Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 6 (2015), pp. 4111-4117

http://www.ripublication.com

Fault-Open Minded Resource Allocation Rescheduling Algorithm and Expense Minimization for Cloud Systems

Albert Mayan. J[1], Yovan Felix. A[2], Manideep Chintalapudi [3], Balaji R[4], Vigneshwaran R [5]

[1, 2]Associate Professor, Department of Computer Science and Engineering, Sathyabama University, Chennai-119
[3, 4, 5] Student, Department of Computer Science and Engineering, Sathyabama University, Chennai-119, India.
E-mail: [1]albertmayan@gmail. com, [2]yovanfelix@gmail. com, [3]manideepec@gmail. com, [4]balajirajendran111@gmail. com, [5]vicky. vigneshwaran2@gmail. com

Abstract

Cloud computing allows business customers to scale up and down their resource usage based on needs. Many of the touted gains in the cloud model come from resource multiplexing through virtualization technology. In this a system that used virtualization technology to allocate data center resources dynamically based on application demands and support by optimizing the number of servers is proposed. Also the concept of "skewness" is introduced to measure the unevenness in the multidimensional resource utilization of a server. By minimizing skewness, different types of workloads are combined nicely and the overall utilization of server resources is improved. A set of heuristics is developed that prevent overload in the system effectively while saving energy used. The resource allocation in cloud computing is much more complex than in other distributed systems like Grid computing platform. In our cloud model, any task will be executed on one or more virtual machines with user-reserved resources and the payment is calculated based on the customized resource. Adopting such a pricing policy is driven by three reasons. First, the efficiencies of many applications usually rely on multiple resources but it is nontrivial to precisely evaluate the exact amount of their consumption separately on individual resources. Second, quite a few users prefer to reserving resources for tolerating usage burst and guaranteeing their service levels. Lastly, the alternative pricing policy, pay-as-you-consume, is rather simple because its payment is always fixed regardless of the resource allocation.

Key words: Resource allocation, skewness, cost minimization.

1. Introduction

Cloud computing has become a new age technology that has got huge potentials in enterprises and markets. Clouds can make it possible to access applications and associated data from anywhere. Companies are able to rent resources from cloud for storage and other computational purposes so that their infrastructure cost can be reduced significantly. Further they can make use of company-wide access to applications, based on pay-as-you-go model. Hence there is no need for getting licenses for individual products. However one of the major pitfalls in cloud computing is related to optimizing the resources being allocated. Because of the uniqueness of the model, resource allocation is performed with the objective of minimizing the costs associated with it. The other challenges of resource allocation are meeting customer demands and application requirements. The customers can dynamically provision virtual servers (i. e., computing instances) in EC2, and then the customers are charged by Amazon on a pay-per-use basis. EC2 offers three options to provision virtual servers, i. e., on- demand, reservation, and spot options. Each option has different price and yields different benefit to the customers. Spot price (i. e., price of spot option) could be the cheapest; however, the spot price is fluctuated and even more expensive than the prices of on-demand and reservation options due to supply-and-demand of available resources in EC2. Although the reservation and on-demand options have stable prices, their costs are mostly more expensive than that of spot option. The challenge is how the customers efficiently purchase the provisioning options under uncertainty of price and demand. To evaluate the performance of the algorithms, numerical studies are extensively performed. The results show that the algorithms can significantly reduce the total provisioning cost incurred to customers.

1. 1 Domain Description

Cloud computing is at an early stage, with a motley crew of providers large and small deliveringa slew of cloud-based services, from full-blown applications to storage services to spam filtering. Yes, utility-style infrastructure providers are part of the mix, but so are SaaS (software as a service)providers such as Salesforce. com. Today, for the most part, IT must plug into cloud-based services individually, but cloud computing aggregators and integrators are already emerging. Based on those discussions, here's a rough breakdown of what cloud computing is all about:

SaaS:

This type of cloud computing delivers a single application through the browser to thousands of customers using a multitenant architecture. On the customer side, it means no upfront investment in servers or software licensing; on the provider side, with just one app to maintain, costs are low compared to conventional hosting.

