様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　３０　年　　５　月　２１　日現在

機関番号：　３７１１１
研究種目：　若手研究(B)
研究期間：　2015～2017
課題番号：　１５Ｋ１７５０１

研究課題名（和文）超巨大天文データからなる全天アーカイブをHadoopにより超低費用で実現する研究

研究課題名（英文）Preliminary study of an inexpensive implementation methodology for an all-sky oriented astronomical data archive system powered by Hadoop for huge observational multi-wavelength data set

研究代表者

江口　智士（Eguchi, Satoshi）

福岡大学・理学部・助教

研究者番号：４０６４７２０２

交付決定額（研究期間全体）：（直接経費）　　1,500,000円

研究成果の概要（和文）：望遠鏡や観測装置の巨大化・複雑化により、そこから得られる天文データが指数関数的に増加している。この増加に安価に対処する方法として民間のクラウド・コンピューティング・サービスに目を付け、分散処理フレームワークHadoopおよびその上で動くHiveにより巨大な天文データを処理する方法論に関して基礎的な調査・開発を行った。その時々の処理内容に応じて必要な計算機資源時間貸しするサービスを利用し、Hiveのパーティショニングにレベル6のHEALPixを、Hiveのエンジンにtezを、データベースのファイル形式にORC形式を使用するのが最適であるという結論を得た。

研究成果の概要（英文）：The size of astronomical observational data has been exponentially inflating due to the hugeness and complexity of modern telescopes and their instruments. Public cloud computing would be an attractive solution for the data explosion to astronomers without sufficient financial support thanks to its flexibility and inexpensiveness. To this end, I investigated the feasibility through the implementation of a simple astronomical database running on Hadoop, a software framework for distributed computing, and Hive, Hadoop-based database software with an SQL-like query language. I found that we should (1) choose clouds enabling the users to get arbitrary computing resources depending on the complexity of problems at the time, instead of virtual private servers providing limited and fixed resources at a lower annual cost, (2) adopt the Tez engine and the ORC file format for the Hive database, (3) partition the Hive table based on the level 6 HEALPix labeling.

研究分野：　データベース天文学

キーワード：　バーチャル天文台　ビッグデータ　分散コンピューティング　クラウドコンピューティング　Hadoop　Hive

１．研究開始当初の背景

観測装置の大型化・複雑化に伴い、天文学の観測データは爆発的に巨大化している。このような巨大データの処理には High Performance Computing (HPC)が必須だが、我々のような地方大学の一研究がそのような環境を自前で用意することは不可能である。いっぽう、コンピュータの仮想化技術の進歩により、HPC などの強力な計算資源を分割化してインターネット経由で時間貸しする、いわゆる「クラウド・コンピューティング（以下『クラウド』と表記）」が本格的に普及し始めた。クラウドを活用することができれば、大学の研究室単位でも、肥大化する天文データの処理に必要な計算資源を低予算で調達可能になると考えられる。

クラウドの普及と並行して、複数の計算機資源をインターネット越しに束ねる分散並列処理の技術も成熟してきた。その中でも分散処理フレームワーク「Hadoop」は、「パソコン」のような信頼性の低い計算機を大量に結ぶことで、ストレージ容量・ストレージ性能・計算能力・信頼性を確保するというコンセプトのもとに開発されている。Hadoop はクラウドとの相性も良く、大量のデータを解析処理する必要がある地理情報システム(GIS)の実装にも使われている[1]。

２．研究の目的

一口にクラウドと言っても、提供形態は様々である。特に計算機資源の貸出期間に注目すると、比較的低性能かつ固定容量の資源（仮想サーバ）を半年～数年単位で安価に貸し出す Virtual Private Server (VPS)と、Amazon Web Services (AWS)のような、動作するアプリケーション・セットを予め限定する代わりに計算資源を利用者のその時々の要求に応じて動的に 1 時間単位で貸し出すものとがある。そこで本研究では、Hadoop 上で簡易的な天文データの保管・検索システムを実際に実装することを通じて、(1)セットアップの容易さなども考慮した場合、どちらのタイプのクラウドがより適しているのかを調査し、(2)10 年後に運用を開始するであろう天文データ・アーカイブ・システムの実装に必要となるノウハウを手に入れることを研究目標とした。

