

JVO プロトタイプシステムの開発

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^{†††}

[†] 国立天文台 〒 181-8588 東京都三鷹市大沢 2-21-1

^{††} 東京大学 宇宙線研究所 〒 277-8582 千葉県柏市柏の葉 5-1-5

^{†††} お茶の水女子大学 理学部 情報科学科 〒 112-8610 東京都文京区大塚 2-1-1

E-mail: [†]{masahiro.tanaka,yuji.shirasaki,honda.satoshi,masatoshi.ohishi,mizumoto.y}@nao.ac.jp,

^{††}yasuda@icrr.u-tokyo.ac.jp, ^{†††}masunaga@is.ocha.ac.jp

あらまし 国立天文台では、すばる望遠鏡などが産み出す膨大かつ多様な天文観測データベースに基づく Japanese Virtual Observatory (JVO) の開発をおこなっている。このシステムは、我々が定義した検索言語で記述された検索条件を受け取ると、必要なデータが保管されたデータベースサーバを自動的に探し、Globus Toolkit 3 のグリッドサービスを用いて検索サービスを遠隔実行する。本システムにより、これまで困難であった多波長データへのアクセスを容易になる。本稿では、JVO プロトタイプシステムの実装について述べる。

キーワード 天文学 DB, 分散 DB, DB 言語, 科学 DB

Development of JVO prototype system

Masahiro TANAKA[†], Yuji SHIRASAKI[†], Satoshi HONDA[†], Masatoshi OHISHI[†], Yoshihiko MIZUMOTO[†], Naoki YASUDA^{††}, and Yoshifumi MASUNAGA^{†††}

[†] National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo, 181-8588 Japan

^{††} Institute for Cosmic Ray Research, University of Tokyo, 5-1-5 Kashiwa-no-Ha, Kashiwa, Chiba, 277-8582 Japan

^{†††} Department of Information Science, Ochanomizu University, 2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 Japan

E-mail: [†]{masahiro.tanaka,yuji.shirasaki,honda.satoshi,masatoshi.ohishi,mizumoto.y}@nao.ac.jp,

^{††}yasuda@icrr.u-tokyo.ac.jp, ^{†††}masunaga@is.ocha.ac.jp

Abstract Japanese Virtual Observatory (JVO) is developed at National Astronomical Observatory of Japan (NAOJ) based on a huge amount and variety of astronomical data produced by Subaru telescope etc. Once the JVO system receives a query condition written in the query language we defined, the system searches for required database servers and executes remote query services using Grid service of Globus Toolkit 3. This system provides easy access to multi-wavelength data, while it has been difficult. This paper describes development of the JVO prototype system.

Key words Astronomical DB, Distributed DB, DB language, Scientific DB, Data Center

1. はじめに

すばる望遠鏡 [1] を始めとする世界中の望遠鏡や宇宙望遠鏡などの最新の天文観測装置により、良質で大規模なデータが大量に生み出される時代になった。そうしたデータは一次処理後アーカイブ化され、一定期間経たのち一般に公開されるようになってきている。こうした天文学データアーカイブの天文学研究における重要性は高まってきている。特に電波からガンマ線にわたる広い波長範囲の電磁波による観測データを使った統計的な研究には、単一の望遠鏡では不十分であり、天文学データ

アーカイブの活用が必須となる。しかし実際にそのような研究をおこなうには大きな労力を必要とする。というのは、天文学データの公開をおこなっている機関は望遠鏡を開発・運用する観測所である場合が多く、公開するデータの形式や方法も機関ごとに統一されていないためである。世界各地に分散配置されたデータアーカイブから研究に必要な数々のデータを取得し、データを変換・加工して解析可能な状態にするまでには、非常に大きな労力が必要である。多くの天文学データを集めて統一的な検索インタフェースを提供する CDS [2] のような機関もあるが、それでも世界中の観測データを集めることは事実上不可

能である。

そこで、究極の天文学データアーカイブ検索システムとして、世界中に分散配置されたアーカイブサーバが連携することにより、世界中のすべての観測データの中から必要なデータを簡単に検索できるようなシステムが模索され始めている。このようなシステムは、計算機に蓄えられた観測データに基づいて仮想的な宇宙を構築し、それを自在に「観測」するための「天文台」であるということから「Virtual Observatory (VO)」と名付けられている。VOにおける大きな課題は、分散配置されたデータアーカイブの間で連携して検索することである。ここにグリッドを代表とする、情報処理の分野で最近発展している分散計算機リソースを有効利用する技術の導入が考えられている。

