero-type parallel processing, science cloud, satellite observation, waveform capture

1 はじめに

近年，気象，大気，海洋，森林など様々な目的で多種多様で膨大な自然環境計測データが取得され，人類の生活面での利便性向上や地球環境の把握と保全などの目的で利活用されている．宇宙空間計測も例外ではなく，地球の種々の環境を計測する地球観測衛星にとどまらず，太陽，地球・惑星圏，天文など，様々な目的で多数の科学衛星が打ち上げられ，膨大なデータが衛星から地上に伝送され，蓄積されている．例えば，1989年から25年余りにわたり観測を続ける磁気圏観測衛星あけぼののアナログ波形データは約30TBに及んでおり，今後も更に増え続けることが予想される．このような背景から，長期間あるいは広範囲にわたる膨大な自然環境計測データを包括的に解析し，自然科学の発展に活かす目的で，自然科学ビッグデータを解析できる研究環境の要求が高まっている．

科学衛星で得られた観測データを用いた研究に科学研究用クラウド環境を活用することで次のような利点が挙げられる．近年の自然科学観測データの解析は，数百GBから数百TBの自然科学データを一度に取り扱うことも多く，特に小中規模研究組織では観測データや解析結果を格納するための計算機リソースと大容量ストレージを独自に準備する必要がない．また，クラウドが提供するCDF(Common Data Format)，HDF(Hierarchical Dara File)などの汎用データフォーマットを取り扱うためのライブラリやIDL（Interactive Data Language）等の標準的データ処理ツールなどの研究環境を利用することができる．本研究で用いるようなタスクスケジューラの利用やその運用支援を期待できる場合もある．大規模プロジェクトにおいては複数の研究機関にまたがるデータ共有や共同研究を行うことが多いため，前述の研究環境を研究テーマごとに個別に整備することは非効率である．これに対し，クラウド上で研究環境を構築することで，これまで研究者個人で独立に整備していた観測データや研究者独自プログラムなどの研究環境および解析結果を共有できる．

NICTサイエンスクラウドは情報通信研究機構（以下，NICT）が次世代の科学研究環境を提供するために構築したクラウドシステムである［1，2］．ビッグデータサイエンスを主対象の一つとしており，ビッグデータサイエンス研究基盤の提供を目的に，広域観測網観測システム，広域分散ストレージシス

テム（Gfarm: GRID datafarm）［3］，世界科学データ保存システム（WSDB：World Science Data Bank）などの科学研究用環境が実装されている［4，5，6，7］．また，クラウド上にデータを保存・公開・移動するためのサーバ環境や，地球近傍の宇宙空間を3次元流体シミュレーションし，可視化するバーチャルオーロラツール（Virtual Aurora Tool）の提供など自然科学研究を行う基盤整備が進められており，計算機シミュレーションデータ処理や科学衛星データ処理システムとして広く活用されている［8］．

NICTサイエンスクラウド有効利用の具体例として，Gfarmシステムによる高速な自然科学データの読み込み・出力結果の書き出しを活用した分散I/O処理環境をクラウド上に構築し10TBを超える地球磁気圏Global MHDシミュレーションデータの高速可視化を行った例があり，96コアで125倍の高速可視化に成功した［9］．また，22年にわたる長期観測を継続的に行っているGEOTAL衛星に搭載されたプラズマ波動測定器（PWI）の観測データのスペクトルプロットを作成する処理において従来のデータ処理環境に比べて100倍以上の高速処理を実現し［10］，そのスペクトル画像をユーザが直感的に操作し閲覧できる異分野横断型時系列データプレビューア（STARStouch）を開発した実績もある［10，11］．

現在，著者らは月周回衛星かぐやで観測された電界波形データから特徴的な波形を抽出する処理のアルゴリズムを開発中であるが，研究室の単一コアの汎用PC Linuxを用いて全データ解析を実施するには約十日の処理時間を要するという問題があった．このような研究環境では処理アルゴリズムの改善を目的とする試行錯誤のターンアラウンドや，解析パラメータを様々に変えながらデータの詳細解析を複数回にわたって繰り返し処理を行う際に効率が悪い［注1］．本論文では，同解析手法をNICTサイエンスクラウド上に移植し，提供される大規模な計算リソースを利用して元のプログラムに大きな改変を加えることなく並列分散処理を実装し，得られる効率を定量評価する．NICTサイエンスクラウドが提供するタスクスケジューラを用いて分散並列処理環境を構築し，分散並列処理性能の評価をすることにより，同様の課題を有する他の研究テーマにおけるクラウドリソース活用の効率について議論する．

本研究の主対象はクラウド上のクラスタ計算機上のタスクスケジューラによる時系列データファイルの並列処理であるが，同程度の規模のシステムは民間クラウドサービスなどでも可能である．以下に，自然科学ビッグデータの研究基盤としてNICTサイエンスクラウド活用の着目点について議論する．

現在，本研究の対象となる太陽地球系科学分野だけを見ても米国を中心に整備されているSPASE（Space Physics Archive Search and Extract）［12］，京都大学・名古屋大学などで進められているIUGONET（Inter-university Upper atmosphere Global Observation NETwork）［13］，愛媛大学・情報通信研究機構のSTARS［11］など様々なメタデータ記述が提案されている．データ解析でもNASA/GSFCのCDAWeb（Coordinated Data Analysis Web）［14］，名古屋大学などによるTDAS/UDAS（Themis Data Analysis Software suite/iUgonet Data Analysis Software）［15］，上記のSTARS［11］などで統合化が試みられている．データ記述形式の標準化も進んでおり，本研究のCDF以外にもHDF

（Hierarchical Dara File）や FITS（Flexible Image Transport System）などの標準形式があり，これらのライブラリが完備していることが求められる．自然科学データを多角的にまた効率的に処理するためにはデータ収集，データ転送，データ保存（冗長化を含む），データ管理，データ処理，データ可視化，データ公開，セキュリティー管理を連動して行うシステムが望ましいが，これらを統合的に実現できているクラウドは太陽地球科学分野ではNICT サイエンスクラウドだけである．

また，今後の分野横断型研究を考える際にもNICTサイエンスクラウドの活用は有効である．前述のSTARStouch［10，11］はNICTサイエンスクラウドが開発した分野横断型データ可視化ツールであり，図5（後述）に示すような波形データのスケーラブル表示も可能である．本研究のデータはクラウド上の他の処理サーバ（STARStouch処理サーバ）からアクセス可能であり，これにより他のデータと比較する環境を比較的容易にSTARStouch上に実現できる．

2 NICTサイエンスクラウドのデータ処理環境

2．1 計算リソース

本節は本研究で活用するNICTサイエンスクラウドのデータ処理環境について述べる．NICTサイエンスクラウドユーザはネットワーク経由でゲートウェイサーバにsshによりアクセスし，クラウド上の大規模分散ストレージや分散データ処理サーバなどの研究用計算リソースを目的に応じて利用することができる［7］．

本研究ではNICTサイエンスクラウドが提供する高速並列ストレージシステムと大規模並列処理クラスタサーバを利用し，特定の波形抽出の並列処理を実装する．本研究で利用する並列処理クラスタサーバおよび高速並列ストレージシステムのスペックを表1，表2に示す．本研究では大規模並列処理クラスタサーバ（ホスト名はn100～n109）の10台を利用した（図2中の並列処理クラスタサーバ）．それぞれ12コアのCPUから構成されており，ハイパースレディングにより仮想的に24コアまで利用できる．また高速並列ストレージシステムとしてDDN社のSFA10000Eを採用した（図2中の高速並列ストレージシステム）．ファイルシステムのプロトコルであるGPFSについては2．2節で述べる．

表1 本研究で用いる大規模並列処理クラスタサーバのスペック

CPU	Xeon X5550 2.67GHz
コア数	12 コア（ハイパースレディングにより 24 コア）
OS	openSUSE11.1（x86_64bit）
メモリ	71GB/node

表2 本研究で用いる高速並列ストレージシステムのスペック（DDN社 SFA10000E）

物理容量	1.8PB
論理容量	620TB（RAID6+1）
1 ユーザあたりの容量	3TB
ファイルシステム	General Parallel File System (GPFS)

2．2 ファイルシステム

本研究では最大 240 コアによるかぐや衛星データの並列処理を行う．このような大規模並列処理ではファイルシステムの選択が重要である．一般的なネットワークストレージ（NAS）では，同時ファイルアクセスによりファイルサーバへのネットワーク I/O（スループット）の処理プロセス集中とネットワークトラフィックの輻輳により，ファイルサイズが小さい場合であってもファイルサー

バへのネットワークのI/O速度がボトルネックになる場合がある［注2］.

