Computers and Electrical Engineering 50 (2016) 166–179

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Novel heuristic speculative execution strategies in heterogeneous distributed environments☆

Xin Huang[a], Longxin Zhang[a,b,*], Renfa Li[a], Lanjun Wan[a], Keqin Li[a,b,c]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[b] National Supercomputing Center in Changsha, Hunan, Changsha 410082, China
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Article history:
Received 24 November 2014
Revised 7 June 2015
Accepted 8 June 2015
Available online 10 March 2016

Keywords:
Cloud computing
Dynamic loading
Hadoop
MapReduce
Speculative execution

ABSTRACT

MapReduce is a promising distributed computing platform for large-scale data processing applications. Hadoop MapReduce has been considered as one of the most extensively used open-source implementations of MapReduce frameworks for its flexible customization and convenient usage. Despite these advantages, a relatively slow running task called straggler task impedes job progress. In this study, two novel speculative strategies, namely, Estimate Remaining time Using Linear relationship model (ERUL) and extensional Maximum Cost Performance (exMCP), are developed to improve the estimation of the remaining time of a task. ERUL is a dynamic system load-aware strategy; using this strategy, we can overcome some drawbacks of the Longest Approximate Time to End (LATE) that misleads speculative execution in some cases. In exMCP, different slot values are considered. Extensive experiments show that ERUL and exMCP are applied to accurately estimate the remaining execution times of running tasks and reduce the running time of a job.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Distributed computing platforms, such as MapReduce [1] and Dryad [2], have been considered as the mainstream computing platforms of data processing, data mining, web indexing, and e-business. Interconnected by commodity computers through networks, these platforms can implement tasks in parallel with high reliability and at low prices, and can also easily add or remove nodes. Hadoop is an open-source implementation of the MapReduce framework and is fully implemented with Java language, which provides program interfaces, such as C++, Python, Perl, and Shell. Large companies, Yahoo!, Aliyun, Facebook and so on replace expensive computers with Hadoop to conduct large-scale computing because Hadoop can be easily customized and used. Hadoop MapReduce includes computing nodes (called TaskTracker) and storage nodes (called DataNodes). In general, a computing node can function as a storage node at the same time. The data blocks of MapReduce are stored in a *Hadoop Distributed File System* (HDFS). The HDFS is a distributed file system designed to run on commodity hardware [3]. A MapReduce job is divided into multiple map tasks and reduce tasks by JobTracker. JobTracker then assigns these tasks to TaskTracker to execute. A map task processes a data block by using a user-customized mapper operator and delivers the corresponding output to reduce tasks. Reduce tasks fetch input data from

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Hui Tian.
* Corresponding author at: College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China.
E-mail addresses: huangxin@hnu.edu.cn (X. Huang), longxinzhang@hnu.edu.cn (L. Zhang), lirenfa@hnu.edu.cn (R. Li), wancanjun2008@163.com (L. Wan), lik@newpaltz.edu (K. Li).

http://dx.doi.org/10.1016/j.compeleceng.2015.06.013
0045-7906/© 2015 Elsevier Ltd. All rights reserved.

the map output through networks and process data by using a user-customized reducer operator. Since a computing node can also be a storage node, map tasks and data blocks on which share the same node, it is known as data locality. Apparently, data locality can reduce the execution time of map tasks. Map tasks and reduce tasks can be executed in parallel because the data of these two types of tasks are independent.

If a task of a job requires an abnormally long execution time, then the total completion time of the job is affected. Such a task is called a straggler. A speculative copy of this task (also called a backup task) is run on another faster node to ensure that this task is finished earlier than the original task [4]. This mechanism is called speculative execution. In heterogeneous distributed environments, computing nodes differ in terms of computing capability and network bandwidth. In addition, a specific job may cause bugs. These problems can cause stragglers. Thus, the completion time of a job is affected. As such, schedulers cannot acquire accurate execution information of nodes and tasks during task assignment. Consequently, the performance of a scheduling strategy is affected. As a fault-tolerant technology, speculative execution can correct the wrong decisions of schedulers to some extent, thereby improving computing efficiency.