VO計画は各国で立ち上がっており、国立天文台においても Japanese Virtual Observatory (JVO) [3] [4] が進められている。JVOでは、独自のVOシステムの開発の他、各国のVOが相互連携するため International Virtual Observatory Alliance (IVOA) [5] にも参加し、VOに関する国際標準仕様の策定などもおこなっている。

本稿では、2002年度から富士通との協力により開発している JVO プロトタイプシステムの実装について述べる。プロトタイプ開発には次のような目的がある。1つはVOの仕様を検討するにあたって、実際に動作するシステムを用いて機能を実証すること、もう1つは本運用のJVOシステムの構築にあたり、採用した技術要素が有効かどうかを評価し、異種分散データベース連携などの機能が実現可能かどうかを実証することである。こうして構築したプロトタイプに基づいて、運用システムを構築する予定である。

2. 天文学観測データ

はじめにVOで扱う天文学観測データについて簡単に紹介する。実際の観測装置が出力する一次データは、装置ごとに異なる形式を持つ。VOでは通常この一次データを対象とはせず、一次データを処理して得られた二次データを扱う。二次データには、大きく分けて次のタイプがある。

2.1 画像・スペクトルデータ

画像データは2次元のグリッドで表わされる。各々のピクセルには通常、放射強度、すなわち明るさのデータが記録される。天文における画像の特徴は、世界地図のように、天球面から平面に投影されるということである。このとき座標や投影法の情報が必要となる。天球座標系には赤道座標系、銀河座標系など複数ある。

天文学では画像の他、電磁波を分光して得られたスペクトルデータを扱うことも多い。各天体に対してスペクトルを測定した結果は、波長方向の1次元のグリッドデータで表わされる。画像の各点に対してスペクトルまで測定している場合は、3次元グリッドデータとなる。

こうしたデータのファイルフォーマットとして、天文学では一般的にバイナリ形式の FITS (Flexible File Transfer) [6] が使われる。FITSは大きくヘッダー部とデータ部からなり、データ部には FITS データを階層的に含むことができる。ヘッダー部に

```
<?xml version="1.0"?>
<!DOCTYPE VOTABLE SYSTEM "http://us-vo.org/xml/VOTable.dtd">
<VOTABLE version="1.0">
<DEFINITIONS>
  <COOSYS ID="myJ2000" equinox="2000." epoch="2000." system="eq_FK5"/>
</DEFINITIONS>
<RESOURCE>
  <TABLE name="Stars">
    <DESCRIPTION>Some bright stars</DESCRIPTION>
    <FIELD ID="ra" name="ra" ucd="POS_EQ_RA_MAIN" unit="deg"
      datatype="double">
      <DESCRIPTION>Right Ascension (degrees FK5)</DESCRIPTION>
    </FIELD>
    <FIELD ID="dec" name="dec" ucd="POS_EQ_DEC_MAIN" unit="deg"
      datatype="double">
      <DESCRIPTION>Declination (degrees FK5)</DESCRIPTION>
    </FIELD>
    <FIELD ID="mag_aper" name="i_mag" ucd="PHOT_SDSS_I" unit="mag"
      datatype="float">
      <DESCRIPTION>Fixed aperture magnitude vector</DESCRIPTION>
    </FIELD>
    <DATA>
      <TABLEDATA>
        <TR><TD>201.0020154</TD> <TD>27.3279589</TD> <TD>19.5707</TD> </TR>
        <TR><TD>201.0296728</TD> <TD>27.3525103</TD> <TD>19.0456</TD> </TR>
      </TABLEDATA>
    </DATA>
  </TABLE>
</RESOURCE>
</VOTABLE>
```

カラム
メタデータ

データ本体

赤経 赤緯 i'バンド等級

図1 VOTableの例
Fig.1 VOTable Example

は、いわゆるメタデータを「キーワード・内容」形式で記録する。重要なメタデータには、観測時刻、観測装置、観測波長、座標系と座標、データの単位系などがある。

2.2 天体カタログ

天文学では画像・スペクトルデータだけでなく、天体カタログも用いる。天体カタログは、個々の天体について、明るさなどの観測値や属性を、テーブル形式で記録したデータである。このため、RDB (Relational DataBase) で扱いやすいデータと言える。天体カタログの項目には、天体の座標が必ず必要となる。その他の項目は決まっていないが、ほとんどの場合はある波長における天体の明るさのデータが含まれる。

天体カタログのデータフォーマットには、FORTRANでの入出力が容易なように、単純な固定幅のテキスト形式が多かった。しかしこれでは計算機がフォーマットを自動的に認識できない。その他に FITS にもテーブルを格納する方式が定められているが、それほど一般的にはなっていない。最近、XML形式のテーブル表現形式として、FITS テーブルの仕様を基に VOTable [7] が制定された。VOTableの例を図1に示す。VOTableに記録されるデータは大雑把に言えば、テーブル形式のデータ本体、およびテーブルのカラムについての詳しい情報(カラムメタデータ)から成る。VOTableは、計算機可読なデータ形式として、VOにおけるデータ交換のための標準的なデータフォーマットとなることが期待されている。

3. JVOに必要な機能

3.1 異種分散データベースの連携

前節で述べたような天文学観測データは、アーカイブデータとして公開されるケースが多くなっている。アーカイブデータを集めて公開しているデータセンターもあり、その中ではCDS [2] が最も充実している。しかしすばるの公開データを取得したいときは、CDSではなく、国立天文台のSMOKA [8] にアクセスする必要がある。というのは、アーカイブデータの管理のためには観測装置のことまで理解していることが求められるため、管理元は必然的に各観測所となるからである。また、アーカイブデータの管理方法も観測所の都合によって異なるこ

とが予想される。したがって、VOを実現するには、地理的に離れているアーカイブデータを統合的に扱うための異種分散データベース連携が必須となる。

JVOでは分散データベースへの問い合わせ言語として、SQLを拡張したJVO Query Language (JVOQL) [9]を定義した。また、各国のVOと連携するため、IVOA標準の検索言語の仕様策定もおこなっている。さらに、VO間のデータ交換を可能にするため、データ転送の際のフォーマットは標準に則ったものを採用すべきである。今後標準フォーマットになるとみられるのは、画像・スペクトルデータについてはFITS、天体カタログについてはVOTableである。

3.2 クロスマッチ

多波長観測データを使った研究の際にまず必要になるのが、異なる天体カタログから同じ天体を探すことである。これはRDBでIDを基にJOINをとることと似ている。JOINと異なるのは、同じ天体かどうかをまず座標位置の近さで判断することである。天体の位置の精度は、望遠鏡の口径や観測波長などで決まる空間分解能や、観測時の空の状態などの条件などによって決まる。天体の位置はこの精度以内で不定である。したがって同じ天体かどうかの判断は、天球面上の距離がある範囲以内であるという条件でおこなう。このような処理を「クロスマッチ」と呼ぶ。クロスマッチは既存のRDBにはないが、天文では使用頻度が高いため、JVOQLではクロスマッチのための仕様拡張をおこなっている。

3.3 画像・スペクトルデータ

SQLには画像やスペクトルデータを扱うための仕様は制定されていないが、天文学においては必須となる。そこで、JVOQLでは、天体の位置に対応する画像を切り出す文法を追加している。

3.4 解析機能

検索して集められたデータは、解析に用いられる。ここで解析とは、画像から天体の明るさを測定する、銀河の形を計算する、天体の「色」によって分類する、などというように、データを基にして各種パラータを求めることである。研究者が必要とするデータは、検索結果よりもこうした解析によって得られたデータだけであることが多い。また一般的には解析結果のデータ量は元となったデータの量より小さい。そこで、VO側で解析までおこない、その結果のみをダウンロードできれば、ネットワーク転送がボトルネックになる場合に有効である。さらに利用者は解析ソフトウェアを用意しなくてもよいという利点もある。

しかし、複雑な解析処理などをVOに依頼する際の一般的な指定方法についてはまだ考えられていない。こうした画像やスペクトルデータをVOで取り扱う方法の一般化は、今後の課題である。

3.5 可視化機能

取得した画像やスペクトルを調べるには、1次元のプロットや2次元の画像として表示する必要がある。VOでもデータの可視化をサポートすれば、利便性が格段に向上する。

クライアント動作の天文可視化ソフトウェアには、SAOim-

age [10]をはじめとしていくつか存在する。そういったものを流用可能か、VO用の可視化ツールが必要であるかどうかは検討中である。

4. JVOプロトタイプの開発方針

4.1 ターゲットユースケース

JVOシステムの開発にあたって、まず機能を限定したプロトタイプの開発をおこない、採用した技術の有効性や機能を評価するという手法をとった。

JVOの利用例(ユースケース)として、すばる望遠鏡の主焦点カメラSuprime-Cam(2k×4kピクセル×10CCDs)によって撮られたSubaru Deep Field(SDF)という領域のデータを用いた。ここで用いたデータの観測波長は*i'*バンド(683-854nm)、*z'*バンド(834-993nm)の2つである。それぞれのバンドごとに、画像データおよびその画像から検出された天体のカタログデータを用意した。分散データベース連携の実証のため、これらはそれぞれ異なるサーバにストアし、検索サービス・クロスマッチサービス・画像切り出しサービスを構築した。

ターゲットとなるクエリは、「*i'*バンドと*z'*バンドでクロスマッチをとり、両方のカタログに含まれる天体のリストおよびその画像を返す」というものである。これをJVOQLで記述すると、以下のようになる。

```
select
  i.POS_EQ_RA_MAIN as ra1,
  i.POS_EQ_DEC_MAIN as dec1,
  i.PHOT_SDSS_I,
  z.POS_EQ_RA_MAIN as ra2,
  z.POS_EQ_DEC_MAIN as dec2,
  z.PHOT_SDSS_Z,
  i.BOX(POINT(ra1,dec1), 0.005, 0.005),
  z.BOX(POINT(ra2,dec2), 0.005, 0.005)
from
  Subaru.SPcam.i i,
  Subaru.SPcam.z z
where
  XMATCH(i, z) < 10 arcsec NEAREST and
  BOX(POINT(201., 27.4), 0.10, 0.10) and
  i.PHOT_SDSS_I - z.PHOT_SDSS_Z > 0.0 and
  i.PHOT_SDSS_I < 20.0 and
  z.PHOT_SDSS_Z < 20.0
```

ここでwhere句にあるXMATCHという条件は、テーブル*i*と*z*とでクロスマッチを取ることを表わし、BOX(...)という条件は、赤経201度、赤緯27.4度を中心とする0.1度四方の領域以内の天体を検索することを表わす。また、select句の*i*.BOX(...), *z*.BOX(...)というカラムは、where句にマッチした天体の位置を中心に、それぞれ*i'*バンドと*z'*バンドの画像を0.005度四方で切り出すということを表わしている。JVOQLの文法についての詳細は白崎他[9]を参照されたい。

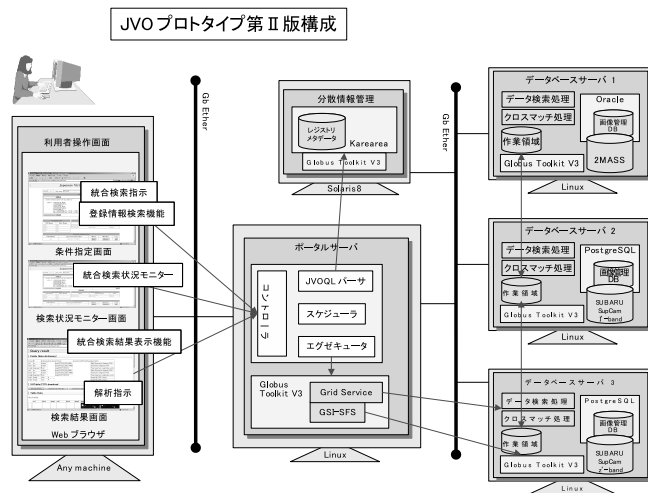


図2 JVO プロトタイプ第2版構成

Fig.2 Configuration of JVO prototype version 2

JVO ポータルがこのようなクエリを受け取ると以下のように動作する、というシステムを開発した。

- (1) JVOQL をパースして個々のサーバへのクエリに分解。
- (2) 個々のクエリを2つのサーバに送り、それぞれ条件にマッチする天体数を取得。
- (3) マッチした天体数が少ない方のサーバで検索を実行。結果をもう一方のサーバへ送る。
- (4) 送られたデータとサーバ内のデータでクロスマッチを実行。
- (5) クロスマッチの結果を基に、画像を切り出す。
- (6) 検索結果を元にテーブルデータおよび画像を表示する。

