

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

山本直孝[†] 建部修見[†] 関口智嗣[†]

観測的天文学の分野では観測装置の大型化により、得られる観測データの量が增大している。個人の研究者が取得した観測データを解析する場合には数台の計算機があれば十分であるが、多くの観測データを一括に処理するようなカタログ作成や、天体サーベイを行うには大規模ストレージ及び分散処理などをサポートしたミドルウェアが必要である。そこで、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のためのアーキテクチャであるグリッドデータファームを用いて大規模な天文学データ解析環境の構築を行った。グリッドデータファームの参照実装である Gfarm を用いて観測的天文学で広く用いられているいくつかのデータ解析ツールの並列化を行い、大規模な分散処理環境下での動作を確認した。この時、30 クラスタノード (60 CPUs) の並列データ解析において 5.9 GB/sec の読み込み、4.0 GB/sec の書き込みトータル I/O 性能が得られた。これにより、観測データの一次処理などのデータ解析を効率的に分散処理する事ができ、10TB にもおよびデータ解析を行う事が可能であることが示された。さらに、FITS ファイルブラウザの動作も確認し基礎的な解析環境を整備することができた。移動天体検出ツールの実行により、2002 年 12 月の観測データから未知の太陽系天体を検出した。

Performance Evaluation of Astronomical Data Analysis Infrastructure on Grid Datafarm Architecture

NAOTAKA YAMAMOTO,[†] OSAMU TATEBE[†]
and SATOSHI SEKIGUCHI[†]

Aiming at a comprehensive study of archival data of astronomical observatories that has a possibility of new science and new knowledge in the observational astronomical field, we constructed a large-scale parallel and distributed data analysis environment using the Gfarm reference implementation of the Grid Datafarm architecture. A set of data analysis tools including data reduction tools and a source extraction tool, were successfully executed in parallel on the Gfarm parallel file system. The parallel I/O operations has achieved data transfer rate of 5.9 GB/s for read and 4.0 GB/s for write using 30 cluster nodes (60 CPUs). A FITS-file browser successfully accessed the Gfarm file system, which gave a fundamental environment of astronomical data analysis. Using this data analysis environment, we found a new unknown solar system object from the archival data taken in December, 2002.

1. はじめに

天文学は、観測によるものと理論計算によるものとで大きく二分される。観測的天文学の分野では大型 CCD などの開発により観測装置から出力される観測データが増大している。日本においても文部科学省国立天文台ハワイ観測所に世界最大級の口径 8-m の主鏡を持つすばる望遠鏡¹⁾ が 1999 年に完成し、日々データが取得されている。すばる望遠鏡に取り付けられた観測装置の一つであるすばる主焦点カメラ²⁾ (Suprime-Cam) を例にとると 1999 年 1 月の観測開始から 2002 年 6 月までの間に観測された 2TB 弱の観測データが研究者に公開されている。すばる望遠鏡は複数の観測

装置を持つため、実際にはその数倍の観測データが取得され、18ヶ月の観測者占有期間を過ぎた観測データが SMOKA^{3)~5)} を通じて順次公開されている。

また、世界中の望遠鏡で観測された観測データが各所でアーカイブされ公開されている^{3),6),7)}。別の手法として掃天観測 (SDSS⁸⁾ など) も観測天文学の一部として行われている。掃天観測の結果も公開されており、研究者は必要な観測データをダウンロードすることができる。これらは、アーカイブデータを用いた新たな観測天文学の一分野としてアーカイブ天文学と呼ばれている。ここでも、多くの研究者はアーカイブされている観測データから興味のある天体のみを選択し、解析を行っている。掃天観測では専用の解析環境を用意し、全ての観測データに対する処理をパイプラインなどで実装している場合はあるものの、大量のデータをダウンロードし研究者もしくは研究グループで解析

[†] 産業技術総合研究所グリッド研究センター
Grid Technology Research Center, AIST

することは解析環境が十分でないため、行われていない。汎用で信頼性のある解析環境の構築のためには大規模ストレージおよび分散処理などをサポートしたミドルウェアが必要とされる。また、グリッド技術を応用した仮想天文台⁹⁾ (VO: Virtual Observatory) の構築が各国で行われている。VO では、データ検索クエリの標準化を行うことで世界中に分散されている観測データの検索を可能にするが、VO 標準化では解析環境については十分な議論がなされていない。

アーカイブデータを一括して処理することにより新たな天文学的知見が得られる可能性はいくつかあり、天体カタログの作成や新天体の検出などが考えられる。アーカイブデータから天体を総あたりで抽出することにより、非常に暗い天体を含む広範囲の星や銀河などのカタログを作成することが可能である。掃天観測でもカタログ作成は行われているが、全天をサーベイするために明るい天体のみをターゲットにしていることが多く性質が異なる。アーカイブされている観測データはより暗い天体の情報を保持しているため、これらを一括処理することにより、全天には及ばないが広域でかつ暗い天体を含むカタログを作成することができる。さらに、アーカイブデータには観測者が意図しない天文学的価値のある情報が埋もれている可能性がある。特に太陽系天体はその位置が移動してしまうため画像の重ね合わせ処理によってその姿は合成画像から消えてしまう。これらの埋もれてしまった移動天体を探索することにより、多くの太陽系小天体の発見が期待できる。

