Allocation of Cloud Resources in a Dynamic Way Using an SLA-Driven Approach

S. Anithakumari and K. Chandrasekaran

Abstract Cloud computing provides a wide access to complex applications running on virtualized hardware with its support for elastic resources that are available in an on-demand manner. In cloud environment, multiple users can request resources simultaneously and so it has to be made available to them in an efficient manner. For the efficient utilization, these computing resources can be dynamically configured according to varying workload. Here in this paper, we proposed an efficient resource management system to allocate elastic resources dynamically according to dynamic workload.

Keywords Cloud computing · Service level agreement (SLA) · Resource allocation · Local manager · Universal manager · Global utility value

1 Introduction

Cloud computing has evolved as the latest computing paradigm in which computing resources can be delivered to users as services through the network (Internet) and can be acquired by the users on demand. Using cloud computing, users can dynamically use computing resources from cloud providers and they have to pay only for what they have used [1]. Usage of these cloud resources can reduce the problem of over-investment and the maintenance cost in large IT industries. Cloud technology can help industries to manage their own pool of computing resources in an efficient manner.

Computing resources are often provisioned using the virtualization technology where computing power, storage and network are encapsulated into a virtual machine (VM). Each data centre is equipped with a large set of virtual machines(VMs) built

S. Anithakumari (✉) · K. Chandrasekaran
NITK Surathkal, Karnataka, India
e-mail: lekshmi03@gmail.com

K. Chandrasekaran
e-mail: kchnitk@gmail.com

A. J. Kulkarni et al. (eds.), *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, Advances in Intelligent Systems and Computing 828, https://doi.org/10.1007/978-981-13-1610-4_42

on physical machines and has the flexibility to configure the VMs according to the diverse requirements and user applications. Multiple applications can be processed separately on dedicated VMs to reduce any conflicts that might happen when an application shares the resources with another application running on the same physical machine. Virtual machines can be migrated from one application environment to another according to the changes in the users' demand. So, we need an efficient system that allows the remote resources to be joined and used as if they were normal resources in a data centre.

Virtualization technology permits the provisioning of physical resources to applications, as the application receives the maximum capacity allotted to it, without considering the workloads generated by other applications. Virtualization helps to implement elasticity of resources by providing the flexibility of expanding or condensing the quantum of computing resources [2]. Here, we define a resource allocation decision to find out the quantum of resource capacity each VM is getting from the corresponding physical machine. This decision algorithm is explained in Sect. 4.1.

2 Related Work

The system model for on-demand provisioning of computing resources to address varying workload has been discussed in many literature. VioCluster [3] and dynamic virtual clustering [4] are such architectures proposed to borrow available resources from nearby sites. Vazquez et al. [5] proposed an architecture to extend grid infrastructure into cloud by using the GridWay meta-scheduler and different resource adapters. Murphy et al. [6] developed a dynamic provisioning system on a shared physical resource pool with Condor job scheduler. Assuncao et al. [7] developed a system which allows a user to take virtual machines from both local resource pool and cloud data centre for processing an application. Silva et al. [8] addressed the problem of finding the optimal number of virtual machines that should be provisioned to maximize the speedup under a given budget. Their proposed heuristic tries to fully utilize CPU time of virtual machines and avoid loss in the one-hour charging scheme used in Amazon EC2. The cost-based scheduling and provisioning [9, 10] policy has been discussed here.

