Hadoop と Spark

岡田 祥，富岡 亮登
Sho OKADA，Ryoto TOMIOKA

1　はじめに

近年，記録媒体の性能向上やインターネットの普及により企業が保持するデータ量・種類が増大している．その結果，数 TB から数 PB の莫大な量かつ，様々な種類のデータ（ビッグデータ）が発生し，利用が盛んに行われている [1]．ビッグデータを処理する場合，莫大なデータを効率的に処理する方法が問題であり，効率的に処理を行う方法として複数のサーバで分担して処理を行う分散処理が注目された．分散処理が注目される中，分散処理を行う仕組みとして MapReduce が発表された．そして，MapReduce の技術理論を取り入れ実装されたのが Hadoop である．次に，Hadoop とは違い，繰り返し処理を含む機械学習やデータマイニングなどの利用に適した Spark が開発された．既に Hadoop は様々な企業で実際に利用されており，Spark も研究が盛んに行われている．

2　Hadoop と Spark

2.1　Hadoop

Hadoop とは，大規模データを効率的に分散処理，管理するための仕組みである [2]．分散処理とはネットワーク接続された複数のサーバを用いて処理を分担し，並列処理を行うことである．分散処理ではサーバを増設することで，サーバ全体の処理能力の向上や処理時間の短縮が可能である．よって，Hadoop も同じ長所を持つ．Hadoop 独自の特徴は，あらかじめ用意されたデータを処理するバッチ処理に適していることである．

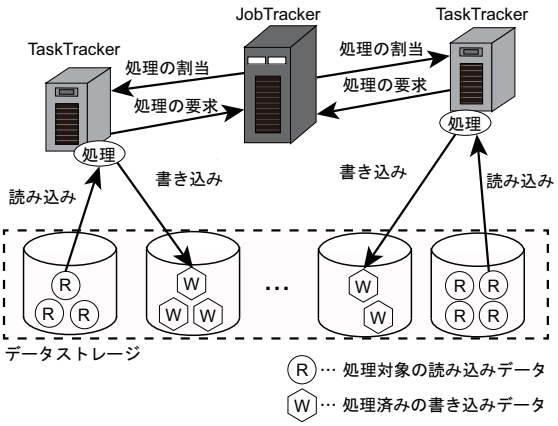

Fig.1　Hadoop の構成

Fig. 1 に Hadoop の構成を示す．Hadoop は処理対象・処理済みのデータを保管する場所であるデータストレージと，処理の割り当てを行う JobTracker および TaskTracker に分かれる．データストレージは複数箇所存在し，データを別々の場所に保存する．TaskTracker は各 TaskTracker で別々の処理を行い，結果をデータストレージに格納する．

次に，Hadoop における処理を述べる．まず，JobTracker が TaskTracker に処理を依頼する．依頼を受けた TaskTracker は依頼を完了するために必要なデータをデータストレージから読み込む．TaskTracker は読み込んだデータを利用して処理を行った後，結果をデータストレージに書き込む．最後に TaskTracker が JobTracker に次の処理を割り当てるように要請する．

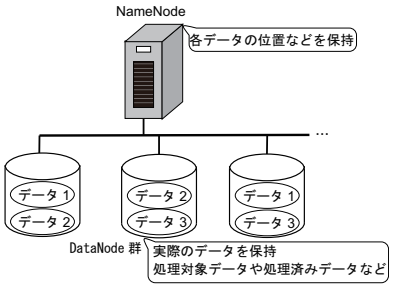

Fig.2　HDFS の構成図

次に，データを保管するストレージについて述べる．Hadoop ではデータを複数のストレージに分散して保管する HDFS（Hadoop Distributed File System）方式のストレージを採用している．HDFS によるストレージの構成を Fig. 2 に示す．HDFS は上位の NameNode と下位の DataNode に分かれている．NameNode は分割したデータの位置情報やファイル名などの情報を格納している．DataNode は実際に処理を行う対象のデータや処理の結果を格納している．HDFS はデータを分割して，いくつかのブロックに分けて保存する．そして，分割したデータのコピーを作成し，保存する．もしデータが破損したとしても，コピーを取ったデータから元のデータを復旧することができるので耐故障性に優れている．

2.2　MapReduce

Hadoop では分散処理の仕組みである MapReduce の技術を基に処理を行っている．MapReduce には手順があり，Map，Shuffle/Sort および Reduce に分かれる．Map ではデータを Key と Value の組に対応付け，分析の準備を行う．Shuffle/Sort では特定の Key の組を集約する．最後に Reduce で結果を得るための演算を行う．

Fig. 3 に MapReduce での処理の一例を示す．ここでは文字列から単語の出現数を数える例を考える．まず Map で出現単語を Key として，出現回数を Value として対

