UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

WAGNER KOLBERG
JULIO C. S. ANJOS

MRSG: A MapReduce Simulator for Desktop Grids

Research Report
RP-364

Supporting Agencies: CNPq

Prof. Dr. Cláudio F. R. Geyer
Advisor

Prof. Dra. Luciana B. Arantes
Coadvisor

Porto Alegre, April 2011

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

DFS	Distributed File System
GFS	Google File System
HDFS	Hadoop Distributed File System
MRSG	MapReduce over SimGrid

LIST OF FIGURES

ABSTRACT

This technical report presents the MRSG simulator for MapReduce platforms. The simulator's goal is to serve as a research tool, and assist in the development of new approaches and solutions to MapReduce platforms. It is described, as well, the decisions made in the simulator's model, MRSG's architecture, and the current development state. An initial validation shows the current precision of the already implemented features, in comparison to real executions of the Hadoop MapReduce platform.

Keywords: Desktop grid, MapReduce, MRSG, simulation.

1 INTRODUCTION

MapReduce (DEAN; GHEMAWAT, 2008) is a parallel programming model, for which there is an associated platform, in order to abstract the implementation details of distributed applications. In other words, the users of the platform shall be concerned only with the development of their solution, which follows the MapReduce model, and not with problems arising from the distribution of the application on a cluster or grid. The MapReduce platform is responsible for controlling the communication between nodes, tasks and data distribution, among other common activities of distributed environments.

The MapReduce model consists of two steps, one for mapping and another for reduction of data. In the map phase, the user function generates a set of key/value pairs, according to some processing done on the input data. All values associated with the same key, are then processed by the reduction function (DEAN; GHEMAWAT, 2008). Figure 1.1 illustrates this behavior.

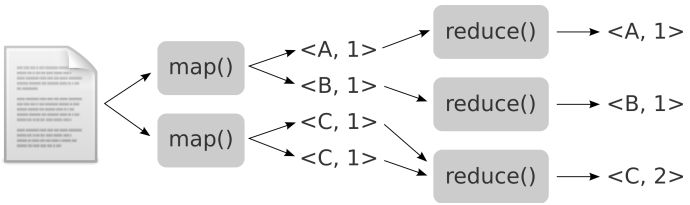

Figure 1.1: MapReduce Data Flow

Since the creation of tasks, data partitioning, and all other aspects of the system's distribution are controlled by the platform, it is very important for its implementation to be efficient. It must present an appropriate behavior for a variety of environments, and also applications. Thus, decisions taken in the design of a MapReduce framework have a major impact on its implementation.

The search for a more efficient and flexible system leads researchers to study new algorithms for job and task scheduling, data distribution, and other improvements in MapReduce features that enable the advancement of this technology. As an axample, a group of researchers studied the impact of *speculative tasks*[1] on the Hadoop MapReduce platform, and developed a new task scheduler algorithm (ZAHARIA et al., 2008).

However, testing these new solutions may have high costs, and requires great effort. When testing new approaches, the researcher must consider issues of scalability, for example, which requires a very large and complex infrastructure. To evaluate a new task scheduler, is necessary to observe its behavior when the tasks are long, short, or when

[1] Speculative tasks are re-executions of slow tasks at the end of a job, which prevent that a few slow tasks delay the entire job.

they have varying sizes. Many other examples could be presented, and like the ones that were exposed, require time, effort, and have high costs.

To facilitate these processes of evaluation and testing, the MRSG simulator (acronym for MapReduce over SimGrid) provides a simplified and high-level model, which emulates the behavior of a MapReduce platform. Through it, a theoretical algorithm can be quickly translated into executable code and analyzed in various simulated environments, and with a variety of different parameters. MRSG also helps in creating new solutions, because it allows researchers to perform simulated deployments of their ideas, and identify possible design errors earlier, before an actual implementation.

This paper is organized as follows. Section 2 introduces the architecture of the MapReduce framework. Section 3 presents existing works on simulating MapReduce environments. Section 4 describes the characteristics and architecture of MRSG. Section 5 presents an initial evaluation of the simulator. In section 6 we make some final considerations.

2 THE MAPREDUCE FRAMEWORK

In this section we briefly introduce the MapReduce framework architecture, and the technologies involved with it. This knowledge is necessary to understand the mechanisms that must be simulated by MRSG.

The implementation of the MapReduce framework, as presented in (DEAN; GHEMAWAT, 2008), consists of a master node and several nodes workers. The master node is responsible for distributing map and reduce tasks to workers, which process the tasks by applying the map and reduce functions programmed by the user.

It is important to note that MapReduce uses a distributed file system (DFS) for storing the input data of the applications. Thus, the data used in the processing of tasks is already distributed among the nodes that comprise the cluster. The MapReduce task scheduler uses this locality information to prevent data transfer over the network (DEAN; GHEMAWAT, 2008). In the Google framework, the DFS used is the Google File System (GFS) (GHEMAWAT; GOBIOFF; LEUNG, 2003).

In a simplified form, the execution of a MapReduce job can be summarized as follows. In the mapping phase, the master tries to schedule tasks according to the location of the DFS data blocks, i.e., when the master finds that a worker is idle, it searches for a task that processes a block of data stored on the worker's local disk. Mapping results are stored locally by workers. Later, in the reduction phase, workers must get the intermediate results directly from the nodes that computed the map tasks.

Some optimization mechanisms are also presented in (DEAN; GHEMAWAT, 2008). As an example we can mention the backup tasks, which are re-executions of slow tasks in the final stages of mapping and reduction. This mechanism prevents that a few slow tasks delay the entire MapReduce job.

Several implementations of this model can be found. Hadoop MapReduce is an open source Java implementation of the framework presented by Google, and maintained by the Apache Software Foundation (HADOOP, 2010). Hadoop is one of the most known and used MapReduce implementations, adopted by major corporations such as Yahoo, Facebook, Amazon and IBM. The project also has an alternative to GFS: the Hadoop Distributed File System (HDFS) (SHVACHKO et al., 2010).

The behavior and mechanisms that were described are examples of emulations to be produced by MRSG to obtain a behavior similar to actual MapReduce platforms.

3 RELATED WORKS

While there is a reasonable amount of general-purpose simulators for distributed environments, few specific simulators for MapReduce can be found. The most recent solutions are described below.

MRPerf (WANG et al., 2009) is a simulator for Hadoop MapReduce, that uses ns-2 as the basis for its network simulation. MRPerf, however, does not simulate important features of the Hadoop platform, which have major impact on system performance, especially when the simulated environment is heterogeneous. As an example we can mention the block replication of the distributed file system, and the speculative execution mechanism. The proposed MRSG simulator implements these features and therefore supports heterogeneous environments and, in future releases, shall support volatile environments as well.

Cardona et al. present a simulator to the MapReduce system for grids based on GridSim and SimJava simulators. It focus on simulating map and reduce functions as well as the distributed file system in order to test different scheduling algorithms. However the simulator simplifies the MapReduce environment, and do not consider many built-in features such as data replication, speculative execution, and data split (CARDONA et al., 2007).

The Mumak MapReduce Simulator (TANG, 2009) is a discrete event simulator that emulates the conditions under which a Hadoop MapReduce scheduler performs on clusters, running a specific workload. It simulates only homogeneous environments and it does not perform I/O or computation simulations. Also, the JobTracker can not detect network operations. These functions are important to identify slow nodes and simulate the speculative execution mechanism.

More recently, the work of (HAMMOUD et al., 2010) proposes the MRSim simulator, which simulates applications on the MapReduce model. That is, from an application, such as matrix multiplication for example, the simulator estimates the execution time and analyzes the behavior of the platform when running this application. However, no interface is provided to modify platform algorithms like task scheduling and data distribution. In contrast, the MRSG simulator is concerned with the analysis of changes in the MapReduce platform itself, and not only in the applications running on top of the MapReduce platform.

