情報処理学会研究報告
IPSJ SIG Technical Report

e-サイエンス推進のための広域分散ファイルシステムの適用と評価

田中昌宏[†1]　建部修見[†1]

地理的に離れた複数組織の共同研究により進められる，大規模データの共有，解析などの e-サイエンスを推進するため，天文観測データの共有による e-サイエンスのケーススタディを行う．これまで天文分野では公開天文データ配信のための国際標準仕様「バーチャル天文台」の構築が行われてきたが，データアーカイブの大規模データ解析のためには課題が多い．本研究では，広域分散ファイルシステム Gfarm を用いた e-サイエンス基盤の提案を行い，その初期性能評価を行う．

Application and Evaluation of Large-Area Distributed File System for e-Science

Masahiro Tanaka[†1] and Osamu Tatebe [†1]

This paper discusses a way of astronomy data analysis to promote e-Science such as large-scale data sharing and analysis conducted in research collaboration with multiple distant organizations. While researchers in astronomy field have been constructed “Virtual Observatory”, which is the international standard for the distribution of public astronomical data, there remain open issues in the large-scale data analysis of data archives. In this paper, we propose an e-Science infrastructure using Gfarm large-area distributed file system, and evaluate its initial performance.

1. はじめに

地理的に分散した研究拠点の間で，それらの情報基盤をインターネットを通じて連携させ，強力な学術活動の研究推進基盤を構築することにより，e-サイエンスを推進する研究開発が行われている．各研究各分野においてその適用がされており，素粒子物理学の分野では，ILDG (International Lattice Data Grid)[1] と呼ばれるデータグリッドの例がある．その国内版である JLDG (Japan Lattice Data Grid)[1] は，広域分散グリッドファイルシステム Gfarm[2] を利用して運用している．

天文分野においても，その膨大な観測データを活かした研究を推進するためには，e-サイエンスの適用は不可欠である．近年，SDSS[3] や 2MASS[4] など，サーベイに特化した大規模な観測が実施され，大量の均質な観測データが得られるようになった．サーベイデータは，アーカイブされ，誰でも研究に利用できるように公開されている．このデータアーカイブを活用した研究は，天文の新しい分野の一つになりつつある．データアーカイブの活用により進展が期待される研究の例として，銀河団の統計的な研究など，多くの天体サンプルを必要とする研究の他，低温度のため可視光では見えない褐色矮星，トーラス雲に隠された活動銀河核，遠方の銀河など，あらかじめ空のどの座標にあるか分からない天体の発見などがある．

こうしたアーカイブされた天文観測データを有効利用するため，バーチャル天文台 (Virtual Observatory，VO) の研究開発が進められてきた．これは天文分野における e-サイエンスの一つである．バーチャル天文台とは，天文データの配信および検索のための国際標準仕様である．この仕様に基づくシステムにより，世界各地で分散配信される天文データの発見・利用を促進する．各国のバーチャル天文台組織が参加する IVOA (International Virtual Observatory Alliance)[5] において，この国際標準仕様の策定が行われている．

図 1 にバーチャル天文台による典型的なデータアクセスの例を簡単に模式的に示す．この図では，国立天文台が開発・運用している JVO (Japanese Virtual Observatory)[6] を利用する場合について示している．ユーザは，JVO ポータルサーバ[7] にログインすれば，バーチャル天文台の仕様を知らなくても Web ブラウザから操作できる．JVO ポータルサーバは，IVOA 標準仕様に基づく VO プロトコルにより，バーチャル天文台のデータ配信サービスからデータを取得する．以降はこの IVOA 標準仕様に基づくデータアクセスの概要について述べる．まず目的のデータがどのサーバから配信されているかを見つけるため，IVOA によって仕様が定められた Resource Metadata[8] を利用する．Resource Metadata は，どのサーバがどのようなサービスでどのような内容のデータを配信しているかが記されている．Resource Metadata は XML で記述され，スキーマによって仕様が定められている．VO サービスはこのリソースメタデータを公開することにより，利用者はそのサービスを発見し

†1 筑波大学
University of Tsukuba

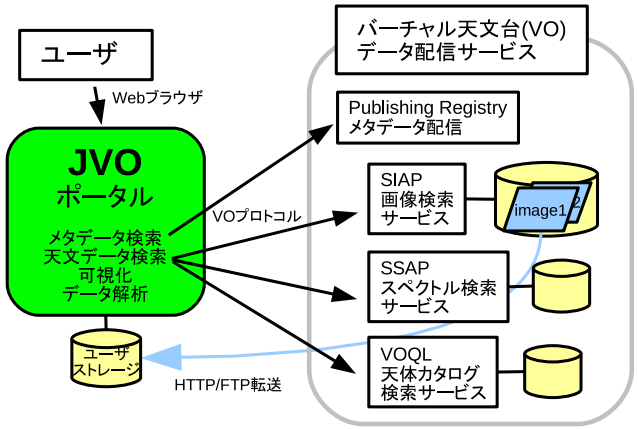

図 1 バーチャル天文台のデータアクセス.
Fig. 1 Data Access to Virtual Observatory.

やすくなる．このメタデータを配信するための規格として，もとは図書館の図書情報のメタデータを配信するための規格である OAI-PMH[9] (Open Archives Initiative - Protocol for Metadata Harvesting) を用いることが IVOA では合意されている．この OAI-PMH によって天文メタデータを配信するサーバを Publishing Registry と呼ぶ．世界各地の Publishing Registry からメタデータを収集し，検索することによって，目的のデータを配信するサーバと取得方法が分かるというわけである．配信サーバが見つかれば，IVOA 標準のプロトコルでアクセスすることにより，データを検索し取得することができる．現在運用されている VO プロトコルとして，SIAP[10] (Simple Image Access Protocol)，SSAP[11] (Simple Spetrum Access Protocol)，ADQL[12] (Astronomical Data Query Language) がある．SIAP は画像，SSAP はスペクトルを取得するためのプロトコルである．これらに検索パラメータに座標などの条件を与えると，検索条件に合う画像ファイルへの URL のリストが返される．クライアントはその URL からデータを HTTP または FTP で転送する．一方，ADQL は，SQL に基づく天文検索言語である．天体カタログの中から座標や明るさなどの条件を与えれば VOTable という仕様の XML フォーマットで天体のリストが得られる．

バーチャル天文台は，これまでに公開天文データを配信する仕組みにおいては標準化に成功している．しかしデータを集めた後の解析処理については，バーチャル天文台には標準の仕組みはない．膨大な天文データを扱う処理では，処理能力が大きく，かつ，処理ソフトウェアが動作する計算機が必要であるから，複数の計算機連携が必要となる．解析処理の計算機連携を実現する簡便な手法の 1 つに，解析処理を SOAP ベースの Web サービスで公開する手法がある．しかし，画像のように大きなデータを転送する際に SOAP メッセージに乗せることは非効率であるという問題，および，ユーザ管理機構がなくジョブの制御などが出来ないという問題がある．これらの問題に対する 1 つの解がグリッドである．天文分野におけるグリッド利用の例として，米国の NVO[13] において TeraGrid[14] を利用して天文データ処理を行った例[15] がある．ヨーロッパの Euro-VO[16] では，EGEE[17] によりバーチャル天文台を構築する試み[18] がある．このように各国のバーチャル天文台では，独自に既存のミドルウェアを利用してグリッドを構築している．

まとめると，大規模データアーカイブの大規模データ解析のため，天文分野ではこれまで

- バーチャル天文台による天文データアーカイブ配信の国際標準化
- 天文データ解析へのグリッドの適用

が行われてきた．

一方，解決されていない課題の 1 つに，

- 広域分散データ解析のための天文データ共有およびストレージ方法

がある．グリッドで天文データを解析する場合には，グリッドのストレージシステムにデータを移動する必要がある．グリッドから利用できるストレージシステムには，TeraGrid の SRB[19]，EGEE の gLite[20] がある．例えば 15) のデータの解析には SRB を用いている．しかし，データ量が膨大になると，ストレージシステムにデータを準備することが研究者にとってはハードルとなる．バーチャル天文台はデータ配信のみ行うため，解析システムへのデータ移動はユーザが行う必要がある．また，解析するデータ量が膨大になり，1 つの拠点では計算機資源が不足する場合は，複数の拠点に処理を分散する必要が出る可能性がある．そのような広域分散データ解析は，ファイル配置などの課題を克服して初めて実現できる．

そこで，本研究では，大規模天文データの広域分散解析を実現するため，広域分散ファイルシステム Gfarm[2] を用いた e-サイエンス基盤の提案を行う．また，この基盤においてバーチャル天文台が配信する公開天文データを分散共有する方法についても考察する．この e-サイエンス基盤が実現することにより，天文データアーカイブの大規模データ処理が実現し，天文分野における新しい一分野が開拓されることが期待される．

本提案によるファイルシステムの設計，共有データの格納手法およびファイルシステムの検索方法について 2 節で述べる．提案による基盤構築の初期設計として，Gfarm 上で天文データ処理の一例について性能評価を行った結果について 3 節で述べる．

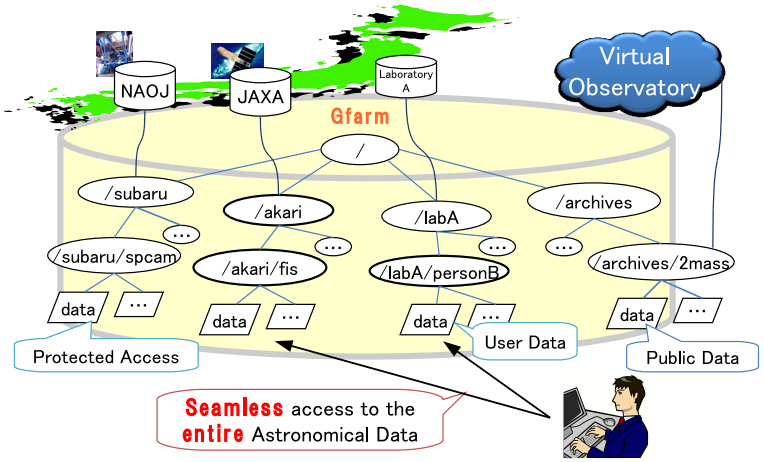

図 2 Gfarm を利用した広域天文データ共有.
Fig. 2 Wide-Area Sharing of Astronomy Data using Gfarm.

