DEWS2007 D8-9

天文データベースと連携した天文学研究用解析システムの構築

白崎　裕治[†]　田中　昌宏[†]　川野元　聡[†]　本田　敏志[†]　大石　雅寿[†]
水本　好彦[†]　大江　将史[†]　増永　良文[††]　安田　直樹[,†††]　石原　康秀[††††]
堤　純平[††††]　中本　啓之[†††††]　小林　佑介[†††††]　坂本　道人[†††††]

† 国立天文台　〒 181-8588 東京都三鷹市大沢 2-21-1
†† お茶の水女子大学　〒 112-8610 東京都文京区大塚 2-1-1
††† 東京大学 宇宙線研究所　〒 277-8582 千葉県柏市柏の葉 5-1-5
†††† 富士通株式会社　〒 261-8588 千葉県千葉市美浜区中瀬 1-9-3
††††† 株式会社セック　〒 150-0031 東京都渋谷区桜丘町 22-14 NES ビル

あらまし　天文観測データは観測装置の性能向上にともないデータ量も爆発的に増加を続けている。データ量が膨大であると，従来のように個人のコンピュータへデータを転送して解析を行うことは困難である。すばる望遠鏡の観測装置の一つである SuprimeCam はこれまでに 6TB のデータを生産しており、これを研究者個人の計算機環境へ一旦コピーしてから解析するのはほぼ不可能である。従って大量のデータを転送するよりも、データアーカイブ側で解析を行ってから、解析後の結果データのみを個人環境へ転送するようなシステムが望まれる。また，すばる望遠鏡のデータ解析は、可視光観測には馴染みの薄い天文研究者にとっては容易ではない。そのような分野外の研究者でも容易にすばる望遠鏡の高品質なデータを扱える環境を構築することを目指し，グリッドの技術を応用しながら Web サービスと FTP を利用した複数のサーバで並行して解析を行えるシステムを構築した。このシステムは，Japanese Virtual Observatory と連携させることで、ユーザは Web ブラウザ経由でこのシステムを利用可能となり，また，汎用性を高めることで他のデータ解析にも応用可能である。
キーワード　天文データベース バーチャル天文台 グリッド ポータル

Construction of the Astronomical Data Analysis System Federated with the Astronomical Databases

Yuji SHIRASAKI ET AL.[†], Masahiro TANAKA[†], Satoshi KAWANOMOTO[†], Satoshi HONDA[†], Masatoshi OHISHI[†], Yoshihiko MIZUMOTO[†], Masashi OE[†], Yoshifumi MASUNAGA[††], Naoki YASUDA[†††], Yasuhide ISHIHARA[††††], Junpei TSUTSUMI[††††], Hiroyuki NAKAMOTO[†††††], Yusuke KOBAYASHI[†††††], and Michito SAKAMOTO[†††††]

† National Astronomical Observatory of Japan
†† Ochanomizu Univerisity
††† University of Tokyo, The Institute for Cosmic Ray Research
†††† FUJITSU LIMITED
††††† Systems Engineering Consultants Co.,LTD.

Abstract　The amount of astronomical observation data is rapidly growing, so the way of traditional analysis appears to be insufficient for using the large amount of data effectively and efficiently. The SuprimeCam, which is one of the instruments equipped with the Subaru telescope, has been generating 6 TB of public data since its start of operation. It is almost impossible to transfer all the data to the local machine to analyze them. It is, therefore, desirable to have an environment where the data is analyzed where it is stored. We have applied grid technology to construct a system that carry out multiple job executions on multiple servers. Integrating this system to the Japanese Virtual Observatory, a user can easily utilize the grid system through the web browser interface.
Key words　Astronomical database, Virtual Observatory, Grid, portal

1. はじめに

天文学の研究分野では、ここ数年の望遠鏡ならびに観測装置の高機能化により良質で大規模な天文データが生産・蓄積されつつある。特に多数の CCD をモザイク状に敷き詰めた大面積カメラ技術の進展と、データストレージ装置の大容量化といった技術的進歩により、天文データは毎年倍増する勢いであり、それを解析する計算機の CPU やデータ転送のためのネットワーク速度の向上速度に比べても急ピッチに増加している。

国立天文台が運用をおこなっているすばる望遠鏡は 2000 年 12 月の共同利用観測開始以来 8 TB のデータが誰でも利用できるデータアーカイブとして公開されている (2006 年 10 月現在)。このような大容量のデータを研究者個人の解析環境にコピーすることは極めて困難であり、データアーカイブに解析機能を持たせる必要性が高まっている。

大容量データの問題に加え、すばる望遠鏡の公開データの解析を行うためには観測装置毎に用意される解析ツールを利用する必要があり、そうしたソフトウエアのインストールはもちろんのこと、利用方法についても学習しなくてはならない。近年の観測的天文学研究は様々な望遠鏡を利用して行われるようになってきており、装置によらない一様な手法で解析が行える環境構築の必要性が高まっている。

我々 Japanese Virtual Observatory (JVO) 開発グループはこれまでに世界中のデータベースを連携して検索を行うためのシステムを開発してきた。我々はさらにこの検索システムに解析システムを融合させることにより、データベースを利用した天文学研究をより行い易くする環境の整備を推めている。その端緒として、今回すばる望遠鏡データアーカイブと連携した解析システムを構築し、それを JVO ポータルサイトから容易に利用できるサービスの開発を行った。また、汎用解析ツールを自由に組み合わせることにより各利用者の要求にあった解析システムを構築するためのワークフローシステムを開発したので、それらについての技術的詳細について報告する。

