Hadoop MapReduce Configuration Parameters and System Performance: a Systematic Review

Ailton S Bonifacio¹, Andre Menolli², and Fabiano Silva¹
¹Department of Informatics, Federal University of Paraná, Curitiba, Paraná, Brazil
²Computer Science Department, Universidade Estadual do Norte do Paraná, Bandeirantes, Paraná, Brazil

Abstract—Hadoop MapReduce assists companies and researchers to deal with processing large volumes of data. Hadoop has a lot of configuration parameters that must be tuned in order to obtain a better application performance. However, the best tuning of the parameters is not easily obtained by inexperienced users. Furthermore, most of studies that address this issue are superficial and cover only some parameters. This paper presents a systematic review by identifying current research papers, which addresses the correlation between Hadoop configuration settings and performance. In this sense, this systematic review identified 743 papers in 5 researched databases. Applying the steps for selecting and a few exclusion criteria, the number of papers was reduced to 40. These papers were analyzed and classified according to Hadoop configuration parameters and the goals of the studies.

Keywords: Configuration Parameters, Hadoop, MapReduce, Performance, Systematic Review

1. Introduction

MapReduce, a framework introduced by Google Inc., is a programming model designed to process large data volumes in parallel in a distributed environment (clusters) [1].

Hadoop [2] is an open-source implementation of MapReduce and currently have been one of the most widely used implementations for processing large amounts of data. Its programming model is divided into two main phases: the Map and Reduce phases. The reduce phase is divided between phases shuffle, sort and reduce. This is the intermediate phase.

A large number of companies, research centers and researchers have used Hadoop MapReduce implementations in applications such as data mining, production of reports, indexing Web pages, analysis of log files, machine learning, financial analysis, scientific simulation, statistics, research in bioinformatics and others [3]. Therefore, Hadoop has been studied to identify several aspects involving the tuning and performance.

Current studies show that the performance of Hadoop MapReduce jobs depends on the cluster configuration, input data type and job configuration settings [1], [3], [4]. Furthermore, some studies point out that most of the applications running on Hadoop, show a large difference between the behavior of the map and reduce phases. For instance, in some cases the Map phase is computationally intensive, in other is Reduce and others both [4].

This paper presents an overview of these studies focused on Hadoop configuration parameters that has influence on the job performance. The paper explores how the tuning parameter of Hadoop impacts on the entire system performance. More specifically, this research might answer the following questions:

(1) Which Hadoop configuration parameters has influences and impact on system performance?
(2) Which parameters are influenced by Hadoop phases?
(3) Which parameters are influenced by workloads characteristics?

This paper is organized as follows. In Section 2, a background on MapReduce and Hadoop framework, concepts and theories are presented. Section 3 describes the research method applied to systematic review. Results are presented in Section 4 and discussed in Section 5. Section 6 presents the final considerations this paper.

2. Background

This section presents the main concepts related to this systematic review.

2.1 MapReduce

MapReduce is a framework of distributed functional programming. Its processing occurs in two phases: Map and Reduce. MapReduce framework divides the work into a set of independent tasks and manage communications and data transfers between nodes in the cluster and the related parts of the system.

Apache Hadoop [2] is the well known and most widely used implementation of MapReduce. However, the improvement of application performance is directly related to the setting of the Hadoop’s parameters. To ascertain the relationship between parameter values and a good performance is not a simple task, even for experienced users.

2.2 Hadoop

The Apache Hadoop software library allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed
to scale up from single servers to thousands of machines, each offering local computation and storage.

Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation.

2.2.1 Execution Parameters

In order to run a program such as a job in Hadoop, an object configuration job is created and the parameters of the job must be specified. That is, since the job was created for MapReduce, it must enabled to run in scale on a cluster [4]. Therefore, in order to avoid some problems such as under-performance, for example, it is necessary to tuning some parameters.

Hadoop provides many configurable parameters that lead to better or worse performance on a cluster. Most of these parameters take effect on execution of the job and cluster. In the case of large clusters of machines, these parameters become quite important and provide an indication of the performance of this cluster in relation to the submitted job. Thus, for better system performance, the framework must be further optimized as is possible.

Hadoop has over 190 parameters that can be specified to control the behavior of a MapReduce job. Of those, more than 25 parameters can impact the performance of the task [5]. Some configuration parameters aim to control various aspects of the behavior of the task at runtime. Some of these aspects include the allocation and use of memory, competition, optimize I/O and network bandwidth usage. If the parameters are not specified, default values are assumed.

