

MapNoReduce Platform

João Pinho
jpepinho@gmail.com

Diogo Rosa
diogomcrosa@hotmail.com

Cláudia Filipe
claudiapbfilipe@gmail.com

Instituto Superior Técnico
Middleware for Distributed Internet Applications
Lisbon, Portugal

Abstract

This project consists in the design and implementation of MapNoReduce Platform, a simplified implementation of the MapReduce middleware and programming model. This platform extracts the input key/value pairs from input files and distributes the Map calls, called Jobs, across multiple machines, the Workers. Also, the platform ensures a good performance by monitoring job's progress, detecting faulty or slow machines and rescheduling their tasks on idle machines. That is assured by Job Trackers, which are distributed in this platform, in contrast to the original implementation of MapReduce, where they are centralized. This mechanism was implemented inspired in Facebook's solution—Corona [1]. Additionally, in order to test the platform it was developed a PuppetMaster component which allows to control the platform, and also to induce some delays and faults to the system in order to perform some tests and evaluation.

lines. The Map invocations, called Jobs, are distributed across multiple machines by automatically partitioning the input data into a set of splits of size S . The input splits can be processed in parallel by those machines, named Workers. The system ensures that for each job submitted, all the input data is processed with a good performance through the monitoring of jobs' progress, fault or slow machines detection and reschedule of idle machine's tasks. In the original MapReduce implementation these tasks are performed by the JobTracker which is a centralized component. If the JobTracker fails the system can't receive new jobs nor processing pending ones, which can be critical in systems that need high availability. Considering JobTracker as a single point of failure [3], it is necessary to replicate this component. As it would add complexity to the system and overhead, this project also introduces a new entity, the CoordinationManager, separating cluster resource management from job coordination, which allows the system to focus on make faster scheduling tasks.

1 Introduction

MapReduce was introduced by Google in 2004 [2] and is currently one of the most popular approaches for large scale data analytics - also thanks to the availability of high quality open-source implementations. When using the MapReduce paradigm, the computation takes a set of input key/value pairs, and produces a set of output key/value pairs. MapReduce users express the computation as two functions: Map and Reduce. This project focuses only the Map part, which uses a Map function given by the user and an input set of key/value pairs to produce a set of key/value pairs. In MapNoReduce Platform the keys are the numbers of the line of the file being read and the values are the content of those

2 MapNoReduce Architecture

2.1 Worker

2.2 JobTracker

2.3 CoordinationManager

2.4 DefaultJobScheduler

2.5 TaskTracker

2.6 TaskRunner

2.7 SlaveReplica

2.8 Distribution manner

2.9 Puppet Master

2.9.1 Status

2.9.2 Create worker

2.9.3 Wait

2.9.4 Slow worker

2.9.5 Freeze/unfreeze worker

2.9.6 Freeze/unfreeze communication

3 Evaluation

4 Conclusions

References

- [1] A. Ching, R. Murthy, D. Molkov, R. Vadali, and P. Yang. Under the hood: Scheduling mapreduce jobs more efficiently with corona. *Facebook Engineering*, November 2012.
- [2] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications ACM*, 51(1):107–113, January 2008.
- [3] V. Kalavri and V. Vlassov. MapReduce: Limitations, optimizations and open issues. *Proceedings - 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013*, pages 1031–1038, 2013.