2. すばる望遠鏡データアーカイブ

すばる望遠鏡はハワイ島マウナケア山頂に設置されている口径 8.2 m の可視・近赤外望遠鏡である。すばる望遠鏡は現在 7 種類の共同利用検出器が用意されている。それらの検出器はそれぞれ異なる特徴をもち、可視光に感度をもつものから近赤外線に感度をもつもの、撮像用のカメラや天体のスペクトルを測定する分光器など様々である。

取得されたデータはハワイ島にあるデータベースシステムである Subaru Telecope Archive System (STARS) に登録される。観測者はこの STARS から自分の観測データを取得することになる。STARS に登録されたデータは日本の三鷹市にある Mitaka Advanced STARS (MASTARS) へミラーリングされ、日本からデータを取得したい場合には MASTARS へアクセスすることになる。すばる望遠鏡により取得されたデータはデータ取得日から 1 年半は観測者に優先利用権が存在し、非公開データとして取り扱われる。STARS や MASTARS はそ

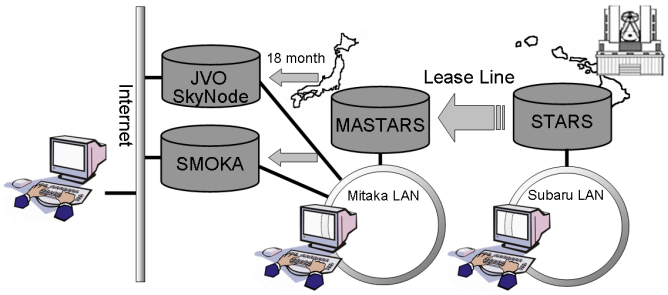

図 1　すばる望遠鏡データの流れ。

うした非公開データ専用のアーカイブ装置であるため、利用するには国立天文台のコンピュータシステム上にアカウントを持つ必要がある。優先利用権の期間を過ぎたデータは SMOKA システム(注1) や JVO システム(注2)を通じて公開データとなる (図 1)。

SMOKA システムは手動でインタラクティブにデータ取得を行えるシステムであり、すばる望遠鏡のデータ以外にも木曽観測所、岡山観測所などで得られたデータの配布も行っている。一方 JVO システムは国際バーチャル天文台により取り決められた標準インターフェイスをもつデータサービスを提供しており、プログラミングによるデータ取得が容易に行えるシステムとなっている。

2006 年 10 月時点における装置別公開データ量を図 2 に示す。この図からもわかるように、SuprimeCam 装置のデータ量は突出して大きく、実際のデータ請求率も 70% 以上は SuprimeCam のデータとなっている。SuprimeCam は 800 万画素の CCD が 10 枚敷きつめられた光学カメラであり、他の装置にくらべ撮像あたりのデータ生成量が多いことに加え、8 m クラスの望遠鏡に取り付けられた撮像装置としては世界最大の視野をもち、宇宙最遠の天体発見など新天体発見において絶大な威力を発揮している。従って、その利用形態としては全データを隈なく調べたいとう要望が多く、請求データ量も多くなる傾向がある。しかしながら、現状で 6TB に及ぶデータを数百人規模のユーザに配布するのはネットワークへの負荷が高く、また利用者側にもそのような大容量のディスクスペースを確保できるとはかぎらない。

SuprimeCam のデータ解析において計算時間ならびに大量

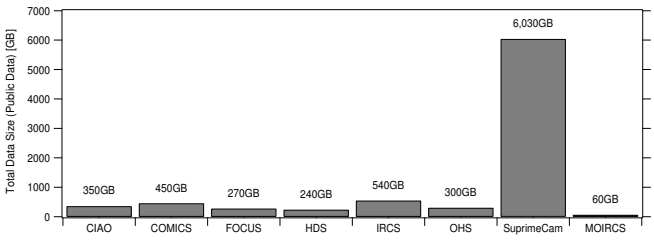

図 2　すばる望遠鏡の装置別公開データ量 (2006 年 10 月現在)

(注1) : http://smoka.nao.ac.jp/
(注2) : http://jvo.nao.ac.jp/portal

のデータを必要とされる処理は以下のものあげられる。

- CCD 面上での集光量の非一様性の補正 (フラット補正)
- 複数回行われる撮像の重ね合わせ処理 (モザイク処理)

フラット処理においては約 1 週間の観測シリーズ中に取得された全データを用いて、非一様性を補正するためのデータを、観測に使われたフィルター毎、CCD チップ毎に作成する。その際、数百枚の CCD 画像のメディアン平均を計算する必要があり、数 GB のデータ入力を必要とする上、8M ピクセル毎のソート処理が必要であり、計算力も必要とされる。ただし、この処理は一度補正用データを作成してしまえば、よほどの高精度なデータ解析を必要としない限り再利用が可能であるので、同じフラット処理の要求がユーザより送られてきた場合にはキャッシュデータを利用することが可能である。