現在公開されている約 2TB の Suprime-Cam の観測データを解析対象にした場合、約 10 倍の 20TB のストレージが必要である。この数字はあくまでも Suprime-Cam によって取得され、現在までに公開されている観測データまでを扱うための見積りであり、新たな観測データも順次追加して公開されている。観測装置や手法によって取得される観測データの量は一定しないが、数十 GB 程度の観測データが毎晩生成されている。これらのデータ解析を如何に円滑に行うことが出来るかが天文学的成果に結びつける上で重要である。

グリッドデータファーム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を可能とするためのアーキテクチャとして、(1) ファイル複製管理を含む広域仮想ファイルシステム (Gfarm ファイルシステム) (2) ファイル集合 (Gfarm ファイル) の管理による分散並列データ処理を提供する。その参照実装として Gfarm が開発され、2003 年 12 月にバージョン 1.0.1 が公開されている¹⁰⁾。Gfarm ファイルを構成するファイルは物理的には分散配置されるが、Gfarm URL と呼ばれるパス名でアクセスすることができる。Gfarm ファイルに対する並列アクセスは、スケジューラによってファイルアクセスがローカルアクセスとなり、ノード数に応じたスケラブルなファイルアクセ

表 1 一次処理コマンドのリストとそれぞれの用途

(1)osmed	画像のトリミングとバイアスの引き算
(2)medianimg	複数画像のメジアン合成
(3)uppercut	バットピクセル除去
(4)arithimg	画像四則演算 (フラットの割算)
(5)distcorr	画像の光学歪み校正
(6)skysb	スカイバックグラウンドの除去
(7)imcio	モザイク合成画像作成

ス性能を実現する。Gfarm の実装に関してより詳細は、文献 11)~13) を参照されたい。

このように、グリッドデータファームアーキテクチャは天文学データ解析で求められている大規模解析環境を構築する機能を提供していると言える。本論文では、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 を用いて天文学データ解析環境の構築を行い、その動作を検証する。同時に、天文学データ解析ツールにより並列 I/O の性能評価を行う。

2. 天文学データ解析

2.1 観測データ

光赤外領域での観測データのほとんどは FITS¹⁴⁾ とよばれる画像形式で保存される。FITS ファイルはアスキーヘッダ部とバイナリデータ部で構成されており、ヘッダ部には観測日時や観測装置固有の情報が記録され、バイナリデータ部には 16bit の 2 次元画像データが格納されている。解析ツールは FITS ファイルのヘッダ部の情報を元に画像データを読み込み、解析結果を出力する。特に、後に述べる一次処理では解析結果も FITS ファイルとして出力されることが多い。

すばる主焦点カメラ (Suprime-Cam) はすばる望遠鏡の主焦点に取り付けられている超広視野カメラであり、約 2000 × 4000 ピクセルの CCD を 10 枚並べたモザイクカメラである。各 CCD チップが約 17MB の FITS ファイルを出力し、1 回の撮像で 10 枚の FITS ファイルが出力される。その時、SUPA00168280.fits のようなファイル名で FITS ファイルを生成し、末尾の 1 桁は 10 枚の CCD チップ毎に 0 から 9 までの数字が割り当てられる。通常、一晩の観測の中で撮像を繰返し複数セットの観測データが取得される。

2.2 データ解析の流れ

一般に、観測装置から出力された観測データはさまざまな較正 (一次処理) が必要であり、元データのままでは天文学的研究に利用する事ができない。すばる望遠鏡の場合、一次処理を行うツールパッケージが研究者に公開されていて、そのコマンドの一部と用途を表 1 に示す。図 1 は N 回撮像した観測データのセットに対する解析の流れを模式的に表したものであり、図中の括弧付き番号は表 1 に示した処理に対応している。

