

# An Greedy-Based Job Scheduling Algorithm in Cloud Computing

Ji Li<sup>a,b</sup>, Longhua Feng<sup>a,b</sup>, Shenglong Fang<sup>c</sup>

<sup>a</sup> College of Computer Science, Chongqing University, Chongqing 400044, China

Email: leedge@cqu.edu.cn, fslfh@163.com

<sup>b</sup> Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400044, China

Email: leedge@cqu.edu.cn, fslfh@163.com

<sup>c</sup> College of Computer Science and Information Engineering, Guangxi Normal University, Guilin 541004, China

Email: yimenwang@foxmail.com

**Abstract**—Nowadays/cloud computing has become a popular platform for scientific applications. Cloud computing intends to share a large number of resources such as equipments for storage and computation, and information and knowledge for scientific researches. Job scheduling algorithm is one of the most challenging theoretical issues in the cloud computing area. How to use cloud computing resources efficiently and increase user satisfaction with jobs scheduling system is one of the cloud computing service providers important goals. Some intensive researches have been done in the area of job scheduling of cloud computing. In this paper we have proposed Greedy-Based Algorithm in cloud computing. In order to prove our opinions we will process this article as the following steps. First of all, we will classify tasks based on QoS. Then, according to the tasks categories, we will select the appropriate branch of the function and compute the justice evaluation. This will also reflect the greedy algorithm to select local optimum. Compare to other methods, it can decrease the completion time of submitted jobs and increase the user satisfaction.

**Index Terms**—cloud computing, job scheduling, Greedy-Based Algorithm, user satisfaction

## I. INTRODUCTION

Job/scheduling system is one of the core and challenging issues in a cloud computing system [1]. However, traditional job scheduling systems in cloud computing only consider how to increase job scheduling efficiency or how to meet the QoS requirements for the resources users, they seldom give description which consider how to combine these above two aspects together. Several job scheduling algorithms have been proposed here as evidences in cloud computing area [2]–[9]. Most of them can be applied in the cloud environment with emphasis on efficiency [3], [4], [6]–[9]. Some of them consider the fairness of users too much to such an extent as to miss the efficiency [2], [5]. Greedy algorithm refers it always make the current best choice in solving the problem. In other words, it is not the whole optimal, but local. Greedy algorithm cannot only be overall optimal solution for all problems, but a wide range of many of the problems that he could produce one or an approximate one. In this paper, we propose the Greedy-Based Algorithm in cloud. The

rest of the paper is organized as follows. In sect.2, we will provide an overview of job scheduling algorithms in cloud computing and make comparisons between them accordingly. In sect.3, we are going to describe the details of algorithm proposed in this paper, which will be supported by sets of data. In sect.4, we intend to offer the descriptions of the simulation experiment and results to further prove our ideas. Finally, we will reach the conclusion part with full descriptions of the core purpose of writing this paper.

## II. RELATED WORKS

Job scheduling has always been a core research area and hence there are a plethora of papers related to this topic can be found in the past decade. Since the development of cloud computing requires technology about virtualization and resource allocation, job scheduling plays a vital role in it. Job scheduling in cloud and grid has an excellent history in research area. A lot of algorithm, technique and strategy have been proposed for this, which include Berger Model [2], Bagof-Tasks [3], Optimal Workflow scheduling [4], Cost-Based Multi-QoS Scheduling [5], Priority Scheduling [6], Utility-Based Scheduling [7] and lots more. [2] Shows a job scheduling algorithm based on Berger Model, the social theories of distributive justice. The algorithm for the establishment of the dual constraint supports the idea of that job scheduling. The simulation result shows that fairness of users is increased to a significant extent with low efficiency. [3] Deals with scheduling multiple applications made of collections of independent and identical tasks on a heterogeneous master-worker platform. The objective is minimizing the maximum stretch i.e. the maximum ratio between the actual time and application has spent in the system and the time. This application would have spent if executed alone. [4] Proposes an Optimal Workflow based Scheduling (OWS) algorithm to find a solution that meets the user-preferred Quality of Service (QoS) parameters. The work focuses on scheduling cloud workflows. By this way a significant improvement in

CPU utilization is achieved. In this paper, we propose the Greedy-Based Algorithm exists in cloud. The algorithm involves 3 phases. First, we classify jobs based on QoS considering the parameters of completion time and bandwidth. Second, According to the job category, we enter different algorithms branch. Finally, we judge the fairness using JEF function [2], [10], [11]. With this algorithm, a signification improvement in CPU utilization is achieved. The experimental results demonstrate that we can decrease the completion time and increases user satisfaction.

