JVO 開発における大規模天文データ処理 : 全天対応天文データ分散検索・解析機構の試験構築

白崎 裕治[1]，小宮 悠[1]，大石 雅寿[1]，水本 好彦[1]
石原 康秀[2]，堤 純平[2]，檜山 貴博[2]，中本 啓之[3]，坂本 道人[3]

Experimental Construction of A Distributed All-Sky Astronomical Data Query and Analysis System

Yuji SHIRASAKI[1], Yutaka KOMIYA[1], Masatoshi OHISHI[1], Yoshihiko MIZUMOTO[1],
Yasuhide ISHIHARA[2], Junpei TSUTSUMI[2], Takahiro HIYAMA[2],
Hiroyuki NAKAMOTO[3] and Michito SAKAMOTO[3]

Abstract

Astronomers have built highly sensitive ground-based and space-based telescopes towards solving various issues of the modern astronomy and obtaining new knowledge of the Universe. As the result produced data volume has been explosively increased. In order to utilize these astronomical resources organically, construction of "virtual observatories" have been advanced through federating the astronomical data archives by means of the ICT technology, which easily collect and analyze observed data necessary for astronomical research. It is necessary to introduce parallel processing into the virtual observatory system in order to search and analyze various distributed astronomical data, we have constructed an experimental scalable parallel data retrieval and analysis system by utilizing the Hadoop. We made performance tests of the system, and obtained very useful insights that can be used in the construction of operational systems; data analysis performance increases in proportional to the number of tasks until when the number of tasks do not exceed the number of CPU cores of the computers used; effective performance decreases due to interference among the tasks when the number of tasks exceed the number of CPU cores. The basic concept of this system may be referred to in the construction of "Virtual Observatory" in other science fields.

概　要

天文学者は，現代天文学が抱える多様な課題を解決し，宇宙に関する新しい知見を得るため，地上や衛星で運用する望遠鏡を建設・運用し，その感度を向上させると共にデータ生産量を爆発的に増やしてきた．これらを有機的に活用するため，世界に分散して存在する天文データアーカイブを ICT 技術により連携させ，研究に必要な観測データを容易に収集し解析するヴァーチャル天文台の構築が進んでいる．全天に分布する多様な天文データを検索・解析するためには並列処理をヴァーチャル天文台システムに導入することが必須であるため，我々は，Hadoop を利用したスケーラブルな並列データ検索・解析システムを試験構築した．構築したシステムの性能試験を実施し，試験構築に用いた計算機群のコア数とほぼ等しいタスク数までは，データ解析性能はタスク数に比例して向上すること，それを越えるタスク数を並列動作させた場合はタスク間の干渉により実効性能が劣化する，など，実運用システムの構築に向けた極めて有益な知見が得られた．本システムの基本概念は，他の科学分野における "Virtual Observatory" の構築の際にも有益な参考情報を与えると考えられる．

[1] 国立天文台 (National Astronomical Observatory of Japan)
[2] 富士通株式会社 (Fujitsu Limited)
[3] 株式会社セック (Systems Engineering Consultants Co.,LTD.)

1 イントロダクション

宇宙は，いわゆるビッグバンで始まったとされている．ビッグバン以降，宇宙膨張の過程の中で輻射と物質が分離し，物質は集積して恒星や銀河を形成した．銀河は恒星の大集団であるがその形成史については未知の部分が多い．いくつかの恒星の周囲には惑星が生まれた．恒星には寿命があり，恒星から放出された物質は再び集積して新しい恒星や惑星系を生み出す．しかし，恒星や惑星の形成過程についても多くの課題が山積している．天文学者は，これら現代天文学が抱える課題を解決し，宇宙に関する新しい知見を得るために地上や衛星で運用する望遠鏡を建設・運用し，その感度を向上させると共にデータ生産量を爆発的に増やしてきた．

一般に天体は，電波からガンマ線までの多波長で放射をしているため，各種天体現象の本質を知るために，多波長データの統計的な活用が求められてきた．すなわち，宇宙の諸現象を深く理解するためには，世界中の天文データを総合する研究基盤が必須である．しかし，天文データアーカイブが世界の主要天文台で構築されているにもかかわらず，その活用のための環境が整っていたとは言いがたい状況にあった．

一方，1990 年代後半からの情報通信技術（ICT）の急激な発展により，高速ネットワーク環境を容易に利用できるようになり，また高機能な計算機が安価に購入できるようになった．このような状況のもとで，ICT を利用すれば世界中の天文アーカイブを連携させて研究に必要な観測データを容易に収集し解析することが可能になるだろうという発想が，世界各地で自然発生的に浮かび上がってきた．これが「バーチャル天文台 (Virtual Observatory = VO) 構想」である．その構築をめざして，世界の主要国が協力して相互の資源を活用するための標準を定めてきた．これらの標準化活動の結果，2011 年 4 月現在，国立天文台が構築したヴァーチャル天文台システムである JVO (Japanese Virtual Observatory) [1,2,3,4] では，1 万を超える日米欧の主要な天文台やデータセンターにあるリソースが VO インターフェースを通じて相互に接続されている．