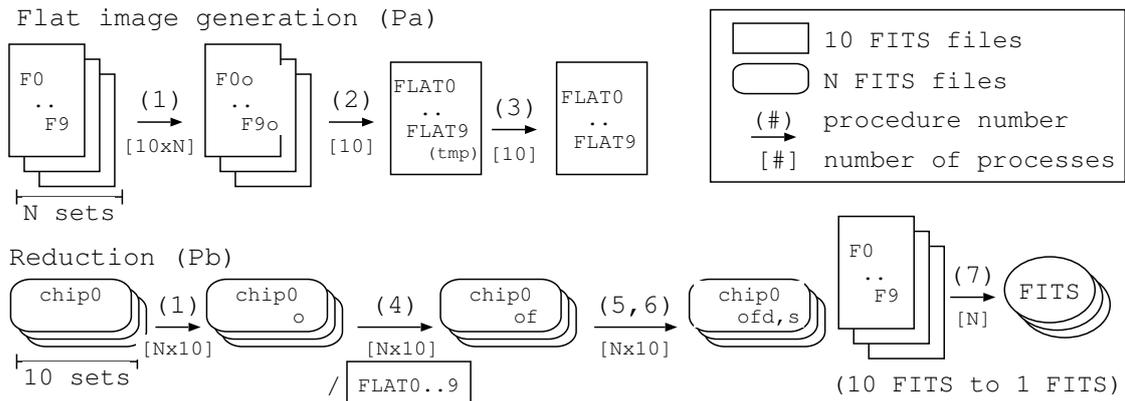


図 1 N 回撮像した観測データのセットに対する解析の流れ

2.2.1 校正処理

観測データの校正処理は、図 1 に示されるフラットの作成 (Pa) と一次処理 (Pb) で構成されている。処理 (Pa) では、CCD のピクセル感度ムラを補正するためのフラット校正用画像の作成を行う。フラット画像の作成は図 1 にある処理 (Pa) に属する処理 (1-3) で行われる。処理 (1) は 10 枚の N セットで完全独立処理になっている。処理 (2) は、チップ ID が等しい N 枚を元に 1 枚の合成画像を出力する。したがって、処理は 10 個の独立プロセスから成り、処理 (3) によってフラット画像の作成が完了する。また、作成された 10 枚のフラット画像は、観測日近辺のデータに対しても再利用が可能であるという特徴をもつ。

処理 (Pa) によって生成されたフラット校正用画像を用いて、処理 (Pb) を行う。処理 (Pb) では処理 (1,4-6) を行う。処理 (4) の画像の割算の入力は $10 \times N$ 枚であり、フラット画像は各 CCD チップに対応したチップ ID のフラット画像が用いられる。図 1 中の chip0o に FLAT0 を適用するので、フラット画像は N 個のプロセスから共有されることになる。処理 (5) は $10 \times N$ の独立処理である。

各々の処理では、新たな FITS ファイルを $10 \times N$ 枚生成する。また、処理 (1) の段階で、データ部分は 16bit から 32bit へ変換されるので元データに対して処理 (Pb) によって 8 倍の新しいデータが生成されることになる。元データ、さらにカタログや校正用データなどを含めると、全ての処理を行うためには元データの 10 倍規模のストレージが必要とされる。引続き述べる解析処理でもさらにストレージが必要とされる。データ解析を行う研究者は、データを複数のノードに分散させ解析を行う場合もしばしばある。これにより並列に処理を行うことが可能であるものの、分散した他のノードにあるデータにアクセスする必要があるため NFS などを用いたファイル共有を行う必要がある。しかし、ここで述べた一次処理ツールの多くは一回の実行でファイルアクセスが占める割合が非常に多く、

NFS では低速なファイルアクセスがボトルネックになってしまう。一方で、複数の CPU と大規模なストレージを持つようなシステムでも解析は行われているが、並列に同時処理を行う場合は、ここでもファイルアクセスがボトルネックになってしまう。また、単体の DISK では容量に限界があり、単純なマルチプロセッサシステムは解析環境として最適ではない。

2.2.2 天体検出および画像の可視化

一次処理が終了した観測データから物理量を抽出する処理を行い、実際の研究成果に結び付ける。その一例として、天体検出ツールである SExtractor¹⁵⁾ (Source Extractor) についてここでは述べる。このツールは FITS ファイルから星や銀河などの天体を抽出し、輝度や形状、分類パラメータをテキスト形式のリストで出力するものである。銀河カタログの作成や、新天体検出などさまざまな分野で広く利用されている。SExtractor は天体検出のための各種パラメータをパラメータファイルから読み込み、FITS ファイルから目的の形状を抽出する。観測データセットから楕円構造をした銀河を選定するというような操作を行うため、すべての観測データに対して共通のパラメータファイルが用いられる。ここでの処理は FITS ファイル一枚毎に独立に行われる。

コマンドラインの処理とは別に、観測データの性質や、処理の正当性を確認するために FITS ファイルブラウザが解析の過程ではしばしば用いられる。図 2 に示した SAOImage ds9¹⁶⁾ は広く利用されている FITS ファイルブラウザの一つである。ds9 は、Tcl/Tk で記述されたグラフィカルなユーザインタフェースを備えており、ファイルダイアログから FITS ファイルを参照することが可能である。さらに、恒星や銀河などのカタログデータを画像の上に描画する機能も備わっており、SExtractor によって検出された天体をプロットすることも可能である。