### III. PROPOSED ALGORITHM

Our scheduling algorithm mainly consists of three phrases including: classification based on QoS Task-classifier, enter different algorithms branch and justice evaluation, as shown in Figure 1. We classifier jobs in the first phrase; then selected the corresponding algorithm branch; Thus we can get the final fairness evaluation by JEF function, feedback the preprocessing unit and task-classifier, and affect its next operation.

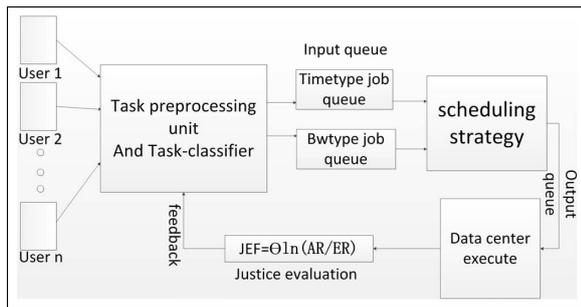


Figure 1. An illustration for Greedy-Based Algorithm.

#### A. Task classification based on QoS

Most of these researches assume that each job has fixed types and amount of execution times. But it is not the case of the real world cloud computing. The above mention issues are highlighted in this section.

- 1) When users submit their job in preprocessing unit, the unit computes the attribute of different jobs and QoS, and encodes the attribute into the users job attribute vector, as following Figure 2.
- 2) Task-classifier, which classifies the task based on attributes, determined by preprocessing unit. For example, job can be classified into different types based on QoS (e.g. completion time, bandwidth). Then it sends jobs to different scheduler branches.

Classype	Start-time	Fin-time	Exp-time	Exp-bw	J-value	Priority
----------	------------	----------	----------	--------	---------	----------

Figure 2. Users Job Attribute Vector.

#### B. Greedy-Based Algorithm

For a set of jobs and the virtual machines, Greedy-Based Algorithm depends on the local optimal method to allocate resources. That is the reason why we called it Greedy-Based Algorithm based on the Greedy algorithm.

A general algorithm for Greedy-Based can be sketched as following Figure 3.

- 1) Users submit their job in preprocessing unit to initial job, and then to form two types of jobs by classifiers.
- 2) According to types of jobs, we created two lists: one for the time type jobs (time type List), another for the bwtype jobs (bwtype List).
- 3) Enter a set of virtual machines named vmList
- 4) Based on the CPUs number of vms and the expectation time of time type jobs, we ascended the vms and time type jobs are ascended.
- 5) At the basis of the actual bandwidth of vms, the expectationbw of bwjobs, we descended the vms and bwtype jobs.
- 6) Using the local optimal algorithm, we respectively bundled jobs in the two tables to a local optimal virtual machine.
- 7) Finally, we calculated JEF function from the expected and actual values, to judge the fairness of users.

Figure 3.

#### C. The justice evaluation function and JEF calculations

[2]Referred to this concept and applied to the fairness of judgment in job scheduling. In our paper, we continued to quote it adding to the Greedy-Based Algorithm to increase the fairness. Please take the definition of justice evaluation function in Berger model [12] as your reference.

$$JEF = \Theta (AR/JR)$$

Where  $\Theta$  denotes to constant ( $0 < \Theta \leq 1$ ). AR is the actual reward, which job obtains actually. JR is the expectation reward which job expected. When JEF is zero, it achieves fairness. Others are not fair. The role of the function is to judge the outcome of the allocation resources whether fair or not .

Before evaluating the function, we first must normalize the jobs and virtual machines. Once the job scheduling has been done, the next step is to normalize the jobs and resources according to the various QoS time, bandwidth.

