Haodoopのノード脱退時，ノード削除時のレプリカ生成挙動に関する評価

日開 朝美† 竹房 あつ子‡ 中田 秀基‡ 小口 正人†
†お茶の水女子大学 ‡産業技術総合研究所

1. はじめに

近年，通信技術や情報処理技術の発展と普及に伴い，データ量が爆発的に増加している．そこで汎用のハードウェアを用いて高度な集約処理を行う分散ファイルシステムに注目が集まっている．分散ファイルシステムは，高いスケーラビリティを持つため，システムの規模に応じた運用が見込まれる．また従来では想定されていなかったような極めて大規模なシステム構成を取る機会が増えたが，全てのマシンを正常に動作させることは難しく，常に一定の割合で故障や動作不安定なマシンが存在してしまうという問題が新たに生じる．

本研究では，Apache Hadoop(以下 Hadoop)[1] の基盤技術であり，Google の分散ファイルシステム GFS[2] に基づいて設計された Hadoop Distributed File System(以下 HDFS)[1][3] に着目した．Hadoop はスケールアウトするシステムであり，システム規模に応じた運用や故障の発生などにより，ノードを脱退もしくは削除する事が想定される．HDFS は一般に複数のレプリカを保持し，ノード脱退時及び削除時にはレプリカが自動的に生成される．しかしその過程が十分に早くないとレプリカが不足してしまう可能性があるため，その高速化が重要である．そこで本稿では，ノード脱退時及びノード削除時のレプリカ生成の挙動の解析を行った．

2. Apache Hadoop

2.1 Apache Hadoop の概要

Hadoop は Apache Software Foundation で開発されている分散コンピューティング関連のプロジェクトである．分散ファイルシステムの HDFS，分散処理アルゴリズムの Hadoop MapReduce[1], データベースの hBase[1] から構成される．HDFS は，ファイルのメタデータやクラスタ内のノード管理を行う一台の NameNode と，実際にデータを格納している複数台の DataNode からなる．ファイルを最小単位であるブロックに分割して DataNode 間で分散して保存することで，耐故障性と対多クライアントでの高いパフォーマンスを維持している．

3. ノード脱退/削除時のレプリカ生成の流れ

HDFS はクラスタからノードを切り離す際，該当ノードが保持しているデータを残りのノードに生成することで，指定したレプリカ数を維持する．本研究ではクラスタからノードを切り離す方法として，ノード脱退とノード削除の 2 種類を用いる．

ノード脱退とは，事前に脱退ノードが保持しているデータを他のノードに複製してから，そのノードをクラスタから切り離す方法である．クラスタ規模の縮小やエラー率が非常に高いノードが存在する時など，意図的にクラスタからノードを切り離す場合を想定している．

ノード削除とは，クラスタからノードが離脱後，残ったノード間で削除ノードが保持していたデータを補完する方法である．ノードの故障や動作不良あるいはネットワークの切断など，想定外にノードが突然クラスタから離脱してしまう場合を表す．

3.1 レプリカ生成の流れ

ノード脱退および削除の処理が行われると，レプリカ生成処理が行われる．レプリカ生成はブロック単位に処理が進み，まず NameNode が定期的に (default:3 秒間隔) レプリカ生成の指示をそれぞれの DataNode に転送する．DataNode は受けとった指示をもとに，レプリカ生成先の DataNode へデータの転送を始める．生成先ではデータの複製が完了すると，生成元の DataNode に生成完了の ACK を転送する．この時 NameNode が定期的に DataNode に転送するレプリカ生成の指示のブロック数は，クラスタに接続されている *DataNode* 数∗ `REPLICATION_WORK_MULTIPLIER_PER_ITERATION`(default:2) で定義される．DataNode が生成完了の ACK を受け取らない状態のまま，レプリカ生成先の DataNode へ転送できるブロック数は `dfs.max-repl-stream`(default:2) で定義される．`REPLICATION_WORK_MULTIPLIER_PER_ITERATION` は FSNamesystem.java の ReplicationMonitor クラスで定義されている変数であり，`dfs.max-repl-stream` はプロパティで変更可能な変数である．

4. 性能評価

4.1 実験概要

本実験ではレプリカ生成に関する評価を行うために，ノードを切り離す方法の違いや，ブロックサイズと同時に処理するデータ量の変化がレプリカ生成処理のスループットに与える影響を調査するための実験を行った．ブロックのレプリカ数を 3 とし，1 台の DataNode をクラスタから切り離す際のスループットを測定した．ここでスループットは以下のように定義した．

$$\text{スループット}\ (MB/sec) = \frac{\text{脱退及び削除ノードが保持しているデータ量}\ (MB)}{\text{データの複製が開始してから完了するまでの時間}\ (sec)}$$

Evaluation of Replica Generation Behavior at Node Decommission and Deletion for Hadoop

† Asami Higai, Masato Oguchi
‡ Atsuko Takefusa, Hidemoto Nakada
Ochanomizu University (†)
National Institute of Advanced Industrial Science and Technology (AIST)(‡)

4.2 実験環境

クラスタツール Lucie を用いて構築したローカルクラスタ上で Hadoop-1.0.3 をインストールしたマシン 7 台を用いた．そのうち 1 台を NameNode とし，残りの 6 台を DataNode とした．マシンのスペックは全て同一で表 1 に示す通りである．