4 THE MRSG SIMULATOR

MRSG aims to facilitate research on the behavior of MapReduce platforms, and possible changes in the technology. To that end, the simulator seeks to provide: a simplified way to translate theoretical algorithms (task scheduling, data distribution, etc.) to executable code, facilitating the analysis of these algorithms; ease of change and test of system configurations; and the possibility to validate platform algorithms on a large scale, since no real infrastructure is required.

It is important to note also that the MRSG is proposed to simulate heterogeneous environments, where the machines are highly variable in terms of processing capacity. The simulation of volatile environments is also a future goal of the tool.

MRSG uses the SimGrid simulator (CASANOVA; LEGRAND; QUINSON, 2008) as a basis for its implementation. SimGrid is a set of tools that provides functionalities for the simulation of general-purpose distributed applications. Its goal is exactly to facilitate research in parallel and distributed platforms. Several researchers, linked to a variety of universities around the world use or have used SimGrid in scientific publications, enhancing the system's reliability (SIMGRID, 2010a).

SimGrid provides several APIs to simulate distributed environments, as shown in Figure 4.1. MRSG uses the MSG interface for the C language, wich provides functions very similar to real code implementation, but with a simplified simulation model (SIMGRID, 2010b).

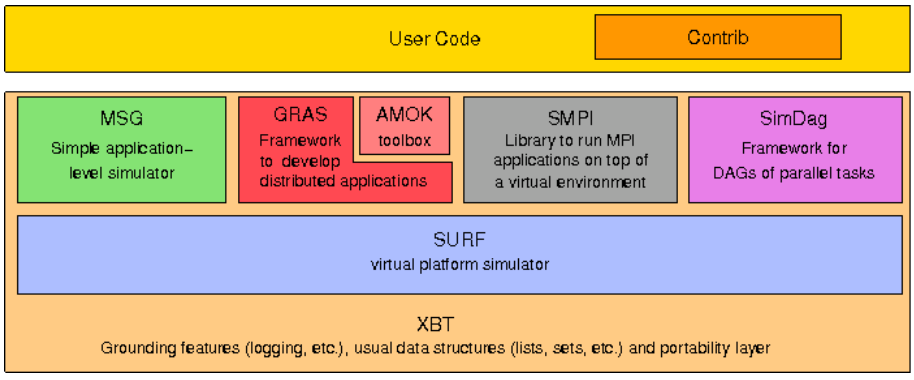

Figure 4.1: SimGrid Modules (SIMGRID, 2010b)

As illustrated in figure 4.2, SimGrid is responsible for the simulation of network communications and task processing. Therefore, MRSG simulates only the behavior of the MapReduce platform, through calls to SimGrid functions, without modifying its source code. In addition, an interface is provided to MRSG's end user, allowing the description

of the environment (platform topology, computational power, etc.), MapReduce platform settings and the development of new algorithms.

