Analysis of Various Applications Proposed for Maintaining Scalability of Resources in Cloud Computing

Dr. Amit Kumar Chaturvedi
Assistant Prof., MCA Deptt.
Govt. Engineering College, Ajmer, India

Malay Upadhyay
M.Tech. Scholar, Department of Computer Science Engineering
Bhagwant University, Ajmer India

Abstract- Among the various qualities of cloud computing, scalability of resources is very important feature of cloud computing. Virtual Machine (VM) technology enables multiple VMs to share resources on the same host. Resources allocated to the VMs should be re-configured dynamically in response to the change of application demands or resource supply. Because VM execution involves privileged domain and VM monitor, this causes uncertainties in VMs' resource to performance mapping and poses challenges in online determination of appropriate VM configurations.

Dynamic Resource Scaling has issue of low resource scaling; where resources are allocating but they lack the actual allocation requirement i.e. lack of resources many times is the problem in this case. Another aspect is high resource allocation in this analysis where resources are allocated to more than the requirement of users. Here in this case, the wastage of resources is the problem found. Third issue is the internet speed issue that is life line of any internet services so as in multi-tenant cloud. Internet speed is widely affecting internet as whole multi-tenant cloud environment. Any kind of service is possible but when we place whole organizations, educational institutions, and business online services round the clock 24x7 any kind of obstacles to speed of internet will lead to big risk to whole organization or business.

Keywords – cloud computing, Virtual Machine, Resource Scaling,.

I. INTRODUCTION

Cloud computing is commonly associated to offering of new mechanisms for infrastructure provisioning. The illusion of a virtually infinite computing infrastructure, the employment of advanced billing mechanisms allowing for a pay-per-use model on shared multitenant resources, the simplified programming mechanisms (platform), etc. are some of the most relevant features. Among the various qualities of cloud computing, scalability of resources is very important feature of cloud computing. Virtual Machine (VM) technology enables multiple VMs to share resources on the same host. Resources allocated to the VMs should be re-configured dynamically in response to the change of application demands or resource supply. Because VM execution involves privileged domain and VM monitor, this causes uncertainties in VMs' resource to performance mapping and poses challenges in online determination of appropriate VM configurations. Various applications have been proposed for maintaining this scalability of resources.

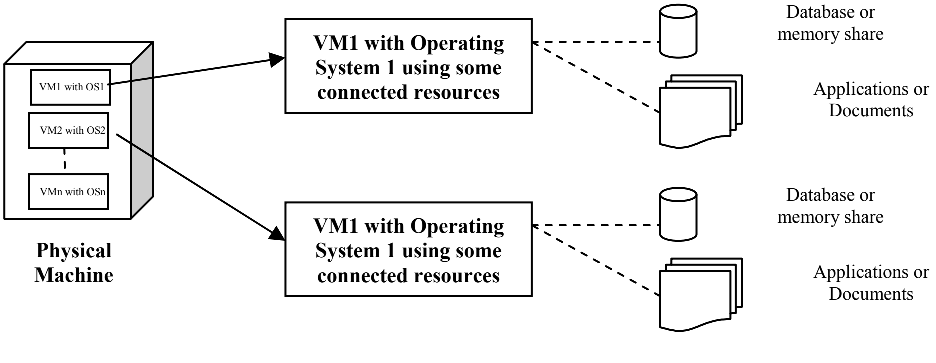

VM1 with OS1
VM2 with OS2
VMn with OSn
Physical Machine
VM1 with Operating System 1 using some connected resources
Database or memory share
Applications or Documents
VM1 with Operating System 1 using some connected resources
Database or memory share
Applications or Documents

Figure 1 : One Physical Machine Have Multiple VMs

In figure 1 it is illustrated that a physical machine containing multiple VMs (Virtual Machines), each VM have different operating system and using some resources of the physical machine.

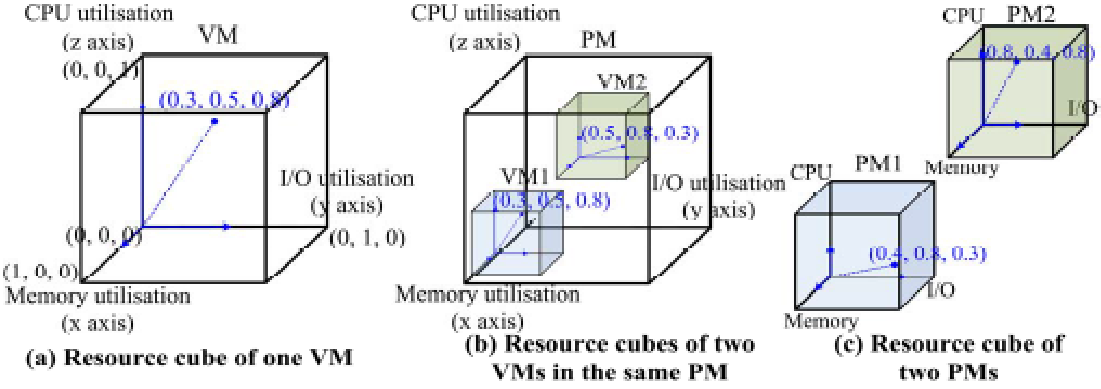
II. RESOURCE SCALING

Although scaling of traditional applications on physical servers shares many commonalities with that of cloud applications, scaling in these two environments has different emphases. Conventional techniques mainly focused on how to schedule compute nodes to meet QoS requirement of applications by predicting their long-term demand changes. In contrast, clouds focus on providing metered resources on demand and on quickly scaling applications up and down whenever the user demand changes.

Most IaaS cloud providers (e.g., Amazon AWS [1]) and vendors (e.g., RightScale [3]) assist application owners to manage cloud infrastructure and employ pre-defined policies (rules) to guide application scaling. For example, vendors like RightScale require application owners to manually specify scaling rules after an application is deployed. These rules specify the upper and lower bounds of the number of servers, the conditions to trigger scaling and the number of changed servers in each scaling. The policy-based mechanism assumes that the application owners have particular knowledge of the application being executed so that they can define proper policies. However, this is not always the case. In addition, a variety of scaling approaches are proposed based on analytical models of the application. In [4], Bacigalupo et al. model an application by a three-tier queueing model, namely application, database and storage disk tiers. Each tier is solved to analyse the servers' mean response time and throughput.

A scaling algorithm is then proposed using these analysis results. Similar to [11, 12], researchers in [5] break down an application's end-to-end response time by tier. They then calculate the number of servers to be allocated to each tier in order to meet the response time target. In addition, a method is presented to support the scaling up of a two-tier web application by actively profiling the VMs' CPU usage. The method also realises scaling down using a regression-model-based predictive mechanism [6]. In [7], a network flow model is introduced to analyse applications to assist application owners in making a trade-off between cost and QoS. In [13], Gong et al. propose a scaling approach that conducts accurate resource allocation using two complementary techniques: patterns-driven (they call repeating patterns as signatures) and state-driven (they use discrete-time Markov chain) demand predictions. In [14], Shen et al. further improve the prediction mechanism by introducing two handling schemas, namely online adaptive padding and reactive error correction.

In cloud, multiple Virtual Machines (VM's) may be on the same Physical Machine (PM) or every Virtual Machine (VM) may have a individual Physical Machine (PM). The physical resources like CPU, Memory, and I/O are shared / used accordingly. This scenario is presented in the figure 2.



(a) Resource cube of one VM

(b) Resource cubes of two VMs in the same PM

(c) Resource cube of two PMs

Figure 2: Resource allocation to VM in PM

In figure 2 (a) one VM is used in PM (server), in this case the resources like CPU, Memory, and I/O are underutilization and in figure 2 (b) two VMs are installed in the same PM (server), the resources in this case are utilized in a better way because when the resource when idle and not in use with one VM they are utilized by the other VM and this increased the utilization of resources. In the third case as shown in figure 2(c), every VM has a different PM (server) and the resources are also individual.

III. RELATED WORK

R. Han, L. Guo, M. M. Ghanem, Y Guo proposed an innovative "Lightweight Resource Scaling for Cloud Applications". They describes that Elastic resource provisioning is a key feature of cloud computing, allowing users to scale up or down resource allocation for their applications at run-time. To date, most practical approaches to managing elasticity are based on allocation/de-allocation of the virtual machine (VM) instances to the application. This VM-level elasticity typically incurs both considerable overhead and extra costs, especially for applications with rapidly fluctuating demands. In this paper, we propose a lightweight approach to enable cost-effective elasticity for cloud applications. Our approach operates fine grained scaling at the resource level itself (CPUs, memory, I/O, etc) in addition to VM-level scaling. We also present the design and implementation of an intelligent platform for light-weight resource management of cloud applications. We describe our algorithms for light-weight scaling and VM-level scaling and show their interaction. We then use an industry standard benchmark to evaluate the effectiveness of our approach and compare its performance against traditional approaches.

