MultiCraft

International Journal of Engineering, Science and Technology
Vol. 7, No. 3, 2015, pp. 118-122

INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND TECHNOLOGY

www.ijest-ng.com
www.ajol.info/index.php/ijest

Energy efficient approach for transient fault recovery in real time systems

Arvind Kumar, Bashir Alam

Department of Computer Engineering, Faculty of Engineering and Technology, JamiaMilliaIslamia New Delhi-110025, INDIA
[*]Corresponding Author: e-mail: arvinddagur@gmail.com

Abstract

Due to increase in complexity of systems, failure rate is increasing in real time systems. A failure in a system can occurs due to presence of fault. A fault can be transient, intermittent or permanent. In this paper we propose a novel approach for transient fault recovery with energy minimization. A faulty system may cause a failure. Fault tolerance approach is used to correct functioning in the presence of fault. DVS (Dynamic Voltage Scaling) is responsible for energy minimization in a system by adjusting the voltage level and clock speed dynamically. Efficiency of DVS depends upon available slack time. Experimental result shows that proposed algorithm is more suitable for transient fault recovery with energy minimization.

Keywords: DVS, Fault tolerance, Real Time System, Transient Fault.

DOI: http://dx.doi.org/10.4314/ijest.v7i3.14S

1. Introduction

Real time systems are highly dependent upon task deadline. Efficiency of these systems are depends not only on task deadline but it is also depend upon their correct functioning. The real time system can be divided into two categories: hard real time system, in which missing the deadline may cause a failure and soft real time system, in which missing deadline may be consider. Most of these systems are time critical systems operate in fault prone environmental conditions. Faulttolerant computing techniques are used to achieve correct result. Fault tolerance is mechanism to provide correct functioning of system (user program and system software) in the presence of fault (Pradhan, 1986). As the system is hard real time system, it is essential that the task complete its functioning before deadlines even in the presence of fault. Fault tolerance approaches are needed in time critical systems to ensure dependability in harsh environment conditions. There are many approaches which have been investigated for fault tolerance. Some of them are active replication, checkpointing, replay logs, etc provide different tradeoff in runtime cost in the absence of fault. Checkpointing is very common approach for tolerate transient fault in the system. At each checkpoint, the system checks the correctness and save the result ina secure device. If a fault is detected, the system rolls back to the previous checkpoint instead of complete re-execution and resumes normal execution (Liu et al, 2010).

These systems are also energy constraint because the lifetime of these systems depends upon battery life. Energy is also main issue for most of the real time devices. It is very crucial due to limited power backup. Energy can be managed by dynamic power management (DPM) in real time systems (Huang *et al*, 2009). DVS (Dynamic Voltage Scaling) is very popular and most effectively approach used for energy minimization. The energy minimization can be achieved by dynamically scaling the voltage and its corresponding frequency (Woomseok et al, 2002) .

There are many fault tolerance and energy minimization approaches have been studied separately. DVS is not considered for fault tolerance and fault tolerance is not considered for dynamic voltage scaling. In this paper, both the approaches are combined to find a suitable approach which can be effectively used for fault tolerance and energy minimization. Fault tolerance approach is combined with DVS to achieve fault tolerance and energy efficiency.

Rest of the paper is organized as follows. The related work with motivation is presented in section II. Problem is described in section III. Section IV includes models and assumptions used. The proposed algorithm is presented in section V. Section VI shows the results and analysis of this paper and finally section VII concludes the paper.

2. Motivation and Related Work

As most of real time systems working in crucial conditions are highly depend upon time with most chance of occurrence of fault. These systems should be fault tolerated system such that if a fault occurs in the system it can be handled on running time. There are several approaches for fault tolerance proposed in many papers (Dima et al, 2001). Most of the approaches are based on checkpointing to tolerate transient fault. The major consideration is to adjust the intervals between checkpoints to reduce energy consumption. The checkpoint interval is the duration between two consecutive checkpoints. The checkpoint interval can be reduced by increasing number of checkpoints. The system is more accurate in fault tolerance if there is more number of checkpoints. Many approaches have been proposed for finding optimal number of checkpoints (Zhang et al, 2003). Checkpointing and rollback recovery are used to improve the reliability and availability of fault-tolerant computing systems. The main problem of checkpointing is how to determine strategies for optimal placement of checkpoints. The performance of the system is depends upon placement of checkpoint. The system will be more efficient and reliable with fault tolerance if an optimal number of checkpoints have been applied. The number of checkpoints affects the execution time, task response time and expected cost (Izosimov et al, 2008).

Energy is also the concern in real time systems. Some of the time critical system working in remote areas with no power supply, where battery is only the energy source or shortage of energy supply we need to minimize the energy consumption. The energy consumption will increase to execute a faulty task. A system will consume more energy to re-execute the task (Melhem et al, 2004). This energy consumption can reduces by applying suitable energy management scheme. Dynamic power management is best approach used for real time applications. DVS dynamically adjust the supply voltage and its corresponding frequency to save the power (Pillai et al, 2001).

If voltage V is applied for execution with f clock frequency then energy consumption E can be expressed by using given equation.

$$\mathbf{E = V^2 f}$$

By scaling the voltage on various voltage levels, energy consumption can be managed. If a task running on maximum voltage and have available slack, then by reducing voltage energy can be saved.

Earliest Deadline First (EDF) and RM (Rate monotonic) scheduling approach are used for scheduling real time task. EDF is most efficient dynamic scheduling approach. EDF does not provide any fault tolerance mechanism and energy management scheme. EDF is responsible for scheduling real time task dynamically. These tasks are aperiodic in nature. Aperiodic task are those which can arrive any time in a system. Every aperiodic task have <ai, ei, di>, where ai is arrival time,ei is execution time anddi is deadline of the task (Alam et al, 2014). Due to high occurrence of transient fault in a system, we consider only transient fault recovery. A transient fault can be recovered using time redundancy. In time redundancy the faulty task is roll back or re-execute during running time. In this paper, we propose an energy efficient approach which can tolerate transient fault in real time applications with energy minimization.

3.Problem Description

In real time critical applications, fault tolerance with energy minimization is a big issue. Many fault tolerance approach has been proposed but fault tolerance is not be able to manage energy consumption. There are some energy management approaches which are not capable for fault tolerance. The main issue of this paper is to minimize the energy consumption with fault tolerance in real time applications.

4. Models and Assumptions

Real Time Task Model: The task model consist a set of aperiodic task T = { T_1, T_2, ... ,T_n } having their respective deadline (d_1, d_2, ….d_n). The task are considered non-preemptive and have to complete the execution before deadline. We consider at least one task arrives at time t=0. The tasks can start the execution after its arrival and it has to finishes its execution on or before deadline. The system is hard read time system which have hard deadline. The tasks are independent and having its own worst case execution time on maximum speed. Each task consists of three parameters $<a_i,e_i,d_i>$, where a_i is arrival time of task, e_i is worst case execution time at full speed and d_i is deadlines. The uniprocessor is considered for scheduling the tasks. Initially we only consider only one task in the task set at time t=0, and start time of the first task can be considered at 0 andd can be assumed as the time interval in which the task can be executed. The utilization of system can be calculated using u=e/d. and total utilization of system is equal to sum of utilization of each task.

Energy Model: DVS is used for power management. In this voltage is dynamically adjusted for running the task. There are V_{max} maximum voltage level and V_{min} minimum voltage level. A task has to execute between these two voltage levels. The voltage is managed in such a way that the task finishes its execution on or before its deadline. At different supply voltage level V = {V_1, V_2, V_3,…V_n}, where each voltage level is associated with its corresponding speed gives discrete power consumption, there are discrete frequency levels on different supply voltage levels. The energy consumption for each task is depends upon supply voltage and frequency.

The energy consumption of a system can be find by multiplying execution interval and energy consumption per unit time. ifE_c is the energy consumption per unit time and N is the execution interval then total energy consumption E_{total} can be calculated as

$$E_{total}(t_i) = N * E_c(t_i)$$

The energy consumption is calculated for each individual running task. As energy is a constraint, we have to minimize the energy consumption. So check the available slack for the running task. If a task running on maximum voltage level may have some available slack. By applying DVS, we adjust the voltage level such that the task finishes on its deadline and energy consumption can minimize. This energy minimization can be achieved by reducing voltage level and check the schedulability of task. If a task is schedulable with low voltage yields the energy minimization. If a fault tolerance approach is applied in the system then a scheduler has to check the fault and adjust the voltage level accordingly. Schedulers always think there is some fault in the system and work accordingly. A task will be schedulable only if available energy is greater than total consumed energy. If a task does not have sufficient energy to execute a task then the scheduler will hold the execution till sufficient energy available.

5. Proposed Approach

Fault tolerance and energy efficient approach is presented in this paper. A transient fault arrive in a system can be tolerated on run time using checkpointing interval. But it can't used for energy minimization. We propose an energy efficient approach for fault tolerance. We applied the EDF approach for scheduling the task on the scheduler. EDF approach is applied in such a way that most of the tasks have scheduled. We also consider the fault recovery in that system and schedule the task.

The worst case execution time is calculated for each task. Here we consider at least one aperiodic task arrives at t = 0 and the tasks are aperiodic. We check the deadline of all arrived tasks and arrange them with their deadline in ascending order. At time t, we check the schedulability of the minimum deadline task on the system with maximum voltage level. If the task has slack then reduce the voltage level using DVS. Again apply the schedulability of and check the acceptance of the system. If it is finishing on or before deadline the task is schedulable. Otherwise execute the task on same voltage level.

As energy is also the concern then before starting to execute a task at time t, check the availability of energy to execute that task. If it has sufficient energy then executes the task with fault tolerance approach and calculates the drain energy for this task. After finishing the task execution again calculate the available energy for next task by subtracting the drain energy from stored energy and continue the same process.

Proposed Scheduling Algorithm

Task scheduling in proposed algorithm is depends upon deadline of task. The task which has minimum deadline will be scheduled first. The algorithm also considers the energy available. It checks the available energy for execution of task. If a system does not have sufficient energy then scheduler will not schedule the task for execution. If a system has sufficient energy for execution then the scheduling is able to run. We start scheduling by initializing at least one task arrives at time t=0, and if more than one task has arrived then select the task having minimum deadline because the task is consider as higher priority task which has minimum deadline. Apply optimal number of checkpoints for fault tolerance and then check schedulability. If the task is schedulable with fault tolerance, then available slack is also checked. If there is available slack after fault tolerance then apply DVS for reducing energy consumption. Then calculate the energy drain to execute that task. If finish time of that task is less than or equal to its deadline, then the task is schedulable with fault tolerance.

Input: (T_1,T_2,T_3,……T_n), D=deadline, A=Arrival Time
Output: Task Scheduling with fault tolerance and energy minimization

1 For task set T= T_1,T_2,T_3,……T_n
do
2 find minimum deadline task as highest priority task and apply EDF scheduling algorithm
3di ← min{d_i: $T_i$belongs to T} // select task from task set at t=0.
4 Set task T_i has minimum deadline for execution and n optimal number of checkpoint with r = 1 as overhead.

5 WCET = $\frac{e_i}{P_k} + \frac{n_i . r}{P_k} + \left[\frac{e_i}{n_i}\right] + \sum_{j\ belongs\ to Thi} \left[\frac{e_j}{P_k} + \frac{mj.r}{P_k} + \left[\frac{e_j}{m_j}\right]\right]$

// P_k= power at which task executed, decreasing the voltage power will decrease, m and n are number of checkpoint applied on higher priority task and current task respectively. T_{hi} are higher priority tasks.

6 Calculate $E_{Total} = N * E_C$ (N is interval of task execution)
7 If there is slack available for the current task.
8 Apply DVS to minimize energy consumption
9 Print task is schedulable with minimum energy
10 otherwise, task is not schedulable.

6. Results

Table 1 shows the results shows the schedulability of task with fault tolerance with energy consumption. Let us consider the task set of four task <0,2,15>, <6,3,25>, <6,4,25>, and,<7,6,38> with their arrival time, execution time and deadline of each task. We write the code in C++ and check the schedulability of each task without fault. We find the worst case execution time of each task and then check the schedulability. We also check the available energy in the system to execute the task. A task will be schedulable if there is sufficient energy in the system.

Table 1: WCET with fault and without fault in the system.

a	e	d.	n	WCET (No fault)	WCET (With fault)
0	2	15	2	2	5
6	3	25	2	9	20
6	4	20	2	6	13
7	6	38	3	15	31

Results shown in table 1 are in both the cases (with fault is proposed approach and without fault is existing approach). All the tasks are schedulable because the WCET of all the tasks are less then deadline of each task. We apply the checkpoint policy to tolerate fault and the table also shows the optimal checkpoint for each task individually. We also check the energy consumption of each task and check the schedulability. Using our proposed approach we can say our proposed approach is better than any other existing approach. Figure 1 shows the simulation results of both existing and proposed approach. WCET with no fault is existing approach and WCET with fault is proposed approach.

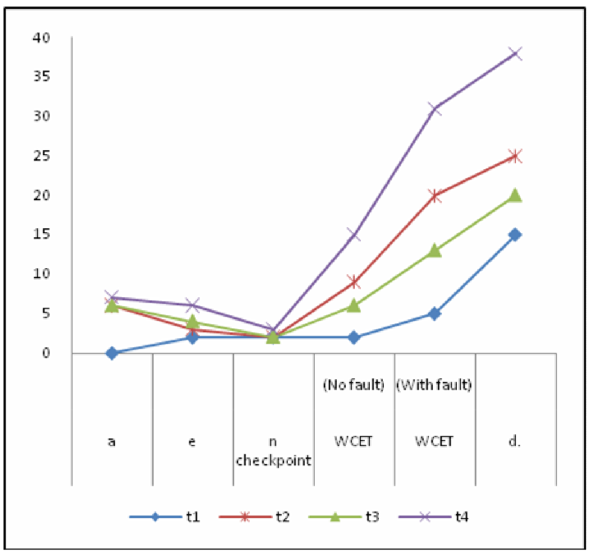

Figure 1: Task scheduling with and without fault

7. Conclusion

The algorithm we have proposed is more efficient to tolerate fault with minimum energy consumption. We check the execution time with fault and without fault. In most of the cases the tasks are scheduled. We presented the formula for finding worst case execution time of individual task. We have also check the energy consumption in each case. The energy consumption is low in our proposed approach. The simulation result shows that all tasks are schedulable in the presence of fault.

References

Huang, K., Santinelli, L., Chen, J., Thiele, L., Buttazzo, G.,“Adaptive dynamicpower management for hard real-time systems”. In: Proceedings of the IEEE Real-Time SystemsSymposium 2009.

Zhang, Y., Chakrabarty, K.,“Energy-aware adaptive checkpointing in embeddedreal-time systems”. In: Proceedings of the DATE 2003.

Melhem, R., Mosse, D., Elnozahy, E., “The interplay of power management andfault recovery in real-time systems’. IEEE Transactions on Computers 53 (2), 2004, 217–231.

D. K. Pradhan, Fault Tolerance Computing: Theory andTechniques. Prentice Hall, 1986.

K. Woonseok, S. Dongkun, Y. Han-Saem, K. Jihong, M. L.Sang, “Performance Comparison of Dynamic Voltage ScalingAlgorithms for Hard Real-Time Systems”, Proceedings of theEighth IEEE Real-Time and Embedded Technology andApplications Symposium (RTAS’02), pp. 219 – 228, 2002.

Yu Liu, Han Liang, Kaijie Wu, "Scheduling for Energy Efficiency and Fault Tolerance in Hard RealTime Systems", 978-3-9810801-6-2/DATE10 © 2010 EDAA, pp.1444-1449.

Dima, Alain Girault, Christophe Lavarenne, Yves Sorel, Off-line real-time fault-tolerant scheduling.In Euromicro Workshop on Parallel and Distributed Processing, Mantova, Italy, February 2001.

Alam, B.; Kumar, A., "A real time scheduling algorithm for tolerating single transient fault," Information Systems and Computer Networks (ISCON), 2014 IEEE International Conference on , vol., no., pp.11,14, 1-2 March 2014

Izosimov, V., Pop, P., Eles, P., Peng, Z., 2008.Scheduling of fault-tolerant embeddedsystems with soft and hard timing constraints. In: Proceedings of the DATE.

Pillai, P., Shin, K., Real-time dynamic voltage scaling for low-power embedded operating systems. In: Proceedings of the ACM Symposium on Operating Systems Principle 2001.

Biographical notes

Arvind Kumar is presently pursuing his PhD (Computer Engineering) from JamiaMilliaIslamia, New Delhi. He received M. Tech. from Motilal Nehru National Institutre of Technology, Allahabad, U.P., India and B. Tech. from UPTU, Lucknow, in Computer Science and Engineering in 2005 and 2011 respectively. He has approx 10 years of experience in teaching and research in various reputed organizations such as AKGEC, Ghaziabad and ABES Institute of Technology, Ghaziabad. He has published more than 16 reseach papers in national/international Journals/book chapters and National/International conferences. Mrarvind’s area of research is Real-Time System, Operating System, Fault Tolerance Computing, Theory of computation, Linux, etc

Dr. Bashir Alam is Assistant Professor in the Department of Computer Engineering, Faculty of Engineering and Technology, JamiaMilliaIslamia, New Delhi, India. He receive his PhD from JamiaMilliaIslamia New Delhi, M.Tech. form IP university New Delhi and B.Tech. form AMU Aligarh. He has more than 15 years of experience in teaching and research. His current area of research includes Real Time System, Algorithm, Microprocessor, Embedded System and Computer Architecture.He has published more than 30 papers in referred international journals and Conferences. He is member of various committees.

Received March 2015
Accepted July 2015
Final acceptance in revised form July 2015