3. Research Method

One method that has gained significant popularity in the various area of research is the Systematic Literature Review (SLR or systematic review). Specifically in Big Data, few systematic reviews are found. Therefore, it is a field to be explored. More specifically, we found no systematic review about performance of Hadoop systems related to their configuration parameters and tuning. In this sense, a review on this subject in a systematic and rigorous manner, will be valuable for research and practice.

SLR is a rigorous methodological review of the research results. Besides adding the existing evidence on a specific research topic, an SLR is also intended to support the development of guidelines and consensus for future studies, based on evidence from several previous studies. Therefore, a systematic review aims to identify, evaluate and interpret relevant research about a particular issue, a thematic area or phenomena of interests using an explicit and rigorous method [6].

Among the several grounds for make a systematic review[6], we highlight for this work:

- Identify the directions that current research on the subject in a matter are taking;
- Provide a background to the new researchers in the field in order to properly position them regarding what has been done and the consensus of the community.

A systematic review involves several distinct activities, which according to [7], summarizes the stages of the systematic review in three main phases: planning, conducting and reporting the review. The next subsections detail the steps taken to produce this study.

3.1 Planning the review

The planning of the systematic review started by developing a protocol. This protocol specified the process and methods that would be applied. The protocol specified the research questions, the research strategy and the criteria for inclusion and exclusion. This systematic review aims to provide an overview of studies which apply to performance of Hadoop systems in regard to their configuration parameters, answering the research questions mentioned in Section 1.

3.2 Conducting the review

Once the protocol has been agreed, the steps for conducting the review are presented below.

3.2.1 Research Identification

A systematic review aims to find the largest possible number of primary studies related to the research topic. Thus, the first step is the identification of keywords and search terms. In this study, the keywords and terms have been used in order to get the relevant work in a broader way, in relation to the universe of this study [8]. All possible permutations of the words "Hadoop" and "parameters" were considered. Moreover, we varied the terms relating to "performance" (see Table 1). For example, the first search string was: "Hadoop", "Performance Tuning" and "Parameter". All search terms were searched in full-text, title, and keywords of all databases listed in Table 2. These databases were chosen because of their relevance in the scientific community [6], and for the indexing of the most important conferences of parallel and distributed computing, workshops and journals. As the objective of this study is to understand the performance of Hadoop framework by setting the its parameters, and this is a relatively new research topic, only studies of 2008 onwards have been searched. The research was carried out in October 2013, which means that the publications of the end of 2013 until now may not have been indexed by the databases. The identification process produced 1181 papers. This formed the basis for the next step.

3.2.2 Selection of Primary Papers

The first step after the preliminary identification of papers was the elimination of duplicate titles. The execution of this
Table 1: Terms for the search

<table>
<thead>
<tr>
<th>Terms about Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Tuning</td>
</tr>
<tr>
<td>Performance Analysis</td>
</tr>
<tr>
<td>Performance Prediction</td>
</tr>
<tr>
<td>Performance of MapReduce</td>
</tr>
<tr>
<td>Hadoop Performance</td>
</tr>
<tr>
<td>Job Performance</td>
</tr>
<tr>
<td>Self-Tuning</td>
</tr>
</tbody>
</table>

Table 2: Databases researched in the systematic review

<table>
<thead>
<tr>
<th>Researched Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM Digital Library</td>
</tr>
<tr>
<td>IEEE Xplore Digital Library</td>
</tr>
<tr>
<td>Science Direct</td>
</tr>
<tr>
<td>Springer Link</td>
</tr>
<tr>
<td>Scopus (Elsevier)</td>
</tr>
</tbody>
</table>

step reduced the set for 743 papers. The next step was to remove the titles clearly not related to the review. This step reduced the set for 296 papers. Then the abstracts were read and evaluated, with the following exclusion criteria:

- exclude papers that were clearly not of the hadoop system area or not applied to hadoop performance or job performance,
- exclude papers that were not related to the terms of performance tuning, analysis, prediction, model or self-tuning,
- exclude papers that were not related to the tuning parameters or configuration as well as their implications on performance,
- exclude papers that were just of literature review, as example, systematic review, and,
- exclude papers without validation method, if it were not clearly a practical study.

After the execution of this process, the set of papers was reduced to 124. The full text for all 124 papers was obtained and read with the same exclusion criteria. The final number of papers selected for the review was 40, which clearly fitted to the criteria defined for accomplishment of the systematic research, which were then analyzed.

3.2.3 Study quality assessment and classification

After reading and analyzing the selected papers, the next step was to classify them according to parameters studied for each paper. Due to lack of specific literature on the topic discussed in this systematic review, the classification of parameters was based on the definitions and classifications adopted in the book Pro Hadoop [4], which is one of the most cited references in the analyzed studies and based on our own understanding since there are no classifications ratified by the scientific community. In this systematic review, the purpose of the classification parameters, is answer the questions raised in Section 1 and show which parameters have been identified and studied in the works that deal with the performance of Hadoop.