一方モザイク処理においては多い場合で数千枚の CCD データから一枚の画像データの合成を行う。本処理では生データをフラット補正データを利用して一様化してからデータ間の相対位置を求め、画像合成を行う。移動天体探しや突発天体の探査を行う場合は、時刻別に分割してモザイク処理を行うので、任意のデータを利用したモザイキング処理が行える必要がある。

モザイキング処理により、数千枚の CCD 画像データはおよそ 10 枚分相当のデータ量へ削減されるため、この段階まで解析が行われればユーザ側の解析環境へ転送してさらなる解析を行うことも可能となる。さらに、以上の処理によって作成された画像に対し、写っている天体の位置、明るさ、形状といったパラメータを抽出しカタログ化することにより、さらにデータ量を小さくすることが可能である。したがって、天体カタログ作成までをアーカイブ側で行う仕組みづくりが我々の目的である。

3. グリッド計算システムの構築

数十人から数百人規模のユーザが利用することが想定され、ひとつのプロセスも最長数時間程度が見込まれる。したがって、解析を行うコンピューティングリソースの効率的な利用が可能なシステムの構築が不可欠である。そうしたシステムを実現するため、グリッドコンピューティングの概念を導入しシステム設計を行った。今回開発を行ったシステムは、ジョブの実行が行われる解析ノード、データの保管場所としてのストレージノード、データの所在を管理するデータ検索ノード、そして解析ノードの負荷情報を管理し適切にジョブの実行先を割り当てるためのジョブ管理ノードからなる。本論文ではこれらのノードを以下の略語で呼ぶことにする。

- Data Analysis Service (DAS)
- Storage Service (SRS)
- Data Search Service (DSS)
- Monitoring and Discovering Service (MDS)

DAS はジョブの実行を行うサービスである。各 DAS サーバは定期的に自分自身のステータス情報を MDS に報告する。ジョブ実行が行われた場合には、そのジョブの実行ステータスの報告も行う。したがって、ジョブを投入したクライアントはその実行ステータスを MDS サーバに問い合わせることで得ることができる。

ジョブの実行ステータスを DAS に直接問い合わせることも可能であるが、DAS の機能簡略化のためステータス管理は MDS にて一括管理できるようにした。DAS はシステム中多数のノードが存在するため、できるだけ機能はシンプルにしておくと便利である。

SRS はデータの送受信機能を提供するほか SRS ノード上のファイル削除要求も受け付けるサービスであり、ノード間でのデータ送受信を実現する。

大容量ディスク装置に接続したファイルサーバに SRS の機能を持たせている。この大容量ファイルサーバに対しては、複数の DAS サーバから同時にデータ要求が発生する場合があるため、ファイルサーバ専用として運用する。現在の試験システムでは 5 台のファイルサーバにそれぞれ 5TB の RAID 装置をとりつけて、ファイルサーバシステムを構成している。

DAS 自身も解析に必要なデータを受信したり、解析結果を送信したりする必要があるため、SRS の機能も有する。

ジョブの実行を行ったクライアントは MDS によるジョブの終了ステータスの確認後、DAS に対して結果取得先 URL をリクエストし、その URL を SRS に対し渡すことにより結果の保存を行う。

本システムではデータ転送は SRS ノード間の FTP 転送により行われる。SRS ノードでは FTP サーバが起動しており、Web サービスインターフェイスとして以下で述べる `copyAsync` や `copy` を実装する。これらの Web サービスインターフェイスを利用することにより、HostA が HostB から HostC へのデータ転送実行を命令することが可能である。

DAS 上で動作する解析プログラム自身が FTP クライアントの機能をもつ場合もあり、その場合第 3 者的ホストからデータ転送を指示せずとも、自身で必要なデータをローカルディスクに保存してから解析が開始される。SRS ノード間では一度に大量のデータが流れる場合が想定されるため、専用ネットワークにより直結する。

MDS はそれぞれの DAS に投入されたジョブ数やロードアベレージ、ハートビートの監視といった DAS のステータス管理を行い、グリッドシステムに投入されるジョブの実行先を適切に振り分ける役割を担う。

OGSA アーキテクチャ [1] における Information Service と Resource Selection Service の機能の一部をまとめたものである。今回構築したシステムでは実行先ホストの決定は単純なラウンドロビン方式を採用しており、複雑なロジックが実装されているわけではないので両者をひとつのサービスとして集約した。

また、OGSA ではジョブの投入は Job Submission Description Language JSDL により Job Manager が行うようになっているが、本システムではクライアントが直接ジョブを投入する。JSDL によるジョブ投入は DAS のインターフェイス実装を複雑にするため、今回は通常の Web サービス呼出をクライアントに行わせる方式にした。この方式のほうが、より柔軟に解析サービスのインターフェイスを定義することが可能であり、

