International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Skewness Algorithm Scheduling Approach for the Energetic Distribution of Resources for Cloud Computing Environment using Virtual Machines

Asha T N[1], Antony P J[2]

Department of Computer Science and Engineering, Visvesvaraya Technological University, Belgaum, KVGCE, SULLIA, DK – 574327.

Director of P G Studies, Department of Computer Science and Engineering
Visvesvaraya Technological University, Belgaum, KVGCE, SULLIA, DK – 574327

Abstract*: Cloud* computing is a new technology that uses the internet. At present cloud computing environment, the forecast approaches for (Virtual Machine) VM resources are mainly focus on the systems current state. In the resource management problems the Dynamic resource allocation problem is one of the most demanding problems. To present a better solution for solving the problem of dynamic resource allocation in a cloud computing environment, the proposed system demonstrates a skewness algorithm to determine the unevenness in the multi-dimensional resource utilization of a server. By reducing the skewness, we can combine different types of workloads and increase the overall utilization of server resources. By using virtualization technology, we can assign resources to the data center dynamically based on application demands and support green computing by optimizing the number of servers in use.

Keywords*:* Cloud computing, dynamic resource allocation, green computing, virtualization technology, skewness.

1. Introduction

Cloud computing is a fast growing technology that currently being studied in[1]. The definition of the cloud computing from the Gartner: "A style of computing where massively scalable IT-related capabilities are provided as a service across the internet to multiple external customers using internet technologies [2]." Cloud computing is the next generation in computation. Possibly people can have everything they need on the cloud. Cloud computing is the next natural step in the evolution of on-demand information technology services and products.

Virtual machine monitors (VMMs) like Xen provide a mechanism for mapping virtual machines (VMs) to physical resources [3]. This mapping is largely hidden from the cloud\ users. Users with the Amazon EC2 service [4], for example, do not know where their VM instances run. It is up to the cloud provider to make sure the underlying physical machines (PMs) have sufficient resources to meet their needs. VM live migration technology makes it possible to change the mapping between VMs and PMs while applications are running [5], [6]. However, a policy issue remains as how to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized. To minimizing skewness, we can combine different types of workloads. Skewness can be measured based on,

a) *Hot spot:* If the utilization of any of its resources is above a hot threshold. This indicates that the server is overloaded and hence some VMs running on it should be migrated away.
b) *Cold spot:* If the utilizations of all its resources are below a cold threshold. This indicates that the server is mostly idle and a potential candidate to turn off to save energy.

The two main goals of proposed system are.

a) Overload avoidance: the capacity of a PM should be sufficient to satisfy the resource needs of all applications running on it. Otherwise, the PM is overloaded and can lead to degraded performance of its application.
b) Green computing: the number of PMs used should be reduced as long as they can still satisfy the needs of all VMs. Inactive PMs can be turned off to save energy.

2. Literature Survey

Cloud computing is the next natural step in the evolution of on-demand information technology services and products. Cloud Computing is an emerging computing technology that is rapidly consolidating itself as the next big step in the development and deployment of an increasing number of distributed applications. Cloud computing nowadays becomes quite popular among a community of cloud users by offering a variety of resources. Cloud computing platforms, such as those provided by Microsoft, Amazon, Google, IBM, and Hewlett-Packard, let developers deploy applications across computers hosted by a central organization. These applications can access a large network of computing resources that are deployed and managed by a cloud computing provider. Developers obtain the advantages of a managed computing platform, without having to commit resources to design, build and maintain the network. Yet, an important problem that must be addressed effectively in the cloud is how to manage QoS and maintain SLA for cloud users that share cloud resources. The cloud computing technology makes the resource as a single point of access to the client and is implemented as pay per usage. Though there are various advantages in cloud computing such as prescribed and abstracted infrastructure, completely virtualized environment, equipped with dynamic infrastructure, pay per consumption, free of software and hardware installations, the major concern is the order in which the requests are satisfied. This evolves the scheduling

of the resources. This allocation of resources must be made efficiently that maximizes the system utilization and overall performance. Cloud computing is sold on demand on the basis of time constrains basically specified in minutes or hours. Thus scheduling should be made in such a way that the resource should be utilized efficiently.

In cloud platforms, resource allocation (or load balancing) takes place at two levels. First, when an application is uploaded to the cloud, the load balancer assigns the requested instances to physical computers, attempting to balance the computational load of multiple applications across physical computers. Second, when an application receives multiple incoming requests, these requests should be each assigned to a specific application instance to balance the computational load across a set of instances of the same application. For example, Amazon EC2 uses elastic load balancing (ELB) to control how incoming requests are handled. Application designers can direct requests to instances in specific availability zones, to specific instances, or to instances demonstrating the shortest response times. In the following sections a review of existing resource allocation techniques like Topology Aware Resource Allocation, Linear Scheduling and Resource Allocation for parallel data processing is described briefly.

