

A Model-driven Approach for Price/Performance Tradeoffs in Cloud-based MapReduce Application Deployment

Shashank Shekhar, Faruk Caglar, Kyoungho An, Takayuki Kuroda, Aniruddha Gokhale¹, and Swapna Gokhale²

¹ Institute for Software Integrated Systems, Dept of EECS
Vanderbilt University, Nashville, TN 37235, USA

Email: { sshekhar,caglarf,kyoungho,kuroda,gokhale}@isis.vanderbilt.edu

² Dept of CSE, Univ of Connecticut
Storrs, CT, USA

Email: ssg@enr.uconn.edu

Abstract. This paper describes preliminary work in developing a model-driven approach to conducting price/performance tradeoffs for Cloud-based MapReduce application deployment. The need for this work stems from the significant variability in both the MapReduce application characteristics and price/performance characteristics of the underlying cloud platform. Our approach involves a model-based machine learning capability that trains itself from executing a variety of MapReduce applications on different cloud service providers, and in turn providing useful price/performance tradeoff information to MapReduce application users. Additionally, the model-based platform serves to automate the deployment of a MapReduce application to the cloud by incorporating the tradeoff choices.

1 Introduction

Emerging trends and technical needs: With the ever growing size of data to be processed, the need for advances in processing very large data sets keeps growing. MapReduce [1] is a widely-used framework for Big Data, which has a parallel programming model and executes on large clusters. Since the number of *map* and *reduce* tasks used for a MapReduce application depends on a variety of factors, such as the size of the data set, the degree of reliability, presence of combiners and number of partitions, executing a MapReduce application requires elastic computing, storage and networking resources. These requirements makes cloud computing a perfect fit to execute MapReduce applications due to its support for elastic computing.

There are two significant challenges faced by MapReduce users. First, Cloud-based provisioning incurs a cost, which is hard to precisely estimate *a priori*. The reason is that the cost varies depending on both the MapReduce job characteristics stemming from the factors outlined above, and also on which cloud service

provider (CSP) is used to execute the job. The latter is due to each CSP having their own pricing model and Service Level Agreements (SLAs). Second, the performance of applications running in the cloud also varies significantly from one CSP to another [6]. This problem is not limited to MapReduce applications alone but can apply to many other types of applications that illustrate variability in the resources they consume. For this paper, however, we scope out the research investigations in the context of MapReduce applications.

Research contributions: In summary, MapReduce application users face significant challenges in making the right price/performance tradeoffs when deciding to use the Cloud to execute their applications, not to mention the learning curve involved in deploying their applications on a particular CSP’s platform. There is an urgent need for a solution that can help a MapReduce user choose from various CSPs by making appropriate price/performance tradeoffs, and subsequently automate the deployment of their application on the chosen cloud provider. To address this need we present preliminary ideas on a model-driven engineering (MDE) approach to making the price/performance tradeoffs and using these decisions to automate the deployment of the application to the cloud. We provide this capability as a web-based service to the end user.

The web service utilizes the MDE approach in the following manner. First, the web service engineers use their MDE tool – developed using domain-specific modeling – to train a machine-learning tool by executing a variety of MapReduce applications with different price and performance characteristics on a variety of CSPs. The deployment of the applications used in the training phase is automated via yet another MDE-based generative tool that can synthesize deployment scripts for the underlying CSP.

The web service offers the end user (i.e., the MapReduce application user) with an interface – essentially an abstract model of a MapReduce application and its properties developed using a MDE approach through the use of domain-specific modeling – so that the user is shielded from details of the CSPs; instead they focus on providing the key requirements of their MapReduce application to the web service. The machine learning tool owned by the web service can then classify an incoming MapReduce application supplied by the end user into a known class of MapReduce application categories. We leverage our prior experience on job classification [10] for this work. Note that the end user is not cognizant of the machine learning tool. The web service subsequently consults its trained engine and provides the end user with different price/performance tradeoffs. The end user in turn uses the web interface to make a deployment choice at which point the web service reuses its automated cloud deployment capabilities to deploy and execute the end user’s MapReduce job on the CSP.

2 Related Work

There has been some existing work to solve the problems we are handling in this research. Ganapathi et al. [3] have used Kernel Canonical Correlation Analysis (KCCA) to predict the performance of MapReduce application in the cloud

environment. Kadirvel et al. [4] have evaluated and compared wide range regression techniques for MapReduce applications. However, these research have not provided any guidance or tool for estimating the cost of deployment and execution. Liew et al. [8] have developed a tool suite called CloudGuide for estimating cloud deployment cost and performance for legacy web applications. Li et al. [7] have also proposed a framework, CloudProphet for achieving the same objective. However, they have not provided anything specific to MapReduce based applications.

