

## Spark における中間データ用ローカルストレージの構成方式の検討

谷村 勇輔, 小川 宏高

産業技術総合技術研究所 人工知能研究センター

### 1. 研究の背景と目的

Spark[1,2]は機械学習やグラフ処理など、反復計算や対話的処理を伴うビッグデータ解析の基盤として注目を集めている。Spark の特徴はメモリベースで処理を行い、処理環境の一部に障害が発生した場合に、失われた中間データを再処理により復旧させる方針をとっている点である。ただし、メモリが不足したり、タスク間の依存関係が連鎖して再処理に時間を要したりする場合には、ローカルストレージに中間データを保持することも可能である。

Spark の耐障害の仕組みは有用であるが、ローカルストレージの利用の有無が一部を除き、ユーザ側のプログラムに委ねられている。つまり、再処理のコストとローカルストレージを用いたデータ入出力のコストのトレードオフを考えて、ユーザ自身が適切にローカルストレージを利用しなければ、性能を損ねてしまう問題がある。また、ローカルストレージの構成に関しても、オフヒープ領域のメモリの利用、メモリとディスクの階層的な利用、高速なディスクの利用などが考えられ、それぞれの構成による性能の違いや費用対効果が明確になっていない問題もある。

本研究では、Spark の中間データの入出力に関して、ワーカノード上のローカルストレージの構成による性能の違いを調査するとともに、ワーカ上で想定される計算処理を踏まえ、ローカルストレージに求められる性能要件を明らかにする。これにより、Spark アプリケーションの性能チューニングや Spark システムの構築のための指針を得ることを目指す。そして、本ポスター発表では、いくつかのローカルストレージの構成において中間データ入出力性能、およびアプリケーションの実行時間を比較した結果を報告する。

### 2. ローカルストレージの構成方式とその評価方法

Spark において、中間データは RDD (Resilient Distributed Dataset) として扱われ、プログラム

内で `persist()` が呼び出されることで、ジョブ実行中の Persistence が確保される。その際の中間データの保存先はメモリ、ディスク、またはメモリとディスクの階層構成、オフヒープのメモリ構成を選択可能である。オフヒープのメモリを用いる場合は、Tachyon[3]を利用することになる。また、シリアライズの有無やレプリケーションに関するオプションも同時に設定可能である。

通常、中間データの入出力を高速にするためには、実行中の処理に制約や影響を与えない範囲で最大限にメモリを利用するアプローチがとられる。そして、メモリが不足する場合に、SSD 等の高速なディスクを用いて速度低下を抑制する方法がとられる。本研究では、これを踏まえてローカルストレージの各構成方式を比較することとし、ディスクを用いる方式では、NVMe-SSD や SATA-SSD を用いる。Tachyon を用いた構成では、バックエンド・ストレージへのデータ転送も含めた評価を行う。

評価には、疑似的に RDD を生成してその入出力を行うプログラムに加え、Spark の機械学習用のライブラリに含まれるプログラムを用いる。

**謝辞** 本研究の一部は、NEDO の委託業務「次世代ロボット中核技術開発プロジェクト」の支援を受けて実施した。

### 参考文献

- [1] Apache Spark, <http://spark.apache.org/>.
- [2] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S. and Stoica, I., "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing", Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (2012).
- [3] Li, H., Ghodsi, A., Zaharia, M., Shenker S. and Stoica, I., "Reliable, Memory Speed Storage for Cluster Computing Frameworks", Technical Report No. UCB/EECS-2014-135, 2014.
- [4] 谷村勇輔, 小川宏高, "Spark RDD のストレージ出力に関する性能評価", 情報処理学会研究報告, 2016-HPC-153(21), 2016.