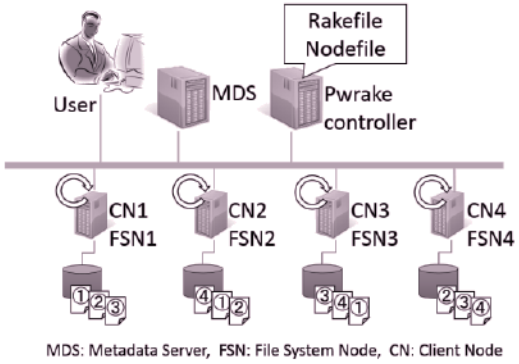

図1 Gfarm および Pwrake の基本システム

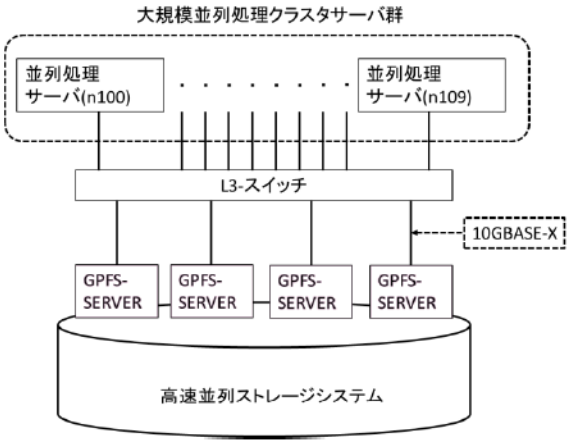

図2 大規模並列処理サーバ群と高速並列ストレージシステムの構成(ネットワークとスイッチはすべて10Gbps）：ホストn100が図1のPwrakeコントローラを兼ねている.

第1章で紹介したMHDシミュレーションの高速可視化事例では，数値データ処理（Processing）時間に対してディスク I/O 時間の割合が高かったため Gfarm[3]と Pwrake（2.3節で後述）によるファイルサーバへのネットワークI/O分散およびローカルファイルアクセスによるI/O高速化が有効であった. Gfarm システムの概要を図 1 に示す．Gfarm では，クライアントサーバ（Gfarm システムではクライアントノードと呼ぶ）がデータファイルにアクセスする際に，コントローラ（Gfarm システムではメタデータサーバ）は最適なファイルデータサーバ（Gfarm システムではファイルシステムノード）上のデータファイルへのアクセスを優先する. 特にクライアントノードがファイルシステムノードを兼ねている場合には，ローカルアクセスを最優先するスケジューリング（Affinity scheduling）がなされる．Affinity scheduling はディスクアクセスのローカリティを高めるためにディスクI/O高速化が期待されるが，データ処理を行う科学研究者には効率的なファイル配置など事前処理が必要である.

本研究で扱う衛星データは後述の通りデータ処理時間と比較してディスクI/O時間は短いため，ディスク/O 速度やネットワーク遅延の影響は小さい. そこで本研究ではスケジューラによる並列化効率の測定を主眼に評価を行う．タスクスケジューラとして図1の Pwrake を用いる. Pwrake については，2.3節および2.4節において説明する.

また本研究は，ストレージシステムの専門知識を有さなくても高速データ処理を行うことを目的の一つとしている. ファイルシステムとしては，上記の通りファイルローカリティー設定をユーザが行わねばならない Gfarm ストレージシステムではなく，ユーザがファイル配置を意識しない設定が可能な並列ストレージシステム GPFS（General Parallel File System）を採用する．2.4節（図3）に示すように，Pwrake では全処理ノードから同じパスでストレージをマウントできる場合に限り，Gfarm 以外のファイルシステムでも利用可能である.

並列処理クラスタサーバ（表1）と高速並列ストレージ（表2）から構成される本研究

の処理システムを図 2 に示す. 各並列処理クラスタサーバは高速並列ストレージシステムに GPFS（General Parallel File System）プロトコルで 10Gbps 接続されている．GPFS は IBM 社が開発した複数ノードから同時アクセス可能な高速並列ファイルシステムである．図 2 のシステムは GPFS サーバ 4 台から構成されており，カタログスペックで最高 40Gbps のネットワーク I/O 速度を提供しており，本研究のデータ処理のディスク I/O はボトルネックにならない. 高速並列ストレージシステムに処理対象となる全データを格納することで，各クライアント（図 2 の並列処理クラスタサーバ）は同じマウントポイント（例えば/gpfs）からデータ読み込み処理を行うことが可能である.

2. 3 タスクスケジューラ（Pwrake）

本研究のような並列分散処理（3. 2 節に詳細を記述）を行うためには，それぞれのタスク処理を各サーバ上のプロセスにスケジューリングする必要がある．本研究では NICT サイエンスクラウド上が環境を提供しているタスクスケジューラ Pwrake（Parallel Workflow extension for Rake）[16] を利用する．Pwrake の基本システムを図 1 に示す．Pwrake は Ruby で記述されたビルドツールである Rake を並列分散処理環境向けに拡張したタスクスケジューラである［注 3］.

Pwrake を実行するにあたっては, Rakefile と Nodesfile の 2 つのファイルを記述する必要がある．Rakefile には入出力ファイル名を含むワークフローを記述し，Nodesfile に使用するノード名および使用するコア数を記述することで，Pwrake タスクスケジューラ（本実験では図 2 中の n100 がタスクスケジューラとして制御を行う）は，Rakefile に記述したワークフローに従い指定したノードにタスクを割り振る．入出力ファイルを含むワークフローと処理ノード記述を分離することにより，ワークフロー記述を変更することなく処理ノードを設定できる．これは並列処理環境構築に不慣れな科学研究者でも効率的にリソース設定（他ユーザと共有）できることを示している.

```
1. i=0
2. input_file=FileList['/gpfs/nfs/yagi/CDF/*.cdf']
3. while input_file[i]!=nil do
4. output_file=input_file[i]slice(20..52)
5. task:default => inputfile[i] do |t|
6. file output_file inputfile[i] do |t|
7. sh"(/main #{t.prerequisites.join(' ')} ->
   /gpfs/nfs/yagi/X_xc/out#{t.name}_out.txt)"
8. end
9. i=i+1
10.end
```

図3 本研究で用いた Rakefile 記述例

```
1. seg-gfarm-n100.nict.go.jp 12
2. seg-gfarm-n101.nict.go.jp 12
3. seg-gfarm-n102.nict.go.jp 12
4. seg-gfarm-n103.nict.go.jp 12
```

図4 本研究で用いた Nodesfile 記述例：seg-gfarm-n100.nict.go.jp はホスト名 n100 に該当する

2. 4 Pwrakeの設定

本研究で使用した Rakefile を図 3 に示す．この Rakefile を実行した際の処理の流れについて説明する．2 行目で処理対象ファイルが保存されたディレクトリ内（/gpfs/nfs/yagi/CDF/）のファイル名を配列に格納している．図 3 ではワイルドカードを用いているためファイル名は明示的に指定されていないが，使用するファイルは拡張子が cdf で表される CDF と呼ばれるフ