3.2.4 Data Synthesis

For the synthesis, were extracted the Hadoop configuration parameters that has influence on the performance, the central objectives and the experimentation methods.

4. Results

The results of this study are divided into three main sections, aiming to answer the initial questions. Section 4.1 focuses on identifying parameters which were used in the detected studies, Section 4.2 focuses on which parameters are affected by each Hadoop phase and Section 4.3 focuses on identifying which parameters are related to the workloads characteristics.

The Table 3 summarizes the parameters. The classification used in the table answers the first three questions of this systematic review. Discussions about it follow in the next sections.

4.1 Configuration Parameters Results

According to the classification adopted, we identified all parameters that somehow influences system performance and which have been studied in the papers that we have identified.

About 29 configuration parameters, which according to studies analyzed impact on system performance, were identified. Figure 1 shows a graph that points out the number of papers by parameters. More details in Section 5.

4.2 Hadoop Phases x Parameters Results

The execution of a MapReduce job is divided into a Map phase and a Reduce phase. The data output by each map task is written into a circular memory buffer. When this buffer reaches a threshold, its content is sorted by key and flushed to a temporary file. These files are then served via HTTP to machines running reduce tasks. Reduce tasks are divided into three sub-phases: shuffle, sort and reduce. The shuffle sub-phase copies the data output from the map nodes to the reducer’s nodes. The sort sub-phase sorts the intermediate data by key. Finally, the reduce sub-phase, which runs the job’s reduce() function and the final result is then written to the distributed file system [47].

Studies point out that most of the applications running on Hadoop, show a large difference between the behavior of the map and reduce phases. Therefore, it is very important know which parameters must be tuned according to the processing phases of an application on the Hadoop. In Table 3, in the column Parameter Phase, we classify the parameters which influence the phases of Hadoop: Map, Reduce and intermediate sub-phases of the Reduce phase. The sub-phases are classified as Merge/Shuffle phase. In
addition, Core Job was listed as a phase that represents those parameters that directly control essential functions of the job. The parameter dfs.block.size was classified by phase Number of maps just to demonstrate that the value this parameter is what defines the number of map tasks.

4.3 Workloads Characteristics x Parameters

Results

Optimizing Hadoop Performance through the characteristics of workloads are important in order to make full use of that cluster resources (CPU, memory, IO and network - see Table 3, in the column Workload Characteristics).

Among other examples, set certain parameters can reduce the IO cost and network transmission, but can cause a CPU overhead. This is an example if we configure the mapred.compress.map.output to parameter to true (default is false), it will decrease the size of data transferred from the Mapper to the Reducers, but will add more processing in the data compression and decompression process [4]. Furthermore, some parameters are correlated with each other. The relation between the io.sort.mb parameter (the amount of buffer space in megabytes to use when sorting streams) and configure the Java heap size (for Map and Reduce JVM processes) so that ample memory is still available for the OS kernel, buffers, and cache subsystems are other examples [23].

Workload characterization by CPU, memory, IO and network (via benchmarks) is essential before performing
any tuning parameters. Furthermore, it is important that we deploy the parameters are related to the characteristics of workloads. This allows identifying the parameters that can be set to obtain best performance.

5. Results Analysis

This section discusses the results of this systematic review and based on the answers from 3 initial questions. Furthermore, the results obtained from our study identified the main purpose of the studies analyzed, and the methods of experimentation and validation used in each study.

5.1 Questions Considerations

Answering the first question, it is clear that about 29 Hadoop configuration parameters are those which more impact on system performance. From these 29 parameters, 10 parameters were covered over 65% of the papers. We observed that the most discussed parameters on these works are `mapred.reduce.tasks` (The suggested number of reduce tasks for a job) with 31 papers of the 40 papers surveyed, `mapred.map.tasks` (The suggested number of map tasks for a job) with 23 papers, `dfs.block.size` (The basic block size for the file system) with 26 papers and `io.sort.factor` (The number of map output partitions to merge at a time) with 18 papers. These parameters are those that allow the framework to attempt to provide data locally for the task that processes the split. Although some parameters have not been widely exploited by most papers, studies have shown their importance in relation to performance.

Answering the second question, we identify the parameters that are affected by each Hadoop phase. This will also allow the targeting of tuning the parameters identified at each stage of Hadoop if this is required. This approach would be rather useful if we know the application characteristics and at what phase it is CPU, IO or network bound, for example. Thus, the tuning of the parameters would be directed by this prior knowledge.