大量の天文データ（画像，スペクトル，カタログ）は，いずれも天球面上に分布している．望遠鏡で観測できる領域の最小の大きさ（空間分解能）は望遠鏡毎に異なり，細かいものでは角度の数マイクロ秒であるが粗いものでは数分から 1 度程度のものもある．また，望遠鏡が天体方向を向いていてもその指向精度には誤差が生じるため，得られた観測データの位置情報にも誤差が生じる．従って，このような天文データの特性を踏まえた上でヴァーチャル天文台機構を構築する必要がある．通常のデータ検索ではデータベースの各レコードに含まれる（誤差なし）数値情報に基づいて検索を行うが，データ量が大量になってくると検索結果をみる人間の負荷が大きくなる．一方，天球面上のどこにどのようなデータが存在するかを可視化することができれば，研究者は検索したい領域を容易に指定することができると期待される．そこで我々は，Google Sky API を利用して，天球面上にどのような観測データが存在するのかを可視化し，そこから既存の検索システムに検索要求を投入する機構 (JVOSky) を構築した．

バーチャル天文台においては，① データ検索，② データ取得，③ データ解析をネットワーク上で行う．データ解析の結果を踏まえて，さらに別のデータを取得して新たな処理を行うこともある．データ検索範囲が空間的（最大，全天を対象とする）もしくは波長（周波数）方向に広範囲にわたる場合，従来のように 1 台の VO ポータルマシン自身が検索先を探して順次検索命令を発行するのでは非効率的となる．この問題を解決するため本稿では，大規模データの分散アプリケーションをサポートするとされる Hadoop を利用し，スケーラブルかつ負荷分散が可能なデータ格納やデータ解析を実現する機構を試験的に構築し，従来方式に比べて非常に効率的な処理が実現できることを示す．

2 構築システム

2.1 天体カタログのクロスマッチと全天検索

天体望遠鏡により観測された画像データから検出される，天体の位置や明るさのデータセットを「天体カタログ」と呼ぶ．天体の識別は通常天球面上での位置情報に基づき行われる．しかしながら，観測量である位置情報は常に測定誤差をもつため，同じ天体でもカタログ毎に登録されている位置は完全には等しくはない．したがって，複数のカタログから個々の天体の情報をまとめるためには，まず位置の誤差を考慮した同定作業を行う必要がある．この同定作業をここでは「クロスマッチ (Cross Match)」と呼ぶことにする．

JVO のバックエンドデータベースとして構築された “Digital Universe” には，これまでに発行された主要な天体カタログを一つにまとめた総計 200 億レコードから成る天文データ測光カタログが登録されている [5,6]．この “Digital Universe” に対して高速に検索する仕組みをこれまで開発してきたが，一度に検索できるデータは半径 1 度程度の限られた天球領域でしかなかった．多数のデータに基づく統計的な天文研究を行うためには，全天のデータを検索対象とする必要があり，多波長のスペクトル情報にもとづく条件で天体を選択できる必要がある．そのためには，あらかじめ 200 億レコードのデータ間の

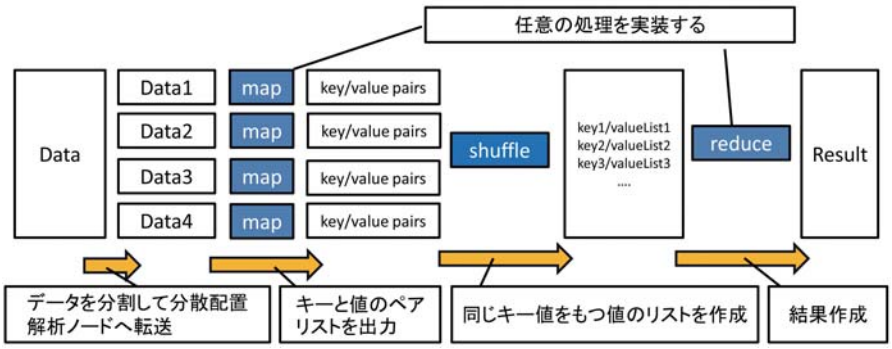

図 1: Hadoop における MapReduce アルゴリズムによる並列データ処理.

クロスマッチを行い，そのクロスマッチ結果にもとづく全天のデータにわたる多波長データ検索システムが必要となる．

200 億レコードのクロスマッチをシングルスレッドで計算する場合約一ヶ月もの時間を要してしまう．天文データは日々増加しており，クロスマッチすべきデータセットも増加の一途をたどっており，高速なクロスマッチ処理システムの導入が不可欠である．また全天検索も様々な条件で検索されるため，あらかじめ全ての検索条件に対してインデックスを作成して置くことは困難であり，全レコードを読む必要がある．こうしたクロスマッチ処理や全天データ検索を高速に行うためには並列処理システムの導入が不可欠となる．

2.2 Hadoop

そこで我々は，高速なクロスマッチ処理ならびに全天検索システムの実現，さらには天文データ解析システムとしての利用を目的として，Apache トッププロジェクトの一つであるフリーウェア，Hadoop[4] を利用した並列分散計算システムを構築した．Hadoop は分散データファイルシステム HDFS (Hadoop Distributed File System) と，それと連携して動作する分散ジョブ実行システムより構成される．Hadoop は様々な企業，大学などでの利用実績が豊富であり，近年では天文データ処理にも利用される例が出てきている．この分散ジョブ実行システムは MapReduce [7] と呼ばれるアルゴリズムにより実行される（図 1）．

一つの「ジョブ」を独立に実行可能な複数の「マップタスク」に分割し，複数の計算機上で各マップタスクが並列に実行される．最後に，それらマップタスクの結果を一つにまとめて最終結果へと変換する処理「リデュースタスク」が実行されジョブの実行が完了する．マップタスクは HDFS 上のファイルを入力データとして実行される．その際，マップタスクの実行は入力データが保存されている計算機上で最優先に行われるようにスケジューリングされる．このジョブのスケジューリングは「ジョブトラッカー」によって行われ，それは Hadoop システム内の一つの計算機上で動作する．タスクの実行は「タスクトラッカー」により行われ，これはタスクを実行するすべての計算機上で動作する．HDFS に登録されるファイルは，予め設定されたブロックサイズに分割されて複数の計算機に分散して配置される．また，複製ファイルを作成して異なる計算機上に配置されるようになっており，計算機ダウンに対する耐障害性も備えている．デフォルトの設定では複製は 2 つ作られる．すなわち同じファイルが 3 つ，異なる計算機上に存在することになる．各ブロックの配置場所に関するメタデータは「ネームノード」サーバーが管理する．各計算機上でのブロックの受信，保存，そして送信は「データノード」サーバーが行う．

マップタスクとリデュースタスクには任意のロジックを実装することが可能であり，アプリケーション毎にこの部分のみを変えることで様々な用途に利用できる．入力データに対する処理は「スプリット」と呼ばれる単位で行われる．スプリットの単位は行単位に設定できるほか，アプリケーションに応じて自由に定義することも可能である．

2.3 分散処理システム

今回構築した Hadoop 分散処理システムは，計算機 34 台を利用している．それぞれの計算機はヘテロなハードウエア構成となっており，搭載している CPU 種別，メモリ搭載量は表 1 に示すように 7 種類に分類される．各計算機のシステム構成，ならびに接続されている記録装置数を表 2 に示す．記録装置はハードディスクドライブ (HDD) またはソリッドステートドライブ (SSD) の 2 種類が利用されている．HDD はいずれも SATA2 インターフェイス，回転数 7200 rpm である．容量

[4] http://hadoop.apache.org/

表 1: 今回構築した分散処理システムに利用した計算機のシステム種別.

システム名	CPU モデル	クロック周波数	CPU 数	総コア数	総メモリ
Opteron	AMD Opteron (2347HE)	1.90 GHz	2	8	32 GB
Core2 A	Intel Core2 Extreme (QX6700)	2.66 GHz	1	4	4 GB
Core2 B	Intel Core2 Quad (Q6600)	2.40 GHz	1	4	4 GB
Xeon A	Intel Xeon (L5420)	2.50 GHz	2	8	24 GB
Xeon B	Intel Xeon (L5520)	2.27 GHz	2	8	24 GB
Xeon C	Intel Xeon (L3426)	1.87 GHz	1	4	16 GB
Athlon	AMD Athlon II X4 (615e)	2.50 GHz	1	4	8 GB

表 2: 使用した各計算機のシステム構成と接続している記録装置数

ノード	システム名	記録装置数	ノード	システム名	記録装置数
jvot	Opteron	161	grid70	Xeon B	410
grid21	Core2 A	12	grid71	Xeon B	410
grid22	Core2 A	23	grid72	Xeon B	410
grid30	Core2 B	23	grid73	Xeon B	410
grid41	Core2 B	44	grid74	Xeon B	410
grid42	Core2 B	45	grid75	Xeon B	410
grid43	Core2 B	46	grid80	Xeon C	411
grid44	Core2 B	27	grid81	Xeon C	411
grid53	Core2 B	28	grid82	Xeon C	411
grid54	Core2 B	28	grid83	Xeon C	411
grid55	Core2 B	28	grid90	Athlon	412
grid56	Core2 B	28	grid91	Athlon	412
grid57	Core2 B	28	grid92	Athlon	412
grid60	Xeon A	49	grid93	Athlon	412
grid61	Xeon A	49	grid94	Athlon	412
grid62	Xeon A	49	grid95	Athlon	412
grid63	Xeon A	49	grid96	Athlon	412

[1] HDD 1TB×16 (RAID6). [2] HDD 500 GB×1. [3] HDD 500 GB×2 (LVM). [4] HDD 500 GB×2, SATA2 SSD 160 GB+128 GB. [5] HDD 500 GB×2, SATA2 SSD 128 GB×2. [6] HDD 1 TB+750 GB. [7] HDD 2 TB ×2. [8] HDD 500 GB×2. [9] HDD 1 TB×4 (RAID5). [10] HDD 2 TB×4 (RAID5). [11] SATA2 SSD 128 GB× 4 (RAID5). [12] HDD 2 TB×2, SATA3 SSD 128 GB2.

