

ビッグデータを支える基盤技術

—「Hadoop」の現状と今後の進化—

近年、ビッグデータが実務に活用されるようになった背景には、大量に発生するデータを高速処理するための基盤技術の革新がある。本稿では、ビッグデータ活用を支える基盤技術の中でも、分散処理のフレームワークとして最も注目され、ビッグデータの中核的な技術となりつつある「Hadoop」に焦点を当て、その処理の仕組みや有効性、今後の進化について解説する。

ビッグデータを処理するための基盤技術

ある試算によると、全世界で生成されるデータの量は、2011年にはPB（ペタバイト）、EB（エクサバイト）を超えてZB（ゼタバイト）に達するとされていた。このような従来の常識を超える大量のデータを処理できる基盤技術とはどのようなものだろうか。

従来、データの保存や分析にはRDB（リレーショナルデータベース）やDWH（データウェアハウス）が基盤技術として用いられてきた。しかしRDBやDWHが現実的な時間内で処理可能なデータ量はせいぜいTB（テラバイト）程度であり、それ以上の大量データを扱うには性能不足となってきた。そこでこれらの技術は低レイテンシー（高速）化とスループット（単位時間当たりの処理量）向上という2つの方向で改良され、商用製品やオープンソースソフトウェアの形で提供されるようになった（図1参照）。

大量のデータを高速処理（保存）

するためには、分散キャッシュ（複数のサーバーのキャッシュを統合的に扱う技術。キャッシュとはデータベースへの問い合わせ結果をメモリー上に保持しておくこと）やインメモリーDB（データをメモリー上に格納するデータベース）が使われてきた。

さらに高速化するためにはNoSQL（リレーショナル型でないデータベース）が利用され

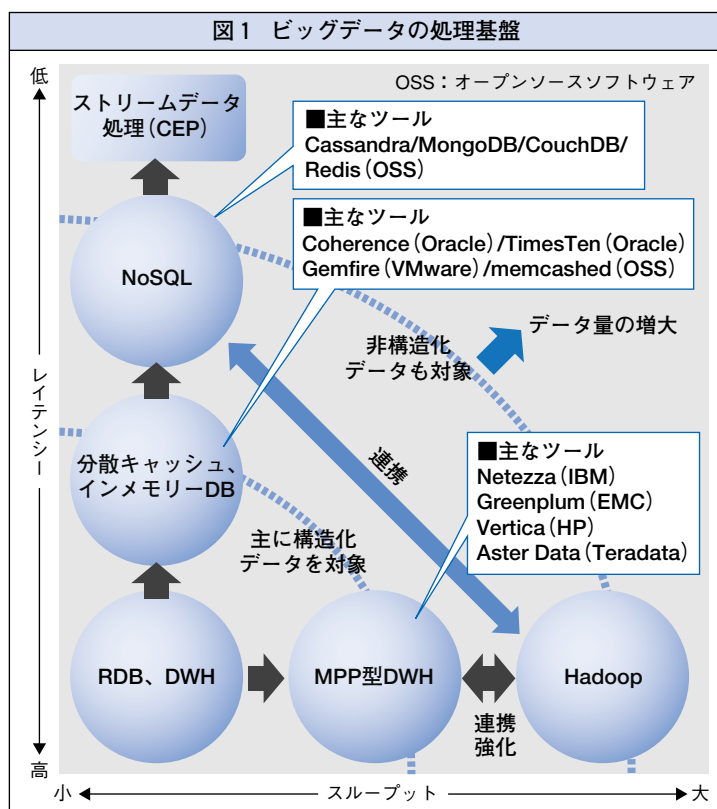
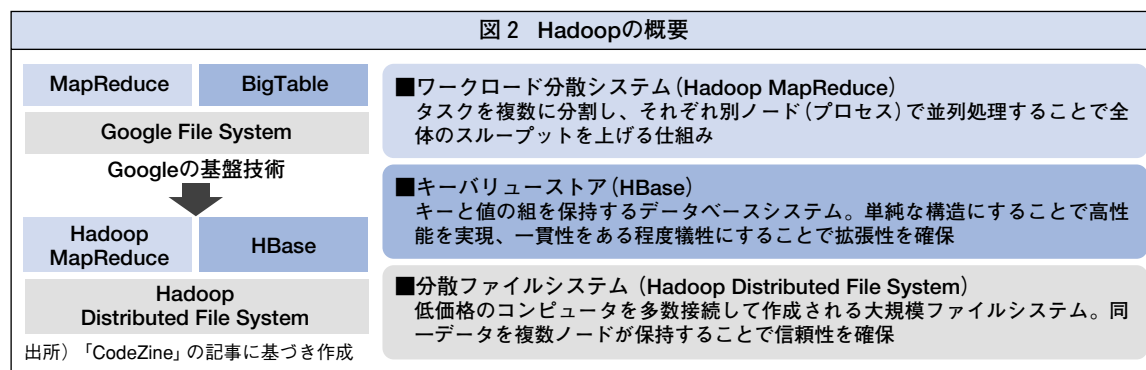