An original implementation of speculative execution (as Hadoop-Naive) [5] is examined in Hadoop-0.20 to enhance performance. However, this strategy cannot work well in heterogeneous MapReduce systems. To schedule tasks in heterogeneous systems, Li [6] analyzed the performance of heuristic power allocation and scheduling algorithms of parallel tasks with constrained precedence. Zhang et al. [7] developed algorithms to enhance task reliability when a dynamic voltage scaling (DVS) technique is applied to achieve low power consumption in heterogeneous computing systems. Xu et al. [8] proposed a genetic algorithm to perform parallel task scheduling on heterogeneous distributed environments by utilizing multiple priority queues. Shen et al. [9] presented several methods to schedule necessary computations to trace the vasculature in retinal images. Tian et al. [10] explored an adaptive data collection strategy by using different communication radii for nodes distributed in different locations to balance the energy consumption in heterogeneous wireless sensor networks. In terms of task management in Hadoop systems, Longest Approximate Time to End (LATE) is proposed as a strategy (Hadoop-LATE) to adapt to heterogeneous environments [11,12]. The LATE strategy can yield some improvements but may cause misjudgment when stragglers are determined. The strategy fails when the misjudgement occurs. Some drawbacks, including inaccurate estimated time and system resource wastage, also exist.

Several other components involving MapReduce and HDFS, such as HBase [13], ZooKeeper [14], Pig [15], and Hive [16], constitute the Hadoop system. The nodes should process the tasks of MapReduce and the tasks of other components at the same time. This process results in an unstable system load of computing nodes, thereby affecting the remaining execution time. Hadoop-LATE and *Maximum Cost Performance* (MCP; as Hadoop-MCP), which was described in a previous study [17], does not consider the effect of system load when the remaining execution time is estimated. As a result, the estimated remaining time becomes inaccurate, thereby affecting the speculation effect. In addition, MCP does not consider the different values in the slots when the values of cluster resources are evaluated.

To solve these problems, we devise a heuristic speculative execution strategy called *Estimate Remaining time Using Linear relationship model* (Hadoop-ERUL). Based on heterogeneous computing systems, the system load used to estimate the remaining time of tasks is considered in Hadoop-ERUL. Hadoop-ERUL can overcome some deficiencies of Hadoop-LATE; the former is also more concise and efficient than Hadoop-MCP. Hadoop-ERUL can estimate the remaining time more accurately and detect stragglers more rapidly and more accurately. Moreover, backup tasks can run on suitable nodes with this strategy. Compared with Hadoop-LATE, Hadoop-ERUL can reduce the execution time of a job by 26%. Hence, we presented the paper entitled “A Heuristic Speculative Execution Strategy in Heterogeneous Distributed Environments” in the Proceedings of the Sixth International Symposium on Parallel Architectures, Algorithms, and Programming 2014 (PAAP-2014) [18]. In this work, a heuristic speculative execution strategy was implemented with ERUL on heterogeneous environments. Another novel strategy called extensional Maximum Cost Performance (exMCP) is devised in this work by fully using the values of different slots that were ignored in MCP.

In our further study, the contents affecting the efficiency of the algorithm are extended. In particular, the conference paper is significantly extended and composed of more than 30% new contents, including new algorithms, discussions, and solid experimental results that are not shown in the conference version. The three major contributions of this further study are listed as follows:

- We propose a heuristic speculative execution strategy with Hadoop-ERUL on heterogeneous environments.
- We implement Hadoop-exMCP to overcome the drawback of ignoring the different values of the slots in heterogeneous computing systems.
- We consider the system load to estimate the remaining time of tasks. This strategy can overcome some defects of Hadoop-LATE; this strategy is also more concise and efficient than Hadoop-MCP.
- We demonstrate that our proposed Hadoop-ERUL can be used not only to estimate the remaining time more precisely but also to detect stragglers more rapidly and more accurately, as revealed by the experimental results of a set of randomly generated task graphs and graphs of real-world problems with various characteristics.

The remainder of this paper is organized as follows: Several related works on speculative execution in the MapReduce framework and the relationship between system load and execution time of tasks are discussed in Section 2. Several drawbacks of previous studies are analyzed in Section 2.3. Our new strategies, ERUL and exMCP, are presented in detail

Download English Version:

https://daneshyari.com/en/article/453938

Download Persian Version:

https://daneshyari.com/article/453938

Daneshyari.com