A NOVEL JOB SCHEDULING MODEL TO ENHANCE EFFICIENCY AND OVERALL USER FAIRNESS OF CLOUD COMPUTING ENVIRONMENT

Wenjuan Li[1, 2], Xuezeng Pan[1]
[1]*College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang, China*
[2]*Hangzhou Normal University, Hangzhou, Zhejiang, China*

Qifei Zhang, Lingdi Ping
College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang, China

Keywords: Cloud computing, Job scheduling model, User fairness, System throughput.

Abstract: Job scheduling and resource allocation are the key factors which affect the performance of parallel and distributed systems. This paper first analyzed several common used job scheduling models in large and distributed environment. Since the main purpose of cloud computing is to provide “low-cost” and “on-demand” services, this paper introduced a novel two-level based cloud scheduling model. In our model, job scheduling in the cloud is divided into User Scheduling and Task Scheduling which were designed to implement different scheduling strategies. In the User Scheduling level, it applies the distribution justice theory in the sociology area to achieve the highest average user fairness. While in Task Scheduling level, it applies adaptive strategies to achieve the binding of micro tasks and the virtual machines according to different objectives of tasks. The new model redefined the messaging mechanism between main cloud entities and a new advertising mode was introduced for cloud providers to promote their services. The results of emulation experiments show that compared to traditional algorithms the proposed model can guarantee the user expectations and the relative fairness of the services on the basis of high overall efficiency of cloud systems.

1 INTRODUCTION[1]

Cloud computing is an emerging business computing model for sharing information infrastructure. It successfully packages all kinds of resources belonged to different cloud providers in different abstract levels as virtualized services over the network and provides end-users low-cost and on-demand services. (Foster I. and Kesselman C., 1999). Obviously, cloud will become one of the most important service types in the next generation network market. Currently, however, cloud computing is faced with many issues needed to be solved urgently, such as security and trust, standardization, uniform architecture, programmable parallel computing and virtualization, etc. (Mladen A. Vouk., 2008).

Resource allocation and task scheduling are the key factors affecting the efficiency of paralleling computing. (Guoliang Chen, Hong An and Lin Chen, 2004). They are also the key of cloud researches. Although scholars have designed many excellent job scheduling models for traditional distributed and parallel environment, there still lacks a perfect and widely-accepted cloud scheduling model. (Fuxi Zhu and Yanxiang He, 2003). What’s more, in cloud all kinds of resources are abstracted into the form of virtual machines. (Foster I., Yong Zhao, Ioan Raicu and Shiyong Lu, 2008). This specific point determines that cloud resource allocation and scheduling mechanisms are quite different from traditional distributed systems.

This paper proposed a novel two-level based cloud scheduling model to guarantee the cloud users’ satisfactory and the total throughput capacity

[1] This project is supported by Chinese Zhejiang Provincial Research Project (2010C31003).

Li W., Pan X., Zhang Q. and Ping L..
A NOVEL JOB SCHEDULING MODEL TO ENHANCE EFFICIENCY AND OVERALL USER FAIRNESS OF CLOUD COMPUTING ENVIRONMENT.
DOI: 10.5220/0003381101520157
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 152-157
ISBN: 978-989-8425-52-2

of cloud systems. It divided cloud scheduling into two levels: User Scheduling and Task Scheduling and designed different strategies for them. User Scheduling can be seen as resource allocation which is the overall scheduling as the unit of cloud user. And the target is to achieve the maximum agreement of user expectation to the quality of service. Task Scheduling is the scheduling within one user's tasks. It realizes the binding of tasks to virtual machines. And its goal changes with the real requirement of tasks. Different scheduling strategies are designed for different scheduling levels. Through message mechanism, cloud providers can advertise their services to customers.

This paper was constructed as follows: part 2 describes the related concepts of scheduling and scheduling model and Part 3 analyzes and compares several existing task scheduling models. Part 4 introduces the new two-level based scheduling model. Part 5 is the design, results and analysis of the simulation experiments and the last part is conclusion and future work.

