大規模環境における HDFS のレプリカ再配置処理のシミュレーションによる評価

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，センサネットワークやソーシャルメディアなどから大量のデータが生み出されるようになり，商業分野や科学技術計算分野を始め，大規模データを効率良く管理，処理することが求められている．このような大規模データを処理するため，汎用的なハードウェアを用いて高度な集約処理を可能にする分散ファイルシステムが広く使用されている．分散ファイルシステムは，複数のノードでレプリカを分散して保持することで耐故障性を維持している．ノードが故障すると，そのノードが管理していたレプリカの再配置が行われる．しかしながら，一般に広く使われている Hadoop Distributed File System(HDFS) のレプリカ再配置時の処理では，レプリカ生成元と生成先がランダムに選ばれるため，データ移動に偏りが生じて，効率良く処理が行われていない [1]．本稿では，これまで提案してきたリング構造に基づく一方向のデータ転送によって負荷分散を行うレプリカ再配置のスケジューリング制御を 0-1 整数計画問題として定式化した最適化手法を実装し，評価する．評価実験から，提案手法によりデータ移動の偏りが解消され，効率良く処理が行われることを示す．

2. 関連研究

Felix[2] らは大規模クラスタにおいて OS イメージを全てのマシンに高速に分配する方法として，star 型，n-ary spanning tree 型，multi-drop-chain 型の 3 つの論理ネットワークトポロジを取り上げ，調査している．star 型はノード数が増加すると，スイッチ部分で輻輳が発生してしまう．n-ary spanning tree 型はネットワーク帯域の限界により複数のストリームを効率良く処理できない．一方で multi-drop-chain 型は，ノード数やネットワーク帯域に大きく影響されることなく，データの転送が可能であり，データの分配に適した方式であると述べられている．

我々は，レプリカ再配置において multi-drop-chain 型と同様の転送処理が行われるように，一方向のリング構造をベースとしたスケジューリングを行う．

3. レプリカ再配置の提案手法

レプリカ再配置処理を効率良く行うには，適切にレプリカの生成元と生成先を選択して各 DataNode の送受信処理を均衡化することが必要である．そこで我々は，生成元が決定すると生成先が一意に決定する，一方向のリング構造に基づいてデータ転送を行いながら，各 DataNode の転送データ量を均衡化するレプリカ再配置のスケジューリング方針を提案している [1]．この方針をもとに，0-1 整数計画問題として定式化して最適化ソルバで求めた最適解を用いる最適化手法を提案する．

3.1 提案するスケジューリング方針の概要

提案するレプリカ再配置のスケジューリング方針を以下に述べる．全ノードが同一ラック上に存在するものとする．

1) DataNode を論理的にリング状に配置し，そのリング構造に従って一方向にデータを転送する．
2) 各 DataNode が送信するデータ量を等しくするために，生成元に選出される回数を等しくする．

3.2 最適化手法

前述で提案したスケジューリング方針を 0-1 整数計画問題として定式化し，その解をレプリカ再配置のスケジューリングに用いる．定義する 0-1 整数計画問題では，レプリカ再配置に要する時間を短縮するために，各 DataNode の送信ブロック数の差を最小化することを目指す．ソルバには無償で提供されている GLPK を用いる．

まず，記号の定義を与える．DataNode i の集合を D，レプリカ再配置が必要なブロック j の集合を B とする．DataNode の総数を N_{dn}，レプリカ再配置が必要なブロックの総数を N_b，レプリカ数を $N_{replica}(\geq 2)$ とすると，DataNode あたりの平均転送ブロック数 N_{avg} は，$N_{avg} = N_b/N_{dn}$ となる．現在のブロックの配置を行列 $Current_{i,j}(i \in D, j \in B)$ とする．$Current_{i,j}$ の値は，DataNode i にブロック j が存在する場合は 1，存在しない場合は 0 とする．また DataNode の隣接関係を行列 $Adj_{from,to}(from, to \in D)$ とする．$Adj_{from,to}$ の値は DataNode $from$ から DataNode to への転送が可能な場合は 1，そうでない場合は 0 とする．レプリカ再配置のスケジューリング結果を格納する変数を $X_{from,to,j}$ で表す．$X_{from,to,j}$ の値は，DataNode $from$ から DataNode to へブロック j の転送をする場合は 1，ない場合は 0 とする．各 DataNode i の転送ブロック数の差を最小化するために利用する変数を z_i とする．この時，レプリカ再配置のスケジューリングは以下のように定式化される．