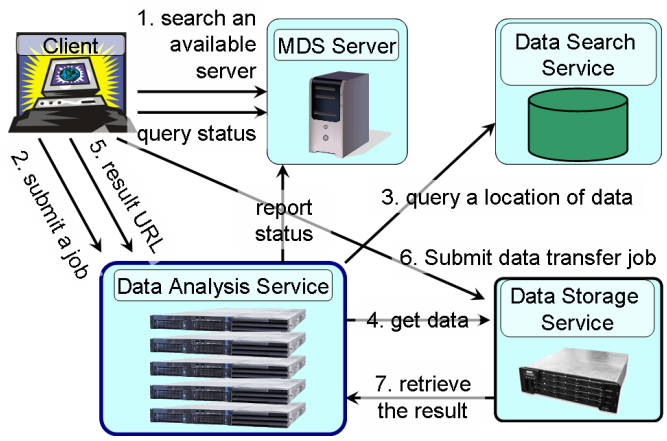

図 3　SuprimeCam フラット補正データ作成における各グリッドコンポーネントの呼出手順。

特に特定解析サービス専用のクライアントを作成する場合に実装が容易になる。

ジョブ投入数の最大値はそれぞれの DAS ノードの CPU コア数に設定してあり、それ以上のジョブ投入は行われないようになっている。全ての DAS ノードの CPU が 100%利用されている場合には、クライアント側で待機するように実装する。ジョブ投入を行うクライアントは、まずジョブの実行先を MDS に問い合わせ、CPU 資源に空きのあるサーバのアドレスを取得する。どのサーバにも CPU 資源に空きがない場合、MDS はその旨クライアントに報告し、クライアントは空きができるまで定期的に MDS に対し空きホストのリクエストの発行を続ける。ジョブ投入に成功した後は、ジョブの終了ステータスの監視を MDS もしくは DAS に対してポーリングすることによって行う。

DSS は天文データの検索サービスであり、国際バーチャル天文台連合で取り決められた標準インターフェイスによる検索を行うことができる。DAS 上で実行されるジョブは必要なデータの所在を DSS　に問い合わせることに得ることができる。

これらのコンポーネントならびにクライアント間のリクエスト・応答は HTTP プロトコルを利用した SOAP メッセージの交換により行われる。各コンポーネントが提供するサービスのアクセスインターフェイスは Web Service Description Language (WSDL) により記述される。各コンポーネントが実装するサービスインターフェイスの例を以下に示す。

```
MDS Interface
    void reportStatus(String hostId, double load,
                                          int njob)
    ServiceInfo resolveService(String serviceId)
    void reportJobStatus(String hostId, int jobId,
                                    String status)
    String getJobStatus(String hostId, int jobId)
    ...
DAS Interface
    int submitJob(String command, String argv)
    String getResultURL(int jobId)
    String query(int jobId)
    String finalize(int jobId)
    ...
SRS Interface
    int copyAsync(String src, String dest)
    void copy(String src, String dest)
    void finalize(int jobId)
    ...
DSS Interface
    VOData performQuery(Select select)
    ...
```

MDS の reportStatus インターフェイスは DAS サーバが自身のステータスを MDS に対して報告する際に利用される。resolveService インターフェイスは、グリッドシステムのクライアントがジョブの実行先を解決する際に利用される。このインターフェイスで返される ServiceInfo はジョブ投入を行う Web サービスインターフェイスのエンドポイント URL を含む構造型データである。

DAS の submitJob インターフェイスはジョブを実行する際に利用され、引数で指定されたコマンドが指定された引数により実行される。ジョブの実行時間は数時間に及ぶ場合も想定されるため、非同期型実行形式をとる。利用可能なコマンドはあらかじめ定義されており、それ以外のコマンドの実行はシステムの安全上許可していない。このインターフェイスの戻り値は、投入したジョブを一意に識別するための ID 番号である。この ID 番号を使ってジョブ終了の監視を行う。submitJob インターフェイスは DAS の必須インターフェイスであり解析サービスの種類によらず一様な呼出方式を実現するものであるが、サービス固有のインターフェイスを定義することが可能であり、より利用しやすいインターフェイスを提供する。サービス固有のインターフェイスを利用する場合には、クライアントアプリケーション側でインターフェイスに渡すべきデータの型などの情報をあらかじめ知っていることが必要である。

getResultURL インターフェイスはジョブ終了後に結果を取得するための URL を DAS サーバに対しリクエストする際に利用される。

SRS の copyAsync ならびに copy インターフェイスはサーバ間もしくはサーバ上のディレクトリ間でデータコピーするためのインターフェイスである。copyAsync は非同期でデータコピーの実行を行うためのものであり、長時間のデータ転送が予想される場合に利用される。copy は同期型インターフェイスであり、短時間で転送が終了されることが予測される場合に利用される。performQuery インターフェイスは国際バーチャル天文台連合において定義されているデータ検索インターフェイスであり、SQL を天文データベース用に拡張した言語である Astronomical Data Query Language により検索要求を行う。検索結果もバーチャル天文台における標準フォーマットである VOTable により返される。

