在雲端環境中以多排程器即時排程之研究

林文鈞、邱紹豐

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

摘 要

現今新興的雲端運算已藉由分散式運算的方式實現，它只需要透過單一的入口即可使用分散式運算提供的服務，而這個入口需要為使用者選擇提供服務的機器。使用者所提出請求的到達的頻率是不可預測的，傳統的排程演算法因為排程器無法事先知道所有將被排程的工作內容資訊，而往往不能產生最佳的工作排程計畫，因此已不適合用來分配使用者的請求給最適合的處理單元。這樣的狀況在大規模的環境中更是嚴重。而若是使用單排程器排程的方法，瓶頸通常在於隊列擠滿了大量的使用者請求等待排程器指派。這個問題擁有一種解決方案，那就是使用多個排程器。 然而，為了產生一個可行的規劃，排程的複雜度將急遽升高，因為排程器必須詢問與維護整體系統中所有處理單元的狀態。有時同步所有排程器與處理單元所需的附加成本會是造成系統效能降低的原因。在此研究中，我們採取資源競爭的方法，其中排程器爭取資源而不必告知其他競爭對手(排程器)，且處理單元只需接受第一個到達的請求，並拒絕其他的請求。我們進一步進行了模擬，在排程複雜度與處理單元使用率等處都得到了正面的結果。

關鍵詞 : 雲端運算、資源競爭、分散式排程

目錄

參考文獻

[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.