HDFS における効率的なレプリカ再配置手法の提案と評価

理学専攻　情報科学コース　日開　朝美

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り，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，大量の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．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が一時的に不足し，そのデータを保持している他の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する．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(Hadoop) プロジェクトの Hadoop Distributed File System(HDFS) が広く用いられている．しかしながら HDFS のレプリカ再配置では，生成元・生成先をランダムに選択するため，データ移動に偏りが生じ，非効率なレプリカ再配置処理が行われている．この問題を解消するために，データ移動の偏りを解消し，各ノードの負荷を均衡化させることにより効率的なレプリカ再配置処理を実現するレプリカ生成元・生成先ノードの選出手法と，可用性を考慮したスケジューリング制御手法を提案する．HDFS はラック構成を考慮したレプリカ配置が行われるため，不足レプリカの再配置先ラックとして，(1)1 ラック，(2)2 ラック，(3)3 つ以上のラックからなるクラスタの 3 つの環境において，その自由度が変わってくる．それぞれの環境において，制御手法を提案し，評価する．本稿では特に，(3)3 つ以上のラックからなるクラスタ環境における HDFS のレプリカ再配置制御について述べる．

2　HDFS のレプリカ配置ポリシーとレプリカ再配置

複数のレプリカはレプリカ配置ポリシーに基づいて，第 1 レプリカはローカルノードに，第 2 レプリカは第 1 レプリカと異なるラックに，第 3 レプリカは第 2 レプリカと同一ラックの異なるノードに配置される．データ転送に関しては，ラック間転送よりもラック内転送が優先される．ノード故障などによりレプリカが不足した場合には，レプリカ配置ポリシーに基づき残りのノード間で不足レプリカを補う．残りのレプリカが異なるラックに存在する場合には，ラック内に再配置を行うラック内転送が行われ，同一ラックに存在する場合には、異なるラックに再配置しなければならないため，ラック間転送が行われる．このレプリカ配置ポリシーを満たしながら，レプリカの生成元・生成先ノードはほぼランダムに選出される．各生成元ノードが同時に送信できるストリーム数は 2 である．

3　効率的なレプリカ再配置手法の提案

HDFS のデフォルトのランダム手法によるレプリカ再配置では，(i) ランダムな生成先ノードの選出による，各ノードの受信ブロック数の偏り，(ii) ラック間転送に関して，削除ノードを含むラックのノードが生成元にはなり得ないが，生成先にはなり得ることによる，削除ノードを含むラックのノードの受信ブロック数の偏り，により非効率な処理が発生している．

これらの問題を解消するために，データ移動の偏りを解消し，各ノードの負荷を均衡化させることにより効率的なレプリカ再配置処理を実現するレプリカ生成元・生成先ノードの選出手法と，可用性を考慮したスケジューリング制御手法を提案する．

3.1　レプリカ生成元・生成先ノードの選出

各ノードの送信及び受信ブロック数が等しくなるように生成元・生成先ノードを選出する．ここで，削除ノードを含むラックを failure rack，それ以外の削除ノードを含まないラックを normal rack と呼ぶ．データが一様に各ノードに保存されているとすると，ラック数を R，1 台のノードが削除された際に複製の必要なブロック数を B とすると，確率的にラック間転送が行われるブロック数 $B_{inter} = \frac{1}{3}B$，ラック内転送が行われるブロック数 $B_{inner} = \frac{2}{3}B$ である．各ラックの送信ブロック数が均衡化するのは $\frac{B}{R}$ であるため，表 1 のように再配置処理を割り当てる．failure rack は送受信の転送ブロック数を均衡化するために，ラック間転送には一切関与しないものとし，その分ラック内転送の負荷を増加させる．

表 1: 各ラックへの再配置処理の割り当て

	normal rack	failure rack
ラック間転送	$B_{inter} \times \frac{1}{R-1}$	0
ラック内転送	$\left(B_{inner} - \frac{B}{R}\right) \times \frac{1}{R-1}$ $= B_{inner} \times \frac{2R-3}{2R(R-1)}$	$B \times \frac{1}{R}$ $= B_{inner} \times \frac{3}{2R}$

生成元の選出に関して，各ノードのラック内転送生成元選出回数とラック間転送生成元選出回数をカウントし，ラック間転送に関しては，生成元候補ノードの中から，ラック間転送生成元選出回数が最小のノードを選出することで，表 1 を満たすことが出来る．ラック内転送に関しては，生成元候補ノードの中から，表 1 の処理の割り当てを考慮し，normal rack の場合は $3(R-1)$ 倍，failure rack の場合は $(2R-3)$ 倍して，ラック内転送生成元選出回数の比較を行い，選出回数が少ないノードを選出する．

生成先の選出は指向性リング構造に基づいて，1 つ先のノードを選出する．指向性リング構造に基づくデータ転送を用いる理由は，大規模クラスタでは multi-drop-chain 型の論理ネットワークトポロジに基づくデータ転送が効

