

# 並列分散処理基盤 Apache Spark における系譜つき中間データに対する効果的なメモリ内キャッシュ指示の検討

米尾 謙史 †      置田 真生 †      萩原 兼一 †  
† 大阪大学 大学院情報科学研究科

## 1 はじめに

近年、ビッグデータ分析を目的とした並列分散処理基盤として Apache Spark [1] が用いられている。Spark では RDD (Resilient Distributed Dataset) [2] という抽象化された分散データセットを介して並列処理を行う。

RDD はデータの実体を持たず、中間処理の系譜情報のみを持つ。データは終端処理で実際に使用される時に中間処理の系譜を辿り計算される。計算されたデータは終端処理終了時にすぐに破棄されるため、データを再利用するためにはプログラマがソースコード中にキャッシュ指示を記述する必要がある。しかしキャッシュの効果は実行環境や入力データにより変化するため、どのアプリケーションでも最適となるキャッシュ指示は難しい。

本研究では効果的なキャッシュ指示方針の発見を目的に、公開ライブラリを改変し実行時間を比較し、既存のライブラリにおけるキャッシュ指示の適切さを確認する。

## 2 RDD

RDD に対する操作は次の3つである。RDD から系譜を派生させる変換および、RDD の系譜を辿って結果値を返すアクション、RDD に対して計算された実体データをメモリに格納する指示を与えるキャッシュ指示である。アクションが実行された際に変換の系譜が辿られ、まとめて計算される。

キャッシュ指示のない RDD を再計算するペナルティは再計算に必要な変換のコストの総和で決定する。そのためキャッシュ指示を与える RDD を選ぶ一般的な基準として系譜中の変換のコストおよび使用回数、系譜の長さ注目する。系譜の例として Spark の機械学習ライブラリ MLlib [3] に含まれる K 平均法の RDD の系譜を図 1 に挙げる。

## 3 実験

本研究の目的は公開ライブラリにおけるキャッシュ指示の適切さの確認および効果的なキャッシュ指示方針の分析である。1つ目の実験では、公開ライブラリに対しキャッシュ指示パターンを変えた改造ライブラリを用意し、メモリ上限と入力データサイズを変えながら実行時間を比較した。2つ目の実験では、キャッシュ指示を与えることで実行時間削減に効果のある RDD を予測し、その中からキャッシュ指示を与える RDD の組み合わせを変えて実行時間を比較する。実験環境として、16 台のノードから成るクラスター、各ノードの主記憶は 24GB、公開ライブラリとして MLlib を用いた。

1つ目の実験では、入力データに対しメモリ上限が十分に大きい場合、単純に2回以上使用する RDD 全てに対しキャッシュ指示を行ったパターンでも実行時間を最大で 46% 削減できた。すなわち最適なキャッシュ指示をしていない公開ライブラリを確認できた。

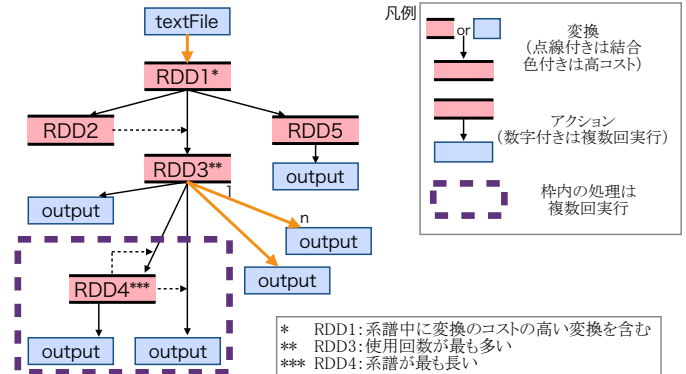


図 1 K 平均法における RDD の系譜

表 1 K 平均法において各 RDD に対しキャッシュ指示を行った場合の実行時間 (5 回の平均値, オリジナルとは公開ライブラリの意味, 単位:秒)

キャッシュ指示を行った RDD	実行時間
RDD1, RDD3, RDD4	78.8
RDD1, RDD3	80.6
RDD1, RDD4	80.7
RDD1	83.8
RDD1, RDD2, RDD4 (オリジナル)	118.5
RDD3, RDD4	152.4
RDD3	160.0

2つ目の実験では公開ライブラリのうち K 平均法の結果を示す。K 平均法において組み合わせを変えてキャッシュ指示を行った場合の実行時間を表 1 に示す。RDD1 を含む組み合わせにおいて実行時間を削減できた。本実験では、他の MLlib ライブラリに含まれる線形回帰、協調フィルタリングでも同様にコストの高い変換のみに着目してキャッシュ指示を与えることで実行時間を効率的に削減できた。

謝辞 本研究の一部は科学研究費補助金 (15H01687, 26730035) の支援による。

## 参考文献

- [1] Apache Spark. <http://spark.apache.org>.
- [2] M. Zaharia, M. Chowdhury, and T. Das et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the NSDI 2012*, San Jose, CA, US, April 2012. 14 pages.
- [3] Machine Learning Library. <https://github.com/apache/spark/tree/master/mllib/src/main/scala/org/apache/spark>.