

IUGONET メタデータ登録・管理システムの処理性能評価

堀 智昭^{*1} 梅村宜生^{*1} 阿部 修司^{*2} 小山 幸伸^{*3} 田中 良昌^{*4} 林 寛生^{*5,*6}
上野 悟^{*7} 新堀 淳樹^{*5} 佐藤 由佳^{*4} 八木 学^{*8}

Performance study of IUGONET metadata management system

Tomoaki HORI^{*1}, Norio UMEMURA^{*1}, Shuji ABE^{*2}, Yukinobu KOYAMA^{*3}, Yoshimasa TANAKA^{*4},
Hiroo HAYASHI^{*5,*6}, Satoru UENO^{*7}, Atsuki SHINBORI^{*5}, Yuka SATO^{*4} and Manabu YAGI^{*8}

Abstract

This paper reports on the benchmark test of the metadata management system developed by the Inter-university Upper atmosphere Global Observation NETwork (IUGONET) project. The IUGONET metadata management system consists of the metadata reception module based on file repositories of Git, which is a widely-used version control software, and the metadata import module to register metadata to the metadata database using the DSpace commands. We made the performance assessments for these two modules in terms of the processing speed for registering/importing metadata both on a native Linux platform and a virtualized Linux platform. As a result, the metadata reception module processes properly even millions of registered metadata within several minutes at most, showing roughly the similar performance for the native and virtualized platform. On the other hand, the metadata import module needs much longer processing times of several to a few tens of hours for the same amount of metadata. As compared with the virtualized platform, the native platform gives a four times faster speed for updating indices for the built-in search engine based on Lucene. From a perspective of the regular operation of the metadata management system, the metadata import module runs more efficiently on the native Linux platform and it is estimated by our performance test that it takes about 5 hours to newly import 10^5 metadata files and subsequently takes about 21 hours to update the Lucene indices for the registered metadata of 5×10^6 in total.

Keyword : IUGONET, metadata, metadata management, Git, DSpace

概要

本論文では、大学間連携プロジェクト「超高層大気長期変動の全球地上ネットワーク観測・研究」(IUGONET)において、作成されたメタデータの受付やメタデータ・データベースへの登録を行っている「メタデータ登録・管理システム」のメタデータ登録処理に関する性能試験の結果について報告する。メタデータ登録・管理システムは、フリーのバージョン管理ソフトウェアである Git を利用したメタデータ受付部と、そこからメタデータ・データベースへメタデータを登録するメタデータインポート部の、2つの処理部で構成される。この2つの処理部の性能を評価するために、実際に作成・登録されたメタデータを用いて、物理マシンおよび仮想マシン上に構築した Linux 環境のそれぞれについて、メタデータの処理速度を計測した。その結果、Gitによるメタデータ受付部は物理マシンか仮想マシンかに依らず、メタデータ総数が500万件レベルまで増えても数分以内で処理が完了するのに対して、メタデータインポート部は処理時間が数時間から数十時間とかなり長くなり、特に検索エンジンである Lucene のインデックス更新については、物理マシンにおける実行速度が仮想マシンのそれより約4倍高速であることがわかった。メタデータインポート部を物理マシン上に実装した場合、メタデータ総数500万件では、新規登録メタデータ数10万件につき5時間弱プラス Lucene インデックス更新で21時間ほど処理時間がかかると見積もられる。

*1 名古屋大学太陽地球環境研究所 (Solar-Terrestrial Environment Laboratory, Nagoya University)

*2 九州大学国際宇宙天気科学・教育センター (International Center for Space Weather Science and Education)

*3 京都大学理学研究科附属地磁気世界資料解析センター

(Data Analysis Center for Geomagnetism and Space Magnetism, Graduate School of Science, Kyoto University)

*4 国立極地研究所 (National Institute of Polar Research)

*5 京大学生存圏研究所 (Research Institute for Sustainable Humanosphere, Kyoto University)

*6 富士通エフ・アイ・ピー株式会社 (FUJITSU FIP CORPORATION)

*7 京都大学理学研究科附属天文台 (Kwasan and Hida Observatories, Graduate School of Science, Kyoto University)

*8 東北大学理学研究科附属惑星プラズマ・大気研究センター

(Planetary Plasma and Atmospheric Research Center, Tohoku University)

1 はじめに

超高層大気長期変動の全球地上ネットワーク観測・研究 (IUGONET: Inter-university Upper atmosphere Global Observation NETwork)¹⁾ は、2009年度より6ヵ年計画で実施されているプロジェクトであり、国立極地研究所宙空圏研究グループ、東北大学大学院理学研究科地球物理専攻太陽惑星空間物理学講座並びに東北大学惑星プラズマ・大気研究センター、名古屋大学太陽地球環境研究所、京大大学生存圏研究所、京都大学理学研究科附属地磁気世界資料解析センター、京都大学理学研究科附属天文台、および九州大学国際宇宙天気科学・教育センターの5機関7組織が参加している。各研究機関が得意とする観測手段、データを持ち寄ることで、幾つかの領域が互いに様々な相互作用をしている超高層大気という領域での、各種物理量の長期変動について研究を行い、地球温暖化など地球規模での大気循環・変動への理解を深めていくことが、このプロジェクトの目的である。

IUGONET 参加研究機関は、これまでレーダー、磁力計、光学観測器、可視光・赤外・紫外線・電波望遠鏡などを用いた超高層大気環境の観測を数十年に渡って行っている。蓄積された大量のデータは各機関が独自に構築したデータベースにアーカイブされているが、データ自体やデータアクセスに関する情報の整備が不足していたり、また内容が非常に専門化されているため、同じ超高層物理分野であっても、別の領域を専門とする研究者にとって、必ずしも利用しやすい状態にはなっていない。その点を改善し、超高層大気の異なる領域に跨るような総合解析に基づく研究を促進するため、IUGONET では各機関が保有するデータに関する様々な情報(メタ情報、またはメタデータ)の抽出・収集を行い、それらをアーカイブして各種検索ができるようなデータベース、すなわちメタデータ・データベースの構築を行っている。またIUGONETでは、このメタデータ・データベースとして、オープンソースの学術リポジトリ用ソフトウェアであるDSpaceⁱをベースにしたシステムの開発を行っている²⁾。

各研究機関から提出されるメタデータ(実際にはSPASEデータモデル³⁾をベースにしたIUGONET共通メタデータフォーマットに沿って作成されたXMLファイル)の登録・管理のために、フリーの版管理ソフトウェアであるGit^jを利用し、Gitによるメタデータリポジトリを核としたメタデータ登録・管理システムを構築して運用している⁴⁾。このシステムによって、新規メタデータの受付や既登録メタデータの改変・修正の履歴管理を行っているほか、IUGONETメタデータ・データベースへの登録もこのメタデータ登録・管理システムを通じて行われる。

このメタデータ登録・管理システムは2010年より本格的に運用を開始し、各部の改良を行いつつ2年余りに渡って運用されている。本論文では、このメタデータ登録・管理システムの運用の現状と、これからも増えていくメタデータ処理に対して運用を最適化するために、その基礎データとなるシステムの性能試験の結果について報告する。またIUGONETでの運用のみならず、フリーソフトであるGitによるメタデータ管理やDSpaceベースのメタデータ・データベースシステムは、モジュール単位で他プロジェクトでも利用可能である。IUGONETでは、プロジェクトで開発したこれらのシステムを他プロジェクト・研究機関に積極的に提供することも行っており、実際に2011年度末より開始した情報通信研究機構からのメタデータ受け入れに伴い、同研究所にメタデータ登録・管理システムおよびメタデータ・データベース一式を提供し、実際に運用も行われている。このように外部プロジェクト・研究機関に提供するという観点からも、システム運用に関する基礎データをまとめておくことは非常に重要である。

2 IUGONET メタデータ登録・管理システムの運用

図1に、メタデータ登録・管理システムを通じて最終的にIUGONETメタデータ・データベースへ登録されたメタデータの総数の推移を示す。ここでは、メタデータ登録・管理システムとメタデータ・データベースとが連動した、定常運用開始(2010年1月)以降のメタデータ数をグラフ化しており、2012年6月末の時点で、メタデータ総数は340万件近くに達している。また図からわかるように、メタデータの増加の仕方は一様ではない。これは新規データセットのメタデータが不定期で登録されるため、その際に、データセットの規模にも依るが、メタデータ数が数千~数十万件程度急増することになる。またグラフがほぼ横ばいになっているように見える期間(例えば2011年7月辺り)でも実はメタデータ数は1日数百件程度ずつ増加しており、これはリアルタイムで観測データが取得されるのに伴って、メタデータが自動生成・登録されるためである。

