International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

A Simulation Based Analysis and Modeling of Workload Patterns using the CloudSim Toolkit

Abhilasha Singh

Department of Computer Science and Engineering, M.S Ramaiah Institute of Technology, Bangalore, Karnataka, India

Abstract: ***The distinctiveness and patterns of workloads in a Cloud computing atmosphere is important in order to progress resource management and operational circumstances as Quality of Service guarantees are maintained. Simulation models based on pragmatic parameters are also directly required for investigating the crash of this workload distinctiveness on novel system designs and operation policies. There is a need of analyses to sustain the progress of workload models that confine the intrinsic range of users and tasks, mostly suitable to the restricted ease of use of Cloud trace logs as well as the difficulty in analyzing such systems. In this research, an extensive examination is performed of the workload distinctiveness consequent from a production Cloud datacenter. The examination focuses on revealing and quantifying the variety of behavioral patterns for users and tasks, as well as identifying model parameters and their values for the simulation of the workload formed by such mechanism. The derivative model is implemented by extending the capabilities of the CloudSim structure and is advance validated during experimental assessment and statistical suggestion tests.***

Keywords: Cloud Computing, Clustering, VM allocation, Virtualization, CloudSim.

1. Introduction

Cloud computing [1] relies on restricting sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services. Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand [2]. This can work for allocating resources to users. For example, a cloud computer facility that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server). This approach should maximize the use of computing power thus reducing environmental damage as well since less power, air conditioning, rackspace, etc. are required for a variety of functions. With cloud computing, multiple users can access a single server to retrieve and update their data without purchasing licenses for different applications.

In the most basic cloud-service model & according to the IETF (Internet Engineering Task Force), providers of IaaS offer computers – physical or (more often) virtual machines – and other resources. (A hypervisor, such as Xen, Oracle VirtualBox, KVM, VMware ESX/ESXi, or Hyper-V runs the virtual machines as guests. Pools of hypervisors within the cloud operational support-system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements.) IaaS clouds often offer additional resources such as a virtual-machine disk image library, raw block storage, and file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles.IaaS-cloud providers supply these resources on-demand from their large pools installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).

Analysis and simulation of Cloud tasks and users significantly benefits both providers and researchers, as it enables a more in-depth understanding of the entire system as well as offering a practical way to improve datacenter functionality. For providers, it enables a method to enhance resource management mechanisms to effectively leverage the diversity of users and tasks to increase the productivity and QoS of their systems. However, deriving such analyses is challenging in two specific areas. The first and most critical problem is that there are few available data sources pertaining to large-scale production utility Clouds, due to business and confidentiality concerns. This is a particular challenge in academia, which relies on the very few publicly available Cloud trace logs. The second problem is analysis and simulation of realistic workloads; this is due to the massive size and complexity of data that a typical production Cloud can generate in terms of sheer volume of users and server events as well as recording resource utilization of tasks.

CloudSim [3] is a simulation tool that allows cloud developers to test the performance of their provisioning policies in a repeatable and controllable environment, free of cost. It helps tune the bottlenecks before real-world deployment. It is a simulator; hence, it doesn't run any actual software. It can be defined as 'running a model of an environment in a model of hardware', where technology-specific details are abstracted. CloudSim is a library for the simulation of cloud scenarios. It provides essential classes for describing data centers, computational resources, virtual machines, applications, users, and policies for the management of various parts of the system such as scheduling and provisioning. Using these components, it is easy to evaluate new strategies governing the use of clouds, while considering policies, scheduling algorithms, load balancing policies, etc. It can also be used to assess the competence of strategies from various perspectives such as cost, application execution

time, etc. It also supports the evaluation of Green IT policies. It can be used as a building block for a simulated cloud environment and can add new policies for scheduling, load balancing and new scenarios. It is flexible enough to be used as a library that allows you to add a desired scenario by writing a Java program. By using CloudSim, organizations, R&D centers and industry-based developers can test the performance of a newly developed application in a controlled and easy to set-up environment.

