The Study of Real-Time Scheduling Using Multiple Schedulers in the Cloud Environments

林文鈞、邱紹豐

E-mail: 345472@mail.dyu.edu.tw

ABSTRACT

The emerging Cloud computing has become the de facto distributed computation. It only requires a portal that users can use services provided by distributed computing, and the portal needs to select the machine what provide services for user. The environment is dynamic in the sense that the arrivals of user requests are unpredictable and due to this versatility, traditional scheduling algorithms fail to match user requests to the most feasible processing elements because no prior knowledge about the requests is available until they arrive at the schedulers. The situation is getting worse particularly in a large scaled system, in which the single scheduler is often the bottleneck that the queue is congested with a large amount of user requests. One of the solutions to this problem is to use multiple schedulers. However, the scheduling complexity can rise dramatically since a scheduler must consult others and maintain a global state of all processing elements in order to generate a feasible plan. Sometimes the imposed costs of synchronizing the schedulers and processing elements can turn into the major cause of system degradation. In this research, we take the resource competition approach in which the schedulers compete for the resources without acknowledging other competitors and the processing elements accept only the first request arrives and reject the others. We conduct thorough simulations, and the results about scheduling complexity and processing elements ’ utility are positive.

Keywords : Cloud Computing、Resource Competition、Distributed Scheduling

Table of Contents

封面內頁 簽名頁 中文摘要 iii ABSTRACT iv 誌謝 v 目錄 vi 圖目錄 viii 表目錄 x 第一章 緒論 1 1.1 研究背景 1 1.2 研究動機與目的 2 1.3 論文各章提要 3 第二章 相關研究 4 2.1 雲端運算環境 4 2.2 排程演算法 7 2.2.1 先到先服務 7 2.2.2 最短工作優先 7 2.2.3 最高回應比優先排程法 8 2.2.4 模擬退火演算法 9 2.2.5 螞蟻系統 10 2.2.6 基因演算法 11 2.2.7 最多連外分支優先 12 2.2.8 後繼工作貢獻度的和 14 2.2.9 可分割的負載理論 16 2.2.10 資源感知型動態漸進式排程 17 2.2.11 公平的分散式運算叢集排程 18 2.2.12 開放式最短路徑優先 19 2.2.13 設置中繼排程器的排程系統 20 第三章 研究方法 22 3.1 目標與假設 22 3.2 定義 24 3.2.1 處理單元 24 3.2.2 區域排程器 25 3.2.3 請求 27 3.2.4 回應 30 3.3 處理單元的選擇 31 3.4 協定範例 33 第四章 實驗成果 35 第五章 結論與未來發展 50 5.1 結論 50 5.2 未來發展 51 參考文獻 52

REFERENCES

[1] http://en.wikipedia.org/wiki/Distributed_computing, Wikipedia.

[2] http://en.wikipedia.org/wiki/Cloud_computing, Wikipedia.

[3]S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, “ Optimization by Simulated Annealing, ” Science, vol. 220, pp. 671-679, 1983.

[4]Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni, “ Ant System: Optimization by a Colony of Cooperating Agents, ” IEEE Transactions System, Man and Cybernetics – Part B, vol. 26, no. 1, pp. 29-41, 1996.

[5]Marco Dorigo and Luca Maria Gambardella, “ Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, ” IEEE Transactions on Evolution Computation, vol. 1, no. 1, pp. 53-66, 1997.

[6]John T. Moy, OSPF: anatomy of an Internet routing protocle. Addison-Wesley, 1998.

[7]Norman Bobroff, Gargi Dasgupta, Liana Fong, Yanbin Liu, Balaji Viswanathan, Fabio Benedetti, Jonathan Wagner, “ A Distributed Job Scheduling and Flow Management System, ” ACM SIGOPS Operating Systems Review, Volume 42 Issue 1, pp. 63-70, January 2008.

[8] http://en.wikipedia.org/wiki/Parallel_computing, Wikipedia.

[9] http://en.wikipedia.org/wiki/Grid_computing, Wikipedia.

[10]Ian Foster, http://dlib.cs.odu.edu/WhatIsTheGrid.pdf, July 20, 2002.

[11] http://www.globus.org/toolkit/, Globus Alliance.

[12] http://aws.amazon.com/ec2/, Amazon.

[13] https://skydrive.live.com/, Microsoft.

[14] http://code.google.com/intl/zh-TW/appengine/, Google.

[15] http://aws.amazon.com/s3/, Amazon.

[16] http://workspace.officelive.com/, Microsoft.
[17] https://mail.google.com/, Google.
[18] http://en.wikipedia.org/wiki/FIFO, Wikipedia.
[19] http://en.wikipedia.org/wiki/Shortest_job_next, Wikipedia.
[20]Harvey M. Deitel, Operating Systems, 2nd ed., Addison-Wesley Pub-lishing Company, 1990.
[21] http://en.wikipedia.org/wiki/Simulated_annealing, Wikipedia.
[22] http://en.wikipedia.org/wiki/Ant_system, Wikipedia.
[23]Edwin S. H. Hou, Nirwan Ansari, and Hong Ren, " A Genetic Algorithm for Multiprocessor Scheduling, " IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 2, pp. 113-120, 1994.
[24]Andy S. Chiou, and Chen-Kun Tsung, " Dynamic Scheduling for Jobs in the Grid Environment, " in Proceedings of the 3rd International Confe-rence on Cybernetics and Information Technologies, Systems and Appli-cations, pp. 288-292, July 2006.
[25]Andy S. Chiou and Jung-Yang Shie, " A Dynamic Scheduling Strategy for the Grid, " in Proceedings of the 2008 International Conference on High Performance Computing, Networking and Communication Systems (HPCNCS-08), pp. 222-228, 2008.
[26]V. Bharadwaj, D. Ghose, and T.G. Robertazzi, " Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems, " Cluster Computing on Divisible Load Scheduling, vol. 6, no. 1, pp. 7-18, Jan. 2003.
[27]Sivakumar Viswanathan, V. Bharadwaj, and Thomas G. Robertazzi, " Resource-Aware Distributed Scheduling Strategies for Large-Scale Computational Cluster/Grid Systems, " IEEE Transactions on Parallel And Distributed Systems, vol. 18, no. 10, pp. 1450-1461, Oct. 2007.
[28]Michael Isard, Vijayan Prabhakaran, Jon Currey, Udi Wieder, Kunal Talwar, and Andrew Goldberg, " Quincy: Fair Scheduling for Distributed Computing Clusters, " in Proceedings of the 22nd ACM Symposium on Operating Systems Principles, pp. 261-276, Oct. 2009.
[29]Keithi Ramamritham, John A. Stankovic, and Wei Zhao, " Distributed Scheduling of Tasks with Deadlines and Resource Requirements, " IEEE Transactions on Computers, vol. 38, no. 8, pp. 1110-1123, Aug. 1989.