図 3 に、すばる望遠鏡に搭載されている主焦点カメラのデータを用いた、カメラ受光面における受光量の非一様性を補正

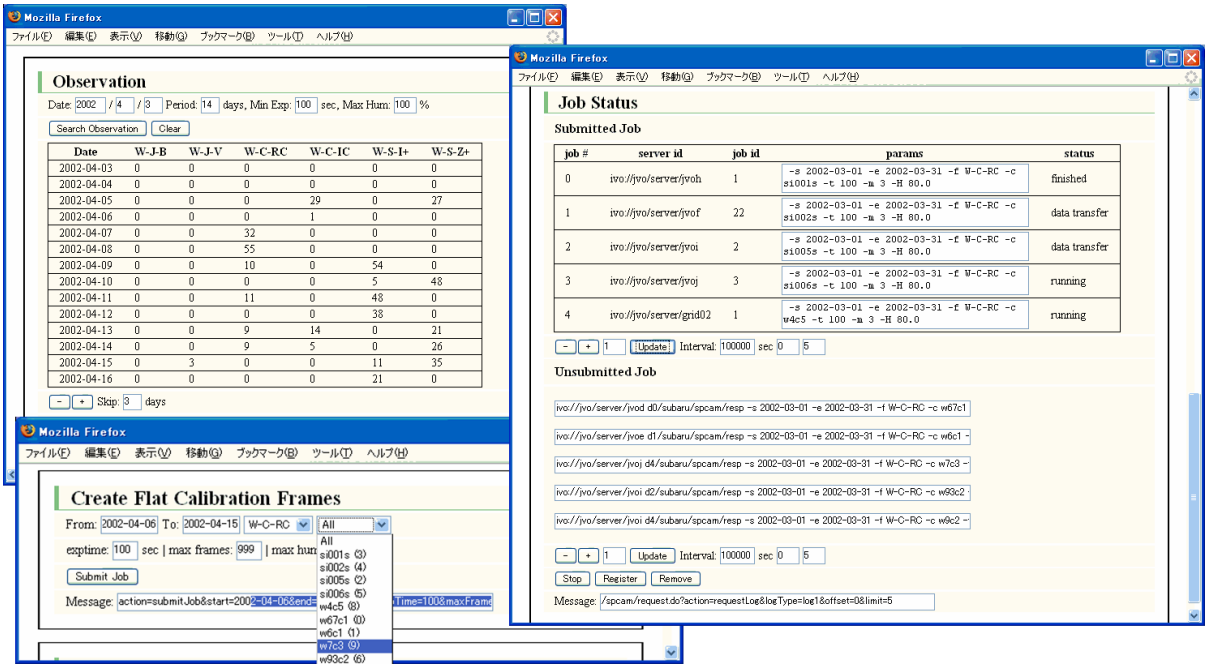

図 4　SuprimeCam 用フラット補正データ計算用 GUI

するレスポンスデータの計算手順を示す。このレスポンス計算を Web ブラウザを利用して実行するための GUI を作成した (図 4)。

Observation セクション では観測データのブラウジングを行い、レスポンス計算に利用するデータの観測期間の決定を行う。通常、1 週間程度の連続して行われた観測のデータを利用する。**Create Flat Calibration Frames** セクション では, **Observation** セクションの情報を使って決定された利用するデータの取得期間の入力と、レスポンスを計算したいフィルターと CCD チップの選択を行う。全てのフィルター、チップを選択することも可能である。その他の計算用パラメータの入力が終了したら **Submit Job** ボタンを押しジョブのグリッドシステムへの投入を行う。

投入されたジョブのステータスは **Job Status** セクションにおいて確認ができ、ステータスは “running”、 “data transfer” そして “finished” と遷移する。エラーが発生した場合には “error” ステータスが表示される。

現在、このグリッドシステムで利用可能な CPU コア数は 30 であり、CPU コアあたり一つまでのジョブ投入が許されている。したがって、30 以上のジョブを投入した場合は CPU の空きができるまで、待機リストにキューイングされる。図 5 は MDS が保持している各 DAS サーバのステータス情報を表示する Web ページのスナップショットである。ここのページにアクセスすることで、利用可能な DAS サーバの台数や各サーバでのジョブの実行状況の他、サーバの性能などについても確認可能である。“living” の列は各 DAS サーバからのハートビート受け付け状況を示し、定期的に報告が行われている DAS サーバは “true” と表示され、報告が断絶した場合には “false” が表示される。“enabled” の列は “true” と表示されているサーバがジョブ投入を受け付けているサーバであることを示す。ここのフラグはグリッドシステムの管理者が設定するものであり、一般のユーザは変更をすることはできない。“load” と “numJob” の列はそれぞれロードアベレージ、サブミットされたジョブ数を表し、各 DAS サーバが定期的に報告してくるステータス情報を元にデータが更新される。

図 6 は 合計 12 CPU コア を利用して 58 個のジョブを解析システムに投入した場合の、各ジョブの実行期間をグラフ化したものである。ジョブ投入は期待されるとおりに行われている。

4.　バーチャル天文台ポータルとの連携

今回開発したグリッド解析システムの Japanese Virtual Observatory (JVO) ポータルへの組み込みを現在行っている。JVO システムの詳細については文献 [3] [4] [2] において述べられている。これまで検索機能を主体としていたサービスに解析機能が付加されることにより、より天文学研究が効率的に行えるようになることが期待される。図 7 に現状の JVO システムの概観を示す。バーチャル天文台からのグリッドシステムの利用は、本グリッドシステム用クライアント機能を実装した Java クラスを用いることで可能である。ポータルシステムはグリッドクライアントクラスインスタンスに実行したいコマンド名とそれに与える引数リスト、結果の転送先を設定してコマンド投入用のメソッドを呼び出すだけである。現状でのグリッドシステムの呼出は、次の節で述べるワークフローを利用することにより行える。より簡便にジョブの実行を行えるよう GUI の開発を現在行っている。

前節において紹介された「フラット計算システム」により用意されたフラット補正データを利用して、オンザフライで生データからフラット補正までを行うデータサービスを構築し、現在 JVO システムから利用可能となっている (図 8)。この新システムにより、これまでは数時間にも及ぶ複雑な解析プロセ

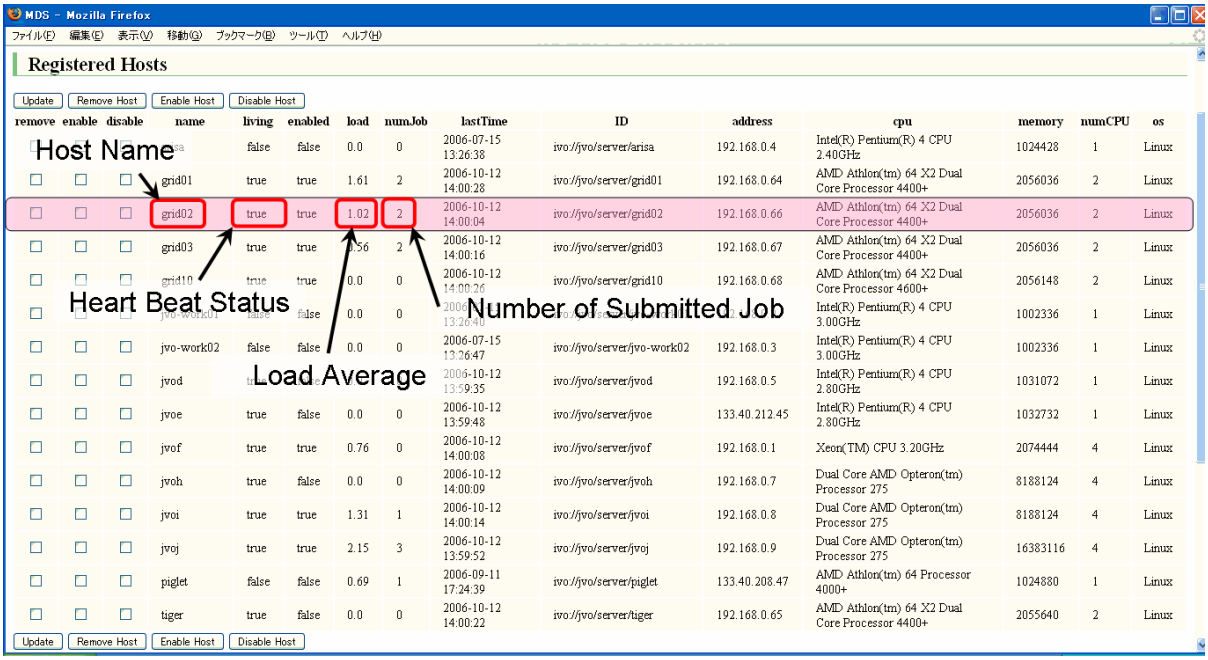

図 5　MDS による各 DAS サーバのステータス表示画面

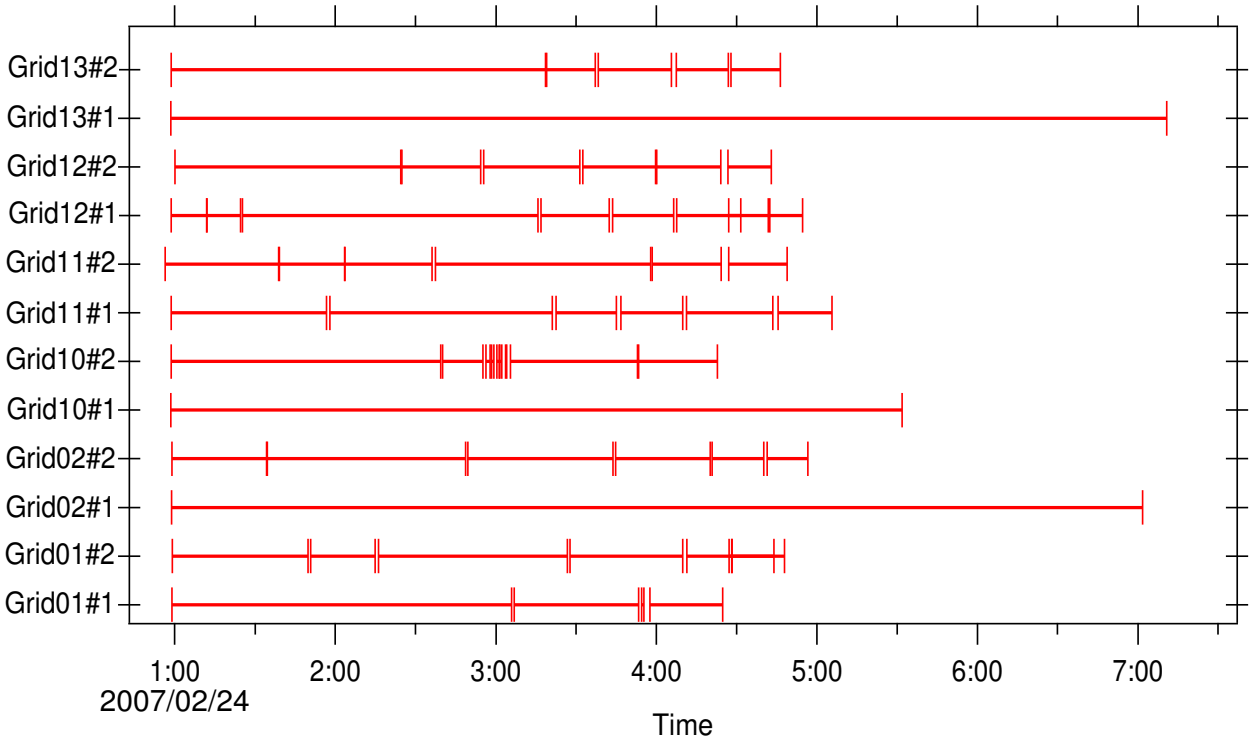

図 6　複数ジョブの並列実行試験結果。縦棒から縦棒までがひとつのジョブに対応する。縦軸の **Grid13#1** はホスト名が Grid13 であるサーバ上で実行されている第一ジョブであることを示す。この試験利用したサーバはすべてデュアルコア CPU であるので同時実行可能なジョブ数は最大 2 である。

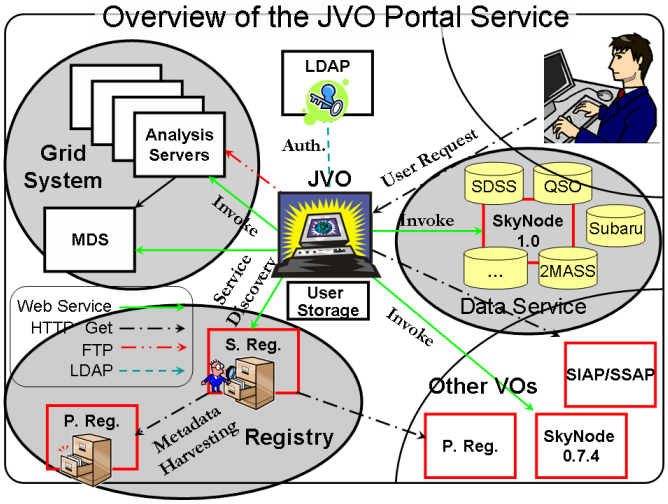

図 7　グリッドシステム組込み後の JVO システムの概観図

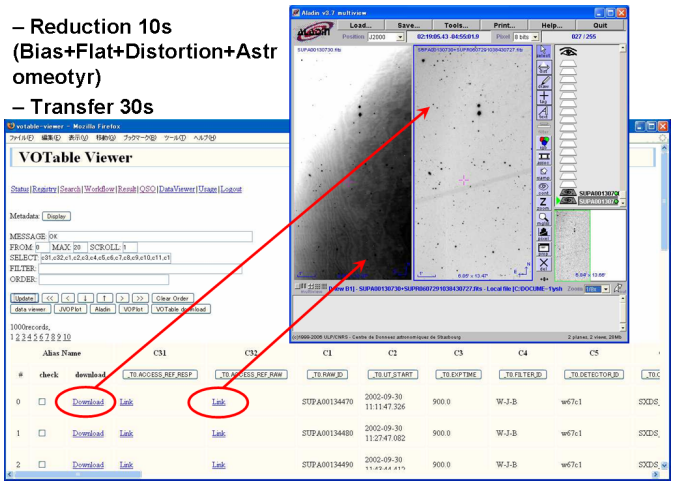

図 8　JVO システムよりフラット補正済みデータの取得が可能となった。

スを経なくてはならなかったデータ処理が、わずか数十秒で HTTP URL リクエストにより行えるようになった。

5.　ワークフローシステム

今回開発した GRID システムをすばる望遠鏡データの解析だけではなく、より汎用的な解析エンジンとして利用するために、ワークフローシステムの開発を進めている。一連のデータ処理手順をワークフローとして記述することにより、何度も繰り返される処理や、長時間に及ぶ処理を容易に実行することが可能となる。

我々が設計したワークフーロシステムは三つのコンポーネントからなる。

- ワークフロー構築用ユーザインターフェイス
- ユーザインターフェイスや実行環境の実装方式に非依存の抽象化されたワークフロー記述言語、
- ワークフローの実行環境

