Cost-Time Performance of Scaling Applications on the Cloud

[Extended Abstract]

Sunimal Rathnayake, Yong Meng Teo
Department of Computer Science
National University of Singapore
Singapore
{sunimalr,teoym}@comp.nus.edu.sg

ABSTRACT

With cloud computing offering scalable resources and pay-per use pricing, it is an attractive platform for scalable computing where application execution is constrained only by the cost budget. Additionally, cloud applications are increasingly becoming larger due to recent advancements in big data processing and machine learning, among others. Due to the large number of resource configurations available on the cloud, it is a non-trivial task for cloud users to determine the largest size of the application executable for a given cost budget and execution time deadline. In addition, the challenge becomes multi-fold as resource demands of the application do not necessarily scale linearly with problem size. Given a cost budget and a time deadline, this paper introduces a measurement-driven analytical modeling approach to determine the largest Pareto-optimal problem size of an application and the corresponding cloud configuration for execution. We evaluate our approach with a subset of representative applications that exhibit range of resource demand growth patterns. We show the existence of cost-time-size Pareto-frontier with multiple sweet spots meeting user constraints. To characterize the cost-performance of cloud resources, we introduce Performance Cost Ratio (PCR) metric. Motivated by Gustafson's law, we investigate fixed-time scaling of applications on cloud and analyze the trade-offs between the application problem size, cost and time.

KEYWORDS

scaling, largest problem size, cloud, cost-time performance, Pareto-optimal configuration

1 INTRODUCTION

Among the many attractive features cloud computing comes with, scalability is of utmost important. Traditionally, in parallel computing, scalability was limited by the resources availability. Now, with the advent of cloud computing which offers theoretically unlimited resources as a utility, scalability is no longer constrained by the resources, but by the consumer's cost budget. Since cloud resources are heterogeneous, and their usage is charged based on the execution time, a greater focus is needed for understanding the impact of scaling on the cost while making scaling decisions. Scalability in the context of parallel computing can be broadly divided into two branches: application scalability and resource scalability. The resource demands of applications change when they scale in terms of the problem size. This is known as application scalability. Such scalable applications exhibit different resource demand growth for different input parameters. A simple example is an n-body simulation application in which the resource demand scales linearly with the number of simulation steps while resource demand scales quadratically with the number of bodies in the simulation. Moreover, due to explosion of big data and recent advancements in computer science such as machine learning and advanced scientific simulations, among others, the size of computer applications are increasingly becoming larger. Due to its cost effectiveness cloud is proving to be an attractive platform for these highly scalable applications. Secondly, the large range of resources available on cloud with different cost and performance leads to resource scalability. As a result of having scalable resources, the cloud consumer has the opportunity to choose a cost-efficient resource configuration on cloud. However, this task is not trivial due to the extremely large cloud resource configuration space.

Historically, studies of scalability in parallel computing have been mainly focused on time-performance [3, 4, 6]. One of the early works by Gustafson, well known as Gustafson's law argues that given a fixed-time, a near-linear parallel speedup could be achieved when the size of the application grows while increasing the parallel compute resources [2]. This is very much applicable for applications that run on on-premise resources. In the context of cloud computing where applications' scaling is constrained only by the cost budget, it is worth investigating the applicability of Gustafson's law and impact of fixed-time scaling of applications.

Given an application with a time deadline and a cost budget, we propose a measurement driven analytical modeling approach to determine cost-time Pareto optimal problem sizes of the application and cloud resource configurations for executing them. Although Pareto-optimal scaling is not new to parallel computing [1, 5], applying Pareto-scaling to investigate the trade-off between the application problem size, and, the cost and time of execution is new in the context of cloud computing. Moreover, to characterize the performance of cloud resources with respect to cost, we use "Performance Cost Ratio (PCR)" metric. We evaluate our approach on a a configuration space of more than ten million configurations from Amazon EC2 cloud and representative application that exhibit a range of different scaling functions.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

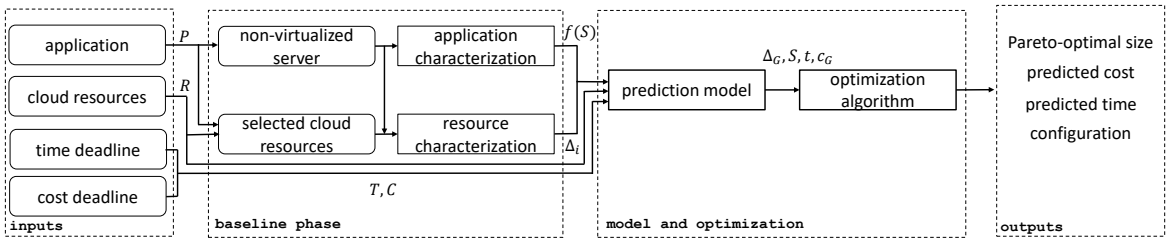

Figure 1: Approach Overview

2 APPROACH

As shown in Figure 1, given an application P with a cost budget C, a time deadline T, and, a set of cloud resources, our approach determines Pareto-optimal problem sizes S of P executable on the cloud. In addition, we determine Pareto-optimal cloud resource configurations to execute these sizes S of P. To obtain an accurate matching between cloud resources and P's resource demand for S, our approach uses a measurent-driven model. Figure 1 shows the outline of the proposed approach consisting of two phases (i) baseline execution to obtain the measurements, and, (ii) the model and optimization phase to determine the Pareto-optimal sizes and cloud resource configurations.

