Kuwait J. Sci. **44 (1)** pp. 58-63, 2017

Speculation resource provisioning in high-performance computing

Leena Sri[1], Balaji Narayanan[2,*]

[1]*Dept. of Information Technology, Thiagarajar College of Engineering, Madurai-625015, India*

[2]*Prof & Head, Dept. of Information Technology, K.L.N. College of Engineering, Madurai, India*

[*]*Corresponding author: balaji_jet@yahoo.com*

Abstract

Distributed computing gives a backing to buyers to diminish their inner foundation, and for suppliers to expand incomes, utilizing their own particular framework. The proper load balancing and dynamic resource provisioning improves cloud performance and attracts the cloud users. In this paper, we propose an automated resource provisioning algorithm, Speculation resource provisioning , prompting load balancing through speculative approach in resource provisioning. As an attempt to quantify resource allocation we use two level adaptive prediction mechanism to check the computational patterns of past resource allocation to the future requirement. The framework guarantees suitable resources required for the application, by dodging over or under-provisioning of resource and supports energy-efficiency in resource allocation. We use estimation methodology to address the variability in the historical data to balance the speculation overhead. We have conveyed our proposed work in an open source cloud structure and contrasted our outcomes and other machine learning approaches. Our Experimental results demonstrate adaptive resource allocation over customer-driven service management under the rapidly changing requirements of cloud computing.

Keywords: Cloud computing; dynamic resource provision; energy efficient resource allocation; speculation resource provision.

1. Introduction

Cloud computing inherently trades the computing paradigm of utility computing and has attracted a large number of service providers and consumers for its elasticity and lack of capital upfront nature to their business compliance. Cloud computing is not a new technology; rather it is a new model emphasizing service-oriented architecture and virtualization. The cloud resources are properly managed and served for the user / developers based on their needs. This is undergone by cloud resource management and allocation system as per Zhuang *et. al.* (2013). The major challenges in cloud datacenter are carbon emission because of uninterrupted service for 24x7. The energy efficient load balancing is needed to be applied in cloud datacenter for the beneficiary of resource provider. Hence our algorithm provides the solution for optimized usage of resources.

For automatic resource provisioning, it is fundamental that resource displaying, submission, monitoring, and selection are given more consideration. The instance set up time is essential for servicing the requests. This may usually takes between 5-15 minutes discussed by Li *et al.* (2011).The efficient instance set up time is possible only when the history of resource consumption and allocation is known in prior. In our paper we have discussed about resource prediction model which is inspired by speculative mechanism. The instance setup time should not violate QoS according to Samimi *et al.* (2014).

The rest of this paper is organized as follows: Section 2 explains Background for cloud resource provisioning model section 3 describes Motivation towards our Resource Provisioning algorithm, Section 4 presents Experimental setup of our approach Section 5 shows Results and discussion, and Section 6 concludes the paper.

2. Background

There are many algorithms implemented in cloud resource management for fast provisioning of resources and load balancing such as Datacenter control algorithm (DCA) by Urgaonkar *et. al.* (2010) framework structural planning made out of dispatcher by Garg & Buyya (2011). Local and global manager, and RC2 algorithm. All such algorithms have the criteria to ensure QoS, availability and responsiveness for the client's SLA as discussed by Cao *et al.* (2014). The resource monitoring module is

the first phase of the cloud resource management which provides the information of current workload of servers Aljazzaf (2015). The cloud resource management is comprised of three phases: Monitoring, Prediction and allocation phases. The monitoring phases measures the application specific performance parameters (resource utilization, energy consumption). Many heuristic and artificial intelligence algorithms are used in Prediction phase. Scheduling and resource allocation algorithms are implemented in Allocation phase. The accounting and metering modules are incorporated with allocation phase. Many pricing models are adopted based on application characteristics like pay-per use model, instance pricing model and spot instance model.

