公共データベースの自動更新システムの開発

１．2002 年度の具体的な研究計画

バイオ情報データの多くが、DDBJ や PDB などの公共データベースに格納されてインターネット上で公開されている。しかし、これら公共データベースの多くはデータベース内部に直接アクセスすることは困難であるため、フラットファイルなどのテキストファイルの形式で利用されることが多く、本プロジェクトでもこの形式のデータを利用することになる。元の公共データベースに対しては更新が毎日のように頻繁に発生するので、データベースの更新を自動的にフラットファイルに反映することが必要となる。

そこで、本研究開発では、元のデータベースの更新がフラットファイルの更新に自動的に反映可能なシステムの開発を目指した。

遺伝子関連データベースの容量は年あたり 1.5 倍から 2 倍という文字通り指数関数的に増大を続けている。データベースを構成するファイルは通常 gzip などの圧縮プログラムで圧縮した形式で提供されているが、それでもファイルサイズが 2GB 以上のファイルが存在する。ファイルサイズが巨大であることは、一つのファイルを、インターネットを介して入手するのに時間がかかることを意味し、その間にネットワーク等に発生する障害への対処が自動更新システムの開発にあたっては重要となる。このため、まずプロトタイプを早期に開発し、巨大なデータベースの入手過程で発生するエラーの洗い出しを行った。ここで得られたエラー情報を基に自動更新システムの仕様を策定した。また仕様策定に当たっては、バイオ関連データベースは年々新たなデータベースが開発されているので容易に新規のデータベースを処理対象に追加できること、データベースのファイル構成はしばしば変更されるのでその変更に柔軟に対応できること、実際にデータベースを利用するにはファイル形式の変更や検索のためのインデックス作成などが必要となるためデータベース入手後に任意の後処理プログラムを実行できることとする。策定した仕様をもとに開発を行い稼動させることになるが、データベース更新時にあらかじめ予期していなかったエラーが発生する可能性があるので、新規のエラーに容易に対処できるものとした。

２．2002 年度の進捗状況と研究成果

１）データベース自動更新時に発生するエラーの洗い出し

通常アノニマス FTP のミラーサイトの維持管理に用いられる mirror プログラムを用いて自動更新システムのプロトタイプをデータグリッド用コンピュータシステム上に構築した。米国 NCBI が提供する BLAST 用データベースに適用することにより、データベース入手時に発生するエラーの洗い出しを行った。その結果、以下のような様々なエラーが発生することが明らかとなった。また各エラー発生の考えられる原因と自動更新システムで行うべき対処も示した。

a) データベースセンターに接続できない例

Cannot connect, skipping package

原因：ネットワークかデータベースセンターの計算機がダウンしている。

対処：再実行

b) アクセスが許されないファイルが存在する例

Failed to get file 550 misc/config/vms/ncbishmh.cfg: Permission denied

原因：何らかの理由で一般のアクセスを許さないファイルが置かれている。

対処：データベースセンターにおいて更新作業中に作られた一時的なファイルの場合がほとんどなので再実行せずに終了する。ただし、このエラーが毎回発生し不必要なファイルと判断される場合はエラーとなるファイルを更新作業からはずす設定を行う。

c) 処理の途中で接続が切れた例

Failed to get file remote server gone away

原因：処理の途中でネットワークかデータベースセンターの計算機がダウンした。

対処：再実行

d) ファイルサイズが、最初に取得したサイズより小さくなった例

Got nr.Z 257321647 (file shrunk from 258218943!) 6918

原因：ファイル転送中にデータベースセンターにおいて更新があった場合に発生する。

対処：再実行（再実行前には後処理を行わない）

e) ファイルサイズが、最初に取得したサイズより大きくなっている例

Got nr.Z 259582111 (file grew from 259402015!) 3828

原因：ファイル転送中にデータベースセンターにおいて更新があった場合に発生する。プロトタイプではファイルサイズの取得を処理の最初に行っているため、複数ファイルからデータベースが構成されている場合、データベースセンターでの更新終了後にファイル転送が行われた時に発生する。

対処：このファイル自体は正常であるが、これ以前に取得したファイルの更新が行われている可能性が高いので、後処理を行わずに、データベース全体に対して再実行を行う。

f) ファイルサイズの減少と増加が同時に発生した例

Got nt.Z 476061696 (file shrunk from 1676570517!) 4621

Got nr.Z 293744383 (file grew from 293263903!) 2850

原因：nt.Z を取得中にデータベースセンターにおいて更新が行われ、それ以降に取得するファイル（nr.Z）も更新されていた場合に発生する。

対処：後処理を行わずに、データベース全体に対して再実行を行う。

g) データベースが置かれているディレクトリが変更になった例

Cannot get remote directory details (/current)

原因：データベースセンターにおいてデータベースを保管するディレクト

リを変更した場合に発生する。
対処：エラーを管理者に報告し、管理者がシステムの設定を変更する

2）仕様策定

プロトタイプの運用により得られたデータベース自動更新時に発生するエラーに対処するとともに、

（1）想定外のエラーが発生した場合にも容易に対処できること
（2）ファイルが正常に取得できているかのチェック機構を備えること
（3）新規データベースの追加が容易にできること
（4）任意のプログラムを後処理として実行できること
（5）管理者の対応が必要と判断されるエラーの発生時には通知すること
（6）データベースの世代管理機能を有すること
（7）Primary サイトに接続できない場合は Mirror サイトに接続し、確実に更新作業を行うための冗長性を有すること
（8）自動更新システムの管理が容易であること
（9）更新状況をデータベースの利用者が容易に確認できる機能を有すること

等の機能を備えた仕様の策定をおこなった。

3）データベース自動更新システムの開発（**付録１参照**）

開発したシステムは大きく「Pre-download check」、「Download」、「Error check」、「Post Processing」の４つのコンポーネントで構成されている。「Pre-download check」では、データベースセンターのデータベースとローカルのデータベースの比較を行い、ファイルが更新されている場合は「Download」を起動しファイルの更新を行う。本来のサイトに接続できない場合は Mirror サイトを利用する機能も実装した。「Download」は、データベースセンターで更新されているファイルのダウンロードを行う。ネットワークが切断された場合は自動的に再接続しダウンロードを継続する機能を有している。「Error check」はダウンロード終了後にダウンロードしたファイルとダウンロード元のファイルのサイズが同じであるかチェックし、サイズが異なる場合はダウンロード中にファイルがデータベースセンターで更新されているので再度自動更新作業をおこなう。「Post Processing」はダウンロードしたファイルに対して、２次データベースなどを作成するプログラムの実行を制御する。

4）その他

データベースの自動更新システムを稼動させる予定の BioGrid コンピュータシステムにプロトタイプをインストールしまず予備実験を行ったが、その段階で一部のファイルが取得できないというトラブルが発生した。調査の結果、公共データベースインストール時に必要となる ftp など多くのコマンドが 2GB 超のファイルに対応していないことが判明した。幸い OS として使用している Linux 自体は 2GB を超える Large File に対応しており、プログラムの再コンパイルにより対処可能であったので必要となるコマンドの再コンパイルとインストールを実施した。

付録１　電子計算機プログラム作成「公共データベースの自動更新システム」

概要

BioDB-Updater は、インターネットに存在する公共データベース（以下、DB）をローカルに回集し、自動的にミラーするツールである。この機能を骨格としてミラー前後に任意のプログラムを実行・監視させることにより、柔軟に独自のローカルデータベース構築管理作業を行うことが可能である。自動化は cron により実現する。BioDB-Updater の処理フローを図 1 に示す。

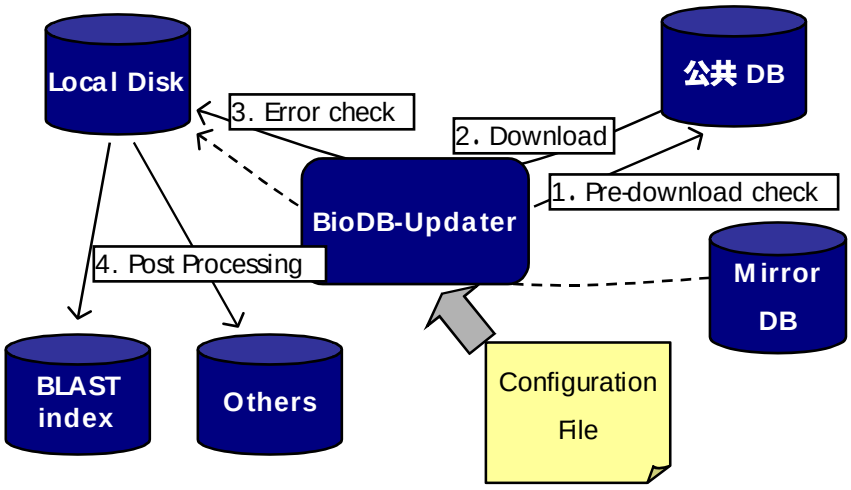

図1　BioDB-Updaterの処理フロー

１． Pre-download check

ダウンロードする前に、Local Disk の現状と比較し、ファイルが更新されているか、ディレクトリ構造に変化がないかなどをチェックする。 Primary サイトに接続できない場合は、Mirror サイトに接続してダウンロードする。

