하이퍼주기 압축을 이용한 주기적 실시간 시스템의 확률적 분석 방법의 개선

김강희[O]
서울대학교 컴퓨터공학부
khkim@archi.snu.ac.kr[O]

Enhancing the Stochastic Analysis of Periodic Real-Time Systems Using the Hyperperiod Compression

Kanghee Kim[0]
School of Computer Science and Engineering, Seoul National University

요 약

본 논문은 우선순위 스케줄링을 사용하는 주기적 실시간 시스템을 위해 제안된 확률적 분석 방법의 개선 방안을 제안한다. 기제안된 확률적 분석 방법은 (1) 시스템 적체 분석, (2) 작업별 적체 분석, (3) 작업별 간섭 분석을 거쳐서 각 태스크의 응답 시간 분포를 얻어냄으로써 태스크들의 종료시한 위반확률(deadline miss probability)을 계산한다. 본 논문은 기제안된 확률적 분석 방법에 대해 주어진 태스크 집합의 하이퍼주기(hyperperiod)를 압축하여 시스템 적체 분석을 개선시킬 수 있음을 보인다. 하이퍼주기 압축 기법은 시스템 적체 분석에 요구되는 계산 시간을 단축시키고, 계산된 각 태스크의 종료시한 위반확률이 항상 원래 확률값의 상한(upper bound)이 된다는 장점이 있다. 실험 결과, 하이퍼주기 압축 기법이 정확도의 손실을 최소화하면서 상당한 계산 시간 단축을 가져올 수 있음을 확인하였다.

1. 서 론

최근에 실시간 시스템 분야에서는 종료시한에 관한 확률적 보장(probabilistic guarantee)을 제공하기 위한 연구들이 활발히 진행되고 있다[1,2,3,4]. 확률적 보장이란 시스템 내에 있는 각 태스크가 주어진 확률로 종료시한 내에 실행을 마칠 수 있음을 보장하는 것을 의미한다. 이러한 확률적 보장의 필요성은 연성 실시간 시스템(soft real-time system)에서부터 하드웨어 고장율 수준에 가까운 종료시한 위반확률을 허용하는 경성 실시간 시스템(hard real-time system)[3]에까지 폭넓게 나타나고 있다. 이러한 확률적 보장을 제공하기 위해서 해당 시스템에서 각 태스크의 종료시한 위반확률을 계산해낼 수 있는 확률적 분석 방법이 필요하다.

우선순위 스케줄링을 사용하는 주기적 실시간 시스템에 대해 본 연구진은 정확한 확률적 분석 방법을 [4]에서 제안하였다. 제안된 확률적 분석 방법은 Rate Monotonic[5]과 같은 고정 우선순위 스케줄링 및 Earliest Deadline First[5]와 같은 동적 우선순위 스케줄링을 사용하는 시스템에 모두 적용가능하며, 마르코프 확률과정(Markov Process) 모델링을 통하여 각 태스크의 극한 응답 시간 분포를 정확히 계산해낼 수 있다. 제안된 분석 방법은 (1) 시스템 적체 분석, (2) 작업별 적체 분석, (3) 작업별 간섭 분석 등의 3단계로 이루어지는데, 이 단계들 중에서 시스템 적체 분석이 일반적으로 가장 높은 복잡도를 갖고 있음이 밝혀졌다[6].

본 논문에서는 시스템 적체 분석을 단순화하기 위해 하이퍼주기 압축(hyperperiod compression) 기법을 제안한다. 하이퍼주기란 각 태스크들의 주기들의 최소공배수로 정의되는 주기로서 시스템 적체 분석 단계에서 행해지는 마르코프 행렬 유도 및 풀이 과정에 큰 영향을 미치는 요소이다. 말하자면 하이퍼주기가 크면 클수록, 시스템 적체 분석에 요구되는 시간은 크게 증가한다. 따라서 본 논문은 하이퍼주기 압축 기법을 통해서 시스템 적체 분석 시간을 단축할 수 있음을 보이고, 그 결과 얻어지는 분석 결과를 나머지 단계들, 즉 작업별 적체 분석과 작업별 간섭 분석에 사용해도 안전(safe)하다는 것을 증명한다. 여기서 안전하다는 의미는 계산된 각 태스크의 종료시한 위반확률이 항상 하이퍼주기 압축을 수행하지 않은 경우에 얻어지는 원래 확률값의 상한이 된다는 뜻이다. 이 성질은 시스템 설계자가 원래의 종료시한 위반확률이 어떤 임계값보다 작을 것을 요구하는 경우 특히 중요하다. 즉, 하이퍼주기 압축을 통해 얻어진 종료시한 위반확률이 그 임계값보다 작으면 그 요구가 충족된다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 우선순위를 사용하는 주기적 실시간 시스템을 위해 [4]에서 제안된 확률적 분석 방법을 요약한다. 3절에서는 그 분석 방법의 1단계인 시스템 적체 분석을 하이퍼주기 압축 기법을 통해서 어떻게 개선시킬 수 있는지 보이고, 안전성을 증명한다. 4절에서는 실험을 통하여 시스템 적체 분석 시간이 단축되고 계산된 확률값이 실제값의 상

한이 된다는 것을 보인다. 마지막으로 5절에서는 본 논문의 결론을 맺는다.