率が良い方法であると示されているからである [1]．ラック内転送のために，ラック毎に論理的なリング構造を構成し，ラック間転送のために，normal rack に含まれる全てのノードを繋ぐ，1 つの論理的なリング構造を作成する．この時，前後のノードは異なるラックに属するノードとなるようにする．このようなリング構造により，生成元ノードが決定すると，ラック間及びラック内転送それぞれにおいて，一意のノードが生成先として選出される．そして各ノードの送信ブロック数の均衡化に伴い，受信ブロック数も付随して均衡化される．

3.2 スケジューリング制御

発生することは稀であるが，万が一レプリカ再配置中にラック全体に渡る障害が発生した場合，同一ラックに残りのレプリカが存在するブロックは復元不可能になってしまう．そのため，ラック間転送が必要なブロックを先に再配置することは可用性の向上に繋がる．そこで，上記の手順により生成元・生成先ノードが決定した不足レプリカについて，ラック間転送を行うブロックに高い優先度をつけて先にスケジューリングした後に，ラック内転送を行うブロックをスケジューリングする制御手法を優先度付手法とする．一方，これら 2 つの状態のブロックを区別することなく，任意の順にスケジューリングする制御手法を優先度無手法とする．

4 提案手法の評価

デフォルト手法と前章で提案した優先度無手法，優先度付手法を用いて，ある 1 つのラックのうちの 1 台のノードを削除した際のレプリカ再配置を表 2 に示す 3 ラック構成の環境において，分散シミュレータの SimGrid-3.10 を用いてシミュレーションにより評価する．

表 2: シミュレーション環境

1 ラックあたりのデータノード数	8, 16, 32
ブロックサイズ	67MB(default)
レプリカ数	3(default)
不足ブロック数 (削除ノードが保持するブロック数)	80*正常なノード数
ラック内ネットワーク帯域幅，遅延	125 MB/sec
ラック内ネットワーク遅延	0.1 msec
ラック間ネットワーク帯域幅，遅延	1.25 GB/sec
ラック間ネットワーク遅延	0.1 msec
ディスク性能	67 MB/sec

各手法のレプリカ再配置の実行時間を図 1 に，1 ラックあたりのデータノード数が 8 台の場合の 1 ブロックあたりの平均転送時間を表 3 に示す．図 1 より，提案手法によりレプリカ再配置の実行時間が減少し，最大で 18% 削減できた．1 ラックあたりのデータノード数が 8，16 台の場合，優先度無手法より優先度付手法の方が実行時間削減に有効である．優先度付手法の性能が高い理由は，優先度付手法は，normal rack のデータ転送に関して，処理の冒頭はラック間転送のみが行われ，それらが終了した後にラック内転送が実行されるため，時系列的にみると生成元と生成先が一対一に対応した転送が行われ，最大受信ブロック数が 2 に制限される．一方で優先度無手法では，ラック間転送とラック内転送が混在し，リング構造に基づいたデータ転送であっても、あるデータノードに関して最大で 4 つのブロックを受信してしまう事態が発生す

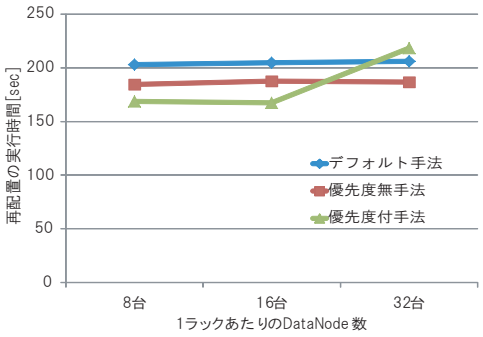

図 1: 各手法におけるレプリカ再配置の実行時間

表 3: 1 ブロックあたりの平均転送時間 [sec]

		デフォルト	優先度無	優先度付
ラック間転送	A to B	4.364	4.666	4.036
	A to C	5.867	0	0
	B to A	4.328	4.565	4.015
	B to C	5.710	0	0
ラック内転送	A	4.260	4.307	3.971
	B	4.323	4.500	3.980
	C	6.225	3.990	3.990

normal rack : A, B　failure rack : C

るからである．このことから，優先度付手法は可用性の向上だけでなく，比較的小規模な環境においてはより効率の良い手法である．しかしながら，ラック内のデータノード数が 32 台と多い場合，優先度付手法ではラック間転送の集中により，ラック間のネットワーク帯域が飽和して，性能が低下している．そのため，適切なストリーム制御が必要であることが分かる．また表 3 からデフォルト時には，failure rackC に関連するデータ転送時間が長くなってしまっていたが，提案手法により，全ての転送形態についてほぼ等しく，効率良く転送が行われていることが分かる．

5 まとめと今後の課題

HDFS のレプリカ再配置において、データ移動の偏りを解消し，各ノードの送受信のブロック数を均衡化することで効率良くレプリカ再配置を行う制御手法を提案し，評価を行った．各ノードの送信ブロック数が均衡化するような処理の割り当て比率に基づいて生成元ノードの選出を行い，生成先ノードは指向性リング構造に基づいて選出し，一対一のデータ転送を行うことで，受信ブロック数も均衡化し効率良く処理を行う手法を提案した．評価実験から，提案手法により，各ラックの送受信ブロック数が均衡化され，最大で再配置の実行時間を 18% 削減することができた．

今後の課題は，ストリーム数を適切に制御し，より大規模な環境においても効率良く転送を行えるように制御することである．

参考文献

[1] 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.