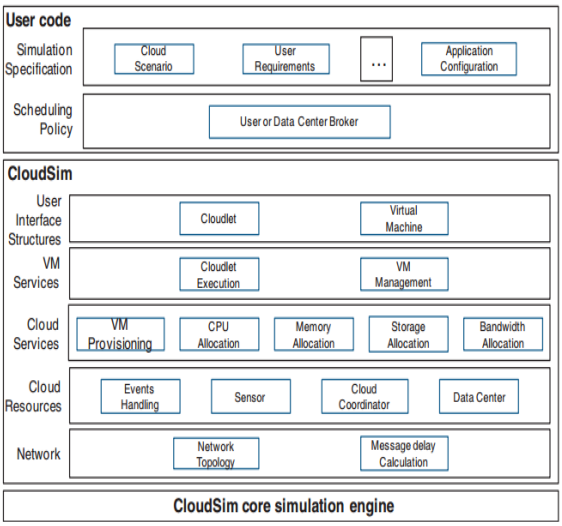

Figure 1: Layered CloudSim architecture

Figure 1.1 shows the multilayered design of the CloudSim software framework and its architectural components. Initial releases of CloudSim as the discrete event simulation engine that supports several core functionalities, such as queuing and processing of events, creation of Cloud system entities (services, host, data center, broker, VMs), communication between components, and management of the simulation clock. However in the current release, the SimJava layer has been removed in order to allow some advanced operations that are not supported by it.

We use a keyword VM allocation which is used in cloud computing for the virtual sharing of physical machine among the data centers. It provides the information of allocated VM to a particular data center ID. This allocation of VM is based on different policies that make it efficient and easy to understand and allocation policies of these VM can be implemented at virtualization level. At the infrastructure layer, the virtualization of cloud element takes place. Cloud infrastructure is highly structured and scalable depending on these allocation policies.

There is a need of analyses to sustain the progress of workload models that confine the intrinsic range of users and tasks, mostly suitable to the restricted ease of use of Cloud trace logs as well as the difficulty in analyzing such systems. This focuses on revealing and quantifying the variety of behavioral patterns for users and tasks, as well as identifying model parameters and their values for the simulation of the workload formed by such mechanism. The derivative model is implemented by extending the capabilities of the CloudSim structure and is advance validated during experimental assessment and statistical suggestion tests. , it aims to provide a validated simulation model that includes parameters of tasks and users to be made available for other researchers to use. It focuses specifically on a substantial analysis of cloud diversity of users and tasks.

The paper is organized as follows: Section II provides an insight into the related work. Section III and IV present the existing and proposed systems respectively. Section V presents the Simulation setup and configurations. Results are presented in Section VI. Finally, Conclusion and Future Work are given in Section VII.

2. Related Work

In 2009 Bryan M. A. El-Refaey and M. A. Rizkaa proposed a paper on Virtual Systems Workload Characterization [4] which introduced an indication of the key requirements and distinctiveness of virtual systems performance metrics and workload characterization which can be measured one step further in implementing virtual systems benchmark and performance model that describe the effect of the applications, host operating system and the hypervisor layer on the performance metrics of virtual workloads. An impression of Intel vCon model and VMware VMmark will be introduced as examples for the consolidated servers' workload evaluation. The vConsolidate (vCon) benchmark is one of the proposed benchmarks for virtualization consolidation developed by Intel and it can be considered a VMM agnostic. The vCon benchmark consists of a compute intensive Workload/application, a database workload, a Web server workload, and a mail server workload. Each of these workloads runs in its own VM.