である。ワークフロー記述言語は　BPEL4WS をベースに天文学研究において必要となる機能を追加し XML として定義した。ユーザインターフェイスとしては、スクリプト形式の言語とグラフィカルユーザーインターフェイスの二方式の開発を行っている。どちらの方式で記述されてもワークフロー記述言語に変換されれば共通に実行が可能である。ワークフローの実行は XML ワークフロー記述言語をスタイルシートにより Java から派生したスクリプト言語である groovy に変換されたから実行される。このように XML からスクリプトに変換することにより、ループや演算処理をスクリプト言語の機能に移譲することができ、開発が短期間ですむ利点がある。また、スタイルシートとして ruby 用あるいは python 用のものを用意することにより様々な言語をワークフロー実行エンジンとして利用可能である。

我々が定義したワークフロー記述言語の仕様を付録にまとめた。その他にも、随時内部コマンドの充実をはかっていく予定である。

6.　まとめと今後の課題

今回開発したグリッドシステムにより多数の解析リクエストを効率的に処理する枠組が完成した。このシステムにより、ユーザーのローカルマシンへの大量のデータ転送が必要であったデータ解析が、データアーカイブに近いところで行うことが可能となる。これにより、データ転送のために天文台外とのネットワーク帯域を使い尽くすことなくデータサービスを行うことができる。現システムの計算能力をフルに利用すると、すばる望遠鏡　SuprimeCam 装置の全アーカイブデータをわずか 4 日で解析することが可能である。これまでのように一旦データをローカルマシンへ転送してからシングル CPU の個人計算機で解析を行う場合、1 年以上は必要であった計算が 1 週間足らずで終了できることになる。