メタデータ登録・管理システムを通じたメタデータの登録は、実際には大きくわけて4つの作業・処理を経て行われる。図2は、メタデータ作成担当者からメタデータ・データベースに登録されるまでのメタデータの流れを概略的に示している。図の右側がメタデータ登録・管理システムであり、左側が各研究機関でメタデータを作成する担当者のメタデータ作成環境を表している。現状

i <http://www.dspace.org/>

j <http://git-scm.com/>

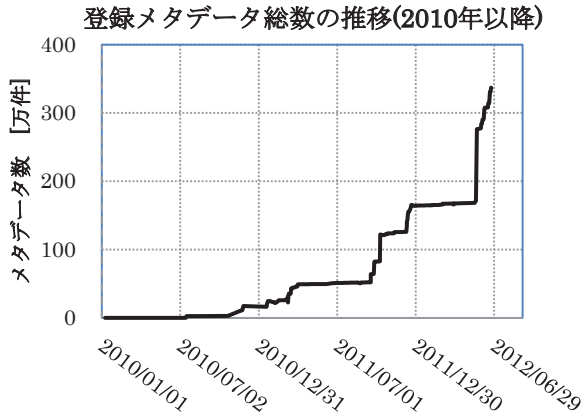


図1 IUGONET メタデータ・データベースに登録された総メタデータ数の推移(2010年1月～2012年6月)

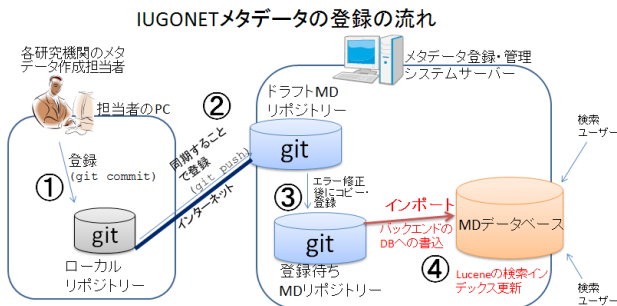


図2 IUGONET メタデータ登録・管理システムを通じたメタデータ提出・登録の流れ

のシステム構成では、メタデータ登録・管理システムとメタデータ・データベース本体は1台のLinuxサーバー上で運用されている。

登録・管理システムのうち、ドラフト/登録待ちメタデータ(MD)リポジトリで構成される部分をメタデータ受付部と呼び、また登録待ちMDリポジトリに入っているメタデータをメタデータ・データベースに登録する機能を担う部分(図中の赤字の部分)をメタデータインポート部と呼ぶ。登録・管理システム側のメタデータ受付部でGitを用いているため、作成担当者側でのメタデータの編集・履歴管理でもGitリポジトリが用いられる。なお、Gitリポジトリ運用の詳細については堀ほか⁴⁾を参照されたい。

メタデータが作成者の手元からメタデータ・データベースに登録されるまでに、主に4つのステップを経ることになる。まず、作成者のローカルな計算機上で作成されたメタデータを、ローカルなGitリポジトリに登録する(図中の①)。具体的には、作成者の計算機上でgit add/commitコマンドが実行されることになる。次にそのローカルリポジトリへの変更分を、メタデータ

登録・管理システムサーバー上のドラフトMDリポジトリとインターネット越しに同期させることで、サーバーに転送する(②)。このローカル-サーバー間の同期はgit pushコマンドによって行われる。このドラフトMDリポジトリ上で簡易的な文法・内容チェックを通過したメタデータは、同サーバー上にある登録待ちMDリポジトリへコピー・登録される(③)。最後に、この登録待ちリポジトリに入ったメタデータは、定期的に自動実行されるDSpaceのimportコマンドでメタデータ・データベースに登録され、また同時に、検索エンジンであるLuceneのインデックス更新が自動的に行われる(④)。このimportコマンドはJavaで実装されており、具体的にはメタデータ・データベースのバックエンドになっているPostgreSQLのDBテーブルへのメタデータの書き込みを行っている。このDBへの書き込みの後、Luceneインデックス更新が完了すると、メタデータ・データベースでの検索でメタデータがヒットようになる。

3 メタデータ登録・管理システムの処理性能評価

現状の登録・管理システムの運用では、図2の①～④が十数時間以内で完了するだろうという見通しのもとで、サーバー上でのメタデータ・データベースへの登録処理を1日1回の頻度で自動起動している。しかしながら、処理プロセスの多重起動を防いだり、作成者がサーバーに転送(図2の②)したメタデータがなるべく早くメタデータ・データベース上で検索可能となるよう登録処理ジョブスケジュールを最適化するためにも、上記の①～④の処理が実際の計算機上でどれくらいの時間を要するかを、あらかじめ見積もっておくことが重要である。また、処理能力の観点で、日々増え続けるメタデータ数に対して、この登録・管理システムがどれくらいのメタデータ総数および1回で登録するメタデータ量まで現実的な運用ができるのかについて、あらかじめ試験しておく、システムの限界などを把握しておくことは不可欠である。

この点については、既に河野ほか⁵⁾でメタデータ・データベースへの登録の部分についてのみベンチマークの結果が報告されている。しかしながら、河野ほか⁵⁾ではメタデータを1000件ずつ計100万件まで追加していった際の登録速度について調査を行ったのみであった。この調査では、Gitリポジトリへの登録、サーバー上のGitリポジトリとの同期に関する性能評価は実施されていない。