は表 2 の脚注に示すように 500 GB から 2 TB である．SSD は SATA2 インターフェイスと SATA3 インターフェイスのもの 2 種類を利用している．

HDFS のネームノードサーバーと MapReduce のジョブトラッカーサーバは jvot 上で動作させた．それ以外の計算機ではデータノードサーバとタスクトラッカーを動作させた．一つのファイルが複数の計算機に分割されないよう，ブロックサイズはマップタスクが処理するファイルよりも大きい値である 256 MB に設定した．個々のマップタスクに割り当てるメモリサイズは，システム全体でメモリ不足にならないよう計算機毎に調節した．なお利用されるメモリサイズの最大値はクロスマッチ処理で 1 GB，全天検索で 400 MB であった．

クロスマッチ処理を行うデータは全部で 200 億レコード，圧縮前のデータサイズで 2TB,gzip による圧縮後は 260 GB である．クロスマッチ処理は天球上の限られた範囲だけで行えばよいので，天球の部分領域毎にファイルを分割し，マップタスクはその 1 ファイルのみに対してクロスマッチ処理を行えばよいようにファイルを用意した．

天球の領域分割は HTM インデック法 [8] で定義されている方法により 32768 の領域に分割した．これは HTM インデックスレベル 6 の分割に相当する．領域の境界にある天体は座標値の誤差も考慮し，誤差円がオーバーラップする部分領域全てに分配した．なお，通常用いられる緯度・経度による天球座標系では，緯度 90 度で経度が不定になる（即ち，北極点と南極点は特異点になる）という問題があるが，HTM インデックスを導入することによりこの特異点問題を回避できる．

ファイルは gzip した状態で HDFS へ登録した．ファイルのフォーマットは図 2 に示すようにパイプ文字 “|” 区切りとなっている．各カラムは左から，ファイル内レコード番号，カタログ名，オリジナルカタログにおける天体識別子，赤経・赤緯（天球上の緯度と経度），観測波長域名，中心波長，波長の単位，明るさ，明るさの誤差，明るさの単位，Jy (=1×10^{-26}W/m^2/Hz) 単位での明るさ，レベル 6 の HTM ID，レベル 18 の HTM ID である．このクロスマッチ処理では，マッチしたレコードの番号リストを出力する．この番号リストは，これとは独立なジョブとして実行される全天検索において，天球分割された個々の領域毎に利用されるので，一つにまとめる必要はなくリデュースタスクは行わない．

```
...
29936|sdss|587731511532453930|19.722875|-0.872348|u'|0.358500|um|21.173000|0.341000|mag|0.000013|32910|552147754841
29937|sdss|587731511532453930|19.722875|-0.872348|g'|0.485800|um|24.163000|2.234000|mag|0.000001|32910|552147754841
29938|sdss|587731511532453930|19.722875|-0.872348|r'|0.629000|um|21.362000|0.326000|mag|0.000010|32910|552147754841
29939|sdss|587731511532453930|19.722875|-0.872348|i'|0.770600|um|21.993000|1.157000|mag|0.000006|32910|552147754841
29940|sdss|587731511532453930|19.722875|-0.872348|z'|0.922200|um|20.980000|0.952000|mag|0.000014|32910|552147754841
...
```

図 2: クロスマッチ処理の入力ファイル例

全天検索では，このマッチしたレコードの番号リストをもとに同一の天体に対応するレコードの組を判断しながら，複数の波長にまたがる条件検索，たとえば z'−J >3.0 といった条件判断を行う．ここで，z' と J はそれぞれ z' バンド，J バンドの明るさである．そのため，この番号リストを分割カタログの先頭にヘッダーとして追記したファイルを用意し，それを全天検索の入力ファイルとした．また，全天検索ではクロスマッチ処理ほど一つのファイルに対する計算時間はかからないため，16 領域のデータをまとめて一つの tar ファイルとして HDFS に登録し，マップタスクはその tar ファイル一つに対して処理を行うようにした．リデュースタスクでは，検索条件にマッチしたすべてのデータを一つのファイルにまとめる処理を行った．

2.4　全天検索システムのユーザインターフェイス

今回構築した全天検索システムを JVO ポータルから利用するための，ユーザインターフェイスを作成した．図 3 に検索条件の入力画面を示す．入力する条件は，利用したいカタログの選択，各波長域での明るさまたは非検出の設定，そして色指標値（天文学では等級の差を「色指標」と呼ぶ）の範囲である．非検出については，カタログがカバーしている天球領域であるにもかかわらず，データが存在しないという条件で検索が行われている．明るい天体があるためにマスクされていたり，データ異常等によりデータ欠損がある場合でも非検出として検索されてしまうが，そういったデータは画像データに戻って確認する必要がある．

