ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

# A Hierarchical Approach for the Resource Management of Very Large Cloud Platforms

R.Kavitha

PG Scholar, Karpagam University, Coimbatore, Tamilnadu, India[1]

**ABSTRACT:** Worldwide interest in the delivery of computing and storage capacity as a service continues to grow at a rapid pace. The complexities of such cloud computing centers require advanced resource management solutions that are capable of dynamically adapting the cloud platform while providing continuous service and performance guarantees. The goal of this paper is to devise resource allocation policies for virtualized cloud environments that satisfy performance and availability guarantees and minimize energy costs in very large cloud service centers. We present a scalable distributed hierarchical framework based on a mixed-integer non-linear optimization of resource management acting at multiple time-scales. Extensive experiments across a wide variety of configurations demonstrate the efficiency and effectiveness of our approach.

Category C.4: Performance attributes; Performance of Systems; Quality concepts.

**KEYWORDS:** Optimization of Cloud Configurations; QoS-based scheduling and load balancing; Virtualized system QoS-based migration policies.

## I. INTRODUCTION

Modern cloud computing systems operate in a new and dynamic world, characterized by continual changes in the environment and in the system and performance requirements that must be satisfied. Continuous changes occur without warning and in an unpredictable manner, which are outside the control of the cloud provider. Therefore, advanced solutions need to be developed that manage the cloud system in a dynamically adaptive fashion, while continuously providing service and performance guarantees: (i) reduce costs, (ii) improve levels of performance, and (iii) enhance availability and dependability. Within this framework, it should first be noted that, at large service centers, the number of servers are growing significantly and the complexity of the network infrastructure is also increasing. This leads to an enormous spike in electricity consumption: IT analysts predict that by the end of 2012, up to 40% of the budgets of cloud service centers will be devoted to energy costs.

Energy efficiency is therefore one of the main focal points on which resource management should be concerned.In addition, providers need to comply with Service Level Agreement (SLA) contracts that determine the revenues gained and penalties incurred on the basis of the level of performance achieved. Quality of Service (QoS) guarantees have to be satisfied despite workload fluctuations, which could span several orders of magnitude within the same business day . Moreover, if end-user data and applications are moved to the cloud, infrastructures need to be as reliable as phone systems : Although 99.9% up-time is advertised, several hours of system outages have been recently experienced by some cloud market leaders (e.g., the Amazon EC2 2011 Easter outage and the Microsoft Azure outage on 29th February 2012. Currently, Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) providers include in SLA contracts only availability, while performance are neglected. In case of availability violations (with current figures, even for large providers, being around 96%, much lower than the values stated in their contracts ), customers are refunded with credits to use the cloud system for free. The concept of virtualization, an enabling technology that allows sharing of the same physical machine by multiple end-user applications with performance guarantees, is fundamental to developing appropriate policies for the management of current cloud systems. From a technological perspective, the consolidation of multiple user workloads on the same physical machine helps to reduce costs, but this also translates into higher utilization of the physical resources. Hence, unforeseen load fluctuations or hardware failures can have a greater impact among multiple applications, making a cost efficient, dependable cloud infrastructures with QoS guarantees of paramount importance in order to realize broad adoption of cloud systems. A number of self-managing