ファイル形式で（CDF 形式の詳細は 3.3 節で述べる），ファイル名に観測日時を含んだ合計 2106 ファイルが時系列順に保存されている．図 3 の 3 行目から 10 行目までは 2 行目で作成した配列数だけループ処理を行っている．4 行目で入力ファイル名の一部を抽出し，出力ファイル名を生成している．5，6 行目では出力ファイルを得るために入力ファイルが必要であることを定義し，7 行目で実行するプログラム（実行ファイル main）の引数に入力ファイルを与え，処理結果を指定した出力ファイル名にテキストで書き込むコマンドを入れている．出力ファイルの保存先は/gpfs/nfs/yagi/X_ch/out（7 行目）となっている．よってループ処理は，処理対象ディレクトリ内の入力ファイル名を実行コマンドの引数に順次与え，それに対応した出力ファイルを生成する構造となっている．以上の 10 行程度の記述で，ループ処理内のタスクを Pwrake コントローラ（タスクコントローラ）が動的に各ノードに割り振り，処理が終了して待ち状態となっているノードに逐次的にタスクを与えるワークフローを実現できる．

次に，Nodesfile 記述例を図 4 に示す．この例では 4 台の計算機でそれぞれ 12 コアずつ使用することを指定しており，ユーザは使用する計算リソースの設定を容易に行うことが可能である．

本研究で扱うデータを含む自然科学研究で取り扱われるデータの多くは観測時系列順にデータが並んでおり，観測時間の割当てや観測条件の変化に依存して，ファイルサイズの異なるデータセットとなる場合が多い．単一プロセスを処理するためにユーザが開発した独自のデータ処理プログラムを用いて非均一なデータセットの処理を行う場合，元となるソースプログラムを MPICH（Message Passing Interface Chameleon）［17］等の並列ライブラリを用いてプロセス間で高い負荷バランスとなるように書き換える作業は，並列プログラミングに精通していない研究者には負担が大きい．

一方，プロセス間通信の必要がない多数のデータファイル処理では，タスクスケジューラを用いることにより MPICH のようにデータ処理用のソースプログラムを書き換える必要なく並列化できる．先に述べた Rakefile と Nodesfile を記述するのみで並列化を実装することができるため，ユーザは並列化を意識することなく単一プロセス処理用に開発した解析プログラムを利用することができる．

本研究が対象とする時系列観測データ解析では，第 1 章で述べた CDF，HDF，FITS などのデータフォーマットによるファイルアーカイブが進められており，データファイル単位での並列分散処理が必要となる．すなわちファイル単位でタスクの管理が可能であり，入力ファイルと出力ファイルの依存関係によるスケジューリングを容易に記述できる Pwrake は，MapReduce［18］や Torque/Maui［19］などファイル単位でのスケジューリングの記述が容易ではない並列分散処理技術と比較すると，自然科学データ処理の分散並列処理のスケジューラとして親和性が高い．また，Pwrake は Gfarm 用に開発されたスケジューラであるため，Gfarm ファイルシステムを利用した際に分散システムの透過性を保証できる点は，ビッグデータサイエンス研究を行う上で必要となる要件を満たしていると言える．

3 かぐやプラズマ波動データ処理

3. 1 かぐやプラズマ波動

月周回衛星かぐやは，月の資源利用・月周辺環境の調査を目的として，2007年9月から2009年6月まで運用された科学衛星である［20，21］．かぐや衛星に搭載された観測装置のひとつである波形捕捉（Wave Form Caputure：以下WFC）は，期間中定常的に月周辺のプラズマ波動の観測を行った[22, 23, 24]．WFCのサブシステムとして100Hzから100kHzの電界波形を観測したWFC-Lでは，特徴的なバイポーラ型の波形が多数確認されている［25］．

WFC-L波動データ上で確認されたバイポーラ型波形観測例を図5に示す．図5は横軸に波形を観測した時刻，縦軸に波形の振幅を示しており，2つの図はWFC-Lの差動観測チャンネルであるXチャンネル（上図）およびYチャンネル（下図）で取得された電界波形を示している．楕円で囲んだ部分が抽出対象となるバイポーラ波形である．このようなバイポーラ波形は周囲の波形に対し突発的に振幅が卓越することが特徴であり，数多くのバイポーラ波形が断続的に観測される時間帯がある．その一方で，長時間にわたりバイポーラ波形が観測されない時間帯もある．図5に示した波形はかぐや衛星によるバイポーラ波形の一例であり，この他にもプラスマイナスの極性が逆転しているものや，左右（時間的には前後）が非対称なもの［26］など，様々な形状を持つバイポーラ型の波形が確認されている．波形形状の違いと物理的成因や観測条件の関係を明らかにすることが，月周辺のプラズマ環境の知見を得る重要な手掛かりとなる．

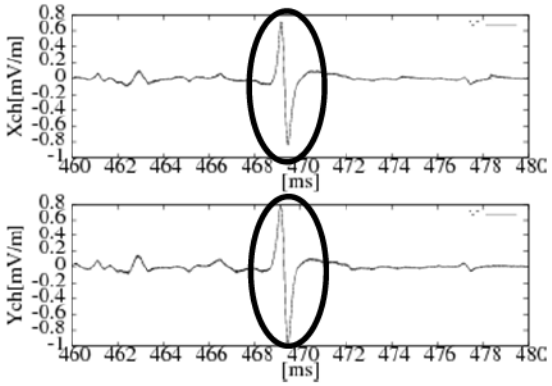

図5 かぐや衛星WFC-Lにより観測されたバイポーラ型波形観測例（2007年12月2日08:20:40 UT）

現在筆者らのグループではこれらの特徴的なバイポーラ波形を自動抽出するプログラム（図3中の実行ファイルmain）を開発中であり，抽出精度の改善や形状の違いによる分類を行うために，様々なアルゴリズムの改良やパラメータサーベイを実施している．しかし，単一コアの汎用PC Linuxを用いて全データ解析を実施するには約十日もの処理時間を要する．本研究の波形解析の目的は様々な軌道条件や衛星の観測モードによらず同一の抽出処理を行い，その結果多様な形状を持つバイポーラ波形パターンの出現を分類することであり，一部のデータセット処理ではアルゴリズムの検証が不十分である．そのため，特徴的なバイポーラ型波形を自動抽出する処理プログラムを書き換えることなく，2. 3節のタスクスケジューラを用いた並列分散処理を実装し，計算効率がどのように改善されるか評価を行う．

かぐや衛星に搭載されたWFC-Lは，1回あたり最大750, 000点，時間長にして1～6秒間の連続波形データを間欠的に取得し［23］，NASAが開発したCDF（Common Data Format）［27］と呼ばれるファイル形式でLevel-2データとして保存されている．CDFは自己表現型

式データフォーマットであり，欠測部をスパースレコードでパディングすることでファイル容量を節約しつつランダムアクセスを可能にするなど，科学衛星の観測データを利用しやすいデータ形式である．

WFC-L波動データは観測時間2時間毎に1つのCDFファイルに保存されており，時刻情報とその時刻に取得した波形が対応する構造となっている．かぐや衛星運用中にWFC-Lが取得した波形データは2990個（約190GB）のCDFファイルに保存されている．その中でも今回の波形抽出処理では，かぐや衛星が月面からの高度100kmで観測を行った定常運用期間で得られた2106個のCDFファイル（合計142GB）を対象としている．各データファイルサイズについては3. 2節で後述する．

3. 2 かぐやWFC-L波形抽出処理

本研究では，図6に示すように，1つのCDFファイルに対して1つの抽出結果をテキストファイルとして出力する．処理対象となるCDFファイルが複数であっても，各ファイル処理は互いに依存しない点が特徴である．一方，3. 1節で述べたようにWFC-LによるCDF形式の出力データは間欠観測データである．観測の時間長は衛星の軌道条件によって大きく異なる．図7は20MB刻みでファイル数をヒストグラムで示しているが，ファイルサイズは最小で3MB，最大で300MBと差があり，ファイルサイズのばらつきは大きい．数MB程度の比較的小さなファイルの処理時間は16秒程度であるのに対し約300MBのファイルの処理時間は27分と，ファイルサイズによる処理時間に大きな差がある．