図2 Hadoopの概要



ている。NoSQLはRDBより高速な処理が可能で拡張性もある。これ以上の高速化を図る技術にはストリームデータ処理がある。CEP (Complex Event Processing: 複合イベント処理) によって発生したデータを保存せずにそのまま処理することで圧倒的な高速化を可能にする。

スループットを高めるための1つの方法は、処理を行うコンピュータ (ノード) の数を増やすスケールアウト構成として全体の処理能力を高めることである。従来のRDBやDWHもこの方法で処理能力を高めることはできるが、基本的に共有ディスク (複数のノードからアクセス可能なディスク領域) 構成を前提としているため、最終的には共有ディスクの処理限界がスループットの限界となる。これを改良したものがMPP (Massively Parallel Processing: 超並列処理) 型DWHである。共有ディスクが不要な構成を採用し、スループットの限界を取り払っている。

しかし、MPP型DWHでもノード数には上限がある。そこで、さらなるスループット向

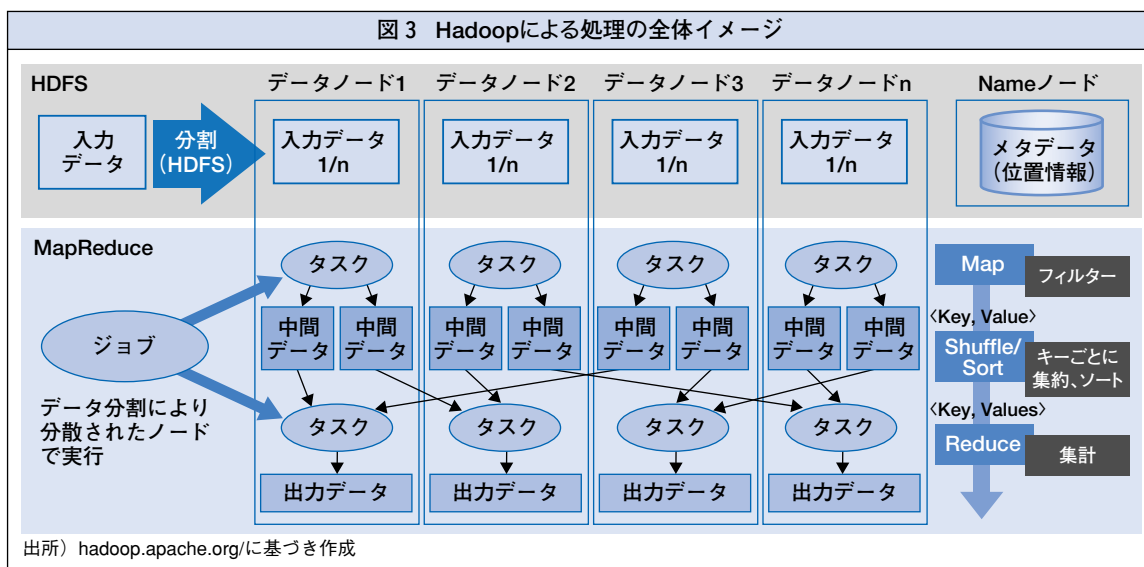
上のために使われるのが大量データの分散処理フレームワーク「Hadoop」である。非構造化データをHadoopで処理し、構造化データをMPP型DWHで処理するといった連携も行われるようになっている。

Hadoopの処理の仕組み

Hadoopは、ビッグデータを扱うためのさまざまな基盤技術の中核的な存在として、今、最も注目されるようになっている。Hadoopはもともと米国Google社が開発した大規模データ処理技術をJava (プログラミング言語の1つ) で実装したオープンソースソフトウェアである。

Hadoopは図2に示すようにMapReduce、HBase、Hadoop Distributed File System (HDFS) という3つの部分で構成され、それぞれワークロード分散、キーバリューストア、分散ファイルシステムの機能を提供している。

Hadoopをビッグデータを支える基盤として見た場合、最も根幹を成す部分はMapReduceとHDFSである。MapReduceがプロセスを分



散する仕組み（プロセスグリッド）、HDFSがデータを分散する仕組み（データグリッド）となっており、2つの仕組みが連動することでスケールアウト可能な分散処理機能が実現されている。

