International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 2, Issue 5(May 2013), PP.40-45
www.irjes.com

Heuristic Based Resource Allocation Using Virtual Machine Migration: A Cloud Computing Perspective

Ts`epoMofolo[1], R Suchithra[2]
[1,2]MS (IT) Department, Jain University Bangalore, India

Abstract: *The emerging cloud computing paradigm provides administrators and IT organizations with tremendous freedom to dynamically migrate virtualized computing services between physical servers in cloud data centers. Virtualization and VM migration capabilities enable the data center to consolidate their computing services and use minimal number of physical servers. VM migration offers great benefits such as load balancing, server consolidation, online maintenance and proactive fault tolerance. However, in cloud computing environments the cost of VM migration requires thorough consideration. Each VM migration may result in SLA violation, hence it is essential to minimize the number of migrations to the extent possible. Failure to do so will result in performance degradation and the cloud provider will have to incur the cost in monetary terms. In previous works, the issue of SLA violation has not received thorough analysis. In this work, we devise an algorithm that will keep the migration time minimum as well as minimizing the number of migrations. This will play a major role in avoiding the performance degradation encountered by a migrating VM.*

I. INTRODUCTION

Cloud Computing is proving to be a phenomenal technology where computing services are provided over the computer networks, with on-demand elastic resources like computing energy, storage capacity, memory and network [1]. Virtualization provides an efficient solution to the objectives of the cloud computing paradigm by facilitating creation of Virtual Machines (VMs) over the underlying physical servers, leading to improved resource utilization and abstraction. Virtualization refers to creating a virtual version of a device or a resource such as a server, a storage device, network or even operating system where the mechanism divides the resource into one or more execution environments. Devices, applications and end-users interact with the virtual resource as if it were a real single logical resource. The factors that a cloud provider must take into account are elasticity, scalability, live migration of VMs and performance isolation. Live migration of VMs, the process of dynamically transferring a virtual machine across different servers on the fly, has proved to represent a new opportunity to enable agile and dynamic resource management in modern data centers [2]. This is of utmost importance since data center networks are fraught with scalability and efficiency issues, which have become aspects of concern among practitioners and researchers [2]. The resource allocation algorithms take the resource requirements of a VM into account and changes the allocated resources, thus making it an on demand elastic cloud. VM placement and migration have become an integral part of resource allocation in cloud data centers. Changes in the resource requirements of VMs are significant information for VM placement and migration considerations. In our previous work in [3], we have emphasized that techniques that ensure very little utilization of resources as a result of the VM migration should not be developed as it leads to high downtime and subsequently service degradation which is not desirable in cloud platforms. Dynamic consolidation methods which aim at minimizing the number of migrations as much as possible should be employed in cloud computing. In a cloud, placement algorithms have a major responsibility of efficiently placing VMs on a physical hosts. In this work, we present a VM placement algorithm which takes into account the resource utilizations of a physical servers by the executing the VMs. The consumed resources are classified as CPU, RAM and the network bandwidth. The storage capacity resource is not taken into consideration in this work since we assume Network Attached Storage environments.Based on the data gathered from the resource utilization we apply the VM placement algorithm.Our work is based on the cloud infrastructure depicted below:

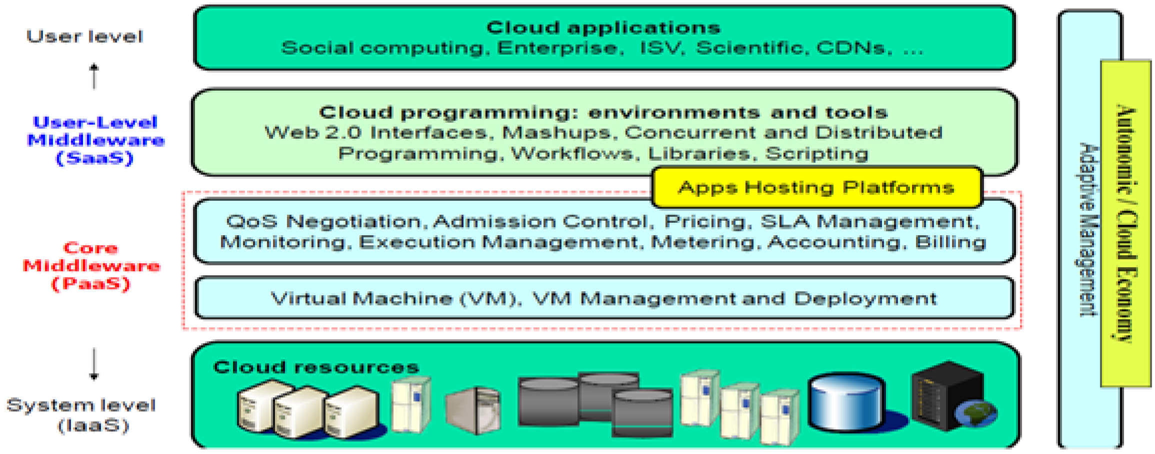

Figure 1: Layered Cloud Computing Architecture [11]

II. RELATED WORK

Some work on carrying out VM migrations while meeting SLA violations has been conducted in recent researches. Jung et al. [5, 6] have studied the problem of dynamic consolidation of VMs executing multi-tier web-application using live migration while meeting SLA requirements. They model SLA requirements as the response time pre-calculated for each type of transactions specific to the web-application. They result in a new VM placement using bin packing algorithm and gradient search techniques. The migration manager determines whether there is a reconfiguration that is effective based on the utility function that accounts for the SLA fulfillment. However, this mechanism can only be applicable to a single web-application environment and, therefore it cannot be incorporated into a multi-tenant Infrastructure as a Service (IaaS) environment. In [7], Zhu et al. have investigated a similar scenario of automated resource allocation and capacity planning. They proposed three individual controllers each executing at a different time scale as follows: longest time scale (hours to days), shorter time scale (minutes) and shortest time scale (seconds). These three controllers place compatible workloads onto groups of servers, react to changing conditions by reallocating VMs, and allocate resources to VMs within the servers to fulfill the imposed SLAs. The middle-scale controller applies a technique based on the idea of setting fixed utilization thresholds. However, fixed thresholds are not suitable for IaaS environments with dynamic workloads that exhibit non-stationary resource utilization patterns [4].In [8], Kumar et al. have proposed a technique for dynamic VM consolidation based on an estimation of “Stability”- the probability that a proposed VM reallocation will remain effective for a certain period of time in the future. Predictions of future resource demands of applications are carried out using a time-variant probability density function. The main drawback is that the authors have assumed that the metrics of distribution such as standard deviation and mean are known *a priori*. They further assumed that these values can be obtained using offline profiling of applications and online calibration. However, offline profiling is unrealistic for IaaS environments. Furthermore, the authors have assumed that the resource utilization follows a normal distribution. However, it has been proved that resource consumption by applications as well as VMs is more complex and cannot be modelled using simple probability distribution.

In [9], Berral et al have investigated the problem the problem of dynamic consolidation of VMs executing applications with deadlines that are defined in the SLAs. Using machine learning techniques, they optimize the combination of power consumption and SLA fulfillment. The proposed mechanism is designed for certain environments, such as High Performance Computing (HPC), where applications have deadline constraints. Hence, such a mechanism is not suitable and feasible for infrastructures with variable workloads. According to Blagodurov et al. in [10], servers in most data centers are often underutilized due to concerns about SLA violations that may result from resource contention as physical server utilization increases. To mitigate this issue they assume a virtualized data center that uses work conserving approach (that is, physical resources are shared among hosted VMs). The work conserving approach however, typically leads to higher utilization but no resource access guarantees can be made. This situation is highly undesirable in cloud data centers. Their solution is to consolidate both batch and interactive workloads on each server, enabling a very high utilization level (80% and above). At higher utilizations, the performance of the hosted workloads is likely to degrade due to shared server resources. This is prevented by providing prioritized access to the physical resources using Linux Control Groups (cgroups) CPU shares. The authors ensure that critical workloads have preferred access to physical resources such that they exhibit similar performance when consolidated with non-critical workloads as compared to when they are not. This is not applicable in IaaS infrastructure as applications exhibit varying workloads over time.In this work, we pay more attention on the increasing CPU utilization that increases due to VMs executions on the physical server over time as well as contention of resources from other applications which maybe executing on the physical server under consideration. Other physical server resource constraints such as the memory and the network bandwidth are taken into consideration while carrying out the VM migration process. We demonstrate techniques for achieving minimal number of migrations as well as minimal migration durations in cloud data center. We carry out the experiments using the CloudSim simulation toolkit [11].