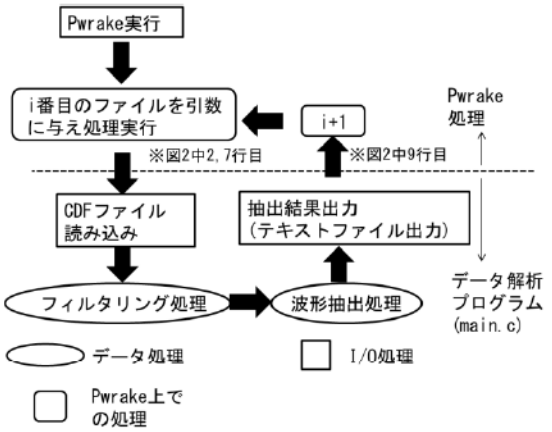

図6 波形抽出処理フロー図

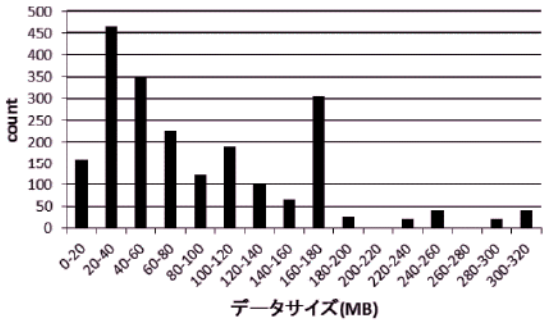

図7 かぐや衛星WFC-Lで観測された2, 106個のCDFファイルのデータサイズについてのヒストグラム

これらの特徴から，かぐやWFC-L波形データの様なヘテロ（非一様）並列処理では逐次的にタスクを空いているノードに割り振る動的スケジューリングが効率的である．この場合のヘテロ並列処理とは，ファイルのサイズによって処理時間が異なる処理（タスク）を管理し，並列に実行することを指す．

かぐやWFC-L波形抽出処理のフロー図を図6に示す．丸で囲まれた部分がデータ処理，四角がディスクI/O処理，角丸で囲まれた部分がPwrake側の処理を表している．まずPwrake側で図3の2行目で取得したファイル名を引数として与え，図3の7行目のように独

自に実装したデータ解析プログラム（実行ファイルmain）を実行する．データ解析プログラム側ではCDFファイルをひとつオープンし，内部に時系列順に格納された波形データとそれに対応した時刻情報を読み出す．次に，抽出対象とする自然波動以外の周波数帯の信号を除去するためバンドパスフィルターを通す．図5にも示したようにバイポーラ型波形は前後に対して顕著な振幅値を持つ．したがって閾値を設けることで，検出点から波形の抽出を行う．最後に抽出した波形の時刻情報を図3の7行目で与えたパス上にテキストファイル（数十KB）として出力する．出力後は，Pwrake側で時系列的に次に格納されているファイルを引数として与え，上記の処理を繰り返す．すべてのファイルが引数として与えられ，出力結果が出揃うと処理終了となる．

かぐやWFC-L波形データのヘテロ並列処理のイメージを図8に示す．Pwrakeでは対象となるファイルごとにタスクを作成し，それぞれのタスクを各ノード（各コア）に割り当てる．逐次的に処理が終わったノードにタスクを割り振るスケジューリングとなっている．今回の様なファイル処理ごとに処理時間が異なり，かつそれぞれのファイル処理に依存関係がない独立処理においては，効率の良いスケジューリング方法である．

図 9 にタスクが与えられたノードがストレージから必要なファイルを読み込む流れを示す．与えられたタスク名がそのままファイル名となっているため，各ノードはタスク名と同じファイルをストレージから読み込む必要がある．図 2 のシステムにおいて，すべての並列処理クラスタサーバ(n100 ～ n109) が同じディレクトリ(/GPFS/NFS/yagi/CDF/) で高速並列ストレージをマウントしているため，読み込みファイルのパス設定が 1 つで済む（出力の書き出しファイルも同様）．そのため Rakefile 内においてパスの指定が容易となる．

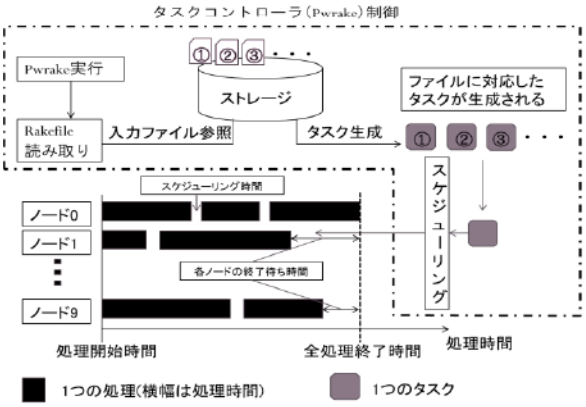

図8 Pwrakeによるヘテロ並列処理イメージ：Pwrakeでは対象となるファイルごとにタスクを作成し，それぞれのタスクを各ノード（各コア）に割り当てる．図ではノード1が最も早く処理を終了し，ノード0が最も時間をかけて処理を終了している．

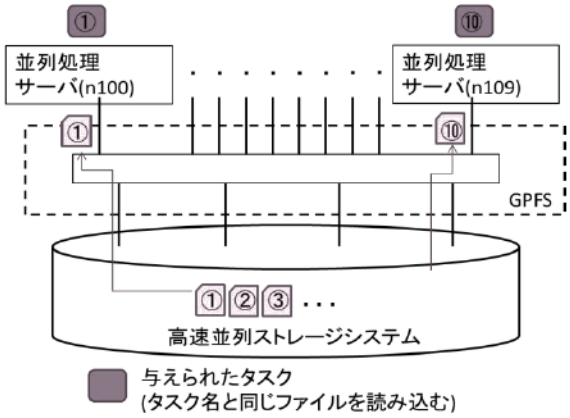

図9 タスクが与えられたノードがストレージから必要なファイルを読み込む流れ

最後に，本研究で用いたスケジューリングにおいて並列化効率を下げる要因について議論する．図8においてスケジューリング時間（タスク間の隙間時間）や，各ノードの最後のタスクの処理終了待ち時間は処理が行

われていない時間であり，並列化効率を下げる要因として挙げられる．第4章ではこれらの要因を考慮して高速化率（並列化効率）の評価を行う．

4 波形抽出処理の高速化評価予備実験

4. 1 評価実験概要

2. 1節で述べたNICTサイエンスクラウド上の大規模並列処理クラスタサーバ（表1）と高速並列ストレージシステム（表2）を使用し，第3章で述べた波形抽出処理についてPwrakeを用いたヘテロ並列処理を行った際の処理時間を測定する．ここでの処理時間とは，Pwrakeを実行開始してから各ノードの中で一番最後のタスクが処理を終えるまでの時間（図8の全処理終了時刻）とする．

処理時間を測定するにあたって，使用するノード数やコア数を変化させてNICTサイエンスクラウド上での並列分散処理の性能について評価する．なお2. 3節で述べたように，実験ではNodesfileを書き換えることで処理ノードやコア数の変更を容易に行うことができる．

4. 2 ノード数と処理速度の関係（予備実験）

予備実験として，任意に抽出したCDFファイル20個（合計2339MB）を対象に，各ノードのコア数を固定し，使用するノード数を変化させた際の処理時間を測定した．測定環境および測定結果を表3に示す．実験はホストn100～n104の5台を用いた．

この測定結果をもとに算出した，1ノード1コアを基準とした処理速度の向上率（高速化率）および並列化効率を図10に示す．図10より，ノード数に応じてほぼ一定に処理速度

表3 ノード数を変化させた処理時間の測定結果

ノード数	各ノードのコア数	処理時間
1	1	0:52:42
2	1	0:26:59
3	1	0:18:54
4	1	0:14:50
5	1	0:12:42

*CDFファイル20個（合計2339MB）について処理

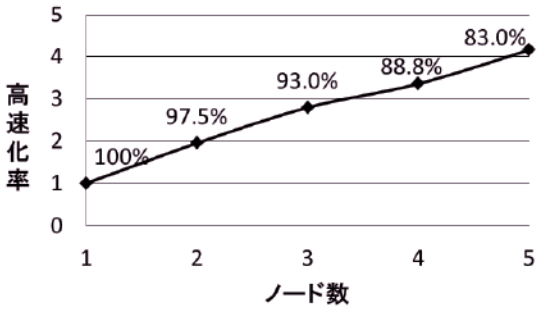

図10 ノード数1の場合を基準にしたノード数別の高速化率：図中の数値は並列化効率

が向上していることが確認できる．これはPwrakeによる並列処理が効果的に行われていることを意味しており，同等のコア数（5ノード・1コア）での処理であれば対象データファイル数を増やしても同等の高速化率が期待できる．

さらに，図10でノード数が増加するにしたがって並列化効率が低下する点に着目し，その原因を考察する．次節の本実験での大規模並列処理ではさらなる並列化効率の低下が予想される．本予備実験では20ファイルを処理するため，5並列の場合にはプロセスあたり平均4ファイルを処理することになる．図7によるとデータファイルサイズはファイルごとに大きく異なり，それに応じて処理時間にもばらつきが生じる．

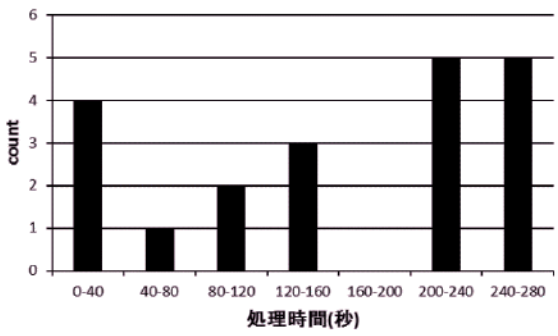

図11 予備実験で扱ったCDFファイル20個に対する処理時間のヒストグラム

図11に全20ファイルの処理時間の分布をヒストグラムで示す. また表4に表3および図10の5コア（5ノード）で処理した場合の各ノードの合計処理時間とファイルサイズ, および最も時間がかかったノード（n100）と最も速く処理を終えたノード（n103）との時間差を示す. 図11と表4によると, データサイズにばらつきがあるファイル群を処理する場合には, 図8に示したように全処理が早く終了するコアと時間がかかるコアがあることが分かる. 高速化率（並列化効率）は最大処理時間（図8の処理終了時刻）で評価するため, 終了時間のばらつきが大きい場合には並列化効率が見かけ上, 減少することになる.

表4 5ノード各1コアでの各ノードの処理ファイル数・サイズ・処理時間：合計ファイルサイズはノード毎の処理ファイルサイズの合計

ノード	ファイル数	合計ファイルサイズ（MB）	処理時間
n100	4	443	0:12:40
n101	3	524	0:10:49
n102	4	210	0:09:11
n103	6	789	0:09:29
n104	3	373	0:10:31
最大処理時間と最小処理時間の差		0:03:29	

表5 図11の20ファイルに対して処理ノード数を変化させた場合の処理時間差の変化：最大処理時間, 最小処理時間は図11ではノード0とノード1に対応する.

ノード数	各ノードのコア数	ノードごとに与えられる平均ファイル数	最大処理時間と最小処理時間の差
2	1	10	0:00:32
3	1	6.67	0:03:13
4	1	5	0:03:20
5	1	4	0:03:29

表5に図11の20データファイル処理において処理ノード数を変化させた際の最大処理時間と最小処理時間の差を示す. この表から, 各ノード（各コア）に与えられるファイル数が多いほど処理時間が均一化され, 図8のノードの終了待ち時間が減少する. すなわち, 処理時間のばらつきによる並列化効率低下は各ノードに与えられる処理対象ファイル数が少なくなるほど顕著である. また同じデータセットを処理する際, 処理プロセス数が増えるほど各ノードに与えられるファイル数が減り, 並列化効率が低下する傾向がある.

本節の予備実験では図11に示すようにファイル処理時間のばらつきが大きい. 例えば5コアでの処理では, 表4からわかるように各ノードの処理ファイル数および処理時間に差がある. したがって各ノードの処理時間の差により発生する他のノードの終了待ち時間（図8）が並列化効率を下げる原因となる. コア数を増やし, 全体の処理時間が短くなると, より顕著に並列化効率に影響を及ぼすため, 図10ではコア数を増やすほど並列化効率が下がる結果となったと考えられる.

5 波形抽出処理の高速化理論(本実験)

5. 1 コア数と処理速度の関係

本実験では，図2に示す全10ノードを用いてCDFファイル2106個の処理時間を測定する．測定は各ノードで用いるコア数を2から24（13以上はハイパースレッディング）まで変化させ，それぞれのコア（プロセス）における各データファイルの処理時間を測定した．処理を行った全ファイル数および測定結果を表6に示す．

表6 ノード数固定でコア数を変えたときの測定環境および測定結果

各ノードのコア数	ファイル数	ファイルサイズ（MB）	処理時間
2	960	70,918	5:22:36
4	960	70,918	2:43:13
8	960	70,918	1:27:46
12	960	70,918	1:02:51
16	1280	89,315	1:15:55
20	1600	111,920	1:26:16
24	2106	144,526	1:38:41

*10ノードで固定して測定

表6の1列目は測定に使用したコア数，2列目および3列目は処理の対象とした全CDFファイル数とその総容量を示している．表中に示すように，10ノード24コアで処理を行う場合は，かぐや衛星定常運用期間中に得られた全観測データである2106ファイル（144GB）について処理を行ったが，少ない計算リソースに対して2106ファイルの処理を行うと処理時間が掛かりすぎるために，段階的に処理ファイル数を減らして実験を行った．表6の4列目（処理時間）が，個々の実験環境下で処理に要した処理時間である．処理時間は予備実験と同様に，Pwrakeを実行してから各ノードの中で最後のタスクが処理を終えるまでの時間とする（図8）．

この測定結果をもとに，10ノード2コアで

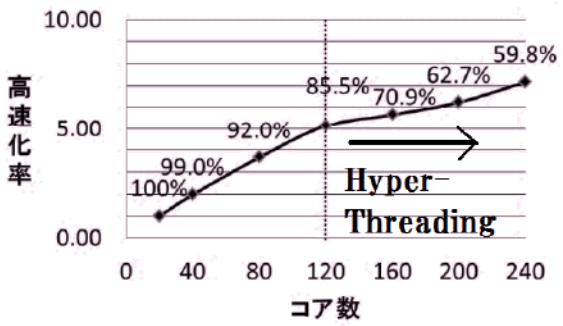

図12 10ノード2コアを基準とした各コア数の高速化率（図中の数値は並列化効率を示す）：測定は各ノードのコア数（プロセス数）を一定にして全10ノードを用いた．

の処理時間を基準として，その他のコア数の処理時間を比較した高速化率を図12に示す．10ノード1コアを測定基準としなかったのは，処理に時間がかかりすぎるためと，10ノード2コアの処理で十分な数のファイルが各ノードに割り振られ（1コアあたり平均48ファイル），各ノードの処理時間のばらつきが少なかったためである．

図12では，10ノード12コア（120コア）での処理においても，予備実験（図10）でも見られたように並列化効率が減少していることが確認できる．120コアの場合の各ノードの処理時間および処理ファイル数，処理サイズを表7に示す．処理時間差は最大で7分30秒であり，これが他のノードの待ち時間の要因となる．4. 2節でも述べたように，コア数を増やし全体の処理時間が短くなると，各ノードの待ち時間が顕著に並列化効率に影響を及ぼす．そのためにコア数を増やすほど並列化効率が下がると考えられる．従って図12で10ノード12コアまでの並列化効率がノード数を増やす毎に低下するのは，予備実験と同様に各ノードの処理終了時間のばらつきが原因である．

