Future Generation Computer Systems 56 (2016) 407–420

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Increasing task consolidation efficiency by using more accurate resource estimations

Jesus Omana Iglesias [a,*], Milan De Cauwer [b,*], Deepak Mehta [b], Barry O'Sullivan [b], Liam Murphy [a]

[a] Lero, Performance Engineering Lab, School of Computer Science and Informatics, University College Dublin, Dublin, Ireland
[b] Insight Centre for Data Analytics, University College Cork, Cork, Ireland

HIGHLIGHTS

- We proposed an algorithm that can balance two conflicting criteria.
- We described three prediction techniques and a scalable scheduling heuristic.
- We investigate the behavior of several eviction strategies.
- We described a simulation framework to test the effectiveness of our approach.

ARTICLE INFO

Article history:
Received 4 March 2015
Received in revised form
28 August 2015
Accepted 31 August 2015
Available online 25 September 2015

Keywords:
Online scheduling
Cloud computing
Forecasting
Resource provisioning
Constraint programming

ABSTRACT

Cloud providers aim to provide computing services for a wide range of applications, such as web applications, emails, web searches, and map-reduce jobs. These applications are commonly scheduled to run on multi-sites clusters that nowadays are becoming larger and more heterogeneous. A major challenge is to efficiently utilize the cluster's available resources, in particular to maximize overall machine utilization levels while minimizing the application waiting time. We propose a methodology for achieving an efficient utilization of the cluster's resources while providing the users with fast and reliable computing services. The methodology consists of three main modules: (i) a prediction module that forecasts the maximum resource requirement of a task; (ii) a scheduling module that efficiently allocates tasks to machines; and (iii) a monitoring module that tracks the levels of utilization of the machines and tasks, and can evict one or more tasks from the machines for rescheduling if required.

There are multiple ways of predicting task requirements, scheduling tasks on machines and evicting task from machines. The decisions made in each module can have significant impact on not only the objective function but also on the efficiency of the decisions made in other components. We therefore study these different combinations and analyze their interaction in order to determine a configuration that meets the objective of the problem. To test our methodology we have developed a simulator and provide a detailed analysis of these interactions between different modules by using a publicly available trace from a large Google cluster (∼12,000 machines). Our results show that the impact of more accurate resource estimations for the scheduling of tasks and evicting lower priority tasks in case of over-utilization can lead to an increase in the average utilization of the cluster, a reduction in the number of tasks being evicted, and a reduction in task waiting time.

1. Introduction

Cloud providers, such as Google, IBM, or Amazon, must deliver reliable and efficient computing resources to a large diversity of situations. The computing resources of these cloud providers reside in data centers. These data centers are composed of one or several large clusters that host heterogeneous machines that have to process jobs submitted to the provider. The cost of acquiring and

* Corresponding authors.
E-mail addresses: jaomania@gmail.com (J.O. Iglesias), milan.decauwer@insight-centre.org (M. De Cauwer), deepak.mehta@insight-centre.org (D. Mehta), barry.osullivan@insight-centre.org (B. O'Sullivan), liam.murphy@ucd.ie (L. Murphy).

http://dx.doi.org/10.1016/j.future.2015.08.018
0167-739X/

especially of maintaining large clusters is very high and foreseen to grow even more [1–3]. Therefore, it is essential for cloud providers to efficiently utilize their clusters. The scheduling of jobs on machines should be performed quickly and in a manner that the amount of resources (e.g., CPU and RAM) allocated to the tasks satisfies their needs.

The problem of scheduling jobs on machines has been actively studied in the past [4,5]. The work presented here focuses on online scheduling, where there is no prior knowledge of when or what type of jobs will be incoming in the system and what will be its resource requirements. More specifically, we study the utilization trace of a Google cluster [6]. A Google cluster is a set of machines, organized into racks and connected by a high-bandwidth cluster network. Workloads on a Google cluster arrive in the form of *jobs* which consist of one or more *tasks*, each of which is accompanied by a set of attributes. Among these attributes, there is an estimate on how much CPU and RAM the task will be required at execution time. Further inspection of the trace leads us to find that these estimated values can differ significantly from the actual resource requirements.

As a first contribution, we are interested in estimating the peak resource requirements for each task. This is motivated by a large difference between maximum resources consumed by tasks as specified by users and those that are actually utilized during their lifetime. Our claim is that it is possible to leverage standard machine learning techniques to produce an estimation of the peak resource consumption of tasks improving on the estimation provided by the users. In turn these forecasts can be used by an online scheduling policy in order to maximize the overall utilization of the cluster while minimizing the mean waiting time of tasks. The underlying idea is to use more accurate information at scheduling time to reduce as much as possible resource over-provisioning at a level of a machine without sacrificing Quality of Service.