In 2010 introduction of An Analysis of Traces from a Production Map Reduce Cluster [5] by S. Kavulya, in which introduced the concept of Hadoop logs from the 400-node M45 supercomputing cluster which Yahoo made freely available to select universities for systems research. These studies track the evolution in cluster utilization 14. Job completion times and cluster allocation patterns followed a long-tailed distribution motivating the need for fair job schedulers to prevent large jobs or heavy users from monopolizing the cluster. Here also observed large error-latencies in some long-running tasks indicating that better diagnosis and recovery approaches are needed. User tended to run the same job repeatedly over short intervals of time thereby allowing us to exploit temporal locality to predict job completion times. Here compared the effectiveness of a distance-weighted algorithm against a locally-weighted linear algorithm at predicting job completion times when scaled the map input sizes of incoming jobs. Locally-weighted linear regression performs better with a mean relative prediction error of 26% compared to 70% for the distance-weighted algorithm. Here measured disequity in the distribution of task durations using the Gini Coeffient and observed that large Gini Coefficients in the reduce durations are correlated with large prediction errors implying that peer-comparison might be a feasible strategy for detecting performance problems.

In 2010Jianfeng Zhan, Lei Wang, Weisong Shi, Shimin Gong and Xiutao Zang introduced Phoenix Cloud-Provisioning Resources for Heterogeneous Cloud Workloads [6]. It presented a RE conformity that state diverse RE necessities and make a novel system Phoenix Cloud to allow creating REs on require according to RE agreements. On behalf of two characteristic heterogeneous workloads: Web servicesand parallel batch jobs, proposed two corresponding resource provisioning solutions in two diverse Cloud scenarios. on behalf of three distinctive workload traces: SDSC BLUE, NASA iPSC and World Cup, experiments showed that: a) in the first Cloud scenario, when the throughput is almost same like that of a DCS, solution decreases the design size of cluster by about 40%; b) in the second Cloud scenario, solution decreases not only the total resource expenditure, but also the peak resource consumption maximally to 31% with respect to that of EC2 + Right Scale solution.

In 2011 Ajit B. Sharma proposed a Modeling and Synthesizing Task Placement Constraints in Google Compute Clusters [7] which addresses the concert impact of task residency constraints. Task residency constraints impact which resources tasks consume. Task placement constraints, such as distinctiveness individual by the Condor Class Ads method, offer a way to contract with machine heterogeneity and diverse software requirements in work out clusters. This understanding at Google suggests that task assignment constraints can have a huge impact on task scheduling delays. This paper is the first to expand a method that addresses the performance impact of task placement constraints. Here showed that in Google compute clusters, constraints can enhance average task scheduling delays by a factor of 2 to 6, which frequently revenue tens of minutes of additional task wait time. To recognize why, to begin a new metric, the Utilization Multiplier (UM) that extends the idea of resource utilization to comprise constraints. It showed that task scheduling delays enhance with UM for the responsibilities to learn. It as well shows how to describe and produce representative task constraints and machine properties, and how to include synthetic constraints and properties into presented performance benchmarks. Applying this come up to Google compute clusters, here find that these constraint characterizations accurately reproduce production performance characteristics.

In 2013 I. Solis Moreno proposed a paper on An Approach for Characterizing Workloads in Google Cloud to Derive Realistic Resource Utilization Models [8] which introduced analyzing behavioral patterns of workloads is critical to understanding Cloud computing environments. However, until now only a limited number of real-world Cloud data center trace logs have been available for analysis. This has led to a lack of methodologies to capture the diversity of patterns that exist in such datasets. This paper presents the first large-scale analysis of real-world Cloud data, using a recently released dataset that features traces from over 12,000 servers over the period of a month. Based on this analysis, we develop a novel approach for characterizing workloads that for the first time considers Cloud workload in the context of both user and task in order to derive a model to capture resource estimation and utilization patterns. The derived model assists in understanding the relationship between users and tasks within workload, and enables further work such as resource optimization, energy-efficiency improvements, and failure correlation. Additionally, it provides a mechanism to create patterns that randomly fluctuate based on realistic parameters. This is critical to emulating dynamic environments instead of statically replaying records in the trace log. Our approach is evaluated by contrasting the logged data against simulation experiments, and our results show that the derived model parameters correctly describe the operational environment within a 5% of error margin, confirming the great variability of patterns that exist in Cloud computing.

