

논문 2006-08-04

신뢰도를 요구하는 임베디드 시스템에서의 저전력 태스크 스케줄링

(Power-aware Real-time Task Scheduling in Dependable Embedded Systems)

김경훈, 김유나, 김종*

(Kyong Hoon Kim, Yuna Kim, Jong Kim)

Abstract : In this paper, we provide an adaptive power-aware checkpointing scheme for fixed priority-based DVS scheduling in dependable real-time systems. In the provided scheme, we analyze the minimum number of tolerable faults of a task and the optimal checkpointing interval in order to meet the deadline and guarantee its specified reliability. The energy-efficient voltage level at a fault arrival is also analyzed and used in the recovery of the faulty task.

Keywords : 저전력, 실시간, 신뢰도 요구사항, 임베디드 시스템, 체크포인트링

I. 서 론

임베디드 시스템의 성능이 계속 발전하여 다양한 실시간 응용 프로그램도 수행할 수 있게 되었다. 임베디드 시스템의 단점인 배터리 한계를 극복하기 위해서 저전력 실시간 시스템에 관한 많은 연구가 진행되고 있다. 대표적인 연구 방법인 동적 전압 조절 (DVS: Dynamic Voltage Scaling) 방법은 주어진 태스크의 마감시간을 만족하면서 전력 소비를 최소로 하도록 전압을 동적으로 조절한다 [1, 2].

임베디드 시스템의 또 다른 문제점으로 저전력과 이동성에 의한 신뢰도 손실이 있다. 이러한 신뢰도 손실을 보강하기 위해서 결함포용 기법을 사용하여 좀 더 안전한 시스템을 제공한다. 실시간 태스크들은 그 중요도에 따라서 서로 다른 신뢰도를 요구한다. 예를 들어, 금융 관련 실시간 응용 프로그램의 경우 그 결과가 금전적으로 중요하므로 높은 신뢰도를 요구한다. 반면에 엔터테인먼트용

멀티미디어 프로그램에서 일시적인 손실은 허용하므로 상대적으로 낮은 신뢰도를 요구하게 된다. 본 연구는 이렇게 태스크별로 서로 다른 신뢰도를 요구하는 태스크 집합이 주어졌을 때, 이를 만족하면서 전력의 손실을 최소로 하는 스케줄링 방법에 관해서 연구한다.

최근 실시간 태스크들의 신뢰도를 고려한 저전력 체크포인트링 관련 연구가 진행되어 왔다 [3-6]. Melhem 등의 연구에서는 주어진 결함포용 수에 대해서 저전력 결함포용 기법에 대해서 연구하였다 [3]. Zhu는 신뢰도를 고려한 저전력 체크포인트링 기법을 제안하였으며 [5], 최대 전압에서 결함이 발생하지 않고 제대로 수행할 확률을 신뢰도로 정의하였다. 본 연구는 고정 우선 순위 기반의 스케줄링 알고리즘을 사용하면서, 태스크들의 주어진 신뢰도 요구 사항을 만족하는 적응적 저전력 체크포인트링 기법을 제안한다.

II. 시스템 모델

1. 에너지 모델

CMOS 회로의 주요 전력 손실은 동적 전력 손실과 정적 전력 손실로 나눌 수 있는데, 더 중요한 동적 전력 손실만 고려한다. 한 태스크가 소비한 동적 에너지는 인가 전압 V_{dd} 의 제곱과 수행 클럭 사이클 수 M_{cyc} 에 비례한다. 또 다른 에너지 모델로 프로세서의 속도와 태스크 실행 시간을 사용하

* 교신저자(Corresponding Author)

논문접수 : 2008. 03. 11., 채택확정 : 2008. 04. 01.
김경훈 : 경상대학교 정보과학과

김유나, 김종 : 포항공과대학교 컴퓨터공학과

* "본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (ITA-2008-C1090-0801-0045)