$$Minimize \quad \sum_{i \in D} z_i \tag{1}$$

$$Subject\ to \quad All_{i,j} = Current_{i,j} + \sum_{from \in D} X_{from,i,j} \quad \forall i \in D, \forall j \in B \tag{2}$$

$$All_{i,j} \leq 1, \ \forall i \in D, \ \forall j \in B \tag{3}$$

$$\sum_{i \in D} All_{i,j} = N_{replica}, \ \forall j \in B \tag{4}$$

$$X_{from,to,j} \in \{0,1\}, \ \forall from, \forall to \in D, \forall j \in B \tag{5}$$

$$Current_{i,j} - \sum_{to \in D} X_{i,to,j} >= 0, \quad \forall i \in D, \forall j \in B \tag{6}$$

A Simulation Evaluation of Effective Replica Reconstruction for a Large-scale HDFS Environment
† Asami Higai, Masato Oguchi
‡ Atsuko Takefusa,Hidemoto Nakada
Ochanomizu University (†)
National Institute of Advanced Industrial Science and Technology (AIST)(‡)

$$\sum_{j \in B} X_{from,to,j} <= M \cdot Adj_{from,to}$$

$$\forall from, \forall to \in D \quad (7)$$

$$\sum_{j \in B} X_{from,to,j} - N_{avg} \geq -z_i, \quad \forall from, \forall to \in D \quad (8)$$

$$\sum_{j \in B} X_{from,to,j} - N_{avg} \leq z_i, \quad \forall from, \forall to \in D \quad (9)$$

$$z_i \geq 0, \ \forall i \in D \quad (10)$$

上記を満たす，$X_{from,to,j}$ を求めレプリカ再配置のスケジューリングに用いる．式 (1) は，各 DataNode の転送ブロック数の差を最小化するための目的関数を表す．式 (2) は，転送後の配置を $All_{i,j}$ で表す．式 (3) は，転送後の配置において同じ DataNode に同じブロックが 2 つ以上配置されないことを表す．式 (4) は，各ブロックのレプリカの総数が $N_{replica}$ になることを表す．式 (5) は，$X_{from,to,j}$ は 0 か 1 の値をとることを表す．式 (6) は，ブロックの生成元となる DataNode がそのブロックを持っていることを表す．式 (7) は，DataNode $from$ と DataNode to が転送リングにおける生成元 DataNode と生成先 DataNode の関係にあることを表す．隣接関係がない DataNode 間の転送ブロック数は 0 で，ある場合は正の値となる．M はある程度大きい値であり，ブロック総数を超えることはないので，ここでは $M = N_b$ とする．式 (8)(9) は，各 DataNode が転送するブロック数とその平均値 N_{avg} の差の下界と上界を表す．式 (10) は，z_i は 0 以上の値をとることを表す．

4. 評価実験

デフォルトの手法と最適化手法，そして [1] で既に提案している制御手法 (以下，ヒューリスティック手法) を用いて，ノード削除時のレプリカ再配置処理の性能を評価する．Hadoop-1.0.3 をインストールしたマシン 7 台からなるクラスタを用いて，その内の 1 台をマスタノードの NameNode とし，残りの 6 台をワーカノードの DataNode とし，DataNode の内の 1 台を削除する．マシンのスペックは全て同一で表 1 に示す．全ノードが Gigabit Ethernet で接続された単一のラックからなる．レプリカ数を 3 とし，HDFS の総データ量は，50GB×3(レプリカ数)=150GB となっている．