A. Topology Aware Resource Allocation:

Different kinds of resource allocation mechanisms are proposed in cloud. The one mentioned in [7] proposes architecture for optimized resource allocation in Infrastructure-as-a-Service (IaaS) based cloud systems. Current IaaS systems are usually unaware of the hosted application's requirements and therefore allocate resources independently of its needs, which can significantly impact performance for distributed data-intensive applications. To address this resource allocation problem, an architecture that adopts a "what if" methodology to guide allocation decisions taken by the IaaS is proposed. The architecture uses a prediction engine with a lightweight simulator to estimate the performance of a given resource allocation and a genetic algorithm to find an optimized solution in the large search space. Results showed that TARA reduced the job completion M time of these applications by up to 59% when compared to application-independent allocation policies.

TARA [7] is composed of two major components: a prediction engine and a fast genetic algorithm-based search technique. The prediction engine is the entity responsible for optimizing resource allocation. When it receives a resource request, the prediction engine iterates through the possible subsets of available resources (each distinct subset is known as a candidate) and identifies an allocation that optimizes estimated job completion time. However, even with a lightweight prediction engine, exhaustively iterating through all possible candidates is infeasible due to the scale of IaaS systems.

Genetic algorithm (GA) is a search technique inspired by evolutionary biology for finding solutions to optimization and search problems. Candidates are represented as genes and they evolve toward better solutions. In comparison to other search techniques, GA was a good match for the resource allocation problem. It was natural to map server selection in an IaaS system to GA's gene representation, and to apply operations during the GA's evolution process. To represent each possible candidate, we use a bit string with the length of n, the number of servers available to host a single VM. The binary value of each bit means whether the corresponding server is selected or not. For each bit in the string, a value of 1 represents the physical server being selected for hosting a VM and a 0 represents the server being excluded. To evaluate a candidate, the prediction engine described above is used. Once we have the genetic representation and the fitness function, GA initializes a population of candidates. It then goes through the evolution process of reproduction and selection until it terminates.

B. Linear Scheduling And Resource Allocation For Parallel Data Processing

Considering the processing time, resource utilization based on CPU usage, memory usage and throughput, the cloud environment with the service node to control all clients request, could provide maximum service to all clients. Scheduling the resource and tasks separately involves more waiting time and response time. A scheduling algorithm named as Linear Scheduling for Tasks and Resources (LSTR) is designed [8], which performs tasks and resources scheduling respectively. Here, the combination of Nimbus and Cumulus services are imported to a server node to establish the IaaS cloud environment and KVM/Xen[3] virtualization along with LSTR scheduling is used to allocate resources which maximize the system throughput and resource utilization. Linear scheduling and resource allocation for parallel data processing not focused on Cloud demand and cloud resource utilization are factors that most of the IT industries and other organization will demand. Hence, apart from scheduling the task and resources, the factors such as time, and wastage of demanded resources from every client should be monitored.

The provision of resource may be made with various virtualization techniques. This may ensure a higher throughput and usage than the existing cloud resource services. The future work is required to deals with the evolutionary techniques that will further result in better resource allocation, leading to improve resource utilization. These resource allocation strategies have the following limitations.

a) Since users rent resources from remote servers for their purpose, they don't have control over their resources.
b) Migration problem occurs, when the users wants to switch to some other provider for the better storage of their data. It's not easy to transfer huge data from one provider to the other.
c) More and deeper knowledge is required for allocating and managing resources in cloud, since all knowledge about the working of the cloud mainly depends upon the cloud service provider.

Hence the existing systems are has the limitations as migration of resources, overloading at server and migrates only working set of an idle VM. To overcome from these limitations this paper presents skewness algorithm which uses green computing technologies. This will be discussed in the following section.

3. Problem Statement

The proposed system aims to find the problem of mapping resources adaptively so that the resource demands of virtual machines are met in the cloud computing environment, while the number of physical machines used is minimized. So, Physical machine is overloaded and can lead to degraded performance of its virtual machines. On other hand, if the resource utilization of active server is too low, while the server is turned on resulting unnecessary use of power. To overcome from these problems the system will keep the utilization of Physical machines low to reduce the possibility of overload, in case the resource needs of virtual machines increases. The proposed system uses green computing technology to save the unnecessary wasting of power and to increase the life time of server.