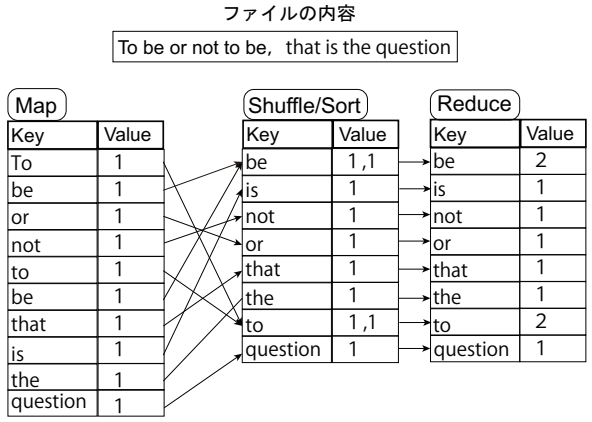

Fig.3　MapReduce 処理の例

応する．例えば，be なら 1 というように対応する．次に Shuffle/Sort で Key ごとに Value を並び替える．結果，単語ごとに値が並ぶので複数回登場した単語と単語に対応した Value が集約する．例では，be の Value が 1，1 に変化している．最後に Reduce で Key の Value を加算すれば結果が出力される．したがって，be なら Value は 2 となり，正しい集計が完了する．

2.3　Spark

Spark は Hadoop とは違い，同じデータを繰り返し利用する処理を効率的に行うことに適した分散処理の仕組みである [3]．Hadoop は処理を行う場合，データストレージへのアクセスが必須であった．しかし Spark は，読み込んだデータをメモリ上に保持するインメモリという技術が用いられている．インメモリの仕組みにより，データストレージからの読み込む回数を減らすことができる．結果として，Hadoop のように処理対象のデータを毎回データストレージから読み込む必要がないので，機械学習やデータマイニングといった，繰り返し同じデータを読み込む必要がある処理において，Hadoop より効果的である．具体的には MapReduce を用いた Hadoop と比較して，インメモリ環境で最大 100 倍高速である．

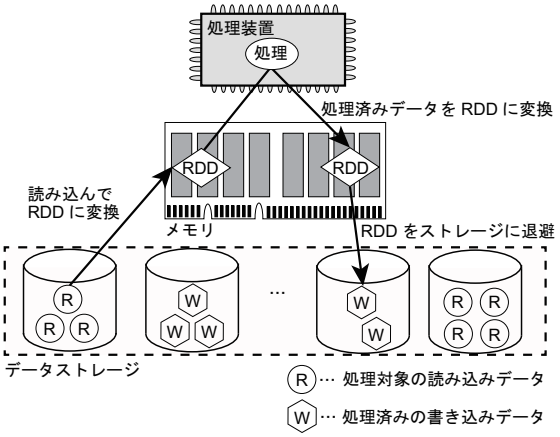

Fig.4　RDD を用いた処理

Spark では処理対象のデータを全てメモリ上に保持する．メモリ上で保持するデータの形式が RDD(Resilient Distributed Datasets) である．RDD を用いた処理は Fig. 4 に示す流れで行われる．まず，メモリ上に処理対象データを保持するため，読み込んだ処理対象データを RDD に変換する．変換が完了したデータは RDD の形式でメモリ上に保持される．次にデータを変換した RDD を用いて処理を行う．そして，処理結果を RDD の形式でメモリ上に保持する．もしメモリ内が飽和した状態で，データストレージから読み込む必要がある場合，メモリ上の RDD をデータストレージに書き込むことでデータを退避する．データの退避が完了すると，メモリに空き領域ができるので，メモリ上で処理すべきデータの置く場所が確保できる．

3　Hadoop/Spark の活用事例

NTT データでは，NTT ドコモ向けに，計算ノードが 1000 台以上の Hadoop のシステムを構築した． システムには大量のシステム運用情報などを処理する目的があり，サーバ増設可能かつ，高い故障耐性を実現していた．しかし，データ処理内容の高度化，多様化に伴い，Hadoop よりも高速かつ柔軟に処理したいという需要が生まれた． 言い換えれば，MapReduce では実現が難しい処理の仕組みが必要となった．高速かつ柔軟なシステムという需要に対応するため，Spark を導入し，機能評価を行うと共に，いくつかの処理をモデル化し，試験的に Spark へと移植が行われた．現在 Spark は研究段階にあるが，企業にも可能性は認識されており，いずれは Hadoop と同様に，業務を円滑に進める手段として利用されると考えられる．

4　今後の展望

Hadoop と Spark は処理可能なデータ量に差が存在する．Hadoop はストレージ読み込みが前提のため，数 PB のデータであっても処理が可能である．対して Spark はインメモリで処理を行うため，数 TB 程度のデータしか処理を行うことができない．よって今後は Hadoop と Spark，どちらの長所も生かした効率的な分析処理の利用が盛んに行われると考えられる．具体的にはドリルダウン分析を用いる．最初に，Hadoop で数 PB のデータを荒い粒度で処理することで分析範囲を数 TB まで縮小する．次に Spark で数 TB のデータを細かな粒度で処理を行う．ドリルダウン分析により Hadoop の数 PB 以上のデータ量であっても処理可能であるという長所，Spark のインメモリによる高速さという長所の両方を利用することができるので，今後はドリルダウン分析の利用が増加すると考えられる．

参考文献

1) Manoochehri, Michael, ビッグデータ テクノロジー完全ガイド,（小林啓倫・訳），マイナビ pp.99-114，2014
2) White，Tom，Hadoop,（玉川隆二・兼田聖士訳），オライリー・ジャパン pp.15-79，2010
3) NTT データほか，Apache Spark 入門 動かして学ぶ最新並列分散処理フレームワーク，翔泳社，pp.3-25,2015