4.2 システム構成

図2に JVO プロトタイプのシステム構成を示す。システムは、1つの JVO ポータルサーバ、複数の分散データベースサーバ、その他のサービス提供サーバから成る。このシステムの大きな動作は以下の通りである。まず利用者は Web ブラウザで JVO ポータルサーバにアクセスする。利用者認証後、JVO のメイン画面が現れる。この画面から JVOQL で記述した検索命令を実行制御コントローラに送る。コントローラは命令を解析し、レジストリを参照して要求されたサービスを提供するサーバを引き当て、検索命令の実行手順を作成する。この手順に従い、グリッド経由で各サーバに実行要求を送る。結果のデータは、天体カタログデータは VOTable、画像データは FITS 形式で得られることとした。ユーザインタフェースの詳細は本田他 [11] を参照されたい。

5. プロトタイプ第1版

5.1 実装

前節で述べたシステム構成でターゲットユースケースの動作を実現することが、プロトタイプの目標である。実際にプロトタイプを実装する上では、個々の技術要素として何を採用するかが大きな問題となる。以下にプロトタイプ第1版の実装について述べる。

5.1.1 実行制御

ユーザが JVOQL で検索条件を書き、JVO ポータルサーバに送ると、JVO システムは実行コントローラに制御を移す。実行コントローラはまず最初に JVOQL パーサを実行する。JVOQL パーサは、ユーザが作成した JVOQL をパースし、個々のリモートサーバへの検索条件へと分解する。次にスケジューラが呼ばれ、実行手順を作成する。この実行手順を元に、実行コントローラは分解された検索条件をリモートサーバの検索サービスに送り、検索を実行するのである。

この実行コントローラは、動的に変化する実行手順にも対応できるようにした。4.1 節で述べたように、クロスマッチ検索では、まず最初に2つの天体カタログそれぞれに対してクロスマッチ以外の条件でマッチする個数を求め、個数が少ない方のサーバから多い方のサーバへデータ転送するようにする。このような手順にするのは、転送量をできるだけ少なくするためである。このように動的に変化する実行手順にも対応するため、スケジューラがサービスの実行結果に基づいて次回以降のサービスの実行手順を決めるという仕組みにした。さらに、1つのサーバへの実行が失敗したときは、別の似た機能を持つサーバへ再試行するという機能も持たせた。

プロトタイプ1では、JVOQL パーサおよびスケジューラもリモートサーバのサービスとして実装している。

5.1.2 遠隔実行・データ転送

世界中に分散した天文データベース連携のため、ネットワーク接続された計算機間での遠隔実行・データ転送を実現するグリッドのミドルウェアが必要となる。世界の VO が連携するために、ミドルウェアとして標準的な技術を使うことが重要である。現在この条件を満たすものとして、Globus Alliance [12] で開発された Globus Toolkit を採用した。

1回の遠隔実行の手順は以下ようになる。まず実行前にリモートサーバで実行したいシェルコマンドを送っておく。そして globus-job-run により、送ったシェルコマンドをリモートサーバで実行する。そのシェルコマンドは実行ステータスおよび出力したファイル名を標準出力に返す。実行コントローラは出力されたファイルを GridFTP (globus-url-copy コマンド) を用いて次に実行をおこなうリモートサーバへ転送する。最後に実行されたサービスの出力は、JVO ポータルサーバへと戻される。

5.1.3 サービスの検索

天体カタログデータは別々のサーバに分散配置されている。そこで実行コントローラは検索したいデータがどのサーバに保管されているかを知る必要がある。このサーバ情報を取得するため、プロトタイプ第1版では UDDI (Universal Description, Discovery, and Integration) を用いた。UDDI は Web サービスを検索・照会するシステムであり、必ずしも RDB 中のテーブルがどこにあるかを検索するために作られたものではない。そこで、天体カタログ1つ1つに検索サービスを対応させるという方法を取った。こうすることにより、UDDI でサービスを検索すれば天体カタログがどのサーバで提供しているかという情報まで得ることができる。ただし、遠隔実行に Web サービスを用いておらず、前述のような独自の遠隔実行方法を用いている

ため、UDDI から WSDL を取得しても JVO 用のサービスのスケルトンを作成できるにはなっていない。

5.1.4 データ検索サービス

データ検索サービスとして、天体カタログ検索サービス、クロスマッチ検索サービス、および画像検索サービスを開発した。

a) 天体カタログ検索サービス

天体カタログ検索サービスの構築には RDB を用いた。このプロトタイプ開発の目的の 1 つは、異種分散データベース同士で連携した検索を実証することである。そこで Solaris マシンと Linux マシンという異機種ハードウェアを組み合わせたテスト環境を用い、DBMS には Oracle と PostgreSQL の 2 種類を使用した。ただし RDB だけでは不足する機能として、JVOQL に特有の領域指定構文を解釈すること、結果を VOTable で出力することがある。

天球座標による領域指定は、ほとんどの天文データ検索において必須である。天球座標は赤経 (RA)、赤緯 (Dec) で表わされるが、これだけで効率的に検索することは難しい。というのは、天球座標は 2 次元であり、直交座標とは異なり距離や領域を表わすために複雑な式変換を必要とするためである。そこで、天球座標を効率的に検索するため、天球座標の領域を 1 次元のインデックスで表わす HTM [13] (Hierarchical Triangular Mesh) を採用した。HTM では天球面をまず 8 分割し、球面三角形の領域に区切る。このとき各々の領域につけるインデックスは、N0 から N3 および S0 から S3 である。さらに分割するときは、三角形の midpoint を結んだ線によって 4 つの三角形に分ける。それぞれの領域は一定のルールによって 0 から 3 までの数字が与えられる。これは 2 進数で 2 桁に相当する。分割された領域のインデックスは、分割される前の親領域のインデックスからこの 2 進数の 2 桁を増やすことによって作られる。このような分割を適当なレベルまで繰り返すことによって、天球は一意的なインデックスを持った多数の三角形の領域に分割される。JVO プロトタイプの検索サービスでも、次のようにして HTM によって天球座標を検索する機能を付け加えた。まず RDB に登録する天体カタログデータには、HTM のカラムを加えた。さらに、JVOQL による検索条件文中に領域指定の構文があると、検索サービスにおいて HTM の条件式に変換する。

天体カタログ検索サービスが出力するデータは、前述の VOTable で出力することとした。VOTable は XML であり、IVOA 標準となることが期待されている。データ本体については、RDB の結果に XML のタグを付加するだけでよい。その他に VOTable に必要な情報として、カラムメタデータがある。VOTable ではそれぞれのカラムに対して、カラム名・UCD [14]・単位・データ型・カラムの説明等の情報を加えることが要求される。そこで RDB にデータ本体に加えてカラムメタデータを別テーブルとして登録することとした。

b) クロスマッチサービス

クロスマッチ検索サービスは、前述のように、2 つの天体カタログの間で天球座標が近い天体を選ぶサービスである。今回構築したサービスの仕様は以下の通り。片方の天体カタログテーブルとして検索サービスが出力した VOTable を与え、もう

片方のテーブルは RDB 中の天体カタログとする。このサービスは、前述の検索サービスを拡張することにより構築した。

c) 画像検索サービス

プロトタイプで開発した画像検索サービスは、入力として検索の結果得られた VOTable を与えると、1 つの FITS 画像から、VOTable に含まれる天体の座標を中心とする指定の大きさの画像を切り出す、というものである。しかし画像データには複数枚の画像に分かれている場合もあり、そういうデータへの対応は今後の開発課題である。

5.2 評価

このプロトタイプによって、分散したデータベースを JVOQL により統一的に扱い、複数の独立なデータベース上のカタログデータを比較し、対応天体がある部分の画像を取得できることを実証した。