３．研究の方法

(1) Hadoop 上で動作する分散データベース・システム Hive の利用

Hadoop 上のデータ処理は Map タスクと Reduce タスクという 2 つの構成要素からなる。巨大データとその加工処理は小さなタスク(Map タスク)に分割され、各 Map タスクによる出力を Reduce タスクで一つの処理結果に集約する。既存の天文データ・アーカイブ・システムのほとんどは SQL 言語を理解するリレーショナル・データベース（RDBMS）で実装されており、これらソフトウェア・リソースを活用するためには、SQL に近い言語(Hive QL)で記述された処理を Hadoop の Map/Reduce タスクに展開してくれるデータベース・エンジン「Hive」を利用するのが合理的である。また、Hive はタブ区切りテキストやカンマ区切りテキストを直接扱えるため、様々なソフトウェアとのデータのやりとりが簡単に行えるという利点がある。しかも Java で書かれているためハードウェア依存性が少なく、機能拡張を Java プログラミングで行うことができる。本研究では Hive を用いてデータベースの実装を行った。なお以下では、特に断らない限り、テスト・データに「2MASS Catalog Server Kit」[2]に付属する「2MASS PSC カタログ・データ」(470,992,970 レコード、174 GB）を使用した。

(2)天球分割アルゴリズムの比較と Hive のパーティショニング

Hadoop クラスタ上に投入されたデータは、HDFS という分散ファイルシステムに小さな断片（デフォルトでは 64 MB）に分割されて保管される。Hive 上でのデータ検索は、この小片を並列かつシーケンシャルに走査することで行われる。従って何もしなければ、検索条件にマッチするデータ（レコード）が僅かであっても、HDFS 全体をスキャンすることになる。これでは性能が全然出ないので、データを小さな「ある単位」（パーティション）に分割し、検索時にはそのパーティション内のデータのみを走査すれば良いようにする必要がある。通常の RDBMS ではデータの場所が非常に限られたハードウェアに限定されるため、「インデックス」と呼ばれるハッシュ値を利用してデータベース全体を高速にスキャンできるのに対し、Hadoop/Hive ではデータが多数の複数のハードウェアに分断されるため、パーティションという仕組みが必要になるのである。従って、Hadoop/Hive のポテンシャルを最大限引き出すためには、パーティションの作り方（パーティショニング）が鍵を握る。

通常の天文学の観測データであれば、それは何らかの天体に付随した現象を記録したものであるので、データの識別（あるいは整理）にその天体の天球座標を用いることができる。つまりパーティション設計の第一歩として、天球を（比較的）大きな単位で分割することを考える。特に HEALPix と呼ばれるアルゴリズムは、天球を等立体角のピクセルに分割して通し番号を付与するものであるが、ピクセル数を「レベル」というパラメータで制御でき（レベルが高いほどピクセルが細かくなる）、高レベルの（複数の小さい）ピクセルが低レベルのひとつのピクセルの中に内包されるという入れ子関係になっているため、Hive のパーティション作成に最適であると言える。しかしながら真っ正直に球面を等立体角に分割するため、天球座標から HEALPix のインデックスを算出する際に複雑

な三角関数の演算が必要になり、その計算でデータベースの検索性能が律速される可能性がある。そこで HEALPix の代わりになるアルゴリズムとして、天球に外接する立方体を考え、この立方体を 3 次元的に小さな賽の目に区切り、それらに**巧いことラベリングする**こと（8 分木および 3 次元のモートン順序の導入）で、天球座標からインデックスを求める作業を（古典的な）ゲームにおける衝突判定の問題に落とし込み高速化を図る方法を考案した。実際に HEALPix および 3 次元衝突判定法を Hive の拡張機能として実装し、両者の性能比較を行った。

(3)VPS サービスと AWS の比較

Hive をクラウドとして使用する方法として、(1)VPS のプロバイダと契約して複数の VPS を購入し、自分ですべてのセットアップ作業を行う方法と、(2) AWS の Hive インスタンスを購入する（=セットアップは自分では行わない）という選択肢がある。前者は性能が低くセットアップの手間はかかるが、安価であるため、もし実用的な（妥協可能な）性能が手に入るのであれば、魅力的な選択肢となり得る。適切なクラウドの形態を見極めるため、データ検索の時間の安定性に注目して、GMO クラウド社の「GMO クラウド VPS スモール（4 コア CPU、4 GB RAM、200 GB HDD)」と AWS の「m3.xlarge インスタンス(4 コア CPU、15 GB RAM)」の性能比較を行った。

(4) Hive のデータベース・エンジンおよび内部ファイル形式の選択による検索性能の変化の測定

Hive を Apache Hive のサイトからダウンロードしてインストールした直後は、Hive のタスクを Hadoop のタスクに変換するエンジンに MapReduce が設定されている。MapReduce では、各 Map タスクと Reduce タスクの間のデータのやりとりに HDFS 上のファイルを利用する。これは単にディスク・アクセスが生じることにより処理性能低下が起こるだけでなく、Hadoop ノード間のインターネット越しのデータ通信が相当数発生することによるさらなる性能低下が起こることを意味する。そこでディスク・アクセスを最小限に抑えつつ Hive の実行性能を向上させる「Tez」と呼ばれるエンジンを選択することもできる。ただし Tez を有効化するためには追加のライブラリのコンパイルとセットアップが必要になる。これは VPS をベースに自力で Hadoop クラスタを構築している場合に手間となるが、AWS ではインスタンスの起動オプションにパラメータをひとつ追加するだけで最初から使用できるようになっている。本研究では、VPS で標準の MapReduce を使用した場合と Tez を使用した場合で、どれくらい性能に差が出るかを調べた。

いっぽうで、Hive の性能を向上させる他のオプションとして、Hive の HDFS 上でのデータ形式が挙げられる。何も指定しない場合は単なるテキスト・ファイルとなり、データ検索の際は各行をその都度パースしながら対象のカラムのデータを抽出するが、「Optimized Row Columnar (ORC)」と呼ばれる形式を指定した場合、データはカラム単位でバイナリ形式に符号化され圧縮される。従って、ORC 形式を選択した場合、Hive は Hive QL を理解するカラム指向データベースのように振る舞い、通常の RDBMS を操作する感覚でビッグ・データを高速に処理できるようになる。そこで、通常のテキスト形式と ORC 形式とで検索性能がどれくらい違うのか調べるために、ベンチマークを行った。

(5) JVO ALMA アーカイブで公開されている巨大 FITS データの Hive への登録

国立天文台の Japanese Virtual Observatory (JVO)プロジェクトでは、世界最大の電波望遠鏡 ALMA で観測されたデータに標準較正を施したものを、3 次元のイメージ FITS ファイルとして配布している（正確には ALMA の Web ページでも配布されているが、JVO 経由の方がアクセスが簡単にできる)。ALMA は空間分解能や波長分解能が非常に高いため、観測データは他の望遠鏡による観測データよりも大きくなるが、2017 年には 1 観測 4 GB を超える（最大約 25 GB）FITS データの配信が始まった。そこでここまでに得られた知見を元に、これらの巨大なイメージデータを AWS で動く Hive に格納し、検索可能な状態にするためのソフトウェア開発を行った。将来的には可視・赤外線望遠鏡から得られるデータも現在の ALMA 望遠鏡並に巨大化することを鑑み、様々な波長で観測されたデータを統一的に扱えるようにソフトウェアの設計を行った。特に重要な点として、可視・赤外線望遠鏡のデータと電波望遠鏡のデータとでは、天球を平面に射影する方法が異なる。そこで、天文学で使用する各種の複雑な座標変換を一手に引き受ける WCSLIB というライブラリを使用して、各 FITS イメージの各ピクセル座標→天球座標→（単一の）レベル=20 の HEALPix のピクセル座標と段階的に変換し、それを一旦テキスト・データとして出力し、それを AWS 上の Hive でパーティショニングを行いながら ORC 形式の Hive データベースへ登録した。この Hive への登録作業を処理ノード数を変えて行うことで、どの程度（自動的に）並列処理が行われているかを測定した。

４．研究成果

(1)僅かに高速な 3 次元衝突判定法、安定度の HEALPix

HEALPix のレベル=6（5×10^4 ピクセル)と 3 次元衝突判定法のレベル=6（3×10^5 セル）でパーティショニングを行ったテーブルそれぞれに対し、検索の中心座標を完全にランダムに決め（天球のどの座標も同じ確率で選択

される)、検索半径を 5 秒角から 5 分角の間で等確率に振り、その天球領域内のデータを検索するクエリを 5000 回投げ、その所要時間の分布を調べた（測定は研究代表者所有のワークステーションで行った)。平均値は HEALPix が 25.47 秒、3 次元衝突判定法が 25.02 秒とほぼ同等であったが、分布のピークは 3 次元衝突判定法の方が 2 秒ほど速かった。いっぽうで、分布の標準偏差を見ると HEALPix は 1.06 秒だったのに対し、3 次元衝突判定法は 9.24 秒だった。詳細にデータを見てみると、HEALPix は検索半径や結果のレコード数が変わっても 23〜30 秒の範囲に収まっていたが、3 次元衝突判定法では該当レコード数が 300 件未満の範囲で 23〜35 秒の範囲に広く分布していた（300 件以上では HEALPix と同等だった)。また、試行 5000 回に対して 10 回程度ではあるが、応答まで 300 秒以上を要しているケースがあった。当初のもくろみ通り HEALPix に比べ 3 次元衝突判定法の方が演算が高速であるいっぽう、検索条件によっては 8 分木の根元に近いところからの探索になり、走査対象となるパーティションが無駄に多くなると考えられる。複数の立方体を用意して互いに 45 度回転させ、それぞれの立方体でのインデックスを計算して保持する等すれば、3 次元衝突判定法の演算が単純であるというメリットを活かせる可能性がある。