3 Cloud Provider's Data Centre

Cloud computing environment is generally viewed as a multi-layer arrangement containing different layers such as cloud provider layer, cloud user layer and end-user layer (as shown in Fig. 1). Cloud provider layer describes the infrastructure arrangement and server organization at the cloud provider's data centre. End-user layer includes the end users, and the cloud user layer describes the interface between

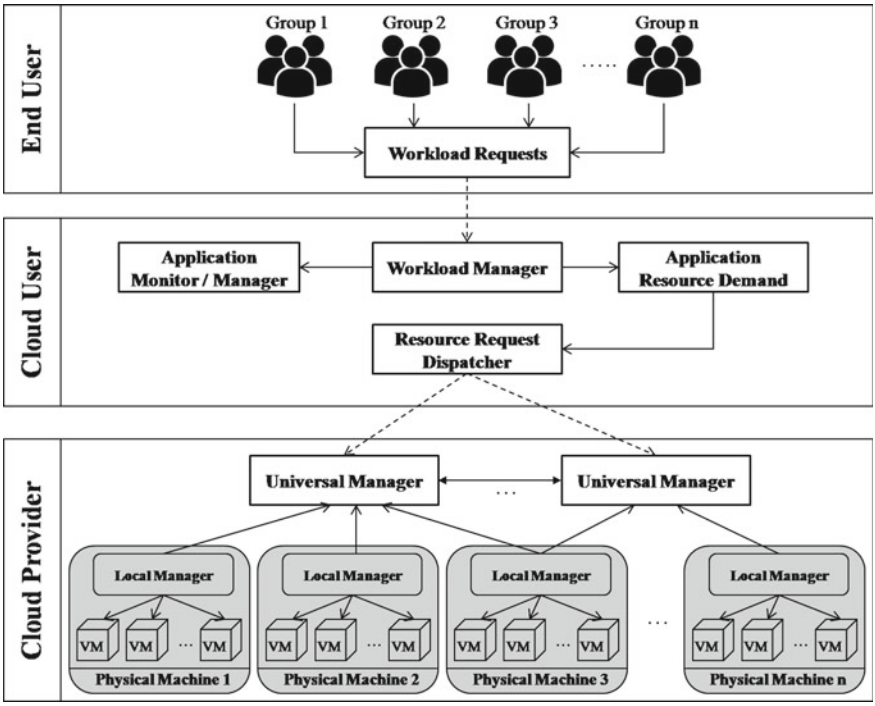

Fig. 1 Overall architecture of a data centre

cloud provider layer and end-user layer. For addressing the resource allocation problem in a dynamic way, we make use of the infrastructure organization at the cloud provider layer. The cloud provider layer contains cloud providers who provide multiple computing resources as services through a shared data centre. These computing resources are able to provide performance isolation and efficient resource sharing through virtualization technology, as per the basic feature of cloud computing. The creation of multiple virtual images from the available physical resources and the maintenance of these virtual images are done by an intermediate virtualization layer. This layer generates multiple virtual machines, on top of the physical infrastructure layer, which are isolated from one another and are capable of serving individual applications. So, these applications are also isolated like running on a dedicated machine and it uses a fraction of the entire resource capacity.

In order to proceed with the mathematical calculations, we assume a standard structure for cloud provider's data centre such as: The data centre contains a total of P physical servers, and the maximum possible VMs that can be created by all servers is taken as V. The set of applications processed by ith server is taken as A_i, and the number of VMs created on ith server is taken as n_i. That is, $\sum_{i=1}^{P} n_i = V$.

The system model for dynamic resource allocation and VM management within a single server machine is discussed in Sect. 4, and the model for global resource management, by considering all the servers in the data centre, is explored in our next paper with complete experimental analysis.

4 System Model for Adaptive VM Management

This section discusses the dynamic VM management in a single server machine by considering the server machines in a cloud provider's data centre where each VM is viewed as a single computing machine and makes use of an admission control policy [11, 12] for allotting or dropping incoming requests. As per the admission control policy, the VMs may drop some of the coming requests, because of the limitations in capacity and excess count in incoming requests. This control policy helps to address the remaining requests without affecting the assured QoS guarantees [13, 14].

The system model assumed for adaptive VM management in a single server machine is shown in Fig. 2. The major component in this model is the *allocation management* module which is to take care of all incoming workload requests and to service these requests by considering system characteristics, application's properties, VM availability and SLA contracts for maximizing the revenue of the service provider. The *allocation management* module is configured with quantitative measures of the application and SLA metrics. These measures are updated according to application change or SLA change.

The *requested workload* module is for monitoring and reading the workload requirement of each application. It contains provision to keep track of all processing applications, and accordingly it predicts the workload requirement of the current scenario. This input is forwarded to the *allocation management*. The *allocation management* decides on the allocation decision and in consultation with the *middleware control*, initiates the virtual resource mappings and generates VMs in the *virtualization layer*. The admission control policies are also taken care before the allocation of the VM images. The *allocation management* decision is based on an optimized performance model which considers SLA parameters and workload conditions of the processing applications.

The adaptive resource allocation is implemented by making frequent resource allocation decisions. The interval between two adjacent decision computing is viewed as *decision interval*, and this can be taken as a fixed value or variable according to the characteristics of the system. By choosing a smaller value for decision interval, we can make a more accurate resource allocation in a single server system. The major component in the system model is *allocation management* module because this is the module responsible for making resource allocation decision. The resource allocation decision is determined based on an optimization model by considering performance and efficiency values.

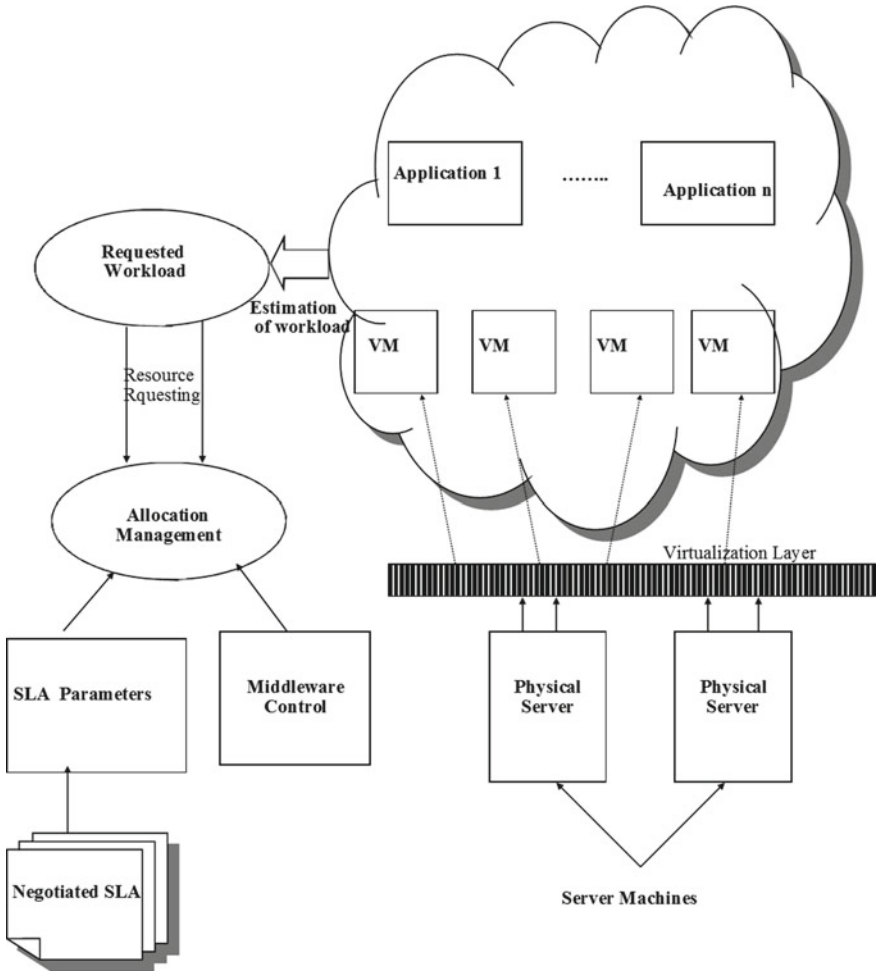

Fig. 2 VM management in a single server

4.1 Resource Allocation Decision

In our analytical system model, the resource allocation is controlled by SLA parameters and some system parameters. The estimated measures calculated from SLA parameters and applications' workload also play a role in this decision-making. The service level efficiency in cloud computing is very much dependent on SLA parameters, and so the SLA performance has related to the VM's ability to service applications by satisfying the application's response time specified in the agreement. The SLA parameters we mainly focusing are throughput(T^{TH}), response time threshold (R^{TH}), probability values(P()), and the system parameters are the total number of