2. Download

公共データベースサイトから、更新されたファイルをダウンロードする。回線が途中で切れても自動的に再接続する。

3．Error check

ダウンロードした後にダウンロード元が更新されていないかなど、ダウンロードしたファイルとダウンロード元が同じかどうかをチェックする。

4. Post Processing

ダウンロードしたファイルに対して、2 次 DB などを作成するための処理を実行する。

構成要素

図 1 にあるような処理フローを実現するために、以下のようなソフトウェア、機能を利用している。

１）再接続、再取得

フリーウェアである lftp を利用した。lftp に実装される再接続、再取得機能を利用して機能を実現している。ファイル取得時には lftp を利用しているためこの機能が利用可能になる。

２）シンボリックリンクの活用

データベースを取得している最中も、ローカルに保存されているデータベースを問題なく参照できる。また、同じデータベースのバージョンを複数保存できるようにするために、UNIX のシンボリックリンクを利用した。Post Processing の部分でシンボリックリンクが多用されており、本システムの利用環境としてもシンボリックリンクが重要な役割を果たしている。

３）FTP サイトの情報取得

通常の ftp クライアントには ftp サイトにあるファイルのタイムスタンプ、ファイルサイズなどの詳細を得ることができないので、lftp を拡張し、必要な情報を得られるようにしている。Pre-download check およびファイル取得後の Error Check において本機能を利用している。

Post Processing について

取得後以下の 3 つの機能を用意し、データベース取得後のデータ取扱を処理している。これらは取得するデータベースの用途によって使い分けられる。

１）データ公開用ポスト処理

公共データベースからデータベースファイルを自動で取得・更新した後、新しいデータセットをユーザが参照できるように準備する処理系。

主にマイナーアップデートデータベース（リリースからの差分・増分データベース）の取得後に利用される。

２）世代管理用ポスト処理

公共データベースからデータベースファイルを自動で取得・更新した後、新しいデータットをユーザが参照できるように準備し、古い世代を規定世代数保存する処理系。

主にメジャーリリースデータベース（GenBank のリリースなど）の取得後に利用される。

３）Blast データベース用ポスト処理

公共データベースからデータベースファイルを自動で取得・更新した後、取得したデータベースを基に Blast インデクスを作成し、取得したフラットファイル、作成した Blast インデクスの新しいデータセットをユーザが参照できるように準備し、古い世代を規定世代数保存する処理系。

主に NCBI の Blasted（FASTA ファイル）を取得した後に利用される。

BioDB-Updater 設定ファイル

BioDB-Updater は、BioDB-Updater 設定ファイル（以下、設定ファイル）を読み、その

内容にしたがって一連の処理を行う。BioDB-Updater の動作は全て設定ファイルにより規定される。管理を容易にするため、設定ファイルはデータベースごとに一つの設定ファイルを用意して、どこのデータベースを取得するのかなど必要項目を設定ファイルに記述しておく。以下に設定ファイル例を示す。この例においては、url=接続先 URL、odir1=取得データベース保存ディレクトリ、odir2=代替ディレクトリ、pre_sh=取得前実行スクリプト、post_sh=取得後実行スクリプト、recursive=サブディレクトリ取得のためのスイッチ、となっている。

```
url =ftp://hoge.hoge.net/pub/genbank_mini
odir1 = /bio/test/genbank1
odir2 = /bio/test/genbank2
pre_sh=;
post_sh=mkidx_gb.pl;link.pl;remove.pl;copy.pl
recursive=n
```

BioDB-Updater ログ管理

BioDB-Updater は、本体と前後に実行したプログラムの全ログを作成する。ログの種類としては、(1)BioDB-Updater メインログ、(2) データベースサイトへの ftp 通信ログ、(3)プレ処理実行ログ、(4)ポスト処理実行ログ、の４種類が出力される。

自動取得方法

BioDB-Updater は cron 設定を行うことで全体を自動化する。通常は標準出力、標準エラー出力に表示されるメッセージを、すべてログに書き出すことで動作結果を後から追うことができる。ログとは前項の BioDB-Updater ログ管理に挙げられているログファイルのことである。

現状の問題点と今後の機能拡張について

BioDB-Updater では、データベースの自動更新に必要な機能については網羅している。しかしながら、現在の BioDB-Updater のインターフェースはすべて CUI であるため、データベース更新管理に多少の習熟が必要になる。そこで本システムでは管理用 GUI を用意することにし、画面設計までを完了している。今後この機能が利用できるよう、内部処理部分を実装していく予定である。また、後処理プログラムの起動条件を柔軟に設定できる仕組みなど、データグリッド技術グループの目標達成のために有用な、より柔軟なデータベース更新が行える機能拡張をはかっていく。