III. HEURISTIC BASED VM PLACEMENT

The heuristic based VM migration scenario is partitioned as follows:

1. Determining when a physical server is considered to be overloaded requiring live migration of one or more VMs from the physical server under consideration.
2. Determining when a physical server is considered as being under loaded hence it becomes a good candidate for hosting VMs that are being migrated from overloaded physical servers.
3. Selection of VMs that should be migrated from an overloaded physical server. VM selection policy (algorithm) has to be applied to carry out the selection process.
4. Finding a new placement of the VMs selected for migration from the overload and physical servers and finding the best physical

The VM placement problem can be considered as a bin packing problem with variable been sizes and items, where bins represent the physical servers, items represent the VMs to be allocated, and bin sizes represent the available CPU capacities of those nodes. Since bin packing problem is NP-hard, to incorporate it into our solution we apply a modification of the Best Fit Decreasing (BFD) algorithm that uses no more than $\frac{11}{9}Z(I)+4$ bins (where Z is the number of bins that provides the optimal solution). In simpler terms it has been proven that $BFD(I) \leq \frac{11}{9}Z(I)+4$ for all instances of I. In the modified BFD algorithm, we take as an input the sorted list of VMs to be migrated in descending order of their current CPU utilizations and allocate each to a selected host that provides the least number of remaining processing capacity caused by the allocation. The host list is also sorted in decreasing order of their remaining capacity to ensure that a VM is allocated to a host that has enough resources for it with the least number of attempts. This ensure high resource utilization as the resources on the target host will not be idle.

The algorithm is presented below. It has the time complexity of nm, where n is the number of physical target hosts and m is the number of VMs that have been selected for migration.

Modified Best Fit Decreasing Algorithm:

```
Input: SortedlistofTargetHosts, SortedVmList, Output: VMs allocations
Foreach VM in SortedVmList do
minCapacity←MAX
allocatedHost←NULL
Foreach host in sortedListofTargetHosts do
        If target host has enough resources for VM
Capacity←estimateCapacity(host,vm)
                If Capacity<minCapacity
                  Allocated host←host
minCapacity←Capacity
 If allocatedHost≠NULL then
Allocation.add(vm,allocatedHost)
Return allocation
```

IV. EMPIRICAL STUDY AND EXPERIMENTAL SETUP

Our empirical study seeks to achieve the following goals:

(i) Carrying out the live migration of VMs in a manner that preserves free resources in order to prevent SLA violations
(ii) Optimal utilization of resources
(iii) Performing minimal number of migrations to the extent possible
(iv) Efficient server consolidation through VM migrations

To this end we came up with a modified best fit decreasing algorithm which performs the migration based on the above mentioned objectives. This algorithm is plays the role of a VM Placement policy and is

implemented through the Cloudsim simulation toolkit which is open source and it plays an important role in guaranteeing statistical significance.
We adopt the following test bed in our simulations:

Processor: Intel (R) Core(TM) i5- 2430M CPU @ 2.40 GHz
OS: Windows 8
RAM: 6GB
System Type: 64-bit OS, x64-based Processor
Storage Capacity: 500 GB
Softwares: JDK 1.7 and Netbeans 7.2
CloudSim 3.01 Configurations.
Scheduling Interval: 300
Simulation Limit : 24*60*60
Job Length= Random
Dynamic Workload
VM Types: 4
VM Ram: {880, 1730, 1740, 620}
VM Bandwidth: 100000 (100Mbits)
VM Storage Capacity: 2000 (2GB)
Host Types : HP Proliant ML110 G4 (1x [Xeon 3040 1860 MHz, 2 Cores], 4GB)
HP Proliant ML110 G5 (1x [Xeon 3075 2660 MHz, 2 Cores], 4GB)