H. Nguyen, Z. Shen, X. Gu stated in their paper "AGILE: elastic distributed resource scaling for Infrastructure-as-a-Service" that Dynamically adjusting the number of virtual machines (VMs) assigned to a cloud application to keep up with load changes and interference from other uses typically requires detailed application knowledge and an ability to know the future, neither of which are readily available to infrastructure service providers or application owners. The result is that systems need to be over-provisioned (costly), or risk missing their performance Service Level Objectives (SLOs) and have to pay penalties (also costly). AGILE deals with both issues: it uses wavelets to provide a medium-term resource demand prediction with enough lead time to start up new application server instances before performance falls short, and it uses dynamic VMcloning to reduce application startup times. Tests using RUBiS and Google cluster traces show that AGILE can predict varying resource demands over the medium-term with up to 3.42× better true positive rate and 0.34× the false positive rate than existing schemes. Given a target SLO violation rate, AGILE can efficiently handle dynamic application workloads, reducing both penalties and user dissatisfaction.

J. Oriol Fit´o, ´I ˜nigo Goiri and Jordi Guitart proposes "SLA-driven Elastic Cloud Hosting Provider". In this paper they declare that It is clear that Cloud computing is and will be a sea change for the Information Technology by changing the way in which both software and hardware are designed and purchased. In this work we address the use of this emerging computing paradigm into web hosting providers in order to avoid its resource management limitations. Thanks to the Cloud approach, resources can be provided in a dynamic way according with the needs of providers and end-users. In this paper, we present an elastic web hosting provider, namely Cloud Hosting Provider (CHP), that makes use of the outsourcing technique in order to take advantage of Cloud computing infrastructures for providing scalability and high availability capabilities to the web applications deployed on it. Furthermore, we pursue the main goal of maximizing the revenue earned by the provider through both the analysis of Service Level Agreements (SLA) and the employment of an economic model. The evaluation exposed demonstrates that the system proposed is able to properly react to the dynamic load received by the web applications and it also achieve the aforesaid revenue maximization of the provider by performing an SLA-aware resource (i.e. web servers) management.

A. Andrzejak, D. Kondo, S. Yi gives an innovative "Decision Model for Cloud Computing under SLA Constraints". With the recent introduction of Spot Instances in the Amazon Elastic Compute Cloud (EC2), users can bid for resources and thus control the balance of reliability versus monetary costs. A critical challenge is to determine bid prices that minimize monetary costs for a user while meeting Service Level Agreement (SLA) constraints (for example, sufficient resource availability to complete a computation within a desired deadline). We propose a probabilistic model for the optimization of monetary costs, performance, and reliability, given user and application requirements and dynamic conditions. Using real instance price traces and workload models, we evaluate our model and demonstrate how users should bid optimally on Spot Instances to reach different objectives with desired levels of confidence.

Z. Gong, X. Gu proposed “PAC: Pattern-driven Application Consolidation for Efficient Cloud Computing”. To reduce cloud system resource cost, application consolidation is a must. In this paper, we present a novel pattern driven application consolidation (PAC) system to achieve efficient resource sharing in virtualized cloud computing infrastructures. PAC employs signal processing techniques to dynamically discover significant patterns called signatures of different applications and hosts. PAC then performs dynamic application consolidation based on the extracted signatures. We have implemented a prototype of the PAC system on top of the Xen virtual machine platform and tested it on the NCSU Virtual Computing Lab. We have tested our system using RUBiS benchmarks, Hadoop data processing systems, and IBM System S stream processing system. Our experiments show that 1) PAC can efficiently discover repeating resource usage patterns in the tested applications; 2) Signatures can reduce resource prediction errors by 50-90% compared to traditional coarse-grained schemes; 3) PAC can improve application performance by up to 50% when running a large number of applications on a shared cluster.