2. 天文データの広域共有の提案

2.1 共有データの格納

図 2 に本提案を模式的に示す．天文データを格納するストレージは天文研究者の所属する研究機関に分散配置される．Gfarm の適用によりこれらを 1 つのファイルシステムとする．これにより天文研究者は各地の計算機から共有データへシームレスにアクセスできるようになる．Gfarm ファイルシステムに格納するデータとして，次の 3 種類を想定している．

2.1.1 観測所データ

観測データ配布の例として，国立天文台が運営するハワイ山頂に設置されたすばる望遠鏡の例を紹介する．すばる望遠鏡によって観測したデータは，ハワイ島ヒロのすばる望遠鏡山麓施設に設置されている STARS[21] および三鷹本部に設置されている MASTARS というシステムが保管，管理している．観測提案者は STARS/MASTARS のアカウントを通じて観測データをダウンロードすることができる．このような観測データを，広域分散ファイルシステムで配布すれば，観測者への配布が容易になる．

観測データを Gfarm に格納する利点として，グリッドによる統一的なアカウントを利用できる点がある．これにより個別のアカウント管理が不要となる．すばる望遠鏡の例を述べると，観測したデータは，観測から 18 か月間は観測提案者に専有権が認められているが，その後は一般公開される．現状のアクセス制限方法は，STARS/MASTARS へのアカウントにより行われている．一方，上記の Gfarm ファイルシステムに置く案では，Gfarm のユーザ・グループによるアクセス制限を行う．JLDG と同じように VOMS (Virtual Organization Membership Service) によってアクセスするユーザを管理し，グループ単位の管理も可能である．これにより個々の観測所のポリシーに応じた運用が可能になる．

表 1 代表的な天文画像データアーカイブ.
Table 1 Performance of Table Partitioning.

アーカイブ名	バンド数	ファイル数	ファイルサイズ	データ量
2MASS	3	4,121,439	2 MB	11.4 TB
SDSS DR7	5		6 MB	15.7 TB
IRAS	4	6,880	1 MB	6.9 GB

2.1.2 ユーザデータ

観測したデータを解析して科学成果を出すことは，観測提案者の仕事である．その解析作業では，一般に観測した生データの何倍，何十倍ものストレージ領域が必要になる．個人でそのようなストレージを確保することは容易ではない．そうしたユーザの解析領域としても広域分散ファイルシステムは有効である．

研究者個人や研究グループのデータを置く場合についても，広域分散ファイルシステムを利用するメリットがある．天文においても，他の分野と同じように，同じ研究テーマを持つグループに属する個人が各地の異なる研究機関に属することが多い．一方，各地に分散する研究グループ内でデータを共有したい場合もある．その場合に広域ファイルシステムを利用すればデータ共有が容易になる．

2.1.3 バーチャル天文台データ

表 1 に，バーチャル天文台から配信される代表的な画像データアーカイブについてのデータ量を示す．他にもギガバイトクラスの天文データアーカイブが数多く存在する．こうしたバーチャル天文台によって配信されるデータは，基本的にはアクセス制限が不要な公開データである．バーチャル天文台という統一規格に基づく天文アーカイブデータの配信は今後ますます増えていくと予想される．一方，1 節で述べたように，バーチャル天文台におけるデータ転送は必ずしも効率的ではなく，天文データを解析システムへ移動する問題も解決されていないという問題がある．こうしたの問題を解決するため，バーチャル天文台のデータ配信として広域分散ファイルシステムを適用する．

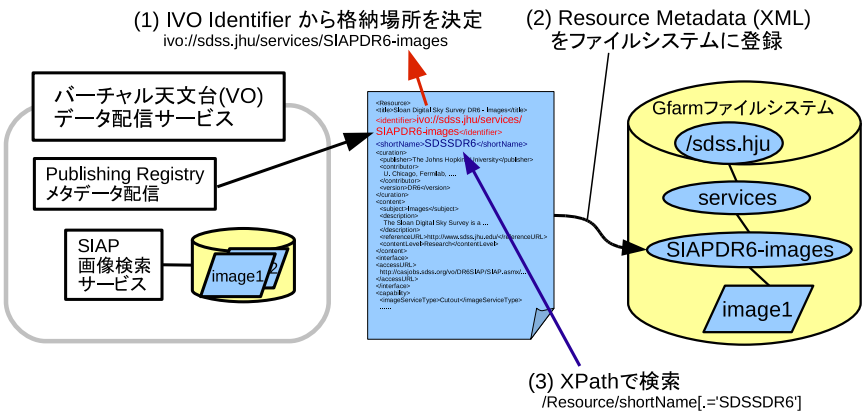

図 3 天文データ格納ディレクトリと検索.
Fig. 3 Directory for Storing Astronomical Data

バーチャル天文台からのデータを Gfarm に格納するには，1 つの VO サービスについて，Gfarm にディレクトリを 1 つ作成する．そのディレクトリ以下に配信データを格納する，ディレクトリの名前の決定するため，IVOA 標準仕様の IVOA Identifiers を用いる．IVOA Identifiers は，サービスなどのリソースを識別するための ID として使用され，Resource Metadata に記述されている (**図 3** の (1))．IVOA Identifiers は，`ivo://` で始まる URI と定められている．例えば，JVO では `ivo://jvo/service-name` などを用いているが，その名前の付け方には特に取り決めはない．しかし ID が衝突しないように，最初は組織が分かる名前とすることが慣例である．そこで，この IVOA Identifiers の本体を Gfarm ファイルシステムのディレクトリ名とする．これによってバーチャル天文台とファイルシステムとの対応付けを行う．