また現在では既にメタデータ総数が300万件を超えて

おり、また通常1回に登録されるメタデータ数は数万～数十万件におよぶこともあり、この点では前回の調査で前提としたメタデータ登録数と、現状が既にかげ離れつつあると言える。さらに、前回の調査時から、登録・管理システムサーバーのOSが新しいものに置き換わっており、かつメタデータ・データベース自体にも幾つかの機能追加・強化が施されており、その意味で現状の環境で再度ベンチマーク試験を行う必要があると考えられる。

また河野ほか⁵⁾では行われなかった、今回の新しい試みとして、従来の物理マシンでのシステムだけでなく、仮想マシン上にも同様のシステムを実装して試験を行った。これは、ここ数年でサーバー管理作業を容易にするため仮想化サーバーで各種サーバープログラムを運用する事例が増えてきており、IUGONETでも将来のサーバー運用形態の有力候補として、仮想サーバー検討しているからである。このサーバーの仮想化により、バックアップ作業や、異なる計算機環境への移行が容易になり可搬性が向上するというメリットが得られる。また、期限付きプロジェクトであるIUGONETが終了した後は、限られた人的リソースで、メタデータ登録・管理システムやメタデータ・データベースの運用・メンテナンスを行っていく必要があり、その意味で、サーバー管理の作業コスト削減を模索していくことは、IUGONETプロジェクト後期での重要な課題である。

表1 性能評価を行った計算機環境

CPU	Intel Xeon 2.93GHz(物理 4core) × 2CPU
メモリー	24GB
OS	Scientific Linux 6.2 64bit (Redhat Linux6 互換)
ベースとなる DSpace	ver. 1.7.0
PostgreSQL	ver. 8.4.11
Apache Lucene	ver. 2.9.3
HDD	Hardware RAID 1 by Intel Workstation board S5520SC
仮想化アプリケーション	Oracle Virtualbox ver. 4.1.14
仮想マシンのスペック	Science Linux 6.2 64bit, 2コア, メモリー 12GB, HDD200GB

表1は、今回のベンチマーク試験を行ったLinux環境についてまとめたものである。メタデータ・デー

タベースのベースとなる DSpace や、バックエンドの PostgreSQL 及び Apache Lucene のバージョンは河野ほか⁵⁾での試験環境と同じであるが、河野ら⁵⁾が使用した Redhat Enterprise Linux 5 互換である CentOS 5.3 64bit 版から、Redhat Enterprise Linux 6 互換である Scientific Linux 6.2 64bit 版に置き換わっている。また、仮想マシンと物理マシンとの性能差を評価するために、今回の試験では Git 及びメタデータ・データベースシステムを物理マシン上と仮想マシン上に構築し、それぞれの環境で同じベンチマーク試験を行った。仮想マシンの構築には、Oracle 社の Virtualbox⁴⁾を用いて、その上にホスト OS と同じ Scientific Linux 6.2 およびアプリケーションをインストールすることで、ソフトウェア的に同一のシステムを準備した。幾つか存在する仮想化ソフトの中で Virtualbox を採用している理由は、フリーであり、かつ他の仮想化ソフトではサポートされないことが多い Mac や Solaris をホスト OS として使うことができ、仮想マシンの可搬性というメリットを最大限享受できるからである。また現時点では Virtualbox を採用して試験を行ったが、他の仮想化ソフトも開発が進められており、将来より良いソフトが現れれば、代わってそちらを使用して運用することも視野に入れるべきである。

3.1 Git リポジトリへのメタデータ登録に関する性能評価

まず、Git リポジトリへのメタデータ登録処理にかかる時間を計測することで、Git リポジトリの性能評価を行った。具体的な試験の方法は、表1のスペックの物理マシン上に空の Git リポジトリを作成してローカルリポジトリとして、さらに上記の物理マシンと同一の物理サブネットワーク内にある別の Linux 機上の空ディレクトリを Git リポジトリとして初期化することでリモートリポジトリとした。次にローカルリポジトリに 10000 件の XML 形式のメタデータをコピーして、git add/commit コマンドで登録、さらに git push コマンドを ssh 経由で使うことでリモートリポジトリと同期、という工程を繰り返し実行し、この3種の処理にかかった時間を計測した。また同様の計測を、物理マシンと仮想マシン (Virtualbox) 上の両方でを行い、実行時間の比較を行った。

物理マシンで行った試験の結果を図3に示す。計測は、空のローカルリポジトリに 10000 件ずつメタデータを追加登録していき、総登録数が 500 万件に達するまで行った。この結果から明らかなのは、メタデータファイルのコピー及びリモートリポジトリとの同期は、その

