Optimized Algorithm For Overloaded Real Time System

Surbhi Singh
Department of Computer Science & Engineering
Research Scholar, Deenbandhu Chhotu Ram University of Science & Technology, Sonipat

Abstract- *In twenty first century, Real Time Systems is studied as one of the important technologies. These systems provide extraordinary opportunities for a variety of civilian and military application, for example Navigation of the Air Traffic, Nuclear Plant management, ICU devices etc. In today environment we use real time systems in so many ways that even we don't know that we are using real time systems cars, planes and entertainment systems are some of them. Scheduling Algorithms are integral part of the real time systems. There are many different algorithm in real time systems. In this study we present an adaptive algorithm that will works well on both uniprocessor and multiprocessor systems.*

Keywords- Uniprocessor, Multiprocessor, Real Time Systems

I. INTRODUCTION

Real-time system are systems in which correctness is depend on the time at which the results are produced. In Real-time systems execution of transactions must be completed within the time limit otherwise the system will fail. In other words real time systems are designed to operate in a timely manner. These systems must meet the predefined deadlines while producing results. Some important operations may be subjected to timing constrains .violation of these timing constraints can cause safety breaches or financial loss.

The unique characteristics present many challenges in the design of real time transaction processing systems, which involve the following main issues:

i. Large and Complex: - Real time systems can have a few hundred lines code or C to the million lines of code that is estimated for the space station freedom. This leads to the increment in complexity to design a processing algorithm.
ii. Concurrent control of system components: - In the real world all devices operates in parallel and needs to effectively manage. A real time
system must effectively control all the device that are a part of the whole system, concurrently.
iii. Facilities to interact with special purpose hardware: -Real time systems provides facilities to interact with special purpose hardware that gives output in real time or near real time. So this also lead to increase complexity of the transaction processing systems in RTS
iv. Mixture of Hardware and Software: - Real time systems are the mixture of hardware and software components. Some specific purpose hardware is assembled with the software to build a reliable system that meet the performance criteria.
v. Extreme reliability and safety: - Real time systems must be extremely reliable because failure of the real time system can cause loss of life or a big financial loss. So that this also lead to increase complexity.
vi. Guaranteed response times: - In real time systems we need to be able to predict the worst case response time with full confidence. Efficiency is important in real time systems but predictability is most necessary.

II. TRANSACTION SCHEDULING IN REAL TIME SYSTEMS

In Real time system for a transaction three type of deadlines can be given Soft, Firm and Hard. Soft deadlines are the deadlines that are desirable but not mission-critical means if system misses a soft deadline it will not cause a system failure example of a soft deadline is operator switchboard for a telephone. Firm Deadline, like soft deadlines, are desirables but not critical A task with a firm deadline cannot be executed after its deadline and so system don't get any value from the task that miss deadline Example of a firm deadline is an autopilot system. Hard Deadline are the critical deadlines means if a hard deadline is missing, it can lead too catastrophic consequences. Logically correct and timely execution is must for the hard deadlines. Hard deadlines are also known as safety critical example of hard deadline is data gathered by a sensor.

III. TRANSACTION SCHEDULING TECHNIQUES IN REAL TIME SYSTEMS

There are two categories of Real-time scheduling techniques.

i) Static, known as fixed-priority where priorities are computed off-line. Static scheduling requires complete priori knowledge of the environment which is deployed. Static scheduling techniques are inflexible this is workable only if all tasks are

effectively periodic. Static schemes can works for simple systems. It cannot works consistently as the load increases.
ii) Dynamic scheduling that assumes unpredictable task-arrival times. Dynamic scheduling attempts to schedule tasks dynamically when task arrived it assigns a priority value to each task dynamically at run time based on parameters of tasks that used in execution. Dynamic scheduling is of two types- static priority and dynamic priority. Static priority is also of two type's rate monotonic and deadline monotonic. EDF and LST are examples of dynamic priority scheduling.

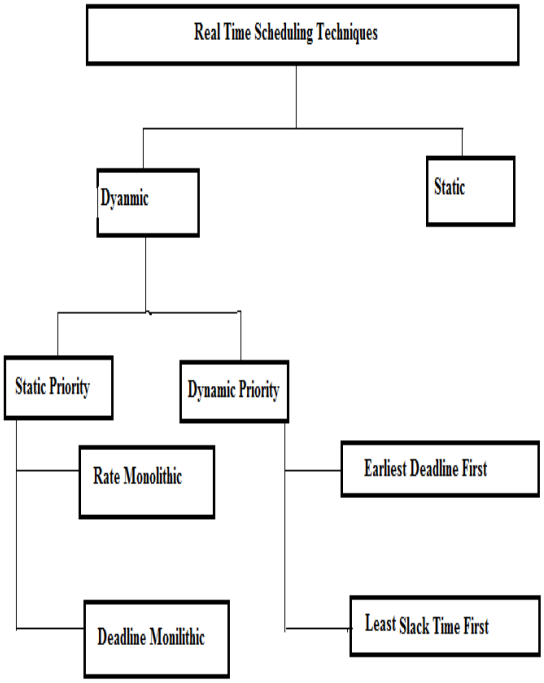

Figure 1: Categories of scheduling techniques

IV. ANT COLONY OPTIMIZATION (ACO) AND LEAST SLACK TIME FIRST (LST) ALGORITHM

Ant Colony Optimization is a type of Swarm Intelligence. Ant Based system provides the advantages such as scalability, parallelism, robustness along with the simplicity of agent. Ant colony optimization (ACO) was developed by M. Dorigo as a solution that is inspired by nature for hard combinatorial optimization problems. ACO is a type of meta-heuristics which is used to find best solutions to hard CO problems in a reasonable amount of computational time.

The behaviour of real ants is the inspiring source of ACO. When ants searches for food, it explore the surrounding area of their nest in a random manner. AS an ant finds the food, it predicts the quantity of the food and bring some of the food to the nest. When ant comes back to the nest it deposits a chemical pheromone trail on the route, which guide other ants to the food source. If obstacle is find onto that path the ants randomly choose another way around the obstacle. Suppose that the obstacle is too high that the ants cannot go wider or over the obstacle then it is assume that at first time 50% of the ants will go right and other fifty percent will go left. On the shorter path ants creates a strong pheromone trail a lot faster than the ants those chooses longer path.

Ant colony optimization applies the same techniques that real ants uses to find the shortest path to the food, in artificial way to solve complex problems. The idea behind this is to use artificial ants as agents to provide new solutions to the problems. ACO uses two steps to solve the problem (1) scheduling is provided using a pheromone model (2) schedules are used to modify the pheromone values.

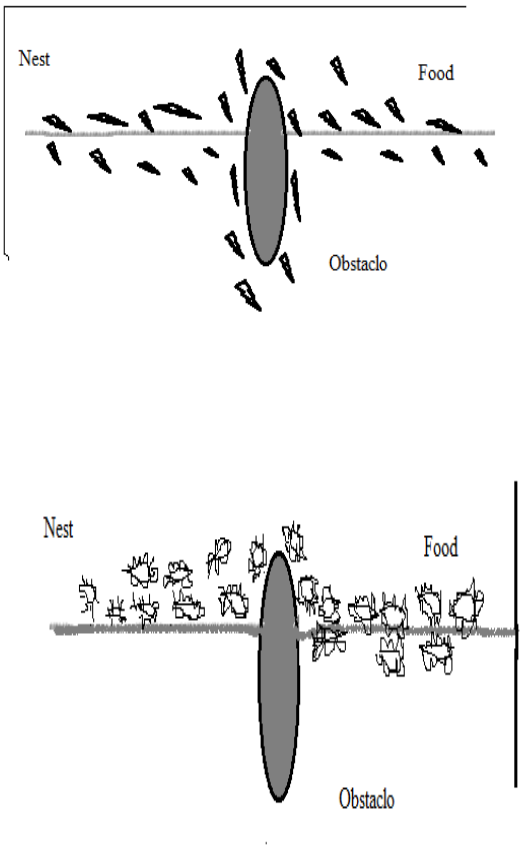

Figure 2: ACO approach

Least slack time first (LST) algorithm is based on the rule that task or transaction that have the smaller slack time will assign the highest priority in the transaction queue. Suppose di= relative deadline, ri = release time, ei = execution time, t = current time. The slack time can be viewed as remaining spare time (di – ei r - t) at the current time. ei r is the time that is required to complete the remaining work of a task.

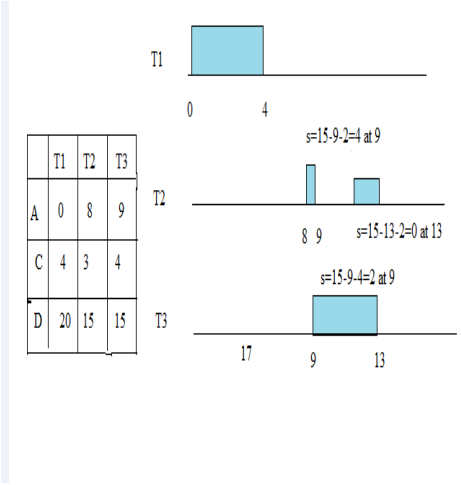

Figure 3: LST Example

The idea behind the LST algorithm is that there is no point to complete a transaction before its deadline. Soft tasks can be executes before them. LST is optimal algorithm in finding feasible schedules for uniprocessor real time systems. In the above example A C and D is used for arrival time, computation time and relative deadline.

V. SCOPE OF THE STUDY

Real time systems are the most important systems that uses in the real world. Like in almost every field human wants results as quickly as possible and some areas such as navigation of air traffic, missile launch wants the results in the real time in these area if some data is delayed or missing then results may be disastrous. So very reliable system have been developed. There are many transaction scheduling algorithms for RT systems but some are optimal for uniprocessors and some for multiprocessors. Many researchers proposed many combinations of the algorithm that increase the efficiency of the system. The scope of this study basically lies in the fact that first understood the real time systems how these systems works and what techniques these system uses for scheduling the transactions. The previous studies in the field of real time transaction scheduling shows that more algorithms should be developed to run on the both multiprocessor and uniprocessor RT systems because most of the RT systems uses more than one processor to schedule the transactions. Now a day research is carried forward to implement RT systems in the new areas where RT systems not implemented yet but scheduling is also a important part of the current studies.

So the scope of this study is to implement a new real time transaction scheduling algorithm that works fine on both uniprocessor and multiprocessors system. This algorithm will use the concepts of existing algorithm but proposed a new implementation style.

VI. PROPOSED SYSTEM

In real time systems some algorithms are optimal for the one processor system that scheduled one transaction at a time such as LST first algorithm and some are optimal for the systems that schedules multiple transactions at a time. The main problem in this type of algorithms is that algorithms that are optimal for uniprocessors does not perform well when system is overloaded means when system misses any deadline. During overload situation ACO perform well but takes more execution time. If we uses LST algorithms then if system become overloaded it does not perform well. But if we uses ACO algorithm then it takes more execution time even in the situation of underloaded. So the problem is to find such a way that if system is in underloaded situation then it uses the concept of LST and if system is in overload situation then it will automatically switch to the ACO.

The proposed algorithm is described as following-

Step1- At the start of the processing, system will use LST algorithm
Step2- If system misses any deadline switch it to the ACO algorithm
Step3- automatically uses the deadline that are defined for ACO
Step4- if system continuously executes 10 transactions within deadline then it will automatically switch to LST.
For this algorithm continuously automatic monitoring of the system is necessary.

VII. CONCULSION

This study is carried forward to review the scheduling techniques of the real time systems and to proposed and implement a new algorithm that works optimal for both the conditions underload and overload. Underload means when system executes the transaction efficiently and no deadline is missed. A system is in overloaded condition when it misses some deadline of the transactions. In this study we review many techniques that are implemented so far and after reviewing these algorithms we implemented a combined algorithm That is optimal for real time scheduling.

REFERENCES

[1] Kaladevi M., Sthhiyabama Dr. S."A Comparative Study Of Scheduling Algorithms For Real Time Task" International

Journal of Advances in Science and Technology November issue, Vol. 1, No. 4, 2010

[2] Zhao Yuehua, Qiu Jing(2010) "A new multi-dynamic priority real-time database scheduling algorithm" 2nd International conference on computer Engineering and Technology V7-178

[3] Wu Xian-De, Sun Zhao-Wei, Xing Ze-Jing "A Data-Centered Transaction Scheduling Strategy of real-time Database in Micro-Satellite Ground Test Systems" International Conference on Mechatronics and Automation August 9-12, Changchun, China 2009 IEEE

[4] Miryani Mohammad Reza, Naghibzadeh Mahmoud (2009) "Hard Real-Time Multobjective Scheduling in Heterogeneous Systems Using Genetic Algorithms" 14th International CSI Computer Conference (CSICC'09) 17

[5] Luo Wei, Qin Xiao, Bellam Kiranmai (2007) "Reliability-Driven Scheduling of Periodic Tasks In Heterogeneous Real-Time Systems" 21st International Conference on Advanced Information Networking and Application Workshop(AINAW'07) IEEE 2007

[6] Idoudi Nizar, Louati nada, Duvallet Claude, Bouaziz Rafik, Sadeg Bruno and Faiez Gargouri" A Framework to Model Real-Time Databases" International Journal of Computing & Information Sciences Vol. 7, No.1, January 2009, On-Line Pages 1-11

[7] Sachindran Narendran, Gupta Manish "Real-Time Problem Localization For Synchronous Transactions In HTTP- Based Composite Enterprise Applications" 978-1-4244- 2066-7/08 2008 IEEE