2 RELATED CONCEPTS

2.1 Scheduling in Cloud

The goal of cloud scheduling is to select an appropriate distribution strategy dispatching paralleling tasks to resource nodes under the premise of priority constraints and certain performance indicators and to achieve the minimal total execution time.

Since all kinds of cloud resources at different abstract layers are virtualized, scheduling will undoubtedly involve two aspects: one is the binding of cloud providers' actual resources to the virtual resources and the other is the binding of users' tasks to virtual resources.

2.2 Distribution Justice

Distributive justice concerns the fair, just or equitable distribution of benefits and burdens. It is widely regarded as an important concept and influential force in philosophy and the social sciences. The socialists have found three distinct principles of justice: the Need Principle, the Efficiency Principle and the Accountability Principle. And also justice is context-based. (Guillermina Jasso, 1989).

In cloud computing environment whose highest objective is to provide "low-cost" and "on-demand" services, how to realize the true sense of distribution justice of users and how to achieve the goal of users' willing burden in line with their gained services is a most important issue that cloud scheduling model has to concern about.

2.3 Berger's Theory

Berger's theory comes from the sociology field and it is about the fair distribution of the society. The theoretical basis of Berger's model is "Expectation States Theory" which makes the model more reasonable and accurate. Berger's theory believes that "balance" is gained through the comparison of reference structure and local structure. (Berger, J., B. P. Cohen and M. Zelditch., 1966). It is easy to extend Berger's theory into computer field. We believe that when majority network users receive considerable quality of service with their expectation, network balance or distribution justice establishes.

3 SCHEDULING MODELS IN TRADITIONAL DISTRIBUTED SYSTEMS [22-24]

3.1 Min-Min Algorithm

Min-Min (Minimum-Minimum Completion Time) algorithm and Max-Min (Maximum-Minimum Completion Time) algorithm are both classical task scheduling algorithms and the research basis of many of today's parallel scheduling. The main idea of Min-Min algorithm is constantly searching for and scheduling the task which is smallest in the Minimum Completion Time when deployed in all cloud resources. Min-Min algorithm has the advantage of fast execution. But since short tasks are always prior scheduled, it may easily lead to uneven host load and low utilization rate. Max-Min has the opposite characteristics. It solves the problem of uneven load while with the slower execution speed.

3.2 Ant Colony Optimization Algorithm

ACO (Ant Colony Optimization) algorithm is one of the most common used heuristic scheduling algorithms in distributed systems. (Dorigo M, Maniezzo V. and Colorni A., 1996). Ant colony algorithm draws on the principles of ants' finding food in nature, uses the history pheromone of each execution experience to achieve positive feedback effect and through iteration keeps closer to the optimal solution. But ACO algorithm is a kind of bomimetic algorithm of which the implementation process has somewhat randomness. In addition, due to the lack of pheromone in the early execution stage, it takes longer time to find the solution of the overall problem. What's more, just like other evolutionary algorithms, ACO algorithm may be premature and easy to fall into local optimum.

3.3 Berger's Model based Algorithm

Chunyan Zhao in her Master's thesis proposed a novel cloud scheduling algorithm based on Berger's theory. (Chunyan Zhao, 2009). Berger's Model based task scheduling algorithm treats the justice distribution of user tasks as its' primary goal. Through the binding of specific user's tasks to specific resources, it attempts to achieve the goal of meeting the greatest consistency of user expectations and the actual allocation of resources.

3.4 Brief Analysis

Berger's model based algorithm only takes into account the distribution relationship between user's tasks and virtual machines without involving the resource side. Although all types of resources in cloud are provided in the form of virtual machines which appears that job scheduling process is simplified. But a complete scheduling model needs consider not only the binding of virtual machines to users' tasks but also real resources to virtual machines. ACO and Min-Min algorithm runs fast which can guarantee high throughput of cloud system. However, they ignore user's expectations and their actual payment capability which may lead to the consequences that some high expectations, strong payment capability users obtain low-quality services, while some users access to expensive and high quality services beyond their capabilities and expectations which make them to bear additional cost.