2.2 広域分散データ解析における利用

以上のように天文データを Gfarm に格納することにより，Gfarm ファイルシステムをマウントするだけで，ファイルアクセスが可能になる．天文データ解析を行う場合，解析システムがファイルシステムをマウントすれば，Gfarm に格納された天文データを入力データとして利用できる．利用頻度が高いデータについては，ネットワーク的に近いノードに複製配置すれば，効率の良いファイルアクセスが可能になる．さらに，提案基盤は，解析結果の出力データの格納場所としても利用できる．天文データ解析では，処理途中に一時的に出力される中間データの量が多い．特に大規模データ解析を行う場合は，解析結果の中間データや出力データが 1 拠点のストレージでは不足することもあり得る．そのような場合でも複数拠点のストレージに出力データを分散することにより，大規模データ解析が可能となる．また，処理速度を高めるため，複数拠点の解析システムを利用して分散処理する場合でも，同一のファイルシステムをマウントすることができるから，データ共有が容易にできる．

2.3 検索方式

2.3.1 ディレクトリ検索

バーチャル天文台では，現在数千ものサービスが利用でき，今後増えることが予想される．数多くのディレクトリから目的のものを発見するため，本提案では，IVOA 標準の Resource Metadata を利用する．Resource Metadata は，1 節で述べたように，Publishing Registry から配信され，VO サービスを検索するために用いられる．一方，Gfarm ファイルシステムでは，メタデータを XML として格納し，XPath で検索することができる．そこで，VO サービスに対応したディレクトリに，Resource Metadata を格納する (図 3 (2))．このメタデータについて XPath などで検索することにより (図 3 (3))，バーチャル天文台におけるサービス検索と同様に，Gfarm ファイルシステム内で目的のディレクトリを発見することができる．

2.3.2 ファイル検索

表 1 にあるように，2MASS の画像データとして 4 百万ものファイルが存在する．そのような大量のファイルをファイルシステムに格納することを想定している．そこで，格納されたファイルを検索する仕組みが必要になる．画像を検索する場合に必要とされるのは座標検索である．座標検索は XPath だけではできないため，VO 標準プロトコルの SIAP を併用する方法を提案する．

ここで SIAP による画像検索について簡単に説明する．SIAP は HTTP または Web サービスでアクセスを受けつける．HTTP による SIAP 検索の基本例を次に示す．

```
http://vo.org/cgi-bin/VOimq?POS=180.567,-30.45&SIZE=0.0125
```

この例では，デフォルトの赤道座標で赤経 18.567 度，赤緯 −30.45 度を中心として半径 0.0125 度以内にある画像を検索する．この URL へアクセスすると，検索にマッチした画像があれば，そのファイルへの URL のリストを，VOTable という XML のテーブルとして返す．

この SIAP 仕様に基づく「ファイルシステム検索サービス」の仕組みを**図 4** に模式的に示す．ユーザはまず「ファイルシステム検索サービス」に対して座標などをキーにして画像

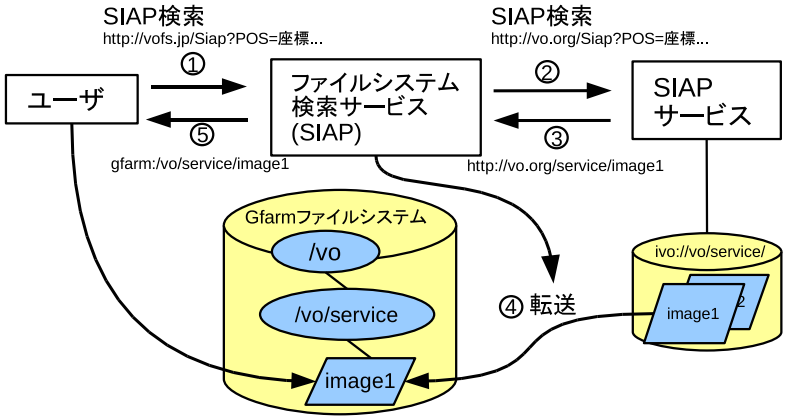

図 4 SIAP プロキシによるファイルシステム検索.
Fig. 4 File System Search using SIAP Proxy.

