「マルチメディア，分散，協調とモバイル
(DICOMO2014)シンポジウム」 平成26年7月

# シミュレーションを用いた HDFSのレプリカ再配置手法の性能評価

日開 朝美[1] 竹房 あつ子[2] 中田 秀基[2] 小口 正人[1]

**概要**：大規模データに対応した処理システムとして，汎用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．オープンソースの Hadoop Distributed File System(HDFS)は広く使用されている分散ファイルシステムの一つである．HDFS は耐障害性を維持するためにデータのレプリカを複数のノードに分散配置し，ノード故障時には不足レプリカを残りのノード間で補うレプリカ再配置処理を行う．我々はこれまで，シングルラックからなる小規模実クラスタ環境において，HDFS ではレプリカ再配置時のデータ移動に偏りが発生し非効率な処理が行われていることを明らかにし，この問題を解消するために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制御手法を提案し，その有効性を示してきた．
本稿では，シングルラック及びマルチラックで構成したより大規模な環境を想定し，シミュレーションを用いてレプリカ再配置に関する提案手法の有効性を評価する．評価実験より，シングルラック環境ではノード数が増加するにつれて再配置のスループットが向上し，提案手法が有効であることが示された．マルチラック環境では，拡張した制御手法を用いた評価を行った．実験からラック間帯域幅が小さい場合はラック間リンク部分が転送のボトルネックとなり，提案手法の顕著な優位性はみられなかった．一方，ラック間帯域幅が十分大きい場合は，提案手法によりレプリカ再配置のスループットが大幅に改善され，提案手法の有効性が示された．

# Performance Evaluation of Replica Reconstruction Schemes on HDFS using Simulation

Asami Higai[1] Atsuko Takefusa[2] Hidemoto Nakada[2] Masato Oguchi[1]

## 1. はじめに

近年，多種・高性能なセンサネットワークやソーシャルメディアの普及により，大量のデータが日々刻々と生成されるようになった．高エネルギー物理学，生命情報工学などの科学技術分野や商業分野への活用を始めとし，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に対応した処理システムとして，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利用されている．分散ファイルシステムは，データに対して複数のレプリカを生成し，大量のデータノードを用いて分散管理することで可用性や耐故障性を維持している．データノードが故障すると，そのデータノードが管理していたレプリカが一時的に不足し，そのデータを保持している他のデータノードへのアクセス負荷が増加して，システム全体の性能が低下してしまう．そのため不足レプリカの再配置を高速に行い，データ処理システム全体の性能低下を防ぐことが重要である．

オープンソースの分散ファイルシステムでは，Apache Hadoop[1](以下 Hadoop) プロジェクトの Hadoop Distributed File System[2](以下 HDFS) が広く用いられている．そこで我々は分散ファイルシステム HDFS のレプリカ再配置処理に着目し評価を行ってきた．これまでシングルラックからなる小規模実環境において，デフォルトのレ

[1] お茶の水女子大学
Ochanomizu University, Bunkyo, Tokyo 112-8610, Japan
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology(AIST), Tsukuba, Ibaraki 305-8568, Japan

プリカ再配置処理にはデータ移動に偏りが発生し非効率な処理が行われていることを明らかにし，この問題を解消するために，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手法を提案し，有効性を示してきた [5]．しかしながら，HDFS は一般に超大量のデータを管理するために多数のデータノードで構成されるが，大規模環境におけるレプリカ再構成手法の性能特性の調査は不十分である．

本稿では，大規模環境において我々が提案してきたレプリカ再配置の制御手法の有効性を調査する．シミュレーションを用いてシングルラック及びマルチラック環境を構築し，HDFS デフォルトの手法と提案手法の性能を検証する．また，マルチラック環境に対しては提案手法の拡張を行う．評価実験より，シングルラック環境ではノード数が増加するにつれて再配置のスループットが向上し，提案手法が有効であることが分かった．マルチラック環境では，ラック間帯域幅が小さい場合，ラックリンク部分で輻輳が生じ転送のボトルネックとなり，提案手法の顕著な優位性はみられない一方，ラック間帯域幅が十分大きい場合，提案手法によりスループットが大幅に改善していた．このことからラック間リンク部分の輻輳による性能低下を回避するために，ラック間帯域幅に応じて適切にストリーム数を制御することが重要だという知見が得られた．

## 2. Hadoop Distributed File System

HDFS は Google の分散ファイルシステム GFS[3] に基づいて設計された分散ファイルシステムである．HDFS はマスタ・ワーカ型の構成であり，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処理を行う複数台の 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，各ブロックに対して複数のレプリカを複数の DataNode 間で分散して保存することで耐障害性を維持している．

### 2.1 レプリカ配置ポリシ

HDFS がシングルラック構成のクラスタで運用されている場合，全てのレプリカは同一ラック上に配置される．しかし HDFS は一般に超大量のデータを管理するために多数のデータノードを用いて，マルチラック構成のクラスタで運用される．この場合，レプリカは信頼性と性能を考慮して，配置され，最新版の v.2.2.4 では，以下のようにレプリカを生成する．

**第 1 レプリカ**
書き込みがクラスタ内部で開始される場合にはローカルノードに，そうでない場合にはクラスタ内からランダムに選ばれたノードに配置

**第 2 レプリカ**
第 1 レプリカとは異なるラックのノードに配置

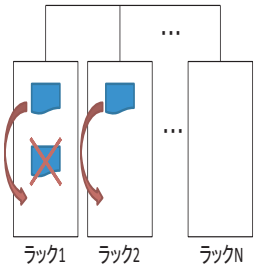

**図 1** 状態 1

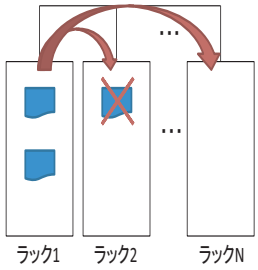

**図 2** 状態 2

**第 3 レプリカ**
第 2 レプリカと同じラック内の異なるノードに配置

**それ以降のレプリカ**
任意のラックの任意のノードに配置

### 2.2 レプリカ再配置処理

ノードが故障した場合にはそのノードが保持するレプリカもなくなるため，その不足レプリカを補うレプリカ再配置処理が行われる．シングルラック時には，生成元も生成先もランダムに選出され，ラック内の転送が行われる．一方マルチラック時には，レプリカ配置ポリシに基づき適当なラックにレプリカを再配置しなければならない．レプリカ数 3 を例にすると，ノードが故障した際のレプリカの配置は状態 1，2 のどちらかである (図 1，2)．Hadoop クラスタでは，経由するスイッチやハブが増加する分ネットワーク帯域幅が減少するという前提の下，ラック間の転送よりラック内の転送を優先する．そのため異なるラックに残りのレプリカがそれぞれ存在する状態 1 においては，同一ラック内にレプリカを再配置するラック内の転送が発生する． 一方，同一ラックに残りのレプリカが存在する状態 2 においては，異なるラックにレプリカを再配置しなければならないので，ラック間の転送が発生する．

## 3. シングルラックにおけるレプリカ再配置処理の評価

我々は効率的なレプリカ再配置を行うために，図 3 のようにリング構造に基づく一方向のデータ転送を行いながら，各ノードの転送ブロック数を均衡化して負荷分散を行う制御手法を提案し，最適化手法とヒューリスティック手法の 2 手法を実装し，評価してきた [5]．評価実験より，6 台の DataNode のうち，1 台を削除した際のレプリカ再配置処理に関して，ヒューリスティック手法が最適化手法と同等の性能を発揮することと，ノード数やデータ量が増加した場合には，最適化手法では最適解の求解時間が指数関数的に増加し非実用的である一方で，ヒューリスティック手法は，デフォルト手法の計算量と同じ計算量でレプリカ再配置のスケジューリングが可能で，大規模環境における有効性を示唆した．

本節ではシングルラックからなるクラスタにおいて，ノー

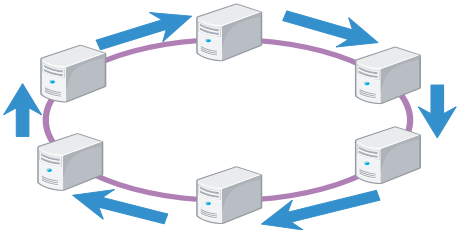

**図 3**　リング構造に基づく一方向のデータ転送

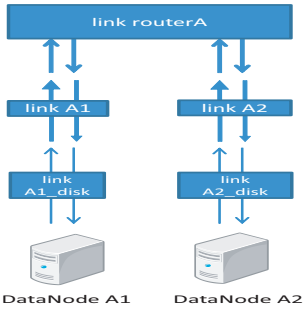


**図 4**　ネットワークトポロジ

**表 1**　実験に用いたパラメータ (シングルラック)

| | |
|---|---|
| DataNode 数 | 6, 8, 16, 32 |
| ブロックサイズ | 67MB |
| レプリカ数 | 3 |
| 削除ノードが保持するブロック数 | 80*(DataNode 数-1) |
| ネットワーク帯域幅 | 125MB/sec |
| ネットワーク遅延 | 0.1msec |
| ディスク性能 | 65，105MB/sec |

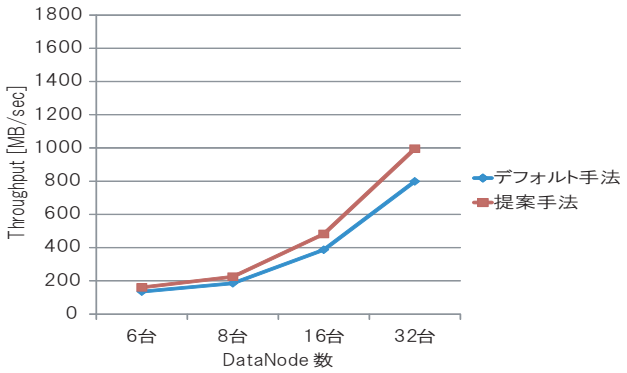


**図 5**　レプリカ再配置のスループット (ディスク性能：65MB/sec)

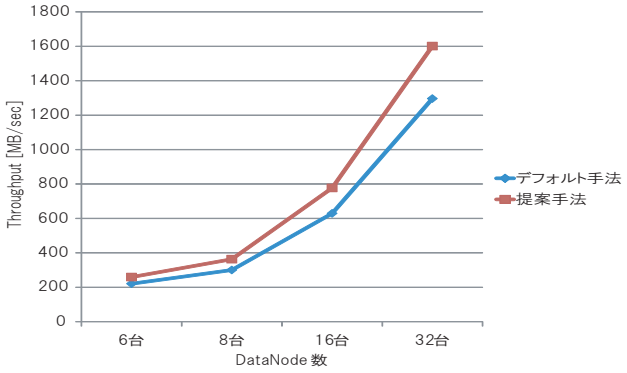


**図 6**　レプリカ再配置のスループット (ディスク性能：105MB/sec)

ド数を増加させた場合においても，提案手法が有効であるかどうかを検証する．ここでは，[5] で提案しているヒューリスティック手法を提案手法と呼んでいる．評価では，再配置のスループットを調査する．

### 3.1　実験概要

HDFS のデフォルト手法と提案手法に対して，ノード数を変化させて，レプリカ再配置処理のスループットをシミュレーションにより評価する．評価には，分散システムのシミュレータ SimGrid[4] を用いた．SimGrid はディスクレベルのシミュレーションを行うことが出来ないため，リンクを追加して，そのネットワーク帯域幅にディスク性能を設定することで，擬似的にディスク処理を表す (図 4)．実験に用いたパラメータを表 1 に示す．DataNode 数は小規模実環境における評価と比較するために，6 台のうちの 1 台を削除する場合と，8, 16, 32 台の場合を評価する．HDFS のデフォルトのブロックサイズは 64MB であり，1 ブロックあたりヘッダなどを含めると転送サイズが 67MB になることから，ブロックサイズは 67MB とした．削除ノードが保持するブロック数は，ノード削除時に各ノードが平均して転送するブロック数 (削除ノードが保持するブロック数/残りのノード数) がノード数に依らず一定になるように表のように設定した．またディスク性能は，小規模実環境における評価と同等のレプリカ再配置のスループットが得られる設定値が 65MB/sec であることと，HDFS 上におけるマシン 1 台当たりのディスク性能が約 105MB/sec であることから，実際のレプリカ再配置処理の結果に基づく値と理想値の 2 つの値を設定した．

### 3.2　実験結果

ノード削除時のレプリカ再配置のスループットを図 5，6 に示す．縦軸が再配置のスループット [MB/sec] で，横軸が DataNode 数である．図 5，6 より，ディスク性能に依らず，DataNode 数の増加に伴い再配置のスループットが増加している．ディスク性能が 65，105MB/sec のときの再配置のスループットの比は，ディスク性能の比にほぼ等しく，ディスク性能が処理速度の要因である．また提案手法により再配置のスループットはデフォルト手法と比較すると 20〜25%向上しており，シングルラックから構成される場合には，大規模環境であっても，提案手法が有効であることが分かる．

## 4. マルチラックにおけるレプリカ再配置処理の評価

### 4.1 マルチラックレプリカ再配置手法

3 章より，提案手法がノード数が増加した場合にも有効であることが分かった．そこでマルチラックにおけるレプリカ再配置の制御手法においても，ラック内の転送，即ち 2.2 節の状態 1 に関しては既提案手法を利用する．また，状態 2 のラック間の転送に関しては，生成先のラックを決めた後に，その生成先ラック内のノードにラウンドロビンに割り当てることで，各ノードの負荷を均衡化することを目指す．生成先のラックの選択は，全ラック数が 2 つの場合は，一意に決定するが，3 つ以上の場合は，任意のラックを選択することとなり，どのラックを生成先のラックに選択するかが重要となってくる．

本節では，まずはシンプルなモデルとして，2 つのラックからなるクラスタでのレプリカ再配置を評価する．そこで，既提案手法を拡張した 2 つのラックからなるクラスタにおけるレプリカ再配置手法について述べる．

まず始めに記号と言葉の定義を与える．DataNode $i$ は，自ラック内への転送回数 $I_i$，異なるラックへの転送回数 $O_i$ の 2 変数を保持する．ラック $j$ は，ラック内の DataNode にラウンドロビンに処理を割り当てる変数 $R_j$ を保持する．”生成元候補ノード”は，不足レプリカの残りのレプリカを保持している DataNode を指す．

1) ラック毎に，DataNode を論理的にリング状に配置する．
2) **(状態 1)** 不足レプリカに関して，生成元候補ノードの中から $I_i$ が最小の DataNode を生成元に選出する．生成先の DataNode は，ラック内のリング構造により一意に決定する．
   **(状態 2)** 不足レプリカに関して．生成元候補ノードの中から $O_i$ が最小の DataNode を生成元に選出する．生成先のラックは一意に決定し，生成先の DataNode は，$R_j$ に基づきラウンドロビンに選出する．

上記のスケジューリングアルゴリズムに基づき生成元と生成先を決定した不足レプリカに対して，状態 2 のレプリカに高い優先度をつけて，先にスケジューリングした後に，状態 1 のレプリカのスケジューリングする手法を優先度付提案手法とする．一方状態 1，2 のレプリカを区別することなく，任意の順にスケジューリングする手法を優先度無提案手法とする．

発生する確率は非常に低いが，ラックに障害が発生した場合には，状態 2 のレプリカに関しては修復不能となってしまうため，状態 2 のレプリカを先にスケジューリングすることは，耐障害性の向上に繋がる．一方でラック間の転送が集中すると，ラック間の帯域に輻輳が発生し，処理スピードの低下に繋がる．以上より，優先度付提案手法は，処理スピードより耐障害性を重視した手法で，優先度無提案手法は，耐障害性より処理スピードを重視した手法である．

**表 2** 実験に用いたパラメータ (マルチラック)

| | |
|---|---|
| DataNode 数 | 8*2, 16*2, 32*2 |
| ブロックサイズ | 67MB |
| レプリカ数 | 3 |
| 削除ノードが保持するブロック数 | 80*(DataNode 数-1) |
| ラック内ネットワーク帯域幅 | 125MB/sec |
| ラック内ネットワーク遅延 | 0.1msec |
| ラック間ネットワーク帯域幅 | 125Mb/sec，1.25GB/sec |
| ラック間ネットワーク遅延 | 0.1，1msec |
| ディスク性能 | 65，105MB/sec |

### 4.2 実験概要

HDFS のデフォルト手法と 2 つの提案手法に対して，ノード数を変化させて，レプリカ再配置処理のスループットを評価する．実験に用いたパラメータを表 2 に示す．同一ノード数からなる 2 つのラックにおいて，ある一方のラックの内の 1 台を削除するものとし，DataNode 数は 1 ラックあたり 8，16，32 台とした．ラック間ネットワークは，1Gigabit Ethernet 及び 10Gigabit Ethernet の場合を想定し，帯域幅 125MB/sec，遅延 0.1msec 及び帯域幅 1.25GB/sec，遅延 1msec の設定とした．また，同様にシミュレーション環境には SimGrid を用いた．

### 4.3 実験結果

各手法を用いた際の，ノード削除時のレプリカ再配置のスループットを図 7，8，9，10 に示す．縦軸が再配置のスループット [MB/sec] で，横軸が DataNode 数である．図 7，8 はラック間帯域幅が 125MB/sec，図 9，10 は 1.25GB/sec のときの結果を表す．