4 PROPOSED SCHEDULING MODEL

We proposed a novel scheduling model for cross-clouds environment. Following is the detail of our model.

4.1 Realization Mechanism

Figure 1 shows the basic realization framework of the new model.

The model divides cloud scheduling into two stages: User Scheduling and Task Scheduling. User Scheduling refers to the scheduling basing on the unit of cloud customers. It realizes the binding of specific customers to specific providers and their real resources according to the overall QoS(quality of services) requirements and the current situation of cloud market. Task Scheduling achieves the binding of tasks within one cloud user and the virtual resources that are allocated to the user. Obviously, User Scheduling stands a higher level than Task Scheduling. The model sets ***User Center (UC)*** and ***Cloud Information Service (CIS) Center***. Cloud users register themselves when first login. And then they login cloud system by the registered account and submit all of their requirements including resource requirements, time and cost constraints to UC each time. Cloud providers register themselves to CIS center through which they can obtain customers' information. Using the user list, providers can publish resource advertisements to users.

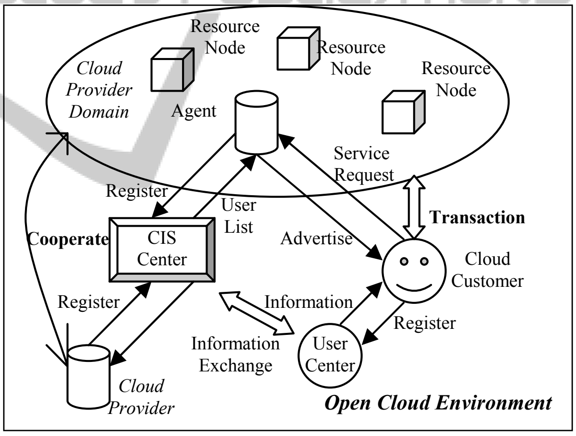

Figure 1: The Basic Realization Framework of the New Model.

In the implementation process, a cloud user can freely choose a provider or a certain set of providers. Also it can use the more reasonable allocation solution recommended by User Scheduler which is obtained according to user expectations and the current situations of the cloud resources. When the user is bound with the providers, the providers will assign a number of virtual resources to complete the overall work of the user. Next task scheduler will be on work. In our model, task scheduler is configurable and its scheduling goal and algorithm changes with different user requirements. It cares about which is sensitive to the user tasks completed time, cost or the other constraints. It will choose a most suitable allocation strategy binding tasks to virtual machines and tasks' execution order to achieve the specific user target.

4.2 User Scheduling Algorithm

In the stage of User Scheduling, the new model references the Berger's model to deploy its scheduling strategy to ensure the overall user fairness. The general process of User Scheduling is as follows: first of all, to calculate the comprehensive expectations of the tasks belonged to one cloud user. And then it clusters cloud resources of different cloud providers according to the user's requirement and calculates different resource set's comprehensive capability. At last, it choose the most suitable resource set to the user according to the Euclidean Distance from user's comprehensive expectation to resource set's comprehensive capability.

Algorithm to bind a Certain User to a Certain Set of Resources belonged to One Provider.

```
AllocateResourcesForUser(CloudProvider cp,
CloudUser user) {
  int requiredCapability=0;
  List<Resource> rList=new
ArrayList<Resource>();
  int resourceNum=cp.getResourceList().size();
    //obtain the number of resources of this
        provider
    Resource r=null;
    rList.clear();
    requiredResource=user.
computeRequireCapability ();
    //compute the user's total resource
       requirements.
    While(requiredResourceCapability>0&&resource
Num>0){
    r=findResource(requiredResourceCapability,cp
.getResourceList())
    //find the resource with the Minimum
       Euclidean Distance to user's requirement.
    rList.add(r);
    //add the chosen resource into the resource
      list.
    requiredResource-=r.capability;
    //required capability minus obtained
       capability from the chosen resource
    resourceNum--;
    }
    Return rList;
    }
```

4.3 Task Scheduling Algorithm

In the new model, Task Scheduling strategy is configurable. The general process of Task Scheduling is as follows: first of all, to judge the type of the task whether it is "time-sensitive" or "cost-sensitive". And then it chooses an appropriate strategy on the basis of the task's category.