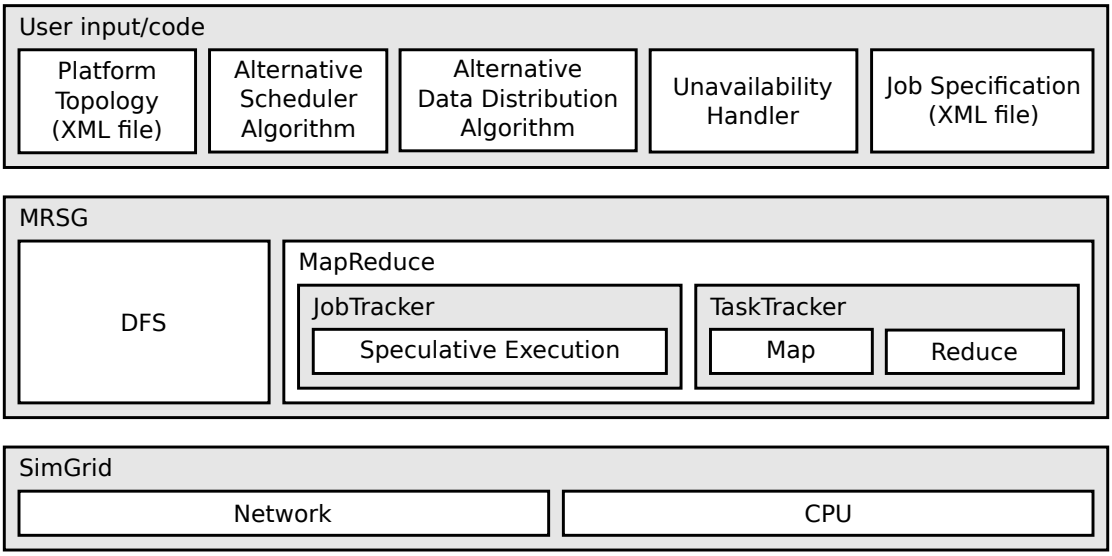

Figure 4.2: MRSG Architecture

In its final release, the MRSG simulator should present the following characteristics concerning the MapReduce features, as shown in Figure 4.2, and environment scenarios:

- Distributed File System:
  - The distributed file system, including all data requests and block localization, is controlled by the simulator;
  - The DFS supports block replication;
  - Users can code new data distribution algorithms, to place data blocks and replicas on specific nodes;
  - The user can describe rack configuration schemas;
- MapReduce:
  - All computation and data transfer required by map and reduce tasks are handled by the simulator, through the SimGrid API;
  - Users can code a new scheduler for map and reduce tasks;
  - The simulator supports the speculative execution mechanism;
- Environment:
  - The platform topology and the MapReduce job configuration is done through SimGrid's platform and deployment XML files;
  - The simulator supports heterogeneous and volatile environments;
  - Users can code a handler system to deal with nodes unavailability in volatile environments, such as Desktop Grids.

In the current development state, MRSG implements: the distributed file system with block replication; the basic flow for the implementation of MapReduce, which includes the execution of the Map and Reduce tasks; all data transfer involved in the process, including the DFS requests and intermediate pairs gathering; and the mechanism for speculative execution. On the other hand, were not yet modeled the rack configuration feature and the ability to handle volatile environments.

5 PRELIMINARY RESULTS

Using the features already developed, the simulator was compared with actual implementations on Hadoop MapReduce (HADOOP, 2010), a Java implementation of the MapReduce platform used by several companies like Yahoo, Facebook and Amazon (HADOOP, 2010). To this end, we used a log-processing application, in two different scenarios: one with 20 cores and 20 tasks for the reduction phase, and another with 200 cores and only one reduce task, causing an overload on the reducer node.

The results are shown in figure 5.1, which presents the comparison between the times of the map, copy, and reduce phases of MapReduce, between the real and simulated execution.