以上が天文学データ解析の概要であり、解析ツールによる FITS ファイルの読み込みおよび書き込み。解

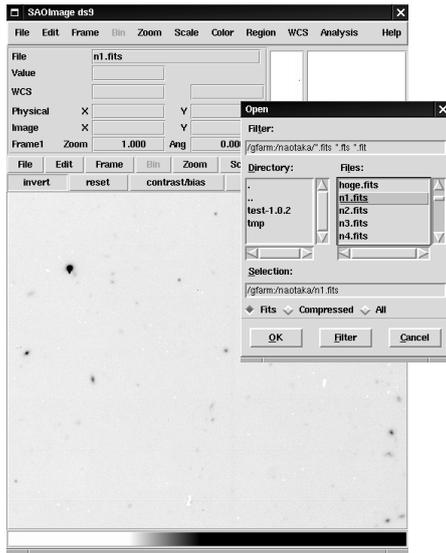


図 2 SAOImage ds9 のキャプチャイメージ

析結果などの出力や、FITS ファイルブラウザなどから構成されている。

3. グリッドデータファームによる実装

本論文では、環太平洋グリッドデータファームテストベッド^{17),18)}を利用して天文学データ解析環境の構築を行った。本章では、2章で述べた観測データのGfarm ファイルシステム上への分散配置、ならびに解析ツールの並列化について述べる。

3.1 Gfarm

Gfarm はグリッドデータファームの参照実装であり、広域仮想ファイルシステム (Gfarm ファイルシステム) を提供する。Gfarm ファイルシステムは実行コマンドおよび、ファイルの集合などを含む Gfarm ファイルを格納する。

Gfarm では、Gfarm ファイルを構成するそれぞれのファイルに対する並列入出力処理を可能とするためにローカルファイルビューと呼ばれるファイルビューの拡張がなされている。また、異なる複数のノードに分散配置された構成ファイルを格納場所で分散して処理するために、ファイル・アフィニティ・スケジューリングが導入されている。これらを組み合わせることにより、分散配置されたファイルの集合に対する並列分散処理が可能となる。

また、Gfarm ファイルのそれぞれの構成ファイルは、ファイルの複製を作成することができ、ファイル・アフィニティ・スケジューリング時には、ファイルの複製を格納しているノードのうち CPU 負荷の低いノードが選ばれる。このとき、故障などで利用不能となっているノードは排除されるため、障害に対する耐性を

もっている。

アプリケーションで Gfarm ファイルを扱うためには Gfarm 並列 I/O API (以下、単に Gfarm API) を利用する必要がある。しかし、ソースコードを Gfarm API へ改変することは、解析ツールのバージョンアップへの対応などを考慮すると現実的ではない。また、天文学に限らずプログラムの変更により誤った結果を得ることは避けなければならない。また、ソースコードが公開されていない場合などもあり得る。そこで、Gfarm ではシステムコールをトラップし Gfarm ファイルを既存のアプリケーションで利用するための仕組みを提供している (フック機能)。open を例にとると、以下のようにファイル名が Gfarm ファイルであるかがチェックされ、Gfarm ファイルの場合は Gfarm API の gfs_pio_open が呼ばれ、それ以外の場合はシステムコールが呼ばれる。

```

_open(file, ...) {
    if (file is a Gfarm file)
        gfs_pio_open(file, ...)
        set appropriate file view
    else
        syscall(SYS_open, file, ...)
    endif
}

```

Gfarm ファイルがオープンされた後、呼ばれた文脈に応じたファイルビューが設定される。並列実行されたプログラムで、並列実行ノード数と Gfarm ファイルを構成するファイルの数が等しい場合には、ローカルファイルビューとなり、それぞれのプロセスは、対応する個々のファイルへのアクセスとなる。それ以外の場合は、グローバルファイルビューとなり、一つの連結したファイルとしてのアクセスとなる。

3.2 観測データの配置

Gfarm による分散並列処理を実現するために、複数の FITS ファイルから 1 つの Gfarm ファイルを作成し、それぞれの FITS ファイルを異なるノードへ配置する。ここでは、図 1 の処理 (Pa)、処理 (Pb) を行うために、2 種類の Gfarm ファイルを作成した。処理 (Pa) を 10 並列で処理するため 10 ファイルからなる Gfarm ファイルと処理 (Pb) を N 並列で行うため N ファイルからなる Gfarm ファイルである。なお、ファイルの集合は Gfarm ファイルとして Gfarm ファイルシステムに登録される時、自動的に複数の異なるノードに分散格納される。また Gfarm ファイルを構成するファイルは自由に複数のノードに複製を作成することができる。

3.3 並列化と Gfarm ファイルへのアクセス

続いて、Gfarm ファイルシステムで解析を行うための解析ツールの整備を行う。ここでは、前に述べたアプリケーションの変更が不要となるフック機能を利用した。フック機能を利用し並列実行する場合、ファイ

