Simulating smarter provisioning in cloud based systems using reinforcement learning to achieve better dynamic scalability and higher efficiency

CS218 PROJECT 2

Fall 2016

SAN JOSE STATE UNIVERSITY

KUSHAL KHANDELWAL

VIKRAM DESHMUKH

Abstract

When it comes to cloud computing, there are many facets like provisioning, interoperability, and portability that have a crucial scope for improvement. Our project 1 focussed on gaining insight on the work done to simplify interoperability and portability using a standardized framework as proposed by mOSAIC. The aim of this project is to improve on the provisioning aspect of Cloud Computing. This project aims to increase provisioning efficiency using auto-reconfiguring of provisioning requests based on application categorization, auto-provisioning resources for seasonal peak load, and further improving provisioning efficiency using reinforcement machine learning[2][3]. This project focuses on 3 phases of the cloud VM life cycle ie. Provisioning, Dynamic work load and Seasonal load.

We propose optimizations at these facets with following goals:

1. Auto-reconfigure provisioning request based on historical data of categorised applications.
2. Auto-provisioning resources for seasonal peak load.
3. Dynamic resource provisioning using reinforced learning.

Table of Contents

Glossary

Action: possible actions that the learning agent can take

BW: Bandwidth

CPU/PE: Central Processing Unit

Performance Index: Ratio of load per Processing Element (PE or CPU)

RAM: Random Access Memory

SLA: Service Level Agreement

State: One of the possible states that system can be in

VM: Virtual Machine

List of Figures

1. Introduction

As part of our project 1, we studied the mOSAIC framework. It was while working on this project that we realized that there is still scope to work on the provisioning aspect in cloud computing to improve overall efficiency of the system.

2. Project Description

This project proposes a 3-pronged approach to enhance resources provisioning and overall efficiency. First approach involves reconfiguring provisioning requests depending upon the type of application the developer wishes to host. This project suggests collecting information of provisioning request across different categories of applications and compiling a knowledge base of provisioning request typical to a particular type of application. For the purpose of these project, we have shortlisted a few categories of applications like e-commerce, educational application, scientific research application, and financial markets. Each of these applications can be classified into further subcategories to narrow down to a provision request that is best suited for the client's application.

For example, an e-commerce application can be further classified depending upon the expected user load. A start-up may expect an initial user loads of a few thousand concurrent users. A mid-level enterprise might need to service users in the range of hundred thousands. While a large online portal may expect user traffic in the millions. Similarly, a scientific research application may be expected to have a continuous uniform load that can be attributed to research related activities in progress. An application about the financial markets can be classified into two types depending upon whether it only dependent on the local markets or international market. A financial application related to local markets may show daily peak usage during trading hours whereas an application that uses international markets data may expect traffic throughout the day. A mapping will be maintained in the application about

ideal provisioning request for each type and subtype of application based upon the classification.

The second approach involves automatically provisioning resources for applications depending upon past usage to accommodate seasonal peak load. This would be a preemptive measure to ensure the setup is capable of handling any seasonal peak load.

As a further enhancement, we are also providing reinforcement learning by implementing a q-learning algorithm that would learn by dynamically provisioning resources during the lifecycle of the system. Being a learning agent, the system would learn and improve over time. This implies that there may be some SLA violations in the initial phase which would reduce overtime as the system gets smarter. Hence, the client would be provided a choice as part of the SLA. The client would have a choice to select between a stringent-but-expensive provisioning approach and an approach that is adaptive and gets better over time. This can be better summarized by the chart taken from the paper [2] on q-learning:

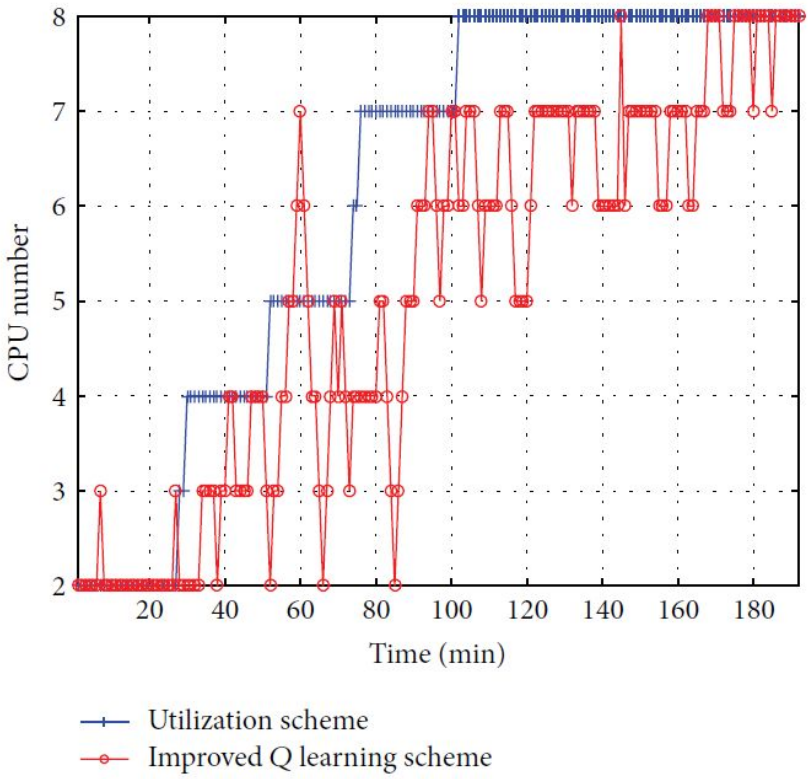

Fig 1. Comparison of CPU number between the improved *Q* learning scheme and the utilization scheme

3. Design

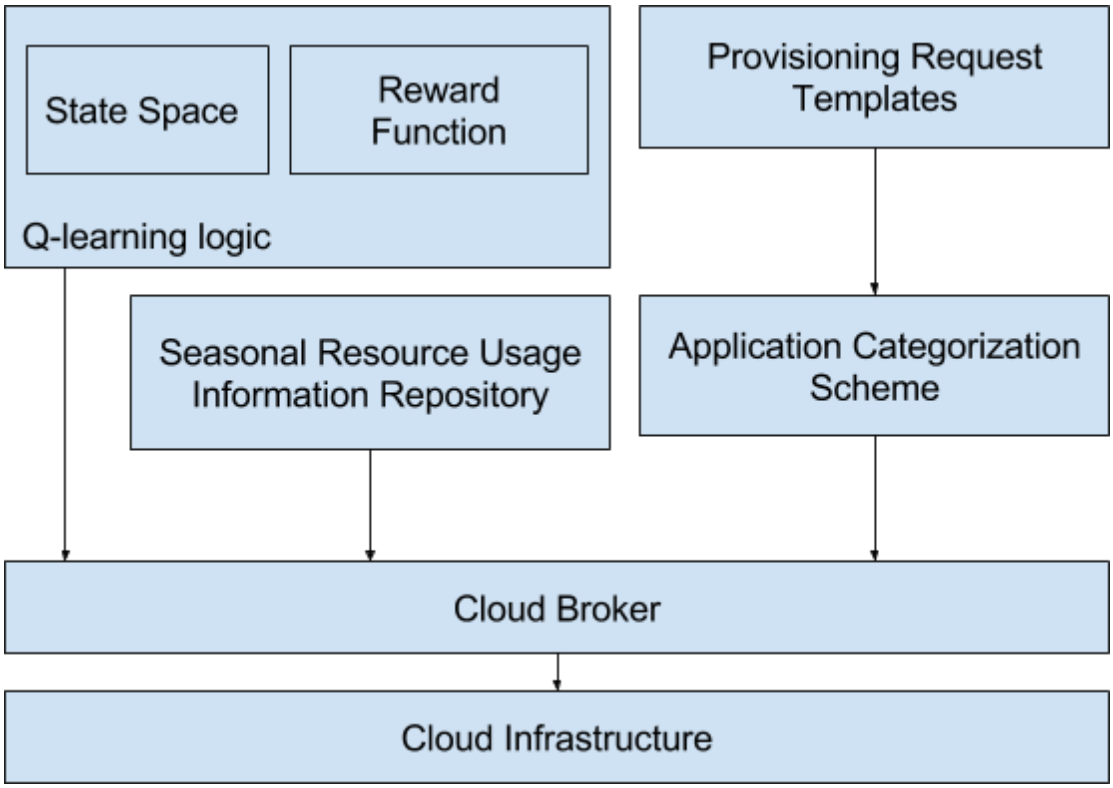

Fig 2. Overall design of the system with q-learning, auto-reconfiguration, and auto-provisioning blocks [2]

4. System Architecture

The three experimental approaches proposed by this project will be implemented as part of a package integrated within a package containing other essential elements of the cloud infrastructure like VMs, Broker, DataCenter, etc.

A bunch of custom data objects are used to represent data used in the demo. A 'State' represents a state of the system at a stage during the execution of q-learning algorithm. It essentially depicts the resources available with the system at a point in time. There is also a provisional request template object that holds the specification for a provisional request for a client application belonging to a particular category.

It includes information about various computing resources like CPUs (PE), BW, and RAM. Given below is the class diagram of the application.