もう少し詳しく見ていこう（図3参照）。まず、実行したいジョブの入力データをHDFS上に用意する。データはHDFSにより複数のデータノードに分割配置され、その位置情報がメタデータとしてNameノードに記録される。次に入力データに対するジョブの実行を司るのがMapReduceである。MapReduceは、Nameノードに格納されたデータの位置情報を参照し、適切なデータノードにジョブをタスクとして実行するよう依頼する。タスクはMap、Shuffle/Sort、Reduceの順に実行され、最終的に処理結果を出力する。このようにデータの分割はHDFSで、プロセスの分割は

MapReduceで自動化されているため、スケールアウト可能な構成とすることが容易である。

耐障害性の観点で付け加えると、データはHDFSレベルではソフトウェア的に多重化されている。またジョブはJobTrackerで、タスクはTaskTrackerで監視されており、何か障害があれば別ノードで再実行する仕組みとなっており、一定の可用性が担保されている。

実証されたHadoopの有効性

野村総合研究所（以下、NRI）は、タクシーなどから得られる移動軌跡情報（プローブデータ）に基づいた独自の交通情報を生成し「全力案内！ナビ」で提供している。プローブデータは、交通規制の参考のために過去の渋滞情報を知りたいといった用途にも利用できる。このようなケースを想定して、Hadoopでは処理をどれだけ高速化できるかを検証した

(図4参照)。具体的には以下のような2ステップのMapReduce処理となる。

①対象日時のプローブデータをデータベースから抽出し、ある車両のIDをキーとしてリンクマッチ処理(車の位置と道路情報のマッチング)を行い、その車両の走行軌跡(リンク情報)を生成する。

②対象となる場所の2次メッシュ(1辺が約10kmの区画)コードをキーとして車両の走行軌跡(リンク情報)を重ね合わせて渋滞情報を生成する。

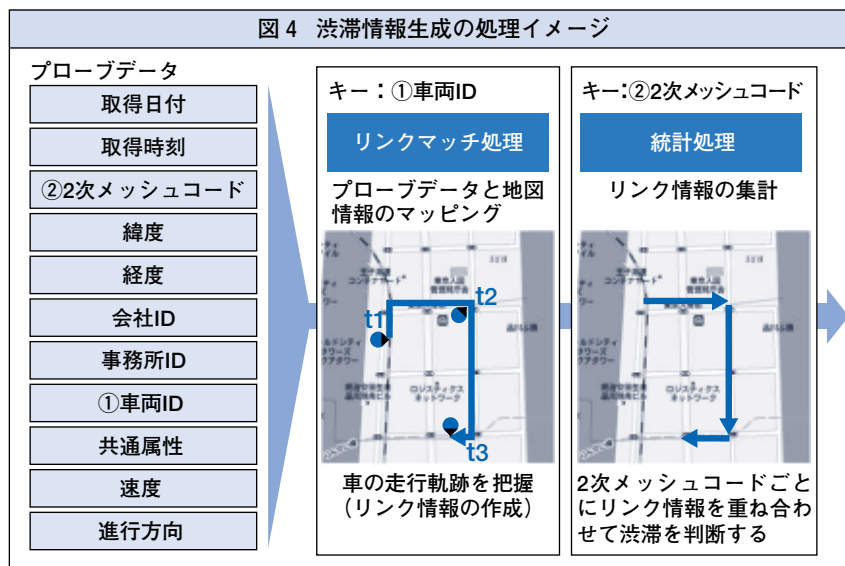
この検証では、RDBを使った場合に比較すれば10倍の高速化(車両が24台の場合)が実現された。キーの選択には工夫が必要だが、適切に設定すればHadoopによる高速化が可能であることが検証できた。

ビッグデータ基盤として進化するHadoop

現在、ますます増大するビッグデータの扱いを強化するために「Hadoop 2.0」の開発が進められており、以下のような機能向上や弱点の補強が図られることになっている。

①現在は4,000台程度が最大のHadoopクラスターが1万台以上に拡張される。

図4 渋滞情報生成の処理イメージ



②複数のNameノードを持てるようになり、さらに大量のHDFSファイルを一元管理することが可能になる。また、これまでSPOF(単一障害点。その故障によりシステム全体が動かなくなるような箇所)となっていたNameノードが複数になることで可用性が飛躍的に高まる。

③MPI、Master-Worker、繰り返しモデルなどMapReduce以外のプログラミングモデルをサポートする。

本稿ではHadoopを中心に解説してきたが、NoSQL、CEP、MPP型DWHを含め、NRIは技術検証を通じてビッグデータを支える基盤技術とその活用ノウハウを蓄積しており、システム構築やSaaS/ASPサービス(ソフトウェア機能をインターネットで提供する仕組み)などでの活用を予定している。 ■