を問い合わせる．この「ファイルシステム検索サービス」は，データ配信元の SIAP サービスへのプロキシといえる．問い合わせをそのまま元の SIAP サービスへ送り，その問い合わせにマッチした画像への URL を得る．もしこの URL のデータがすでにファイルシステムに存在すれば，そのファイルパスをユーザに返す．もし無い場合は，URL から Gfarm ファイルシステムにデータを転送し，そのファイルパスをユーザに返す．ユーザからの問い合わせるプロトコルを SIAP とすれば，バーチャル天文台と互換性があるものとなる．

この方法には，ファイルシステムの他に別途「ファイルシステム検索サービス」の設置が必要になるというデメリットがある．しかし，このような仕組みを導入すれば，ファイルシステム内のファイルに対して座標検索が可能になるだけでなく，必要になった段階で転送することにより，大量のデータを格納するストレージをあらかじめ用意しなくてもよい，というメリットがある．一方，アクセス頻度が高いと予想されるデータについては，あらかじめ転送しておけば，利用者はすぐにデータを利用することができる．

3. 性 能 評 価

前節で提案した e-サイエンスの基盤を天文研究に活用できることを示すため，実際に Gfarm 上のファイルに対して天文データ処理を実施し初期性能評価を行う．天文データ処理の代表例として，ここでは，Montage[22] という画像処理ソフトウェアを用いた．Montage は，複数の天文画像を合成して 1 枚の画像にする（モザイキング）という汎用的な処理を行うオープンソースのソフトウェアである．入出力画像のフォーマットは，天文において標準の FITS 形式[23] である．FITS は，天球座標系などの情報がヘッダ部に記述されていることが特徴である．Montage は，FITS データの読み書きに CFITSIO[24] という天文で広く使われているライブラリを用いている．Montage のモザイキング処理はいくつかの処理から成る．最初に行う処理は，入力画像を出力画像の座標系へ投影する処理である．この投影処理を 1 枚の画像について行うプログラムが mProjectPP である．mProjectPP は，複数の画像に対して並行処理が可能である．また，この投影処理は，モザイキング処理全体のうち，半分以上を占める部分である[25]．そこで，天文データ処理の代表として mProjectPP を Gfarm 上で並列に実行し，その実行時間を測定した．

3.1 測 定 環 境

地理的に分散した広域ファイルシステムの性能を測定するため，InTrigger プラットフォーム[26] を用いた．この測定では，InTrigger ノードのうち 広島と筑波のノードを用いた．これらのノードはすべて 1 ノード当たり 8 コアの CPU を搭載している．Gfarm のメタデータサーバは筑波ノード上で動作する．メタデータサーバへの RTT (Round-Trip Time) は，筑波ノードからは 0.15 ms，広島ノードからは 29 ms である．

測定に用いた Gfarm のバージョンは 2.2.1 である．FUSE を用いて実装された gfarm2fs により，Gfarm ファイルシステムをマウントし，通常のファイルシステムとしてアクセスした．gfarm2fs はシングルスレッド動作するため，同時に 1 つのマウントポイントにアクセスすると性能が落ちる．そこで，1 つのノード内で並行実行する際は，並列プロセスの数だけマウントポイントを用意し，プロセス毎に異なるマウントポイントにアクセスした．比較対象として，クラスタ内で NFS マウントされているストレージ，および，実行ノード上のローカルストレージにファイルを置いた場合についても同様に処理を行い，実行時間を測定した．

3.2 実行時間の測定

3.2.1 測 定 内 容

mProjectPP は，入力ファイルとして，画像ファイル 1 つと出力画像の投影パラメータが記述された設定ファイル 1 つを読み込み，出力ファイルとして，投影結果の画像ファイル 1 つと，入力画像が投影されたエリアを表す画像ファイル 1 つを書き出す．入力画像は，公開されている 2MASS の FITS 画像データ 32 枚を用いた．1 枚のファイルサイズは約 2 MB である．

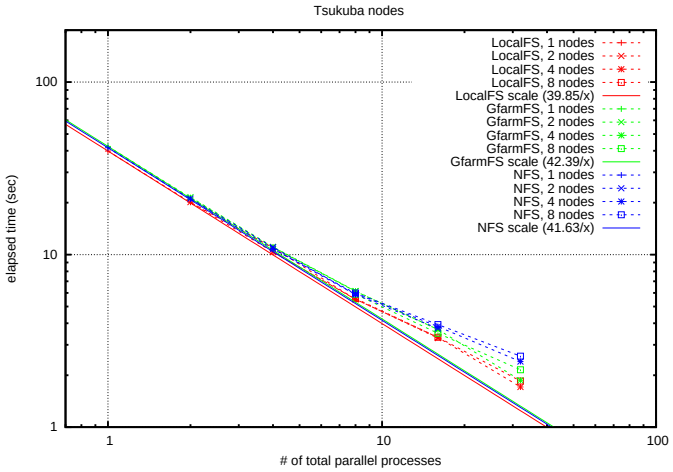

図 5 mProjectPP 実行時間の比較（筑波ノード）.
Fig. 5 Comparison of Elapsed Time of mProjectPP (Tsukuba nodes).

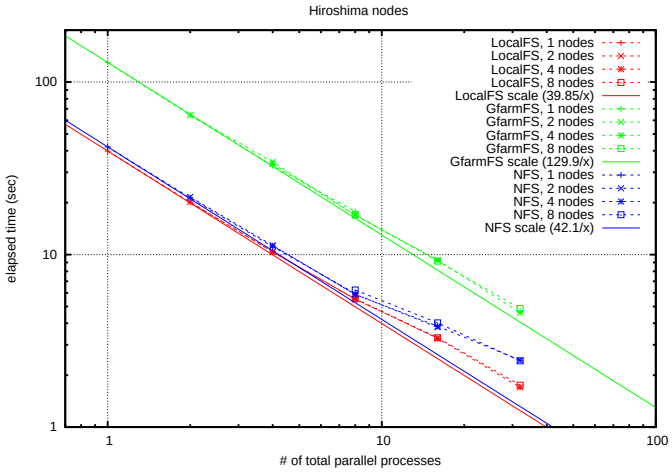

図 6 mProjectPP 実行時間の比較（広島ノード）.
Fig. 6 Comparison of Elapsed Time of mProjectPP (Hiroshima nodes).

