Analysis and Simulation of a Virtual Environment for the Organization of Competitions in Programming with Local and Remote Clients

Deyan Atanasov [1], Trifon Ruskov [2] and Vladimir Nikolov [3]

Abstract - **The aim of this paper is to analyze the services provided in a distributed system by establishing a virtual environment, which simulates the system usage, load and performance while conducting a programming contest (National Collegiate Programming Contest) where the teams are local or remote clients.**

Keywords - **Cloud computing, data center, simulation, resource management.**

I. INTRODUCTION

For the past few years there is a significant development in the distributed processing of information, based on the existence of a wide variety of computing environments and platforms. This rapid development provided the foundation for the development of new more flexible architectural models and concepts such as Grid computing and Cloud computing.

Cloud computing is based on virtualization technology and provides its customers with on-demand, reliable, secure and easy access to resources and services. Cloud computing is a dynamic model in which Cloud service providers could sell or resell services to their customers. The dynamics of the model is determined by the fact that the customer uses the service only when needed, and pays only for the period of its use.

Cloud providers offer their services according to several fundamental delivery models [1]:

- Software as a Service (SaaS);
- Platform as a Service (PaaS);
- Infrastructure as a Service (IaaS).

IaaS is the delivery model that we are simulating and describing in this paper. It is a self-service model for accessing, monitoring, and managing remote datacenter infrastructures, such as computer hardware (servers, networking technology, storage, and data center space) as a service. IaaS may also include the delivery of operating systems and virtualization technology to manage the resources.

[1]Deyan Atanasov is PhD student at the Computer Science and Engineering Department at Technical University of Varna, 1Studentska Str., Varna 9010, Bulgaria, E-mail: dido.paraskevov@tu-varna.bg.

[2]Trifon Ruskov is with the Computer Science and Engineering Department at Technical University of Varna,1Studentska Str., Varna 9010, Bulgaria, E-mail: ruskov@tu-varna.bg.

[3]Vladimir Nikolov is with the Computer Science and Engineering Department at Technical University of Varna,1Studentska Str., Varna 9010, Bulgaria, E-mail: nikolov_vn@tu-varna.bg.

This paper is organized as follows. Related problem and used instruments are described in Section II. Section III presents the experiment scenario. The results from the experiments are given in Section IV. The conclusion and future work are specified in Section V.

II. PROBLEM BACKGROUND AND USED TOOLS

Our simulation experiment aims to analyze the services provided in a distributed system by establishing a virtual environment. This virtual environment simulates the system usage, workload and performance while conducting a programming contest (National Collegiate Programming Contest), where the teams are local or remote clients. The problem arose some time ago and during the last few editions of this competition the organizers experienced some technical issues concerning load balance and system performance. Our department hosted the last contest, so we had the opportunity to examine and analyze the parameters which affected the performance. By this analysis we have obtained the values of some important parameters concerning system and network performance. Parameters such as size of tasks, amount of data to transfer, network bandwidth, etc. were taken into account and used in establishing the simulating environment.

To build such a virtual environment we used the open source simulation instrument CloudSim [2, 3]. CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud computing systems and environments. We decided to use CloudSim for our simulation model because of its time efficiency, flexibility and applicability to different Cloud concepts. Some specific to Cloud environments functionalities that CloudSim can provide are simulating of large Cloud environments (data centers) on a single physical compute node and simulating a federation of clouds (multiple inter-networked clouds) [4].

Our simulation of distributed computing environment is also based on the use of NetBeans IDE [5] for the development and deployment of our scenario. We used a web-based instrument called CloudReports to show the results from the experiment. This is an open source tool with a graphical interface using the CloudSim kernel.

III. SIMULATION SCENARIO

A distributed system is used to support the learning process at the Computer Sciences and Engineering Department at the Technical University of Varna. This system also aims to provide various other online educational services. It consists of three blade servers and each of them has the following parameters:

- 2 x CPU Intel XEON E5-2600v2 6 core 2 GHz;
- 32GB RAM;
- 2 x HDD 146GB 15K rpm;
- 10GbE network adapter.

These characteristics of the physical distributed infrastructure are used as parameters in our simulation environment. Thus we obtained the opportunity to analyze and compare the simulation results with the real distributed system. The simulation environment receives and processes requests and tasks that match with the tasks sent by the participating teams in the real programming contest.

For this purpose, the simulation model is based on the configuration of a datacenter, containing three servers (hosts) with the parameters of the actual blade servers. The datacenter is used by both local and remote clients. These clients are sending tasks (cloudlets) to the virtual machines running on the datacenter. The cloudlet is a class that models the Cloud-based application services (such as content delivery, social networking, and business workflow) and represent the usage Cloud-based applications and services [2, 3]. Cloudlets have predetermined parameters such as size, amount of data to transfer and length of instructions measured in millions of instructions per second (MIPS). After analyzing the results from the actual contest, these parameters were set with values that correspond to the actual tasks from the programming competition.