Utility computing:

The idea is not new, but this form of cloud computing is getting new life from Amazon. com, Sun, IBM, and others who now offer storage and virtual servers that IT can access on demand. Early enterprise adopters mainly use utility computing for

supplemental, non-mission-critical needs, but one day, they may replace parts of the datacenter.

Web Services in Cloud:
Closely related to SaaS, Web service providers offer APIs that enable developers to exploit functionality over the Internet, rather than delivering full- blown applications. They range from providers offering discrete business services.

Platform as a service:
Another SaaS variation, this form of cloud computing delivers development environments as a service. You build your own applications that run on the provider's infrastructure and are delivered to your users via the Internet from the provider's servers.

2. SYSTEM ANALYSIS

2. 1 EXISTING SYSTEM

Traditional job scheduling is often formulated as a kind of combinatorial optimization problem or queue based multiprocessor scheduling problem due to non-guaranteed performance isolation for multiple tasks running on same machines. That is most of the existing deadline driven task scheduling solutions are strictly subject to queuing model under which a single machine's multiple resources cannot be further split to smaller fractions at will. This will eventually cause the raw grained resource allocation, relatively low resource utilization and suboptimal task execution efficiency. Users may wish to minimize the payment when guaranteeing their service level such that their tasks can be finished before deadlines. But this is not guaranteed here.

Drawbacks of Existing System
Users may wish to minimize their payments when guaranteeing their service level such that their tasks can be finished before deadlines. Such a deadline-guaranteed resource allocation with minimized payment is rarely studied in literatures. Moreover, inevitable errors in predicting task workloads will definitely make the problem harder.

2. 2 Proposed System

In the proposed system a deadline-driven resource allocation problem based on the cloud environment facilitated with VM resource isolation technology is used and also a solution to minimize the users' payment in terms of their expected deadlines is proposed. Further an error tolerant method is proposed to find whether the tasks are completed within its deadline. Also the concept of "skewness" is introduced to measure the unevenness in the multidimensional resource utilization of a server.
By minimizing skewness, different types of workloads are combined nicely and the overall utilization of server resources is improved. A set of heuristics is developed that prevent overload in the system effectively while saving energy used.

Advantages of Proposed System

All the theoretical conclusions are confirmed with our experiments. Specifically, in the situation with relatively sufficient resources, the worst case tasks under the stricter deadline-based allocation only take as about 0. 75 times as their deadlines to complete, as compared to the 1. 2 times of the deadlines under the original user-predefined deadline based allocation. We also observe that in the competitive environment, the latter algorithm performs much more stable than the former instead, which means that the latter tolerates the resource competition better. We also confirm the effectiveness of our solution via the distribution of the number of tasks with respect to execution times and user payments: in the competitive situation, majority of tasks can be guaranteed to be completed within deadlines.

Algorithm:

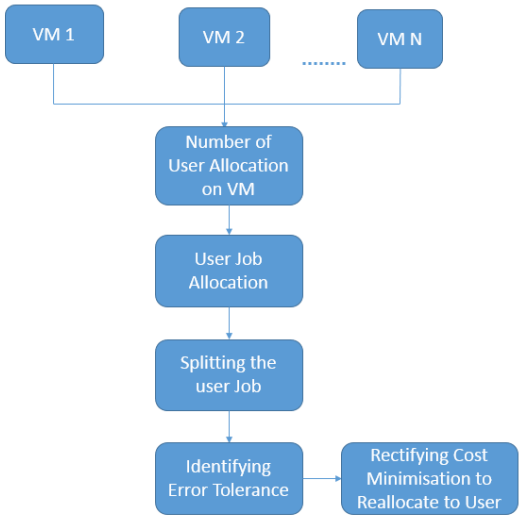

Fig 1: System Architecture

2. 3 System Implementation Modules

1. Network Model
2. Optimal Resource Allocation
3. Predicting Error Tolerance
4. Scheduling and Payment Minimization

Network Model:

Multiple VMs can be started and stopped dynamically on a single physical machine to meet accepted service requests. Hence providing maximum flexibility to configure various partitions of resources on the same physical machine to different specific requirements of service requests. In addition, multiple VMs can concurrently run applications based on different operating system environments on a single physical

machine since every VM is completely isolated from one another on the same physical machine.

Optimal Resource Allocation:
The resource allocation in a cloud computing environment can be modeled as allocating the required amount of multiple types of resource simultaneously from a common resource pool for a certain period of time for each request. The allocated resources are dedicated to each request. Upon receiving the request, the scheduler checks the recollected availability states of all candidate nodes, and estimates the minimal payment of running the task within its deadline on each of them. The host that requires the lowest payment will run the task via a customized VM instance with isolated resources.

Predicting Error Tolerance:
Users may wish to minimize their payments when guaranteeing their service level such that their tasks can be finished before deadlines. An error tolerant method is proposed to find whether the tasks are completed within the deadline. Moreover, inevitable errors in predicting task workloads will definitely make the problem harder.

Scheduling and Payment Minimization:
To improve scheduling and to measure the unevenness in the multidimensional resource utilization of a server the concept of "skewness" is introduced. By minimizing skewness, different types of workloads are combined nicely and the overall utilization of server resources is improved. A set of heuristics is developed that prevent overload in the system effectively while saving energy used. Also payment minimization is provided to minimize the users' payment in terms of their expected deadlines.

2. 4 Experimental Results:
The proposed system is executed in the following step wise procedure.
Firstly, User home is initiated in the application to provide a interface for the user to submit jobs. The next step is followed by Scheduler home, which provides the interface to select the Resource creation, Scheduling, Prediction process or Exit.
From the scheduler home the application is redirected to Resource allocation i. e. VM Creation. Finally the jobs all are submitted and redirected to Relative error calculation for the calculation of the scheduling and to measure the unevenness in the multidimensional resource utilization of a server the concept of "skewness".