⁴⁾ <https://www.virtualbox.org/>

時の登録メタデータ総数に依らず、常に数秒以内で完了していることである。一方、ローカルリポジトリへの登録は、その時のメタデータ総数に比例して実行時間が長くなっていることがわかる。これは、Gitの動作として、ローカルリポジトリへの登録時にリポジトリ内の全ファイルをスキャンして更新の有無を確認するために、リポジトリ内のメタデータが増えてスキャンするファイル数が増えれば増えるほど実行に時間がかかる、と考えると理解できる。

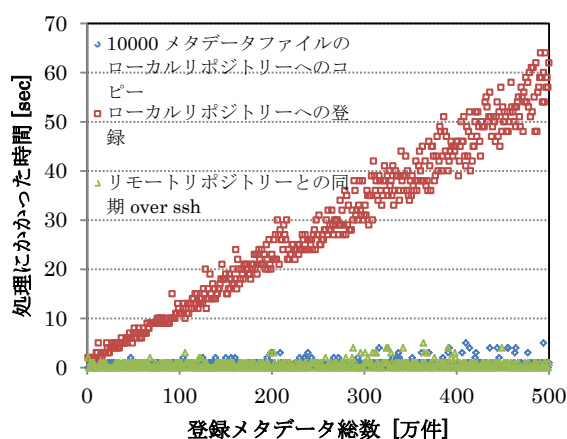


図3 物理マシン上において、10000個のメタデータファイルのコピー、ローカルGitリポジトリへの登録、及びリモートリポジトリとの同期を繰り返した際の、各処理の実行時間。

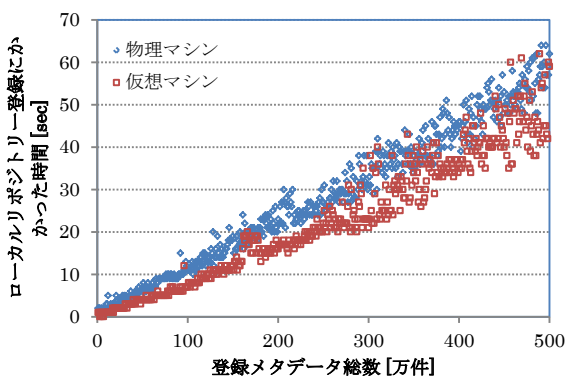


図4 物理マシンおよび仮想マシン (Virtualbox) 上のLinux環境での、ローカルリポジトリへの登録にかかる時間の比較

次にローカルリポジトリへの登録にかかる時間について、物理マシンと仮想マシン (Virtualbox) とで比較した結果を図4に示す。仮想マシンについても物理マシンで行ったのと全く同じ処理を行った。実際にはファイルコピーとリモートリポジトリとの同期についても計測を行ったが、これらは物理マシンと同様に数秒以内であったので、違いが顕著であったローカルリポジトリ

登録の結果のみ図示している。

得られた結果は予想に反し、物理マシンよりも仮想マシンでの処理の方が20-30%ほど高速であった。これには2つ理由が考えられる。1つは、仮想マシン (Virtualbox 上のLinux) では必要最低限のプロセスしか動いておらず、またネットワークを通じた外部からの通信もないため、余計な負荷が全く無い状態で試験を行ったため、今回の試験単体は速く実行できた、という可能性が考えられる。またもう1つ原因として考えられるのは、仮想マシンでのファイルアクセスは、ホストOSのメモリーなどを使った読込/書込処理の高速化が図られているかもしれないということである。今回の試験で用いた仮想マシンには12ギガバイトという比較的大きなメモリーが割り当てられているのに対して、10000件のメタデータXMLファイルの総量はわずか20メガバイト程度であるので、新規に書き込まれたメタデータファイルがキャッシュ機能の範囲内で扱われ、それゆえに全体としてGitの処理が高速化しているというのは十分考えられることである。これはVirtualboxという仮想化アプリケーションの内部仕様に関わることなので、残念ながらこのことを裏付ける資料をOracle社サイト等で見つけることはできなかった。しかし、もしメモリー等を利用したキャッシュ機能でファイルアクセスを見かけ上高速化しているとすれば、そのキャッシュ量の限界以上のアクセスをしようとする、恐らく本来のnative環境より遅い速度になると考えられる。