表 1: マシンスペック

OS	Linux 2.6.32-5-amd64 Debian GNU/Linux 6.0.4
CPU	Quad-Core Intel(R) Xeon(R) CPU @ 2.66GHz
Main Memory	2GB
HDD	73GB SAS × 2(RAID0)
RAID Controller	SAS5/iR

4.3 ノード脱退とノード削除における基本性能

ブロックサイズを 64MB(default) とし，ノード脱退とノード削除の 2 つの方法でクラスタから DataNode 切り離した際のスループットを測定した結果を図 1 に示す．図 1 より，ノード脱退時の方がノード削除時に比べてスループットが高いことが分かる．これはノード削除時には，削除ノードを除く残りのノードでレプリカ生成の処理が行われるのに対して，ノード脱退時には，脱退ノード自身がレプリカ生成元としてレプリカ生成の処理に参加するためである．

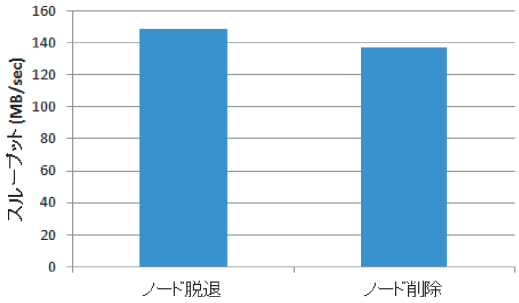

図 1: ノード脱退時とノード削除時のスループット

4.4 同時転送ブロック数を変化させた実験

ブロックサイズ 16，32，64，128，256MB に対して，レプリカ生成時の NameNode と DataNode の同時転送ブロック数に関連する，`REPLICATION_WORK_MULTIPLIER_PER_ITERATION` と `dfs.max-repl-stream` の 2 つの変数を共に同じ値 α とし，α を 1〜8 に変化させてノード削除を行った際のスループットを図 2 に示す．図 2 より，α が小さいとき，すなわち同時転送するブロック数が少ない場合，ブロックサイズが小さい時にはスループットが低くなる．これはブロックサイズが小さいと DataNode はレプリカ生成の処理が完了して，NameNode から定期的に送られてくるレプリカ生成の指示を待っている状態が生じるからと考えられる．またこの状態にある間は，α に比例してほぼ線形にスループットが増加している．本実験のスループットの上限値は約 138MB/sec であり，上限値に到達後は α の増加に伴い若干スループットが低下している．これは輻輳制御やスケジューリングの影響の可能性が考えられる．

また本実験で最もスループットが高かったブロックサイズが 128MB，α が 2 のノード削除時のデータの移動状況を図 3 に示す．横軸は時間 (sec) で縦軸の数値はそれぞれ DataNode を示している．矢印の始点がレプリカ生成元で，終点が生成先であり，その間データ転送が行われていることを表している．図 3 より，最もスループットが高かった時でも，ノード間でデータ移動に偏りが存在し，この偏りをバランスさせることでスループットの向上を見込める．

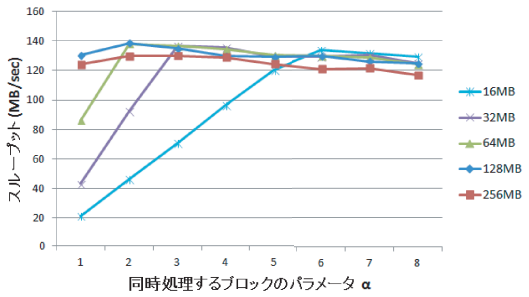

図 2: 各ブロックサイズにおける同時処理するブロック数に応じたスループット

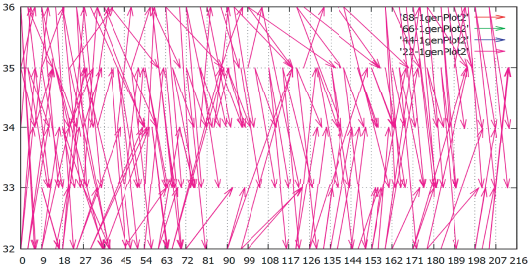

図 3: レプリカ生成時のデータの移動状況

5. まとめと今後の課題

分散ファイルシステムの一つである Hadoop 上で，ノード脱退時とノード削除時のレプリカ生成に関する基本性能の評価を行った．ノード脱退時の方が削除時に比べてスループットが高いことと，レプリカ生成処理の同時転送するブロック数とブロックサイズを変化させると性能が低下することを確認した．またレプリカ生成処理のノード間のデータ移動に偏りが存在することも確認した．

今後の課題としては，レプリカ生成時の詳細なデータ移動を解析するとともに，レプリカ生成処理の高速化に向けて，ノード脱退時及び削除時のノード間のデータ移動の偏りをバランスさせる制御手法の提案と実装を進めていきたい．

参考文献

[1] Tom White(著), 玉川竜司，兼田聖士 (訳):Hadoop, オライリー・ジャパン,2010

[2] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, ”The Google File System”, Proc. 19th ACM Symposium on Operating Systems Principles, pp.29-43, Octorber 2003.

[3] Dhruba Borthakur, ”HDFS Architecture”, 2008 The Apache Software Foundation.