図 7，8 より，ラック間帯域幅が小さい場合には，ノード数の増加に伴い若干再配置のスループットが向上しているものの，約 385MB/sec で飽和している．これは，ラック内転送速度がディスク性能の 65，105MB/sec であり，ラック間転送速度が 125MB/sec であるため，DataNode 数が増えるとラック間転送部分に輻輳が発生してボトルネックとなっているためである．ノード数が増加すると再配置のスループットが若干向上しているのは，3 章よりラック内の転送性能はノード数の増加に伴い向上するので，ラック間での転送処理の停滞の影響が中和されたためと考えられる．また優先度付提案手法の再配置のスループットがデフォルト手法よりも低くなっている．これはラック間転送を伴うレプリカ再配置を先に優先的に行っているため，処理の序盤では他の手法よりも一層ラック間リンク部分に処理が集中し処理が停滞しているためだと考えられる．

また図 9，10 よりラック間帯域幅が大きい場合には，ノー

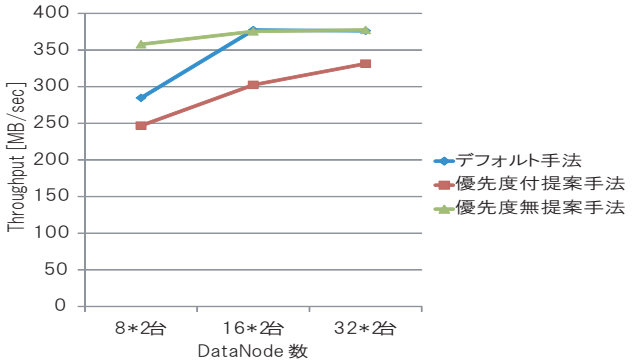


**図 7** レプリカ再配置のスループット
(ラック間帯域幅：125MB/sec，ディスク性能：65MB/sec)

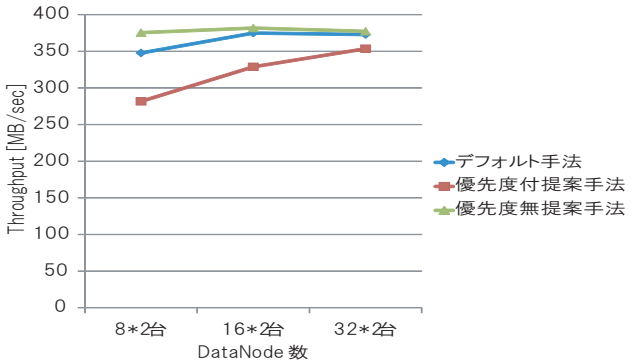


**図 8** レプリカ再配置のスループット
(ラック間帯域幅：125MB/sec，ディスク性能：105MB/sec)

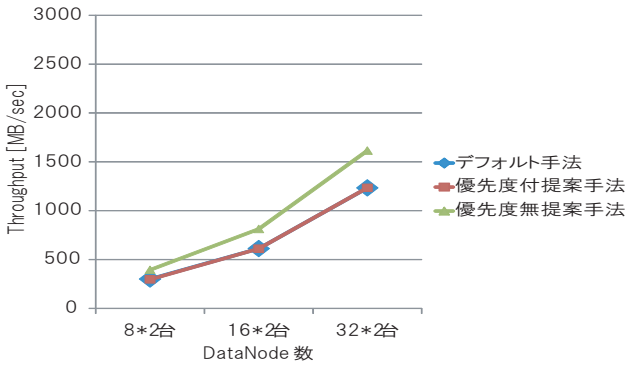


**図 9** レプリカ再配置のスループット
(ラック間帯域幅：1.25GB/sec，ディスク性能：65MB/sec)

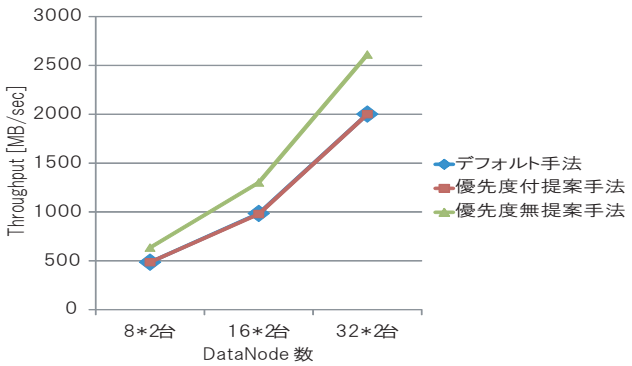


**図 10** レプリカ再配置のスループット
(ラック間帯域幅：1.25GB/sec，ディスク性能：105MB/sec)

ド数の増加に伴い再配置のスループットが増加し，ディスク性能比と再配置のスループット比がほぼ等しくなっている．優先度付提案手法とデフォルト手法の再配置のスループットはほぼ同一であるが，優先度無提案手法はデフォルト手法と比較すると約 30%スループットが向上している．これはラック間帯域幅が十分に大きく，処理のボトルネックの要因がデータ移動の偏り及びディスク性能であるため，各 DataNode の負荷を均衡化しようとする制御手法が有効に作用したためである．

以上より，マルチラック環境においては，ラック間リンク部分の輻輳による性能低下を回避するために，ラック間帯域幅に応じて適切にストリーム数を制御することが重要だと分かった．また優先度付提案手法は，デフォルト手法の性能と同等かそれ以下となってしまうが，耐障害性を考慮すると実用範囲内であることが分かった．

## 5. 関連研究

鈴木らは，広域環境におけるクラスタ間で複数のファイルの複製を効率的に行う手法について考察し，アルゴリズムの提案と評価を行っている [6]．クラスタ間で効率良くファイルを複製するには，ネットワークの通信性能低下の回避と単一ディスクへのアクセス集中による性能低下の回避が重要であると述べている．そこで，クラスタ間のファイル複製に対する適切な転送ファイルの選択と適切な並列ファイル転送スケジューリングをグラフ問題としてモデル化し，更にそれらの結果から，特定ノードに複製処理が偏っている場合，その送信元クラスタ内の他のノードにファイルを複製し処理を肩代わりしてもらい，所望の外部ノードにファイル転送を行うアルゴリズムを提案している．評価実験より，クラスタ間での複数ファイルの複製において提案アルゴリズムの有効性を示している．