The calculation of these parameters is specified as follows:

- 1) **Time Preference (TP):** This preference indicates the purpose of choosing resources instances, that which gives the least execution time for jobs. TPI given by Equation 1 as:

$$TP_i = \frac{TR_i}{TR_{max} - TR_{min} + 1},$$

$$0 \leq timetypeLIST.size() - 1 \quad (1)$$

Where  $TR_i = T_{iFin} - T_{iStart}$ ,  $T_{iFin}$  is the finish time of the  $i^{th}$  job,  $T_{iStart}$  is the start time of the  $i^{th}$  job,  $timetypeList.size()$  is the number of jobs preferring time,  $TR_{max}$  is time taken by slowest job,  $TR_{min}$  is time taken by fastest job.  $TR_i$  is time taken by executing  $i^{th}$  job in the job queue.

- 2) **Bandwidth Preference (BP):** This preference indicates the purpose of choosing resources instances which give the appropriate bandwidth for jobs. BPI is given by Equation 2 as:

$$BP_i = \frac{BR_i}{BR_{max} - BR_{min} + 1},$$

$$0 \leq bwtypeLIST.size() \quad (2)$$

Where  $BR_i = BW_{vm.getbw(i)}/bwtypeList.size()$  is the number of jobs preferring bandwidth,  $BR_{max}$  is bandwidth taken by the greatest demand job,  $BR_{min}$  is bandwidth taken by the min demand job,  $BR_i$  is bandwidth taken by executing  $i^{th}$  job in the job queue.

- 3) **Expectation Time Preference (ETP):**This preference indicates the expectation time of job in the first type,  $ETP_i$  is given by Equation 3 as:

$$ETP_i = \frac{ETR_i}{ETR_{max} - ETR_{min} + 1},$$

$$0 \leq i \leq timetypeLIST.size() - 1 \quad (3)$$

Where  $ETR_i = T_{iExpect.time}$ ,  $ETR_i$  is the expectation time of the  $i^{th}$  job in the first type.

- 4) **Expectation Bandwidth Preference (EBP):**This preference indicate the expectation bandwidth of the  $i^{th}$  job in the second type,  $EBR_i$  is given by Equation 4 as:

$$EBP_i = \frac{EBR_i}{EBR_{max} - EBR_{min} + 1},$$

$$0 \leq i \leq bwtypeLIST.size() - 1 \quad (4)$$

Where  $EBR_i = BW_{iExpect.BW}$ ,  $EBR_i$  is the expectation bandwidth of the  $i^{th}$  job in the second type.

- 5) **JEF function(J) and Function Result**

After normalization, we start JEF function as following Equation 5 and Equation 6.

$$J_{0i} = \Theta \ln(TP_i/ETP_i),$$

$$0 \leq timetypeLIST.size() - 1 \quad (5)$$

Where  $\Theta$  denotes constant ( $0 < \Theta \leq 1$ ),  $J_{0i}$  is the J-value of  $i^{th}$  job in the first type,  $TP_i$  is the  $i^{th}$  jobs actual time after normalization,  $ETP_i$  is the  $i^{th}$  jobs expectation time after normalization.

$$J_{1i} = \Theta \ln(BP_i/EBP_i),$$

$$0 \leq i \leq bwtypeLIST.size() - 1 \quad (6)$$

Where  $\Theta$  denotes constant ( $0 < \Theta \leq 1$ ),  $J_{1i}$  is the J-value of  $i^{th}$  job in the second type,  $BP_i$  is the  $i^{th}$  jobs actual bandwidth after normalization,  $EBP_i$  is the  $i^{th}$  jobs expectation bandwidth after normalization. When  $J_{0i}$  ( $J_{1i}$ ) is zero, it achieves fairness. Others are not fair. The role of the function is to judge the outcome of the allocation resources whether or not fair. We also give feedbacks of  $J\_value$  to the classifier, so that it can be modified.

#### IV. SIMULATION EXPERIMENT AND EXPERIMENTAL RESULTS

##### A. Implementation environment

We implement Greedy-Based Algorithm in the CloudSim platform [1], which aims to simulate programs. Of course, we add the algorithm presented by this paper, with overloading the bindCloudletToVM() method in DatacenterBroker class of CloudSim. Cloudlet class is also need to be extended. We add attributes for Cloudlet class, e.g. classtype, Expectationtime, ExpectationBW and related methods,such as setClasstype(), getClasstype(), setExpectationtime(), getExpectationtime(), setExpectationBW(), getExpectationBW(). We also overload the bindCloudletToVM() method in DatacenterBroker class. According to the tasks categories, we select the appropriate branch of the function. This also reflects the greedy algorithm to select local optimum.

##### B. Implementation Data

A group of tasks after the preprocessing unit and task classifier, with the parameters list as following Figure 4. In Figure 4, task 0.3 belong to the preference of time; task 4-7 belongs to the preference of bandwidth by classifier learned, in order to create a group of virtual machine with different performance and preference, as shown in Figure 5.

Cloud -letld	Class -type	Length	File _size	Output _size	Expecta -tiontime	Expecta -tionBW
0	1	4000	2500	500	400	--
1	1	3000	2000	400	200	--
2	1	2000	800	300	150	--
3	1	5000	5000	2000	500	--
4	2	2000	800	300	--	2000
5	2	3000	2000	400	--	3000
6	2	800	300	300	--	1200
7	2	2500	1000	500	--	2000

Figure 4. Cloudlet parameters.

##### C. Experimental results

The experimental results are shown in Figure 6-9. Here you can find we compare the performance of our algorithm (Algorithm 1) with other two scheduling algorithms in the cloud environment, which named job scheduling

algorithm based on Berger model (Algorithm 2) and job scheduling algorithm based on the optimal completion time (Algorithm 3). Algorithm 3 is implemented by using the existing scheduling strategy of CloudSim. Figure 6 shows the task execution time achieved by comparative experimental results. Overall, execution efficiency of algorithm 1 is slightly better than others.

Figure 7 shows the J Value (user satisfaction) achieved by comparative experimental results. Finally, user satisfaction of algorithm 1 is somewhat better than the others.

Figure 8 shows the first type of task achieved by comparative experimental results. It obtains good computing power.

Figure 9 shows comparison of the allocated virtual machines bandwidth for the second type of task. It obtains better fairness than algorithm 3, but slightly worse than

VmId	CPU	Memory	Bandwidth
0	4	2048	2000
1	2	1024	3000
2	2	1024	1200
3	1	512	2500

Figure 5. Virtual machine parameters.

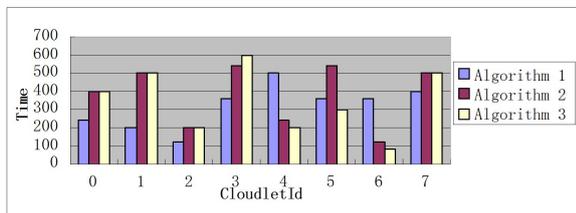


Figure 6. Task execution time comparisons.

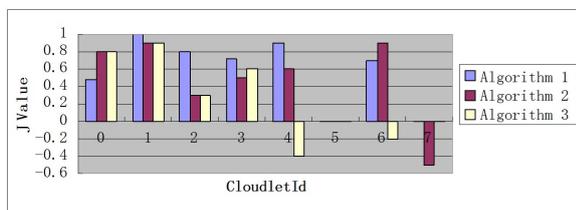


Figure 7. User satisfaction comparisons.

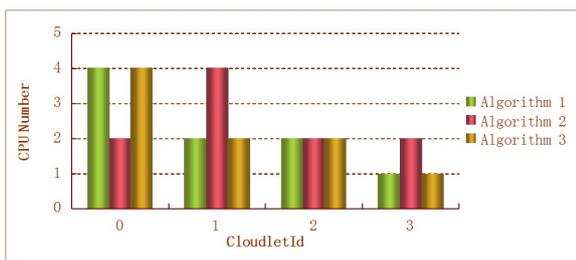


Figure 8. Comparison of the first class task.

algorithm 2.

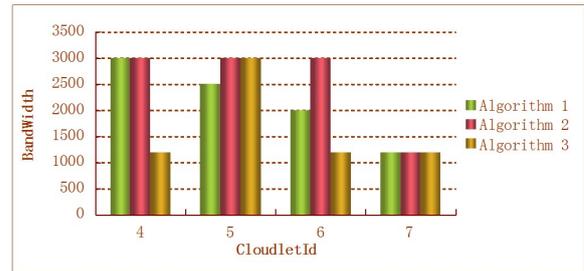


Figure 9. Comparison of the second class task.

### V. CONCLUSIONS

User fairness and efficiency are important issues for job scheduling in cloud environments. As cloud is a business-oriented service, it must concern about both shorter completion time as well as better QoS of cloud customer. In this paper we have proposed Greedy-Based job scheduling algorithm, which can be applied in cloud environments. Result of this paper indicates that the proposed algorithm has decreased the completion time of submitted jobs and increased user satisfaction. In addition, improvement of the proposed algorithm in order to gain more fairness is considered as future work.

### REFERENCES

- [1] P. LIU, *cloud computing*. Publishing House of Electronics Industry, 2011.
- [2] C. Z. Baomin XU, "Job scheduling algorithm based on berger model in cloud environment," *Advances in Engineering Software*, vol. 42, pp. 419-425, 2011.
- [3] M. L. P. J.-F. R. Y. V. F. Benoit, A., "Offline and online master-worker scheduling of concurrent bags-of-tasks on heterogeneous platforms," *Parallel and Distributed Processing*, vol. 190 CCIS, pp. 1-8, 2011.
- [4] A. B. a. P. V. P.Varalakshmi, Aravindh Ramaswamy, "An optimal workflow based scheduling and resource allocation in cloud," *Advances in Computing and Communications*, pp. 411-420, 2011.
- [5] R. C. J. D Dutta, "A genetic algorithm approach to cost-based multi-qos job scheduling in cloud computing environment," *International Conference and Workshop on Emerging Trends in Technology*, 2011.
- [6] M. O. Shamsollah Ghanbari, "A priority based job scheduling algorithm in cloud computing," *International Conference on Advances Science and Contemporary Engineering*, 2012.
- [7] J. H. Zheng HU, Kaijun WU, "An utility-based job scheduling algorithm for current computing cloud considering reliability factor," *Information of China Construction*, pp. 296-299, 2012.
- [8] C.-F. L. Ruay-Shiung Chang, Chih-Yuan Lin, "An adaptive scoring job scheduling algorithm for grid computing," *Information Sciences*, vol. 207, pp. 79-89, 2012.
- [9] L. LI, "An optimistic differentiated service job scheduling system for cloud computing service users and providers," *Multimedia and Ubiquitous Engineering, 2009. MUE '09. Third International Conference on*, pp. 295-299, 2009.
- [10] Z. M. Berger J, Cohen BP, "Status characteristics and expectation states," *Sociological theories in progress*, vol. 1, pp. 29-46, 1966.

- [11] C. T. Z. M. Berger J, Cohen BP, "Status characteristics and expectation states: a process model," *Sociological theories in progress*, vol. 1, pp. 47–74, 1966.
- [12] G. Jasso, "The theory of the distributive-justice force in human affairs:analyzing the three central questions," *Sociological theories in progress:new equation*, vol. SAGE Publications, pp. 357–87, 2012.

**Ji Li** received his Ph.D. degree from Chongqing University in Computer Science. He is currently a professor in the College of Computer Science at the Chongqing University, Chongqing, China. His areas of research are cloud computing and service computing. He has published more than 20 papers.

**Longhua Feng** received her B.E. degree in computer science and technology from Suzhou University, Suzhou, China in June 2011. She is currently working towards her M.E. degree in Technology of Computer Application from Chongqing University, Chongqing 400044, China. Her current research interest includes cloud computing and massive data.

**Shenglong Fang** received his B.E. degree in computer science and technology from Suzhou University, Suzhou, China in June 2011. He is currently working towards his M.E. degree in Computer Science and Information Technology from Guangxi Normal University, Guilin 541000, China. His current research interest includes the Internet of things and flushbonading and cloud computing.