ノード数は 1，2，4，8 ノード，1 ノード当たりの並列プロセス数は 1，2，4，8 プロセスと変化させ，画像の枚数 32 を並列プロセス数で割った数だけ逐次に mProjectPP を実行する．4 ノード × 8 プロセスおよび 8 ノード × 4 プロセスの場合は，32 枚の画像について完全に並列処理が行われる．1 ノード × 1 プロセスの場合は，32 枚の画像について逐次的に処理が行われる．

Gfarm ファイルシステム (GfarmFS) の測定の場合，入力画像ファイルはあらかじめ gfreg を用いて実行ノード上のファイルサーバに登録した．したがって，Gfarm ファイルシステムの各ファイルへのアクセスはネットワークを経由していない．NFS の測定の場合，すべて同一クラスタ内の別のファイルサーバへのネットワーク経由のアクセスである．ローカルファイルシステム (LocalFS) の測定の場合，実行ノード上のローカルストレージへのアクセスとなるため，理想的な場合についての測定となる．通常，プログラム実行前のステージングにより共有ストレージからローカルストレージに移動する必要があるが，この測定にはその処理は含まれていないことに注意が必要である．

3.2.2 結　　果

GfarmFS，LocalFS，NFS のファイルに対して mProjectPP を実行し，経過時間を計測した結果を，**図 5** および**図 6** に示す．筑波ノードにおける結果を図 5 に，広島ノードにおける結果を図 6 に示す．両図とも，並列プロセス数に対する経過時間を表す対数プロットであり，緑のマークが GfarmFS，赤のマークが LocalFS，青のマークが NFS を示す．右下がりの実線の直線は，1 ノード × 1 プロセスの場合の処理時間から，並列プロセス数で割ったグラフを表す．LocalFS，NFS，GfarmFS いずれの結果も 8 並列あたりまでこの直線にほぼ乗っており，プロセス数に比例して実行時間が短縮されていることが分かる．16，32 並列では若干実線より上にずれている．実行時間はいずれも並列数に依存しており，ノード内で並列にしたかノードに分散したかの違いはみられない．

メタデータサーバと同じクラスタ内の筑波ノードでの結果 (図 5) は，LocalFS，NFS，GfarmFS の処理時間は相対的にほとんど差がないことを示している．詳細に見ると，1 プロセスで逐次処理した GfarmFS の実行時間は，LocalFS に対して約 6 % の増加，NFS に対して約 2 % の増加である．一方，32 並列実行の場合の GfarmFS の実行時間は，LocalFS に対して約 9 % の増加，NFS に対しては逆転して約 8 % の減少となった．Gfarm の測定の場合は実行ノード上にファイルを置き，NFS の場合は別のノードのファイルにアクセスしたというアドバンテージがある．Gfarm は複数のファイルシステムを束ねて 1 つのファイ

表 2 mProjectPP の入出力関数の経過時間.
Table 2 Elapsed time of IO functions in mProjectPP.

		長短時間の合計	関数の経過時間 (μs) 広島ノード		筑波ノード		広島/筑波
関数	呼出回数 /	呼出回数	合計	平均	合計	平均	の比
fopen	14 /	14	1,150,978	82,213	9,555	683	120.46
fseeko	4 /	14	419,922	104,981	3,211	803	130.78
ftello	3 /	6	89,031	29,677	666	222	133.68
fgets	4 /	64	443,287	110,822	3,823	956	115.95
fread	4 /	744	299,200	74,800	5,296	1,324	56.50
fwrite	2 /	2,968	236,883	118,442	1,467	733	161.47
remove	2 /	2	161,651	80,825	1,104	552	146.42
長時間 計	33 /	3,823	2,800,952	84,877	25,122	761	111.49
短時間 計	3,790 /	3,823	105,049	28	103,306	27	1.02

ルシステムとするものであるから，このような運用が可能である．実際の運用でも，Gfarm 上で適切にファイル配置を行うことにより，NFS よりもよい性能が得られるということをこの結果は示している．

一方，メタデータサーバから地理的に遠い広島ノードでは (図 6)，GfarmFS は LocalFS に比べて実行時間が約 3 倍になっている．どちらもローカルディスクへのアクセスであるから，データ転送とは別の処理に時間がかかっていることになる．

3.3 入出力関数の経過時間

メタデータサーバから遠い広島ノードにおいて，Gfarm 上での実行時間が増えた原因を探るため，入出力関数の呼び出しにかかる時間を計測した．計測のため，mProjectPP および CFITSIO のソースコードの入出力関数の前後に，経過時間をログに出力するマクロを挿入した．実行環境は，前出の InTrigger 筑波ノードおよび広島ノードである．Gfarm ファイルシステムにファイルを配置し，そのノードで mProjectPP を 1 回実行し，入出力関数の経過時間について集計した．この計測の際の mProjectPP 全体の実行時間は，広島ノードでは 4.59 s，筑波ノードでは 1.82 s である．