さらに，120コア以上の場合には，10ノー

ド12コア（120コア）での並列化効率は86%であり10ノード24コア（240コア）では60%と，コア数（プロセス数）を上げると並列化効率がさらに落ち込むことが図12より読み取れる．これは12コアを越えるコア数を使用するとハイパースレッドの影響が現れ，処理速度がコア数に比例しなくなるためと考えられる．

表7 120コアの場合の各ノードの処理時間のばらつきと処理を行った処理ファイル数:最大差は各項目の最大値と最小値の差を示す．

ノード	処理時間	ファイル数	合計サイズ（MB）
n100	0:57:54	89	6,809.05
n101	0:55:21	95	5,998.37
n102	0:58:47	106	7,557.22
n103	0:58:37	126	6,002.85
n104	0:58:48	93	6,683.33
n105	1:02:51	111	8,786.70
n106	0:54:58	108	6,901.10
n107	0:58:52	75	6,911.40
n108	0:58:57	82	6,578.77
n109	0:59:57	76	8,680.01
最大差	0:07:30	51	2788.33
標準偏差	126	15.7	925.85

5. 2 スケジューリング処理の評価

5. 2. 1 スケジューリング処理効率評価

本節ではPwakeによるスケジューリングについて，定量的に評価を行う．まず表6および図12の10ノード24コアで処理を行った場合について，各ノードの処理時間の詳細を調べた．各ノードが処理に必要とした時間およびその間に処理を行ったファイル数とファイルサイズを表8に示す．

各ノードの処理したファイル数と処理時間のばらつきを相対的に比較するために変動係数（標準偏差/平均値）を比較すると，ファイル数の変動係数は0.109（23.039個/210.7個）であり処理時間の変動係数は0.007（9250秒/66,215秒）であった．したがって各ノードの処理するファイル数のばらつきよりも各ノードの処理時間のばらつきのほうが十分に小さい．以上より，本実験ではスケジューリングが効率良く行われ，各ノードがほぼ均等に処理を実施したことが確認できた．このように，ノード・コアごとに処理ファイル数にばらつきがあるにもかかわらず高い高速化率を達成しているのはタスクスケジューリングが有効に働いていることを示している．

表8 240コアの場合の各ノードの処理時間のばらつきと処理を行った処理ファイル数:最大差は各項目の最大値と最小値の差を示す．

ノード	処理時間	ファイル数	合計サイズ（MB）
n100	1:34:43	221	13,883.39
n101	1:34:04	231	14,374.04
n102	1:33:21	216	15,399.62
n103	1:33:55	184	14,291.70
n104	1:34:57	230	14,055.56
n105	1:33:30	248	14,301.57
n106	1:34:29	198	14,996.70
n107	1:37:15	177	14,166.73
n108	1:38:41	181	14,521.43
n109	1:33:50	221	14,535.32
最大差	0:03:54	71	1,516.23
標準偏差	0:01:06	23.04	426.93

5. 2. 2 タスクスケジューリング時間評価

次にPwrakeによる処理タスクのスケジューリングに必要とした時間を測定した．測定時間評価にはPwrakeを用いた分散並列処理を終えた際に得られるログファイルを使用した．ログファイルの一部分を図13に示す．2795行目の波線を引いた部分のtask[start]に記載されている日時はタスク処理が実行された日時であり，workerの番号がどのノードのどのコアを使用しているかを示している（worker#1がn100ノードの1コア目，worker#2がn101ノードの2コア目…，worker#240がn109ノードの24コア目となっ

ている）．また2794行目の下線を引いた部分のtask[end]にも同様にタスクが終了した日時が記載されている．

同コア中でタスクが終了し次のタスクが開始されるまでの時間をスケジューリング時間（図8）とし，ログファイルより算出した．10ノード24コアで2106ファイル，すなわち2106個のタスクをスケジューリングするのに必要とした時間は合計0.168秒であった．1タスクあたりの平均は約80μ秒であるため，1時間38分41秒の処理時間と比較すると無視できる時間でスケジューリングが行われたことがわかる．以上からPwrakeのスケジューリングに必要とする時間は，本実験では並列分散処理の効率に大きく影響を与えないと考えられる．

```
      /gpfs/nfs/yagi/X_ch/output_all/selene_h0_wfcl_20
      080410080000_v00_out.txt¥n¥t¥t)"
2794. task[end]:2013-09-11T16:29:35.482849
      elap=0.384 worker#1
      task=selene_h0_wfcl_20080410080000_v00
2795. task[start]:2013-09-11T16:29:35.482897
      worker#1
      task=selene_h0_wfcl_20080807000000_v00
2796. ** Execute selene_h0_wfcl_20080807000000_v00
```

図13　ログファイルの出力例（一部抜粋）

5.3 高速化評価

表6に示すように，10ノード24コアで2106ファイルの並列処理を行った場合の処理時間は1時間38分41秒であった．これに対し，図10および図12の結果から処理速度はノード数に対して線形であること，またノード毎のコア数が12以下の場合はコア数に対しても線形であることを考慮すると，1ノード1コアで全2106ファイルを処理するために必要な処理時間は約230時間と見積もられる．すなわち，10ノード24コアは1ノード1コアで処理する場合と比較して，約140倍の高速化（1/140の時間短縮）が達成された．本研究で使用した波形抽出アルゴリズムに，計算量の多いフィルタリングや波形の特徴をパラメータ化して詳細に分類する機能を追加するなどの改良を加えても，各試行に対する結果を得るまで時間の大幅な削減に繋がる．これによりバイポーラ型波形の物理学的知見に関する研究効率が向上することが期待できる．

最後に，本研究で扱った波形抽出処理のディスクI/O処理時間とデータ処理時間の割合について調べた．測定にあたっては．今回用いたデータ解析プログラム（図6のプログラム）にC言語の標準ライブラリであるタイム関数（time）を用いてディスクI/Oに関する処理部分（CDFファイルの読み込みおよび出力テキストファイルの書き出し）とデータ処理部分の開始と終了の時刻を記録した．測定対象としたのは図7のヒストグラムのほぼ中央値である163MBおよび最大ファイルサイズである309MB，最小ファイルサイズである3MBの3つのCDFファイルである．これらのファイルを処理した際のデータ処理時間とディスクI/O処理時間の内訳を図14に示す．

図14(a)で163MBのCDFファイル全処理に要した時間は13分12秒であり，ディスクI/O処理は6秒（全体の0.8%），データ処理は13分6秒（全体の99.2%）であった，ディスクI/O処理中の読み込み処理の割合は1.4%，書き込み処理の割合は98.6%であった．またデータ処理については，波形抽出処理部分の割合が6.1%，フィルタリング処理の割合が93.9%であった．以上から波形抽出処理はデータ処理（フィルタリング処理）がほとんどの時間を占め，ディスクI/O処理は僅かであることが分かる．

また図14(b)に最大ファイルサイズである309MBと，最小である3MBのデータと

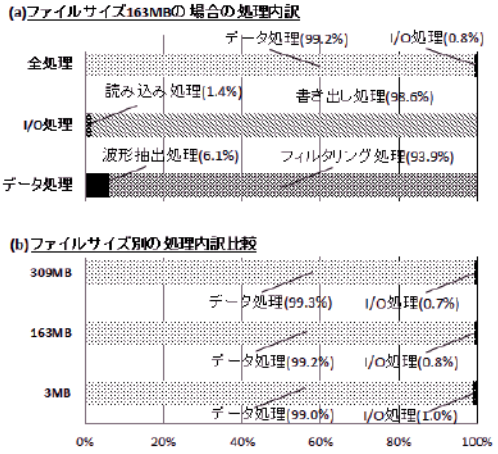

図14　ファイルサイズと波形抽出処理の関係：(a) ファイルサイズが163MBの場合の処理時間の内訳
(b) 異なるファイルサイズ（309MB・163MB・3MBのデータ処理時間およびディスクI/O時間の内訳比較

163MBのデータ処理内訳の比較を示す. 最大ファイルサイズ（309MB）処理においては全処理時間が27分3秒であり，データ処理は26分52秒（全体の99. 3%），ディスクI/O処理は11秒（全体の0. 7%）であった．また最小ファイルサイズ（3MB）処理においては全処理時間が16秒であり，データ処理は14. 6秒（全体の99. 0%），ディスクI/O処理は1. 6秒（全体の1. 0%）であった．以上より，今回対象とした波形データ処理ではディスクI/O処理部分はボトルネックとならず，5. 1節で述べたようにタスクのばらつきによる各ノードの待ち時間とスケジューリング時間や5. 2節のハイパースレッディングの効果が高速化率を低下させる要因であることがわかる. ディスクI/O処理時間の割合が高い処理に対して本研究の並列分散処理を実装する場合には，2. 2節で示した分散ファイルシステムを採用するなどの方法によりディスクI/O処理部分の負荷を軽減する必要がある.

