Caching Opportunities in Spark Frameworks

Ahmed M. Abdelmoniem and Byron Yi
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
{amas, byi}@cse.ust.hk

Abstract

Spark Framework and its successors have recently demonstrated its position as a new fault-tolerant eco-system for large-scale batch data analysis. They have shown to be highly scalable, support coarse-grained fault tolerance and provide easy to learn Application Programming Interface (API). Most applications are built to execute different jobs on these frameworks where they often share similar work (for instance, several jobs may use the same input data and/or produce the same output data which is used by next job). Hence, we can spot many opportunities to optimise the execution plan performances for majority of batch jobs. In this report, we plan to explore possible caching techniques in the literature for multi-job optimisation specifically for spark framework. We plan to go further and propose a simple yet efficient caching techniques and policies. Our contribution in this project would be surveying the current and recent literature and proposals of caching and optimisation algorithms that given an input batch of jobs, produces an optimal plan while identifying caching opportunities. Our other contribution is proposing a straightforward caching algorithm that would improve batch job's performance. If possible, we will report our experimental results on Spark deployment to demonstrate that our technique would improve the average completion time of Spark jobs.

1 Introduction

1.1 Subsection