CLOUD WORKFLOW SCHEDULING WITH DEADLINES AND TIME SLOT AVAILABIITY

Mr. M.RAMASAMY

S.NANDHINI
B.E CSE
JEI MATHAAJEE COLLEGE OF ENGINEERING
KANCHIPURAM

S.THENMOZHI
B.E CSE
JEI MATHAAJ EE COLLEGE OF ENGINEERING
KANCHIPURAM

ABSTRACT: ***Cloud computing becomes critically important to economic understanding will be more and more popular. To maximize the profit, a service provider should understand both service and business costs, and how they are determined by the characteristics of the applications and the configuration of a Multiserver system. Profit maximization of cloud computing and multiserver configuration of optimal problem in environment will be studied. Our pricing model takes such factors into considerations as the amount of a service, the workload of an application environment, the configuration of a multiserver system, the service-level agreement, the satisfaction of a consumer, the quality of a service, the penalty of a low-quality service, the cost of renting, the cost of energy consumption, and a service provider's margin and profit. Our approach is to treat a multiserver system as an M/M/m queuing model, such that our optimization problem can be formulated and solved analytically. Our main contributions are as follows. Allocate the job based on working rate. It will be reduce the waiting time also increased the profit maximization.***

INTRODUCTION

Cloud computing can efficiently provide on-demand computing resource over the network to consumer worldwide [1]. Typically, computing resources in cloud data centers are dynamically delivered to consumers using a pay-as-you-go pricing model [2]. In addition, the economy of scale brought by cloud computing attracts an increasing number of companies to deploy their applications in cloud data centers. As a typical part of cloud, Infrastructure as a Service (IaaS) provides the foundation for applications [3]. Typical IaaS providers such as Rackspace and Amazon EC2 [4] provide services to consumers based on a pay-per-use model. An IaaS provider manages its own limited resources. Therefore, similar to the definition in [5], from the perspective of an IaaS provider, private cloud in this paper denotes a resource-constrained IaaS provider that may outsource some of its tasks to execute in external public clouds when it cannot deliver promised quality-of-service (QoS) with its resources. A private cloud provider aims to provide services to consumers' tasks in the most cost-effective way while guaranteeing the specified QoS.

Therefore, profit maximization is a critically important goal for a private cloud provider [6].

The uncertainty and aperiodicity of arrival tasks makes it difficult to predict the future arrival tasks, and brings a major challenge to operators of a private cloud. Therefore, it is possible that a private cloud provider cannot satisfy all arrival tasks with its limited resources if the arrival tasks are massive. The existing works usually provide an admission control mechanism to refuse some of arrival tasks that exceed the capacity of a private cloud [7]. Nevertheless, this will decrease the throughput of a private cloud, and inevitably cause revenue loss to the private cloud provider. However, the mechanism of hybrid clouds enables a private cloud provider to make use of public clouds where resources are delivered in the form of virtual machines (VMs) when resources of a private cloud is fully occupied [8]. Delay-tolerant tasks usually have a strict service delay bound to meet. Moreover, cloud providers such as Amazon EC2 pro- vide resources in the form of VMs to paying consumers. In the real life, the execution prices of each VM type offered by public cloud providers vary with time [9]. Besides, the power prices in a private cloud also express temporal diversity [7]. There- fore, this presents an opportunity to maximize the profit of a private cloud by a temporal task scheduling while guaranteeing the strict service delay bound of all tasks. The temporal task scheduling can intelligently dispatch arrival tasks to execute in private and public clouds within the service delay bound.

There are several differences between the proposed temporal task scheduling in hybrid clouds and conventional areas including manufacturing, transportation, etc. Therefore, we clearly summarize the differences as follows.

1) The virtualization of cloud enables resources to be encapsulated as VMs that can be delivered on demand. Besides, services in a cloud can be dynamically configured and de- livered. Therefore, a private cloud can keep scalable and schedule tasks using VMs provided by other public clouds even if resources in a private cloud are fully occupied. However, the available resources in conventional scheduling are usually limited and fixed.
2) The economy of scale enabled by the cloud makes a private cloud provider focus on maximizing its profit by providing a pay-as-you-go pricing model. However, scheduling in conventional areas usually cannot support a pay-as-you-go model because it is difficult to deliver essential resources (e.g., manufacturing tools) on demand.
3) The prices in hybrid clouds vary during the service delay bound. The temporal diversity in prices presents an opportunity to propose a temporal task scheduling that maximizes the profit of a private cloud provider in hybrid clouds. However, compared to applications in a cloud, the scheduling period in conventional areas is much longer. Therefore, the service delay bound of arrival tasks for cloud applications cannot be guaranteed with conventional scheduling.