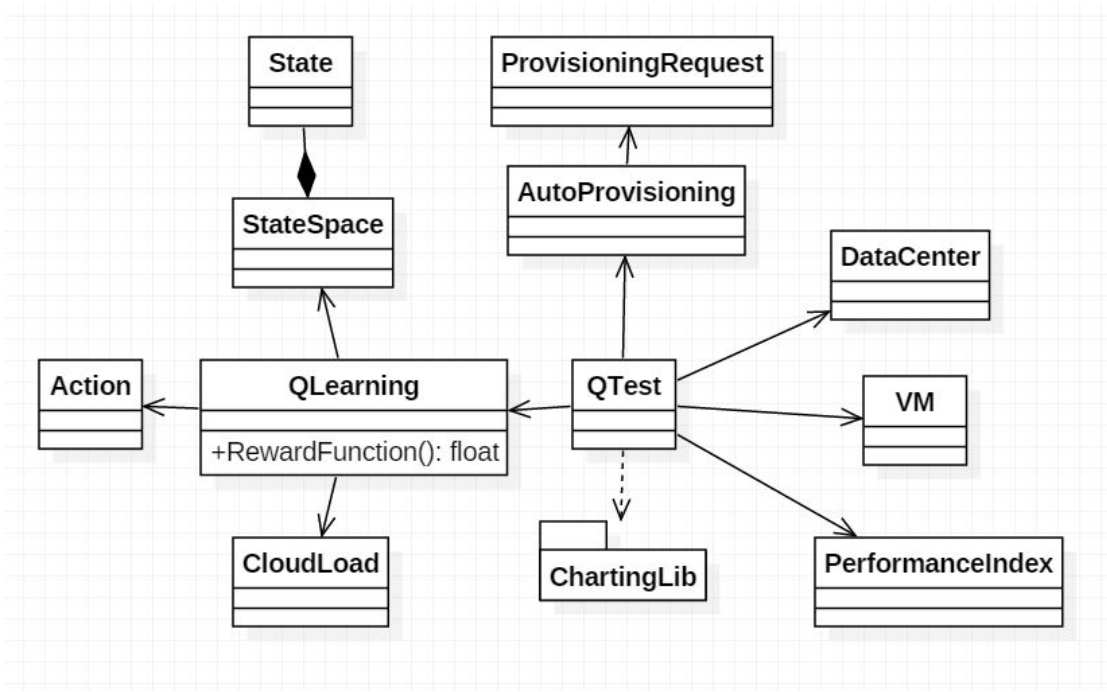

Fig 3. Class Diagram of the project application

5. Performance Evaluation

The simulation would be run against a dataset assumed after evaluation of various cloud applications across different categories.

5.1. Results

The test runs of our simulation at the moment are producing results that are not in line with our findings. Further refinements to our model and data are needed. These would eventually prove conclusively how the approaches suggested by this project fare in the simulation.

6. Conclusion

We expect our experiment to demonstrate an approach to dynamically scale and auto-reconfigure required cloud resources to effectively meet right requirements for the customer applications in a simulated environment. The auto reconfiguration logic is built to simulate learning from historical resource usage of other applications belonging to same category. This project also takes into account seasonal change in effective resource usage for similar applications to proactively reconfiguring customers resources who belong to same category by forecasting future requirements.

Acknowledgements

We would like to take this opportunity to express our gratitude and sincerely thank Prof Dr Melody Moh for giving us this opportunity to work on this exciting project and for continuously providing guidance whenever we were in doubt and needed some clarity. Thank you professor for all the valuable suggestions that helped us in finalizing on our project topic and the valuable feedback that you provided throughout the course of this project.

References

1. Al-Ayyoub, M., Jararweh, Y., Daraghmeh, M. et al., “Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure”, in Cluster Computing, Volume 18, Issue 2. Irbid, Jordan: Springer US, 2015, pp 919–932. (http://link.springer.com/article/10.1007/s10586-015-0449-5)
2. Z. Peng, et al., "Research on cloud computing resources provisioning based on reinforcement learning," Mathematical Problems in Engineering, vol. 2015, pp. 1–12, 2015. (https://www.hindawi.com/journals/mpe/2015/916418/)
3. G. C. Chasparis, "Reinforcement-learning-based efficient resource allocation with demand-side adjustments," 2015 European Control Conference (ECC), Jul. 2015. (http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7331004)
4. P. Jamshidi, et al., "Self-learning cloud controllers: Fuzzy Q-Learning for knowledge evolution," 2015 International Conference on Cloud and Autonomic Computing, Sep. 2015. (http://ieeexplore.ieee.org/document/7312157/)
5. E. Barrett, E. Howley, and J. Duggan, "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud," Concurrency and Computation: Practice and Experience, vol. 25, no. 12, pp. 1656–1674, May 2012.