Azerbaijan Journal of High Performance Computing, Vol 3, Issue 2, 2020, pp. 190-195
https://doi.org/10.32010/26166127.2020.3.2.190.195

Challenges of Using Different Mathematical Models for Load Balancing Optimization in Multi-Core Computing Systems

Nigar T. Ismayilova

High Performance Computing Research Advance Center, Department of General and Applied Mathematics, Azerbaijan State Oil and Industry University, Baku, Azerbaijan, nigar.ismailova@asoiu.edu.az

*Correspondence: Nigar T. Ismayilova, High Performance Computing Research Advance Center, Department of General and Applied Mathematics, Azerbaijan State Oil and Industry University, Baku, Azerbaijan, nigar.ismailova@asoiu.edu.az

Abstract

This paper examines the role of applying different artificial intelligence techniques for the implementation of load balancing in the dynamic environment of distributed multi-core computing systems. Were investigated several methods to optimize the assignment process between computing nodes and executing tasks after the occurrence of a dynamic and interactive event, when traditional discrete load balancing techniques are ineffective.

Keyword: Exascale Computing, AI, Load Balancer, Graph Matching, Hybrid techniques.

1. Introduction

Load balancing is the process of distributing tasks among computing machines under the conditions that all tasks are executing and all capabilities of resources are using (Alakeel, A. M., 2010). A common strategy used to study optimization of task scheduling process in distributed computing systems is to apply different optimization methods for the solution of matching problem (Ramezani, F., Lu, J., & Hussain, F. K., 2014, Visalakshi, P., & Sivanandam, S. N., 2009, Catalyurek, U. V. et al., 2009). These approaches aimed to solve load-balancing problems for both static and dynamic distributed systems (Godfrey, B., et al., 2004; Di Nitto, E., et al., 2008; Muñoz, P., Barco, R., & de la Bandera, I., 2013).

In static systems (e.g., cluster computing systems), resource attributes and process requirements are predetermined, and load balancing starts its work before execution in the system (Sharma, D., & Aggarwal, V. B., 2015). Optimal load balancing is usually found in these systems to minimize execution time and maximization of resource usage.

For dynamic processes, in which resources, tasks, or both are not static and can be changed during the execution process, different approaches for optimizing load balancing were studied. One of the significant topics investigated in the task scheduling field in dynamic systems is to develop an optimal scheduling strategy sustainable to the changing of resources and types of tasks (Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D., 2011).

Multi-core computing systems or Exascale distributed systems are characterizing

by the occurring of dynamic and interactive events that changes process requirements and resource attributes in the system and requires reloading the task assignment process. The mentioned problem can be solved using two approaches: to allow the occurrence of the dynamic and interactive event and then reload the balancing process or to predict the occurrence of the dynamic and interactive event and to reload the system in a given period (Bakhishoff, U., et al., 2020).

This paper was investigated different approaches based on graph theory and techniques of artificial intelligence for reloading the task matching process periodically during the execution process in multi-core computing systems. The similarities between load balancing and matching problems in graphs were described graph theory approach and proposed load balancer using matching algorithms in bipartite graphs. Besides this, suggested turning to account the advantages of Bayesian networks in missing data to optimize load balancing after occurring of dynamic and interactive event. Besides were discussed opportunities to aggregate power of fuzzy logic and neural networks and apply adaptive neuro-fuzzy inference system (ANFIS) and evolutionary algorithms for developing sustainable load balancer.

2. Related Work

There have been multiple attempts to express the dynamic load balancing system in distributed computing environments by bipartite graphs and to apply different optimization algorithms to a maximum-matching problem in a weighted bipartite graph (Lin, C. C., Chin, H. H., & Deng, D. J., 2013; Wang, Y. C., Peng, W. C., et al., 2007; Kuila, P., & Jana, P. K., 2012). Several approaches have been investigated to express the matching process among computing nodes and task queues in distributed computing systems by bipartite graphs.

Characteristics and capabilities of belief networks allow bipartite graphs representing the dynamic load balancing process in multi-core computing systems smoothly transformed to Bayesian networks. An advantage of modeling the management process by Bayesian networks or belief networks is their ability to analyze data from different sources, including expert knowledge, and handle the learning process in missing data.

The Bayesian approach has been considered by different researchers for implementing the overloading in the load balancing process, as well as for finding most likely haplotypes through parallel search in dynamic systems (Zhao, J., Yang, K., 2015; Otten, L., & Dechter, R., 2011).

As load balancing in HPC systems remains the best assignment between tasks and machines, combinatorial optimization problem solutions can become a useful tool. A recent study by several researchers was analyzed, compared, and critiqued the relevance of using genetic algorithms for task scheduling and load balancing in different HPC systems (Ramezani, F., Lu, J., & Hussain, F. K., 2014; Visalakshi, P., & Sivanandam, S. N., 2009).

The necessity of predicting resource utilization for incoming tasks and, as a result

handling the task scheduling problem enables to use of neural networks in distributed systems (Sigal, L., & Glauberman, A., 2012), the application of neural networks for forecasting loads in cloud data centers under the condition of minimizing energy costs was successfully tested in cloud computing platforms by Prevost and others (Li, J., Luo, G., Cheng, N., Yuan, Q., Wu, Z., Gao, S., & Liu, Z., 2018).

3. Proposition

The main characteristics of multi-core computing systems (or exascale computing systems) are the unstable nature of the execution process and the occurrence of a dynamic and interactive event. Dynamic and interactive events damage the execution process in the distributed system and bring to inconsistency between resources and executed tasks. The goal of load balancing is to control the assignment process and arrange the best matching between resources and tasks according to resource attributes and process requirements. This section was investigated different methods for optimization of the load balancing process in multi-core computing systems.

Specifically, we aim to investigate how to obtain tremendous performance on tackling the time dynamics problem in load balancing to divide time into different time steps and represent each time step by weighted bipartite graphs. Several techniques can be used for evaluating weights on edges between nodes of mentioned graphs.

To outline a similar approach for tackling load balancing problems in multi-core computing systems, it is useful to model the process as time series and represent each time step by parts of weighted bipartite graphs. Weights in these graphs can be determinate as similarities of the assignment process in consecutive time steps. As a result, we get a series of weighted bipartite graph for each consecutive pair of time steps; the second part of the previous time step becomes the first part of the next time step. The machines' optimal assignment tasks can be determined after the implementation of maximum matching algorithms proposed for bipartite graphs. Obtained graph networks allow applying statistical inference and soft computing techniques for automatization of optimal load balancing, which is discussed in the next parts of the section.

The paper suggests applying Bayesian networks to the modeling of dynamic distributed computing systems. This is mainly caused by information deficiency and the impossibility of collecting enormous data for training using learning systems such as deep neural networks or convolutional neural networks. However, the opportunity to express dynamic load balancing process by time series and connect them in the directed acyclic graph gives a chance to apply Bayesian learning algorithms based on conditional similarities between resource attributes and task requirements in each time step, as well as between assignments in each time step and to determine optimal load balancing in each time step.

Dynamic changing both resources and tasks require new approaches and different methods of AI would give beneficial results in developing of Exascale computing

systems. In this case, traditional discrete load balancing mapping finally becomes useless and appears necessary for hybrid load balancing mapping, which can be characterized as a continuous function. The immediate practical solution for this problem might be the application of AI methods and fuzzy logic. Representing load balancing assignment by fuzzy graphs or definition of fuzzy relations between processes or resources at different time moments might be one of the best solutions for job scheduling in distributed systems.

Our description of modeling load balancing in dynamic systems in dynamic and interactive events based on the ANFIS approach is similarly founded on expressing the assignment process by time series and on training ANFIS on each time step. The critical task here is to define the input and output parameters of ANFIS. These parameters can be defined in several ways: task requirements as input variables and resources as output variables, the output node that gets the highest value after implementation will be determined as the best matching for the given task; the next approach is representing input variables as resource attributes and process requirements and as output to define optimal matching as the highest value.

4. Conclusion and Future Work

The paper was studied different approaches defined by mathematical models based on graph theory and artificial intelligence techniques for optimization of load balancing on the exascale computing system. The dynamic and interactive event, which can occur anytime during the execution of a complex task in a multi-core computing system, makes it challenging to manage the job scheduling as the solution was proposed to express the process by consecutive time series and in each time step to apply different techniques of artificial intelligence. The evidence of the missing data and difficulty collecting training examples makes it ineffective to apply deep neural networks. However, different artificial intelligence techniques created as variable based models and logic-based models can help implement effective load balancing in dynamic distributed systems. There is a great deal of work to implement each of these approaches and identify the most successful method for different tasks.

References

Alakeel, A. M. (2010). A guide to dynamic load balancing in distributed computer systems. *International Journal of Computer Science and Information Security, 10*(6), 153-160.

Atayero, A. A., & Luka, M. K. (2012). Adaptive neuro-fuzzy inference system for dynamic load balancing in 3GPP LTE. *International Journal of Advanced Research in Artificial Intelligence, 1(*1), 11-16.

Bakhishoff, U., Khaneghah, E. M., Aliev, A. R., & Showkatabadi, A. R. (2020). DTHMM ExaLB: discrete-time hidden Markov model for load balancing in distributed exascale computing environment. *Cogent Engineering, 7*(1), 1743404.

Catalyurek, U. V., Boman, E. G., Devine, K. D., Bozdağ, D., Heaphy, R. T., & Riesen, L. A. (2009). A repartitioning hypergraph model for dynamic load balancing. *Journal of Parallel and Distributed Computing, 69*(8), 711-724.

Di Nitto, E., Dubois, D. J., Mirandola, R., Saffre, F., & Tateson, R. (2008, November). Applying self-aggregation to load balancing: experimental results. In *Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Sytems* (pp. 1-8).

Godfrey, B., Lakshminarayanan, K., Surana, S., Karp, R., & Stoica, I. (2004, March). Load balancing in dynamic structured P2P systems. In *IEEE INFOCOM 2004 (Vol. 4,* pp. 2253-2262). IEEE.

Kuila, P., & Jana, P. K. (2012). Energy efficient load-balanced clustering algorithm for wireless sensor networks. *Procedia Technology, 6,* 771-777.

Kwok, Y. K., & Cheung, L. S. (2004). A new fuzzy-decision based load balancing system for distributed object computing. *Journal of Parallel and Distributed Computing, 64*(2), 238-253.

Lee, S. P., & Nahm, E. S. (2012, August). Development of an optimal load balancing algorithm based on ANFIS modeling for the clustering web-server. In *International Conference on Hybrid Information Technology* (pp. 783-790). Springer, Berlin, Heidelberg.

Li, J., Luo, G., Cheng, N., Yuan, Q., Wu, Z., Gao, S., & Liu, Z. (2018). An end-to-end load balancer based on deep learning for vehicular network traffic control. *IEEE Internet of Things Journal, 6*(1), 953-966.

Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D. (2011, August). Cloud task scheduling based on load balancing ant colony optimization. In *2011 Sixth Annual China Grid Conference* (pp. 3-9). IEEE.

Lin, C. C., Chin, H. H., & Deng, D. J. (2013). Dynamic multiservice load balancing in cloud-based multimedia system. *IEEE Systems Journal, 8*(1), 225-234.

Muñoz, P., Barco, R., & de la Bandera, I. (2013). Optimization of load balancing using fuzzy Q-learning for next generation wireless networks. *Expert Systems with Applications, 40*(4), 984-994.

Naaz, S., Alam, A., & Biswas, R. (2011). Effect of different defuzzification methods in a fuzzy based load balancing application. *International Journal of Computer Science Issues (IJCSI), 8*(5), 261.

Otten, L., & Dechter, R. (2011). Finding most likely haplotypes in general pedigrees through parallel search with dynamic load balancing. In *Biocomputing 2011* (pp. 26-37).

Ramezani, F., Lu, J., & Hussain, F. K. (2014). Task-based system load balancing in cloud computing using particle swarm optimization. *International Journal of Parallel Programming, 42*(5), 739-754.

Sethi, S., Sahu, A., & Jena, S. K. (2012). Efficient load balancing in cloud computing using fuzzy logic. *IOSR Journal of Engineering, 2*(7), 65-71.

Shaout, A., & McAuliffe, P. (1998). Job scheduling using fuzzy load balancing in distributed system. *Electronics Letters, 34*(20), 1983-1985.

Sharma, D., & Aggarwal, V. B. (2015). An effective mechanism for improving performance of load balancing system in cluster computing. *International Journal of Computer Applications, 115* (7), 21-27.

Sigal, L., & Glauberman, A. (2012). U.S. Patent No. 8,185,909. Washington, DC: U.S. Patent and Trademark Office.

Suresh, M., & Karthik, S. (2014, March). A load balancing model in public cloud using ANFIS and GSO. In *2014 International Conference on Intelligent Computing Applications* (pp. 85-89). IEEE.

Visalakshi, P., & Sivanandam, S. N. (2009). Dynamic task scheduling with load balancing using hybrid particle swarm optimization. *Int. J. Open Problems Compt. Math, 2*(3), 475-488.

Wang, Y. C., Peng, W. C., Chang, M. H., & Tseng, Y. C. (2007, August). Exploring load-balance to dispatch mobile sensors in wireless sensor networks. In *2007 16th International Conference on Computer Communications and Networks* (pp. 669-674). IEEE.

Zhao, J., Yang, K., Wei, X., Ding, Y., Hu, L., & Xu, G. (2015). A heuristic clustering-based task deployment approach for load balancing using Bayes theorem in cloud environment. *IEEE Transactions on Parallel and Distributed Systems, 27*(2), 305-316.

Submitted: 26.06.2020
Accepted: 23.11.2020