検索に要する時間は，可視光でのサーベイである Sloan Digital Sky Survey (SDSS) と近赤外 2 m での全天サーベイである TWOMASS カタログを利用する場合で約 30 分である．検索中は実行状況を示す天球マップが表示され（図 4），検索済みの領域を色付している．データが見付かった領域は赤で示してある．検索結果はユーザ用の個人ストレージ領域に保存さ

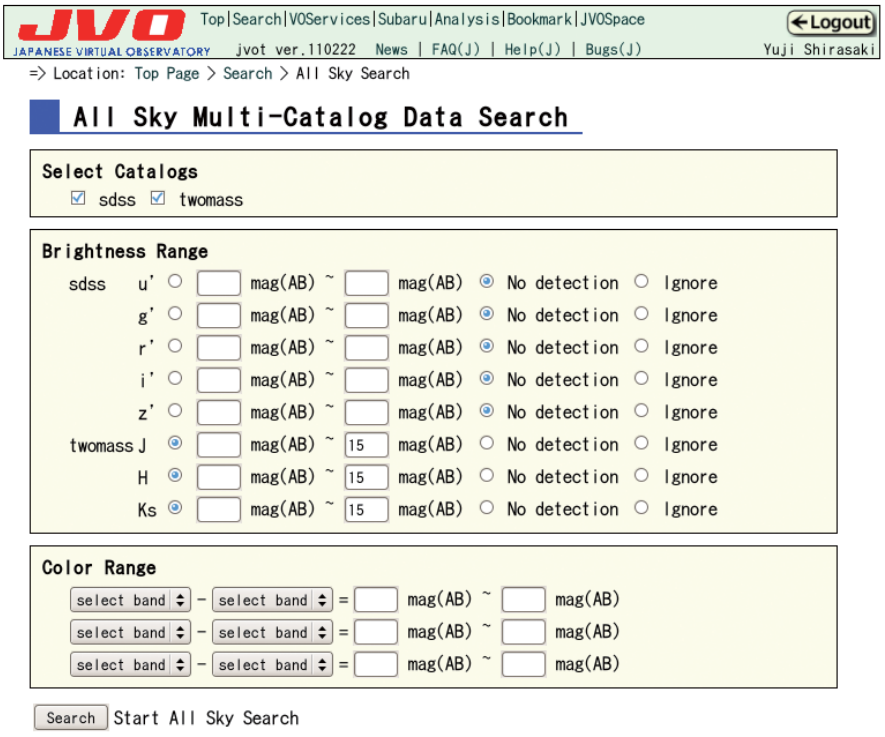

図 3: 全天検索システムの検索条件入力画面

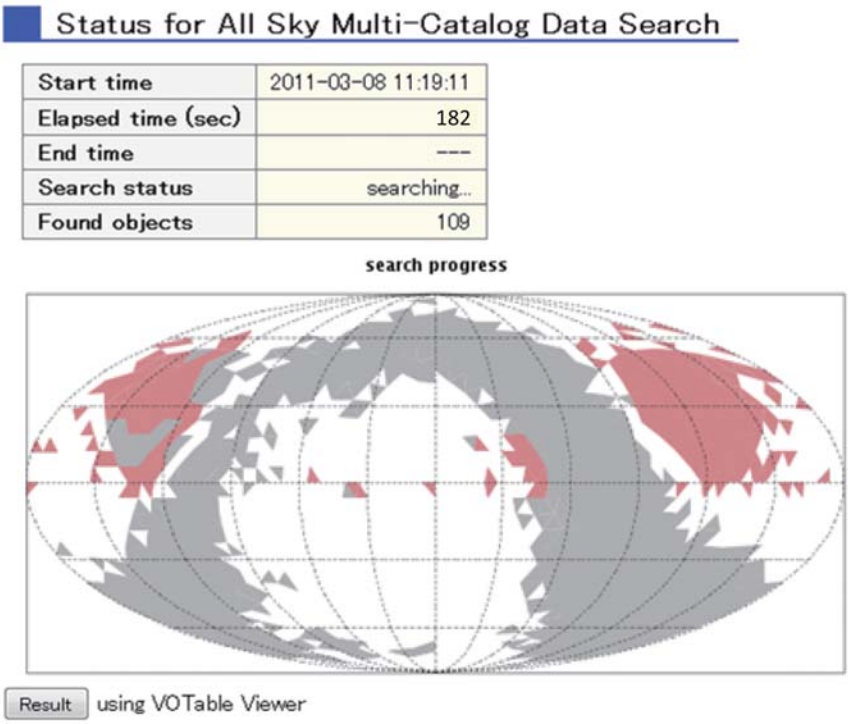

図 4: 全天検索実行中のステータス画面

れるので，後からいつでも結果を参照することが可能である．検索実行中でも，Result ボタンをクリックすることで，途中結果を見ることができる．

3 システムの性能評価

3.1 単体性能試験

まず計算機単体での性能を評価を行った．計算性能を調べるために，円周率計算プログラムを動作させて計算時間の測定を行った．測定は搭載されている CPU コア数を上限として，同時実行数を 1，2，4，8 と変化させ，全てのプログラムが終了するまでの時間を計測した．その結果を表 3 に示す．Xeon C システムでは同時実行スレッド数の増加とともに計算時

表 3: 各システム計算性能試験結果．円周率計算プログラムの実行時間（秒）の比較．

ホスト名	システム名	スレッド数 1	2	4	8
grid22	Core2 A	454	453	457	
grid30	Core2 B	503	503	506	
grid60	Xeon A	624	627	625	626
grid70	Xeon B	637	636	642	668
grid80	Xeon C	517	554	744	
grid90	Athlon	631	630	634	

表 4: 記録装置別 I/O 性能試験結果．単位は MB/s.