VMs(V) created by the virtualization layer, the utilization value(u) of each VM and the service time average(S) of the application on a physical server. Among these values, the utilization value we are seeing as the maximum possible value provided by the service provider and the throughput value is the throughput maximum limit or throughput threshold(T^{TH}).

The *allocation management* component gets an estimate a_i (arrival rate of requests) from the *requested workload* component for each application during the next controller interval. If there occur some deviations in arrival rate from the estimated value, then a load optimizer unit is initiated and the deviations are optimized with the optimizer unit. Here, a_i is the rate of arrival requests over the considered controller interval. From these arrived requests, some may be rejected because of resource limitations and so the actual arrival rate becomes lesser than a_i. Among the processed set of requests, some may violate the agreed response time values and so they are not taken for the calculation of actual throughput, T_i.

In case of fixed controller intervals, the events which are considerably smaller than the controller interval could mislead the allocation manager such as bulk quantum of requests (with less duration), coming from some applications can stop the allocation of resources to some other class of applications because of deficiency in resource availability and this will lead to heavy penalties to the provider. For minimizing this undesired effect, the *requested workload component* provides the estimated probability (P_i) of a class of requests having higher arrival rate for the next controller interval. The parameter P_i is to represent the certainty level to assure maximum profit to the provider for VM_i, and it can be bypassed, to consider workload changes, by assigning a value 1.

To proceed with the analytical model, we assume that the application coming from a user is a unique entity and is submitted to particular VM. That is application coming from customer i is submitted to VM_i which is serviced in a mean service time S_i, and the utilization value upper limit of VM_i is u_i. In this model, each VM is eligible for a guaranteed fraction of available physical server and so we estimate the average service time by taking f_i, the fraction of service time given to VM_i, as S_i/f_i. Correspondingly, resource allocation decision is the decision-making of allocation fractions f_i $(i = 1, 2 \ldots V)$ given to each VM_i. So, f_i is viewed as the important decision variable in our resource allocation problem and analytical system model.

5 Experimental Analysis

We have conducted the experimental studies in an environment simulated by arena. The quantitative results are taken from two different VMs which are running on top of the same physical infrastructure. The experimentation of the proposed resource allocation model is done with a conservative admission control policy using tokens. That is, a fixed number of tokens are ready for a fixed slot and the transactions need to acquire these tokens for getting into the system.