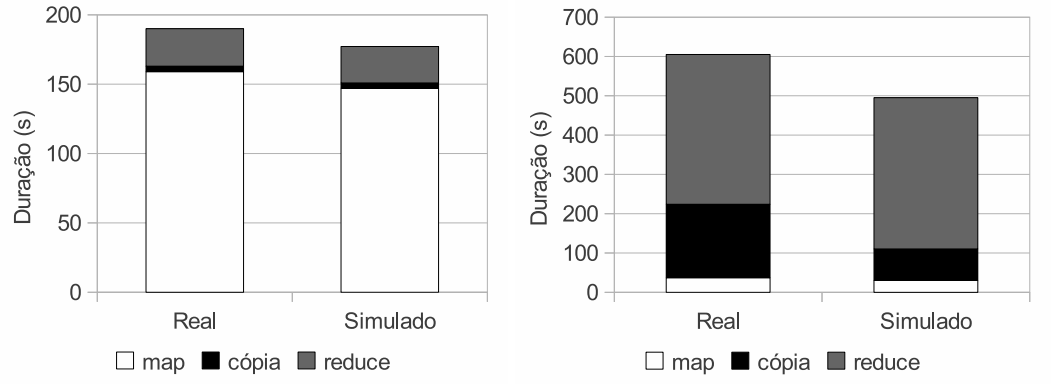

Figure 5.1: MRSG Preliminary Evaluation

It is possible to see that there is a considerable proximity between simulation and actual execution, except in the copy phase of the second test. This difference in the copy phase was due to a discrepancy between the simulator model and the Hadoop environment, and is currently being adjusted. However, one can observe that despite this difference, the behavior of the simulator was very close to the actual execution of the application on the real grid.

The amount of work on local data and speculative tasks, although not shown in charts, were equivalent in the real and simulated executions. In future work, a more comprehensive evaluation will be conducted in order to obtain a more meaningful and consistent validation of MRSG's behavior.

6 FINAL CONSIDERATIONS

This paper presented the MRSG, a simulator for MapReduce platforms, describing its objectives and architecture. Although yet in development, it was possible to perform initial evaluations of the simulator, which demonstrated a considerable proximity between simulations and real implementations of the MapReduce platform. However, it is possible to see that improvements can be made in the simulator, and new features should be added.

We are currently working on these improvements to correct the copy time discrepancy and to make the MapReduce behavior as close as possible to the real system execution, based on extended tests performed with the Hadoop MapReduce on the Grid'5000. In future works, we intend to present a thorough validation of the simulator with some new features.

REFERENCES

CARDONA, K. et al. **A Grid Based System for Data Mining Using MapReduce**. [S.l.]: The AMALTHEA REU Program, 2007. Acessado em Março 2011. (Technical Report TR-2007-02).

CASANOVA, H.; LEGRAND, A.; QUINSON, M. SimGrid: a generic framework for large-scale distributed experiments. In: IEEE INTL. CONF. ON COMPUTER MODELING AND SIMULATION, 10. **Proceedings...** [S.l.: s.n.], 2008.

DEAN, J.; GHEMAWAT, S. MapReduce: simplified data processing on large clusters. **Commun. ACM**, New York, NY, USA, v.51, n.1, p.107–113, 2008.

GHEMAWAT, S.; GOBIOFF, H.; LEUNG, S.-T. The Google file system. In: SOSP '03: PROCEEDINGS OF THE NINETEENTH ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, New York, NY, USA. **Anais...** ACM, 2003. p.29–43.

HADOOP. **Welcome to Apache Hadoop!** Disponível em: <http://hadoop.apache.org/>. Acesso em: abril 2010.

HAMMOUD, S. et al. MRSim: a discrete event based mapreduce simulator. In: FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY (FSKD), 2010 SEVENTH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. v.6, p.2993–2997.

SHVACHKO, K. et al. **The hadoop distributed file system**. 2010. 1–10p.

SIMGRID. **SimGrid**: people around simgrid. Disponível em: <http://simgrid.gforge.inria.fr/doc/people.html>. Acesso em: agosto 2010.

SIMGRID. **SimGrid Home Page**. Disponível em: <http://simgrid.gforge.inria.fr/>. Acesso em: agosto 2010.

TANG, H. **Mumak**: map-reduce simulator. [S.l.]: Apache Software Foundation, 2009. [ONLINE], Disponível em https://issues.apache.org/jira/browse/MAPREDUCE-728 acessado em março 2011. (Technical Report MAPREDUCE-728).

WANG, G. et al. A simulation approach to evaluating design decisions in MapReduce setups. In: **Anais...** [S.l.: s.n.], 2009. p.1–11.

ZAHARIA, M. et al. **Improving MapReduce Performance in Heterogeneous Environments**. [S.l.: s.n.], 2008. (UCB/EECS-2008-99).