実際の運用という観点では、図2でのメタデータの登録処理の中で、Gitに関する処理が伴うのは①～③である。今回の性能試験より、メタデータ総数が500万件でもGitリポジトリは正常に機能し、また処理時間は①～③合計で、せいぜい数分程度と見積もることができる。ただし、②の処理については、ローカルリポジトリとリモートリポジトリとの間のネットワーク速度にも依存すると考えられる。しかしながら、後述のように、④の処理時間の方がずっと長いので、メタデータ処理にかかる全体の時間の中で①～③が占める割合は非常に小さいと言える。

3.2 メタデータ・データベースへの登録に関する性能評価

次に、図2の④の部分に当たる、メタデータ・データベースへの登録に関する性能評価の結果について述べる。この処理は2つの要素とそのための準備作業にわけることができる。それぞれ、メタデータ・データベースのバックエンドであるPostgreSQLのDBテーブルへのメタデータの書込、検索エンジンであるLucene用の内部インデックスの更新、及びこの2つの処理を行うための

Java 実装のコマンドの実行スクリプト作成 (Ruby¹ スクリプト及び Unix shell スクリプトで実装) などの前後処理がある。ただし、メタデータ・データベースへの登録は非常に時間がかかるため、前述の Git リポジトリの性能試験とは異なり、10 万件ずつメタデータを登録していった時の、各処理の実行時間を計測している。それでも、現在のところ物理マシンについては 130 万件、仮想マシンについては 150 万件までしかデータが得られていない。この試験は現在も継続中であり、少なくとも Git リポジトリと同じ 500 万件までは試験を続ける予定である。以下では、それぞれ 130 万件および 150 万件までのデータに基づいて議論を行う。

図 5 は、物理マシンで行ったメタデータ・データベースへの登録に関する処理の実行時間の内訳と、登録メタデータ総数への依存性を示したものである。青丸が PostgreSQL の DB テーブルへの書き込みにかかった時間、赤の四角がその後に実行される Lucene インデックス更新にかかった時間、緑の三角が登録の全処理にかかった時間を表す。それぞれのメタデータ総数において、青丸と赤い四角の和は緑の三角が表す時間の約 70-80% 程度となっており、このことから、Ruby+shell スクリプトによる前後処理にかかる時間は、登録にかかる全時間のうちの 20-30% であることがわかる。

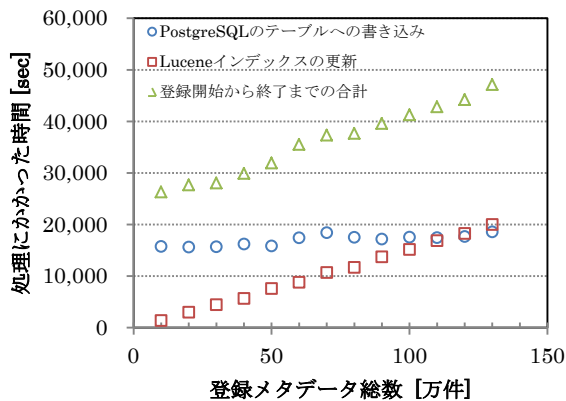


図 5 物理マシンでのメタデータ・データベースへの登録に関する処理の実行時間の内訳と、メタデータ総数への依存性

また図 5 より、PostgreSQL テーブルへの書き込みはメタデータ総数に依らずほぼ一定なのに対して、Lucene インデックス更新にかかる時間はメタデータ総数にほぼ比例して長くなっていることがわかる。この違いは、定性的には両者の処理の性質の違いとして理解できる。つまり、PostgreSQL テーブルへの書き込みはまさにその時に登録するメタデータの量で決まるので、毎回 10 万件とい

う同じ件数を処理しているがゆえに実行時間は毎回同じとなる。逆に Lucene インデックス更新は新しいメタデータの追加によりインデックス構成自体が変わることがあり、その場合 DB テーブル中の全メタデータを読みなおしてインデックスを再構築する必要があるため、実行時間が登録済みのメタデータ総数に比例することになる、と考えられる。また重要なのは、総数 110 万件的の辺りで、PostgreSQL テーブル書き込みにかかる時間と、Lucene インデックス更新にかかる時間の大小が逆転している。つまり、システムとしての運用を考えた場合、メタデータ数が多くなってくると、処理時間としてネックになるのは Lucene インデックス更新になるということである。