Algorithm to bind a Certain Task to a Certain Virtual Machine.

```
allocateVMForTask(List<VirtualMachine>vmList,
List<Cloudlet> cloudletlist){
while(cloudletList.iterator().hasNext()){
Cloudlet c1 = (Cloudlet) iter.next();
int tasktype=c1.getClassType();
  //obtain the type of the task
int virtualmachineID=0;
switch(tasktype){
// choose appropriate virtual machine in vmList
    to the task according to its type.
case 1:
vmID=this.findVMforTimeCaredTask(vmList);
       c1.setVmId(vmID);
      break;
case 2:
vmID=this. findVMforCostCaredTask (vmList);
       c1.setVmId(vmID);
      break;
case 3:
…
}
}
```


5 SIMULATION EXPERIMENTS

5.1 The Design of Simulation Experiments

This paper designed simulation experiments to test the performances of the new scheduling model, Min-Min and Berger's Model based scheduling algorithm. Our simulation experiments only took into account "time-sensitive" tasks, that is, cloud users cared only about their tasks' completion time. The simulation experiments set up three evaluation factors: task sets' ***final completion time***, ***total implementation Cost*** and ***the overall customer satisfaction evaluation***. Implementation cost means the execution cost to complete users' tasks. It is calculated based on pre-specified unit costs of resources and the consumption time. Overall customer satisfaction evaluation means the average fairness of all tasks. Since our experiments only considered "computation-sensitive" tasks, the overall customer satisfaction is evaluated by tasks' actual CPU occupation time and their expectation CPU time. Formula 1 below shows the calculation method.

$$Jvalue = \log(\frac{actualCPUTime}{\exp ectationCPUTime}) \qquad (1)$$

The simulation experiments simulated cross-clouds system that contains one CIS center, one UC(user center), two cloud providers and a certain number of cloud users. Each provider contains random number of resource nodes. Each customer should complete a certain number of cloud tasks on resource nodes. According to the specified task number, a function generates cloud task and randomly deploy each task's parameters. In the initial time, providers register themselves to CIS center and customers register themselves to UC. CIS center and UC can often exchange data. When cloud customers want to finish their tasks, they will first send a message to UC and UC will help user to find an appropriate provider set. When transaction relationship is successfully established, user scheduler will help to choose the appropriate resource list for the customer. And task scheduler will work to bind the specific task to the specific virtual machine.

5.2 Experiments' Results and Analysis

The following three figures (Figure2-Figure4) show the results of the simulation experiments.

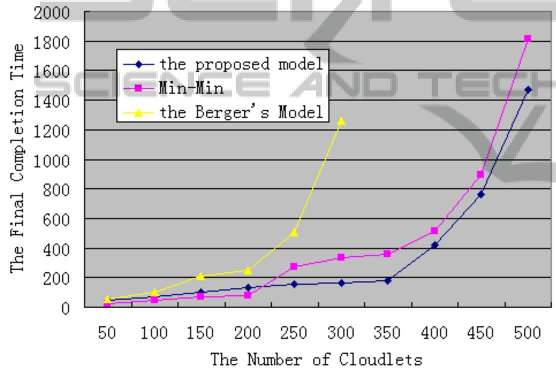

Figure 2: Comparison of task set's final completion time.

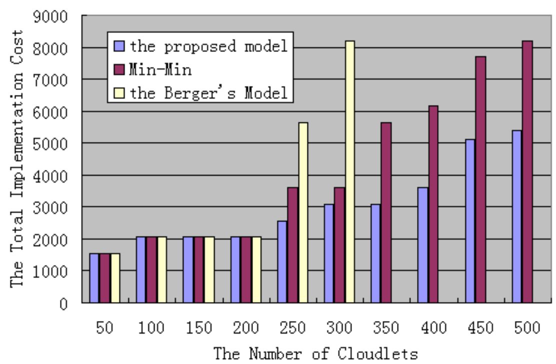

Figure 3: Comparison of task set's total implementation cost.