는 것이다. 프로세서 속도는 전압에 비례하고 태스크 실행 시간은 클럭 사이클 수에 비례하므로, 본 연구에서의 태스크 에너지 소비는 다음과 같이 가정한다 [3]. 단, S 는 프로세서의 상대적인 속도이며 ($S_{\min} \leq S \leq S_{\max} = 1$) t 는 태스크의 실행 시간, a 는 비례 상수이다,

$$E = \alpha S^2 t \quad (1)$$

본 연구에서는 동적 전압 조절이 가능한 임베디드 프로세서를 대상으로 하여 [1, 2, 7], m 개의 다른 전압 조절이 가능하다고 가정한다. 따라서 V_1 부터 V_m 까지 전압 조절이 가능하고, 각 전압에서의 프로세서 상대 속도를 S_i 로 나타낸다. S_{\min} 을 프로세서 최고 속도로 정의하고, 이 값을 기준으로 각 전압에서의 상대 속도로 표시한다.

2. 태스크 모델

태스크 집합 T 는 M 개의 태스크로 구성되며 ($T = \{\tau_1, \tau_2, \dots, \tau_M\}$) 각 태스크 τ_i 는 다섯 개의 파라미터 (P_i, E_i, D_i, C_i, R_i)로 정의가 된다.

- P_i : 작업을 생성하는 주기. 각 태스크는 매 P_i 시각에 하나의 작업을 생성한다.
- E_i : 무결함시 작업의 실행 시간. 생성된 작업의 실행 시간은 최대 성능의 (S_{\min}) 프로세서에서 실행되었을 때, 결함이 없는 경우 E_i 시간이 걸린다. 이때 작업 실행에 필요한 최악 실행 시간 (worst-case execution time)을 사용한다.
- D_i : 작업의 상대 마감 시각. 생성된 작업은 생성 후 D_i 시간 이내에 실행을 마쳐야 한다.
- C_i : 체크포인트링 시간. 시스템은 주기적으로 작업의 상태를 저장하는데, 이때 최대 성능의 프로세서에서의 걸리는 시간을 말한다.
- R_i : 신뢰도 요구 사항. 태스크의 한 작업이 주어진 마감 시각 내에 수행을 마칠 확률을 신뢰도로 정의한다.

시스템은 M 개의 태스크로 이루어지며, 이 태스크들은 RM(Rate Monotonic)과 같은 고정 우선순위 기반 스케줄링 알고리즘으로 스케줄링 된다. τ_i 는 τ_{i+1} 보다 높은 우선순위를 가진다.

각 전압 수준에 따라서 태스크의 실행 시간과 체크포인트링 시간이 다르게 정해진다. E_i 와 C_i 가 최대 성능($S_{\min} = 1$)에서 정의가 되었으므로, 전압 V_i 에서의 실행 시간과 체크포인트링 시간은 각각 E_i/S_i 와 C_i/S_i 로 정의된다.

3. 결함 및 복구 모델

본 연구에서는 시스템에 발생하는 일시 결함(Transient Fault)을 대상으로 하며, 이러한 결함이

발생했을 때 수행 중이던 태스크에만 영향을 미친다. 그리고 그 결함은 재실행함으로써 복구할 수 있다. 결함 발생 시 재실행 시간을 줄이기 위해서 체크포인트링 (Checkpointing) 기법을 사용하여, 주기적으로 작업의 상태를 안전한 저장 장치에 저장한다. 따라서 결함이 발생하면 가장 최근의 체크포인트된 상태부터 재실행 한다.

일반적으로 결함 발생 모델은 포아송 (Poisson) 분포를 따르며 [7], 발생률이 λ 인 경우에 시간 t 동안 k 번의 결함이 발생할 확률은 $(\lambda t)^k e^{-\lambda t} / k!$ 이 된다. DVS 기반의 프로세서에서 결함 발생률은 인가 전압에 의존한다 [5]. 각 전압 V_i 에서의 결함 발생률을 λ_i 로 정의하며, 일반적으로 저전압 프로세서에서 타이밍 결함 발생률이 높으므로 $\lambda_i \geq \lambda_{i+1}$ 이라고 말할 수 있다.

III. 체크 포인트링을 고려한 적응적 저전력 태스크 스케줄링

1. 동적 저전력 결함포용 스케줄링 기법

체크포인트링 기반의 결함 포용 시스템에서 동적 전압 관리 기법을 적용하여 소모 전력을 줄인다. 그림 1(a)처럼 고전압 인가시 태스크 수행 시간이 짧고 결함 발생 확률이 낮지만, 소비하는 에너지가 많아진다. 따라서 마감 시각을 맞출 수 있는 한 최대한 낮은 전압에서 실행함으로써 소비 에너지를 줄일 수 있다 (그림 1(b)).

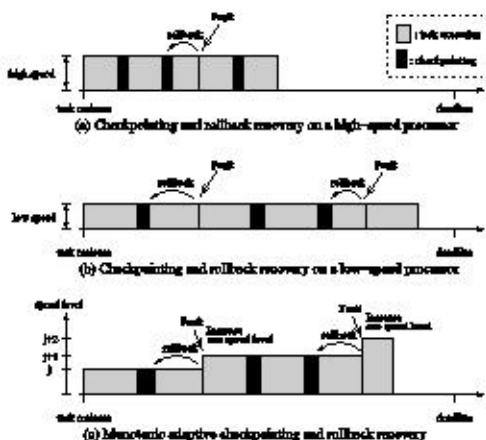


그림 1. DVS 체크포인트링 기법

Fig 1. DVS checkpointing schemes

단계적 동적 저전력 체크포인팅 기법[6]은 태스크의 작업 수행 도중 결함이 발생하면 그 태스크의 신뢰도를 높이기 위하여 인가 전압의 수준을 한 단계 상승시켜 준다. 즉, 현재 전압이 V_j 인데, 만약 결함이 발생하면 한 단계 올려서 V_{j+1} 에서 수행하도록 한다 (그림 1(c)).

2. 분석 및 스케줄링 가능성 검사

2.1 신뢰도 보장을 위한 포용 결함 수 분석

각 태스크의 신뢰도를 보장하기 위해서 포용할 수 있는 결함 수를 분석한다. 만약 태스크의 작업이 수행하는 동안 발생하는 결함의 수가 이 값을 넘지 않으면 그 태스크의 신뢰도를 보장하게 된다.

먼저 결함 발생률이 λ 일 때 시간 t 동안 최대 k 번의 결함이 발생할 확률을 $PF(\lambda, t, k)$ 라고 정의한다. 이 값은 결함 발생 사건이 포아송 분포를 따르므로, 다음과 같이 정의 된다.

$$PF(\lambda, t, k) = \sum_{i=0}^k \frac{(\lambda t)^i e^{-\lambda t}}{i!} \quad (2)$$

만약 태스크가 k 번의 결함을 포용할 수 있다고 할 때, 그 태스크의 신뢰도를 $PF(\lambda, t, k)$ 라고 말할 수 있다. 따라서 주어진 신뢰도 r 를 만족하는 결함 수 k 는 $PF(\lambda, t, k) \geq r$ 를 만족하는 최소의 정수가 된다. 각 전압 V_j 에서 태스크 τ_i 의 신뢰도를 만족하기 위한 포용 결함 수를 $K_{i,j}$ 라고 표시한다. V_j 에서의 결함 발생률은 λ_j 이며 태스크 τ_i 의 실행 시간은 E_i/S_j 이므로, $K_{i,j}$ 는 $PF(\lambda_j, E_i/S_j, k) \geq r_i$ 인 최소 정수이다.

2.2 최적의 체크포인팅 회수

너무 많은 체크포인팅은 결함 발생 시 재실행 시간은 줄어들지만, 체크포인팅에 드는 비용이 증가하게 된다. 반면에 너무 작은 체크포인팅은 부가적인 비용은 적지만, 결함 발생 시 재실행 시간이 증가하게 된다. 따라서 성능의 향상을 위해서 최적의 체크포인팅 회수 또는 체크포인팅 주기를 분석한다.

만약 태스크 τ_i 의 한 작업이 전압 V_j 에서 실행되는 동안 $(n-1)$ 번의 체크포인팅을 수행한다고 가정하자. 이 경우 $K_{i,j}$ 번의 결함이 발생할 경우 최악의 실행 시간 $W_{i,j}^*(n)$ 은 아래 수식과 같이 정의된다.

$$W_{i,j}^*(n) = \max_{t=1}^{K_{i,j}} \left\{ \left(\frac{E_i}{nS_j} + \frac{Q_j}{S_j} \right) (n-k) + \sum_{m=1}^k \left(\frac{E_i}{nS_{j+m}} + \frac{Q_j}{S_{j+m}} \right) + \sum_{m=b}^{t-1} \frac{E_i}{S_{j+m}} - \frac{Q_j}{S_{j+t}} \right\} \quad (3)$$

최악의 시간을 고려하기 때문에 $W_{i,j}^*(n)$ 대신에 아래 수식 $W_{i,j}^o(n)$ 을 근사값으로 사용한다.

$$W_{i,j}^o(n) = \frac{E_i}{S_j} + (n-1) \frac{Q_j}{S_j} + K_{i,j} \frac{E_i}{nS_j} \quad (4)$$

프로세서 전압 V_j 에서의 최적의 체크포인팅 회수 $m_{i,j}$ 는 $W_{i,j}^o(n)$ 을 최소로 하는 m 으로, 그 값은 $\sqrt{K_{i,j}E_i/Q_j}$ 이다. $m_{i,j}$ 는 정수이므로, 최적의 체크포인팅 수는 $\lfloor \sqrt{K_{i,j}E_i/Q_j} \rfloor$ 또는 $\lceil \sqrt{K_{i,j}E_i/Q_j} \rceil$ 이 된다.

```

Algorithm Feasibility Test with DVS ( $T$ )
 $\mu = T = \{ \tau_i (P_i, E_i, D_i, C_i, R_i) \mid i = 1 \dots N \}$ 
1: for  $i$  from 1 to  $N$  do
2:    $\sigma_i = 0; \omega_i = 0;$ 
3:    $\varepsilon_i = \text{LARGE\_VALUE};$ 
4:   for  $j$  from 1 to  $m$  do
5:      $n_{i,j} =$  the optimal checkpointing number
6:      $s_{i,j} = \alpha_j^2 W_{i,j}^o(n_{i,j})$ 
7:     if  $(\exists t \leq D_i$  such that  $RT_{i,j}(t) \leq t)$  then
8:       if  $(\varepsilon_i < s_{i,j})$  then
9:          $\sigma_i = j$ 
10:         $s_i = s_{i,j}$ 
11:         $\omega_i = W_{i,j}^o(n_{i,j})$ 
12:      endif
13:    endif
14:  endif
15:  if  $(\sigma_i == 0)$  then return Not Feasible
16: endif
17: return Feasible
    
```

그림 2 스케줄링 테스트 및 전압 설정 알고리즘
Fig 2. Algorithm for testing the feasibility with DVS

2.3 태스크의 전압 설정 및 실행가능성 분석

고정 우선순위 기반의 스케줄링 알고리즘을 사용하므로, 우선순위에별 태스크 스케줄링 테스트와 전압 설정을 동시에 하는 알고리즘은 그림 2와 같다.

그림 2에서 태스크 τ_i 의 전압 설정을 σ_i 로 표시하며, $\varepsilon_{i,j}$ 는 전압 V_j 에서의 예상 소비 전력이다. 7 번째 줄의 스케줄링 가능성 테스트는 최악의 경우에 대한 응답 시간 분석을 통해서 알 수 있다. 아래 수식에서 $W_{i,j}^o(m_{i,j})$ 는 태스크 τ_i 가 전압 V_j 에서의 최악의 실행 시간이며, 그 보다 우선순위가 높은 태스크들의 실행 시간을 합한 것이 응답 시간 $RT_{i,j}(t)$ 가 된다.

$$RT_{i,j}(t) = W_{i,j}^o(m_{i,j}) + \sum_{h=1}^{i-1} \left\lceil \frac{t}{P_h} \right\rceil \cdot W_{h,\sigma_h}(n_{h,\sigma_h}) \quad (5)$$

3. 결함 복구 시 적응적 전압 설정

단계적 저전력 결함포용 기법에서 실행 도중 결함이 발생하면 전압 수준을 항상 한 단계씩 증가시킨다. 하지만 결함 발생 시점에서 남은 실행 시간을 고려해서 최적의 전압 수준을 설정함으로써 보다 전력 소비를 줄일 수 있다.

그림 3은 결함 발생 시 인가 전압을 재설정하는 알고리즘을 보여준다. 태스크 τ_i 에게 보장된 최악의 실행 시간은 그림 2에서 w_i 로 정해져 있다. 따라서 이 값을 넘지 않는 범위에서 전력 소비를 최소화하는 것을 택하게 된다. 그림 3의 1번째, 5번째 줄에서 최적 체크포인트링 수를 정할 때, 현재 남은 시간은 E_i 를 사용한다. 7번째 줄에서처럼 최악의 실행 시간이 w_i 를 넘지 않으면 태스크의 신뢰도를 보장하게 된다.

```

Algorithm DVS_Task_At_Recovery ( $\tau_i$ )
/*  $E_i$ : the remaining execution time */
1:  $n_{m,i}$  = the checkpointing number with  $E_i$  at level  $m$ 
2:  $E_i = S_m = \alpha_i S^m W_i(\tau_i)$ 
3:  $s_i = m$ ;
4: for  $i$  from 1 to  $m-1$  do
5:    $n_{i,j}$  = the checkpointing number with  $E_i$  at level  $j$ 
6:    $S_j = \alpha_i S^j W_i(\tau_i)$ 
7:   if ( $W_i(\tau_i) + E_i - E_i \leq w_i$  and  $S_j < E_i$ ) then
8:      $s_i = j$ 
9:      $S_i = S_j$ 
10:  endif
11: endfor
12: Set the current supply voltage with  $s_i$ .

```

그림 3. 복구 시 태스크 전압 설정

Fig. 3. Algorithm for setting the supply voltage at recovery

IV. 성능 평가

본 연구에서 제시한 방법의 성능을 모의실험을 통하여 비교한다. 모의실험에서는 6개의 전압을 가지는 프로세서를 가정하며, ($S = \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$) 결함 발생률도 0.001에서 0.050 사이에 서로 다른 값으로 설정한다. 태스크의 실행 시간 E_i 는 5에서 20 사이의 임의의 값으로 생성하며, 주기 P_i 는 50에서 200 사이에서 선택한다. 신뢰도 R_i 는 0.8보다 큰 값으로 생성하며, 체크포인트링 비용 C_i 는 0.1에서 0.9 사이에서 선택한다.

그림 4는 모의실험 결과를 보여 주며, 생성한 태스크 집합의 이용률을 0.3에서 0.7로 바꿔가면서 실험을 하였다. 그림 4에서 Static_Check는 최대 성능에서의 실행 결과이며, Adaptive_Check는 제안하는 방식이다. 결과 수치는 상대적인 에너지 소비를 나타낸 것으로 제안하는 방법이 정적인 방법보다 24% ~ 33%만큼의 에너지 소비를 줄여 준다.

V. 결론

본 연구는 고정 우선순위 기반의 임베디드 시스템에서의 저전력 결함포용 스케줄링 방법을 제안하였다. 제안하는 방법은 태스크의 신뢰도 요구 사항

을 만족하기 위한 허용 결함 수를 분석하고, 최적의 체크포인트링 최수를 유도한다. 태스크들의 스케줄링 가능성을 검사하고 에너지 소비를 줄이는 알고리즘을 제안하였다. 또한 결함 발생 시에 신뢰도 보장 한도 내에서의 최적 전압 설정 방법을 제안하였다. 모의실험을 통해서 소비 에너지의 양이 줄어들음을 보였다. 향후 제안하는 방법을 기반으로 저전력 결함포용 임베디드 시스템을 디자인하고 개발한다.

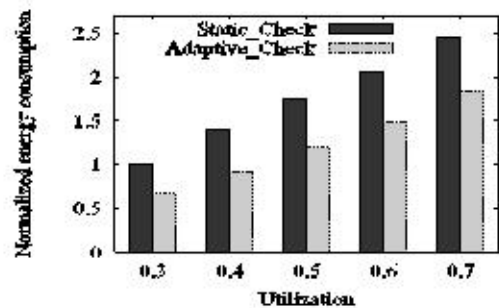


그림 4. 모의실험 결과

Fig. 4. Simulation result

참고문헌

- [1] Y. Lee and C.M. Krishna, "Voltage-clock scaling for low energy consumption in real-time embedded systems," Proc. of the 6th Intl. Conf. on Real-Time Computing Systems and applications, pp. 272-279, 1999.
- [2] P. Pillei and K. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," Proc. of 18th ACM Symp. on Operating System Principles, pp. 89-102, 2000.
- [3] R. Melhem, D. Mosse, and E.N. Elnozahy, "The interplay of power management and fault recovery in real-time systems," IEEE Trans. on Computers, Vol. 53, No. 2, pp. 217-231, 2004.
- [4] Y. Zhang and K. Chakrabarty, "A unified approach for fault tolerance and dynamic power management in fixed-priority real-time embedded systems," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 25, No. 1, pp. 111-125, 2006.
- [5] D. Zhu, "Reliability-aware dynamic energy

management in dependable embedded real-time systems," Proc. of 21th IEEE Real-Time and Embedded Technologies and Applications Symposium, April 2006.

- [6] K.H. Kim and J. Kim, "An adaptive DVS checkpointing scheme for fixed-priority tasks with reliability constraints in dependable real-time embedded systems," Proc. of Intl Conf. on Embedded Software and System (ICCESS-07), Daegu, Korea, May 14-16, 2007.
- [7] Y. Zhang and K. Chakrabarty, "Energy-aware adaptive checkpointing in embedded real-time systems," In Proc. of IEEE/ACM Design, Automation and Test in Europe Conference (DATE), 2003.

저 자 소 개

김 경 훈 (Kyong Hoon Kim)



1998년 포항공과대학교 전자계산학과 학사. 2000년 포항공과대학교 컴퓨터공학과 석사. 2005년 포항공과대학교 컴퓨터공학과 박사. 현재, 경상대학교 정보과학과 교수.

관심분야: 임베디드 소프트웨어, 실시간 스케줄링

Email: khkim@gnu.ac.kr

김 유 나 (Yuna Kim)



2001년 포항공과대학교 컴퓨터공학과 학사. 2003년 포항공과대학교 컴퓨터공학과 석사. 현재, 포항공과대학교 컴퓨터공학과 박사과정 중.

관심분야: 서비스 지향 컴퓨팅, 정보 및 시스템 보안, 서비스 통합 기술

Email: existion@postech.ac.kr

김 종 (Jong Kim)



1981년 한양대학교 전자공학과 학사. 1983년 한국과학기술원 컴퓨터공학과 석사. 1991년 Pennsylvania State University 박사. 현재, 포항공과대학교 컴퓨터공학

과 교수.

관심분야: 병렬 처리 및 고성능 컴퓨팅, 임베디드 시스템 소프트웨어, 정보 및 시스템 보안

Email: jkim@postech.ac.kr