ホスト名	記録装置種別	逐次書き込み	逐次読み込み
grid21	HDD×1	55	55
grid22	HDD×2 LVM	49	59
grid41	SSD×1 SATA2	100	214
grid60	HDD×4 RAID5	22	95
grid70	HDD×4 RAID5	17	277
grid80	SSD×4 RAID5	53	291
grid90	SSD×1 SATA3	130	341

間が長くなっていることが確認された．1 スレッド実行時に比べ，4 スレッド同時実行時では約 50% ほど処理性能が低下している．Xeon B システムにおいても若干であるが，8 スレッド実行時に 5% 程度の性能低下が認められる．その他のシステムについては，コア数に等しいスレッド数までは性能低下は認められなかった．

記録装置の I/O 性能について，bonnie++[5] を利用して測定を行った．結果を表 4 に示す．HDD 単体の場合の読み込み性能は 50 MB/s であるのに対し，SSD (SATA3) 単体の場合は 340 MB/s と大幅な性能向上が認められる．大量のデータを読み書きし，ディスク I/O の占有率が高いアプリケーションの場合には SSD を利用することで大幅な性能向上が期待できる．本測定によると，RAID 化によっても読み込み性能は向上しているが，bonnie++ による性能測定は 1 スレッドによる結果であり，複数スレッドが同時に I/O を行う場合には性能低下が見られるはずである．grid60 と grid70 は利用している RAID コントローラの違いに起因する性能差が出ている．

3.2 Hadoop 性能試験

次に，実際に Hadoop を利用して全天検索を実行し，Hadoop の性能試験を行った結果について報告する．

試験は各計算機で同時実行できるタスク数の上限を変化させて，データの入出力に要した総時間，マップタスクの総実行時間，ジョブの実行開始から終了までの経過時間を測定した．測定は Java の System.nanoTime() メソッドを使用し，データ入力を行うメソッドの実行前から実行後までの経過時間と，マップタスク関数内の先頭行から最終行までの経過時間を測定し，その合計を計算した．同時に，入力データがどのホストから呼ばれているのかを出力するよう Hadoop のソースコードを変更し，どれだけの割合で他計算機からデータ転送が行われているのかを調査した．同時実行タスク数は接続している記録装置 (HDD/SSD) 数の 2 倍を越えない範囲で CPU コア数を上限とし 1，2，4，8 と変化させた．

タスク数 1 の場合の結果から，マップタスクの開始時刻と終了時刻のタイムスタンプをもとにジョブの実行間隔を測定した．図 5 に grid80 におけるマップタスクの終了時刻から次のタスクの開始時刻まで経過時間の分布を示す．図から明らかなように，約 1 秒を単位として離散的に分布しており，このタイミングでタスクが実行可能かを定期的にポーリングしていることが分かる．平均のジョブ実行までの経過時間は約 3 秒である．

図 6 に同時実行タスク数に対するジョブの実行実時間を示す．タスク数 66 すなわち，各計算機での同時実行タスク数を 2 とした場合までは，タスク数に反比例して実行実時間は短縮していることが分かる．また，タスク数 137 以上では，タス

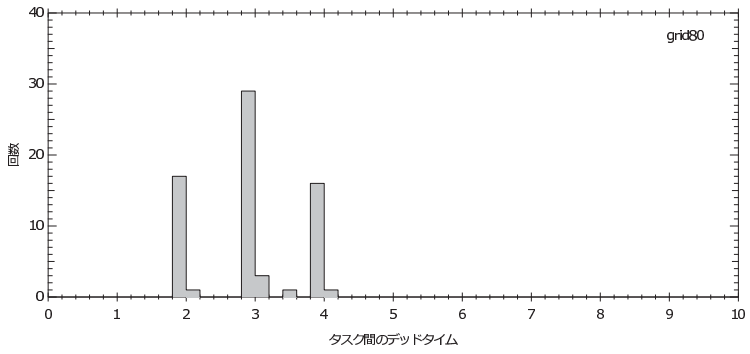

図 5: タスクの実行終了から次のタスクの実行開始までの遅れ時間の分布.