The results show that when the total number of tasks is not significant, the Berger's Model has some advantages in the customer satisfaction evaluation for the reason that its original purpose is to ensure the overall fairness of tasks. However, the pre-processing cost spent on generating scheduling sequence is very large which leads to the rapid increase on execution time when the number of tasks increases. And the proposed model can ensure high transaction throughput and relatively low implementation cost on the basis of high overall customer fairness compared to traditional scheduling models in cross-domain environment.

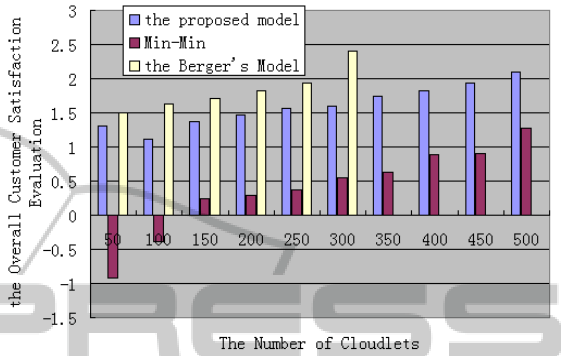

Figure 4: Comparison of task set's overall customer satisfaction evaluation.

6 CONCLUSIONS AND FUTURE WORK

This paper introduces a novel two-level based scheduling model that can be used in cross-clouds environment. We divided cloud scheduling into two different stages: User Scheduling and Task Scheduling. We set up a User Center (UC) to manage cloud customers' information. In the stage of User Scheduling, the model tries to achieve the target of highest user fairness and satisfaction which meets the goal of cloud computing. In the stage of Task Scheduling, its strategies can be justified according to actual task requirement which can increase the total throughput of cloud system. Simulation experiments show the proposed model can establish transaction relationship between cloud customer and provider and gain relatively good performance.

In future, there are still a lot of issues to be studied. For example, how the two scheduling levels interact, how to cluster resources, etc. We will establish a cross-clouds prototype system and implement and perfect the proposed model in the test-bed.

REFERENCES

Foster, I., Kesselman, C., 1999. The Grid: Blueprint for a New Computing Infrastructure. *Morgan Kaufmann*, San Francisco.

Mladen A. Vouk., 2008. Cloud Computing-Issues Research and Implementations, In *ITI 2008, the 30th International Conference on Information Technology Interfaces*. IEEE.

Guoliang Chen, Hong An, Lin Chen, 2004. Practice of Parallel Algorithm. *Higher Education Press*. Beijing. 1st edition.

Fuxi Zhu, Yanxiang He, 2003. Parallel Scheduling Algorithm in Distributed Computing Theory and Design. *Wuhang University Press*. Wuhang. 1st edition.

Foster I., Yong Zhao, Ioan Raicu, Shiyong Lu, 2008. Cloud Computing and Grid Computing 360-Degree Compared. In *GCE '08 (2008), the Grid Computing Environments Workshop*. IEEE.

Guillermina Jasso, 1989. The Theory of the Distributive-Justice Force in Human Affairs: Analyzing the Three Central Questions. Sociological Theories in Progress: New Formulation. *Sage Pub*. Newbury Park, Calif.

Berger, J., B. P. Cohen, M. Zelditch, 1966. Status Characteristics and Expectation States. Sociological Theories in Progress (Volume One). Houghton-Miffin. Boston.

Dorigo M, Maniezzo V. and Colorni A., 1996. Ant system: Optimization by a colony of cooperating agent. Man and Cybemetics-Part B: Cybemetics, 26(1). IEEE Trans on Systems.

Chunyan Zhao, 2009. Research and Realization of Job Scheduling Algorithm in Cloud Environment. Beijing Jiaotong University Master’s Thesis, Beijing.