以下ではパーティショニング以外の要素によるパフォーマンス測定を行う関係上、パーティショニングには HEALPix を用いる。

(2)パフォーマンス変動の多い VPS、パフォーマンスに変動のない AWS

Hadoop クラスタは、実際にデータ処理を行うデータ・ノードとデータがどのノードにアルかを管理するネーム・ノードから構成される。GMO クラウド VPS で 1 ネーム・ノード+7 データ・ノードの Hadoop/Hive クラスタを構築し、研究代表者のワークステーションで予めパーティショニングを行ったデータを VPS へ転送し、(1)と同じ方法によるクエリの処理時間の測定を、日を変えて行った。VPS ではメモリ容量がかなり制限されるため、HEALPix のレベルは 3 とした。同様に、AWS で 1 ネーム・ノード+3 データ・ノードの Hive クラスタを構築し（測定ごとに新規にインスタンスを構築)、インスタンス間でのパフォーマンスの差を計測した。このときセットアップの関係で VPS は Tez が無効、AWS は Tez が有効という差と、ノード数の差があるが、処理時間の日々のばらつきを見ることが目的のため、これらの差は結論には影響しないと考える。VPS では処理時間の分布の平均値も分散も日ごとに異なり、10%程度ばらついたが、AWS はどのインスタンスであっても処理時間の分布自体に再現性があり、平均値のばらつきは 3%未満であった。クラスタを構築する際に必要なノード数が確実に予測できること、セットアップが容易であることから、天文データベースの構築という目的には AWS が合致していると考える。

この方向で、同じクラスタ構成でパーティショニングの HEALPix のレベルを変化させた場合に、パフォーマンスがどのように変動するのか測定した。レベル=3 からレベル=6 まで 1 刻みで変化させた場合（レベル=7 以上はメモリ不足のため測定不能であった)、前者に比べ後者では平均の処理時間がほぼ半分になった。従って、以下ではレベル=6 に固定した。

(3)絶対に使用すべき Tez と ORC 形式

再びワークステーション内での作業に戻り、Hive のエンジンをデフォルトの MapReduce から Tez に変えた場合にどれだけ性能が向上するか調べた。測定方法は(1)と同じである。MapReduce では処理時間の分布の平均は 25 秒であったが、Tez では 9 秒になった。自分で Hadoop/Hive を一からセットアップする場合、Tez を有効にするためにはソースコードのコンパイルが必要で手間がかかるが、クエリのチューニングを行わなくても性能が 3 倍近く向上するメリットは大きい。AWS ではオプションに 1 パラメータ追加するだけなので、使わない理由がない。よって以下では Tez を有効にした。

続いて Hive のファイル形式にテキスト形式を用いた場合と ORC 形式を用いた場合とで、性能がどれだけ変化するのか調べた。純粋にファイル形式による性能の違いを見るために、パーティショニングをしないまま 2MASS カタログをテーブルに登録し、「SELECT COUNT(*) FROM (テーブル名)」に掛かる時間を測定した。テキスト形式では 750 秒程度掛かっていたが、ORC 形式では 60 秒以下になった。Map タスク数に注目すると、テキスト形式では 204 であったのに対し、ORC 形式では 49 だった。つまり ORC 形式へ変えたことによる性能向上（12 倍）のうち、4 倍分は ORC 形式によるデータ圧縮の効果（パーティショニングされていないテーブルの処理は、あるサイズごとにひとつの Map タスクが割り当てられる）で、残りはカラム指向のデータ構造にしたことによる性能向上と推定される。よって ORC 形式を積極的に採用すべきと結論した。

(4)多波長・巨大天文データ検索用アーカイブ実装に向けて〜ALMA 望遠鏡による巨大 FITS イメージの登録処理から見えたこと

ここまでに得られた知見をまとめると、（1）クラウドには VPS ではなく AWS を使う、（2）パーティショニングのアルゴリズムにはレベル=6 の HEALPix を用いる、（3）Tez と ORC 形式を使用する、となる。最後にこれらを組み合わせ、簡単な多波長・巨大データ・アーカイブの実装を行ってみた。研究期間の制限から、システムのバックエンドにのみ注目して、単純なコーン・サーチ（中心座標、