次に同じ性能試験を Virtualbox 上の仮想マシンで行った場合の結果を図 6 に示す。結果として、定性的には同じようなメタデータ総数への依存性を示しているが、Lucene インデックス更新にかかる時間が、物理マシンのそれと比べて約 4 倍に伸びている。この結果は、Virtualbox 上でのファイルアクセス、特に仮想ディスクへのファイル書き込みに何らかのキャッシュ機能が介在しているという前述の推測と整合する。もちろんこれは Virtualbox の内部仕様が不明であるがゆえに想像の域を出ないのだが、PostgreSQL テーブルへの書き込み量はメタデータ 10 万件ということから約 200 メガバイト程度とそれほど大きな量ではないが、登録メタデータ総数が 100 万件を超えると Lucene インデックスは数十ギガバイトのオーダーとなり、またインデックス作成に伴い大量の新規ファイルを作成し実際に仮想ディスクに書き込むことになるため、キャッシュ機能で吸収できなくなるのではないかと考えることができる。その結果、仮想マシンではメタデータ総数が 30 万件を超えると Lucene インデックス更新の方が処理時間が長くなり、150 万件的段階ではメタデータ登録の全時間のうちの 8 割強がこのインデックス更新に費やされることになる。

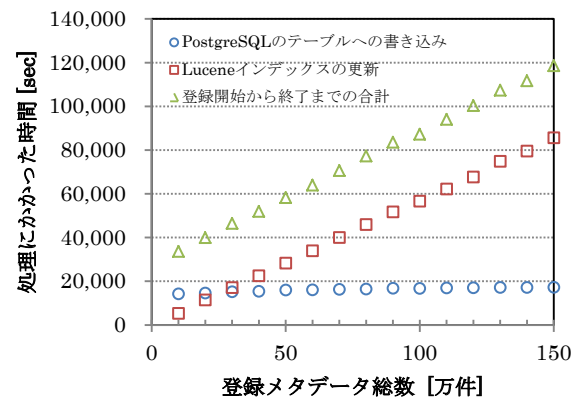


図 6 仮想マシン (Virtualbox) でのメタデータ・データベース登録処理にかかった時間の内訳

¹ <http://www.ruby-lang.org>

4 考察

以上の性能試験から、現状のメタデータ登録・管理システムの運用方針について、幾つかの見通しを得ることができる。

まず、図2に示す①～④の処理のうち、処理時間の観点で一番のネックになるのは④のメタデータ・データベースへの登録の部分であり、かつその中でも Lucene インデックス更新の処理が支配的である。また通常はメタデータの新規追加量はせいぜい数十～数百件/1日であることを考えると、このインデックス更新の処理時間が、登録・管理システムの実行可能な最短サイクル、つまりメタデータ・データベースの更新可能頻度の上限を決めることになる。

例えば今回の物理マシンを用いて、かつメタデータ総数が130万件の場合、書込メタデータ数に比例すると予想される PostgreSQL テーブル書込速度は約17000秒(5時間弱)/10万件、Lucene インデックス更新は総数130万件に対して約20000秒(6時間弱)かかり、前後処理を含めると計13時間かかることになる。つまり、この場合に Git リポジトリ経由で新規メタデータを10万件登録しても数分程度しかかからないが、それが実際にメタデータ・データベースで検索可能となるまでに、理論的には最短で13時間かかるという計算である。

また今回の性能試験は総数150万件までしかデータが取得できていないが、2012年5月下旬に実際に IUGONET で運用中のメタデータ・データベース(現状は物理マシンで運用)に一度に約105万件の新規メタデータが登録され、結果として登録総数は約270万件となった。この時の記録によれば、PostgreSQL テーブルの書込と Lucene インデックス更新に、それぞれ約144,000秒(約40時間)、47000秒(約13時間)かかっており、今回の試験の結果からそれぞれ線形外挿して得られる値におおよそ近いものになっている。この線形外挿からメタデータ総数が500万件に達した場合を予想すると、Lucene インデックス更新にかかる時間は約21時間となる。1つ注意すべきことは、追加メタデータ数が少なくともインデックス更新の時間は変わらない(登録総数に比例)ので、例えば追加メタデータが1件であろうが1万件であろうが、結果として総数500万件であればインデックス更新にはやはり約21時間かかってしまうということである。

この21時間という値は、システムの性能的にメタデータ総数500万件あたりからデータベースの毎日更新が間に合わなくなるということを暗示している。ここで、実際にメタデータ・データベースの機能として要求される更新頻度との兼ね合いで評価する必要があるが、

IUGONET で取り扱う観測データは毎日生産されるものが多く、可能であればメタデータの方も毎日更新され新規観測データのメタデータが即座に検索可能となるのが望ましい。もちろんデータベース更新の頻度は高ければ高いほどよいのであるが、現実的には、例えば毎日1回の観測データのデータベース更新にメタデータ・データベースの方が遅延なく同期できるかどうか、1つの目安と言えよう。この観点で今回の結果を評価すると、現状のシステム性能では、仮想マシンに比べて高速な物理マシン上のシステムを用いたとしても、メタデータ総数が500万件というレベルになると、更新ジョブを走らせる頻度を1日1回以下の頻度に下げざるをえなくなり、つまり毎日更新には追従できないことを意味している。

