7T-06

Hadoopにおけるreduceタスク割り当ての改善手法

白水 達也† 芝 公仁†
†龍谷大学理工学部

1 はじめに

近年，複数台の汎用計算機で処理を実行する並列分散処理基盤 Hadoop が活用されている．Hadoop は大きく分けて 3 つのシステムから構成される [1]．ファイルシステムの Hadoop Distributed File System(以下 HDFS) と，リソース管理・スケジューリングなどを行うアプリケーション実行フレームワーク YARN，YARN 上で動作する MapReduce アプリケーションである．MapReduce では，ユーザは入力を並列に処理する Map 処理とその結果を集約する Reduce 処理を定義することで，ノード間通信や障害対策などを意識せずに分散処理アプリケーションを作成できる．

Hadoop では膨大なデータを扱うため，ノード間のデータ通信がボトルネックとなる場合がある．Hadoop のタスクスケジューラは Map タスクの場合，入力データがあるノードに割り当てを試みるが，Reduce タスクの場合は単純に先行するタスクから割り当てを行い，各ノードの中間データ量を考慮しない．結果として，ネットワークリソースを無駄に使用し，処理速度を低下させる場合がある．

本研究は，Map タスクの出力した各ノードの中間データの量を考慮し Reduce タスクの割り当てを行うことで，データ通信量を削減し Hadoop の効率を高めることを目的とする．

以下，本稿では，2 章で MapReduce アプリケーション，3 章で Reduce タスクスケジューリングのシステム構成，4 章で割り当てタスク選択手法について述べる．

2 MapReduce アプリケーション

MapReduce とは，Google によって 2004 年に導入された分散処理のためのプログラミングモデルである．MapReduce アプリケーションのデータフローを図 1 に示す．

Map タスクの入力となるデータはスプリットという単位に分割される．Hadoop の場合，スプリットは HDFS により，各ノードに保存されている．スプリットはそれぞれに対応する Map タスクがあり，スプリットがあるノードに優先的に Map タスクを割り当てることで通信量を削減している．また，ノードローカルに割り当てられなかった場合でも，ラックローカルの割り当てを試みる．Map タスクの出力である，キーとバリューで構成される中間データは，対応する Reduce タスクごとにまとめられる．この中間データをパーティションという．Reduce タスクを割り当てられたノードは，他のノードに保存されている対応するパーティションをネットワークを介して取得する．すべての対応するパーティションの取得を終えると，パーティションはマージされ，Reduce タスクの入力データとなる．Hadoop の場合，Reduce タスクの出力は HDFS に格納される．

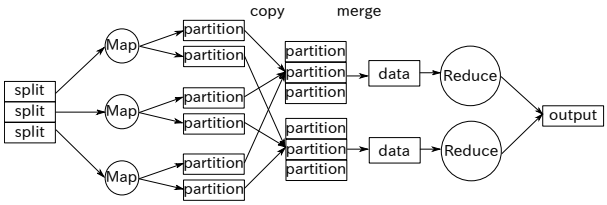

図 1 MapReduce アプリケーションのデータフロー

3 システム構成

Reduce タスクスケジューリングのシステム構成を図 2 に示す．図の RM はリソースマネージャノード，AM はアプリケーションマスタノードを表す．

まず，Map タスク処理ノードで Map タスクが出力したデータ量をカウントし，タスクカウンタに格納する．タスクカウンタはハートビートにより，アプリケーションマスタに送信される．この通信は各ノード 3 秒に 1 回の頻度で行われる．リソースマネージャがアプリケーションマスタのリクエストに応えコンテナを割り振ると，アプリケーションマスタはパーティション容量とコンテナのノードを基に割り当てる Reduce タスクを決定する．この Reduce タスクを選択する手法については次章で述べる．アプリケーションマスタは割り当てタスクが決定すると，当該コンテナのノードマネージャと通信して処理を開始させ，Reduce タスクの割り当ては完了する．通常，タスクカウンタのパーティション容量のデータは，中間データの容量に対し

Improving the reduce task scheduler in Hadoop
Tatsuya Shirouzu†, Masahito Shiba†
†Faculty of Science and Technology, Ryukoku University

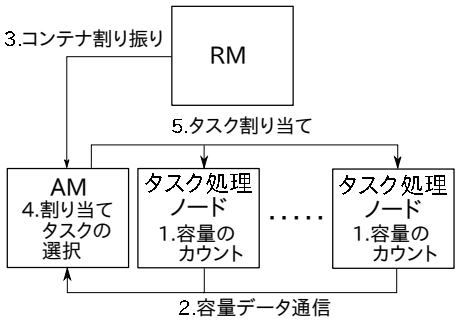

図 2　システム構成

て十分に小さい．

4　割り当てタスク選択手法

4.1　最大格納データ量割り当て

最大格納データ量割り当てでは，タスクを割り当てるノードが格納しているパーティションごとの容量を参照し，最も容量の大きいパーティションに対応する Reduce タスクを割り当てる [2]．これにより，入力データが少ないノードへの割り当てを防ぐことができるため，結果として通信量が削減される．

4.2　ノード最小通信量割り当て

ノード最小通信量割り当てでは，タスクのために他ノードから受信するデータ量が最も少ないタスクを割り当てる．具体的には，割り当てノード以外のノードにある対応パーティション容量を集計することで，タスク割り当て時にそのタスクのために受信する必要があるデータ量を求め，これが最も少なくなるタスクを割り当てる．これにより，クラスタ全体でも通信量を削減できる．

4.3　最大通信量差順割り当て

最大通信量差順割り当ては，ノードに割り当て順を設け，その順番で通信量の小さい割り当てタスクを選択することによって，クラスタ全体の通信量を削減する手法である．

具体的には，各タスクを割り当てたそれぞれの場合について，他ノードから受信する必要があるデータ量を計算する．その結果からノードごとに最も小さい通信量と最も大きい通信量の差を求める．この差が大きいほど，どのタスクが割り当てられるかによる通信量の差が大きくなりやすく，順番に小さい通信量のタスクを選択することで通信量を削減できる．

ノード n1 に Reduce タスクを割り当てるときの最大通信量差順割り当ての処理を図 3 に示す．最初に，ノード最小通信量割り当てと同様に受信する必要のあるデータ量を求め，通信データ量表とする．次にノード別に最大通信量と最小通信量の差を算出し，これが大きいノードから順番を付ける．図 3 では，ノード n2，n1，n3，n4 の順で割り当てを行う．最初に選択するノードは n2 で最も小さい通信量 37 となる．タスク r3 を選択する．二番目のノード n1 で最も小さいのは 49 の r3 だが，すでにノード n2 に r3 を選択したため，次に小さい 40 に対応するタスク r4 を選択する．割り当てコンテナのあるノード n1 への割り当てタスクを選択したため，ノード n3，n4 への割り当てを計算する必要はなく，処理は終了する．

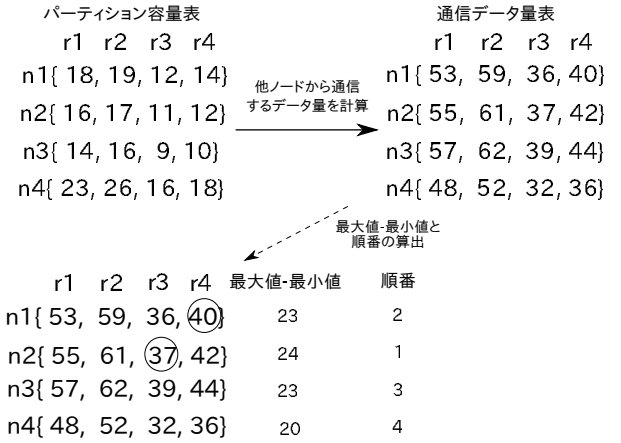

図 3　最大通信量差順割り当ての処理

この割り当ては，各 Map タスクが出力するパーティション容量に偏りがないとき，クラスタ全体で最小通信量の割り当てを行える．しかし，いずれかのノードで一つのパーティション容量のみ極端に大きく，他のパーティション容量が小さい場合などは通信量を増加させる場合がある．また，前述した二つの割り当てに比べ計算量は多くなるため，処理時間に悪影響が出る可能性がある．

5　おわりに

本稿では Hadoop におけるデータ通信量を考慮した Reduce タスク割り当て手法を提案した．本手法により，Reduce タスク割り当てを改善し，Hadoop クラスタの通信量を削減した．

参考文献

[1] Tom White，Hadoop．オライリー・ジャパン，2013．-

[2] 阪東 幸二，芝 公仁．MapReduce における通信データ量を考慮した Reduce タスク割り当て手法 IPSJ SIG Technical Report, 2014.