しかし多くの課題も出た。その 1 つはサービスの遠隔実行が非常に遅いことである。個々のサービスの実処理時間は 1 秒以下であるにもかかわらず、遠隔実行にすると 1 分以上もかかり、全体の処理が終わるまでに 10 数分もかかった。この理由は、globus-job-run で発行したジョブが GRAM (Globus Resource Allocation Manager) で管理されており、GRAM は長時間ジョブの遠隔実行を想定しているため、リアルタイム処理には向かないためである。globus-job-run はジョブを投げると、結果を受け取るために定期的にポーリングする。このポーリング間隔が Globus Toolkit のデフォルトでは 30 秒となっている。ソースを修正してポーリング間隔を 3 秒程度にすると、サービス 1 個の実行時間が 22 秒程度に短縮された。リアルタイム処理には globus-job-run ではなく GSI-SSH を用いることで改善されるかもしれない。

遠隔サービスの実装方法にも課題があった。プロトタイプ第 1 版では、天体カタログ 1 つ 1 つに検索サービスを作るという方法を取った。こうすることにより、天体カタログが保管されているサーバの検索は、UDDI でのサービス検索に置き換えられる。しかしこの場合、天体カタログを新たに加えるときはサービスも新たに構築することになる。そうすると、天体カタログが増えた場合に管理が煩雑になる。また、WSDL の独自拡張は他の VO との連携に支障が出る恐れがある。

6. プロトタイプ第 2 版

6.1 実装の見直し

JVO プロトタイプ第 1 版の評価で判明した問題点を改善するため、JVO プロトタイプ第 2 版では採用した技術要素を以下のように見直した。

6.1.1 遠隔実行

プロトタイプ第 2 版では Globus Toolkit 3 を採用し、そこで新たに導入された「グリッドサービス」を遠隔実行に用いた。グリッドサービスは、Web サービスをグリッド用に拡張したものであり、サービスの記述方法も WSDL を拡張した GWSDL が定められている。これはプロトタイプ第 1 版で構築したサービスと似ている。しかし独自仕様でなく、グリッドの標準仕様になることにより、サービスの構築を容易にしたり、外部との

連携をしやすくすることを図る。また、グリッドサービスの仕様に合わせて各サービスも再構築した。

6.1.2 データ転送

コントローラとデータベースサーバとの間のデータ転送には、認証付きのファイルシステムである SFS を利用した。また、各データベースサーバ間の転送には、Globus Toolkit 3 に含まれる RFT (Reliable File Transfer) を利用した。RFT は GridFTP の転送の信頼性を向上したものである。

6.1.3 レジストリ

JVO システムのレジストリは、各データベースサーバのサービス情報、および保管された天文観測データのメタデータを集約し、検索できるようにしたものである。レジストリは次のような場合に必要になる。例えば、どんなカタログがあるかわからないが「クエーサーカタログのデータが欲しい」というように、特定の種類のカタログを見つけたい場合。また、「この領域に含まれる天体のデータをすべて取得したい」というように、複数の天体カタログにまたがる検索をおこないたい場合。

プロトタイプ第 1 版では、天体カタログの検索は、UDDI を使ったサービス検索に置き換えたため、メタデータの検索ができなかった。そこでプロトタイプ第 2 版では、天体カタログのメタデータをレジストリに登録し、レジストリに対して検索することによって実現することにした。

メタデータには、アーカイブデータの情報(観測装置、天域、波長域、データの所有者等)や、サービス提供サーバの情報が含まれる。メタデータの仕様は、国際的な VO における標準化案と互換性を持つように設計した。これは将来諸外国の VO との連携を容易にするためである。このデータを XML 化し、XML データベース(製品「カレアレア」を利用)に入力し、XPath で検索することとした。

このレジストリに検索をかけることで、多波長の天体カタログを検索したり、検索サービスや解析サービスを提供するサーバを自動的に見つけたりすることが可能となる。

6.2 評価

プロトタイプ第 2 版では、第 1 版より実行時間が短縮された。検索処理だけに 2.3、13 秒かかるジョブをグリッドサービスで発行すると、遠隔実行を含めた経過時間はそれぞれ 2.8、16 秒程度と、わずかな増加にとどまったのである。このように、余分な待ち時間はほとんどなくなり、パフォーマンスは大きく改善された。理由は調査中であるが、Web サービススペースのグリッドサービスに変更したことにより、ポーリング間隔による待ち時間がなくなったためであろう。さらに JVOQL パーサおよびスケジューラをリモートサービスではなくメソッド呼び出しにし、メタデータ検索を UDDI から XML データベースにしたことにより、それらにかかる時間がほとんど無視できるようになった。こうして、JVO プロトタイプ第 2 版は、実用システムに大きく近づいたと言える。

