3J-7

Hadoopによるバックアップシステムの提案と実装

武井優樹† 中山泰一†
†電気通信大学情報工学科

1 はじめに

近年，データのバックアップが肝要と言われている．しかし，現状ではバックアップに伴う計算コスト，処理にかかる計算資源などを理由にバックアップは行われていないことも多い．本研究ではそのバックアップを行うために，分散支援フレームワークである Hadoop を使用したバックアップシステムを提案し，実装する．

バックアップには幾つか種類がある．代表的には，完全バックアップ，差分バックアップである．それぞれに利点，欠点はあるものの，データを一箇所に保存しておいたのでは冗長性が得られない．一つの災害でたやすく失われてしまう可能性がある．

本研究は，非累積型差分バックアップを行い，その処理，データを分散処理機構に移す．目的として次のものがあげられる．

- 分散保存により冗長性を確保する．
- 分散計算を行うことによって，ローカルでの差分，復元の処理を減らし，処理時間を短縮する．

分散ストレージによるバックアップを提案している研究は幾つかある [1] が，完全バックアップか，もしくは差分の計算をローカルで行っている．本研究では，非累積型差分バックアップによるバックアップを行い，差分の計算を分散クラスタに行わせることを提案している．

2 バックアップ

バックアップには，完全バックアップの他，差分バックアップがある．さらに差分バックアップには，累積型と非累積型 [2] があり，現在では非累積型差分バックアップは増分バックアップと呼ばれている．これらについて以下で説明する．

Proposal and Implementation of Backup System by Hadoop
Yuki TAKEI†, Ysuichi NAKAYAMA†
†Department of Computer Science, The University of Electro-Communications
182-8585, Chohu,Tokyo, Japan
takei-y@igo.cs.uec.ac.jp

2.1 累積型差分バックアップ

累積型差分バックアップは，一度完全バックアップをとった後，今度は完全バックアップからの変更部分のみをコピーするものである．完全バックアップに比べディスク使用量を削減できるが，次第にデータ量が増加していく．それにつれて処理時間が増大していくが，バックアップ媒体の管理，復元は容易にできることが特徴である．

2.2 非累積型差分バックアップ

非累積型差分バックアップは本研究で採用したバックアップ手法である．非累積型差分バックアップは累積型同様一度完全バックアップをとった後，次回からはその都度の変更部分のみをコピーする手法である．非累積型差分バックアップのメリットは，差分の計算処理が累積型に比べ増えないこと，ディスク使用量を削減できることである．デメリットはバックアップを行うに連れて媒体管理や復元処理が複雑化することである．本研究ではバックアップ時の処理時間が最短であることから，非累積型差分バックアップ方式を採用した．

3 Hadoop

本研究では，分散処理システムとして Hadoop 0.20.203.0 を採用した．Hadoop は大容量のテキストデータの処理に向いており，大規模なデータマイニングに使用されることが多く，Yahoo や Livedoor，Facebook などが採用している．

今回提案するバックアップシステムも大容量のテキストを処理すること，また分散処理モデルである Map/Reduce 処理を比較的簡単に記述できることが Hadoop を選択した理由である．ここでの大容量のテキストデータとは，数百 MB から数 PB を指す．また，これらは既存のネットワーク上，コモディティサーバ上で動き，要求されるハードウェア性能が低いのも特徴である．Hadoop は，現在 apache のオープンソースプロジェクトとして開発が進められている．

Hadoop は HDFS という通常のファイルシステムの上

に成り立つ分散ファイルシステムを持っており，Hadoopで処理したいデータはこのHDFS上に置かれる[3][4]．分散処理に関しては，Hadoop Map/Reduceを自作することによって行うことができる．

Map/Reduceは基本的にはJavaによって組まれ，Hadoopの提供するAPIと併用しつつ様々な処理が行える．Hadoop上で動作するでデータベースと言える，HiveやHBaseなどもある．

3.1 Map/Reduce

本節では，前述したMap/Reduceについて簡単に説明する．Map/Reduceは大規模データの集合を処理するためのプログラミングモデルと言える．並列に実行するため，ひとつの処理を独立したタスクの集合に分けて実行させる．Map/Reduceは大きくはMap処理とReduce処理に分かれ，Map処理は入力ファイルを切り分けたものに対してフィルタリングを行い，Reduce処理はMap処理から渡されたデータの集約を行う．

Hadoop Map/Reduceでは、テキストは一行ずつに分割され，それぞれにおいて処理が行われる．本研究の場合，ファイルリストが大容量のテキストに当たり，これを効率的に処理するためにHadoopを選択した．

4 システムの設計

本研究では，非累積型差分バックアップによる実装を行った．以下に簡単な処理の流れを示す．

1. バックアップしたい計算機のファイルリストを取得.
2. 全てのファイルについて3から5を行う.
3. ファイルリストからファイル情報を一行とってくる.
4. 新規ファイルならバックアップリストへ追加，データベースに書き込みして5へ．前回バックアップがあるなら4へ.
5. データベースからファイル情報を取り出し比較，更新されているならバックアップリストへ追加しデータベースも更新する.
6. 全てのファイルについて終了した後，リストに従ってバックアップを行う.

以上をC言語，PostgreSQLとHadoop，HBaseでそれぞれ実装する．

5 システムの評価方法

本システムは大まかに三つに分けられる．ファイルリストの取得，差分の計算，そしてファイルコピーである．その内ファイルリストの取得はローカルにおいても分散においても結局バックアップしたい計算機によるため考慮しない．今回の目的の一つは差分の計算時間の短縮なので，特にその部分について計測する．

実験方法は，ファイル数を一定のまま更新ファイル数を変更し測定する．ローカルで実験する場合と分散クラスタで実験する場合ではどうしても計算機自体の性能が変わってきてしまうが，本研究の目的が通常時にバックアップ行った場合よりも処理時間を短縮するというものであるから，ローカルでの実験の場合は普段使用している程度の計算機を使用する．そのため計算機の性能差については考慮しない．

6 まとめと今後の課題

本研究の目的は，差分の計算時間短縮，データの冗長性確保であった．分散クラスタによって差分の計算時間は短縮できると予想され，HDFSによって冗長性も確保されているだろう．

本研究を始めた当時Hadoopは正式版ではなかったが，先日Hadoop 1.0がリリースされた．本研究ではHadoop 1.0への移植が間に合わなかったが，いずれ移植したいと考えている．

参考文献

[1] 宇田隆哉，井上亮分，伊藤雅仁，市村哲, 田胡和哉，星徹：ファイル分散バックアップシステムの開発，東京工科大学研究報告，03，2008.

[2] 藤武光夫，太田貞夫：バックアップの高速化における一手法，情報処理学会第47回全国大会，1993.

[3] Tom White(著)，玉川竜司，兼田聖士 (訳)：Hadoop:The Definitive Guide，第二版, オライリー・ジャパン，2011.

[4] 太田一樹他：Hadoop 徹底入門 オープンソース分散処理環境の構築，初版，翔泳社，2011.