ルの集合に対するアクセスはローカルファイルビューとなる。このとき、解析ツールは FITS ファイルを単位として並列にファイルアクセスを行うのでソースコードを改変すること無く動作する。例えば、図 1 の処理 (Pb) にある処理 (1) では、 N 枚の入力に対して N 並列で動作し、 N 枚の FITS ファイルで構成された Gfarm ファイルを並列に出力する。

例えば、SExtractor を逐次に行うにはパラメータファイルを引数として指定し次のように行う。

```
% sex -c detect.param input.fits
```

ここで、出力ファイルはパラメータファイルの中で指定される。これを Gfarm で並列実行する場合には、次のようになる。

```
% gfrun -r -G gfarm:input.fits gfarm:sex
-c gfarm:detect.param gfarm:input.fits
```

このときプログラム `gfarm:sex` は Gfarm ファイル `gfarm:input.fits` を構成するファイルの数で並列実行される。また実行ノードは `gfarm:input.fits` の格納ノードと CPU 負荷により決定される。ここで、パラメータファイル `gfarm:detect.param` は単一のファイルからなる Gfarm ファイルであり、並列実行ノード数と異なるため、グローバルファイルビューでアクセスされる。`gfrun` の `-r` オプションはファイルを遠隔参照する場合、参照に先立ち動的に実行ホストにファイル複製を作成する。パラメータファイルである `gfarm:detect.param` は、その後の SExtractor の実行で繰り返し再利用されるものであるがこのオプションを付けることで複製が作成され、2 度目以降のファイルアクセスを高速にする。

更に、C++ と Tcl/Tk で書かれた FITS ファイルブラウザである SAOImage ds9 もフック機能を用いることで Gfarm ファイルへのアクセスが可能となった。ds9 は並列実行を行う必要はないので、`gfrun` を用いずに直接コマンドを起動する。このとき、Gfarm ファイルに対するアクセスはグローバルビューとなり、図 2 のように Gfarm ファイルを指定することができる。データ解析用のコマンドに加えて、FITS ファイルブラウザが動作することで Gfarm ファイルシステムを用いた基本的な天文学データ解析環境を整えることが出来た。

4. 実行時性能測定

前章で、解析ツールの動作が確認されたことを述べた。本章では、Gfarm の並列 I/O および並列実行の性能評価を行う。実行時性能測定では、テストベッドの一部である AIST Gfarm Cluster I⁽²⁰⁾ の 30 ノードを占有して行った。また、実行ノードとは別にクライアントおよびメタサーバノードも性能評価用にクラスター内部に用意し外的要因を排除した。利用したクラスターは全てのノードが dual Xeon 2.8GHz, 1GB メモ

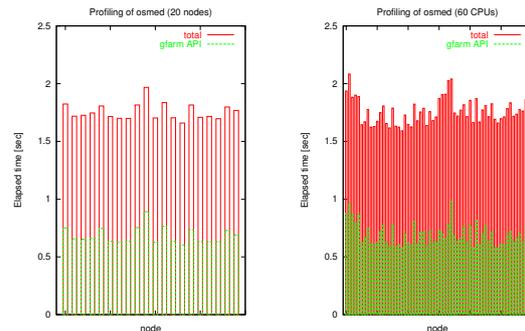


図 3 ノード数を変化させた場合の実行時間と Gfarm API の所要時間

リ、ギガビットイーサネットで構成されている。ディスクは 4 台の 200GB IDE HDD からなる RAID0 で構成されている。Gfarm の性能評価を行うため、以下の Gfarm API の所要時間を計測した。

```
gfs_pio_create, gfs_pio_open, gfs_pio_close
gfs_pio_seek, gfs_pio_read, gfs_pio_write
gfs_pio_getline, gfs_pio_set_view_global
gfs_stat, gfs_unlink
```

並列実行性能を測定するために、各ノード毎の処理が均一になるよう 2002 年 12 月 1 日に観測された 1 枚の FITS ファイルのみで複数の Gfarm ファイルを構成した。ここでは、測定対象のノード数と等しい数で構成された Gfarm ファイルを用意した。

4.1 並列入出力性能評価

バイアスを画像から引き算するための `osmed` を用いてノード数を 60 まで変化させた場合のトータルの実行時間に対する Gfarm API の割合の測定を行なった。ノード数を 20, 60 とした場合の解析実行時間に対する Gfarm API の所要時間を図 3 に示した。どちらのノード数でも実行時間 2 秒弱に対して、Gfarm API の所要時間は 0.6 から 0.9 秒程度になっている事が分かる。Gfarm API の所要時間が増加していないことから、ファイル I/O バンド幅はスケールしていることが伺える。そこで、60 並列 (30 ノード, 60CPU) までノード数を変化させた場合の、読み込みおよび書き込みのトータル I/O バンド幅を計測した結果を図 4 に示した。`osmed` は、約 17MB の FITS ファイルを読み込み、約 34MB の FITS ファイルを出力する。この処理では、FITS ファイルのバイナリデータ部分を 16bit から 32bit への変換するためファイルサイズが約 2 倍になる。この図から、60 ノードまでは非常に良くスケラビリティが保たれている事が分かる。ただし、書き込み性能が 30 ノードを超えたところで低下しているのは、1 ノードあたり 2 プロセスが動作したためである。