Cloudlets use the computing resources of the deployed virtual machines via one of the following scheduling policies. They are space-shared, time-shared and dynamic workload:

1. Space-shared scheduling policy – by implementing this policy, the virtual machines executes the tasks one by one. When a cloudlet is being executed, all the other cloudlets are waiting in standby mode;
2. Time-shared scheduling policy – by implementing this policy, each virtual machine executes several tasks simultaneously. In time-shared policy, each task is using the virtual machine for a certain period of time after which access to VM resources is given to the next task;
3. Dynamic workload policy – this policy allows dynamic resource load generation, which means dynamic usage of the virtual machines resources. Under this policy, each cloudlet is given a VM with available resources.

Other important parameters that we take into account for the purpose of the experiment are:

- Network bandwidth;

- Packets transfer delay – parameter which represents the network latency that a message can experience. We use this parameter for data sent from remote clients. In this model, there are no actual entities available for simulating network equipment, such as routers or switches. Instead, network latency that a message can experience on its path from one CloudSim entity to another is simulated;

- Resource allocation method - resource allocation (VM allocation) is the process of creating virtual machine instances on hosts that meet the characteristics and configuration requirements of the Cloud service providers in order to achieve certain levels of quality of service (QoS). The default allocation method used is First-Come, First-Served (FCFS) [6]. This is a well known allocation policy in which items are processed in order of arrival.

We deployed a total of 30 virtual machines, configured with single-core processors, performance 1000 MIPS and 1GB of RAM in the datacenter. The VMs are 30, because there were 30 teams in the actual programming contest (every competitor used 1 VM). Remote clients (competitors) used 20 of them and the other 10 were used by the local clients (competitors). The simulation test was 30 min long. The analysis of the actual competition showed that the distribution of executed tasks represents a normal (Gaussian) distribution with displacement of the maximum to the last 30 minutes of the race. Therefore, our simulation time interval is chosen to match the final 30 minutes of the contest, when the workload and the number of client requests are significant.

IV. Results

The results of the simulation experiment are shown in Fig. 1, Fig. 2 and Fig. 3. They show the workload and resource utilization of the simulated infrastructure during the simulation. For the purposes of the simulation we used all three scheduling policies. The best results were obtained by using the Dynamic workload scheduling policy, therefore Fig.1-3 are representing the usage of this policy.

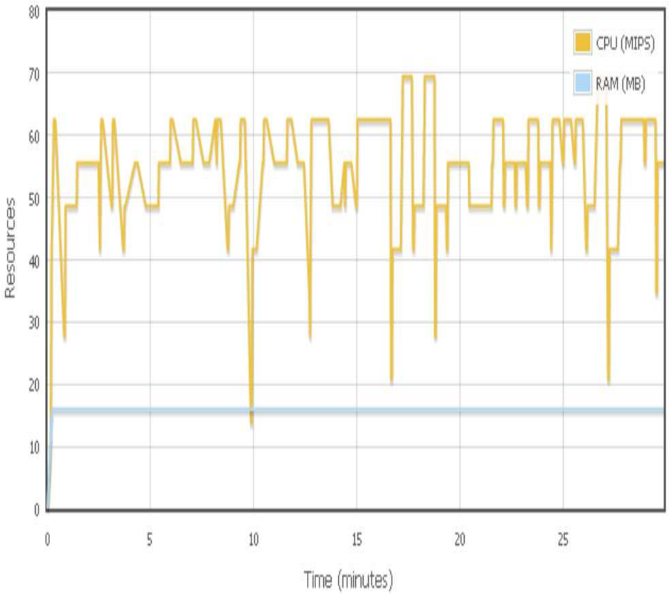

Fig.1. Used resources in Host 1.

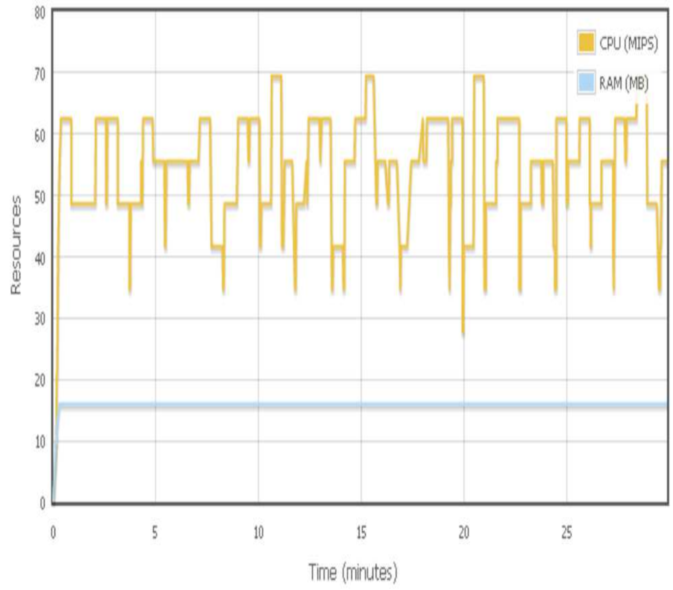

Fig.2. Used resources in Host 2.