To determine Pareto-optimal problem sizes and the cloud resource configurations to execute them, our approach requires (i) characterizing the application and determining the resource demand growth function, and, (ii) characterizing the set of cloud resources and determining their execution rates. We use baseline measurements from a non-virtualized server and cloud resource instances to characterize the application and cloud resources. For characterizing the application, we measure the number instructions executed on the non-virtualized server while running the application for different problem sizes. These measurements are utilized to derive the application resource demand growth function. For characterizing cloud resources, we execute the application on cloud resource instances and record the execution time for each problem size. The execution rate for each cloud resource instance is computed by dividing the number of instructions measured on the non-virtualized server for each problem size of the application by the corresponding execution time recording on the respective cloud resource instance.

To minimize the measurement overhead, we measure the instructions executed on the non-virtualized server using non-intrusive hardware performance counters. A more accurate means of determining the instruction execution rate of cloud resources would be measuring the instruction count directly on the cloud resource itself. However, as commercial cloud vendors restrict access to physical layer of the cloud resources due to virtualization and security reasons, we are constrained to use a non-virtualized server similar to cloud resources with the same Instruction Set Architecture (ISA) for measuring the number of instructions executed.

We formulate analytical models to compute the instruction execution rate of a cloud resource configuration made up of a combination of one or more cloud resource types, compute the problem size of the application given the execution time and the cloud configuration, and, determine the execution cost for running a cloud configuration for a given time duration. The optimization algorithm takes as input C, T, the set of cloud configurations G and the growth function of P, $f(S)$ and determines largest Pareto-optimal sizes of P, S^{max}, optimal cost C', optimal time T' and, the corresponding cloud resource configuration.

3 EVALUATION

Our approach is evaluated on Amazon EC2 cloud using a representative subset of applications to cover a range of scaling functions with respect to the problem size. Thus, we first show that the chosen applications exhibit a range of scaling functions through a detailed workload characterization. Next, we introduce "Performance Cost Ratio" (ratio between the instruction execution rate and the cost for unit time) to capture the impact of resource capacity with respect to the cost of resource. Thirdly, we address the challenge of of having a large cloud resource configuration space and show the application of our approach to determine Pareto-optimal problem sizes to execute the application within given cost budget and time deadline. Validation of our approach for a subset of predictions from our model on Amazon EC2 reported a prediction accuracy of more than 81%. Lastly, we study the impact and trade-offs scaling applications on cloud using predictions from our approach where we discuss the Pareto-optimal problem sizes, effect of cloud resource PCR, and, impact of time deadline and cost budget on Pareto-optimal problem sizes.

4 CONCLUSION

This paper presents a measurement driven analytical modeling approach for determining the cost-time Pareto-optimal problem sizes executable for a given cloud application with a time deadline and a cost budget, and, cloud resource configuration for executing them. Our approach is validated on Amazon EC2 cloud using applications that exhibit a range of scaling functions. Using the model predictions, we study the impact and trade-offs for cost and time of scaling applications on cloud.

REFERENCES

[1] M. R. H. Farahabady, Y. C. Lee, A. Y. Zomaya, Pareto-optimal cloud bursting, *IEEE Transactions on Parallel and Distributed Systems*, 25(10):2670–2682, 2014.
[2] J. L. Gustafson, Reevaluating Amdahl's Law, *Communications of the ACM*, 31(5):532–533, 1988.
[3] M. D. Hill, M. R. Marty, Amdahl's Law in the Multicore Era, *Computer*, 41(7), 2008.
[4] K. Hwang, X. Bai, Y. Shi, M. Li, W.-G. Chen, Y. Wu, Cloud Performance Modeling with Benchmark Evaluation of Elastic Scaling Strategies, *IEEE Transactions on Parallel and Distributed Systems*, 27(1):130–143, 2016.
[5] L. Ramapantulu, D. Loghin, Y. M. Teo, An Approach for Energy Efficient Execution of Hybrid Parallel Programs, *International Parallel and Distributed Processing Symposium*, pages 1000–1009, 2015.
[6] X.-H. Sun, Y. Chen, Reevaluating AmdahlâĂŹs Law in the Multicore Era, *Journal of Parallel and Distributed Computing*, 70(2):183 – 188, 2010.