Int. Journal of Electrical, Electronics, Computer Engineering and Technology www.teslapub.com
Vol. 01, Issue 01, January- 2020, pp. 15-19

RESEARCH ARTICLE OPEN ACCESS

A NEW TASK SCHEDULING APPROACH FOR CLOUD BASED APPLICATION

Ashtosh Pangasa*, Antony Vasanth*
***Cardiff University, UK**
Email: pangasaashutosh@gmail.com

ABSTRACT

To achieve high performance, thousands of servers in cloud datacenters coordinate tasks to provide reliable and highly available cloud computing services. Task scheduling plays a key role in cloud computing systems. Scheduling of tasks cannot be done on the basis of single criteria but under a lot of rules and regulations. Scheduling is nothing but the mapping of tasks and resources in accordance with some certain principles for achieving the desired goal. Cloud computing paradigm simplifies the mapping of tasks to resources; the required resources together form to be virtual machines (VMs), the process of search the desired resource package is same as the process of searching the various VMs. The scheduling of tasks in cloud means choosing the best suitable resource available for execution of tasks or to allocate computer machines to tasks in such a manner that the completion time is minimized as possible. The tasks are assigned based on scheduling algorithm we use. And the task then chooses their available Data Centers, Hosts and VMs.

Keywords: Virtual Machines, Service-Oriented Architecture, Datacenter, Round-Robin, Earliest Start time

1. INTRODUCTION

Cloud computing is a computing term or metaphor that evolved in the late 2000s, based on utility and consumption of computing resource Cloud computing involves deploying groups of remote servers and software networks that allow centralized data storage and online access to computer services or resources. Cloud computing is typically defined as a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications.

In cloud computing, the word cloud (also phrased as "the cloud") is used as a metaphor for "the Internet," so the phrase cloud computing means "a type of Internet-based computing," where different services — such as servers, storage and applications —are delivered to an organization's computers and devices through the Internet. Cloud computing enables companies to consume compute resources as a utility -- just like electricity -- rather than having to build and maintain computing infrastructures in-house.

Cloud computing is the result of evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. The cloud aims to cut costs, and helps the users focus on their core business instead of being impeded by IT obstacles.

The main enabling technology for cloud computing is virtualization. Virtualization software separates a physical computing device into one or more "virtual" devices, each of which can be easily used and managed to perform computing tasks. With operating system–level virtualization essentially creating a scalable system of multiple independent computing devices, idle computing resources can be allocated and used more efficiently. Virtualization provides the agility required to speed up IT operations, and reduces cost by increasing infrastructure utilization. Autonomic computing automates the process through which the user can provision resources on-demand. By minimizing user involvement, automation speeds up the process, reduces labor costs and reduces the possibility of human errors.

Users routinely face difficult business problems. Cloud computing adopts concepts from Service-oriented Architecture (SOA) that can help the user break these problems into services that can be integrated to provide a solution. Cloud computing provides all of its resources as services, and makes use of the well-established standards and best practices gained in the domain of SOA to allow global and easy access to cloud services in a standardized way.

Task scheduling plays a key role to improve flexibility and reliability of systems in cloud. The main reason behind scheduling tasks to the resources in accordance with the given time bound, which involves finding out a complete and best sequence in which various tasks can be executed to give the best and satisfactory result to the user. In cloud computing, resources in any form i.e. cups, firewall, network are always dynamically allocated according to the sequence and requirements of the task, subtasks. So, this leads task scheduling in cloud to be a dynamic problem means no earlier defined sequence may be useful during processing of task. The reason behind the scheduling to be dynamic is that because flow of task is uncertain, execution paths are also uncertain and at the same time resources available are also uncertain because there is a number of tasks are present that are sharing them simultaneously at the same time. The scheduling of tasks in cloud means choose the best suitable resource available for execution of tasks or to allocate computer machines to tasks in such a manner that the completion time is minimized as possible. In scheduling algorithms list of tasks is created by giving priority to each and every task where setting of priority to different tasks can be based on various parameters. Tasks are then choosing according to their priorities and assigned to available processors and computer machines which satisfy a predefined objective function.