さらにサービスおよびテーブルのメタデータを XML データベースに登録することにより、統一的にメタデータを検索できるようになった。特に観測波長等の天体カタログについてのメタデータを検索することにより、JVO からどんな天

体カタログが利用できるかをわかりやすく表示することが可能になった。

JVO プロトタイプでは Globus Toolkit を利用して実装してきた。しかし VO を構築する際に Globus Toolkit では不足すると感じられる点もある。それは、サービス実行とデータ転送を別々に実装しなければならないということである。グリッドサービスの引数としてデータを転送するならば別にデータ転送を実装する必要はない。しかしこの場合は中間データを一旦実行コントローラに戻すことになり、さらにその結果を別のサーバに渡す場合には無駄な転送が発生する。この問題は GridFTP または RFT を用いてリモートサーバ間で転送することにより解決する。そこでプロトタイプ第 2 版では、グリッドサービスと RFT とを独自に連携させる方式を採った。しかし独自仕様はメンテナンスや外部との接続の時に問題となるため、遠隔実行とデータ転送を統合したサービスの仕様がミドルウェアレベルで標準的に提供されることが望ましい。

また、グリッドサービスを利用できる言語が現状では Java のみであることも問題である。天文学データ処理ツールは Java 以外のいろいろな言語で記述されており、コマンドとして利用することも多い。そこで、コマンドやいろいろな言語からグリッドサービスを容易に利用できるような仕組みを期待する。

7. おわりに

本稿では、Globus Toolkit を利用して JVO プロトタイプシステムを実装し、分散データベースとして機能することを実証したこと、プロトタイプ第 2 版でグリッドサービスを採用することにより、実行時間の改善したことなどについて述べた。このプロトタイプにより、分散データベースとしての JVO システムの枠組みが確立できた。

プロトタイプの評価を踏まえ、今後、JVO 運用システムの開発を計画している。実用に耐えうるシステムにするには、検索サービスの高性能化、解析ツールの充実などの改良が必要である。また運用システムではユーザ認証およびリソースの管理が必要になる。その他の大きな課題として、組織間での VO の連携がある。別組織で保管されるデータは管理や運用のポリシーが異なる。そこで他の組織に配置された天文学データを利用するためには、どのようなプロトコルが必要か、どのような技術が使えるかになどについて検証することを考えている。

謝辞 本研究の JVO プロトタイプの開発においては、富士通(株)および(株)セックの方々のご支援を頂いた。本研究は科学研究費補助金 特定領域研究「IT の深化の基盤を築く情報学研究」の補助を受けておこなわれた。

文 献

- [1] Kaifu, Norio, “SUBARU Telescope” Proceedings of SPIE, Vol.3352, pp.14–22, 1998
- [2] CDS (Centre de Données astronomiques de Strasbourg) <http://cdsweb.u-strasbg.fr/>
- [3] 大石雅寿, “Japanese Virtual Observatory の構築” 天文月報, Vol.95, No.12, pp.566–575, 東京, 2002.
- [4] JVO <http://jvo.nao.ac.jp/>
- [5] International Virtual Observatory Alliance <http://www.ivoa.net/>
- [6] FITS Support Office <http://fits.gsfc.nasa.gov/>
- [7] VOTable Documentation <http://vizier.u-strasbg.fr/doc/VOTable/>
- [8] SMOKA Science Archive <http://smoka.nao.ac.jp/>
- [9] 白崎裕治他, “天文学 DB 用検索言語の開発” DEWS2004, 2004
- [10] SAOImage DS9 <http://hea-www.harvard.edu/saord/ds9/>
- [11] 本田敏志他, “バーチャル天文台の研究開発” DEWS2004, 2004
- [12] Globus Alliance <http://www.globus.org/>
- [13] Hierarchical Triangular Mesh <http://www.sdss.jhu.edu/htrm/>
- [14] Unified Content Descriptors <http://vizier.u-strasbg.fr/doc/UCD.htm>