3. Existing System

In the existing system, the examination takes place with limited cloud traces from Google and yahoo to provide mechanisms to analyze and categorize workload patterns. In this system the main objective it to obtain coarse grain statistical data about jobs and tasks to classify them by duration. This characteristic limits the work's application to the study of timing problems. It makes it unsuitable to analyze the cloud computing issues related to resource usage patterns. In this system, group jobs with similar characteristics using clustering to analyze the resulting centroids. Unfortunately there is a lack of analyses to support the development of workload models that capture the inherent diversity of users and tasks, largely due to the limited availability of Cloud trace logs as well as the complexity in analyzing such systems.

Here they develop Cloud computing workload classifications based on task resource consumption patterns. The existing approach identifies workload characteristics, constructs the task classification, identifies the qualitative boundaries of each cluster and then reduces the number of clusters by merging adjacent clusters. This approach is useful to create the classification of tasks, but does not perform an analysis of the characteristics of the formed clusters in order to derive a detailed workload model. Finally, it is entirely focused on task modeling, neglecting user patterns.

In this system, it used Intra Cluster analysis algorithm. The cluster analysis and intra–cluster analysis do not contain sufficient detail to quantify the diversity of workload, instead presenting high–level observations. Furthermore, there is insufficient detail about the parameter distributions used; more detail is necessary in order for other researchers to simulate the workload obtained. Finally, the validation of the simulated model against that of the empirical data is based only on a visual match of the patterns from one single execution, and does not consider more rigorous statistical techniques. The following figure shows the existing system architecture:

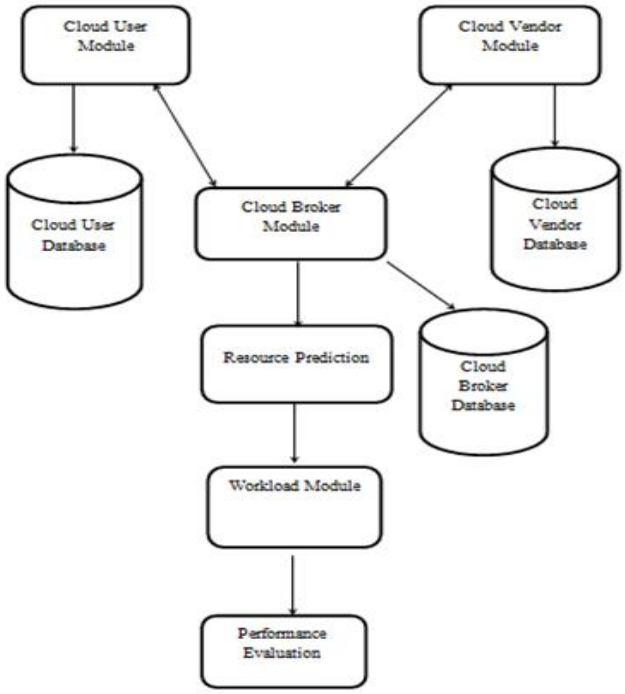

Figure 2: Existing System Architecture

4. Proposed Dynamic VM Allocation Algorithm

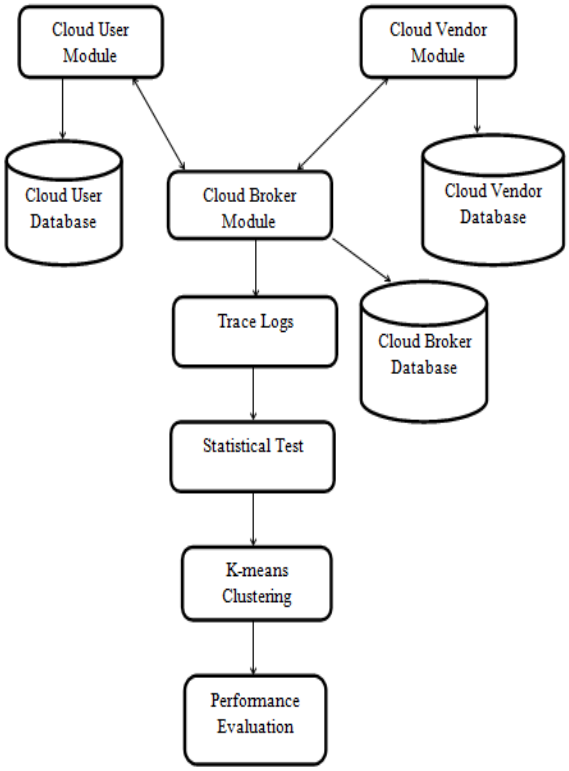

Figure 3: Proposed System Architecture

Clustering is the method which includes the grouping of similar type of objects into one cluster. These clusters include the objects of data set that is chosen in order to minimize some measure of dissimilarity. K-means clustering algorithm is used for scheduling these virtual machines. K-means clustering algorithm is a clustering method in which the given data set is partition into k number of clusters and is well-known for the partitioning method. In this objects are classified as belonging to one of k-groups. The results of clustering method are a set of k partitions where each object of the data set belonging to one cluster. In each partition, there may be a centroid or a cluster representative.

5. Allocation of Dynamic VM using K-means clustering algorithm:

K-means clustering algorithm follows the partitioned or non-hierarchical clustering approach. It involves the partitioning of given data set into the particular number of groups called clusters. Each cluster is associated with a center point called centroid. Each cluster point is assigned to a cluster with the closest centroid.
Proposed dynamic VM allocation algorithm using k-means clustering algorithm is as:

Input:

- List V of Virtual Machine`s with their location around the globe.
- List D of datacenters

Algorithm:

1. Select K points according to the number of datacenters in D
2. Choose datacenter from D
3. Form K clusters of VM`s from V by assigning closest centroid
4. Recomputed the centroid of each cluster
5. Arrange all the requested VM`s in cluster form
6. Allocate the VM`s to the available Host
7. If all the VM`s are allocated
8. Assign the VM`s cluster to the selected datacenter
9. Endif
10. Repeat Step 2 until D is empty
11. If all the VM`s are created in the datacenters
12. Send the cloudlets to the created VM`s
13. Endif
14. Compute the results

The initial centroid is chosen randomly. Centroid is the mean of the points in the cluster. Euclidean distance is used to measure the closeness. K-means clustering generates the different clusters in different runs.

6. Simulation

NetBeans is used to develop the application which is simulated in the CloudSim simulator. WampServer is also used. 20 data centers, 25 data centre brokers and 40 virtual machines were created. The virtual machines are then allocated to the brokers, and 40 jobs or cloudlets were provisioned to these brokers. Normal and speed kind jobs are prioritized. Finally, the resources are collected and clustering is performed.

7. Results

The following results were obtained after the simulation.

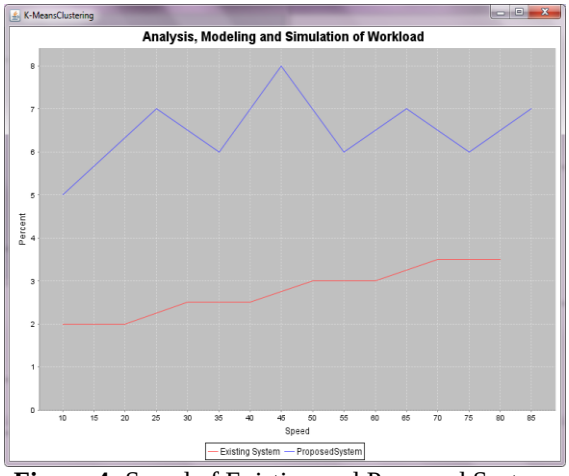

Figure 4: Speed of Existing and Proposed System

Figure 4 shows the graph obtained after simulation. It shows the comparison of speeds of existing and proposed system. From this graph, it can be seen that there is 3-4.5% increase in the speed of the proposed system.