表 5: 全天検索システムの性能評価試験結果

試験名	同時タスク数	I/O 総時間	マップタスク総時間	ジョブ実行時間	ローカルデータ実行率
Search33[1]	33	86237	261001	8130	87%
Search66[2]	66	81663	255234	4132	85%
Search137[3]	137	87615	277624	2145	88%
Search154[4]	154	90135	288421	1970	88%

[1] 全ノードでタスク数の上限を 1 に設定．[2] 全ノードでタスク数の上限を 2 に設定．[3] タスク数の上限を記憶装置数に設定．[4] タスク数の上限を記憶装置数の 2 倍または CPU コア数を越えない最大値に設定.

[5] http://www.coker.com.au/bonnie++/

ク数に対し線形にスケールしなくなっていることが分かる．

図 7 に各タスクの実行時間の合計と I/O に要した時間の合計を同時タスク数に対してプロットした．I/O に要した時間はタスク数 154 まで大きな変化は見られず，変化率は 5% 以下である．一方，タスクの総実行時間はタスク数 137 以上で増加しており，タスク数 154 の場合で 10% の増加が見られた．このことから，タスク数が多い場合に実行実時間が期待値より大きくなる原因として，ディスク I/O 以上に，スレッド間での干渉による計算性能の低下，例えば「メモリーウォール問題」

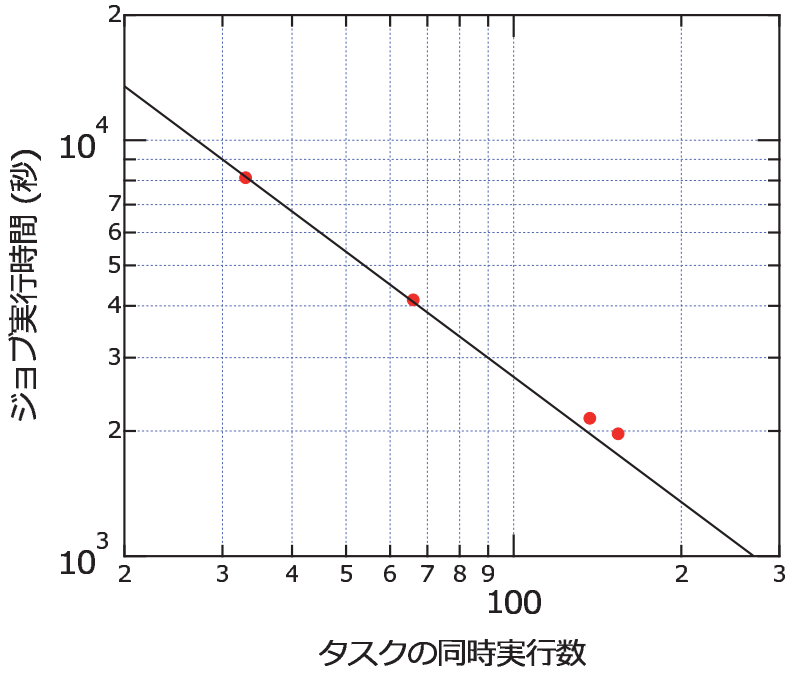

図 6: タスクの同時実行数に対するジョブの実行時間

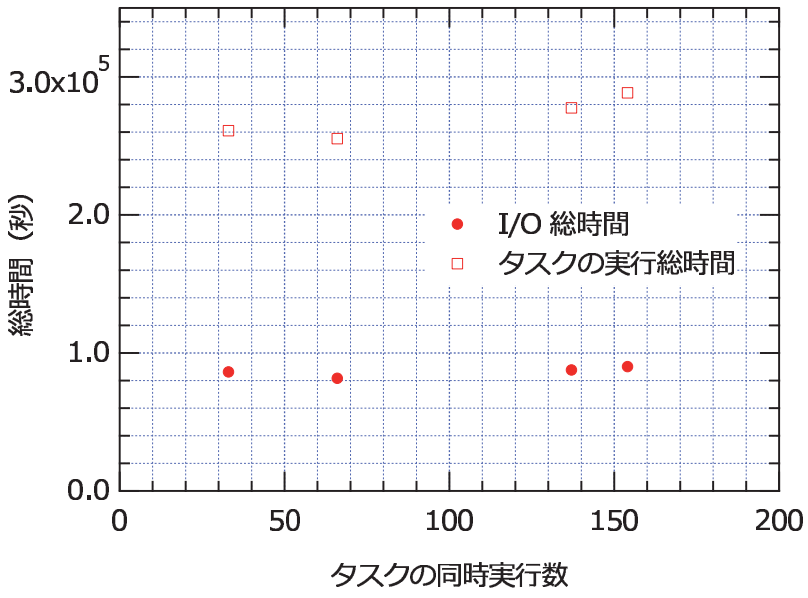

図 7: タスクの同時実行数に対する I/O 総時間とタスクの総実行時間

として良く知られている，CPU の性能にメモリの性能が追いつかなくなる問題や，タスクマネージャスレッド等のシステム上で動作している他のジョブとの干渉等の方が大きく寄与していると考えられる．