計測結果を**表 2** に示す．ここにリストした fopen から remove までは，次の条件を満たす関数経過時間の集計である：

- 関数 1 回の経過時間が，広島ノードで 1 ms 以上，かつ，広島ノードは筑波ノードに対して 2 倍以上

便宜的にこれらを「長時間」関数と呼ぶ．一方，表 2 の一番下の行は，それ以外の「短時間」関数について集計した．

表 2 をみると，各「長時間」関数 1 回の経過時間の平均は，広島ノードでは 30–118 ms であるのに対し，筑波ノードでは 0.2–1.3 ms であり，両者の比は 56–161 である．「長時間」関数の経過時間の合計は，広島ノードでは 2.80 s であり，これは mProjectPP 実行時間 4.59 s の 61 % にも達する．残りの時間は 1.79 s であり，ローカルファイルシステムに対する実行時間とほぼ同じである．一方，筑波ノードでの同じ関数の経過時間は 25 ms であり，mProjectPP 実行時間の 1.4 % である．このことから，広島ノードでは，特定の入出力関数の経過時間の増加が，実行時間の増加につながったことが分かる．

一方，表 2 最下行の「短時間」関数の集計では，関数の経過時間の合計はどちらのノードも約 100 ms とほぼ同じであった．

広島ノードにおける「長時間」関数 1 回の経過時間 30–118 ms は，広島筑波間の RTT 29 ms の 1–4 倍である．Gfarm のファイルシステムメタデータの操作は RTT の数倍かかる[27] から，表 2 の「長時間」関数の処理には，メタデータサーバへのアクセスが伴っていると考えられる．また，関数経過時間の筑波ノードに対する広島ノードの比は，fread 関数を除いて 116–161 である．この比は，それぞれのクラスタから筑波にあるメタデータサーバまでの RTT の比 29 ms / 0.15 ms = 193 より小さいが半分よりは大きい．メタデータサーバへの RTT の差が，関数経過時間の差に相当している．

表 2 を見ると，fopen 関数が 14 回呼ばれているが，14 回とも「長時間」に分類されている．ファイルシステムからファイル名を削除する remove 関数も，2 回とも「長時間」に分類されている．これらの関数は常にメタデータサーバへのアクセスが発生している．対照的に fclose 関数は「長時間」に分類されていない．他方，fseeko, ftello, fgets, fread, fwrite の各関数は,「短時間」を含めた合計回数のうち，2–4 回のみ「長時間」に分類されている．これらの関数には，ファイルオープンの直後はメタデータアクセスが発生するが，それ以降の呼び出しでは発生しない，という特徴がある．

mProjectPP は 2 つの入力ファイルを読み，2 つの出力ファイルに書き出すから，open に必要な回数は 4 回である．表 2 にある fopen 関数の呼び出し回数 14 回というのは，この必要回数より 10 回多い．ソフトウェアの設計を見直し，open 回数を 4 回に減らすことができれば，fopen だけでなく fread などの関数でもメタデータアクセスの減少が期待できるから,「長時間」関数の経過時間の合計 2.8 s を，14 分の 4 倍の 0.8 s 程度までに減らすことができると予想される．Montage のソースを見ると，mProjectPP で fopen で テンプレートファイルを開いており，さらに CFITSIO の関数経由でもテンプレートファイルと

入力画像にアクセスしている．さらに CFITSIO では，最初の open の際，FITS ヘッダからキーワードを得る際，および FITS 画像データを読む際にそれぞれ open-close の操作を行っている．このようなソフトウェアの仕様が，実行時間が増加した一因である．

本性能評価によって，天文データ処理では，メタデータサーバまでの RTT 距離によって実行時間に影響を与えるということが分かった．特に CFITSIO のような天文で広く使われているライブラリの仕様がその影響を大きくしていることも分かった．ソフトウェア設計以外の対策として，メタデータサーバを実行ノードの近く，特にクラスタ内に設置することにより，実行時間を短くすることができる．現状では Gfarm ファイルシステム 1 つにつきに 1 台しかメタデータサーバを持てず，別のファイルシステムとしなければならない．そこで，Gfarm で複数のメタデータサーバを設置可能にし，近くにメタデータサーバを設置することにより，効率的なファイルアクセスを行う研究[28] が進められている．これが実現すれば，クラスタ内部にメタデータサーバを設置することにより，Gfarm 上のファイルに対してもでも十分な性能の処理を行うことができる．

4. おわりに

天文研究者の間で天文観測データを広域共有するため，e-サイエンス推進のための研究基盤の 1 つである広域分散グリッドファイルシステム Gfarm を適用する方法について考察した．観測所が配布する観測データや，ユーザ・研究グループが所有するデータだけでなく，公開天文データ流通のための国際共通仕様であるバーチャル天文台が配信するデータまでもを 1 つのファイルシステムからアクセスする手法について提案した．特にファイル検索の手法として，SIAP へのプロキシを設置する方法を提案した．さらに，Gfarm 上で天文データ処理を行う際の性能を評価するため，Montage の実行時間を測定した．その結果，メタデータサーバからの RTT 距離が近い場合についてはローカルディスクや NFS とほぼ同等の性能が得られることが分かった．一方，メタデータサーバから遠い場合については，ファイルオープンの回数が最適化してない場合に処理時間が余分にかかることが分かった．今後は，その他の天文データ処理についても広く実行性能を評価しつつ，提案した天文データ共有環境を構築し，実際の天文データ処理へ適用して評価する計画である．

