Recent Insights into Proactive Resource Provisioning Utilizing Machine Learning Techniques

Syed.Karimunnis[1],

Department of Computer Science and Engineering, Koneru Lakshmaiah
Education Foundation Vaddesvaram, Guntur, AP, India-522302, karimun1.syed@gmail.com

Supriya Menon M[2]

Department of Computer Science and Engineering, Koneru Lakshmaiah
Education Foundation Vaddesvaram, Guntur, AP, India-522302,
supriyamenon05@gmail.com

ABSTRACT:

Dynamic management of cloud resources in a specific project dedicated to delivering a private cloud (or virtual private cloud) to a sole client venture. This operation relies on acquiring resources as needed from a public cloud. An innovative proactive strategy has been introduced to automatically adapt resource scaling within the private cloud according to real-time system demands. This approach caters to both immediate and pre-planned resource requests, utilizing machine learning to forecast future workloads based on historical data. The research findings strongly indicate that this method substantially enhances profitability for the service-providing enterprise while concurrently lowering costs for the client enterprise.

KEYWORDS: Virtual Machines, Decision Maker, cloud

1. Introduction

A critical and an essential aspect of cloud computing is its adaptability: the ability of leveraging or pulling down from a resource repository [1]. Auto-scaling capability empowers an organization to manage the costs associated with computer resource architecture. Current automatically scaling tools and techniques [2] mainly depend on parameters such as CPU cycles used, memory utilization, and traffic at network sites. Eventually, accurately determining the scaling metrics and thresholds is challenging, particularly with complex application models and limited, low-level resource usage indicators [2]. The system presented in this paper avoids using these conventional markers and instead employs evolving techniques of ML to forecast the upcoming demands of resource requirements for executing automatically scaling operations.

In the majority of existing cloud environments, workload requests are typically executed based on available resources, operating on a optimal efficiency metrics. Such appeal is primarily coined as an On-Demand (OD) service-request [3]. Neverthless, cloud environment mainly get centric around workloads with aforementioned requirements in terms, like Advance Reservation Requests (ARs) [4]. This type of reservation requests are vital for needs that are intended to be served before a specific deadline, thus ensuring a guaranteed responsibility in terms of Quality of Service (QoS) [4]. The Service provider engages into an alliance with the customer in terms of Service level characteristics i.e., an Service Level Agreement (SLA). The service provider and the customer enter into a Service Level Agreement (SLA) when the service provider grants an AR request. ARs are essential parts of distributed and cloud systems [5]. Still, most public cloud providers don't enable augmented reality capabilities at this time. In a context where best-effort request execution is necessary, maintaining service levels may be difficult in the absence of adequate AR assistance.

2. Related Work

[1], Managing assets everywhere scale while giving execution detachment and effective utilization of fundamental equipment is a key test for any cloud the executives programming. Most virtual machine (VM) asset the board frameworks like VMware DRS bunches, Microsoft PRO and Eucalyptus, don't right now scale to the quantity of hosts and VMs bolstered by cloud specialist organizations. Notwithstanding scale, different difficulties incorporate heterogeneity of frameworks, similarity limitations between virtual machines and hidden equipment, islands of assets made because of capacity and system network and restricted size of capacity assets.

[2], In order to reduce administrative expenses, we offer the broker new and creative ways to effectively reserve individual slots. These methods quickly handle high demand by utilising approximation algorithms and dynamic programming. Our large-scale simulations on large-scale Google clusters show that the broker can save a significant amount of money using these techniques.

Objectives

1. Investigating machine learning as a potential substitute for resource provisioning in order to address the problem of operating cloud-based scientific workflows.

2. To ascertain and evaluate whether a machine learning model that has been trained can produce answers much faster than an algorithm while maintaining an identical degree of quality.

3. To determine appropriate resources to provide (using machine learning) in order to feed the knowledge into the algorithm.

4. Using machine learning in conjunction with a scheduling algorithm, resource provisioning will be finished in order to expeditiously identify optimal scheduling options for scientific processes and maintain a good make span for the process within the allocated budget.

Problem Statement

Common auto-scaling systems typically rely on metrics such as Processor utilization, memory utilization, and traffic over network. However, accurately determining the scaling metrics and thresholds is challenging, particularly in complex application models where the resource usage implications have restrictions with low level.

