PADIMapNoReduce 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 PADIMapNoReduce 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, henceforth called Jobs, across multiple machines, the Workers. Also, the platform ensures a good performance by monitoring jobs' progress, detecting faulty or slow machines and rescheduling their tasks on idle machines. That is assured by JobTrackers, which are distributed in this platform, in contrast to the original implementation of MapReduce, where they are centralised. This mechanism was implemented inspired by Facebook's solution Corona [2]. Additionally, in order to test the platform it was developed a PuppetMaster component which allows to control the platform, and also allows to induce some delays and faults to the system.

1. Introduction

MapReduce was introduced by Google in 2004 [1] 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 of MapReduce, which uses a Map function given by the user and takes an input set of key/value pairs and produces a set of key/value pairs. In PADIMapNoReduce Platform the keys are the numbers of the line of the file being read and the values are the content of those lines.

The Map invocations 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 job's progress, fault or slow machines detection and reschedule of idle machines' tasks. In the original MapReduce implementation these tasks are performed by the JobTracker which is a centralised 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.

2. Related work

3. Puppet Master

3.1. Status

3.2. Create worker

3.3. Wait

3.4. Slow worker

3.5. Freeze/unfreeze worker

3.6. Freeze/unfreeze communication

4. Worker

5. JobTracker

6. CoordinationManager

7. DefaultJobScheduler

8. TaskTracker

9. TaskRunner

10. SlaveReplica

11. Distribution manner

12. Stuff and stuffy

Figure 1. Example of caption.

Figure 2. Example of long caption requiring more than one line. It is not typed centered but aligned on both sides and indented with an additional margin on both sides of 1 pica.

13. Conclusions

References

[1] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. 2008.

[2] TODO. Some related article I wrote. *Some Fine Journal*, 99(7):1–100, January 1999.