謝辞 本研究は，文部科学省の科学技術試験研究委託事業による委託業務：次世代ＩＴ基盤構築のための研究開発「e-サイエンス実現のためのシステム統合・連携ソフトウェアの研究開発」における研究課題「研究コミュニティ形成のための資源連携技術に関する研究」（データ共有技術に関する研究）の支援を受けています．また本研究は，科学技術研究費特定領域 「情報爆発時代に向けた新しい IT 基盤技術の研究」において構築された研究用プラットフォーム InTrigger を利用しました．謹んで感謝の意を表します．

参 考 文 献

1) 佐藤三久，建部修見，吉江友照，石井理修， 朴泰祐， 宇川彰：計算素粒子物理学分野の国際データグリッド ILDG と国内グリッド JLDG，情報処理学会研究報告 2007-HPC-113，pp.13–18 (2007).
2) Gfarm: http://datafarm.apgrid.org/.
3) SDSS: Sloan Digital Sky Survey, http://www.sdss.org/.
4) Skrutskie, M.F., Cutri, R.M., Stiening, R., Weinberg, M.D. et al.: The Two Micron All Sky Survey (2MASS), *Astronomical Journal*, Vol.131, pp.1163–1183 (2006).
5) IVOA: International Virtual Observatory Alliance, http://www.ivoa.net/.
6) Ohishi, M., Shirasaki, Y., Tanaka, M., Honda, S. et al.: Development of Japanese Virtual Observatory (JVO) : Experience on Interoperation with other Virtual Observatories and its Future Plan, *Astronomical Data Analysis Software and Systems XV* (Gabriel, C., Arviset, C., Ponz, D. and Enrique, S., eds.), Astronomical Society of the Pacific Conference Series, Vol.351, p.375 (2006).
7) JVO portal: http://jvo.nao.ac.jp/portal/.
8) Resource Metadata: http://www.ivoa.net/Documents/latest/RM.html.
9) OAH-PMH: Open Archives Initiative Protocol for Metadata Harvesting, http://www.openarchives.org/pmh/.
10) SIA: Simple Image Access, http://www.ivoa.net/Documents/latest/SIA.html.
11) SSA: Simple Spectral Access, http://www.ivoa.net/Documents/latest/SSA.html.
12) ADQL: Astronomical Data Query Language, http://www.ivoa.net/Documents/latest/ADQL.html.
13) NVO: US National Virtual Observatory, http://www.us-vo.org/.
14) TeraGrid: http://www.teragrid.org/.
15) Laity, A., Berriman, G. B., Good, J. C. K. D. S., Jacob, J. C. B. L., Moore, R.W.R.D.E., Singh, G. and Su, M.-H.: An All-Sky 2MASS Mosaic Constructed on the TeraGrid, *Astronomical Data Analysis Software and Systems XVI* (Shaw, R.A., Hill, F. and Bell, D.J., eds.), Astronomical Society of the Pacific Conference Series, Vol.376, pp.65–+ (2007).
16) Euro-VO: European Virtual Observatory, http://www.euro-vo.org/.
17) EGEE: Enabling Grids for E-sciencE, http://www.eu-egee.org/.
18) Taffoni, G., Vuerli, C., Pasian, F. and Rixon, G.: Bridging VO and Computational Grid Applications within Euro-VO, *Astronomical Data Analysis Software and Systems XVII* (Argyle, R.W., Bunclark, P.S. and Lewis, J.R., eds.), Astronomical

Society of the Pacific Conference Series, Vol.394, pp.289–+ (2008).
19) SRB: Storage Resource Broker, http://www.sdsc.edu/srb/index.php/Main_Page.
20) gLite: http://glite.web.cern.ch/glite/.
21) Takata, T., Ogasawara, R., Kosugi, G., Mizumoto, Y. et al.: STARS (Subaru Telescope archive system) for the effective return from Subaru Telescope, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (Quinn, P.J., ed.), Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, Vol.4010, pp.181–189 (2000).
22) Montage: http://montage.ipac.caltech.edu/.
23) FITS: http://fits.gsfc.nasa.gov/.
24) FITSIO: http://heasarc.gsfc.nasa.gov/fitsio/.
25) Katz, D.S., Anagnostou, N., Berriman, G.B., Deelman, E. et al.: Astronomical Image Mosaicking on a Grid: Initial Experiences, *Engineering the Grid - Status and Perspective* (Martino, B.D., Dongarra, J., Hoisie, A., Yang, L. and Zima, H., eds.), American Scientific Publishers (2006).
26) 斎藤秀雄，鴨志田良和，澤井省吾，　弘中健，　高橋慧，関谷岳史，　頓楠，柴田剛志，横山大作，田浦健次朗：InTrigger: 柔軟な構成変化を考慮した多拠点に渡る分散計算機環境，情報処理学会研究報告 2007-HPC-111，pp.237–242 (2007).
27) 建部修見，曽田哲之：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).
28) 平賀弘平，建部修見：広域ファイルシステムにおける分散メタデータサーバの検討，情報処理学会研究報告 2009-HPC-119，pp.139–144 (2009).