The model acquainted here doesn't rely on these conventional markers. Instead, it leverages ML efficiency in forecasting the expected resource needs for executing automatically scaled tasks.

3. Methodology

The third party charges the clients belonging to the private cloud a more expensive rate for each unit time than the cloud suppliers so as to procure a benefit.

Nonetheless, the agent charges clients every second instead of on a hourly premise. These guarantees clients pay just for time their demand is running. The dealer auto-scales the assets proactively utilizing a forecast framework dependent on a machine learning procedure and figures the present number of asset required for taking care of the up and coming solicitations while guaranteeing that a benefit is created for the go-between cloud supplier.

The Broker

End-user requirements must be handled by the broker. When a request is received, task scheduling modules and component mapping are used to determine its viability. In addition, the broker sends user requirements to MLE via DM, which enables MLE to learn from the patterns of these incoming user demands and improve its predictive capabilities.

The Match-Make Scheduler module is also utilized by the Broker for matchmaking and scheduling, two essential tasks in resource management. Additionally, MLE replicates duties related to matchmaking and scheduling for anticipated requests using this module.

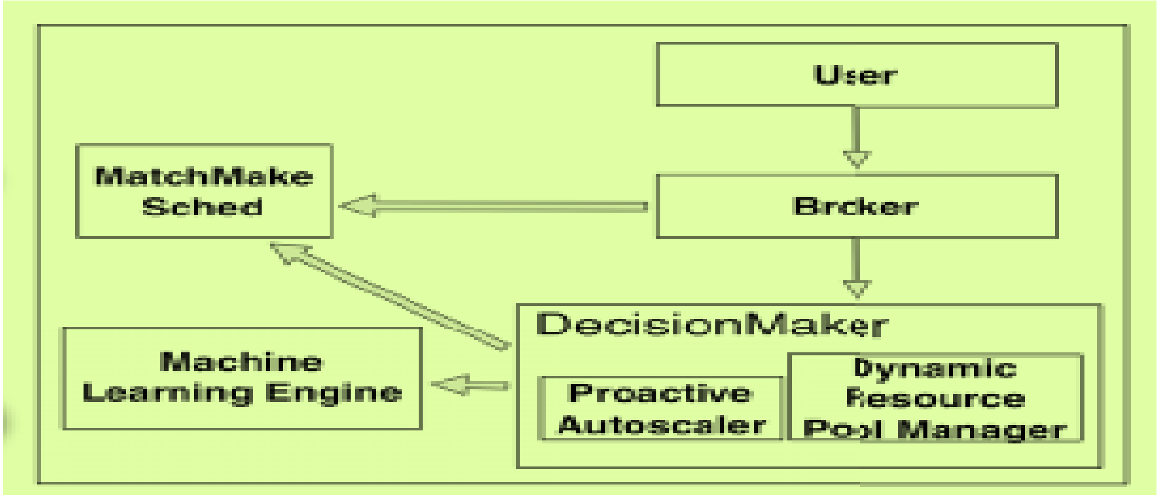

Figure1 Proposed Framework

Decision Maker

After the Decision Maker receives an answer from MLE to its request prediction inquiry, it decides how to scale. Using a program named Weka, MLE forecasts upcoming requests using a machine learning technique based on Linear Regression (LR). For a given volume of queries, this machine learning approach—which is especially aided by LR—reveals expected traits. In addition, several machine learning algorithms, like Support Vector Machines (SVM) [8], are being considered for simulation in addition to LR.

Proactive Algorithm

By examining the start and finish timings of individual resources, the proactive algorithm in data mining seeks to operate at regular intervals and forecast the exact quantity inside the chosen system. MLE goes through an initial training phase in this procedure to learn the demand patterns. For every resource in the system, the Decision Maker (DM) assesses the quantity of resources that are now available as well as the scheduled requests. The DM then asks MLE to determine the characteristics of the k upcoming requests that are anticipated as well as the existing state of the N assets that the DM has already purchased.

After that, the DM considers the requests that are predicted and models the matching sequence of execution and matchmaking for the k expected request needs that are anticipated to be incorporated into the system.

4. Results and Discussion

Simulation tests are conducted on a PC with an Intel Core i7 CPU, which has eight cores (2.8 GHz) and four gigabytes of RAM. The prototype broker and other related parts are housed in the machine stated before, which powers the prototype system. Concurrently, the user module, which generates requests, operates on a different PC that has an Intel Dual Core 3.0 GHz CPU and 4 GB of RAM.

Every simulation was executed for an extended duration to ensure the system reached a stable state. For each performance metric, there were enough iterations of each experiment to produce a range of within ± 5% at a 95% confidence level. Request arrival rates can change depending on the day of the week and time of day. Although there was room for variation in the arrival rates, the rates listed below were thought suitable for carrying out the comparative analysis presented in this work.

The system experiences a low arrival rate at first, which progressively changes to a high arrival rate when the load factor, f, indicates that the ratio of received requests to all simulated requests has reached a certain value.

i) Arrival Rate

The effect of a variable on the profit earned in System I and System II is shown in Figure 2. There appears to be a direct correlation between the arrival rate and the rise in broker profit in both systems. This correlation arises from the fact that higher arrival rates raise the possibility of making a bigger profit per hour by generating more requests per unit of time. Between 2.5 and 4 times the profit that System II makes is what System I consistently produces.

ii) Load Factor

The comparison of System I and System II's accrued profit for a range of f values is shown in Figure 3. System I consistently generates more profit than System II in all instances. The profit and f have an inverse relationship; that is, when f rises, so does the profit. Interestingly, the hourly profit peaks at f = 0. The system is now just experiencing the increased arrival rate. The mix of greater and lower arrival rates, however, gets smaller as f rises. The system only encounters the reduced arrival rate when f = 1, which results in lesser earned income.

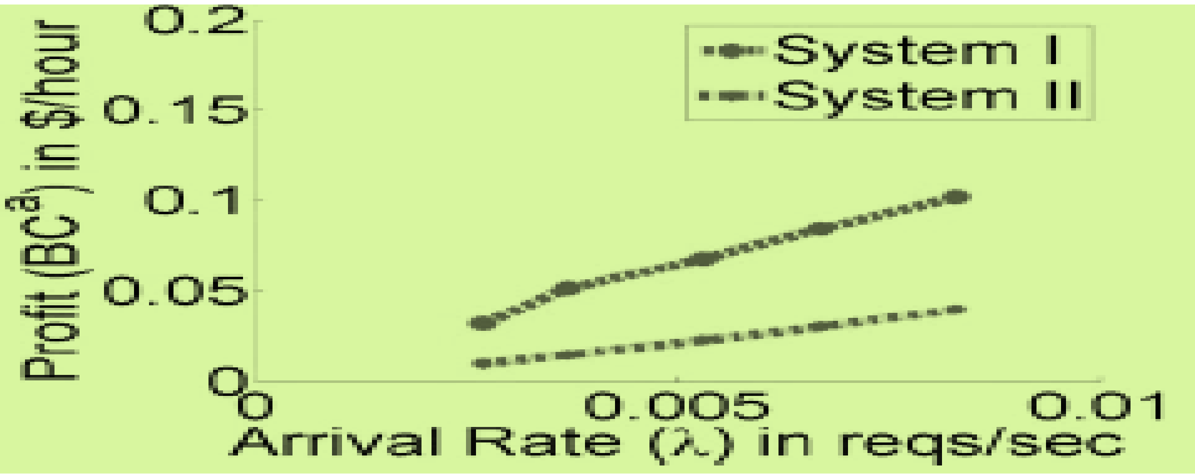

Figure 2 Impact of Arrival rate

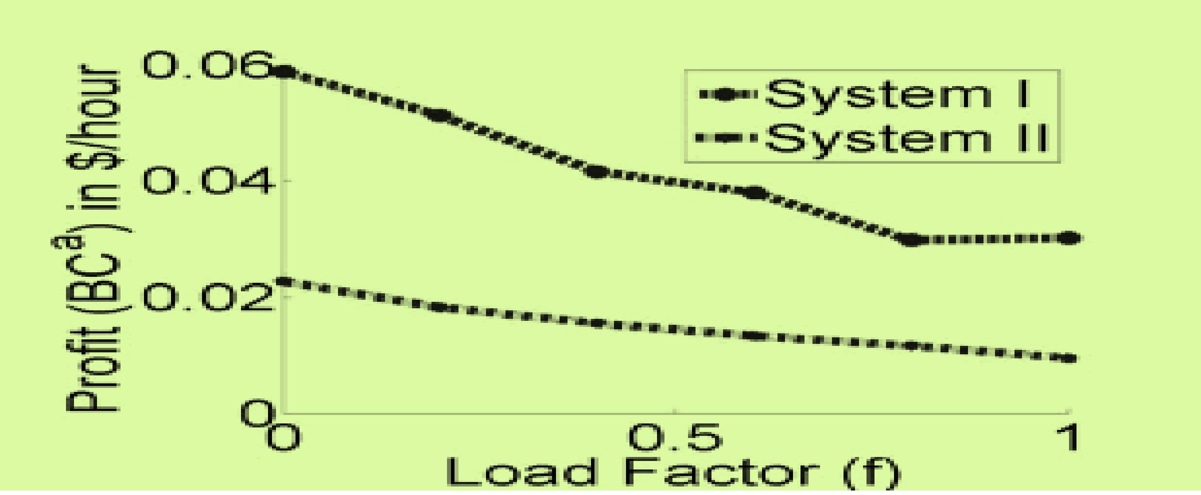

Figure 3 Impact of Load Factor

5. Conclusion

The proposed study introduces an alternative system and associated computations that underpin a proactive auto-scaling approach. The envisioned middle-tier architecture assesses the required system resources by anticipating the characteristics of incoming requests. This suggested system not only reduces costs for clients but also generates profits for the

intermediary cloud provider hosting the broker, as evidenced by simulations and model evaluations.

References

[1] T. Lorido-Botrán, J. Miguel-Alonso and J. A. Lozano, "Auto-scaling Techniques for Elastic Applications in Cloud Environments," 2012.

[2] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows," in *High Performance Computing, Networking, Storage and Analysis*, 2011.

[3] J. O. Melendez and S. Majumdar, "Matchmaking on Clouds and Grids," *J. Internet Techonology*, vol. 13, no. 6, pp. 853-866, 2012.

[4] I. Foster, C. Kesselman, C. Lee, B. Lindell, C. Nahrstedt and A. Roy, "A distributed resource management architecture that supports advance reservations and co-allocation," in *Quality of Service, IWQoS*, London, 1999.

[5] R. Buyya, S. K. Garg and R. N. Calheiros, "SLA-Oriented Resource Provisioning for Cloud Computing: Challenges, Architecture, and Solutions," in *CSC '11 Proceedings of the 2011 International Conference on Cloud and Service Computing*, 2011.

[6] Amazon Web Services, "Amazon VPC," [Online]. Available: http://aws.amazon.com/vpc/. [Accessed 08 2014].

[7] U. o. Waikato, "Weka 3: Data Mining Software in Java," [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed 06 2014].

[8] Pentaho, "Time Series Analysis and Forecasting with Weka," [Online]. Available:http://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka. [Accessed 08 2014].

[9] A. Gulati, G. Shanmuganathan, A. Holler and I. Ahmad, "Cloud-scale resource management: challenges and techniques," in *HotCloud'11 Proceedings of the 3rd USENIX conference on Hot topics in cloud computing*, Portland, 2011.

[10] B. Sotomayor, R. S. Montero, I. M. Llorente and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," *IEEE Computer Society*, vol. 13, no. 5, pp. 14 - 22, 2009.

[11] "Amazon CloudWatch," [Online]. Available:http://aws.amazon.com/cloudwatch/. [Accessed 06 2014].

[12] W. Wang, D. Niu†, B. Li and B. Liang, "Dynamic Cloud Resource Reservation via Cloud Brokerage," in *Distributed Computing Systems (ICDCS)*, Philadelphia, 2013.

[13] L. R. Moore, K. Bean and T. Ellahi, "A Coordinated Reactive and Predictive Approach to Cloud Elasticity," in *Fourth InternationalConference on Cloud Computing, GRIDs, and Virtualization*, Spain, 2013.

[14] H. Wang, J. Jin, Z. Wang and L. Shu, "On a novel property of the earliest deadline first algorithm," in *Eighth International Conference on Fuzzy Systems and Knowledge Discovery*, Shanghai, 2011.

[15] J. Yu, R. Buyya and C. K. Tham, "Cost-based scheduling of scientific workflow applications on utility grids," in *First International Conference on e-Science and Grid Computing*, Melbourne, 2005.