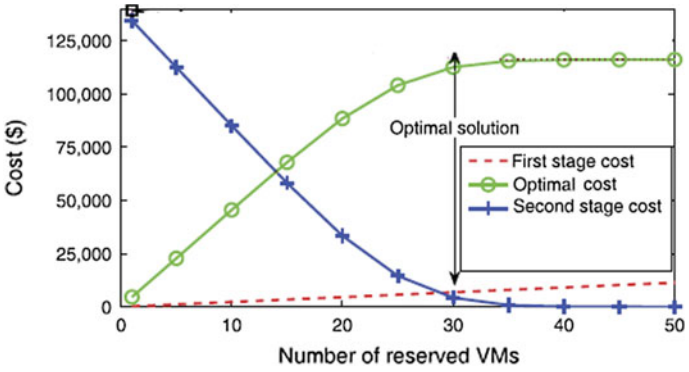

Fig. 3 VM management

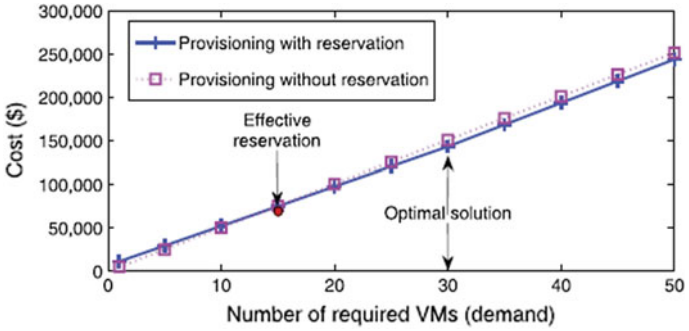

Fig. 4 VM management with reservation

The optimal allocation of created virtual machines according to varying workload is shown in Fig. 3, and the comparison between provisioning of resources within a single physical machine and a global system is illustrated in Fig. 4. The complete experimentation results are included in our next paper because of space restrictions.

6 Conclusion

We introduced a resource allocation system that is aware of the varying resource requirement in each application and adapt the system in a dynamic way. The proposed system is ready to self-organize resource provisioning according to the changing demand. We take into account both high-level performance goals of the hosted applications and objectives related to the placement of virtual machines on physical machines. An efficient way to express and quantify application satisfaction with

regard to SLA is provided by using utility functions and is seen as a balanced trade-off between multiple objectives which might conflict with each other. The complete experimentation results are included in our second paper with all experimental investigations.

References

1. Ali S, Jing S-Y, Kun S (2013) Profit-aware dvfs enabled resource management of iaas cloud. Int J Comput Sci Issues (IJCSI) 10:237
2. Padala P, Shin KG, Zhu X, Uysal M, Wang Z, Singhal S, Merchant A, Salem K (2007) Adaptive control of virtualized resources in utility computing environments. In: ACM SIGOPS operating systems review, vol 41, no 3. ACM, 2007, pp 289–302
3. Ruth P, McGachey P, Xu D (2005) Viocluster: virtualization for dynamic computational domains. In: IEEE international cluster computing, IEEE pp 1–10
4. Emeneker W, Stanzione D (2007) Dynamic virtual clustering. In: IEEE international conference on cluster computing (2007). IEEE pp 84–90
5. Blanco CV, Huedo E, Montero RS, Llorente IM (2009) Dynamic provision of computing resources from grid infrastructures and cloud providers. In: Grid and pervasive computing conference, (2009) GPC'09. Workshops at the. IEEE pp 113–120
6. Murphy MA, Kagey B, Fenn M, Goasguen S (2009) Dynamic provisioning of virtual organization clusters. In: Proceedings of the 2009 9th IEEE/ACM international symposium on cluster computing and the grid. IEEE computer society, pp 364–371
7. De Assunção MD, Di Costanzo A, Buyya R (2009) Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters. In: Proceedings of the 18th ACM international symposium on high performance distributed computing. ACM, pp 141–150
8. Silva JN, Veiga L, Ferreira P (2008) Heuristic for resources allocation on utility computing infrastructures. In: Proceedings of the 6th international workshop on middleware for grid computing. ACM, p 9
9. Zhang L, Ardagna D (2004) Sla based profit optimization in web systems. In: Proceedings of the 13th international world wide web conference on alternate track papers & posters. ACM, pp 462–463
10. Salehi MA, Buyya R (2010) Adapting market-oriented scheduling policies for cloud computing. In: International conference on algorithms and architectures for parallel processing. Springer, pp 351–362
11. Perros HG, Elsayed KM (1996) Call admission control schemes: a review. IEEE Commun Mag 34(11):82–91
12. Kleinrock L (1975) Queuing systems. Wiley
13. Menasce DA, Almeida VA, Dowdy LW, Dowdy L (2004) Performance by design: computer capacity planning by example. Prentice Hall Professional
14. Papoulis A, Pillai SU (2002) Probability, random variables, and stochastic processes. Tata McGraw-Hill Education