Host MIPS= {1870, 2770}
RAM= {4096, 4096}
Bandwidth= 1 000 000 (1 Gbit/s)
Storage Capacity= 1 000 000 (1 GB)
DATA CENTER
System Architecture= x86
OS= linux
VMM= Xen
Number of physical hosts=50
Number of VMs = 60

To validate and understand the efficiency of our algorithm we carry out experiments that are based on dynamic workload loads generated as is the case in real world cloud data centers. Our modified best fit decreasing algorithm makes use of the already available policies in Cloudsim simulation toolkit to achieve the desired objectives. The already existing policies include the minimum migration policy which has been proved to be more efficient than the other evaluated policies.

V. ANALYSIS OF RESULTS

The first experiment is based on local regression robust VM allocation policy and the second is based on the modified best fit decreasing algorithm that we devised. Statistical analysis of the simulations of both algorithms is presented below.

Simulations Analysis

Column1	Column3	Column2
	Simulation 1	Simulation 2
Number of VM migrations	3537	4172
SLA	0.06023	0.06431
SLA perf degradation due to migration	0.26	0.32
SLA time per active host	22.92	20.13
Overall SLA violation	8.11	7.42
Average SLA violation	17.31	16.14
Number of host shutdowns	927	1346

Table 1: General Simulation Results

Column1	Column3	Column2
	Simulation 1	Simulation 2
Mean time before a host shutdown	910.38	692.28
StDev time before a host shutdown	1556.8	815.96
Mean time before a VM migration	15.76	17.03
StDev time before a VM migration	7.38	7.8
Execution time - VM selection mean	0.00009	0.00014
Execution time - VM selection stDev	0.00029	0.00035
Execution time - host selection mean	0.00036	0.00038
Execution time - host selection stDev	0.00059	0.00049
Execution time - VM reallocation mean	0.00195	0.00105
Execution time - VM reallocation stDev	0.00662	0.00182
Execution time - total mean	0.00751	0.00661
Execution time - total stDev	0.01296	0.01154

