TELKOMNIKA, Vol. 11, No. 6, June 2013, pp. 2956 ~ 2961
e-ISSN: 2087-278X

An Intelligent Job Scheduling System for Web Service in Cloud Computing

Jing Liu*, Xingguo Luo, Bainan Li, Xingming Zhang, Fan Zhang
National Digital Switching System Engineering & Technology Research Center, Zhengzhou 450002, China
*Corresponding author, e-mail: tobelj@126.com

Abstract

Cloud computing is a new computing and business paradigm with flexible and powerful computational architecture to offer universal services to users via Internet. Web service is one of the most active and widely adopted implementation in cloud computing. The performance of the scheduling system influences the benefits of both sides in this computing paradigm. In this paper, we present an intelligent scheduling system for web service, which considers both the requirements of different service requests and the circumstances of the computing infrastructure which consists of various resources. We have validated our hardware based server WSVP by conducting a performance comparison to the traditional Apache and Nginx servers. The results demonstrate that our WSVP equipped with intelligent scheduling system has immense potential as it offers better performance while consumes much lower energy.

Keywords*: intelligent job scheduling, cloud computing, web service, resource allocation, cognitive decision*

1. Introduction

Cloud computing, the long-held dream of "computing as a utility", is a large-scale distributed computing paradigm driven by economies of scale, in which a pool of heterogeneous, virtualized, dynamically-scalable, highly available, and configurable and reconfigurable computing resources (e.g., networks, computing units, storage, applications, data) can be rapidly provisioned and released with minimal management effort in the data centers [1-8]. However, the overwhelming computing requests from kinds of users and from different geographical regions, calls for a powerful and cost-effective scheduling paradigm, which will help in effective utilization of resources and will also satisfy the computing needs of the users [9].

The scheduler takes a virtual network computing request constrained by the Service Level Agreement (SLA) and attempts to map it onto the available computing resources. If a mapping is found, the computing resources in that mapping are allocated to the user. Management of heterogeneous virtualized resources is an important and challenging task, especially when dealing with fluctuating workloads and performance interference.

Several research initiatives have investigated the issue of job scheduling for cloud computing and have proposed various architectures. An auto-regressive-moving-average (ARMA) model is applied to represent the allocation to application performance relationship [10], which introduces a MIMO controller to automatically allocate CPU share and I/O bandwidth to multiple virtual machines (VMs), however, the model does not emphasize on the release of unused resources and may not be effective under steady workload; the reinforce learning based systems are applied in optimal server allocation and VM resource management [11, 12], however, the complexity of training and maintaining the models under different scenarios becomes expensive when the number of VMs increases; should be scalable and highly adaptive; dynamic scheduling systems are put forward for cloud computing services in the approach of QoS performance analysis [13, 14], however, these researches rarely mention the differential service-oriented QoS guarantee in the scheduling system.

In this paper, we first present a macroscopic architecture for the cognition and decision model for web service in the cloud computing, in which the intelligent scheduling system (ISS) is included to recognize and classify the service requests, and provide the information for the

Received January 5, 2013; Revised April 2, 2013; Accepted April 9, 2013

learning and reasoning machine to enrich the knowledge base; then we introduce the main components of the intelligent scheduling system in detail, mainly focusing on the service process and special requirements for the cloud computing, at last, we take some experiments to validate our design and compare our web server to traditional software based web servers, the results shows the effectiveness of our system.

2. Architecture Overview

Figure 1 shows theholistic architecture of cognition and decision model for the cloud computing. Fundamentally, the cognition and decision system in the model can perceive both characteristics of the service request and current states of the computing system(such as in-network storage space, computation intelligence, bandwidth capacity and so on), provide the learning machine and reasoning machine with the perception information after identification and classification operation.

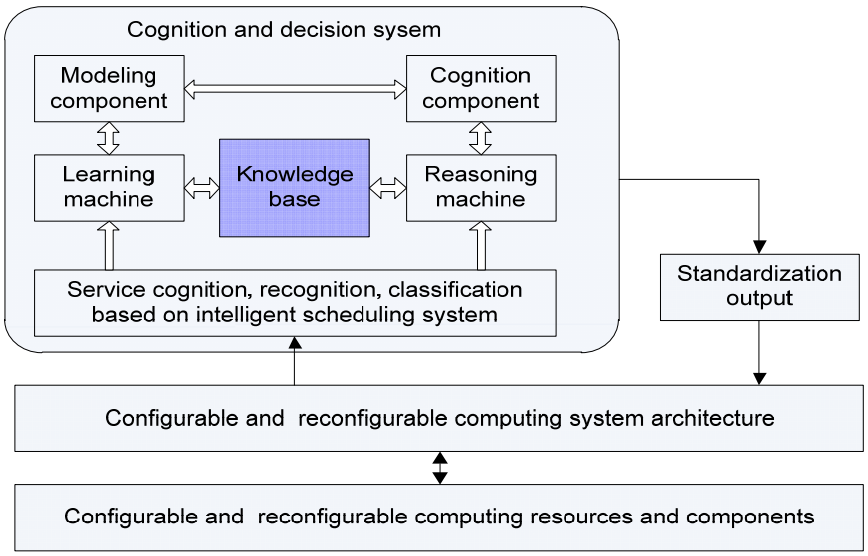

Figure 1. Architecture of Cognition and Decision Model for the Cloud Computing

Based on the perceptive, learned and interactive information, modeling component can improve the performance of established cognition model; with the help of human-computer interaction, the learning machine adds the learned cases and rules to the knowledge base, which can be used as references for the similar request in the future; combining cases and rules in knowledge base with perceived information, under the guidance of cognition model, the reasoning machine can provide the standardization output to express the decision of the system under current situation; according to the output, the computing system is configured or reconfigured to form an optimal computing architecture under the constrained of current resource status.

3. Intelligent Job Scheduling System in Cloud Computing

In cloud computing, the high volatility in customer demand makes it ideal to provide cloud resource dynamically. On the service providers side, a key issue is how to partition and configure their own computing resources for establishing a computing paradigm in order to provide the customers with QoS satisfied services and maximize revenues with cost-efficient resource utilizations for the providers. Fig. 2 shows the functional architecture for intelligent job scheduling system for cloud computing, in the following, we will introduce the main components in detail.

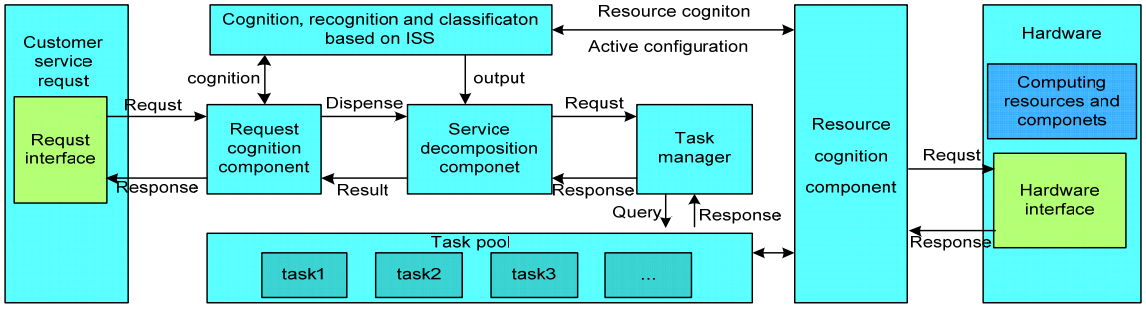

Figure 2. Functional Architecture for Intelligent Job Scheduling System

3.1. Request Cognition Component

According to heterogeneous characteristics of the service requests submitted by different costumers, such as data encryption requirements in encrypted communication, computational accuracy requirements in scientific calculation, the high-definition requirements in video on demand (VOD), the request cognition component should be fully aware of the special needs for different businesses and provide wanted services.

The cognitive content includes type, function, requirements for computing, storage and communication, arrival law and concurrent conditions, requirements for security and privacy and QoS of the business. The feasible solutions includes business requirements customization method which declares the resource needs when subitting the business, request script analysis method and personalized perception method.

3.2. Service Decomposition Component

The cloud computing is a hybrid computer structure which consist of general purpose processors, special purpose processors, reconfigurable processors, each kind of processors may perform well in handling some special tasks, while get a poor result for other businesses, thus, we should decompose the service request into different level of granularities with different processor preferences. In the next procedure, the task manager will analyze the resource requirements of each granularity, and mapping it on to optimal processors to reach a effective solution.

Figure 3 shows an example of service decomposition for the web service of static webpage browse, in the figure, the service is represented in a directed acyclic graph located above, and H1~H7 indicate the basic granularities after decomposition.

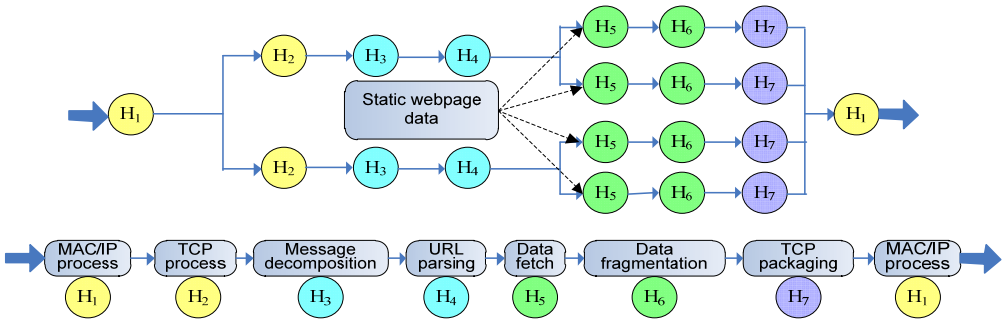

Figure 3. Service Decomposition for the Static Webpage Browse

3.3. Task Manager

The main function of this manager includes:

1) Request management. Task status maintenance(start, stop, cancel, run to complete, wait ....), job submission and results retrieve, detail information visualization;

2) Job scheduling. According to the scheduling strategies and known information, determine the scheduling sequence and the resource assignment for the requests. The scheduler may subscribe different status information to the system considering the needs, such as job status variation, resource release status;
3) Job mapping management. Based on the scheduling strategies, submitted resource requirements of the user and current distribution of the resources, determine the resource types and amount for each job and the allocation scheme to make the jobs run on suitable resources.

3.4. Resource Cognition Component

The main function of this component includes:

1) Resource management. Manage the available resources(discovery, allocation, provision, release), where, the resources can be physical resources or virtual resources;
2) Performance of the resource monitoring. Monitor the types and capacity of hardware, network load, I/O load, communication bandwidth of the interconnection network, utilization of memory and computing component (CPU, FPGA, GPU, etc.), temperature and voltage of motherboard and each computing component, fan speed, etc. and add these information to the state base;
3) Dynamic optimization of scheduling strategy. According to the real time distribution of the task pool and resource pool, dynamically optimize the scheduling strategies by adjusting deployment of the computing resources;
4) Error notification. Notify the system administrator in time when there is a failure, then the system can make an adjustment to the scheduling scheme, avoid allocating tasks to the failed resources.

4. Experiments and Discussion

Web service is one of the most active and widely adopted implementation in cloud computing, in which, all communication between the server to client, client to client or server to server and in general application to application, uses the same protocol. Due to the above reasons, we choose web server as a typical example of the intelligent job scheduling system in cloud computing environment.

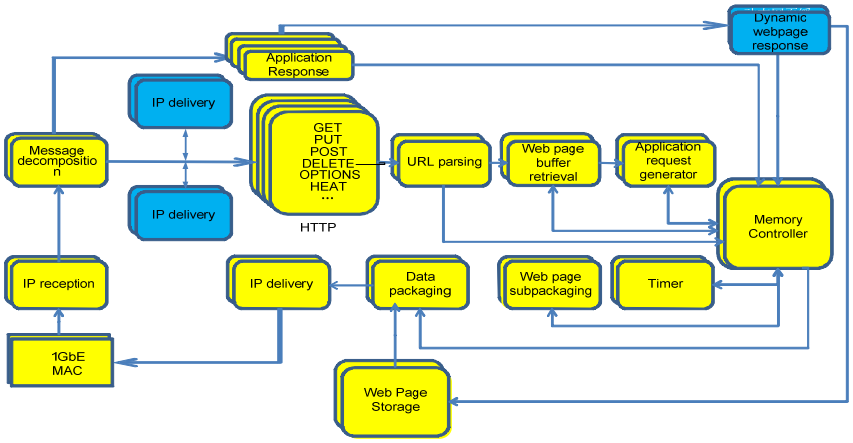

Figure 4. Framework of Web Server Verification Platform

Based on Field Programmable Gate Arrays (FPGAs), we design a software and hardware collaborative web server verification platform named WSVP3.0, the framework of which is shown in Figure 4. Each platform consists of 4 Virtex-5 LX110T FPGAs as the central processor, up to 64GB DDR2 ECC DRAM as the memory and 8 Ethernet interfaces for communication between modules with the rate of 1Gbps.

The experiments aim to demonstrate and evaluate the contribution of the intelligent job scheduling system for web service. It also aims to compare the obtained results of the configurable hardware based HTTP web server to the traditional Apache and Nginx software based servers.

4.1. Experimental Settings

The HTTP requests are generated by Spirent stress tool Avalanche 2900, and the data analysis runs on console application TestCenter 3.51. The configuration parameters of the three servers are shown in Table 1, where, the DDR2 type is DDR2 800 and the hard disk interface type is SATA II.

Table 1. Configuration Parameters of the Servers

Server name	Processor	Memory (GB)	Hard disk (GB)	Network interface(bps)	Idle Power(W)
Apache 2.2	2 Xeon 5520 (4 core 2.27 GHz)	24	640	1G	320
Nginx 0.7.6.1	2 Xeon 5520 (4 core 2.27 GHz)	24	640	1G	320
WSVP3.0	4 Xilinx V5 LX110T	24	640	1G	28

The test set is consist of two kind of web pages, one is static web pages, with the average size of 8.3kb; the other is dynamic web pages which may include audios, videos and forms, with the average size of 710.68kb.

4.2. Results and Discussion

Tables 2 to 5 show the experimental results of the three web server platforms for web service requests with different web page size.

Table 2. Results of Requests Size of 4k

Server name	WSVP	Apache	Nginx
Throughput(Mbps)	682	600	770
Total requests	19193358	20380929	27843182
Failed requests	1	2097527	654371
Success pages/s	13459	12821	19066

Table 3. Results of Requests Size of 10k

Server name	WSVP	Apache	Nginx
Throughput(Mbps)	831	801	870
Total requests	11521673	15153751	13883487
Failed requests	1603	2283689	413759
Success pages/s	10078	9025	9445

Table 4. Results of Requests Size of 100k

Server name	WSVP	Apache	Nginx
Throughput(Mbps)	986	850	928
Total requests	1645460	4065041	1555448
Failed requests	1	2648673	87157
Success pages/s	1153	993	1029

Table 5. Results of Requests Size of 1M

Server name	WSVP	Apache	Nginx
Throughput(Mbps)	987	860	950
Total requests	233537	269833	196779
Failed requests	69492	2554866	90744
Success pages/s	121	100	74

It can be seen from above tables, the lager the web page size is, the higher the failed page ratio is, and the less web pages return to the clients per second.

Considering the throughput, the Apache server performs worst all the time, and the WSVP could provide higher throughput when the web page size increases. Compared to the other two software based web servers Apache and Nginx, we can see, that the WSVP server shows an excellent performance in failed request ratio; the same phenomenon can be seen when considering the success pages returned to clients per second, this is because the WSVP is a hardware based server, providing an pipeline structure which is good at processing the high concurrent transactions such as web services.

Considering the power consumption of the three web servers when they provide services, in the experiment, the WSVP consumes much lower energy than the other two, the power of WSVP is about 70W, while Apache and Nginx are about 325W. We can also see that the idle power and the working of WSVP differs a lot, while the software based web servers may not vary much. What is more, the power consumption of software based servers are much more than WSVP, no matter in idle or working state.

From a further performance comparison of the WSVP and traditional Apache, we can get that the power consumption ratio, the connection establishment speed ration, first package arrival rate ratio and the success pages returned to clients ratio of WSVP to Apache is 12.5, 17970-174, 0.37-0.39 and 1.05-1.21 respectively. We can conclude that, our WSVP server achieves better performance, while consumes much lower energy, which is very important to the IT industry in the pursuit of green computing.

5. Conclusion

In this article, we have presented a framework of intelligent job scheduling systems based on the cognition and decision technology for web service in cloud computing environment. In this system, after the analysis of the requirements of different service requests, combining the cognitive result of the available resources, a scheduling scheme is generated aiming to benefit both the costumers and service providers. To validate the effctiveness, we conduct experiments on the hardware based web server WSVP designed based on the proposed idea. Experimental results show that our WSVP server achieves better performance, while consumes much lower energy.

References

[1] Armbrust M, Fox A, Griffith R, Joseph A D, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I. A view of cloud computing. *Communications of the ACM*. 2010; 53(4): 50-58.
[2] Buyya R, Chee SY, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*. 2009; 25(6): 599-616.
[3] Iosup A, Ostermann S, Yigitbasi MN, Prodan R, Fahringer T, Epema DHJ. Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. *IEEE Transactions on Parallel and Distributed Systems*. 2011; 22(6): 931-945.
[4] Baliga J, Ayre RWA, Hinton K, Tucker RS. *Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport*. Proceedings of the IEEE. 2011; 99(1): 149-167.
[5] Almutairi A, Sarfraz M, Basalamah S, Aref W, Ghafoor A. A Distributed Access Control Architecture for Cloud Computing. *IEEE Software*. 2012; 29(2): 36-44.
[6] Dikaiakos MD, Katsaros D, Mehra P, Pallis G, Vakali A. Cloud Computing: Distributed Internet Computing for IT and Scientific Research. *IEEE Internet Computing*. 2009; 13(5): 10-13.
[7] Ashish Kumar. World of Cloud Computing & Security. *International Journal of Cloud Computing and Services Science*. 2012; 1(2): 53-58.
[8] Sean Carlin, Kevin Curran. Cloud Computing Technologies. *International Journal of Cloud Computing and Services Science*. 2012; 1(2): 59-65.
[9] L Shyamala, Saswati Mukherjee. *EduCloud: An Institutional Cloud with Optimal Scheduling Policies*. 4th International Conference on Computing, Communication & Information Technologies. Vellore. 2011: Part I, 114-123.
[10] P Padala, KY Hou, KG Shin, X Zhu, M Uysal, Z Wang, S Singhal A, Merchant. *Automated control of multiple virtualized resources*. Proceedings of the 4th ACM European conference on Computer systems. New York. 2009: 13-26.
[11] G Tesauro, NK Jong, R Das, MN Bennani. On the use of hybrid reinforcement learning for autonomic resource allocation. *Cluster Computing*. 2007; 10(3): 287-299.
[12] J Rao, X Bu, CZ Xu, L Wang, G Yin. *VCONF: a reinforcement learning approach to virtual machines auto-configuration*. Proceedings of the 6th international conference on Autonomic computing. New York. 2009: 137-146.
[13] Afzal A, AS McGough, J Darlington. Capacity planning and scheduling in Grid computing environments. *Future Generation Computer Systems*. 2008; 24(5): 404-414.
[14] Yuan-Shun D, X Min, P Kim-Leng. Availability Modeling and Cost Optimization for the Grid Resource Management System. *IEEE Transactions on Systems, Man and Cybernetics, Part A.* 2008; 38(1): 170-179.