In this paper, we study the profit maximization problem for a private cloud provider in hybrid clouds. The execution prices of VMs in public clouds, and the power prices in the private cloud exhibit temporal diversity. We formulate the profit maximization problem for a private cloud provider and propose PMA to solve it. Then, we adopt public real-life workload to evaluate the proposed method. Extensive simulations have shown that the proposed method outperforms existing task scheduling methods in terms of throughput and profit.

The main contribution of this paper is described as follows. First, this work focuses on delay-tolerant tasks and strictly guarantees the service delay bound of all tasks. Second, we propose an architecture of temporal task scheduling in hybrid clouds where a private cloud can outsource some of its arrival tasks that

exceed the capacity of a private cloud to public clouds pro- vided that the service delay bound of all task is strictly guaranteed. Third, this work proposes a PMA algorithm to maximize the profit of a private cloud provider by intelligently scheduling arrival tasks to execute in private and public clouds.

EXISTING SYSTEM

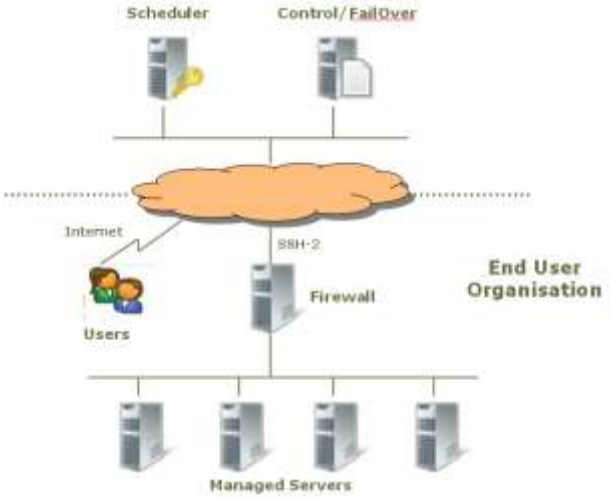
To increase the revenue of business, a service provider can construct and configure a multiserver system with many servers of high speed. Since the actual service time (i.e., the task response time) contains task waiting time and task execution time, more servers reduce the waiting time and faster servers reduce both waiting time and execution time. More servers increase the cost of facility renting from the infrastructure vendors and the cost of base power consumption.

- Previous works usually provide an admission control to intelligently refuse some of arrival tasks.
- Existing methods usually provide an admission control to refuse some of arrival tasks that exceed the capacity of a private cloud.
- Dispatching arrival tasks to execute in private and public clouds within the service delay bound.

OUR WORK

Profit maximization of cloud computing and multiserver configuration of optimal problem in environment will be studied. Our approach is to treat a multiserver system as an M/M/m queuing model, such that our optimization problem can be formulated and solved analytically. We consider two servers based on working rate. Our main contributions are as follows. Allocate the job based on working rate. It will be reduce the waiting time also increased the profit maximization.

- First Come First Serve.
- The waiting time will be reduced.
- Allocate job for particular cloud priority based server.



MICROSOFT VISUAL STUDIO 2008

Microsoft has a wide variety of products; web designed in ASP.Net and code Written in VB.net for front end designed. And as reports in Crystal Reports is built in Micro Soft Visual Studio 2008. Vb.Net is very flexible and easy to understand any application developer.

MYSQL

Microsoft SQL is Structured Query Language (SQL) based on client/server relational database. Each of these terms describes a fundamental part of the architecture of SQL Server.

OVERVIEW OF ASP.NET

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

Enhanced Performance: ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.

World-Class Tool Support: The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

Power and Flexibility: Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers.

Simplicity: ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

Manageability: ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

Scalability and Availability: ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

Customizability and Extensibility: ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.

Security: With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.

Language Support: The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic.Net, and JScript.

What is ASP.NET Web Forms?

The ASP.NET Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages. It provides the ability to create and use reusable UI controls that can encapsulate common functionality and thus reduce the amount of code that a page developer has to write.

- The ability for developers to cleanly structure their page logic in an orderly fashion (not "spaghetti code").
- The ability for development tools to provide strong WYSIWYG design support for pages (existing ASP code is opaque to tools).

ADVANTAGES OF ASP.NET

WEB FORM AND SERVER CONTROL

- ASP.NET Web Forms provide an easy and powerful way to build dynamic Web UI.

- ASP.NET Web Forms pages can target any browser client (there are no script library or cookie requirements).
- ASP.NET Web Forms pages provide syntax compatibility with existing ASP pages.
- ASP.NET server controls provide an easy way to encapsulate common functionality.
- ASP.NET ships with 45 built-in server controls. Developers can also use controls built by third parties.
- ASP.NET server controls can automatically project both up level and down-level HTML.
- ASP.NET templates provide an easy way to customize the look and feel of list server control.

MODULES

- Server Formation
- Analysis of Performance
- Task Scheduling
- Predicting Result

SERVER FORMATION

In our project, the server calculates which cloud doing which job. That is monitoring cloud access, cost calculation and equal sharing of jobs in cloud.

ANALYSIS AND PERFORMANCE

We analyze and compare the performance offered by different configurations of the computing cluster, focused in the execution of loosely coupled applications.

In particular, we have choose different cluster configurations with different number of worker nodes from the three cloud providers and different number of Jobs, we use the following acronyms infrastructure; Amazon EC2 Europe cloud Azure EC2 US cloud.

TASK SCHEDULING

Each and every user assigns the task to cloud, so that task will assign to the cloud in priority scheduling basis or if anyone cloud is free mean, user job assign to that cloud

PREDICTING RESULT

If we assign the job in priority scheduling way to a cloud, we got an output correctly and shortly. The amount or cost will reduced and transferred to cloud owner of the using of cloud.

CONCLUSION

Based on the changing demand, we use to present the design, implementation, and evaluation of a resource management system for cloud computing services. We present a system that uses virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. Both overload avoidance and green computing for systems with multi resource constraints algorithm are achieved. Cloud-Analyst to have cost effective results and development are proposed by new strategy. From the work done, we can conclude that the simulation process can be improved by modifying or adding new strategies for traffic routing, load balancing etc. to make researchers and developers able to do. Overload in the system prevented to develop a set of heuristics. Trace driven simulation and experiment results demonstrate that our algorithm achieves good performance. It contributes to a significant portion of the operational expenses in large data centers saves electricity.

FUTURE WORK

For the future work, scenario reduction techniques will be applied to reduce the number of scenarios. In addition, the optimal pricing scheme for cloud providers with the consideration of competition in the market will be investigated. Scenario reduction techniques will be applied to reduce the number of scenarios. In addition, the optimal pricing scheme for cloud providers with the consideration of competition in the market will be investigated. We need to predict the future resource needs of VMs. Application level statistics in VM is a solution to be looked inside. e.g., by parsing logs of pending requests. Doing so requires modification of the VM which may not always be possible. Instead of, we make our prediction based on the past external behaviors of VMs.

REFERENCE:

[1] S. Lyden and M. E. Haque:
"A simulated annealing global maximum power point tracking approach for PV module sunder partial shading conditions,"
IEEE Trans. Power Electron., vol. 31, no. 6, pp.4171–4181, Jun. 2016.

[2] R. Buyya, C.S.Yeo, S.Venugopal, J. Broberg,andI.Brandic:
"Revenue maximization with optimal capacity control in infrastructure as a service cloud markets,"
IEEE Trans. Cloud Comput, vol. 3, no. 3, pp. 261–274, Jul. 2015.

[3] D. Bruneo:
"A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems,"
IEEE Trans. Parallel Distribute.Syst. vol. 25, no. 3, pp. 560–569, Mar. 2014.

[4] K. Hwang, X. Bai, Y. Shi, M. Li, W. G.Chen, and Y. Wu:
"Isthe same instance type created equal? Exploiting heterogeneity of public clouds,"
IEEE Trans. Parallel Distribute. Syst., vol. 27, no. 1, pp.130–143, Jan. 2016.

[5] X. Zuo, G. Zhang, and W. Tan:
"Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IaaS cloud," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 564–573, Apr. 2014.

[6] J. Mei, K. Li, A. Ouyang, and K. Li:
"A profit maximization scheme with guaranteed quality of service in cloud computing,"
IEEE Trans.Compute., vol. 64, no. 11, pp. 3064–3078, Nov. 2015.

[7] J. Luo, L. Rao, and X. Liu:
"Temporal load balancing with service delay Guarantees for data center energy cost optimization,"
IEEE Trans. Parallel Distribute. Syst., vol. 25, no. 3, pp. 775–784, Mar. 2014.

[8] M.Menzel,R.Ranjan,L.Wang, S.U.Khan,and J. Chen:
"CloudGenius:A hybrid decision support method for automating the migration of web application clusters to public clouds,"
IEEE Trans. Comput.,vol. 64, no. 5, pp. 1336–1348, May 2015.

[9] H. Zhao, M. Pan, X. Liu, X. Li, and Y. Fang:
"Exploring fine-grained resource rental planning in cloud computing,"
IEEE Trans. Cloud Comput., vol. 3, no. 3, pp. 304–317, Jul. 2015.

MODULES Server Formation