4.2 複製の自動生成機能による効果

次に、複製の存在による性能改善の評価を行った。

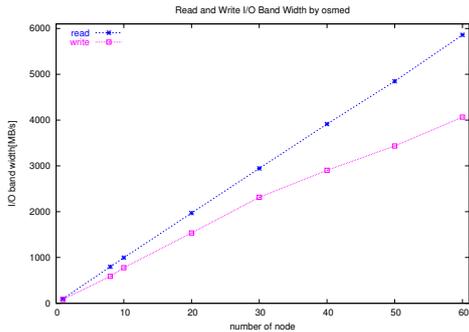


図 4 ノード数を変化させた場合の I/O band 幅

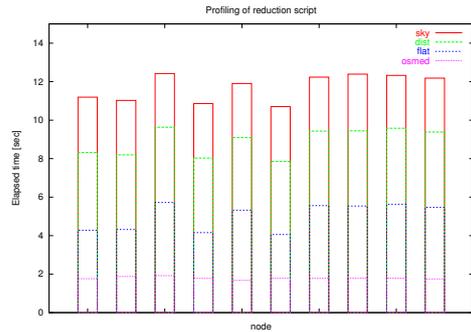


図 6 スクリプトによる連続実行時の各コマンドの所要時間

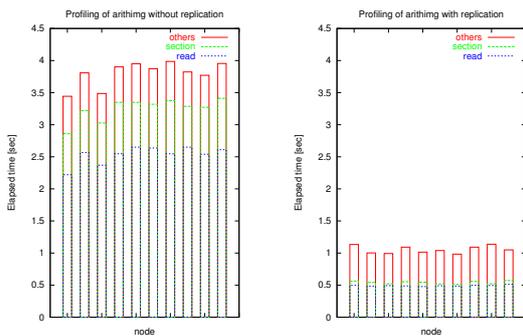


図 5 複製の自動生成機能による効果．左図はフラット画像が 1 ノードのみに存在する場合．右図はフラット画像の複製が実行ノード全てに存在する場合．

一次処理の一つである arithing を用いた観測データのフラット処理ではフラット画像が複数のプロセスから参照される．先に述べた gfrun の -r オプションによる動的ファイル複製作成が有効に働くものと予想される．そこで、フラット画像が一つのノードに作成された直後に実行した arithing と、2 回目の実行を比較した結果が図 5 である．

1 度目の実行では、フラット画像がフラット画像を作成したノードのみに存在しており、ファイル複製が作成されていないため各プロセスから一つのノードへアクセスが集中してしまう．この時の、gfs_pio_read, gfs_pio_set_view_global の所要時間が図 5 の左図である．一度目の実行で参照されたフラット画像は並列実行した全てのノードへ複製が作成され、2 度目の実行ではローカルアクセスが可能となる．この時の Gfarm API の所要時間が図 5 の右図である．この図から二つの Gfarm API の所要時間が短縮されている事がわかる．これらの Gfarm API は、実際にファイルの読み込みを行う API であり、ローカルファイルアクセスの実現により性能が向上している．その他の Gfarm API については大きな変化がみられないのは、2 回の実行で書き込み性能が変化していないためである．

このように、繰り返し参照される Gfarm ファイルについては複製を自動生成することで 2 回目以降の実行の効率を向上させることができる．

4.3 スクリプトによる実行

解析ツールを実行するには、gfrun コマンドによる実行が必要であることを述べた．しかし、図 3 に示した osmed の実行では各プロセスは 2 秒弱で終了しているものの、gfrun の実行終了までには 3 秒程度かかっている．gfrun は、スケジューリングなどを事前に行うためのコストがかかっているためである．一次処理では、4 つの処理が最低でも必要であり、このコストは大きなオーバーヘッドとなる．そこで、4 つの処理を連続して行うスクリプトを利用し、オーバーヘッドの削減を図る．一次処理手順はデータの性質が大幅に変化しない限り同一の手順を用いることができることが多いため、スクリプトの修正は頻繁には行う必要はない．

スクリプトの中では、Gfarm ファイルシステムでの exec コマンドに相当する gfexec コマンドを利用して、各解析ツールを実行した．図 6 はスクリプトによって連続実行された各解析処理の所要時間を示しており、4 つの処理の合計所要時間は 12 秒程度であった．スクリプトを実行する gfrun 全体の実行時間は約 16 秒であったのに対し、4 回の gfrun による実行では約 21 秒であったので、連続処理のスクリプト化は有効に機能したといえる．