In practice cloud brokers plays vital role for resource reservation and resource pricing model. Moreover, clients may not predict and reserve sufficient resources for their applications, and hence demand resource migration may occur as per Jamshidi *et al.* (2013). The resource migration may violate the availability QoS because of network bandwidth overhead according to Voorsluys *et al.* (2009). The intelligent and adaptable resource allocation algorithm is to be devised for addressing all the phases of resource allocation and withstanding the dynamic scaling environment as suggested by Huang *et al.* (2014).

3. Motivation of our algorithm

So as to attain intention of research we have designed the resource provisioning algorithm.

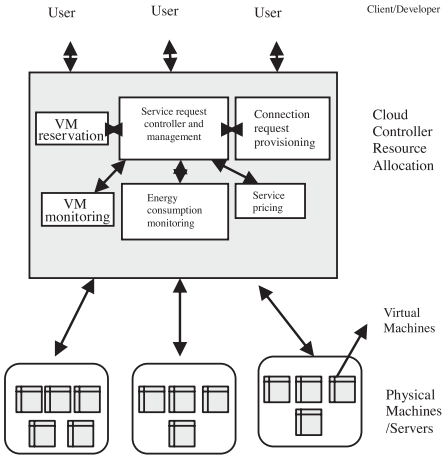

Fig. 1. Datacenter Architecture

VM resource allocation is the process of mapping the virtual resources into physical host machines as in Figure 1. After receiving requests from the user, the Resource Allocation System (RAS) in the data center controller will first apply Evaluation filters to select eligible hosts to provide computing capacities, CPU cores, RAM and image properties. The RAS in cloud controller should decide which host to run the user application in a VM. Early studies shows that there are two host level resource allocation strategies like Random (rand) and Max Core First (mcf) are used during resource provisioning by Hu *et al.* (2013).

3.1. State-of-the art

In our RAS, we focus on auto resource provisioning and an energy efficient cloud datacenter. Our research problem actually relates to time-series analysis. For automatic resource provisioning, it is fundamental that resource displaying, submission, monitoring, and selection are given more consideration. We have arranged the resources as active and idle based on the resource utilization: this mechanism ensures the active PMs and puts the idle PMs into standby mode, hence supporting an energy efficient cloud. For appropriate resource selection, our resource provisioning algorithm utilizes posteriori optimization techniques called Speculation (spec) algorithms. The shared Datacenter model is used for evaluating our proposed work. The time domain queuing model is suggested which expose weighted fair queuing mechanism to collect all the jobs in a queue. Each queue corresponds to particular application. The random arrival rate and exponential probability service time of the queues are measured. Whenever jobs arrived at the queue the monitoring modules are initiated to estimate the workload of the datacenter like load factor, resource density and energy consumption. The efficient host to handle the requested service is predicted by the prediction module. The outcome of the prediction module is to provide expected service time in an expected load to yield a minimum response time. The Expected service time can be predicted only knowing its history of occurrence of a job and its servicing factor. We use speculation mechanism for predicting the appropriate hosts to handle the service. The speculation works in two ways 1. Predicting the correct host among ‘n’ number of hosts and verifying accordingly during its runtime and 2. discarding the host at any point of time if its outcome doesn’t match the goal.

3.2. Resource provisioning mechanism

The history of service and the pattern of the requests are maintained in History Table and Pattern Table. The history table of size ‘n’ is adopted for recording the current requests and it is of fixed bound. The current requests are overridden in a very old entry when it exceeds the

size. The resource patterns associated with each record are saved in a pattern table for ensuring users SLA and for non violating providers QoS. The predictors are used for dynamic capacity planning, on demand resource provisioning and administering the cloud resource usage Tuah *et al*. (2003). When the requests arrive with a similar SLA pattern the application execution characteristics are learned from pattern table. The pattern table records the best application characteristics of each different request along with SLA. The characteristics of an application are the resource specific metrics such as resource usage (CPU, memory, Storage), Server Load Rate and Execution Time. The Execution time gets quantify with the service metering and accounting. The string pattern matching algorithm is used for estimating similar resource patterns of a request as per Caron *et al.* (2010).

4. Experimental setup

The evaluation of the resource provisioning algorithm is to be in a targeted cloud environment i.e IaaS. Constructing controlled and repeatable real time cloud test bed in a real environment is practically time consuming and very expensive. Hence usage of simulators is economically boon to the researchers who can test their algorithm and ideas free of cost and evaluating performance bottlenecks before testing it in a real environment. We have chosen low simulation overhead tool CloudSim by Garg & Buyya (2011), for implementing our resource allocation algorithm induced by speculative mechanism. The workload traces and data used in simulation have chosen from CoMon project, a monitoring infrastructure for PlanetLab: http://comon.cs.princeton.edu/ and an Application service benchmark, such as TPC-App García *et al.* (2006).

The list of simulation parameter is shown in Table 1.

Table 1. Simulation parameters

Simulation Parameter	Value
Total number of users	600
Total number of Host	20
Total number of Virtual Machines	150
Initial price of VM in *Cost/MI $*	[10, 500]
Electrical energy in *MW*	[0.1, 1.0]
Bandwidth in *Bits per Second*	[100, 1000]
Computing power in *MIPS*	[100, 1000]
RAM in *MB*	[256- 2048]
Energy price in *Cost/MI $*	[1, 100]
Deadline in *msec*	[100,400]

The application workload is estimated by request arrival rate and service distribution. Our synthetic workload uses Poisson request arrival and deterministic request size as in Figure 2 & 3.

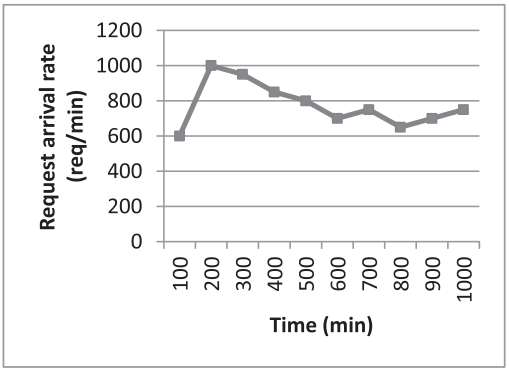

Fig. 2. Request arrival rate for random sampling CoMon project

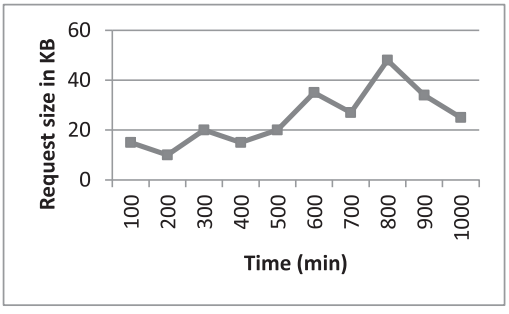

Fig. 3. Average request size for random sampling CoMon project

The criterion and the heuristic approach used in the resource allocation and VM allocation is given in Table 2.

Table 2. Criteria for cloudlet and resource allocation

Identifier	Criterion Name	Identifier	Heuristic Name
LLF	Least-Full First	FF	First Fit
		LF	Least-Full First
PAL	Percent Allocated	MF	Most-Full First
		NF	Next Fit
RAN	Random	RA	Random
		TP	Tag and Pack

As per Table 2, we have implemented our algorithm and evaluated the outcome by initiating and not initiating speculation algorithm.

5. Results and discussion

We have inferred the results of our proposed algorithm in various resource allocation phases like monitoring, prediction and allocation modules. The service time in each module is accumulated as the total service time of an application. Hence results of each phase is discussed as follows

5.1. Resource monitoring phase

The resource monitoring phase is induced by fair queuing model. The queuing model is estimated through arrival rate and the service rate. The size of the queue denotes the service rate of the queue. When the queue size larger for a time interval then it may leads to service degradation. The efficient load balancing algorithm is devised for reducing hotspot and number of live migration. The migration is the major criteria for load balancing but it may even cause the network overhead due to SAN storage transferring. Our algorithm provides (0,0) hotspot and migration as compared to other resource allocation algorithm since of applying speculation mechanism. The speculation algorithm is trained from its history of hotspot occurrences. From Figure 4 the outcome of load balancing algorithm is given as follows.

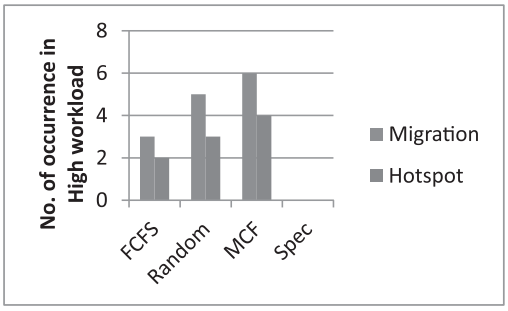

Fig. 4. Occurrences of Hotspot and Migration in synthetic workload of 200 jobs per min

5.2. Resource prediction phase

Many resource prediction algorithms are proposed for quick instantiating resources for better response time and service time. The sample prediction algorithms are neural network and linear regression for dynamic resource allocation as per (Bataineh, 2012; Islam *et al.* 2012).

The expected service time for the provided workload is estimated by the efficient prediction of successive hosts for the request. The response time measures the outcome of robust instantiating the VM images to the host as a result of prediction algorithm. The response time of various prediction based machine learning algorithm is evaluated with our algorithm and the results are in favor of our work as per Figure 5.

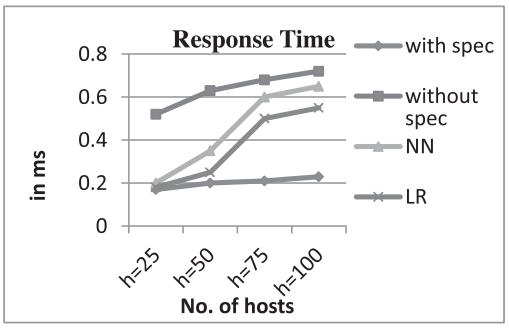

Fig. 5. Response time of jobs with varying hosts

The resource utilization of the jobs is considered under various workload. The resources must be efficiently utilized for the beneficiary of resource providers. Our algorithm efficiently predicts the resource pattern for the expected load through the pattern table entry. The pattern table records the best resource patterns under various time series and requests size. The resource utilization results are shown in Figure 6.

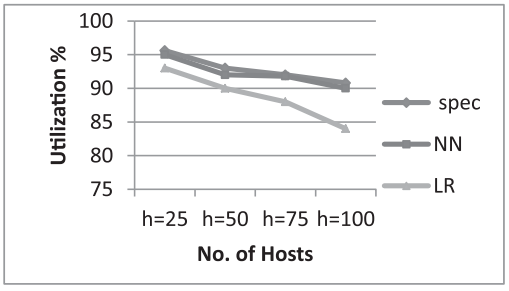

Fig. 6. Resource utilization vs No. of hosts

5.3. Resource Allocation phase

The outcome of the prediction module is the expected service rate of the job and the resources to be provided for the efficient service time and the response time. There are many resource allocation algorithms are there especially in cloud simulation environment the resources are provided based on time and space shared events. The service time of the random sampled jobs are chosen and compared with traditional resource allocation algorithm Hussain *et al.* (2013) with our proposed algorithm is shown in Figure 7.

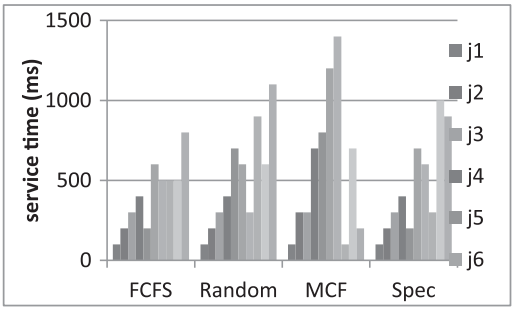

Fig.7. Mean service time of sampled jobs

The resource allocation algorithm is estimated by prediction accuracy rate. The prediction accuracy is measured by the difference between the expected and actual outcome. The successive decision taken for mapping the right VM image to the hosts results the prediction accuracy. In our work the prediction accuracy is measured for the sampled jobs and comparing the service time with other resource provisioning algorithm. We have achieved 84% of prediction accuracy rate and further analyzing factor is to be considered for improving this accuracy in future. The prediction Accuracy rate of our work is given in the Figure 8.

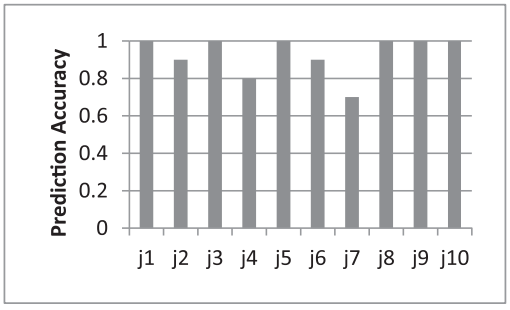

Fig. 8. Prediction accuracy of sampled jobs

6. Conclusion

This paper provides an evolutionary approach to constructing an adaptive resource provisioning in the cloud in order to facilitate dynamic and proactive resource management, scheduling and capacity planning for interactive web service applications, where immediacy and responsiveness are vitally important. Throughout the study, we have evaluated our speculative approach with several major machine learning algorithms, with a view to provide accurate forecasting ahead of time.

In order to ensure the real time environment we have chosen the resultant data set of resource monitoring project CoMon and online simulated book shopping web application TPC-App benchmarking tool. We also provided estimation for validating the accuracy of the proposed methodology. The integration of prediction strategies mentioned in this paper with the auto - scaling process will certainly enhance the effectiveness of adaptive resource allocation procedure in the cloud in terms of both performance and cost.

References

Aljazzaf, Z.M. (2015). Modeling and measuring the quality of online services. Kuwait Journal of Science, **42**(3):134-157.

Bataineh, M.H. (2012). Artificial neural network for studying human performance, MSc. Thesis, The University of IOWA, IOWA, USA.

Cao, J., Li, K. & Stojmenovic, I. (2014). Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers. IEEE Transactions on Computers, **63**(1):45-58.

Caron, E., Desprez, F. & Muresan, A. (2010). Forecasting for cloud computing on-demand resources based on pattern matching (Doctoral Dissertation, INRIA).

García, D.F., García, J., García, M., Peteira, I., García, R. & Valledor, P. (2006). Benchmarking of web services platforms. In Proceedings of 2nd International Conference on Web Information Systems and Technologies, WEBIST, Setubal, Portugal (pp.75-80).

Garg, S.K. & Buyya, R. (2011). Networkcloudsim: Modelling parallel applications in cloud simulations. In Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference, pp. 105-113. IEEE, Victoria, NSW.

Hussain, H., Malik, S.U.R., Hameed, A., Khan, S.U., Bickler, G. *et al*. (2013). A survey on resource allocation in high performance distributed computing systems. Parallel Computing, **39**(11):709-736.

Hu, W., Hicks, A., Zhang, L., Dow, E.M., Soni, V., *et al.* (2013). A quantitative study of virtual machine live migration. In Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference, p. 11. ACM.

Huang, D., He, B. & Miao, C. (2014). A survey of resource management in multi-tier web applications. IEEE Communications Surveys & Tutorials, **16**(3):1574-1590

Islam, S., Keung, J., Lee, K. & Liu, A. (2012). Empirical prediction models for adaptive resource provisioning in the cloud. Future Generation Computer Systems, **28**(1):155-162.

Jamshidi, P., Ahmad, A. & Pahl, C. (2013). Cloud migration research: a systematic review. IEEE Transactions on Cloud Computing, **1**(2):142-157.

Li, A., Yang, X., Kandula, S. & Zhang, M., (2011). Comparing public cloud providers. IEEE Internet Computing, **15**(2):50.

Samimi, P., Teimouri, Y., & Mukhtar, M. (2014). A combinatorial double auction resource allocation model in cloud computing. Information Sciences. DOI: 10.1016/j.ins.2014.02.008.

Tuah, N.J., Kumar, M. & Venkatesh, S. (2003). Resource-aware speculative prefetching in wireless networks. Wireless Networks, **9**(1):61-72.

Urgaonkar, R., Kozat, U.C., Igarashi, K. & Neely, M.J. (2010). Dynamic resource allocation and power management in virtualized data centers. 2010 IEEE Network Operations and Management Symposium-NOMS, pp. 479-486. IEEE.

Voorsluys, W., Broberg, J., Venugopal, S. & Buyya, R. (2009). Cost of virtual machine live migration in clouds: A performance evaluation. IEEE International Conference on Cloud Computing, pp. 254-265. Springer Berlin Heidelberg.

Zhuang, H., Liu, X., Ou, Z. & Aberer, K. (2013). Impact of instance seeking strategies on resource allocation in cloud data centers. IEEE CLOUD, pp. 27-34.

Submitted **:** 20/02/2015
Revised **:** 13/10/2015
Accepted **:** 13/10/2015

المضاربة في توفير الموارد للحوسبة عالية الأداء

لينا سري، [1] *،[2] بلاجي نارايانان

[1]قسم تقنية المعلومات، كلية ثياجاراجار للهندسة، مادوراي-625015، الهند.

[2]قسم تقنية المعلومات، كلية .K.L.N للهندسة، مادوراي، الهند.

*المؤلف المراسل: balaji_jet@yahoo.com

خلاصة

الحوسبة الموزعة تعطي دعم للمشترين لتقليل الأسس الداخلية، وللموردين لتوسيع الدخل، وذلك باستخدام إطار خاص بكل مستخدم. الموازنة السليمة للتحميل وتوفير الموارد ديناميكيا يحسن أداء السحابة ويجذب المستخدمين للسحابة. في هذه الورقة، نقترح خوارزمية لتوفير الموارد الآلي، توفير الموارد بطريقة المضاربة، وموازنة التحميل من خلال نهج المضاربة في توفير الموارد. وفي محاولة لقياس تخصيص الموارد نستخدم آلية ذات مستويين للتنبؤ والتحقق من أنماط تخصيص الموارد في الماضي وملائمتها لمتطلبات المستقبل. ويضمن هذا الإطار الموارد المناسبة اللازمة للتطبيقات، من خلال المرونة في المخصصات ودعم كفاءة استخدام الطاقة في تخصيص الموارد. نحن نستخدم منهجية تقديرية لمعالجة التباين في البيانات التاريخية لتحقيق التوازن بين النفقات الإضافية للمضاربة. وقد تم نقل العمل المقترح لدينا في بنية سحابة مفتوحة المصدر ومقارنة النتائج بآلية أخرى تستخدم نهج تعلم الألة. تظهر لنا النتائج التجريبية أن تخصيص الموارد يتكيف مع خدمة العملاء في إطار المتطلبات سريعة التغير في الحوسبة السحابية.

الكلمات المفتاحية: الحوسبة السحابية. توفير الموارد الديناميكي. تخصيص الموارد بكفاءة الطاقة؛ توفيرالموارد عن طريق التكهنات.