4. Approach

The proposed system presents the design and implementation of an automated resource management system that achieves a good balance. The proposed system makes the following three contributions:

a) Develops a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used.
b) Introduces the concept of "skewness" to measure the uneven utilization of a server. By minimizing skewness, thus we can improve the overall utilization of servers in the face of multi-dimensional resource constraints.

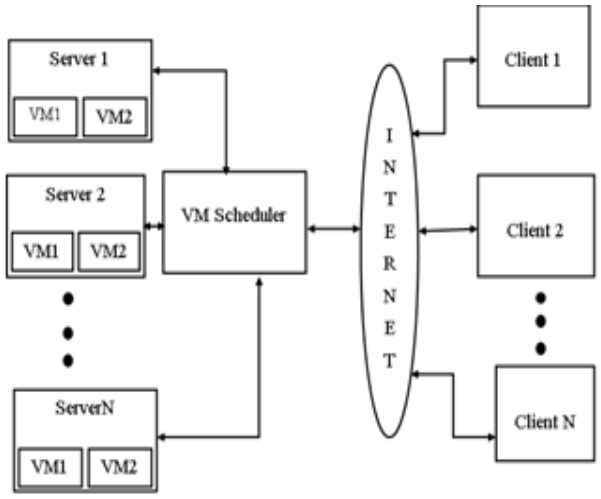


Figure 1: Architecture of the proposed method.

c) Designs a load prediction algorithm that can capture the future resource usages of applications accurately without looking inside the VMs. The algorithm can capture the rising trend of resource usage patterns and help reduce the placement churn significantly.

The main goal of proposed system is "overload avoidance", which is the capacity of a PM should be sufficient to satisfy the resource needs of all VMs running on it. Otherwise, the PM is overloaded and can lead to degraded performance of its VMs. For overload avoidance, we should keep the utilization of PMs low to reduce the possibility of overload in case the resource needs of VMs increase later.

A. System Overview

The Fig.1 represents the architecture of the dynamic resource allocation for cloud computing environment, which consists of N servers each server consists of two virtual machines(VM) those are connected to the VM scheduler is connected to the internet to distribute the resources dynamically to the clients ,the clients are accessing resources through the internet. Virtual machine (VM) is a software implementation of computing environment in which operating system or program can be installed and run. The VM Scheduler is invoked periodically and receives the resource demand history of VMs, the capacity and the load history of server, and the current layout of VMs on servers. The skewness algorithm and predicting future resources are two main modules for the proposed system that are described below.

1) Skewness Algorithm

The paper introduces the concept of skewness to quantify the unevenness in the utilization of multiple resources on a server. By minimizing the skewness, we can combine different types of workloads nicely and improve the overall utilization of server resources.

Let n be the number of PMs and m be the number of VMs in the system, respectively. The number of resources such as CPU, memory, I/O, network, etc. that should be considered is usually a small constant. Thus the calculation of the skewness and the temperature metrics for a single server takes a invariable amount of time. During load prediction, we need to apply the FUSD algorithm to each VM and each PM. The time complexity is O (n+m).The skewness algorithm consists of three parts such as load prediction, hot spot mitigation, and green computing as described below.

The Load predication in skewness algorithm executes periodically to evaluate the resource allocation status based on the predicted future resource demands of VMs. The proposed system defines a server as a hot spot if the utilization of any of its resources is above a hot threshold. This indicates that the server is overloaded and hence some VMs running on it should be migrated away.

The proposed system defines a server as a cold spot if the utilizations of all its resources are below a cold threshold. This indicates that the server is mostly idle and a potential candidate to turn off to save energy. However, by doing so only when the average resource utilization of all actively used servers (i.e., APMs) in the system is below a green computing threshold [5]. A server is actively used if it has at least one VM running. Otherwise, it is inactive. Finally, the proposed system define the warm threshold to be a level of resource utilization that is sufficiently high to justify having the server running but not so high as to risk becoming a hot spot in the face of temporary fluctuation of application resource demands. Different types of resources can have different thresholds.

The Hot Spot Mitigation sorts the list of hot spots in the system in descending temperature. The goal of the proposed system is to eliminate all hot spots if possible. Otherwise, keep their temperature as low as possible. For each server p, we first decide which of its VMs should be

migrated away. We sort its list of VMs based on the resulting temperature of the server if that VM is migrated away.