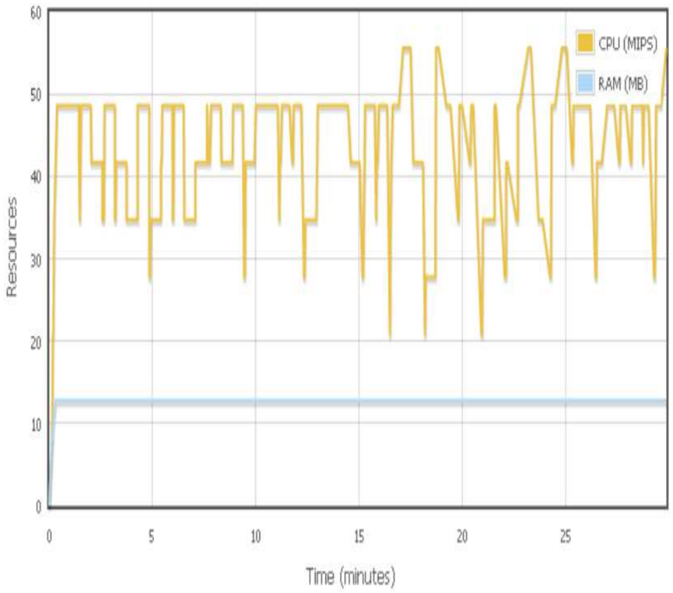

Fig.3. Used resources in Host 3.

The obtained results indicate the following:

1. The Dynamic workload scheduling policy gives the best results in terms of uniform (balanced) load of servers and VMs;

2. Used resources in the datacenter, are distributed evenly among the servers (hosts);

3. The simulated infrastructure has enough free resources and can process applications and tasks from a much larger number of customers, both local and remote;

4. During the simulation, the average number of processed tasks by remote clients is smaller than the average number of task processed by the local clients, but the difference is negligible. Remote clients processed 660 cloudlets, which is an average of 33 cloudlets per VM. Local clients executed a total of 378 cloudlets, which is an average of 37,8 cloudlets/VM. The number of tasks processed by each virtual machine is shown in Fig.4.

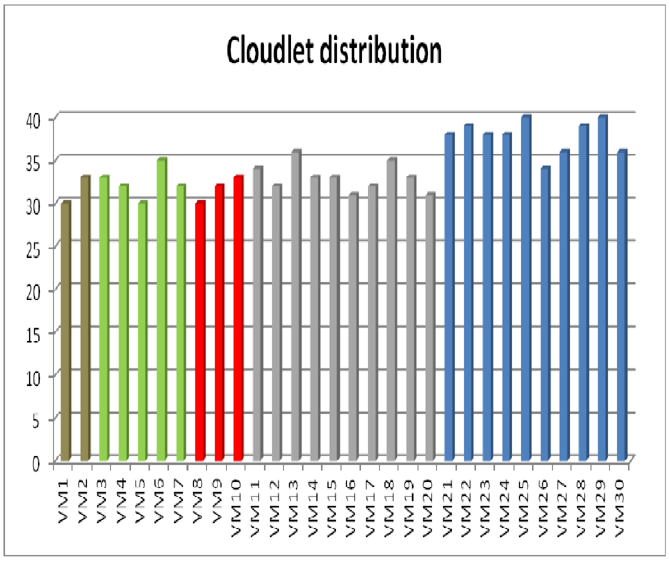

Fig.4. Distribution of executed tasks (cloudlets) by VM

Different colors in the graph are corresponding to different packet transfer delays. This delay is used to simulate different remote clients in the simulation. We have set the virtual machines (VM1-VM20) with the following packet transfer delays:

VM1 and VM2 – 15ms;
VM3 to VM7 – 20ms;
VM8 to VM10 – 18ms;
VM11 to VM20 – 10ms.

VMs with numbers 21-30 represent the local clients in the simulation. They all have been set with packet delay of 1ms.

5. The number of cloudlets executed by the VMs is much greater than the number of tasks from the real contest, which indicates that the infrastructure is able to cope with the workload and the large number of requests during the competition.

V. CONCLUSION AND FUTURE WORK

The conducted in this work experiments show the applicability of CloudSim for simulation related to the efficiency of processing applications and tasks from local and remote clients. Such a configuration of the system would handle the workload in this type of competitions.

As a guideline for future work on the simulator may be development and implementation of different allocation algorithms and scheduling policies. Such developments would give us the opportunity to compare and analyze different methods and Cloud scenarios using wide variety of policies and algorithms in order to achieve high level of QoS and performance in distributed systems.

REFERENCES

[1] P. Mell, T. Grance, "The NIST Definition of Cloud Computing", NIST Special Publication 800-145, 2011.

[2] R. Calheiros, R. Ranjan, C. A. F. DeRose, R. Buyya, "Cloudsim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services", 2009.

[3] R. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. DeRose, R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Software: Practice and Experience (SPE), Volume 41, Number 1, pp: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, 2011.

[4] R. Buyya, R. Ranjan, "InterCloud: Utility-oriented Federation of Cloud Computing Environments for Scaling of Application Services", Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing, pp:328-336, Busan, South Korea, 2010.

[5] NetBeans IDE, Available at: https://netbeans.org/.

[6] Dictionary of Algorithms and Data Structures, Available at: http://www.nist.gov/dads/HTML/firstcome.html.