International Journal of New Technology and Research (IJNTR)
ISSN:2454-4116, Volume-2, Issue-4, April 2016 Pages 56-58

Fuzzy Real Time Scheduling on Distributed Systems to Meet the Deadline of Applications

Mrs Rekha A Kulkarni , Dr. Suhas H Patil, Dr N.Balaji

Abstract— **As there is fast change in the design of computing systems, distributed computing is gaining prominence. Fast processing with large data sets is becoming need of the era.. In distributed systems different kinds of hardware and software work together in cooperative fashion to solve problems involving large amount of data. In a heterogeneous environment handling real time tasks poses challenges of processor scheduling and load balancing to meet the deadlines. Implementation of real-time tasks poses a lot of problems due to the unpredictability of the tasks involved and due to the lack of complete task knowledge prior to the execution process. Fuzzy logic can be used in dynamic scheduling of these tasks based on the load information at various processing units. Fuzzy logic uses the system's deadline miss ratio (DMR) and system throughput to calculate a value which provides a system performance metric used to drive the system to the desired level of performance.**

Index Terms— **Distributed systems, Fuzzy logic, Dead line miss ratio.**

I. Introduction

Most of the practical real-time scheduling algorithms in heterogeneous systems present a trade-off between their computational complexity and performance. In real-time systems, tasks have to be performed correctly and timely. Finding minimal schedule in multiprocessor systems with real-time constraints is shown to be NP-hard. The practical scheduling algorithms in real-time systems don't have deterministic response time. Deterministic timing behavior is an important parameter for system robustness analysis. The intrinsic uncertainty in dynamic real-time systems increases the difficulties of scheduling problem. To alleviate these difficulties, A fuzzy scheduling approach can be employed to arrange real-time periodic and non-periodic tasks in systems. Static and dynamic optimal scheduling algorithms fail with non-critical overload. The knowledge based algorithms can be devised to improve the performance of the heterogeneous system.

II. Related Work

Cloud computing provides a distributed computing environment in which there is a pool of virtual, dynamically scale-able heterogeneous computing and storage platforms. The computing power and the storage are provided as Services on demand to external user over the internet. Cloud computing aims at making computing as a service whereby shared resources, software, and information are provided to users who are requesting the service as a utility over a network. One of the key technologies which plays an important role in Cloud data-center is resource scheduling. Real time processing on the cloud systems requires satisfying timing constraints of real time systems. In the scenario of time dependent application deadline misses are undesirable. One of the challenging scheduling problems in Cloud data center is to consider allocation and migration of re-configurable virtual machines (VMs) and integrating features of hosting physical machines.

Earliest Deadline First (EDF)[1] algorithm always chooses the task with the earliest deadline. It has been proved that this algorithm is optimal in a uni-processor system. Since it cannot consider priority and therefore cannot analyze it, this algorithm fails under overloading conditions.

Least Laxity First (LLF) [1]algorithm selects the task that has the lowest laxity among all the ready ones whenever a processor becomes idle, and executes it to completion. This algorithm is non-preemptive and avoids the well-known problem of its preemptive counterpart that sometimes degenerates to a processor-sharing policy.

While using the facilities provided by cloud environment in the area of real time processing the challenge is that of meeting the deadlines. A real-time scheduler must ensure that processes meet deadlines, regardless of system load or inherent delays caused by the various components of the cloud environment. One solution to this is by selecting the scheduling policy at run time. After studying the various real time scheduling algorithms, in this research dynamic selection of scheduling policy based on fuzzy logic has been proposed. This is possible by identifying and using the appropriate parameters in selecting the scheduling policy. In the proposed system we have tried to identify the parameters and the effect of these different parameters on the selection policy. The proposed system will try to improve the performance in real time.

III. Proposed System

The performance of a scheduling algorithm is measured in terms of additional processor required to be added at a schedule without deadline violations as compared to optimal algorithm it has been proved that finding a minimal schedule for a set of real-time tasks in multiprocessor system is NP-hard.

With the above analysis, the ability to satisfy timing constraints of such real-time applications plays a significant role in distributed environment. However, the existing

Mrs Rekha A Kulkarni , PICT, Pune
Dr. Suhas H Patil, BVU,COE ,Pune
Dr N.Balaji, JNTUK Vijaynagarum

schedulers are not perfectly suitable for real-time tasks, because they lack strict requirement of deadlines. A real-time scheduler must ensure that processes meet deadlines, regardless of system load or inherent delays which are caused by various components of distributed systems.

Our Proposed system S will be represented as

S={ I,O,F,Fa,Su}

Where

I=Set of inputs {t1……..tn}{p1…..pn}

O={ Set of output } finding optimal mapping of tasks to processors.

F= { Set of functions f1,f2,f3…}

Where f1 is gathering of information about various resources in system. F2 generation of scheduling policy based on the parameters collected from f2. F3 is to decide which is the best policy for given application.

Su : Success case when scheduling policy improves the performance by meeting the dead line of real time task.

Fa : failure case when deadline of time critical applications are not being met.

Fuzzy [2] inference is the process of formulating the mapping from a given input set to an output using fuzzy logic.

Laxity is the maximum time that a task can wait before being executed (i.e., laxity = deadline - computation time).

Much of the power of fuzzy logic is derived from its ability to draw conclusion and generate responses based on vague, incomplete, and imprecise qualitative data.

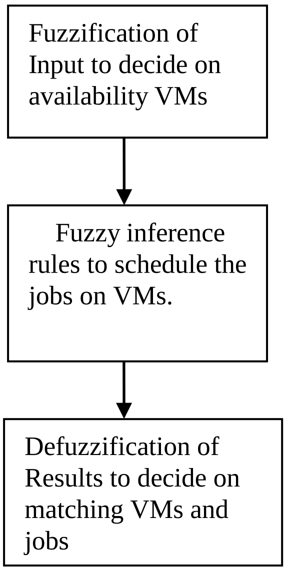

Fig 1 : Use of Fuzzy Logic for Scheduling

Fuzzy logic contributions could be in the form of approximate reasoning, where it provides decision-support and expert systems with powerful reasoning capabilities bound by a minimum number of rules. With a reliable set of inference rules, inputs are converted into their fuzzy representations during the fuzzification process, and the output generated is then converted back into the "crisp" or numerically precise solutions during the defuzzification process. The rules that determine the fuzzy and crisp predicates for both the fuzzification and the defuzzification processes are constructed from sources like past history, neural network/neuro-fuzzy training and numerical approximation.

Fuzzy logic model for dynamic task scheduling for multi-processing systems that performs load balancing is proposed. It takes Processor Execution Length and Processor Delay and Task Length as inputs in the fuzzy inference system and produces the degree of "Processor Acceptance" as output, which determines which processor to allocate to it the current task and its associated degree or value of acceptance.

IV. Implementation and Results

Implementation of above said approach can be done with the help of cloud simulator which can be used in the laboratories for experimentation. We have used CLOUDS in as simulation tool to study the effect of fuzzy logic on scheduling policy. The fuzzy logic helps to meet the deadlines of the tasks.

V. Conclusion

Meeting the deadline for Resource and Timing Constraints is a Challenge in a cloud system. While working with shared environment users will compete for resources to meet their timing deadlines. In such a scenario, resource scheduling plays a critical part in meeting an application's expectations. In large clusters, tasks complete at such a high rate that resources can be reassigned to new jobs on a timescale much smaller than job duration.

The job to be scheduled next according to fairness might not have data on the nodes that are currently free. In a heterogeneous environment like cloud where processing speeds are likely to vary among nodes, high processing nodes are expected to complete more tasks. In such a scenario, the concept of equally distributing data in order to provide data locality is likely to create network bottleneck. For heterogeneous systems, effective data placement strategies are required in order to ensure efficient task scheduling. So the proposed system tries to select a scheduler at run time which can meet the timing requirements based on the information available locally and globally.

With the help of fuzzy logic it is possible to address the uncertainties involved while dealing with large amount of processors. With the help of Cloud simulators it is possible to create an infrastructure for meeting the deadlines in real time tasks.

References

[1] K. Liu, H. Jin, J. Chen, X. Liu, D. Yuan, and Y. Yang, "A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost- Constrained Workflows on Cloud Computing Platform." J. of High Performance Computing Applications, 24:445–456, Nov 2010.

[2] J. Gu, J. Hu, T. Zhao, and G. Sun, "A new resource scheduling strategy based on genetic algorithm in cloud computing environment," J. Comput., vol. 7, no. 1, pp. 42–52, 2012.

[3] A. Oprescu and T. Kielmann, "Bag-of-Tasks Scheduling under Budget Constraints." in 2nd International Conference on Cloud Computing, Los Angeles, USA, 2009.

[4] A. Oprescu and T. Kielmann, "Bag-of-Tasks Scheduling under Budget Constraints." in 2nd International Conference on Cloud Computing, Los Angeles, USA, 2009.

[5] S. H. Jang, T. Y. Kim, J. K. Kim and J. S. Lee, "The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing," IJCA, vol. 5, no. 4, pp. 157-162, December 2012.

[6] H. Ye, J. Wong, G. Iszlai and M. Litoiu, "Resource Provisioning for Cloud Computing," Proceedings of CASCON 2009, November 2009.

[7] T. Henzinge, A. V. Singh, V. Singh, T. Wies and D. Zufferey, "Flex-PRICE: Flexible Provisioning of Resources in a Cloud Environment," In CLOUD, 2010.

[8] M. Paul and G. Sanyal, "Task-scheduling in cloud computing using credit based assignment problem," International Journal on Computer Science and Engineering (IJCSE) 3.10, pp. 3426-3430, 2011.

[9] S. Xue and W. Wu, "Scheduling workflow in cloud computing based on hybrid particle swarm algorithm," TELKOMNIKA Indonesian Journal of Electrical Engineering 10.7, pp.1560-1566, 2012.

[10] X. Wu, M. Deng, R. Zhang, B. Zeng and S. Zhou, "A task scheduling algorithm based on QoS driven in cloud computing," Advances in Engineering Software, Procedia Computer Science, 17, pp. 1162-1169, 2013.

[11] B. M. Xu, C. Y. Zhao, E. Z. Hu and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," Advances in Engineering Software, 42(7), 419-425, 2011.

[12] I. A. Moschakis and H. D. Karatza, "Performance and cost evaluation of Gang Scheduling in a Cloud Computing system with job migrations and starvation handling," In: Proceedings of IEEE Symposium on Computers and Communications (ISCC), pp. 418–423, 2011.

[13] S. Ghanbari and M. Othman, "A priority based job scheduling algorithm in cloud computing," Procedia Engineering, 50, pp. 778-785, 2012.

[14] L. F. Bittencourt and E. R. M. Madeira, "HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds," Journal of Internet Services and Applications, vol. 2, pp. 207–227, 2011.