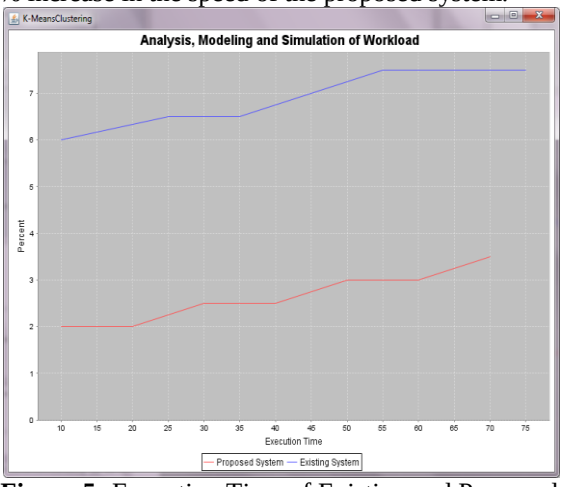

Figure 5: Execution Time of Existing and Proposed System

Figure 5 shows the graph obtained after simulation. It shows the comparison of execution times of existing and proposed system. From this graph, it can be seen that there is 3.5-4% decrease in the execution time of the proposed system. Therefore, this validates that the proposed system is better than the existing system.

8. Conclusion and Future Work

This paper presents an examination that quantifies the diversity of Cloud workloads and derives a workload model from a large-scale construction Cloud datacenter. The obtainable examination and model captures the distinctiveness and behavioral patterns of user and task variability across the entire system as well as different observational periods. The derivative model is implemented by means of the CloudSim construction and comprehensively validated during empirical comparison and statistical tests. From the explanation obtainable within this work and the outcome obtained from the simulations, a number of conclusions can be ended.

Future research includes extending the model to include tasks constraints based on server characteristics; this will allows us to analyze the impact of hardware heterogeneity on workload behavior. Other extensions include analyzing the workload from the jobs perspective specifically modeling the behavior and relationship of users and submitted jobs, accurately emulating and analyzing workload energy consumption and reliability enabling further research into energy-efficiency, resource optimization and failure-analysis in the Cloud environment. Finally, it is important to enable a collaboration link with the CloudSim group in order to integrate the proposed workload generator as an add-in of the current framework implementation allowing it to be made publicly available.

References

[1] Veeramallu, G. K. S. B. 2014. "Dynamically Allocating the Resources Using Virtual Machines." (IJCSIT) *International Journal of Computer Science and Information Technologies.* 5(3):4646-4648

[2] CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services https://code.google.com/p/cloudsim/

[3] El-Refaey, M. A. and Rizkaa, M. A. 2009. "Virtual Systems Workload Characterization: An Overview." *IEEE Intl. Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises.* 72-77.

[4] Kavulya, S., 2010. "An Analysis of Traces from a Production MapReduce Cluster." *IEEE/ACT int. Conf. Cluster, Cloud and Grid Computing.* 94-103.

[5] J. Zhan,., 2010. "PhoenixCloud: Provisioning Resources for Heterogeneous Workloads in Cloud Computing," *arXiv preprint arXiv*:1006.1401.

[6] B. Sharma., 2011. "Modeling and synthesizing task placement constraints in Google compute clusters." *Proc. ACM Symp. on Cloud Computing*. 1-14.

[7] I. Solis Moreno, 2013. "An Approach for Characterizing Workloads in Google Cloud to Derive Realistic Resource Utilization Models," *Int. Symp. on IEEE Service Oriented System Engineering*. 49-60.