M. Mao, J. Li, M. Humphrey introduced a new “Cloud Auto-scaling with Deadline and Budget Constraints”. Clouds have become an attractive computing platform which offers on-demand computing power and storage capacity. Its dynamic scalability enables users to quickly scale up and scale down underlying infrastructure in response to business volume, performance desire and other dynamic behaviors. However, challenges arise when considering computing instance non-deterministic acquisition time, multiple VM instance types, unique cloud billing models and user budget constraints. Planning enough computing resources for user desired performance with less cost, which can also automatically adapt to workload changes, is not a trivial problem. In this paper, we present a cloud auto-scaling mechanism to automatically scale computing instances based on workload information and performance desire. Our mechanism schedules VM instance startup and shut-down activities. It enables cloud applications to finish submitted jobs within the deadline by controlling underlying instance numbers and reduces user cost by choosing appropriate instance types. We have implemented our mechanism in Windows Azure platform, and evaluated it using both simulations and a real scientific cloud application. Results show that our cloud auto-scaling mechanism can meet user specified performance goal with less cost.
U. Sharma, P. Shenoy, S. Sahu and A. Shaikh proposed “A Cost-aware Elasticity Provisioning System for the Cloud”. In this paper we present Kingfisher, a cost-aware system that provides efficient support for elasticity in the cloud by (i) leveraging multiple mechanisms to reduce the time to transition to new configurations, and (ii) optimizing the selection of a virtual server configuration that minimizes the cost. We have implemented a prototype of Kingfisher and have evaluated its efficacy on a laboratory cloud platform. Our experiments with varying application workloads demonstrate that Kingfisher is able to (i) decrease the cost of virtual server resources by as much as 24% compared to the current cost unaware approach, (ii) reduce by an order of magnitude the time to transition to a new configuration through multiple elasticity mechanisms in the cloud, and (iii), illustrate the opportunity for design alternatives which trade-off the cost of server resources with the time required to scale the application.

A. Ali-Eldin, J. Tordsson and E. Elmroth described “An Adaptive Hybrid Elasticity Controller for Cloud Infrastructures”. This paper state that Abstract—Cloud elasticity is the ability of the cloud infrastructure to rapidly change the amount of resources allocated to a service in order to meet the actual varying demands on the service while enforcing SLAs. In this paper, we focus on horizontal elasticity, the ability of the infrastructure to add or remove virtual machines allocated to a service deployed in the cloud. We model a cloud service using queuing theory. Using that model we build two adaptive proactive controllers that estimate the future load on a service. We explore the different possible scenarios for deploying a proactive elasticity controller coupled with a reactive elasticity controller in the cloud. Using simulation with workload traces from the FIFA world-cup web servers, we show that a hybrid controller that incorporates a reactive controller for scale up coupled with our proactive controllers for scale down decisions reduces SLA violations by a factor of 2 to 10 compared to a regression based controller or a completely reactive controller.

M. Z. Hasan, E.Magana, A. Clemm, L. Tucker, S. Lakshmi, D. Gudreddi discussed “Integrated and Autonomic Cloud Resource Scaling” and state that A Cloud is a very dynamic environment where resources offered by a Cloud Service Provider (CSP), out of one or more Cloud Data Centers (DCs) are acquired or released (by an enterprise (tenant) on-demand and at any scale. Typically a tenant will use Cloud service interfaces to acquire or release resources directly. This process can be automated by a CSP by providing auto-scaling capability where a tenant sets policies indicating under what condition resources should be auto-scaled. This is specially needed in a Cloud environment because of the huge scale at which a Cloud operates. Typical solutions are naïve causing spurious auto-scaling decisions. For example, they are based on only thresholding triggers and the thresholding mechanisms themselves are not Cloud-ready. In a Cloud, resources from three separate domains, compute, storage and network, are acquired or released on-demand. But in typical solutions resources from these three domains are not auto-scaled in an integrated fashion. Integrated auto-scaling prevents further spurious scaling and reduces the number of auto-scaling systems to be supported in a Cloud management system. In addition, network resources typically are not auto-scaled. In this paper we describe a Cloud resource auto-scaling system that addresses and overcomes above limitations.

Z. Shen, S. Subbiah, X. Gu, J. Wilkes proposed “CloudScale: Elastic Resource Scaling for Multi-Tenant Cloud Systems”. Elastic resource scaling lets cloud systems meet application service level objectives (SLOs) with minimum resource provisioning costs. In this paper, we present CloudScale, a system that automates finegrained elastic resource scaling for multi-tenant cloud computing infrastructures. CloudScale employs online resource demand prediction and prediction error handling to achieve adaptive resource allocation without assuming any prior knowledge about the applications running inside the cloud. CloudScale can resolve scaling conflicts between applications using migration, and integrates dynamic CPU voltage/frequency scaling to achieve energy savings with minimal effect on application SLOs. We have implemented CloudScale on top of Xen and conducted extensive experiments using a set of CPU and memory intensive applications (RUBiS, Hadoop, IBM System S). The results show that CloudScale can achieve significantly higher SLO conformance than other alternatives with low resource and energy cost. CloudScale is non-intrusive and light-weight, and imposes negligible overhead (< 2% CPU in Domain 0) to the virtualized computing cluster.

IV.CONCLUSION

After analyzing the various schemes and models proposed for resource scaling in cloud computing. We concluded that the resource scaling is important factor and cost aware elasticity in scaling of resources is best option to do the job. In this analysis, we find that resources scaling has issues of allocation in multi-tenant environment. There are various solutions proposed for resource scaling like Lightweight Resource Scaling for Cloud Applications by R.Han, L. Guo, M.M. Ghanem, SLA-driven Elastic Cloud Hosting Provider by J.Oriol Fit’o, Goiri and J. Guitart and Cloud Auto-scaling with Deadline and Budget Constraints by M. Mao, J. Li, M. Humphrey. These are the innovative approaches to solve the issue. Dynamic Resource Scaling has issue of low resource scaling; where resources are allocating but they lack the actual allocation requirement i.e. lack of resources many times is the problem in this case. Another aspect is high resource allocation in this analysis where resources are allocated to more than the requirement of users. Here in this case, the wastage of resources is the problem found. Third issue is the internet speed issue that is life line of any internet services so as in multi-tenant cloud. Internet speed is widely affecting internet as whole multi-tenant cloud environment. Any kind of service is possible but when we place whole organizations, educational institutions, and business online services round the clock 24x7 any kind of obstacles to speed of internet will lead to big risk to whole organization or business.

So, we proposed that researcher should focus on developing such solutions that will allocate the resources dynamically and on demand. The solutions should be so dynamic, if some resources are occupied for a long time and another resource requirement occurs. The provider should adjust the balance from the already booked resource this will be a most economical solution. So, future researcher may work in this direction to develop some solutions.

REFERENCES

[1] Abhishek Chandra, Weibo Gong, PrashantSheno.Dynamic Resource Allocation for Shared DataCentres Using Online Measurements 2003

[2] J. Chase, D. Anderson, P. N. Thakar, and A. M. Vahdat.

[3] Managing energy and server resources in hosting centers. In*Proc. SOSP*, 2001.

[4] X. Fan, W.-D.Weber, and L. A. Barroso. Power provisioningfor a warehouse-sized computer. In *Proc. ISCA*, 2007.

[5] 2008the System S declarative stream processing engine

[6] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. Capacitymanagement and demand prediction for next generation datacenters. In *Proc. ICWS*, 2007.

[7] E. Kalyvianaki, T. Charalambous, and S. Hand. Self-adaptiveand self-configured CPU resource provisioning forvirtualized servers using Kalman filters. In *Proc. ICAC*,2009.

[8] H. Lim, S. Babu, and J. Chase. Automated control for elasticstorage. In *Proc. ICAC*, 2010.

[9] Xiaoyun Zhu, Zhikui Wang, SharadSinghal Utility-driven workloadmanagement using nested control design. In *Proc. AmericanControl Conference*, 2006.

[10] B. Urgaonkar, M. S. G. Pacifici, P. J. Shenoy, and A. N.Tantawi. An analytical model for multi-tier internet servicesand its applications. In Proc. SIGMETRICS, 2005.

[11] Z. Gong, X. Gu, and J. Wilkes. PRESS: PRedictiveElasticReSource Scaling for Cloud Systems.In*Proc. CNSM*, 2010.

[12] M. Armbrust, A. Fox, D. A. Patterson, N. Lanham,B. Trushkowsky, J. Trutna, and H. Oh. Scads:Scale-independent storage for social computing applications.In Proc. CIDR, 2009.

[13] ZhimingShen, SethuramanSubbiah, XiaohuiGu, John Wilkes, CloudScale: Elastic Resource Scaling for Multi-Tenant Cloud Systems 2011

[14] Djamal Ziani1,configuration in erpsaas multi-tenancy,

[15] VenkatanathanVaradarajan†, Yinqian Zhang‡, Thomas Ristenpart_, and Michael Swift†,Placement Vulnerability Study in Multi-Tenant Public Clouds

[16] Jing Zhu_, Dan Li_*z*, Jianping Wu_, Hongnan Liu_, Ying Zhang*y*, JingchengZhang_*Towards Bandwidth Guarantee in Multi-tenancy Cloud Computing Networks*

[17] Twinkle Garg1, Rajender Kumar2, JagtarSinghA way to cloud computing basic to multitenant environment