Tolerating Single Transient Fault in Time Critical System using Energy Harvesting

Vineeta Anand , Swasti Sharma, Vijay Verma ,Shivangi Singh
Galgotia's Educational Institution, Greater Noida UP.

Abstract - In this paper, we propose fault tolerance approach to manage the harvesting energy in a time critical system, through dynamic voltage scaling (DVS) while satisfying the system timing constraints. The energy aware DVFS algorithm adjusts the processor's behaviour depending on the stored energy and the harvested energy in the system. Specifically, if the system has sufficient energy, tasks are executed at full speed; otherwise the processor slows down task execution to save energy. We are presenting a checkpointing approach for tolerating single transient fault in time critical applications where source of energy is variable.

1.0 Introduction

Real time critical applications such as space missions, nuclear reactors, air-traffic control systems, etc. highly depend on their correct functioning within its deadline.
Some of real time applications are battery driven having energy as constraint. These systems such as tiny sensor nodes are dependent on battery power. Every task running on real time system have a deadline. It may be soft, hard, or firm. A task running on real time system should finish on or before its deadline. If the task is not finishing on deadline then results may vary which may be useless after its deadline.
A fault in real time system can result a system into failure if not properly detected and recovered at time. These systems must function with high availability even under faults. Fault tolerance enables a system to continue operating properly in the event of the failure of (or faults) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure. In order to solve the energy problem and prolong the system operating duration, energy harvesting is employed. Energy harvesting is regarded as an especially prospective approach to draw parts or all of its operating energy from its ambient energy sources(solar cells etc.).

The performance of a practical energy-harvesting-capable real-time system, measured by the deadline miss rate, heavily depends upon the stored energy and the energy harvested from the environment. Unfortunately, the harvested power is time-varying and thus very unstable. Therefore, the accurate modelling for energy source plays a key role in designing a good policy to schedule the task and reduce the deadline miss rate.
Using checkpoints, total re-executions are replaced with partial rollback and recovery. On the other hand, as one of the most commonly used power management techniques, DVS techniques can be used to save energy.

1.1 Motivation

In the case of hard real-time systems, the stringency of timing requirements, the safety critical features, the fault-tolerance requirements and the need for efficiency are problems which need to be addressed during the design phase of the system. The task of showing that such systems do simultaneously meet all the functional,

ISBN: 9788175157538

timing and fault tolerance requirements, remains an extremely challenging and complicated problem.

1.2 Related Work

Fault tolerance is a typical policy for system reliability in real-time systems. Fault tolerance computing refers to the correct execution of user programs and system software in the presence of transient faults. It is typically achieved through online fault detection, checkpointing, and rollback recovery. Power management can be achieved at various design levels. There are two main types of dynamic power management (DPM) techniques.

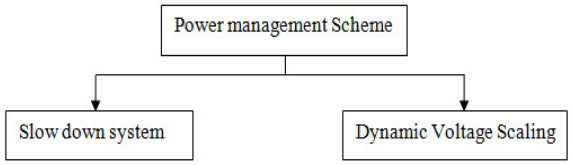


Figure 1: Types of PMS

The first includes selective shut-off or slow down of system components that are idle or underutilized. The second, which is often termed as dynamic voltage and frequency scaling (DVFS), refers to the dynamic control of the supply voltage level for the various components in a system.

DVS tries to address the trade off between performance and battery life by taking into account two important characteristics of most current computer systems:

(1) The peak computing rate needed is much higher than the average throughput that must be sustained.

(2) The processors are based on CMOS logic.

The first characteristic effectively means that high performance is needed only for a small fraction of the time, while for the rest of the time, a low-performance, low-power processor would suffice. We can achieve the low performance by simply lowering the operating frequency of the processor when the full speed is not needed. DVS goes beyond this and scales the operating voltage of the processor along with the frequency. This is possible because static CMOS logic [1], used in the vast majority of microprocessors today, has a voltage-dependent maximum operating frequency, so when used at a reduced frequency, the processor can operate at a lower supply voltage.