Answering the third question, we can observe the parameters are impacted according to workloads characteristics. Thus, it is possible, in future work, observe the workloads characteristics and working with the tuning of Hadoop parameters targeted at them.

5.2 Purpose and Experimentation Approach

Results

Related work in this systematic review have had in common the exploration of configuration parameters and performance of Hadoop. However, we note that the studies have different core purposes, as well as different methods of experimentation. Thus, for better understanding, we classified the papers according to the main purpose and the methods applied.

For classifying the experimentation method, it was used the taxonomy proposed by Zelkowitz and Wallace [48]. In this taxonomy there are four approaches toward experimentation: (1) Scientific Method, (2) Engineering Method, (3) Empirical Method and (4) Analytical Method.

Table 4 shows the main purpose of each work and the main experimentation methods found in the analyzed papers. Importantly, the identification of the method used in the analyzed studies are not always as clear to identify. Some studies show a twofold interpretation. Thus, even with dubious features, in some cases, we classify according to the predominant method.

We observed that many studies focused between the purpose of obtaining performance models (about 22.5%) and performance prediction (37.5%). Other were divided into performance analysis and tuning performance, and only one job to prediction energy and performance penalties. Furthermore, we observed that 67.5% of the analyzed papers have used the empirical approach as experimentation method, and 32.5% have used the analytical method.

Most studies have adopted an empirical approach in experimental methods. This means that proposed a hypothesis and
validated by a method of statistical analysis. In other words, data were collected and statistically verified the hypothesis. On the other hand, some studies have utilized the analytical method. These studies have developed a formal theory to a problem, and results derived from this theory were compared with empirical observations.

5.3 Validation Methods

A further analysis was achieved through the researched information by this systematic review is about the validation methods used by the studies. Validation methods were analyzed in the same manner as the experimentation methods and were classified according to the taxonomy for software engineering experimentation [49].

The studies may be classified into three of the twelve methods considered in the taxonomy. It is noteworthy that once again, here, papers with dubious interpretations. Thus, the validation methods with predominant features were classified. It is possible to observe in Table 5 that most of the works used mainly dynamic analysis (benchmark) and simulation to validate their researches. As can be seen, the controlled dynamic analysis method is the method more used in researches projects and formal theoretical analysis method is the least used method.

6. Final Considerations

A systematic review on the relationship of the Hadoop configuration parameters and system performance was presented. The goal was a study to review the most recent research into this context and answer the following questions: (1) Which Hadoop configuration parameters has influences and impacts on system performance? (2) Which parameters are influenced by workloads characteristics? (3) Which parameters are influenced by Hadoop phases?

The results achieved were able to answer these questions and provide references for future works directed to its subject matter. Besides the expected answers, the results analysis identified the main purpose and the experimentation and validation methods approached by the examined papers. Once again, with great importance for reference in future works.

Table 4: Purpose and Experimentation Approach Results

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Approach</th>
<th>Scientific Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy Prediction</td>
<td>Analytical</td>
<td>[13]</td>
</tr>
<tr>
<td>Performance Analysis</td>
<td>Empirical</td>
<td>[32], [43], [12], [26], [21], [15]</td>
</tr>
<tr>
<td>Performance Model</td>
<td>Empirical</td>
<td>[18], [40], [33]</td>
</tr>
<tr>
<td>Performance Penalties</td>
<td>Empirical</td>
<td>[44], [19], [16], [35], [30], [25]</td>
</tr>
<tr>
<td>Performance Prediction</td>
<td>Empirical</td>
<td>[24], [11], [42], [37], [3], [28], [38], [9], [10], [29]</td>
</tr>
<tr>
<td>Performance Tuning</td>
<td>Empirical</td>
<td>[36], [41], [46], [31], [17]</td>
</tr>
<tr>
<td></td>
<td>Analytical</td>
<td>[5], [22], [23], [20], [34], [27], [39]</td>
</tr>
</tbody>
</table>

Table 5: Validation Results

<table>
<thead>
<tr>
<th>Validation Type</th>
<th>Validation Method</th>
<th>Scientific Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlled</td>
<td>Dynamic Analysis</td>
<td>[18], [24], [5], [45], [42], [22], [36], [42], [37], [3], [19], [28], [40], [32], [43], [12], [26], [41], [38], [13], [23], [21], [34], [27], [46], [31], [30], [39], [25], [29], [17], [33]</td>
</tr>
<tr>
<td></td>
<td>Analyss (Benchmark)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Simulation</td>
<td>[11], [14], [20], [35], [9], [10], [15]</td>
</tr>
<tr>
<td>Formal</td>
<td>Theoretical Analysis</td>
<td>[44], [16]</td>
</tr>
</tbody>
</table>

References