5. 解析結果および考察

5.1 天文学的結果

2002 年 11 月から 12 月の間に W-C-RC フィルタを用いて行われた観測データから太陽系内小天体探索を行った．太陽系内小天体は太陽の周囲をケプラー運動しているため、連続的に撮像された画像中でその動きをみる事ができる．また、撮影された画像の露出時間が長い場合は 1 枚の画像中でも棒状に伸びて観測される事がある．

移動天体を検出するための方法はいくつか実装され

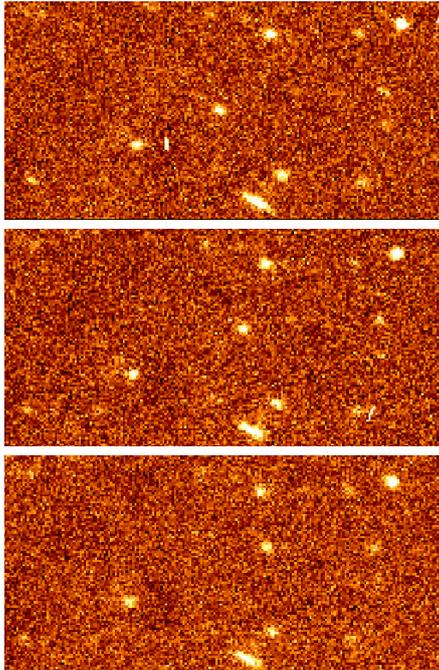


図 7 検出された移動天体

ているがここでは、もっともシンプルな方法を用いた。その手法は、画像からすべての天体をピックアップし、リストから移動している天体を抽出するというものである。前半の処理は、天体検出ツールである SExtractor を用いた。図 7 は検出された天体の例である。3 枚の画像は時間変化を表しており、中央左から右方向へ移動している天体が検出された天体である。

5.2 考察と今後の課題

グリッドデータファームの参照実装である Gfarm を用いて天文学データ解析環境の構築を行った。Gfarm で提供されているシステムコールのトラップ機能を用いる事により解析ツールの修正を行うことなく、想定していた全ての機能が動作する事が確認された。天文学データ解析においてグリッドデータファームが有効に機能し、大規模な解析を行う事ができるだけでなく、高い並列性によって解析の効率が向上することも示すことができた。また、コマンドによる解析ツールだけでなく、FITS ファイルブラウザである SAOImage ds9 の動作も確認できた事によって従来の解析環境が Gfarm ファイルシステム上で実現する事ができた。

さらに、グリッドデータファームの複製管理機能が有効に働き、フラット画像や、カタログデータなど繰り返し利用されるファイルへの高速なアクセスを行う事ができることが示された。これにより、全アーカイブデータを対象とするような大規模なデータ解析を行うためのインフラが整備され、今後より多くの天文学的成果が得られる事が期待される。

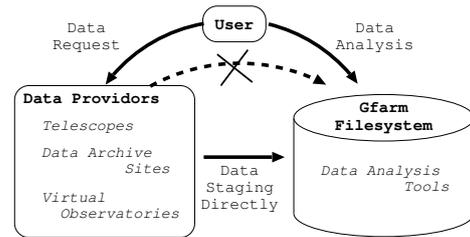


図 8 VO との関係によるデータフロー

一方で、解析環境を Gfarm ファイルシステム上に構築するためにはアプリケーションの修正が少なからず必要とされる。ソースコードやオブジェクトファイルなどが公開されているアプリケーションの場合はフック機能を利用することができるが、実行バイナリのみが配布されている場合には、フック機能を現在の Gfarm の実装では利用することができない。実行バイナリのみが配布されている場合への対応は現在検討中である。さらに、Gfarm ファイルシステムをマウントすることでアプリケーションの修正が一切必要無いような実装も検討中である。

また、Gfarm ファイルシステムを利用するためには 3 章で述べたように Gfarm ファイルシステムへ観測データを登録しなければならない。本研究におけるシステム構築では、すばる望遠鏡データアーカイブサイトである SMOKA から観測データをローカルにダウンロードした後、Gfarm ファイルシステムへ登録している。Gfarm を利用することで分散並列処理は実現されるものの、観測データの転送と配置がボトルネックとなり、トータルの実行効率を下げってしまう。この問題を解決するためには、図 8 に示したようにユーザーからのデータ請求に対して、データ提供者となりうる望遠鏡やアーカイブサイト、仮想天文台が直接 Gfarm ファイルシステムへ観測データを配置する構成が望ましい。日本の VO である JVO²¹⁾ と関係し図 8 の実現の可能性について今後検討していきたい。