Table 2: Performance analysis based on time dimension

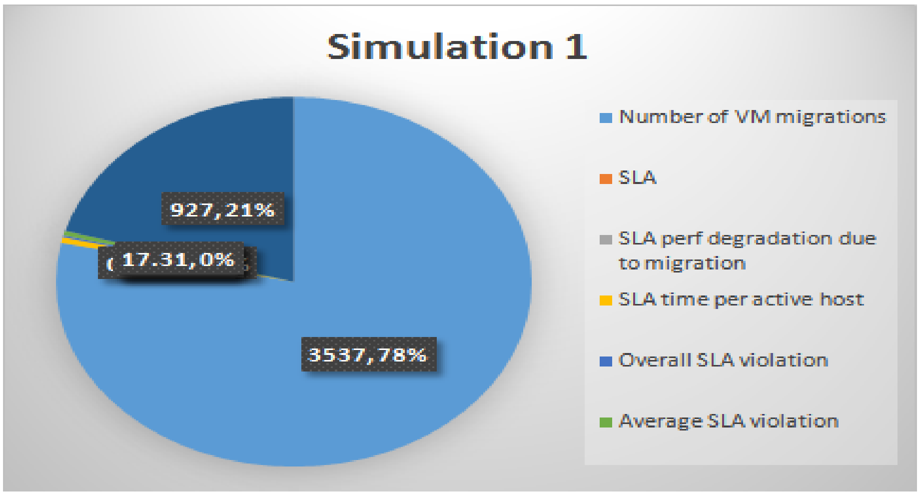

Fig 2. Bar chart for Experiment 1

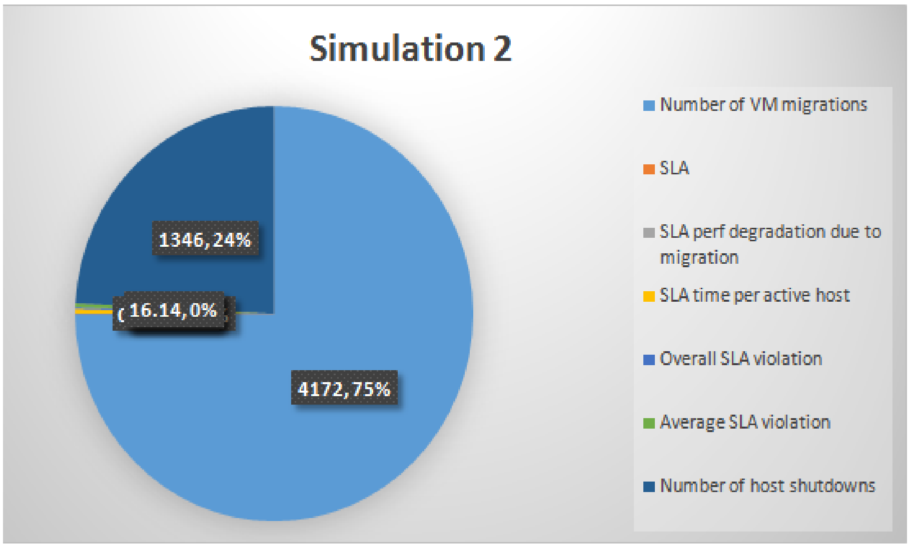

Fig 3. Bar chart for Experiment 2

VI. CONCLUDING REMARKS

Through our modified best fit decreasing algorithm we have achieved migrations of VMs on the basis of overloading that occurs in physical servers in cloud data centers. As indicated in our results, our algorithm provides relatively less performance degradation that occurs due to VM migrations. The SLA violation is also less compared to other techniques and this means the cloud provider will incur less cost from VM migrations. Further, our algorithm performs host selection and VM reallocation quicker than the existing algorithms. Maintenance of physical servers can be efficiently achieved through our algorithm as it leads to efficient consolidation of VMs.

REFERENCES

[1]. VUPIC Virtual Machine Usage Based Placement in IaaS CloudGaurav Somani, member, IEEE, PrateekKhandelwal, and KapilPhatnani

[2]. 2.Egalitarian Stable Matching for VM Migration in Cloud Computing Hong Xu, Baochun Li3. Resource Allocation using Virtual Machine Migration: A Survey Ts'epoMofolo, R. Suchithra and N. Rajkumar

[3]. Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers

[4]. Jung G, Joshi KR, Hiltunen MA, Schlichting RD, Pu C. Generating adaptation policies for multi-tier applications in consolidated server environments. Proceedings of the 5th IEEE International Conference on Autonomic Computing (ICAC 2008), Chicago, IL, USA, 2008; 23–32.

[5]. Jung G, Joshi KR, Hiltunen MA, Schlichting RD, Pu C. A cost-sensitive adaptation engine for server consolidation of multitier applications. Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware (Middleware 2009), Urbana Champaign, IL, USA, 2009; 1–20

[6]. Zhu X, Young D, Watson BJ, Wang Z, Rolia J, Singhal S, McKee B, Hyser C, Gmach D, Gardner R, et al.. 1000 islands: Integrated capacity and workload management for the next generation data center. Proceedings of the 5thInternational Conference on Autonomic Computing (ICAC 2008), Chicago, IL, USA, 2008; 172–181.

[7]. Kumar S, Talwar V, Kumar V, Ranganathan P, Schwan K. vManage: loosely coupled platform and virtualizationmanagement in data centers. Proceedings of the 6th international conference on Autonomic computing (ICAC 2009), Barcelona, Spain, 2009; 127–136.

[8]. Berral JL, Goiri ,Nou R, Juli F, Guitart J, Gavald R, Torres J. Towards energy-aware scheduling in data centers using machine learning. Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, Passau, Germany, 2010; 215–224.

[9]. 10.Sergey Blagodurov, Daniel Gmach, Martin Arlitt, Yuan Chen, Chris Hyser, Alexandra Fedorova. Maximizing Server Utilization while Meeting Critical SLAs via Weight-Based Collocation Management

[10]. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose and RajkumarBuyya. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms