International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 85-88 | Cosmos Impact Factor (Germany): 5.195
Received: 24.03.2018 Published : 01.04.2018

EFFICIENT LOAD BALANCER USING MACHINE LEARNING AND HEURISTICS

[1]S Sridhar, [2]Saikruth Rao, [3]Ayush Sanghi
[1]Assistant Professor, Department of Computer Science and Engineering,
SRM Institute of Science and Technology
[2,3,]UG Scholars, Department of Computer Science and Engineering,
SRM Institute of Science and Technology
Corresponding Author: saikruthr@gmail.com

ABSTRACT

Cloud computing allows business customers to scale up and down their resource usage based on needs. Many of the gains in the cloud model are achieved from resource multiplexing through virtualization technology. In this paper, we present a system that uses virtualization technology based on machine learning algorithms and heuristic functions to allocate data center resources dynamically based on application demands and support green Computing by optimizing the number of servers in use. We introduce the concept of "Time Slot Filtering" to measure the unevenness in the multidimensional resource utilization of a server. By minimizing time, we can combine different types of workloads and improve the overall utilization of server resources. We develop a set of heuristics that prevent overload in the system effectively while saving energy used.

Index Terms- Load balancing, Time Slot Filtering, Virtualization, Heuristics.

INTRODUCTION

BACKGROUND

Service capacities are usually considered to be unlimited in cloud computing, which can be used any time. However, from the CSP's perspective, service capacities are limited. Available service capacities change with workloads, i.e., they cannot satisfy user's requests at any time when a cloud service is shared by multiple tasks. Only some available time slots are provided for new coming users by CSPs in terms of their remaining capacities.

For example, each activity has different candidate services with various execution times, costs and available time slots. For activity 4, there are two candidate services with different workloads. If service 1 is selected for activity 4, the execution time is 4 with the price 6 and available time slots. Time slot is unavailable because there is no remaining capacity. The considered WSDT problem is similar to the Discrete Time/Cost Trade-off Problem (DTCTP) to some extent.

We can modify existing algorithms for the latter to the problem under study with less than 200 activities and no more than 20 candidate services in the service pool, spending thousands of seconds. However, the number of activities is usually far more than 200 in practical workflow applications, which makes the modified versions, and are not suitable for the problem under study.

The problem was modeled as we proved they had different properties. The ILAH (iterated local adjusting heuristic) framework was proposed. Three initial solution construction strategies were developed.

Two improvement strategies were introduced which had similar effects on the solution improvement. By integrating the worst and best construction strategies with two improvement strategies, four based algorithms were developed. However, the worst initial solution construction strategy was strange that showed the best performance. However, they obtained the worst performance. In addition, this was not sensitive to parameters while were affected by most of the parameters.

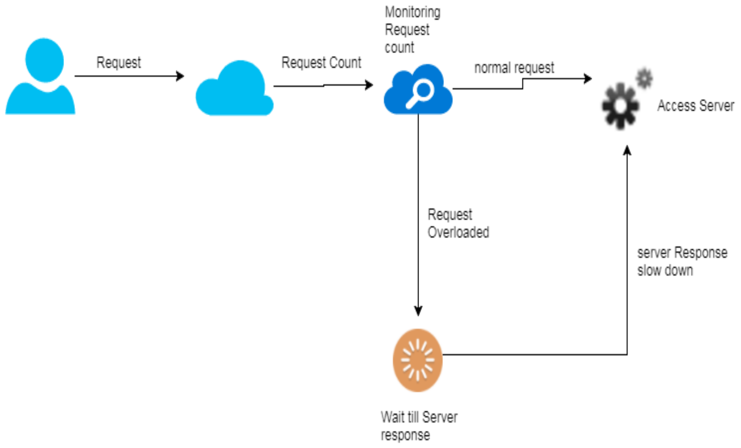
RELATED WORK

Most existing methods for workflow scheduling in cloud computing consider only task constraints (e.g., deadlines) from the perspective of users. Services are rented with an interval-based pricing model. Rented intervals are exclusively reserved and owned by users, i.e., cloud resources (services) are assumed unlimited during these intervals. Capital Presented service scheduling with start time constraints in distributed collaborative manufacturing systems. They modeled this problem as a Discrete Time-Cost Tradeoff Problem with Start Time Constraints (DTCTP-STC) and proved it NP-hard.
Service capacities are usually considered to be unlimited in cloud computing, which can be used at any time. However, from the CSP's perspective, service capacities are bound. Available service capacities change with workloads, i.e., they cannot satisfy user's requests at any time when a cloud service is shared by multiple tasks. Only some available time slots are provided for new coming users by CSPs in terms of their remaining capacities.

METHODS

1. Procedure proposed by Brian Kamburowski, Stallman for finding the minimal reductions possible for a given set of data for easy scheduling.
2. Approximation algorithms and heuristic algorithms proposed by Cannakan, Andreas.
3. Dynamic programming and its complexity by Long Chen.

EXISTING DIAGRAM



A dynamic programming algorithm were presented for small instances, considering the traditional service selection problem with time/cost trade off under the workflow deadline constraint. Based on the proposed Critical Path Iterative heuristic (CPI), they considered share service provisioning for workflows in public clouds. The List-based Heuristic considering Cost minimization and Match degree maximization heuristic were proposed for increasing utilization, which was based on several priority rules for assigning tasks to rented services.

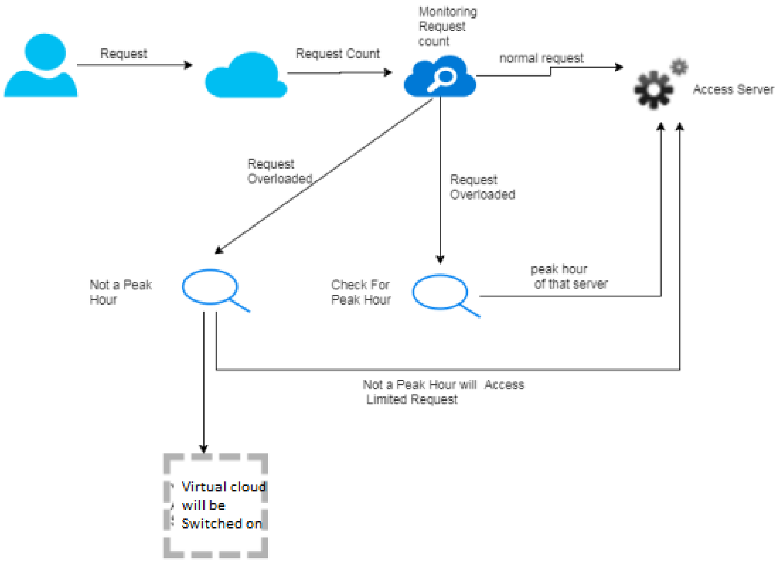
DISADVANTAGES

- ✓ Requests are overloaded.
- ✓ Server Response slow down.
- ✓ Time Consuming.

PROPOSED METHODOLOGY

Based on the proposed Critical Path Iterative heuristic (CPI), share service provisioning for workflows in public clouds was considered. The List-based Heuristic considering Cost minimization and Match degree maximization heuristic were proposed for maximizing utilization, which were based on several priority rules for assigning tasks to rented services. Guided users were used to choose proper type and number of share able services for batch or Message Passing Interface (MPI) tasks. A special type of share able service provisioning problem was considered in , where one task could use only one Virtual Machine (VM) instance. Reserved intervals resulted in time slots for the cloud services from the perspective of resource providers. In addition, there would be some time slots in reserved intervals because the required amount of resources is less than that of the rented resources.

SYSTEM ARCHITECTURE



The proposed system would achieve an important goal namely overload avoidance . The proposed load balancer is created in such a way that a required server can be accessed at any time without any buffering or slow response. A virtual machine is created at the time of overloading which processes the extra requests .The system aims at tackling the overloaded requests in an efficient way while consuming less energy and speeding up the process at the same time.

ALGORITHMS

ILAH(Iterated Local Adjusting Heuristics)-It is an iterative process which consists of four main components namely time slot filtering, initial solution construction, solution improvement and perturbation. It starts with an initial solution pi and improves iteratively until end criteria is satisfied.
Initially the requests are analyzed based on a particular duration. Once this is done, the time frame in which every server is busy is known. Based on this, peak hour is assigned to the respective servers. Hence, we get to know the period where the virtual cloud needs to be activated for functioning. The virtual cloud handles the overloaded requests such that the performance of the system is not impacted in any manner.

CONCLUSION

The proposed method helps in managing the load in the form of requests for faster functioning. The functioning of the system is very simple. Any server in the system during peak hour can be accessed. It helps in reducing the wait time as well as helps in green computing. The server response time is improved.This can be used on large scale networks to improve network trafficking.

REFERENCES

[1] Market Oriented cloud computing: Vision, Hype and reality for delivering IT services as computing utilities (10th IEEE international conference).
[2] Optimal procedures for the discrete time/cost trade off problem in project networks 1994.
[3] Network decomposition based benchmark results for the discrete time–cost tradeoff problem 2005.
[4] Dynamic programming for services scheduling with start time constraints in distributed collaborative manufacturing systems October 14-17 2012.