1.2.1 Dynamic Voltage Scheduling (DVS)

Dynamic voltage scaling and dynamic frequency scaling (DFS) allow adjusting processor voltage and frequency at runtime. Usually, higher processor voltage and frequency leads to higher system throughput while energy reduction can be obtained using lower voltage and frequency. Dynamic Voltage Scaling is an effective way to minimize CPU energy. The main principle that is exploited in DVS is that the voltage of a processor is proportional to its frequency.

Fault tolerance is achieved via checkpointing, and energy is saved using Dynamic Voltage Scaling(DVS). Issues here are like faults during checkpointing, rollback recovery time, memory access time, and energy needed for checkpointing, as well as DVS and context switching overhead.

1.2.2 Energy Aware EDF Scheduling

The EDF scheduler always decides to execute the task based on earliest deadline. The task with earliest deadline executes first. These tasks may run on different speed based upon the deadline. A collection of independent jobs, each characterized by an arrival time, an execution requirement, and a deadline, can be scheduled such that all the jobs complete by their deadlines, then EDF schedule this collection of jobs such that

ISBN: 9788175157538

they all complete by their deadlines . To save energy, we need to reduce the speed of processor along with voltage supply. In [3], they have proposed an energy-efficient scheduling algorithm (TDVAS) using the dynamic voltage scaling technique to provide significant energy savings for clusters. The TDVAS algorithm aims at judiciously leveraging processor idle times to lower processor voltages (i.e., the dynamic voltage scaling technique or DVS), thereby reducing energy consumption experienced by parallel applications running on clusters.

Reducing processor voltages, however, can inevitably lead to increased execution times of parallel task. The salient feature of the TDVAS algorithm is to tackle this problem by exploiting tasks precedence constraints.

1.2.3 Fault Tolerance with Power Management

Fault tolerance is achieved by checkpointing, and power management is carried out via dynamic voltage and frequency scaling (DVFS).

We consider a fault tolerant schedulability analysis for aperiodic tasks and derive the optimal number of checkpoints presented. The optimal number of checkpoints can help the task to guarantee the timing constraints and minimize the worst case execution time in the presence of faults. A scheduling scheme which carries out DVFS on the basis of the schedulability analysis for the problem of static task scheduling and voltage allocation is proposed. The problem is addressed and formulated as a linear programming (LP) problem. A transient fault is typically achieved through online fault detection, checkpointing, and rollback recovery.

2.0 Proposed Approach

We propose a new checkpointing approach for tolerating single transient fault in real time system. In this work we will calculate optimal number of transient fault. The optimality of checkpoints depends upon deadline of task and speed of processor. We calculate optimal number of checkpoint such that it consumes minimum energy within its deadline. We propose a scheduling algorithm for single transient fault tolerance that offers lesser energy consumption for battery powered real time system modeled with energy harvesting. In the proposed scheme we find the optimal number of checkpoints with respect to energy. Energy is managed dynamically by DVS.

2.1 Task Scheduling using EDF (Earliest Deadline Scheduling)

For task Scheduling we are using Earliest Deadline Scheduling (EDF). The EDF scheduler always decides to execute the task based on earliest deadline. The task with earliest deadline executes first. These tasks may run on different speed based upon the deadline.EDF schedule the collection of jobs (each characterized by their arrival time,execution time and deadline) such that they all complete by their deadlines.

2.2 Dynamic Voltage Scheduling

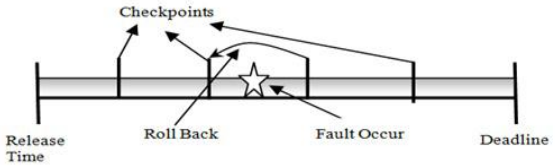
Dynamic voltage scaling and dynamic frequency scaling (DFS) allow adjusting processor voltage and frequency at runtime. The main principle that is exploited in DVS is that the voltage of a processor is proportional to its frequency. When the processor works at lower frequencies than its maximum frequency, it uses less energy. In DVS technique, the processor executes each task at various frequencies in such a way that the task is executed within its

ISBN: 9788175157538

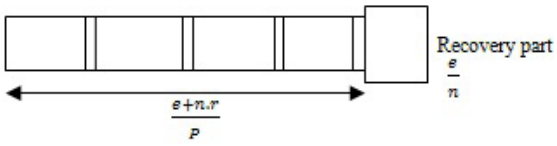
deadline and also the energy utilization is minimized. The power consumption can be represented by:
P proportional to V^2 x f
where v is supply voltage and f is system clock frequency of CMOS circuit.

2.3 Fault Tolerance through Checkpointing

It is the process of saving to rollback the complete execution of a task for tolerating transient faults.



We check the acceptance test on each checkpoint, if it fails then go to previous checkpoint where acceptance test has been passed successfully, instead of starting execution of task from beginning.
We calculate optimal number of checkpoints that can tolerate fault with minimum energy consumption.



Here e is the execution time, n is the number of checkpoints, r is the checkpointing overhead and p is the processor speed. The recovery part e/n utilizes the extra time (slack time) to perform the recovery operation.

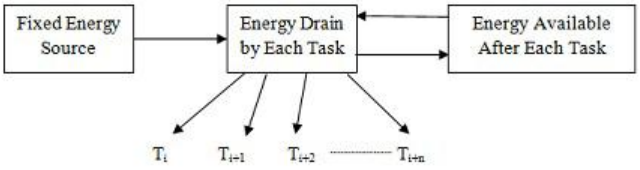
2.4 Energy Harvesting

Let t1 and t2 are the task execution starting time and task finishing time respectively.
After processing each task the available energy can be calculated by equation below

$$E^{i+1}_c = E^i_c - E^i_d$$

This available energy is then used to process next task. Energy requirement of each task is depend upon its execution time of the task and speed of processor.
After processing each task the available energy can be calculated by equation below
At some time *t*, an energy source converts the harvested energy into electrical power P(t) and stores this power in a device that has maximum capacity C.

$$E(t1,t2)=\int_t \quad dt$$



There is always some amount of energy wasted in the process of charging and discharging. So we ignore this loss and assume an ideal battery for our case. There is an upper limit to the storage device denoted by E_{max} which is the maximum capacity of the capacitor. The lower limit of the capacitor is assumed as E_{thres} and not zero which is the energy reserved in the capacitor for worst case scenarios. This is done in accordance with the solar energy predicted by AFP algorithm. An adaptive technique was employed to predict the future availability of solar energy as it tracks small changes in the input signal and dynamically adjust to the changes.
We prefer to predict energy each hour (or 30 minutes) rather than minute as the solar energy does not vary so frequently. The real energy harvested each hour is represented by $x(n-t)$. Here n represents the time of the day in hours and t represents the position of previous inputs from time n where $t = \{0, 1, 2 ...\}$.

3.0 Conclusion

There are various approaches for transient fault tolerance. Many of the real time

ISBN: 9788175157538

applications are fixed battery operated and some are chargeable battery operated.
Tolerating a transient fault in a battery operated critical real time applications with energy minimization is a big issue.
We are presenting a checkpointing approach for tolerating single transient fault in real time critical applications where source of energy is variable.

REFERENCES

[1] S. Kang and Y.Leblebici, CMOS Digital Integrated Circuits Analysis and Design. In McGraw-Hill, 2002.

[2] Y. Zhang and K. Chakrabarty, A Unified Approach for Fault Tolerance and Dynamic Power Management in Fixed- Priority Real-Time Embedded Systems, in IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, Vol. 25(1), pp. 111-125, 2006.

[3] Xiaojun Ruan, Xiao Qin, Ziliang Zong, Kiranmai Bellam, Mais Nijim,An Energy-Efficient Scheduling Algorithm Using Dynamic Voltage Scaling for Parallel Applications on clusters in IEEE International Conference on Computer Communications and Networks (ICCCN), Honolulu, Hawaii, Aug. 2007.

[4] Chaeseok Im and Soonhoi Ha, Dynamic Voltage Scaling for Real-Time Multi-task Scheduling Using Buffers in LCTES04, June 1113, 2004, Washington, DC, USA. ACM 1-58113-806-7/04/0006

[5] Guohui Li, Fangxiao Hu, and Ling Yuan, An Energy-Efficient Fault-Tolerant Scheduling Scheme for Aperiodic Tasks in Embedded Real-Time Systems in IEEE, DOI 10.1109/MUE.2009.69

ISBN: 9788175157538