The task scheduling problem can be viewed as the finding or searching an optimal mapping/assignment of set of tasks over the available set of resources (Data Centers, Host and VM) so that we can achieve the desired goals for tasks. Cloud computing uses virtualization technique for mapping the resources of cloud to the virtual machine layer, implement the user's task, so the task scheduling of cloud computing environment achieves at the applications layer and the virtual layer of resources. In this paper we are performing comparative study for enhancing their suitability, feasibility, adaptability in the context of cloud scenario, after that we try to propose an approach that can be adopted to enhance the existing platform further.

The paper has been organized as follows, section 2 discussed the important research findings about task scheduling in cloud environment, section 3 introduces system modules and description. Section 4 discussed various types of sensing. Finally the paper has been concluded in section 5.

2. RELATED WORKS

The efficient allocation strategy of the independent computational jobs among different Virtual Machines (VM) in a Datacenter (DC) is a distinguishable challenge in the Cloud Computing domain and finding out an optimal job allocation strategy guided by a good scheduling heuristic for such an environment is an NP-complete problem. So different heuristic approaches may be used for better result. The related works with other NP-complete problems have shown that solutions guided by heuristic approaches can often be improved by applying local scheduling procedure for allocating independent jobs in the virtual machines (VM) inside a Datacenter (DC), which, when combined with fast construction heuristics, can find out the shorter schedules on benchmark problems than the other solution methodologies found in different literatures, and in significant less time. This paper [1] highlights a smart job allocation strategy for a CSP by applying Round-Robin (RR) scheduling policy.

Job scheduling is one of the major issues in cloud computing environment. In job scheduling priority is the biggest issue because some jobs need to be scheduled first then all other remaining jobs which can wait for a long time. In this paper [2], proposed an improvement in priority-based job scheduling algorithm in cloud computing which is based on multiple criteria and multiple attribute decision making model. In this work [3], the impact of First Come First Served and Earliest Start time (EST) based Job scheduling are compared. It is shown that when EST based Job Scheduling is used, priorities can be set on Cloudlets execution based on their EST. After the scheduling of Cloudlets, Single Threshold based VM provisioning or Two Threshold based VM provisioning is done at the Cloud Provider. In the proposed Two Threshold and Single Threshold based VM provisioning algorithms, a cost-effective Datacenter selection policy is used for selecting the Datacenter with minimum cost. It can be shown that Service Level Agreement (SLA) violations can be minimized by using Two Threshold based VM provisioning algorithm when compared to Single Threshold based VM provisioning algorithm. It is observed that profit of Cloud Service is increased by Two Threshold based VM provisioning algorithm. Cloud computing will be a major technology in the development of the future Internet of Services. Service providers want to remove the bottle neck of the cloud computing system in order to satisfy user requirement. And in order to save energy consumption they also need to apply new energy-efficiency mechanism and observe its effects on a datacenter. As it is difficult to test new mechanism in real cloud computing environment and researchers often cannot reach the real cloud environment, simulation to model the mechanism and evaluate the results is necessary. Simulating a datacenter avoids spending time and effort to configure a real testing environment. Moreover, as real machines are not used for testing purposes, their computational power can be allocated to profit-making applications. In paper [4] introduces a simulation framework called CloudSim which provides simulation, power to manage services and modeling of cloud infrastructure.

3. SYSTEM MODULE DESCRIPTION

3.1 INTERNAL CLOUD

Every User can enter the File Size that needs to be scheduled, if it is below certain limit it gets scheduled within the local host itself. If it is above the limit, then the task is given to the cloud for scheduling.

3.2 AGENT:

Agent consists of three parameters: Request, Response and Latency (Delay). Request, Response values are generated from proposed scheduling algorithm. Latency is nothing but the difference between Request and Response depending on which the task gets assigned to each data centers. It also displays the Scheduling path of the task. The path has Data Center, Host and VM. In addition to that it also has the Configuration of each Data Centers.

3.3 DATA CENTER & VM SELECTION

The Cloud comprises of various Data Centers, Hosts and VMs. The task gets assigned to the particular Data Center based on the Scheduling algorithm. Each Data Centers has several Host and VMs. The task gets scheduled to particular hosts and VMs which is free. Once the scheduling gets completed the processing time of each task gets displayed.

3.4 CODING AND TESTING

Once the design aspect of the system is finalizing the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

4. SYSTEM TESTING

Testing is performed to identify errors [5-8]. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected [9-10]. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the work testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

4.1 TESTING TECHNIQUES / TESTING STRATERGIES

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet-undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

4.2 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

4.3 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

4.4 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

4.5 INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

4.6 PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

4.7 SECURITY TESTING:

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

4.8 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted.
- A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this work is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus, the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

4.9 USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

5. CONCLUSION

We have used CloudSim to simulate a cloud computing datacenter which avoids spending time and effort to configure a real testing environment. We have presented a task scheduling approach to reduce processing time in order to improve the reliability and performance of cloud. As cloud is a business-oriented service, it must concern about shorter processing time. Thus, tasks are scheduled without waiting in a queue so the processing time gets decreased much efficiently. Task scheduling is one of the most famous problems in cloud computing. So, there is always a chance of modification of previously completed work in this particular field. In future it would be better to schedule the task by uploading files and observe the processing time and simulate various results using CloudSim.

REFERENCES

[1]. Sourav Banerjee, Mainak Adhikari, Utpal Biswas, “Development of a Smart Job Allocation Model for a Cloud Service Provider” 2014 2nd International Conference on Business and Information Management (ICBIM).

[2]. Swachil J. Patel, Upendra R. Bhoi, “Improved Priority based Job Scheduling Algorithm in Cloud Computing using Iterative Method” 2014 Fourth International Conference on Advances in Computing and Communications.

[3]. Chinnu Edwin A, A.Neela Madheswari, “Job Scheduling and VM Provisioning in Clouds” 2013 Third International Conference on Advances in Computing and Communications.

[4]. Wang Long, Lan Yuqing and Xia Qingxin,” Using CloudSim to Model and Simulate Cloud Computing Environment” 2013.

[5]. Shohei Gotoda, Minoru Ito, “Task scheduling algorithm for multicore processor system for minimizing recovery time in case of single node fault,” Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium. pp. 260-267, Ottawa.

[6]. Jiadong Yang, Hua Xu, LiPan, Peifa Jia, Fei Long, Ming Jie, ”Task scheduling using Bayesian optimization algorithm for heterogeneous computing environments”, Applied Soft Computing. Volume 11, Issue 4, June 2011, pp. 3297–3310.

[7]. Siva Theja Maguluri and R. Srikant, "Stochastic Models of Load Balancing and Scheduling in Cloud Computing Clusters", 2012 Proceedings IEEE Infocom.

[8]. Cloudsim, “A Framework for Modeling And Simulation of Cloud Computing Infrastructures And Services”, https://code.google.com /p/ cloudsim /downloads/list.

[9]. Shamsollah Ghanbari and Mohamed Othman, “A Priority based Job Scheduling Algorithm in Cloud Computing”, International Conference on Advances Science and Contemporary Engineering. ICASCE 2012, Volume 50, 2012, pp. 778–785.

[10]. Nguyen Doan Man and Eui-Nam Huh, “Cost and Efficiency-based Scheduling on a General Framework Combining between Cloud Computing and Local Thick Clients”, International Conference on Computing, Management and Telecommunications. (ComManTel), Jan. 2013, Ho Chi Minh City, Vietnam, pp. 258 - 263.