

MapReduce 処理系 SSS の実装と評価

小川 宏 高[†] 中田 秀 基[†]
高野 了 成[†] 工藤 知 宏[†]

1. はじめに

MapReduce の代表的な実装である Hadoop は、非常に大規模なデータに対して、一組の Map と Reduce を実行することに特化した設計となっており、比較的小規模な MapReduce を多数繰り返すアプリケーションや、複雑なワークフローで実現されるアプリケーションには適していない。この問題を解決するべく、われわれはイタレーションが高速で柔軟なワークフローの構成が可能な MapReduce システム SSS¹⁾ を開発している。本稿では SSS の実装と評価について述べる。

2. SSS の構成

SSS は Hadoop の HDFS のようなファイルシステムを基盤とせず、分散 KVS を基盤とする点に特徴がある。入力データは予めキーとバリューの形で分散 KVS にアップロードしておき、出力結果も分散 KVS からダウンロードする形となる。

SSS ではデータをキーに対するハッシュで分散した上で、Owner Compute ルールにしたがって計算を行う。つまり、各ノード上の Mapper/Reducer は自ノード内のキーバリューペアのみを対象として処理を行う。このためスケジューリングのオーバーヘッドが少なく、イタレーションが高速である。

また、SSS では Map と Reduce の間でやりとりされるデータも KVS に蓄積されるため、Map と Reduce が 1 対 1 に対応している必要がなく、任意個数、段数の Map と Reduce から構成される柔軟なデータフロー構造を実行することができる。

3. SSS の評価

疎行列ベクトル積を用いて評価した。ベクトルのサイズが Map 時にメモリに乗る程度のサイズである場合に特化した手法として、row 方向に分割した行列を

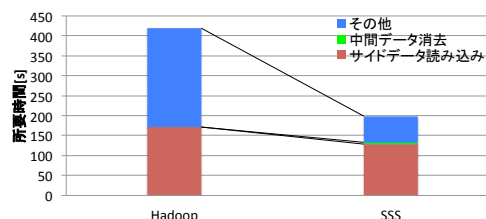


図 1 256Mi 次元の疎行列ベクトル積実行時間

Map の入力とし、ベクトルをサイドデータとする方法がある。この方法では Map で乗算と加算の双方を実行することができるため、Reduce フェイズでは演算を行う必要はない。

評価には、1 台のマスターノードと 16 台のワーカーノードからなる小規模クラスタを用いた。各ノードは 10Gbit Ethernet で接続され、各ワーカーノードは Fusion-io ioDrive Duo 320GB を備える。比較対象の Hadoop のバージョンは 0.20.2、レプリカ数は 1 としている。

256Mi 次元の行列を用い、中間データの消去を行いつつ繰り返し実行をした場合の、1 回のイタレーションの実行時間を図 1 に示す。Hadoop に対しておよそ 2 倍の速度向上を確認できた。これは、SSS では Reduce 段を省略できたことによるものと思われる。

謝 辞

本研究の一部は、独立行政法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務「グリーンネットワーク・システム技術研究開発プロジェクト (グリーン IT プロジェクト)」の成果を活用している。

参 考 文 献

- 1) Ogawa, H., Nakada, H., Takano, R. and Kudoh, T.: SSS: An Implementation of Key-value Store based MapReduce Framework, *First International Workshop on Theory and Practice of MapReduce (MAPRED'2010)*, pp. 754-761 (2010).

[†] 独立行政法人 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology