Efficient Load Balancing Algorithm for Cloud Computing Network

Che-Lun Hung[1], Hsiao-hsi Wang[2] and Yu-Chen Hu[2]

[1] Dept. of Computer Science & Communication Engineering, Providence University
200 Chung Chi Rd., Taichung 43301, Republic of China (Taiwan)
[2] Dept. of Computer Science & Information Management, Providence University
200 Chung Chi Rd., Taichung 43301, Republic of China (Taiwan)
{clhung, hhwang, ychu}@pu.edu.tw

Abstract. Recently, Network bandwidth and hardware technologies and Internet services are developing rapidly. Cloud computing as a new Internet service concept has become popular to provide various services to user. Cloud computing employs a variety of computing resources to facilitate the execution of large-scale tasks. Therefore, to select appropriate node for executing a task is able to enhance the performance of large-scale cloud computing environment. In this paper, we propose a scheduling algorithm, LB3M, which combines minimum completion time and load balancing strategies. For the case study, LB3M can provide efficient utilization of computing resources and maintain the load balancing in cloud computing environment.

Keywords: Load Balancing, Cloud Computing, Distributed System, Scheduling.

1 Introduction

Recently, cloud computing as a new internet service concept has become popular to provide various services to user such as multi-media sharing, on-line office software, game and on-line storage.

In a cloud environment, each host as a computational node performs a task or a subtask. The Opportunistic Load Balancing algorithm (OLB) intends to keep each node busy regardless of the current workload of each node [1, 2, 4]. OLB assigns tasks to available nodes in random order. The Minimum Completion Time algorithm (MCT) assigns a task to the node that has the expected minimum completion time of this task over other nodes [4]. The Min-Min scheduling algorithm (MM) adopts the same scheduling approach as the Minimum Completion Time algorithm (MCT) [4] to assign a task the node that can finish this task with minimum completion time over other nodes [5]. The Load Balance Min-Min (LBMM) scheduling algorithm [6] adopts MM scheduling approach and load balancing strategy. It can avoid the unnecessary duplicated assignment. In this paper, we propose an efficient load balance algorithm, named Load Balance Max-Min and Max algorithm (LB3M). From the case study, LB3M achieves better load balancing and minimum completion time for completing all tasks than other algorithms such as MM and LBMM.

2 Method

The progress of LB3M is presented is as following:

Step 1: It is to calculate the average completion time of each task for all nodes, respectively.

Step 2: It is to find the task that has the maximum average completion time.

Step 3: It is to find the unassigned node that has the minimum completion time less than the maximum average completion time for the task selected in Step 2. Then, this task is dispatched to the selected node for computation.

Step 4: If there is no unassigned node can be selected in Step 2, all nodes including unassigned and assigned nodes should be reevaluated. The minimum completion time of an assigned node is the sum of minimum completion time of assigned task on this node and the minimum completion time of the current task. The minimum completion time of an unassigned node is the current minimum completion time for the task. It is to find the unassigned node or assigned node that has the minimum completion time less than the maximum average completion time for the task selected in Step 2. Then, this task is dispatched to the selected node for computation.

Step 5: Repeat Step 2 to Step 4, until all tasks have been completed totally.

In the following section, an example to be executed by using the proposed algorithm is given.

3 Case study

Table 1 shows the completion time for each task at different computing nodes. The threshold is the average completion time of task t_i in all computing nodes. To evaluate the performance of LB3M, LB3M is compared with MM and LBMM by the case shown in Table 1. Figure 1 demonstrates the comparison completion time of each computing node among LB3M, LBMM and MM. The completion times for completing all tasks by using LB3M, LBMM and MM are 24, 33 and 35 seconds, respectively. LB3M achieves the minimum completion time and better load balancing than other algorithms, such as MM and LBMM in this case.

4 Conclusion

In this paper, we proposed an efficient scheduling algorithm, LB3M, for the cloud computing network to assign tasks to computing nodes according to their resource capability. Similarly, LB3M can achieve better load balancing and performance than other algorithms, such as MM and LBMM from the case study.

Table 1. Completion time (second) of each task at different computing nodes.

Node Task	C_{11}	C_{12}	C_{13}	C_{14}	Threshold
t_1	12	13	10	14	12.25
t_2	16	24	13	25	19.5
t_3	26	31	12	33	25.5
t_4	17	24	18	31	22.5

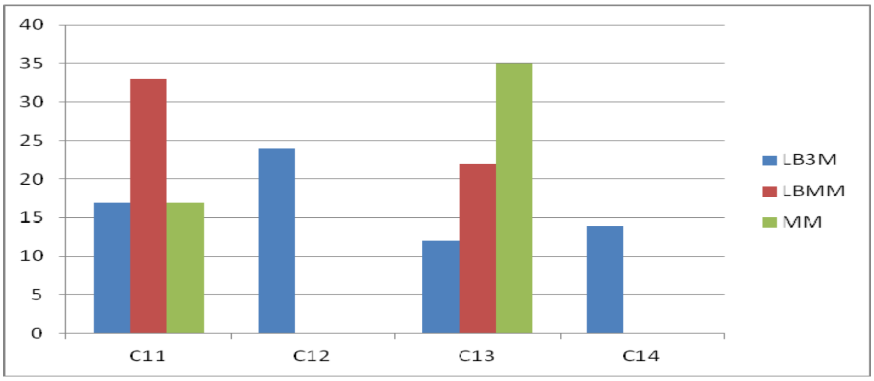

Fig 1. The comparison of completion time of each task at different node for case study.

Acknowledgments. This research was partially supported by the National Science Council under the Grants NSC-99-2632-E-126-001-MY3.

References

1. Armstrong, R., Hensgen, D., Kidd, T.: The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions. In: 7th IEEE Heterogeneous Computing Workshop, pp. 79—87, (1998)
2. Freund, R., Gherrity, M., Ambrosius, S., Campbell, M., Halderman, M., Hensgen, D., Keith, E., Kidd, T., Kussow, M., Lima, J., Mirabile, F., Moore, L., Rust, B., Siegel, H.: Scheduling resources in multi-user, heterogeneous, computing environments with SmartNet. In: 7th IEEE Heterogeneous Computing Workshop, pp. 184—199, (1998)
3. Freund, R. F., Siegel, H. J. : Heterogeneous processing. IEEE Computer, vol. 26, pp.13—17, (1993)
4. Ritchie, G., Levine, J.: A Fast, Effective Local Search for Scheduling Independent Jobs in Heterogeneous Computing Environments. Journal of Computer Applications, vol. 25, pp. 1190—1192, (2005)
5. Braun, T. D., Siegel, H. J., Beck, N., Bölöni, L. L., Maheswaran, M., Reuther, A. I., Robertson, J. P., Theys, M. D., Yao, B., Hensgen, D., Freund, R. F.: A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems. Journal of Parallel and Distributed Computing, vol. 61, pp. 810—837, (2001)
6. Wang, S. C., Yan, K. Q., Liao, W. P., Wang, S. S.: Towards a Load Balancing in a three-level cloud computing network. In: Computer Science and Information Technology, pp. 108—113, (2010)