When the resource utilization of active servers is too low, some of them can be turned off to save energy. This is handled in our green computing algorithm. The challenge here is to reduce the number of active servers during low load without sacrificing performance either now or in the future. The proposed green computing algorithm is invoked when the average utilizations of all resources on active servers are below the green computing threshold. We sort the list of cold spots in the system based on the ascending order of their memory size. Since the proposed system needs to migrate away all its VMs before we can shut down an under-utilized server, we define the memory size of a cold spot as the aggregate memory size of all VMs running on it. Recall that our model assumes all VMs connect to share back-end storage. Hence, the cost of a VM live migration is determined mostly by its memory footprint.

2) Predicting Future Resource Needs

We need to predict the future resource needs of VMs. As said earlier, our focus is on Internet applications. One solution is to look inside a VM for application level statistics, e.g., by parsing logs of pending requests. Doing so requires modification of the VM which may not always be possible. Instead, we make our prediction based on the past external behaviors of VMs. This project first attempt was to calculate an exponentially weighted moving average (EWMA) using a TCP-like scheme:

$$NE(I) = a * E(I) + (1 - a) * O(I), 0 \leq a \leq 1 \ldots..(1)$$

Where in exp(1) $E(I)$ and $O(I)$ are the estimated and the observed load at time t and for the I th iteration, respectively. 'a' reflects a tradeoff between stability and responsiveness. The EWMA formula is use to predict the CPU load on the DNS server.

5. Implementation

The implementation is conducted using a group of 3 servers with CPU and 24GB of RAM. It consist of two parts admin part and user session part. The admin part consists of

- Server details
- Application details
- Server monitoring
- Active connection
- Server spot
- Change password

Here server details can add, edit and delete the server. An application can add, edit and delete the application. The server monitoring will periodically collects the statics of the server to allocate the resources to the server. The servers are connected over a Gigabit ethernet to a group of three servers where the VM Scheduler runs. This starts with a small scale experiment consisting of three PMs and three VMs so that we can present the results for all servers. Different shades are used for each VM. All VMs are configured with 128 MB of RAM. An Apache server runs on each VM. It uses http reference to invoke CPU intensive java scripts on the Apache server. This allows us to subject the VMs to different degrees of CPU load by adjusting the client request rates. The utilization of other resources are kept low.

Recall that the goal of the skewness algorithm is to mix workloads with different resource requirements together so that the overall utilization of server capacity is improved. By above technique proposed algorithm handles a mix of CPU, memory, and network intensive workloads.

6. Result

By considering the performance of the proposed method with the existing Topology Aware Resource Allocation (TARA) and Linear Scheduling and Resource Allocation For Parallel Data Processing we can find the gain in the performance as we are using skeweness algorithm in combination with green computing to avoid overloading at the server, hence minimizing the usage of number of resources.

7. Conclusion and Future Scope

Cloud computing technology is progressively more used in enterprises and business markets. A review shows that dynamic resource allocation is growing need of cloud providers for more number of users and with the less number of systems. Based on the changing demand the proposed system multiplexes virtual to physical resources. The system uses the skewness metric to mix VMs with different resources. The proposed algorithm achieve overload avoidance by predicting the future needs and by using green computing technology we can turn off the idle serves. The skewness algorithm supports load balance as well as green computing. We also adopt load predicting to improve scheduling effectiveness. The investigation on prediction algorithms are left as future work.

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, "Abovethe clouds: A berkeley view of cloud computing," UCB/EECS-2009-28.
[2] Cloud Computing Definition Gartner.
[3] http://www.gartner.com/it/page.jsp?id=1035013
[4] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho,R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proc. of the ACM Symposium on Operating Systems Principles (SOSP'03)*, Oct. 2003.
[5] "Amazon elastic compute cloud (Amazon EC2), http://aws.amazon.com/ec2/."
[6] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in Proc. of the Symposium on Networked Systems Design and Implementation(NSDI'05), May 2005.
[7] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast transparent migration forvirtual machines," in Proc. of the USENIX Annual Technical Conference,2005.

[8] Gunho Lee, Niraj Tolia, Parthasarathy Ranganathan, and Randy H.Katz, Topology aware resorce allocation for data-intensive workloads,ACM SIGCOMM Computer Communication Review, 41(1):120--124,2011.
[9] Abirami S.P. and Shalini Ramanathan, Linear scheduling strategy for resource allocation in cloud environment, International Journal on Cloud Computing: Services and Architecture(IJCCSA), 2(1):9--17,2012.

Author Profile



Asha T N received the B.E degree in Computer Science and Engineering from BTL Institute of Technology in 2012. Now studying M.Tech in KVG College of Engineering.