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

solutions have been developed that support dynamic allocation of resources among running applications on the basis of workload predictions. Most of these approaches have implemented centralized solutions: A central controller manages all resources comprising the infrastructure, usually acting on a hourly basis. This time-scale is often adopted [8] because it represents a compromise in the trade-off between the frequency of decisions and the overhead of these decisions. For example, self-managing frameworks may decide to shut-down or power up a server, or to move virtual machines (VMs) from one server to another by exploiting the VM live-migration mechanism. However, these actions consume a significant amount of energy and cannot be performed too frequently. As previously noted, cloud systems are continuously growing in terms of size and complexity: Today, cloud service centers include up to 10,000 servers , and each server hosts several VMs. In this context, centralized solutions are subject to critical design limitations , including a lack of scalability and expensive monitoring communication costs, and cannot provide effective control within a one hour time horizon. In this paper we take the perspective of a PaaS provider that manages the transactional service applications of its customers to satisfy response time and availability guarantees and minimize energy costs in very large cloud service centers. We propose a distributed hierarchical framework based on a mixed-integer non-linear optimization of resource management across multiple time-scales. At the root of the hierarchy, a Central Manager (CM) partitions the workload (applications) and resources (physical servers) over a 24-hour horizon to obtain clusters with homogeneous workload profiles one day in advance. For each cluster, an Application Manager (AM) determines the optimal resource allocation in a centralized manner across a subsystem of the cloud platform on a hourly basis. More precisely, AMs can decide the set of applications executed by each server of the cluster (i.e., application placement), the request volumes at various servers inside the cluster (i.e., load balancing), and the capacity devoted to the execution of each application at each server (i.e., capacity allocation). AMs can also decide to switch servers into a low-power sleep state depending on the cluster load (i.e., server switching) or to reduce the operational frequency of servers (i.e., frequency scaling). Furthermore, since load balancing, capacity allocation, and frequency scaling require a small amount of time to be performed, they are executed by AMs every few (5-15) minutes. The periodicity of the placement decisions is investigated within the experimental analysis in §VI. Extensive experiments across a wide variety of configurations demonstrate the effectiveness of our approach, where our solutions are very close to those achieved by a centralized method that requires orders of magnitude more time to solve the global resource allocation problem based on a complete view of the cloud platform. Our solution further renders net-revenue improvements of around 25-30% over other heuristics from the literature that are currently deployed by IaaS providers . Moreover, our algorithms are suitable for parallel implementations that achieve nearly linear speed-up. Finally, we have also investigated the mutual influences and interrelationships among the multiple time-scales (24h/1h/5-15 minutes) by varying the characteristics of the prediction errors and indentifying the best fine-grained time-scale for run-time control. The remainder of the paper is organized as follows. §II reviews previous work from the literature, while §III introduces our hierarchical framework. The optimization formulations and their solutions are presented in §IV and §V, respectively. §VI is dedicated to experimental results, with concluding remarks in §VII.

## II. RELATED WORK

Many solutions have been proposed for the self-management of cloud service centers, each seeking to meet applicationrequirementswhilecontrollingtheunderlyinginfrastructure.Fivemainproblemareashavebeenconsidered in system allocation policy design: 1) application/VM placement, 2) admission control, 3) capacity allocation, 4) load balancing, and 5) energy consumption. While each area has often been addressed separately, it is noteworthy that these problem solutions are closely related. For example, the request volume determined for a given application at a given server depends on the capacity allocated to that application on the server. A primary contribution of this paper is to integrate all five problem areas within a unifying framework, providing very efficient and robust solutions at multiple time-scales. Three main approaches have been developed: (i) control theoretic feedback loop techniques, (ii) adaptive machine learning approaches, and (iii) utility-based optimization techniques. A main advantage of a control theoretic feedback loop is system stability guarantees. Upon workload changes, these techniques can also accurately model transient behaviour and adjust system configurations within a transitory period, which can be fixed at design time. A previous study implemented a limited lookahead controller that determines servers in the active state, their operating frequency, and the allocation of VMs to physical servers. However, this implementation considers the VM placement and capacity allocation problems separately, and scalability of the proposed approach is not considered. A recent study

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

proposed hierarchical control solutions, particularly providing a cluster-level control architecture that coordinates multiple server power controllers within a virtualized server cluster . The higher layer controller determines capacity allocation and VM migration within a cluster, while the inner controllers determine the power level of individual servers. However, application and VM placement within each service center cluster is assumed to be given. Similarly, coordination among multiple power controllers acting at rack enclosures and at the server level is done without providing performance guarantees for the running applications . Machine learning techniques are based on live system learning sessions, without a need for analytical models of applications and the underlying infrastructure. autonomic managers with different goals, integrating a performance manager with a power manager in order to satisfy performance constraints while minimizing energy expenses exploiting server frequency scaling. Recent studies provide solutions for server provisioning and VM placement and propose an overall framework for autonomic cloud management. An advantage of machine learning techniques is that they accurately capture system behaviour without any explicit performance or traffic model and with little built-in system-specific knowledge. However, training sessions tend to extend over several hours, retraining is required for evolving workloads, and existing techniques are often restricted to separately applying actuation mechanisms to a limited set of managed applications. Utility based approaches have been introduced to optimize the degree of user satisfaction by expressing their goals in term so user level performance metrics.Typically,the system is modelledbymeansofaperformancemodelembedded within an optimization framework. Optimization can provide global optimal solutions or sub-optimal solutions by means of heuristics, depending on the complexity of the optimization model. Optimization is typically applied to each one of the five problems separately. Some research studies address admission control for overload protection of servers . Capacity allocation is typically viewed as a separate optimization activity which operates under the assumption that servers are protected from overload. VM placement recently has been widely studied. A previous study has presented a multi-layer and multiple time-scale solution for the management of virtualized systems, but the trade-off between performance and system costs is not considered. A hierarchical framework for maximizing cloud provider profits, taking into account server provisioning and VM placement problems, has been similarly proposed , but only very small systems have been analysed. The problem of VM placement within an IaaS provider has been considered , providing a novel stochastic performance model for estimating the time required for VMs startup. However, each VM is considered as a black box and performance guarantees at the application level are not provided. Finally, frameworks for the co-location of VMs into clusters based on an analysis of 24-hour application workload profiles have been proposed, but only the server provisioning and capacity allocation problems have been solved without considering the frequency scaling mechanism available with current technology. In this paper we propose a utility-based optimization approach, extending our previous works through a scalable hierarchical implementation across multiple time-scales. In particular has provided a theoretical framework investigating the conditions under which a hierarchical approch can provide the same solution of a centralized one. The hierarchical implementation we propose here is built on that framework and includes the CM and the AMs while in only a CM was considered which implied more time to solve the global resource allocation problem as well as a complete view of the cloud platform. To the best of our knowledge, this is the first integrated framework that also provides availability guarantees to running applications, since all foregoing work considered only performance and energy management problems.

## III. HIERARCHICAL RESOURCE MANAGEMENT

This section provides an overview of our approach for hierarchical resource management across multiple time-scales in very large scale cloud platforms. The main components of our run-time management framework are described in III-A, while our performance and power models are discussed in III-B.

A. Run-time Framework We assume the PaaS provider supports the execution of multiple transactional services, each representing a different customer application. The hosted services can be heterogeneous with respect to resource demands, workload intensities and QoS requirements. Services with different quality and workload profiles are categorized into independent request classes, where the system overall serves a set R of request classes. The architecture of the service center under consideration. The system includes a set S of heterogeneous servers, each runing a Virtual Machine Monitor (VMM) (such as IBM POWER Hypervisor, VMWare or Xen) configured in non-work-conserving mode (i.e., computing resources are capped and reserved for the execution of individual VMs). The physical resources (e.g., CPU, disks, communication network) of a server are partitioned among multiple virtual

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

machines, each running a dedicated application. Among the many physical resources, we focus on the CPU and RAM as representative resources for the resource allocation problem, consistent with . Any class can be supported by multiple application tiers (e.g., front-end, application logic, and data management). Each VM hosts a single application tier, where multiple VMs implementing the same application tier can be run in parallel on different hosts. Under light-load conditions, servers may be shut-down or placed in a stand-by state and moved to a free server pool in order to reduce energy costs. These servers may be subsequently moved back to an active state and re-introduced into the running infrastructure during peaks in the load. To go into details, we assume that each server has a single CPU supporting Dynamic Voltage and Frequency Scaling (DVFS) by choosing both its supply voltage and operating frequency from a limited set of values, noting that this single-CPU assumption is without loss of generality in heavy traffic [56] and can be easily relaxed in general. The adoption of DVFS is very promising since it does not introduce any system overhead, while hibernating and restoring a server both require time and energy.