また、利用者もウェッブブラウザーがあるだけですばる望遠鏡のデータを利用することができ、天文学研究の効率が格段に向上することが期待される。現在はまだ試験システムの段階であり、ジョブのキャンセル、一時停止、途中からの再開、長時間たっても終了しないジョブの取扱い、といった管理機能が不足しており、今後の課題である。今後はこれを実運用システムとして耐えうる堅牢で使い易いものにしていく予定である。そのためには、一人のユーザにのみ計算資源が占有されるのを防ぐため、ユーザ毎のジョブ実行制御をきめ細かく行う機能が不可欠である。また、解析結果の再現性という観点から、ジョブの実行時の条件等をデータベース化し過、去に実行した解析を再度実行可能なシステムにする必要がある。今後はそうした機能を追加していく予定である。

文　　献

[1] I. Foster, H. Kishimoto, and et al. The open grid services architecture version 1.0. Jan 2005. http://www.ggf.org/documents/GFD.30.pdf.

[2] Shirasaki Yuji, Masahiro Tanaka, Satoshi Kawanomoto, Satoshi Honda, Masatoshi Ohishi, Yoshihiko Mizumoto, Naoki Yasuda, Yoshifumi Masunaga, Yasuhide Ishihara, Jumpei Tsutsumi, Hiroyuki Nakamoto, Yuusuke Kobayashi, and Michito Sakamoto. Japanese virtual observatory (jvo) as an advanced astronomical research environment. In Lewis Hilton and Bridger Alan, editors, *Advanced Software and*

Control for Astronomy, Vol. 6274 of *Proceedings of the SPIE*, 2006.

[3] 田中昌宏, 白崎裕治, 本田敏志, 大石雅寿, 水本好彦, 安田 直樹 増永, 良文. バーチャル天文台 jvo プロトタイプシステムの開発. 日本データベース学会 Letters, Vol. 3, No. 1, pp. 81–84, 6 2004.

[4] 本田敏志, 大石雅寿, 白崎裕治, 田中昌宏, 川野元聡, 水本好彦. 天文学連携データベースシステム（ヴァーチャル天文台）の開発・計算機資源の国際連携機構. 日本データベース学会 Letters, Vol. 4, No. 1, pp. 173–176, 6 2005.

付　　録

ワークフロー記述言語の仕様について記述する。ワークフロー記述言語は XML により定義される。ルート要素 `workflow` は `variables` 要素 と activity 型要素を含む。`variables` 要素によりワークフロー内部で利用される変数の定義ならびに初期化を行う。`activity` 型要素としては以下の要素が定義されている。

- `sequence` 子要素に `activity` 型要素を複数個とれ、それらが順番に実行される。
- `flow` 子要素に `activity` 型要素を複数個とれ、それらが並行して実行される。
- `script` ワークフロー実行環境がサポートするスクリプト言語をそのまま記述できる。
- `command` ワークフロー実行環境の内部コマンドを実行。
- `if` 条件分岐を記述する。プログラミング言語の if 文に相当する機能をもつ。
- `switch` 条件分岐を記述する。プログラミング言語の switch 文に相当する機能をもつ。
- `for` ループ処理を記述する。プログラミング言語の for 文に相当する機能をもつ。
- `while` ループ処理を記述する。プログラミング言語の while 文に相当する機能をもつ。
- `continue` ループ中で以後の処理をスキップする。
- `break` ループから抜ける。
- `exit` ワークフローを停止する。
- `parfor`, ループ処理を記述する。プログラミング言語の for 文に相当する機能をもち、ループ内の処理が並行して処理される。
- `awk` 指定したファイルに対し一行毎に処理を行う。shell コマンドの awk の機能の一部を実行できる。
- `set`. 変数に値を代入する。

用意されている内部コマンドは、

- `copyText` テキストデータファイルのコピー
- `executeHyperZ` 測光データを用いた赤方偏移計算を行うソフトウエアの実行
- `executeQuery` バーチャル天文台データサービスに対し SQL によるデータ検索を実行
- `getCurrentWorkDir` 現在のワーキングディレクトリを返す
- `invoke` 任意の Web サービスを呼び出す
- `loadAsDataHandler` ディスク上のファイルを DataHandler オブジェクトとしてメモリ上にインスタンス化する
- `loadVOTable` ディスク上の VOTable ファイルを VOTable オブジェクトとしてメモリ上にインスタンス化する
- `storeVOTable` メモリ上の VOTable オブジェクトをディスク上にファイルとして保存する
- `getFile` 指定したパスの File オブジェクトを返す
- `loadAsString` ディスク上のファイルの内容を String オブジェクトとてメモリ上にインスタンス化する
- `replace` 文字列の置換を行う
- `split` 文字列を、指定した区切り文字で分割する
- `getColumnByName`VOTable から指定したカラムを抜き出す
- `echo` ファイルに文字列を書き出す
- `rm` ファイルを削除する
- `sort` ファイルの内容をソートする
- `xmatch` 複数の VOTable を座標値で結合する
- `filenameMatch` 指定した文字列パターンにマッチするファイル名の配列を返す