6 おわりに

一般にクラウドの利用形態はSaaS，PaaS，IaaSに分類される．従来のサイエンスデータ処理環境は，計算機リソース（IaaS）と解析（SaaS）の自由度がトレードオフの関係になる二極化した傾向にあった. 既存のデータセンターが提供しているデータ処理環境（Webアプリケーション，ツールなど）のSaaSと，従来の研究者個人計算環境であるIaaSを両立させることは容易ではない．本研究のNICTサイエンスクラウド［1, 2］は，これらを両立するPaaSとして，並列分散処理に必要な全工程の下位レイヤー（IaaS）である計算機リソースおよび上位レイヤであるデータ収集・転送・保存などのバックグランド処理（SaaS）を満たしている．これにより，専門的な情報処理技術や処理環境を持たない自然科学者でも比較的簡便にビッグデータ処理を行うことが可能となった.

昨今，種々の目的で打ち上げられる科学衛星観測・地球観測衛星データはデジタルファイルとして蓄積され続け，多くのプロジェクトでTBからPBスケールの膨大なデータ量となっている．たとえば，2014年に打ち上げが予定されているひまわり衛星8号は，年間150TBのデータを出力する予定である．それに伴ってアーカイブされた自然科学ビッグデータの処理技術が必要となっているが，既存の処理プログラムをMPICH等の並列ライブラリで書き換えるのは専門的な知識を持たないユーザにとって容易ではない．また，多くの研究者はビッグデータ保存と処理を行うための大規模計算環境を有しておらず，研究プロジェクトごとの環境整備も研究者の負担となる.

一般に科学衛星観測・地球観測衛星データ

は時系列ファイルで保存されるが，時系列観測データ処理の多くはファイル間の相互処理が必要でない．特に一次データ処理では，蓄積された全データファイルに同じプログラムを適用することが多い．このようなサーベイ処理では，並列処理におけるプロセス間通信が発生しない利便性の高いタスクスケジューラを用いることで既存のプログラムを変更することなく並列処理が実現できる．

本研究では，自然科学分野のビッグデータを解析するための外部研究環境としてサイエンスクラウドに着目し，かぐや衛星波動データ処理の並列分散処理化を通してその性能の評価を行った．並列分散処理の対象としたかぐや衛星WFC-LのCDFファイルは個々のファイルサイズが大きく異なること，また波形抽出処理が適用するファイル毎に独立していることから，通常のプロセス間通信を用いた並列処理は効率的ではない．本研究ではタスクスケジューラPwrakeを使いヘテロ並列処理を実装し，実データを用いた処理時間を測定した．その際，既存のデータ解析プログラム（波形パターン抽出プログラム）に手を加えることなく10行程度のスクリプトを記述するだけで多数データファイルの並列処理環境を実現した．さらに，最大計算リソースである10ノード各24コアを用いて並列処理を行い，1ノード1コアによる処理に比べて約1/140の時間で処理を終えることが実証できた．またこのときの並列化効率は約60%と決して高い数値ではないが，より並列化効率を挙げるための改善点として，処理時間の長いタスクを優先的に処理することで各ノードの終了時間のばらつきを小さくすることなどが挙げられる．

GfarmによるFIFO型の並列分散処理のスケジューリングについては，これまでにも報告例がある［28，29，30］が，タスクスケジューラを使わずに独自のワークフローにより並列諸事を実現している．2.2節で述べたとおり，本研究では専門的なワークフロー記述なしに大規模計算を目指しており，本研究成果はこの点で今後の自然科学ビッグデータ処理に一石を投じると期待される．なお，本研究では，第一著者（矢木）は，サイエンスクラウドを初めて利用してから実作業時間2週間ほどで240コアによる高速化に成功した．この利便性は，Ready for useである科学研究クラウドの有効性を示している．計算環境だけではなく，第1章で述べたデータ伝送，データ保存，データ管理，セキュリティー対策などの総合的計算環境とビッグデータ処理が有効に連動することによる効果を示している．

このような高速化によって今後期待される効果について議論する．今回用いた波形抽出処理は，アルゴリズムの高度化により，バイポーラ型波形が観測されやすい月の太陽や地球との相対的な位置関係や，月に対する衛星の観測緯度・経度など，詳細な発生条件が統計的に解明されることが期待されている．これまでの筆者らの研究では，高精度の波形抽出を行うために様々なアルゴリズムの検証が行われ，波形の抽出頻度と観測条件の依存関係について様々な検討がなされてきたが，処理時間の長さが実処理上の障害となっていた．本研究の並列分散処理の実装により波形抽出アルゴリズム改善のターンアラウンドが向上し，この問題点は大幅に改善された．今後，波形抽出アルゴリズムの更なる精度向上，新アルゴリズムを実装したプログラムによる処理時間の大幅な向上，それに伴ったバイポーラ型波形の物理的解析の進歩が期待できる．

最後に，今後の衛星観測ビッグデータの高速並列処理の展望について述べる．今回は対象とした処理は，対象データのファイルサイズが約144GBとビッグデータと呼ぶほどの規模ではなく，またディスクI/Oに関する処理時間が十分に短い処理であった．しかし，長期観測を続けているあけぼの衛星や1日に数百GBのデータを地球に送信する地球観測衛星では，保存されるデータが数百TBからPBのビッグデータとなる．このような衛星観測ビッグデータのディスクI/O処理部分の割合が大きい処理（例えばグラフなどのサイズの大きい出力を伴う並列分散処理）を実装する際は，ファイルサーバとストレージのネットワークI/O並列化等により高速なデータI/Oが行うことが必要となる．本研究で提案したPwrakeは本来，分散ストレージシステムGfarmのために開発された［16］．今後，本研究をさらに拡張する場合や類似の並列データ処理においてI/O処理がボトルネックとなる場合には，GPFSではなくGfarmを用いることで，本研究による技法（とくにタスクスケジューラ設定）をそのまま適用してビッグデータ処理が実現できる．

謝辞

本論文のデータ処理の一部は情報通信研究機構のNICTサイエンスクラウドを用いて行われました．ここに深く感謝の意を申し上げます．

参考文献

［1］村田健史：「NICTサイエンスクラウドによる新しい科学研究の姿」，情報管理，Vol.55，No.8，pp.552-561，2012.

［2］村田健史：「サイエンスクラウドは第四の研究基盤となるか？」，学術の動向，Vol.17，No.6，pp.42-47，2012.

［3］Tatebe, O.; Hiraga, K.; Soda, N.: “Gfarm Grid File System.New Generation Computing”, Vol.28, No.3, pp.257-275, 2010.

［4］Murata, K. T.; Watari, S.; Nagatsuma, T.; Kunitake, M.; Watanabe, H.; Yamamoto, K.; Kubota, Y.; Kato, H.; Tsugawa, T.; Ukawa, K.; Muranaga, K.; Kimura, E.; Tatebe, O.; Fukazawa, K.; Murayama, Y.: “A Science Cloud for Data Intensive Sciences”, Data Science Journal, Vol.12, pp.WDS139-WDS146, 2013.