このように、メタデータ総数が増えてきた時に Lucene インデックス更新にかかる時間が益々ネックになるという予想を踏まえると、この部分の処理を速く行うためには、メタデータインポート部を含むメタデータ・データベース本体については、仮想マシンではなく、物理マシンで運用する方がよいと思われる。ただし仮想化によって得られるシステム保守等の利便性も大きく、また将来的にハードウェア性能が向上すれば実際の処理時間は現実の運用では問題にならない程度に短縮されるかもしれない。このように、実際のデータベース運用でどちらを選択するかは、将来のハードウェア性能向上に合わせて継続的な評価を行いつつ、要求されるデータベースの更新性能と、仮想化による利便性向上との兼ね合いで判断されるべきである。

またメタデータ・データベース更新処理の高速化について、IUGONET 開発チームでも幾つかの施策を検討している。各部の処理を少しずつ改善していくなどの地道な手段の他に、メタデータ・データベースを新バージョンの DSpace ベースに移行することも検討しており、それによって PostgreSQL などのバックエンドが新バージョンとなり、より効率的なデータベース更新処理ができるようになると期待される。

5 結論

IUGONET プロジェクトで作成したメタデータを管理しメタデータ・データベースへの登録を行うメタデータ登録・管理システムに関して、その中枢機能を構成している Git リポジトリの処理とメタデータ・データベースへの登録処理に関する性能評価を行った。Git リポジトリでの処理はメタデータ総数が500万件レベルになってもほぼ数分以内で完了し、処理速度は物理マシンでも仮想マシンでもおおよそ同じくらいであると推測さ

れる。一方メタデータ・データベース登録部分の処理は千倍以上の時間がかかり、特にメタデータ総数が数百万件レベルになってきた場合、処理時間上ネックになるのは Lucene インデックス更新処理である。この Lucene インデックス更新については、物理マシンの方が仮想マシンより約4倍高速に処理できることがわかった。

謝辞

大学間連携プロジェクト「超高層大気長期変動の全球地上ネットワーク観測・研究」は、文部科学省特別教育研究経費(研究推進)[平成21年度]および特別経費(プロジェクト分)[平成22年度~]の交付を受けて、平成21年度より6ヵ年計画で実施している事業である。IUGONET プロジェクトで使用される観測データのデータベース構築に携わった全ての方々に感謝する。特に、国立極地研究所宙空圏研究グループのデータ整備に尽力下さった余川 真純氏、京大大学生存圏研究所のデータ整備に尽力下さった橋口 典子氏、京都大学理学研究科附属天文台のデータ整備に尽力下さった金田 直樹氏に感謝の意を表す。

参考文献

- 1) 林 寛生, 小山幸伸, 堀 智昭, 田中良昌, 新堀淳樹, 鍵谷将人, 阿部修司, 河野貴久, 吉田大紀, 上野 悟, 金田直樹, 米田瑞生, 田所裕康, 元場哲郎, 大学間連携プロジェクト「超高層大気長期変動の全球地上ネットワーク観測・研究」, 宇宙科学情報解析論文誌, 第1号, 2012年3月, 113-120.
- 2) 小山幸伸, 河野貴久, 堀 智昭, 阿部修司, 吉田大紀, 林 寛生, 田中良昌, 新堀淳樹, 上野悟, 金田直樹, 米田瑞生, 元場哲郎, 鍵谷将人, 田所裕康, 超高層物理学分野の為のメタデータ・データベースの開発, 宇宙科学情報解析論文誌, 第1号, 2012年3月, 99-104.
- 3) King, T., J. Thieman and D. A. Roberts, SPASE 2.0: a standard data model for space physics, *Earth Sci. Inform.*, vol. 3, 2010, 67-73.
- 4) 堀 智昭, 鍵谷将人, 田中良昌, 林 寛生, 上野 悟, 吉田大紀, 阿部修司, 小山幸伸, 河野貴久, 金田直樹, 新堀淳樹, 田所裕康, 米田瑞生, IUGONET 共通メタデータフォーマットの策定とメタデータ登録管理システムの開発, 宇宙科学情報解析論文誌, 第1号, 2012年3月, 105-111.
- 5) 河野貴久, 小山幸伸, 堀 智昭, 阿部修司, 吉田大紀, 林

寛生, 新堀淳樹, 田中良昌, 鍵谷将人, 上野悟, 金田直樹, 田所裕康, DSpace を用いた超高層物理学のためのメタデータ・データベースの構築, *Proc. of the 3rd Forum on Data Engineering and Information Management*, 2011, C8-5, <http://db-event.jpn.org/deim2011/proceedings/pdf/c8-5.pdf>.