

Fault Tolerance with Minimum Checkpoint Scheme and Low Energy Consumption Technique

Mr. Anil Vadhwa^[1], Mr. SS Yadav^[2], Ms.Preeti Yadav^[3]

[1]M.Tech (CSE) Student, RPS College of Engineering and Technology, Balana Mohindergarh

[2] Associate Professor, RPS College of Engineering and Technology, Balana Mohindergarh

[3] Assistant Professor, RPS College of Engineering and Technology, Balana Mohindergarh

anilvadhwa018@gmail.com, ssyadav78@gmail.com, preetyadav27@gmail.com

Abstract: Fault tolerance is the property that enables a system (often computer-based) to continue operating properly in the event of the failure of (or one or more faults within) some of its components. Fault tolerance approach is used to give the expected services in the presence of fault. Check pointing is the fault tolerance approach for transient faults used for rollback recovery where we detect the Fault by applying acceptance test and then roll back to the previous checkpoint if a fault occurs.

The optimality of checkpoints is dependent upon energy and deadline of the task. The energy minimization is achieved by DVS (Dynamic Voltage Scaling) techniques and we apply for both static and dynamic voltage

Keywords: Checkpoint, Fault, DVS Techniques for both static and dynamic voltage.

Introduction

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. A checkpoint may be system level, application level, or mixed level depends on its characteristics

A lot of research efforts have been made to solve the energy problem of a battery-based system, but it continues to be a chief challenge for researchers. This is particularly true for systems where replacing or recharging batteries manually is impracticable. One example is locating sensors in remote areas for monitoring/sensing purposes. This is important as we can't reach these remote places all the time to replace the battery after it is discharged. The energy constraint has become a main obstacle for increasing lifespan of battery-based systems. In order to solve the energy problem and prolong the system operating duration, a new technology called energy harvesting, also known as *energy scavenging*, has recently been explored. Energy harvesting is regarded as an especially prospective approach to decouple the energy constraint of battery-powered systems and increase system autonomy. Simply put, energy scavenging system is a system that draws parts or all of its operating energy from its ambient energy sources.

Issues

There are four main issues such as: **Feasibility**- this means that a task running should be finished on its deadline even though there is a fault in the system. Dead line in real time system is the major issue because there is no meaning of such a task which is not finishing before its deadline. So the question is that which method is to be applied by which the task can finish on deadline in the presence of fault. **Reliability**- in real time distributed system reliability means availability of end to end services and the ability to experience failures or systematic attacks, without impacting operations. **Scalability**- it is about the ability to handle growing amount of work, and the capability of a system to increase total throughput under an increased load when resources are added. **Energy**- to operate any system energy is the main requirement. The system is more reliable if it is consuming lesser energy and give desired output even there is fault in a system.

System Requirements

HARDWARE REQUIREMENTS:

1. Memory
2. Hard Disk

SOFTWARE REQUIREMENTS:

1. Operating system: Windows XP/7/8
2. Language: C

ENERGY HARVESTING ALGORITHM

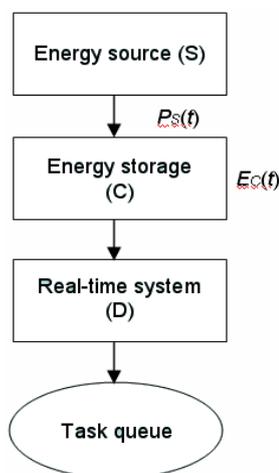
[Initialize array A and E for generating energy profile]

1. Divide day(24 hrs) into k equal slots
2. $\forall i, i=0,1,2,..(k-1)$

Calculate

$$E[i] = \alpha \cdot E[i] + (1 - \alpha) \cdot A [i-1]$$

3. Exit



A real-time system with energy harvesting module

Proposed Model

PROPOSED CHECKPOINTING TECHNIQUE

In this work we will calculate optimal number of checkpoints.

TASK SCHEDULING ALGORITHM

[Initialize ready queue Q of tasks T_m]

1. While(true)
 - do
 - At $t = 0$
 - $d_1 \leftarrow \min\{d_m : T_m \in Q\}$
 - 2. Calculate checkpoints and energy consumption for T_i
 - 3. Find Finish time F_i of task T_i
 - 4. At $t=F_i$
 - $d_1 \leftarrow \min\{d_m : T_m \in Q \ \&\& \ a_m \leq F_i\}$
 - If ($d_{i+1} \geq F_i$)
 - Calculate checkpoints and energy consumption for T_i
 - Else
 - Remove task from Queue Q
 - 5. Go to step 3
 - 6. Exit

CHECKPOINTING WITH STATIC VOLTAGE

[Initialize $n=1, r=1, V=$ Maximum Voltage]

$\forall T_i, i=0,1,2,..(n-1)$

1. Calculate

$$F_n = e_i + n_i \cdot r + \text{ceil}(e_i / n_i) + \sum_{j \in T_h} [e_j + n_j \cdot r + \text{ceil}(e_j / n_j)]$$

Compare F_n, F_{n-1}

If ($F_n \leq F_{n-1}$)

Do $n = n + 1$

Go to step 1

Else

If ($F_n \leq d_i$ && $E_d^i \leq E_c^i$)
 Print n-1, optimal number of checkpoint
 Else
 Print, task cannot be scheduled
 2. Exit

CHECKPOINTING WITH DYNAMIC VOLTAGE

[Initialize n=1,r=1,V=Minimum Voltage]
 $\forall T_i, i=0,1,2,..(n-1)$
 1. Calculate
 $F_n = e_i / p_k + n_i \cdot r / p_k + \text{ceil}(e_i / n_i) + \sum_{j \in T_n} [e_j / p_k + n_j \cdot r / p_k + \text{ceil}(e_j / n_j)]$
 2. Compare F_n, F_{n-1}
 If ($F_n \leq F_{n-1}$)
 Do $n = n + 1$
 Go to step 1
 Else
 If ($F_n \leq d_i$ && $E_d^i \leq E_c^i$)
 Print n-1, optimal number of checkpoint
 Else
 Increase voltage level V , upto maximum; where $V=\{V_1, V_2, V_3,.. V_m\}$
 Else
 Print, task cannot be scheduled
 3. Exit

ENERGY CONSUMPTION ALGORITHM

[Initialize E to total capacity of battery]
 1. Find E_c^i , energy available for task T_i
 2. Calculate $E_d^i = \int_{t_1}^{t_2} E_{dput}^i, T_i \in Q$
 3. $E_c^{i+1} = E_c^i - E_d^i$
 4. Exit

RESULTS

The performance of algorithm proposed has been measured through various results generated by implementing in C++. All the results are calculated on both static and dynamic voltage. In static voltage, voltage $V = 5V$ as the maximum voltage. In dynamic voltage, we are assuming three different voltage level 3V, 4V and 5V with frequencies 0.5Hz, 0.75Hz and 1Hz respectively. We also consider a set of task with attributes $\langle a_i; e_i; d_i \rangle$ generated by random number generator. The arrival time of each task is generated by adding a random number(0-4) in arrival time of previous task. Execution time of a task is also generated by uniform random number generator from 2 - 6 and the deadline of each task is calculated by $a_i + e_i + \{5 \text{ to } 8\} \cdot e_i$. Every task has its individual number of checkpoint. The system energy initially is 250J. The energy being harvested will be added to the system energy. The energy consumed by a set of task at static and dynamic voltage is shown in following table:

| a_i | e_i | d_i | V | n | EWC | TEWC | E_c | E_d | Ec(new) |
|-------|-------|-------|---|---|-----|------|--------|-------|---------|
| 0 | 3 | 24 | 5 | 3 | 7 | 7 | 250 | 25 | 323.19 |
| 4 | 2 | 16 | 5 | 2 | 5 | 12 | 323.19 | 25 | 388.40 |
| 2 | 2 | 20 | 5 | 2 | 5 | 17 | 388.40 | 25 | 489 |

Checkpoint and Energy Consumption at Static Voltage V=5

| a_i | e_i | d_i | V | N | EWC | TEWC | E_c | E_d | Ec(new) |
|-------|-------|-------|---|---|-----|------|--------|-------|---------|
| 0 | 3 | 24 | 3 | 1 | 8 | 8 | 250 | 36 | 331.23 |
| 4 | 2 | 16 | 3 | 2 | 6 | 13 | 331.23 | 27 | 420.97 |
| 2 | 2 | 16 | 3 | 2 | 6 | 19 | 420.97 | 27 | 494.32 |

Checkpoint and Energy Consumption at Dynamic Voltage

Conclusion/Future Work

These systems are fixed battery operated or battery operated. Success of such systems depends on various constraints including energy requirements, fault tolerance and deadline. In this project we present a check pointing approach for tolerating fault in real time critical applications where source of energy is variable i.e. system with energy harvesting. Our goal is to find the best placement of checkpoints to avoid of missing the task's deadline when the occurrence of one transient fault is probable, while efficiently using the harvested energy. We find optimal number of checkpoints in order to minimize the energy consumption. For predicting the energy being harvested, we are using EWMA that estimates the amount of energy harvested in a day based on the previous. Further, we implement an energy aware dynamic voltage and frequency selection algorithm.

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
- [6] C. Rusu, R. G. Melhen, and D. Mosse, "Multi-version scheduling in rechargeable energy-aware real-time systems", In 15th Euromicro Conference on Real-time systems, ECRTS 2003, pp95-104, Berlin, Germany, Jan., 2004.
- [7] A. Kansal, J. Hsu, "Harvesting aware power management for sensor networks", In IEEE DAC 2006.