## 6. まとめと今後の課題

HDFS のレプリカ再配置処理に関して，既発表研究においてラック内におけるレプリカ再配置手法を提案し，実環境で有効性を示してきた．本研究では，大規模環境を想定して，シングルラック及びマルチラックからなるクラスタ上でレプリカ再配置手法の性能を検証した．マルチラックの場合には，ラック内の転送に関しては既提案手法を利用し，ラック間の転送に関しては，ラウンドロビンに処理を割り当てて負荷分散を図るように制御手法を拡張し，実装した．評価実験より，シングルラックの場合には，ノード数の増加に伴い再配置のスループットが向上し，提案手法が有効であることを示した．マルチラックの場合には，ラック間帯域幅が小さい場合は，このリンク部分の転送がボトルネックとなるため，デフォルト手法に対する提案手法の優位性は見られなかった一方，ラック間帯域幅が十分大きい場合，優先度無提案手法が有効であることを示した．

これらよりラック間リンク部分の輻輳による性能低下を回避するために，ラック間帯域幅に応じて適切にストリーム数を制御することが重要だという知見が得られた．

今後の課題として，ラック間転送の際に，ラック間のリンク部分のストリーム数を適切に制御し，効率的な処理を実現するように提案手法を改良することと，ネットワークの帯域や遅延を変化させて評価を行うことが挙げられる．また本稿では 2 つのラックからなるシンプルなモデルの評価を行ったが，今後は更に複数のラックからなるモデルについても評価していきたい．

## 参考文献


[1] Tom White, Hadoop: The definitive guide, trans. Ryuji Tamagawa. O'Reilly JAPAN, 2010.

[2] Dhruba Borthakur. "HDFS Architecture," 2008 The Apache Software Foundation.

[3] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung (October 2003), "The Google File System," 19th Symposium on Operating Systems Principles (conference), Lake George, NY: The Association for Computing Machinery, CiteSeerX: 10.1.1.125.789, retrieved 2012-07-12.

[4] SimGrid. http://simgrid.gforge.inria.fr/

[5] Asami Higai, Atsuko Takefusa, Hidemoto Nakada, Masato Oguchi, "A Study of Effective Replica Reconstruction Schemes at Node Deletion for HDFS", 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, (To appear)

[6] 鈴木克典，建部修見，"ＰＣクラスタ間ファイル複製スケジューリング" 論文誌コンピューティングシステム (ACS)，情報処理学会，Vol.3，No.3，pp.113-125, 2010 年 9 月