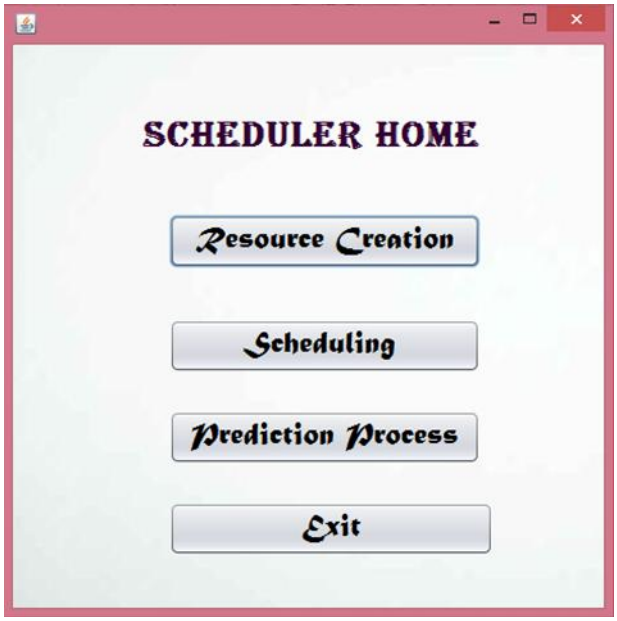

Fig 3: Scheduler Home

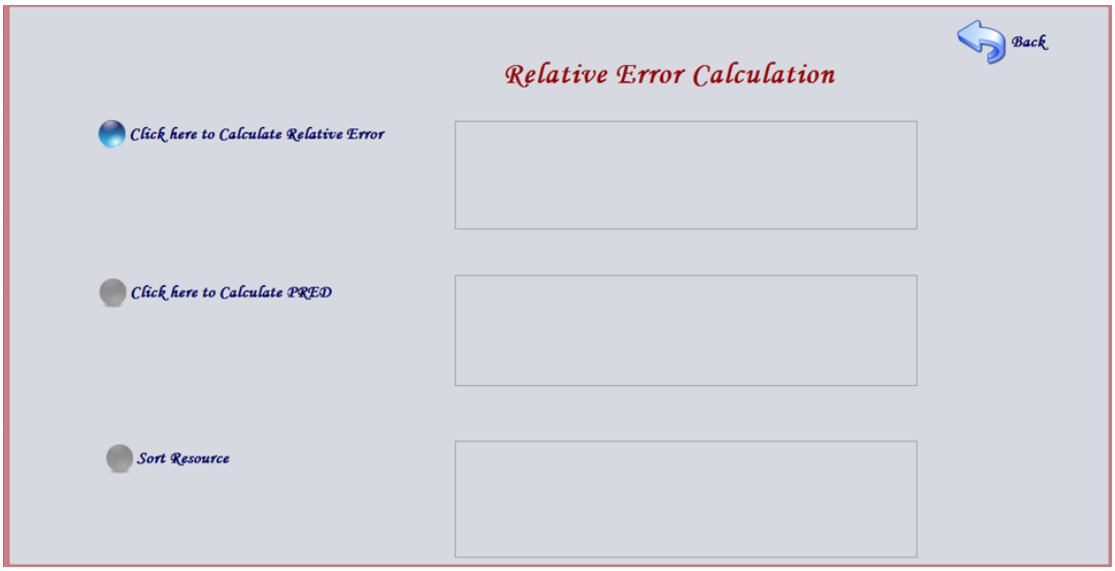

Fig4: Relative Error Calculation

3. Conclusion

This paper proposes a novel resource allocation algorithm for cloud system that supports VM multiplexing technology, aiming to minimize user‘s payment of task and also endeavor to guarantee its execution deadline meanwhile. This is proven that the output of this algorithm is optimal based on the KKT condition, which means any other solutions would definitely cause larger payment cost.

Future Enhancement

In the future, the plan to integrate this algorithms with stricter/original deadlines into some excellent management tools, for maximizing the system wide performance. To improve the resource utilization and reduce the user payment the future work to fix the KKT based on dual parameters like deadline, cost or bandwidth.

References:

1. Armbrust. M, Fox. A, Griffith. R, Joseph. A. D, Katz. R. H, Konwinski. A, Lee. G, Patterson. D. A, Rabkin. A, Stoica. I, and Zaharia. M, (2009), ―Above the Clouds: A Berkeley View of Cloud Computing, ‖ Technical Report UCB/EECS-2009-28, EECS Dept., Univ. California, Berkeley.
2. Chaisiri. S, R. Kaewpuang, B. -S. Lee, and D. Niyato, "Cost Minimization for Provisioning Virtual Servers in Amazon Elastic Compute Cloud, " Proc. 19th Ann. IEEE/ACM Int'l Symp. Modeling, Analysis and Simulation of Computer and Telecomm. Systems (MASCOTS '11), pp. 85-95, 2011.
3. Chang. F, J. Ren, and R. Viswanathan, "Optimal Resource Allocation in Clouds, " Proc. IEEE Int'l Conf. Cloud Computing, pp. 418-425, 2010.
4. Mao. M and M. Humphrey, "Auto-Scaling to Minimize Cost and Meet Application Deadlines in Cloud Workflows, " Proc. Int'l Conf. High Performance Computing, Networking, Storage & Analysis (SC'11), pp. 49:1-49:12, 2011.
5. Meng. X, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, "Efficient Resource Provisioning in Compute Clouds via VM Multiplexing, " Proc. Seventh Int'l Conf. Autonomic Computing (ICAC '10), pp. 11-20, 2010.
6. Nathuji. R, Kansal. A, and Ghaffarkhah. A, (2010), ―Q-Clouds: Managing Performance Interference Effects for Qos-Aware Clouds, ‖ Proc. European Conf. Computer Systems (EuroSys '10), pp. 237-250.
7. Wu. L, S. K. Garg, and R. Buyya, "SLA-Based Resource Allocation for Software as a Service Provider (SAAS) in Cloud Computing Environments, " Proc. 11th IEEE/ACM Int'l Symp. Cluster, Cloud and Grid Computing (CCGRID '11), pp. 195-204, 2011.