To go into details, we assume that each server has a single CPU supporting Dynamic Voltage and Frequency Scaling (DVFS) by choosing both its supply voltage and operating frequency from a limited set of values, noting that this single-CPU assumption is without loss of generality in heavy traffic and can be easily relaxed in general. The adoption of DVFS is very promising since it does not introduce any system overhead, while hibernating and restoring a server both require time and energy. Following we adopt full system power models and assume that the power consumption of a server depends on its operating frequency/voltage as well as the current CPU utilization. Moreover, as in but without limiting the generality, servers performance is assumed to be linear in its operating frequency. Finally, each physical server s has availability as, and in order to provide availability guarantees, the VMs running the application tiers are replicated on multiple physical servers and work under the so-called load-sharing configuration. Our resource management framework is based on a hierarchical architecture. At the highest level of the hierarchy, a Central Manager (CM) acts on a time scale of T1 (i.e., every 24 hours) and is responsible for partitioning the classes and servers into clusters one day in advance (long-term problem). The objective is to obtain a set of clusters C, each element of which has a homogeneous profile in order to reduce server switching at finer grained time scales. Furthermore, we denote by Rc the set of request classes assigned to cluster c ∈Cand by St c the set of physical servers in cluster c at time interval t. At a lower level of the hierarchy, Application Managers (AMs) centrally manage individual clusters acting on a time scale of T2 (i.e., on a hourly basis). Each AM can decide (medium-term problem): (i) application placement within the cluster, (ii) load balancing of requests among parallel VMs supporting the same application tier, (iii) capacity allocation of competing VMs running on each server, (iv) switching servers into active or low power sleep states, and (v) increasing/decreasing the operating frequency operation of the CPU of a server.

B. System Performance and Power Models The cloud service center is modelled as a queueing network composed of a set of multi-class single-server queues and a delay center. Each layer of queues represents the collection of applications supporting the execution of requests at each tier.

## IV. FORMULATION OF OPTIMIZATION PROBLEMS

In this section we introduce resource management optimization problems at multiple time scales (T1,T 2, T3). The general objective is to maximize profit, namely the difference between revenues from SLA contracts and costs associated with servers switching and VM migrations, while providing availability guarantees. We formulate the optimization problem at the time scale of T2 (medium-term problem) from which, at the end of this section, will straightforwardly render the formulations at the other time scales. As previously explained, the CM determines one day in advance the class partition Rc and the server assignment St c for each cluster c and every time interval t =1,...,nT2. The AM controlling cluster c has to determine at time scale T2: (i) the servers in active state, introducing for each server s a binary variable xt s equal to 1 if server s is running in interval t and 0 otherwise; (ii) the operating frequency of the CPU of each server, given by the binary variable yts,f equal to 1 if server s is working with frequency f in interval t and 0 otherwise; (iii) the set of applications executed by each server, introducing the binary variable zts,r,τ equal to 1 if the application tier τ for class r is assigned to server s in interval t and 0 otherwise; (iv) the rate of execution for class r at tier τ on server s in interval t (i.e., λts,r,τ); (v) the fraction of capacity devoted to executing VM at tier τ for class r at server s in interval t (i.e., φts,r,τ).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

## V. SOLUTION OF OPTIMIZATION PROBLEMS

In this section we describe an efficient and effective approach for solving the computationally hard optimization problems of the previous section. We first consider the medium-term (RAPT2 c ) and short-term (RAPT3 c ) resource allocation problems, then turn to the long-term (RAPT1) optimization problem, and end by briefly discussing the interactions among them. A. Medium and Short-Term Solutions In this section we discuss the optimization techniques used for solving the medium-term (RAPT2 c ) and short-term (RAPT3 c ) resource allocation problems. Computational results demonstrate that only small instances of these problems can be solved to optimality with commercial non-linear optimization packages. To handle representative cloud service center problem sizes, we develop a heuristic approach. Three basic components comprise our overall solution approach. Step 1. An initial solution for the medium-term problem is found by applying a greedy algorithm.

**TABLE IV PERFORMANCE OF THE HIERARCHICAL ALGORITHM**

C. Availability Impact Analysis In this section we compare our medium-term algorithm with our solution in [8]. The algorithm presented in this paper extends our previous work providing availability guarantees to the running applications and implementing the diversification strategy that periodically relaxes availability constraints and recovers a feasible solution. In order to show the effectiveness of our diversification procedure, reports the execution trace of a significant run of our algorithms for a system including 50 classes and 400 servers. The x-axis reports the execution time, while the y-axis reports the objective function value. The plot on the left has been obtained by the previous solution in and shows that the standard Local Search withAvailability() procedure got stuck in a local optima after around 25 sec. In the plot on the right, Solution A depicts the best found feasible solution, while Solution B is the current solution of the new Alg. 1. This plot shows that after almost 25 sec, Alg. 1 finds the first local optimum and Solution A is not further improved. Then, constraint (9) is relaxed and Alg. 1 can continue to explore the solution space. In around 35 sec Alg. 1 finds a better optimum feasible solution and the plot of Solution A coincides with Solution B. Note that, during the search the value of Solution B can increase or decrease. Solution B usually decreases suddenly when Alg. 1 goes back in the feasible solution set. Indeed, in that case additional servers are turned on to improve the system availability and additional energy costs are incurred accordingly. As a final consideration, the plot shows that the Alg. 1 is effective since the transition in the infeasible region allows finding after around 55 sec a local optimum feasible solution which is almost 30% greater than the first local optimum at time 25 sec. In order to evaluate the impact of availability constraints on the net revenues we have also considered a case study based on realistic workloads created from the logs of the Web site of a large University in Italy. From these logs, we have extracted 50 requests classes that correspond to the days with the highest workloads experienced during the year. The workload follows a bimodal distribution with two peaks around 11.00 and 16.00 . Results are illustrated in Fig. 9 which reports the number of on-line servers and net revenues for a Service center including 400 servers with and without availability constraints. As the plots show, under light load conditions between 1.00 and 9.00, the solution with availability constraints introduces almost 100% additional servers with respect to the solution without availability. Conversely, under heavy load conditions, the system naturally adopts a larger number of servers and, in that case, the solution with availability introduces only 30% additional servers. Furthermore, by inspecting the obtained solutions, results have shown that the Alg. 1 is able to exploit the DVFS capability and to manage service resources effectively since the net revenues of the solution with availability are only 20% lower on average over the 24 hours.

## VI. CONCLUSIONS

In this paper we have proposed resource allocation policies for the management of multi-tier virtualized cloud systems with the aim at maximizing the profits associated with multiple-class of SLAs and minimizing the infras- tructure energy costs. A hierarchical framework acting at multiple time-scales and providing availability guarantees for the running applications was developed. Cloud service centres including up to 7,200 servers, 700 request classes and 60,000 VMs can be managed efficiently by a parallel implementation of a local search algorithm achieving an almost linear speed-up. The effectiveness of our approach has been assessed by considering realistic workloads. The interrelationships which characterize the time-scales that govern the management of very large scales systems have also been analyzed. In our current work, we are extending the resource allocation problem by considering at finer grained

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

time-scale the adoption of pure control theory models at time scales of seconds. At larger time granularity we are considering the management of resources among multiple distributed service centres, exploiting also the local availability of green energy sources. Finally, more advanced availability models for characterizing cloud system behaviours and a peer-to-peer architecture for the AMs will be also investigated.