各手法のレプリカ再配置のスループットを図 1 に示す．縦軸がスループット [MB/sec]，横軸がブロック (データ) サイズである．図 1 より，ブロックサイズが 64MB 以上の場合は，提案手法によりスループットが向上しており，デフォルト手法と比較するとヒューリスティック手法では最大で 44%，最適化手法では最大で 45%向上している．ブロックサイズが 16MB，32MB と小さい場合は，制御前後でスループットに変化がないが，ディスク帯域を使い切っておらず処理に余力がある為である．またヒューリスティック手法と最適化手法は同等の性能であることが確認できる．

またデフォルト手法とヒューリスティック手法を用いた際の各 DataNode のディスク I/O スループットと受信ブロック数の時系列データをそれぞれ図 2，3 に示す．縦軸がディスク I/O スループット [MB/sec] および受信ブロック数，横軸が時間 [sec] である．デフォルト手法では各 DataNode の受信ブロック数に大きく差があり，ディスク I/O も不安定であったのに対し，ヒューリスティック手法では，各 DataNode の受信ブロック数が安定し，ディスク I/O は比較的安定した高い値が全 DataNode で維持されている．

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@1.60GHz
Main Memory	2GB
HDD	73GB SAS×2(RAID0)
RAID Controller	SAS5/iR
Network	Gigabit Ethernet

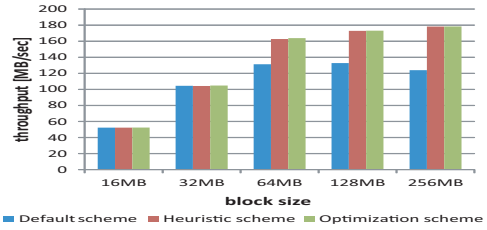

図 1: 各手法におけるレプリカ再配置のスループット

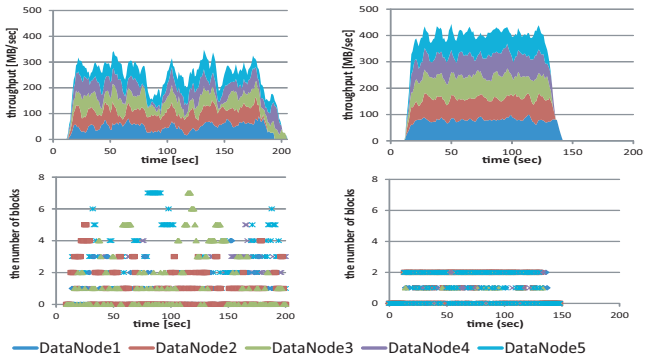

図 2: 上：各 DataNode のディスク I/O スループット (積み上げグラフ)
図 3: 下：各 DataNode の受信ブロック数
(左：デフォルト手法，右：ヒューリスティック手法)

5. まとめ

HDFS 上において効率良くレプリカ再配置を行うために，リング構造に基づく一方向のデータ転送を行い，各ノードの負荷を均衡化するスケジューリング方針をもとにした最適化手法を実装し，評価した．評価実験から，提案手法によりデータ移動の偏りが解消され，その有効性を示した．またヒューリスティック手法が，最適化手法と同程度の性能が得られることを示した．

今後の課題は，ラック構成を導入した大規模環境における提案手法の有効性を検証することである．

参考文献

[1] 日開朝美，竹房あつ子，中田秀基，小口正人，”Hadoop のノード削除時のレプリカ生成の高速化手法の提案” Multimedia, Distributed, Cooperative, and Mobile Symposium(DICOMO2013), 7H-2, July 2013.

[2] Felix Rauch, Christian Kurmann, Tomas M.Stricker, ”Partition Cast — Modelling and Optimizing the Distribution of Large Data Sets in PC Clusters”, Euro-Par 2000, LNCS 1900, pp.1118-1131, 2000.