6. まとめ

グリッドデータファームの参照実装である Gfarm を用いて天文学データ解析環境の構築を行い、その性能評価を行った。必要とされる解析ツール、FITS ファイルブラウザは全て Gfarm 上で動作が確認され、並列実行性能も確認できた。また、データ解析ツールのブレイクダウンにより 30 クラスタノード (60 CPUs) を用いた並列実行で読み込みおよび書き込みそれぞれ 5.9 GB/sec, 4.0 GB/sec のトータル I/O 性能が得られ、スケーラビリティも確認できた。今後は、さらに Gfarm ファイルシステムを容易に利用できるような Gfarm の実装を改良していく。また、より円滑な天文学データ解析環境の構築のために VO との密接な

連係を実現していきたい。

謝辞 本研究を遂行するにあたり貴重なご助言を頂いた Gfarm プロジェクトメンバ諸氏，グリッド研究センターのメンバ諸氏に感謝致します。特に，クラスタの改修，保守にご尽力頂いた清水敏行氏に感謝致します。本論文で用いた観測データは，国立天文台・天文学データ解析センターによってアーカイブされ公開されている観測データを利用致しました。

参 考 文 献

- 1) Kaifu, N., Usuda, T., Hayashi, S. et al.: The First Light of the Subaru Telescope: A New Infrared Image of the Orion Nebula, *PASJ*, Vol. 52, p. 1 (2000).
- 2) Komiyama, Y., Yagi, M., Miyazaki, S. et al.: High-Resolution Images of the Ring Nebula Taken with the Subaru Telescope, *PASJ*, Vol. 52, p. 93 (2000).
- 3) NAOJ: *Subaru Mitaka Okayama Kiso Archive system*. <http://smoka.nao.ac.jp/>.
- 4) 馬場 肇, 安田 直樹, 市川 伸一, 他: すばる望遠鏡公開データアーカイブシステムの開発, 国立天文台報, Vol. 6, pp. 23–36 (2002).
- 5) 山本 直孝, 野田 祥代, 多賀 正敏, 他: すばる望遠鏡公開データアーカイブシステムの開発 2, 国立天文台報, Vol. 6, pp. 79–100 (2003).
- 6) NASA: *HST Archive*. <http://hst.nao.ac.jp/>.
- 7) The Canadian Astronomy Data Centre (CADC): *The CFHT Archive*. <http://cadwww.dao.nrc.ca/cfht/>.
- 8) Sloan Foundation: *Sloan Digital Sky Survey*. <http://www.sdss.org/>.
- 9) International Virtual Observatory Alliance: *International Virtual Observatory Alliance*. <http://www.ivoa.net/>.
- 10) AIST: *Grid Datafarm*. <http://datafarm.apgrid.org/>.
- 11) 建部 修見, 森田 洋平, 松岡 聡, 関口 智嗣, 曾田 哲之: ペタバイトスケールデータインテンシブコンピューティングのための Grid Datafarm アーキテクチャ, 情報処理学会論文誌: ハイパフォーマンスコンピューティングシステム, Vol. 43, No. SIG 6 (HPS 5), pp. 184–195 (2002).
- 12) Tatebe, O., Morita, Y., Matsuoka, S., Soda, N. and Sekiguchi, S.: Grid Datafarm Architecture for Petascale Data Intensive Computing, *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2002)*, pp. 102–110 (2002).
- 13) Tatebe, O., Morita, Y., Matsuoka, S., Soda, N., Sato, H., Tanaka, Y., Sekiguchi, S., Watase, Y., Imori, M. and Kobayashi, T.: Grid Data Farm for Petascale Data Intensive Computing, Technical Report ETL-TR2001-4, Electrotechnical Laboratory (2001). <http://datafarm.apgrid.org/pdf/gfarm-ETL-TR2001-4.pdf>.
- 14) NASA: *Flexible Image Transport System (FITS)*. <http://fits.gsfc.nasa.gov/>.
- 15) Bertin, E. and Arnouts, S.: SExtractor: Software for source extraction, *Astronomy and Astrophysics Supplement*, Vol. 117, pp. 393–404 (1996).
- 16) SAO: *SAOImage DS9*. <http://hea-www.harvard.edu/RD/ds9/>.
- 17) Tatebe, O., Ogawa, H. et al.: The Second Trans-Pacific Grid Datafarm Testbed and Experiments for SC2003, *Proceedings of 2004 International Symposium on Applications and the Internet Workshops (SAINT 2004 Workshops)*, pp. 0–0 (2004).
- 18) 建部修見, 小川宏高, 児玉祐悦, 工藤知宏, 関口智嗣: グリッドデータファームと GNET-1 による日米間高速ファイル複製, 情報処理学会研究報告, 2004-HPC-97, HOKKE2004, pp. 31–36 (2004).
- 19) Monet, D., Bird, A., Canzian, B. et al.: The USNO-A2.0 Catalogue (Monet+ 1998), *VizieR On-line Data Catalog* (1998).
- 20) 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, pp. 185–190 (2003).
- 21) NAOJ: *Japanese Virtual Observatory*. <http://jvo.nao.ac.jp/>.