SPACE4CLOUD [2] follows a model-driven approach for estimating cost and performance for different cloud systems and defines three meta-models: Cloud Independent Model (CIM), Cloud Provider Independent Model (CPIM) and Cloud Provider Specific Model (CPSM) for different levels of abstractions. The solution tries to address any type of cloud application, but the results are not encouraging for heavy workloads which is the primary criteria for any MapReduce solution. Thus, we need a solution which could help the customers and to cater to the variabilities and commonalities amongst the CSPs for which a model driven approach seems an appropriate choice.

3 Project Status and Ongoing Work

In present form we have developed an MDE-based cloud deployment framework that automates the deployment and execution of MapReduce applications used for training our machine learning-based engine as well as deploying a user-supplied application. We focus on the Hadoop framework, which is the most widely used MapReduce framework. For developing the modeling capabilities, we have used the Generic Modeling Environment [5] to model and interpret the deployment configurations and executing the MapReduce jobs.

For training our machine learning engine that can illustrate the price/ performance tradeoffs, we have used a Java-based open source machine learning software called Weka and the algorithm used is multilayer perceptron [9]. The training is thus far conducted on cloud platforms that include an OpenNebula-based private cloud infrastructure, and public clouds such as Amazon EC2 and Windows Azure.

Our ongoing work involves the process of finding the right set of factors that affect the performance of a MapReduce application. Kadirvel et al. [4] have identified and divided the factors into six categories i.e. resource, data, program, configuration, faults and environment. We are expanding the list of these factors by including the variabilities within and across CSPs. Some of these include performance variabilities due to different workload at different time of the day and at different zones of the CSPs' datacenters. Some of the factors which we are still investigating to how to take into account are network topology of the datacenter, variability in processor speed/ architecture and data locality. We are also performing sensitivity analysis to figure out the factors which affect the performance the most and leave out the factors which do not, which will reduce

the complexity. This will also help to refine the models and make the MDE tool more effective.

Our future work will involve using the predicted resources to estimate the cost of deploying and executing MapReduce jobs, thus helping the user to select the best CSP for the job. The framework can then deploy and execute the MapReduce job on the CSP without the user needing to deal with the intricacies of the CSP's infrastructure. We also need to develop the modeling abstractions for different CSPs and expanding the models for various factors that impact the performance.

4 Conclusion

In this work, we proposed a framework for conducting price/performance trade-offs in executing MapReduce jobs at various CSPs, selecting the best option and deploying and executing the job on the selected CSP infrastructure. All of these capabilities are driven by a model-driven engineering (MDE) framework that shields the users from the variabilities in the cloud service providers (CSPs) and also automates the deployment of the application on the CSP platform. We propose to offer such a solution as a web-hosted service, possibly as a software-as-a-service, by a providers who is willing to incur the initial cost of training the tool for various configurations on different CSPs. They can then provide services to customers who will pay a fraction of the cost of the saving they get by using the solution. We have conducted preliminary work including training a system on a private cloud platform as well as public clouds. The MDE abstractions are being developed and the realization as a web-hosted service is still under development.

5 Acknowledgment

This work was supported in part by NSF CNS SHF 0915976 and CNS CAREER 0845789. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
2. Davide Franceschelli, Danilo Ardagna, Michele Ciavotta, and Elisabetta Di Nitto. Space4cloud: a tool for system performance and costevaluation of cloud systems. In *Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds*, pages 27–34. ACM, 2013.
3. Archana Ganapathi, Yanpei Chen, Armando Fox, Randy Katz, and David Patterson. Statistics-driven workload modeling for the cloud. In *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on*, pages 87–92. IEEE, 2010.

4. Selvi Kadirvel and José AB Fortes. Grey-box approach for performance prediction in map-reduce based platforms. In *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pages 1–9. IEEE, 2012.
5. Ákos Lédeczi, Árpád Bakay, Miklós Maróti, Péter Völgyesi, Greg Nordstrom, Jonathan Sprinkle, and Gábor Karsai. Composing Domain-Specific Design Environments. *Computer*, 34(11):44–51, 2001.
6. Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. Cloudcmp: comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM, 2010.
7. Ang Li, Xuanran Zong, Srikanth Kandula, Xiaowei Yang, and Ming Zhang. Cloudprophet: towards application performance prediction in cloud. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 426–427. ACM, 2011.
8. Siew Huei Liew and Ya-Yunn Su. Cloudguide: Helping users estimate cloud deployment cost and performance for legacy web applications. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 90–98. IEEE, 2012.
9. Sankar K Pal and Sushmita Mitra. Multilayer perceptron, fuzzy sets, and classification. *Neural Networks, IEEE Transactions on*, 3(5):683–697, 1992.
10. Dili Wu and Aniruddha Gokhale. A self-tuning system based on application profiling and performance analysis for optimizing hadoop mapreduce cluster configuration. In *Proceedings of the 20th IEEE International Conference on High Performance Computing, to appear*, 18-21 Dec, 2013, Bangalore, India.