Unlike other traditional scheduling problems [7], the actual utilization of the machines cannot be known while solving the problem. The actual consumption of resources is varying as the tasks are processed. Therefore, the actual utilization of the machines can only be computed for the time that has already passed, that is, after making the scheduling decisions. Another significant discrepancy from traditionally studied scheduling problems is that tasks cannot be preempted [8,9] nor simply paused and resumed later on an other machine (migrated). In the problem at hand, an external mechanism is responsible for *evicting* one or more tasks from an overloaded machine in order to guaranty quality of service for the remaining tasks running concurrently on the same machine. Those evicted tasks are to be resubmitted for scheduling and processed on a machine from the beginning. In addition, the resource requirements and the durations of tasks can vary significantly, and the number of tasks arriving at any time-point could be tens of thousands. The challenge is to efficiently solve this highly dynamic multi-dimensional online resource constrained scheduling problem consisting of heterogeneous machines and tasks. For this purpose, it is necessary to mitigate the effects of having uncertainties. We therefore investigate the impact of different predictive methods applied to task requirements on the actual utilization of machines and the eviction of tasks.

As second contribution, we study the behavior of classical scheduling policies in this highly dynamic and uncertain environment. Our focus is laid on the two aforementioned criteria i.e., maximizing the overall system's utilization and minimizing a metric reflecting the average waiting time of tasks in the system. We then design two algorithmic enhancements over simpler policies and show how one can benefit from them. Naturally, since the actual requirements of tasks are uncertain at decision time, we study the impact of uncertain information used by the simple policies in contrast of more advanced scheduling policies.

This paper is an extended version of the work originally published in [10] where the authors presented and investigated the two former aspects – machine learning and scheduling policies – in the context of a Google cluster. We here provide an expanded version in which the techniques leveraged are explained in much finer details including the details of our simulation framework. Moreover, as an original contribution, we investigate the behavior of several eviction strategies. Indeed, a last aspect discussed in this paper is related to the eviction mechanism. Beyond the tasks' scheduling policies, the implementation of the eviction mechanism proved to influence the performance of the scheduler sensibly. The eviction policy implemented in Google's scheduler selects tasks to evict from an overloaded machine based on task priorities. We claim that this mechanism can impact drastically the quality of the schedule and explore alternative *eviction policies*.

As an experimental framework, we provide a methodology for efficiently scheduling tasks to machines, without precise knowledge of task resource consumption and arrival time, given the user specified resource requirements of tasks and job attributes. The methodology includes three modules: *prediction*, *monitoring* and *scheduling* allowing us to simulate the behavior of a large scale cloud computing system. We argue that these modules are necessary in order to efficiently assess how well-utilized a cluster will be, based on historical data and characteristics of the cluster. We present the results from three different predictors in order to understand the benefits of more accurate resource predictions on the scheduling process. In particular, we apply two machine learning techniques, namely multiple linear regression and random forest [11]. In addition, we also analyze an ad-hoc predictor that assigns to every task a fixed percentage of the user-defined limit. Moreover, the scheduling process requires that we constantly monitor the state of the machines in the cluster as well as the current utilization levels for each task. This information is extremely valuable in order to periodically update the prediction models. Furthermore, the scheduler also requires to know the status of the machines before deciding where to schedule each task. Finally, we describe a mathematical formulation of the scheduling problem and propose an adaptive and scalable greedy method for allocating tasks to machines.

The remainder of the paper is organized as follows. Section 2 provides background about the Google trace. Section 3 introduces our methodology, including a detailed description of all of its modules in Sections 4–6. We then present in Section 7 how these modules are contributing to our more general purpose simulation framework. Section 8 presents the evaluation and results of the proposed methodology. Section 9 gives an overview of the related work. Section 10 concludes the paper and presents the future work.

2. Background: Google trace

Unlike more traditional scientific or supercomputing environments, cloud computing serves a much broader variety of workload profiles, for instance, long-running Internet services, large-scale data analysis or even testing and developing of software applications [12,13]. The motivation of this work is stemming from the release of a data center usage data set from a Google cluster [6]. This trace contains of information about the computing power hosted locally and the workload to be processed on it. To a large extent the data set illustrates well the unique challenges that cloud providers have to face. The trace includes information for 29 days, in which millions of tasks were scheduled across 12,583 heterogeneous machines. The trace consists of more than 40,000 applications, which are called numerous times by thousands of users in the form of jobs [13]. In this paper, we study the first 48 h of the Google trace. In that period, almost 1.8 million tasks were submitted and scheduled, corresponding to approximately 40,000 jobs.