半径および波長範囲を指定して、その中に入る画像イメージを取り出す）を行うシステムを考える。ALMA 望遠鏡による 3 次元イメージ FITS の中でも特に大きな 4 つ（合計 43 GB）をこのシステムに登録した。（ALMA 以外のものも含む）様々な観測機器で得られた画像を横断的に検索することを考え、各 FITS イメージの各ピクセル値（(赤経，赤緯，波長)からなる 3 次元のピクセル）をレベル=20 の HEALPix のメッシュ（角度分解能=0.2 秒角）で再サンプリングを行い、FITS に含まれる各種キーワードとともに「(3 次元的な）1 ピクセル=1 レコード」という関係で SQLite3 ファイルへと変換した。この際オリジナルのピクセルと HEALPix のピクセルとで形状が違うことも考慮した。SQLite3 のテーブルを HEALPix のインデックス順にソートした後、タブ区切りテキスト・ファイルに出力した。このテキスト・ファイルを AWS のストレージにアップロードし、AWS の Hive クラスタで HEALPix のレベル=6 のパーティショニングを行いながら、ORC 形式の Hive データベースへと変換した。この作業をクラスタのデータ・ノード数を 1 から 15 までの範囲で変えて行い、その結果をアムダールの法則[3]でフィットし、Hive データベースへの登録作業の並列化の度合いを測定した。途中タブ区切りテキスト・ファイルの段階でファイル・サイズが 670 GB まで膨らんだが、最終的な ORC 形式の Hive データベースのサイズは 25 GB で済み、並列度は 65%であった。つまり、理論的にはデータ・ノード数を無限に増やしても性能向上は 3 倍程度ということになる。注意すべきは、これはデータのデータベースへの登録時の性能向上に関してであり、データ検索の性能については、また違った値になると思われる。

以上のように、本研究では、「巨大天文データをクラウドで捌く」という、これまで誰も考えなかった切り口で、将来（今後 10 年以内）の天文データのサイズ爆発という問題への対処法の一端を示した。自分で管理する HPC とは異なり、クラウドには自身でコントロールできない最適化パラメータがいくつもあり、それらを順番に絞り込んでいった過程が本研究である。天文データの検索クエリは、昨今の「ビッグ・データ」の中でも比較的シンプルである。それでもなお、今後様々な分野で起こるデータのサイズ爆発の問題に対し、限られた経済的資源で解決しようとする際、今回得られた知見がきっと役立つに違いない。本報告書の内容を査読論文というきちんとした形にまとめ上げ、同様の問題に悩む人たちの道しるべとしたい。それが今筆者に課せられた責務だと思う。

＜引用文献＞

[1] Aji et al., Hadoop-GIS: A High Performance Spatial Data Warehousing System over MapReduce, Proceedings of the VLDB Endowment, Volume 6 Issue 11, 2013, 1009-1020
[2] https://www.ir.isas.jaxa.jp/~cyamauch/2masskit/index.ja.html
[3] Amdahl, Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities, AFIPS Conference Proceedings (30), 483-48

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 2 件）

①Satoshi Eguchi、Pre-feasibility Study of Astronomical Data Archive Systems Powered by Public Cloud Computing and Hadoop Hive、査読なし、2016、arXiv:1611.06039
https://arxiv.org/abs/1611.06039

②Satoshi Eguchi et al.、Blade Runner -What kind objects are there in the JVO ALMA Archive?- 、査読なし、2015、arXiv:1511.06533
https://arxiv.org/abs/1511.06533

〔学会発表〕（計 3 件）

①Satoshi Eguchi、Pre-feasibility Study of Astronomical Data Archive Systems Powered by Public Cloud Computing and Hadoop Hive、Astronomical Data Analysis Software and Systems XXVI、2016 年 10 月 16～20 日、トリエステ（イタリア）

②Satoshi Eguchi et al.、Blade Runner -What kind objects are there in the JVO ALMA Archive?- 、Astronomical Data Analysis Software and Systems XXV、2015 年 10 月 25～30 日、シドニー（オーストラリア）

③江口　智士他、VO ALMA アーカイブの天体同定(I)、日本天文学会 2015 年秋季年会、2015 年 9 月 9～11 日、甲南大学（兵庫県・神戸市）

〔その他〕
ホームページ等
https://www.cis.fukuoka-u.ac.jp/~satoshieguchi/

６．研究組織
(1)研究代表者
江口　智士（EGUCHI, Satoshi）
福岡大学・理学部物理科学科・助教
研究者番号：40647202