［5］NICTサイエンスクラウドWebサイト，http://sc-web.nict.go.jp （2014年5月23日参照）．

［6］村田健史；渡邉英伸；鵜川健太郎；村永和哉；鈴木豊；磯田総子；山本和憲；久保田康文；長妻努；坂口歌織；津川卓也；西岡未知；建部修見；田中昌宏；深沢圭一郎；才田聡子；海老原祐輔；藤田茂；木村映善；黒澤隆；村山泰啓；永井亨；水原隆道：「科学研究用クラウドシステム（NICTサイエンスクラウド）の提案」，宇宙科学情報解析論文誌（ISSN 1349-1113），Vol.3，pp.39-56，2014.

［7］村田健史；渡邉英伸；鵜川健太郎；村永和哉；鈴木豊；山本和憲；木村映善：「NICTサイエンスクラウド運用および利活用報告」，情報知識学会誌，Vol.24，No.3，pp.275-290，2014.

［8］村田健史；磯田総子；渡邉英伸；鵜川健太郎；村永和哉；鈴木豊；黒澤隆；木村映善；建部修見；田中昌宏；山本和憲；長妻努；津川卓也；佐藤晋介；笠井康子：「NICTサイエンスクラウド～地球規模観測ネットワークからのデータ収集・データベース・データ処理～」，情報処理学会パターン認識・メディア理解研究会（PRMU），IEICE-PRMU2013-61，Vol.IEICE-113，

No.230，pp.23−28，2013.
［9］村田健史；磯田総子；渡邉英伸；深沢圭一郎；山本和憲；建部修見；田中昌宏；木村映善：「NICTサイエンスクラウドによる大規模シュミレーションデータ分散可視化処理」，宇宙科学情報解析論文誌，Vol.3，pp.57−70，2014.
［10］Murata, K. T.; Watanabe, H.; Yamamoto, K.; Kimura, E.; Tanaka, M.; Tatebe, O.; Ukawa, K.; Muranaga, K.; Suzuki, S.; Kojima, H.: “A high-speed data processing technique for time-sequential satellite observation data”, IEICE Communications Express, Vol.3 , No.2, pp.74-79, 2014(doi:10.1587/comex.3.74).
［11］村田健史；鵜川健太郎；村永和哉；鈴木豊；渡邉英伸；是津耕司；山本和憲；篠原育；笠原禎也；岡田雅樹；小嶋浩嗣；能勢正仁；木村映善；建部修見；田中昌宏：「世界科学データシステム（WDS）のための学際的科学データ表示Webの提案」，情報知識学会誌，Vol.24，No. 3，pp. 297−320，2014.
［12］Space Physics Archive Search and Extract (SPASE),
http://www.spase-group.org/（2014年11月12日参照）.
［13］Inter-university Upper atmosphere Global Observation NETwork (IUGONET),
http://www.iugonet.org/（2014年11月12日参照）.
［14］Coordinated Data Analysis Web(CDAWeb)
http://cdaweb.gsfc.nasa.gov/（2014年11月12日参照）.
［15］Themis Data Analysis Software suite/iUgonet Data Analysis Software (TDAS/UDAS)
http://www.iugonet.org/software.html（2014年11月12日参照）.
［16］田中昌宏；建部修見：「並列分散ワークフローシステムPwrakeによる大規模データ処理」，宇宙航空研究開発機構研究開発報告，宇宙科学情報解析論文誌，Vol.1，JAXA−RR−11−007，pp.67−75，2012.
［17］Message Passing Interface Chameleon(MPICH)
http://www.mpich.org/（2014年11月12日参照）.
［18］J. Dean;S. Ghemawat;” MapReduce: Simplified Data Processing on Large Clusters”, OSDI, pp.137-150, 2004
［19］Torque/ Maui
http://www.adaptivecomputing.com/products/open-source/maui/（2014年11月12日参照）.
［20］Kato, M.; Sasaki, S.; Takizawa, Y., the Kaguya project team: “The Kaguya mission overview”, Space Science Reviews, Vol.154, No. 1-4, pp.3-19, 2010 (doi:10.1007/s11214-010-9678-3).
［21］宇宙航空研究開発機構(JAXA)ホームページ，http://www.jaxa.jp/projects/sat/selene/(2014年9月16日参照）.
［22］Ono, T.; Kumamoto, A.; Yamaguchi, Y.; Yamaji, A.; Kobayashi, T.; Kasahara, Y.; Oya, H.: “Instrumentation and observation target of the Lunar Radar Sounder (LRS) experiment on-board the SELENE spacecraft”, Earth Planets Space, Vol.60, No.4, pp.321-332, 2008.
［23］Kasahara, Y.; Goto, Y.; Hashimoto, K.; Imachi, T.; Kumamoto, A.; Ono, T.; Matsumoto, H.: “Plasma wave observation using waveform capture in the Lunar Radar Sounder on board the SELENE spacecraft”, Earth, Planets and Space, Vol.60, No.4, pp.341-351, 2008.
［24］Ono, T.; Kumamoto, A.; Kasahara, Y.;

Yamaguchi, Y.; Yamaji, A.; Kobayashi, T.; Oshigami, S.; Nakagawa, H.; Goto, Y.; Hashimoto, K.; Omura, Y.; Imachi, T.; Matsumoto, H.; Oya, H.; "The Lunar Radar Sounder (LRS) onboard the KAGUYA (SELENE) spacecraft", Space Science Reviews, Vol.154, No. 1-4, pp.145-192, 2010 (doi:10.1007/s11214-010-9673-8).

[25] Hashimoto, K.; Hashitani, M.; Kasahara, Y.; Omura, Y.; Nishino, M. N.; Saito, Y.; Yokota, Y.; Ono, T.; Tsunakawa, H.; Shibuya, H.; Matsushima, M.; Shimizu, H.; Takahashi, F.; "Electrostatic solitary waves associated with magnetic anomalies and wake boundary of the moon observed by KAGUYA", Geophysical Research Letters, Vol.37, No.L19204, 2010(doi:10.1029/2010GL044529).

[26] Kasahara, Y.; Horie, H.; Hashimoto, K.; Omura, Y.; Goto, Y.; Kumamoto, A.; Ono, T.; Tsunakawa, H.; LRS/WFC Team; MAP/LMAG Team: "Bipolar-pulses observed by the LRS/WFC-L onboard KAGUYA --- Plausible evidence of lunar dust impact --- ", Abstracts of EGU General Assembly 2010, p.10056, 2010.

[27] Common Data Format (CDF), http://cdf.gsfc.nasa.gov/ (2014年5月23日参照).

[28] 山本直孝；建部修見；関口智嗣：「Grid Datafarmにおける天文学データ解析ツールの性能評価」，情報処理学会研究報告，2003-HPC-95，SWoPP2003，pp. 185−190，2003. http://datafarm.apgrid.org/pdf/SACSIS2004-yamamoto.pdf

[29] 山本和憲；村田健史；木村映善：「Grid Datafarm における太陽地球系観測データの大規模統計解析の試み」，情報処理学会研究報告，Vol. 2006−HPC−107，No. 87，pp. 233−238，2006.

[30] 山本和憲：「グリッドデータファームによる並列分散処理」，情報通信研究機構研究報告，Vol. 55 No. 1 − 4，2009. http://www.nict.go.jp/publication/shuppan/kihou-journal/kihou-vol55no1.2.3.4/040201.pdf

注

[注1] 第 3.1 節でも述べるように，バイポーラ型波形はいくつかの形状に分類でき，様々な軌道条件や衛星の観測モードによらず適切な抽出が行われているか調べるため，部分データのみを用いたアルゴリズムの評価ができない.

[注2] 本研究では第 5.3 節で述べるようにデジタル処理時間に比して I/O 時間は十分に小さいが，プロセスやネットワークの輻輳により I/O 時間が無視できなくなることがある.

[注3] 拡張点としては，ssh によるリモート実行機能，Gfarm ファイルシステムのサポート機能，ローカリティを考慮したタスク配置（Affinity scheduling）実装が挙げられる.

（2014年5月23日受付）
（2015年2月14日採択）