2. 확률적 분석 방법

[4]에서 제안된 확률적 분석 방법이 가정하는 시스템은 n개의 독립된 주기적 태스크들로 구성된다. 각 태스크 $\tau_i (1 \leq i \leq n)$는 주기 T_i, 위상 ϕ_i, 실행 시간 C_i, 그리고 상대적 종료시한 D_i 등 4가지 요소로 모델링된다. 여기서 태스크 실행 시간 C_i는 확률 변수로서 최소 실행 시간 C_i^{min}부터 최대 실행 시간 C_i^{max}까지의 값을 취할 수 있고 그 확률 분포는 확률 질량 함수(probability mass function)으로 주어진다고 가정한다. 반면 각 태스크 τ_i의 j번째 작업 $J_{i,j}$의 도착 시간은 $\phi_i + (j-1)T_i$으로 고정되어 있다.

이렇게 정의된 주기적 실시간 시스템에 대해서 하이퍼주기에 속한 각 작업 $J_{i,j}$의 응답 시간 $R_{i,j}$ 역시 확률 변수이다. k번째 하이퍼주기에서 $J_{i,j}$의 응답 시간의 확률 분포를 $f_{R_{i,j}^{(k)}}(r)$라고 하면, 다음 극한 확률 분포가 존재한다는 것이 증명되었다.

$$\lim_{k \to \infty} f_{R_{i,j}^{(k)}}(r) = f_{R_{i,j}^{(\infty)}}(r)$$

따라서 하이퍼주기에 속한 모든 작업들의 극한 응답 시간 분포를 얻은 후에 태스크 τ_i에 속한 작업들의 극한 응답 시간 분포를 평균함으로써 τ_i의 응답 시간 R_i의 분포를 결정할 수 있다. 그러면, τ_i의 종료시한 위반확률 DMP_i는 다음과 같이 계산된다.

$$DMP_i = P\{R_i > D_i\} = 1 - P\{R_i \leq D_i\}$$

하이퍼주기에 속한 각 작업 $J_{i,j}$의 극한 응답 시간 분포를 계산하기 위해서는 다음 단계들을 거쳐야 한다.

(1) 시스템 적체 분석(system backlog analysis): 매 하이퍼주기 시작 시점에서 관찰되는 총 적체량의 극한 분포를 계산한다. 여기서 적체량이란 해당 시점에서 실행을 마치지 못한 작업들의 남은 실행 시간들의 총합을 말한다.
(2) 작업별 적체 분석(per-job backlog analysis): 1단계에서 얻어진 극한 시스템 적체 분포를 입력으로 하여, 하이퍼주기에 속한 각 작업 $J_{i,j}$에 대해서 그 작업의 도착 시간에 관찰되는 적체량 중 $J_{i,j}$의 실행을 지연시킬 수 있는 적체량, 즉 작업별 적체량의 극한 분포를 계산한다.
(3) 작업별 간섭 분석(per-job interference analysis): 2단계에서 얻어진 작업별 극한 적체 분포를 입력으로 하여 작업 $J_{i,j}$의 실행 시간 분포와 $J_{i,j}$를 간섭(혹은 선점)할 가능성이 있는 작업들의 실행 시간 분포를 고려하면서 $J_{i,j}$의 극한 응답 시간 분포를 최종적으로 결정한다.

이들 단계들 중에서 시스템 적체 분석 단계는 마르코프 행렬 유도 및 풀이가 요구되는 단계로서 전체 분석 시간 중에 가장 큰 시간을 차지한다[6].

3. 하이퍼주기 압축을 이용한 시스템 적체 분석

시스템 적체 분석에 소요되는 시간은 하이퍼주기의 길이에 의해서 좌우된다. 마르코프 행렬 유도는 행렬 풀이에 필요한 규칙성이 발견될 때까지, 한 하이퍼주기 시작 시점부터 끝 시점까지의 모든 작업들을 고려하여 끝 시점에 발견될 수 있는 가능한 모든 종류의 시스템 적체 분포를 계산할 것을 요구한다. 따라서, 하이퍼주기가 길면 길수록 분석에 연루되는 작업들의 수가 많아지고, 결과적으로 행렬 유도 시간 및 풀이 시간이 심각하게 커지게 된다.

따라서 시스템 적체 분석을 단순화하기 위해서는 주어진 태스크 집합의 하이퍼주기를 압축할 필요가 있다. 하이퍼주기 압축은 태스크들의 주기들의 최소공배수가 작아지도록 어떤 태스크 τ_i의 주기를 감소시킴으로써 가능하다. 예를 들어 주기가 각각 61, 100, 150인 3개의 태스크들을 생각해보자. 이 경우 하이퍼주기의 길이는 18300이나 되고 그 하이퍼주기에 속한 작업들의 수는 605개나 된다. 그러나 이 때 주기가 61인 태스크의 주기를 60으로 줄이면 하이퍼주기의 길이는 300으로 크게 줄어들 수가 있다. 이 때 이 하이퍼주기에 속한 작업들의 수는 겨우 10개이기 때문에 마르코프 행렬 유도 및 풀이는 크게 단순화되고, 따라서 시스템 적체 분석에 소요되는 시간은 대폭 줄어들 수가 있다.

그러나 하이퍼주기 압축은 각 태스크의 실행 시간 분포를 그대로 상속하면서 이루어지기 때문에 평균 시스템 이용률 $\overline{U} = \sum_{i=1}^{n} \overline{C_i}/T_i$ 의 증가를 수반하게 된다. 이러한 이용률의 증가는 압축된 하이퍼주기에서 평균 시스템 적체량이 원래의 평균 적체량보다 더 커진다는 것을 의미한다. 이것을 확률 분포의 측면에서 이해하면 다음과 같이 표현할 수 있다.

$$\sum_{w=0}^{t} f_{B_\infty}'(w) \leq \sum_{w=0}^{t} f_{B_\infty}(w) \quad \text{for any } t$$

여기서 $f_{B_\infty}'(w)$는 압축된 하이퍼주기를 대상으로 계산된 극한 시스템 적체 분포이고 $f_{B_\infty}(w)$는 원래의 하이퍼주기를 대상으로 계산된 극한 시스템 적체 분포이다. 다음 정리는 이렇게 조금 더 비관적으로(pessimistically) 계산된 극한 시스템 적체 분포가 나머지 분석 단계들, 즉 작업별 적체 분석과 간섭 분석의 입력으로 사용되면, 각 태스크의 원래의 종료시한 위반확률의 상한을 얻어낼 수 있음을 기술한다.

정리1. *압축된 하이퍼주기를 대상으로 계산된 극한 시스템 적체 분포 $f_{B_\infty}'(w)$를 사용하여 계산한 (원래의 하이퍼주기에 속한) 각 작업들 $J_{i,j}$의 극한 응답 시간 분포를 $f_{R_{i,j}}'(r)$라고 하자. 그리고, 원래의 하이퍼주기를 대상으로 계산된 극한 시스템 적체 분포 $f_{B_\infty}(w)$를 사용하여 계산한 $J_{i,j}$의 극한 응답 시간 분포를 $f_{R_{i,j}}(r)$라고 하자. 그러면, 두 분포 $f_{R_{i,j}}'(r)$와 $f_{R_{i,j}}(r)$ 사이에서도 다음 관계가 성립한다.*

$$\sum_{r=0}^{t} f_{R_{r,i}}'(r) \leq \sum_{r=0}^{t} f_{R_{r,i}}(r) \quad \text{for any } t$$

따라서, $f_{B_x}'(w)$를 사용하여 얻어진 각 태스크의 종료시한 위반확률은 원래 확률값의 상한(upper bound)이 된다.

4. 실험 결과

하이퍼주기 압축 기법이 얼마나 시스템 적체 분석 시간을 단축시킬 수 있는지와 분석 결과의 안전성을 보장하는지를 확인하기 위해 표1에서 제시된 태스크 집합을 대상으로 실험을 수행하였다. 각 태스크 τ_i는 $[C_i^{\min}, C_i^{\max}]$ 구간에서 균일 실행 시간 분포를 갖는다고 가정하였다. 그리고 이 태스크들은 Rate Monotonic 알고리즘으로 스케줄된다고 가정하였다.

표1. 태스크 집합

τ_i	T_i	D_i	$C_i^{\min}$	$\overline{C}_i$	$C_i^{\max}$
τ_1	40	40	1	9	17
τ_2	60	60	1	12	23
τ_3	100	100	1	21	41

Rate Monotonic과 같은 고정 우선순위 스케줄링을 가정하는 경우 태스크 τ_i의 하이퍼주기는 태스크 τ_1부터 τ_i까지를 포함하여 계산된다. 표1에서 제시된 태스크 집합의 경우 모든 태스크들을 고려할 때 (즉 태스크 τ_3 수준의) 하이퍼주기 H는 600이다. 이 하이퍼주기를 각 태스크의 주기를 번갈아 줄여가면서 다음 표2와 같이 300, 180, 90 등으로 압축할 수 있다. 하이퍼주기를 압축하면, 상대적으로 시스템 이용률(평균 $\overline{U}$, 최대 $U^{\max}$)은 증가하게 된다.

표2. 하이퍼주기 설정

H	$\{T_1, T_2, T_3\}$	$\overline{U}$	$U^{\max}$
600	{40, 60, 100}	0.64	1.22
300	{30, 60, 100}	0.71	1.36
180	{30, 60, 90}	0.73	1.41
90	{30, 45, 90}	0.80	1.53

표2에서 제시된 하이퍼주기 설정 각각에 대하여 시스템 적체 분석을 수행하고, 그 분석에 걸린 시간을 측정하였다. 그리고, 시스템 적체 분석 결과 얻어진 극한 시스템 적체 분포를 입력으로 작업별 적체 분석과 간섭 분석을 수행하고 태스크 τ_3의 종료시한 위반확률 DMP_3을 계산하였다(다른 태스크의 종료시한 위반확률을 계산하려면 그 태스크 수준의 하이퍼주기를 별도로 고려해야 한다).

표3으로부터, 하이퍼주기가 압축되면 그 결과 얻어진 τ_3의 종료시한 위반확률은 원래값보다 항상 크다는 것을 알 수 있다. 즉 하이퍼주기 압축은 분석 결과의 안전성을 보장한다. 그러나 τ_3의 종료시한 위반확률은 모두 2% 미만으로서 원래값에 아주 가깝다는 점도 또한 중요하다. 즉, 하이퍼주기 압축은 분석 결과의 정확성을 크게 훼손하지 않는다 (참고로, 표1에서 제시된 태스크 집합에 대해, 5000회의 하이퍼주기 동안의 시스템 진행을 1회 모의실험으로 정의할 때, 100회 모의실험을 수행한 결과 얻어진 태스크 τ_3의 종료시한 위반확률의 평균값±표준편차는 0.0143±0.0007이다). 그리고 표3으로부터 시스템 적체 분석에 소요된 시간은 하이퍼주기가 줄어들수록 크게 향상된다는 것도 확인할 수가 있다.

표3. 태스크 τ_3의 종료시한 위반확률과 시스템 적체 분석에 소요된 시간

하이퍼주기 설정	DMP_3	분석 시간 (초)
H = 600	0.0144	18.70
H = 300	0.0145	3.66
H = 180	0.0149	0.88
H = 90	0.0186	0.16

5. 결 론

본 논문은 우선순위 스케줄링을 사용하는 주기적 실시간 시스템을 위해 제안된 확률적 분석 방법의 개선 방안으로서 하이퍼주기 압축 기법을 제안하였다. 이 기법은 마르코프 행렬 유도 및 행렬 풀이 과정에서 고려해야 할 작업들의 수를 대폭 줄임으로써 분석 결과의 안전성을 보장하면서 시스템 적체 분석 시간을 크게 줄인다. 실험 결과, 제안된 기법이 정확도의 손실을 최소화하면서 상당한 계산 시간 단축을 가져올 수 있음을 확인하였다.

참고 문헌

[1] M. K. Gardner and J. W. Liu, "Analyzing Stochastic Fixed-Priority Real-Time Systems," in *Proc. of the 5th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Mar. 1999.

[2] J. P. Lehoczky, "Real-Time Queueing Theory," in *Proc. of the 17th IEEE Real-Time Systems Symposium*, pp. 186-195, Dec. 1996.

[3] G. Bernat, A. Colin, and S. Petters, "WCET Analysis of Probabilistic Hard Real-Time Systems," in *Proc. of the 23rd Real-Time Systems Symposium*, Dec. 2002.

[4] J. L. Díaz, D. F. García, K. Kim, C.-G. Lee, L. Lo Bello, J. M. López, S. L. Min, and O. Mirabella, "Stochastic Analysis of Periodic Real-Time Systems," in *Proc. of the 23rd Real-Time Systems Symposium*, Dec. 2002, pp 289-300.

[5] L. Liu and J. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *Journal of ACM*, vol. 20, no. 1, pp. 46-61, 1973.

[6] 김강희, "일반적인 우선순위 스케줄링을 사용하는 실시간 시스템을 위한 정확한 확률적 분석 방법," *정보과학회논문지 심사중*, 2003.