ローカルデータの利用率はいずれの試験でも約 90% であった．リモート計算機からのデータ転送が発生するケースはジョブの開始直後に集中しており，それ以降はほぼ 100% のローカルデータ実行率が達成されていた．最初のジョブ実行スケジューリングがローカルデータ実行に最適化されていないようである．

4　今後の展望と課題

現在国立天文台では，すばる望遠鏡に搭載予定の次世代広視野撮像カメラである HyperSuprime-CAM (HSC) の開発や日米欧による共同国際プロジェクトである Atacama Large Millimeter/submillimeter Array (ALMA) の建設が進行している．HSC は，約 2 度というかつてない広視野を撮像できるカメラであり，カメラには 2k×4k 素子を持つ CCD176 枚が収められる．従って，HSC は 1 回の撮像毎に 2.8 GB もの大きさを持つ画像（群）が生み出される．HSC 開発の大きな目的は，ダークエネルギーを始めとする現代天文学の謎を解明することにあり，そのためには長期に及ぶサーベイ観測で得られる大量データの質を可能な限り均質に保たなければならない．即ち，各 CCD からの出力を並列パイプライン処理する必要があり，本研究で得られた並列データ解析システムは，HSC データ解析パイプラインに応用することも可能であると考えられる．

また ALMA は，最終的には 66 台のアンテナを建設することとなっているが，2011 年後半には 16 台のアンテナを用いた Early Science が始まることとなっている．ALMA は電波干渉計システムであるため，空間方向 2 次元＋周波数（波長）方向 1 次元の 3 次元データキューブを生み出す．ALMA が full operation を迎えると，そのデータ産出量は年間 1PB 弱と予想され，そのデータは ALMA アーカイブシステムからだけではなくヴァーチャル天文台用インターフェースを介して全世界の研究者に公開されることとなっている．従って世界の ALMA ユーザーが ALMA のデータを並行して処理することが容易に考えられ，その処理をまかなうシステムにも，本研究で得られた成果が応用できるであろう．

一方，本研究で明かになったように，並列処理の効率を高めるためにはスレッド間の干渉を可能な限り排除するようにデータ処理システムを設計することが肝要である．従って, HSC や ALMA のデータ処理に本研究成果を応用するためには，それぞれの科学的目的やそのための要求要件を science use cases としてまとめ，その use cases を満たすように実運用のためのハードウエア設計や並列処理システムの設計を進める必要があると考えられる．

5　結論

我々は，データ生産量が爆発的に増えている天文学の要請に応えるため，全天対応並列データ検索・解析システムを Hadoop を用いて試験構築した．性能測定の結果，試験構築に用いた計算機群のコア数とほぼ等しいタスク数になる以前に実効性能が劣化することが分かった．この原因としては，ディスク I/O によるボトルネックよりも，複数タスクを並列動作させたことによるタスク間の干渉，例えば「メモリーウォール問題」が発生するためであることが分かり，実運用システムを構築するために極めて有益な知見が得られた．

本研究は天文学におけるデータ検索や解析を対象としたものであるが，世界では様々な科学分野において世界中のデータを連携させようとする “Virtual Observatory” 構想が進んでおり，本研究で得られた知見は，これらの “Virtual Observatory” を構築する際の有益な参考情報を与えるものと考えられる．

acknowledgment

本研究は，文部科学省科学研究費補助金特定領域研究「情報爆発」公募研究 (18049074，19024070, 及び 21013048) の支援により実施された．また，研究にあたり様々な支援をいただいた国立天文台天文データセンターのスタッフの方々に深く感謝致します．

参考文献

[1] 田中昌宏ほか : 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 (2004)

[2] 本田敏志ほか : 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 (2005)

[3] Shirasaki, Y. et al.: Japanese Virtual Observatory (JVO) as an advanced astronomical research environment, *Proc. of the SPIE, Advanced Software and Control for Astronomy*, Edited by Lewis, Hilton; Bridger, Alan., Vol. 6274, pp. 62741D (2006)

[4] Shirasaki, Y. et al.: The Japanese Virtual Observatory in Action, ASP Conference Series, Vol. 411, Proc. of *ADASS XVIII*, Edited by David A. Bohlender, Daniel Durand, and Patrick Dowler, pp. 396 (2009)

[5] Tanaka, M. et al.:Construction of Multiple-Catalog Database for JVO, ASP Conference Series, Vol. 394, Proc. of *ADASS XVII*, Edited by Robert W. Argyle, Peter S. Bunclark, and James R. Lewis., pp, 261 (2008)

[6] 田中昌宏ほか：膨大な天体データを効率的に検索する方法の考察と実装，DEWS 2008, C9-3 (2008)

[7] Dean J. and Ghemawat S.: MapReduce: Simplified Data Processing on Large Clusters Appeared in: OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004. http://labs.google.com/papers/mapreduce.html (2004)

[8] Kunszt, P. Z., Szalay, A. S., Thakar, A. R., 2001, *Mining the Sky: Proceedings of the MPA/ESO/MPE Workshop, ESO Astrophysics Symposia*, Edited by A.J. Banday, S. Zaroubi, and M. Bartelmann. pp. 631 (2001)