

이용률 요구 분석: 주기와 비주기 혼합 태스크 집합에 대한 통합 수락 제어 기법

박정근^{*0} 유민수^{**} 홍성수^{*}

^{*}서울대학교 전기·컴퓨터공학부
{jkpark⁰, sshong}@redwood.snu.ac.kr

^{**}한양대학교 정보통신대학
msryu@hanyang.ac.kr

Utilization Demand Analysis: An Integrated Admission Control Approach for a Mixed Set of Periodic and Aperiodic Tasks

Jungkeun Park^{*0}, Minsoo Ryu^{**}, Seongsoo Hong^{*}

^{*}School of Electrical Engineering & Computer Science
Seoul National University

^{**}College of Information & Communications
Hanyang University

요약

본 논문에서는 주기와 비주기 실시간 태스크가 혼합된 태스크 집합에 대하여 이용률 요구 분석(utilization demand analysis)에 기반한 수락제어 기법을 제안한다. 이용률 요구는 혼합 태스크 집합의 모든 태스크가 종료시한 내에 수행을 마치기 위한 프로세서의 이용률로 정의된다. 본 논문에서는 비주기 태스크 집합과 주기 태스크 집합 각각의 이용률 요구를 정의하고 혼합 태스크 집합에 대한 이용률 요구 분석 기법을 제공한다. 제안된 기법은 비주기 태스크에 대한 이용률 요구를 재귀적으로 계산하여 태스크의 수락여부를 결정한다. 이러한 기법은 $O(n)$ 에 스케줄링 분석을 수행하고 복잡한 자료구조가 필요하지 않아서 실시간 수락제어를 가능하게 한다.

1. 서론

실시간 스케줄링에서의 핵심은 주어진 태스크 집합에 대하여 스케줄링 가능성을 분석하는 문제이다. 현재까지 많은 스케줄링 알고리즘과 이에 대한 스케줄링 분석 방법이 제안되었다[1,2,3]. 이들 방법은 대부분 주기 태스크 모델에 중점을 두고 있다. 그러나 최근 네트워크 중심의 서버/클라이언트 시스템에서는 주기와 비주기 요청이 혼합되어 존재하며 QoS 보장과 같은 실시간성 보장이 요구되고 있다. 비주기 태스크에 대한 스케줄링 분석이 어려운 근본적인 원인은 비주기 태스크의 도착 시간과 수행시간이 태스크 도착 이전에 예측 불가능하다는 점이다. 최근 제한적인 비주기 태스크 모델에 의해 주기와 비주기 혼합 태스크 집합에 대한 스케줄링 분석 방법이 제안되고 있으나 이들은 비판적 가정에 의해 수행되어 자원 이용률(utilization)이 매우 낮거나[4] 연성 실시간 시스템에만 적용되는[5] 등의 단점이 있다.

본 논문에서는 이용률 요구 분석(utilization demand analysis)에 기반하여 주기와 비주기 혼합 태스크 집합에 대한 새로운 스케줄링 가능성 분석과 수락제어 기법을 제안한다. 이용률 요구는 혼합 태스크 집합의 모든 태스크가 종료시한 이내에 끝나기 위한 프로세서의 이용률로 정의된다. 제안된 수락제어 기법은 주기와 비주기 태스크가 혼합된 태스크 집합에서 비주기 태스크에 대한 이용률 요구를 재귀적으로 계산하여 태스크의 수락여부를 결정한다. 이 기법은 $O(n)$ 의 복잡도를 가지며 매우 정확한 수락제어 조건을 제공한다.

2. 모델과 가정

우선순위 순서로 정렬된 비주기 태스크의 집합을

$Q = \{\tau_1, \tau_2, \dots, \tau_i, \dots\}$ 로 나타낸다. $Q(t) \subset Q$ 는 시간 t 까지 수락되었지만 아직 종료하지 않은 태스크의 집합이다. 비주기 태스크 $\tau_i \in Q$ 의 도착시간은 A_i , 도착시간으로부터 상대 종료시한은 d_i 로 표시하며 절대 종료시한 $D_i = A_i + d_i$ 이다. 각 비주기 태스크의 최악 수행시간은 e_i 이다.

주기 태스크는 $\bar{\tau}_i$ 로 표시한다. 주기 태스크는 주기는 \bar{T}_i 이고 주기 태스크의 각 인스턴스 $\bar{\tau}_{i,j}$ 의 상대 종료시한은 주기와 같고 \bar{d}_i 로 표시한다. 각 주기 태스크의 최악 수행시간은 \bar{e}_i 이다.

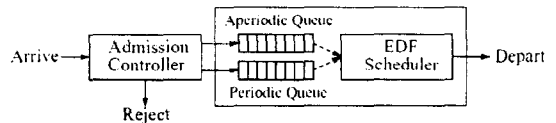


그림 1: 시스템 구조

본 논문에서는 그림 1과 같이 수락 제어기와 태스크 큐 그리고 태스크 스케줄러로 구성된 단순한 시스템 구조를 고려한다. 또한 그림에서와 같이 본 논문에서는 EDF 스케줄만을 고려한다.

3. 이용률 요구 분석

3.1 비주기 태스크에 대한 이용률 요구 분석

독립적 비주기 태스크의 집합 $Q(t) = \{\tau_1, \tau_2, \dots, \tau_i, \dots\}$ 가 EDF에 의해 스케줄링 된다고 가정한다. $\tau_i \in Q(t)$ 에 대해 τ_i

가 종료하기까지 남은 잉여 수행시간(residual execution time)을 $e_{i,t}$ 로 정의하고 절대 종료시한과 현재 시간의 차이를 리드 시간(lead time) $d_{i,t} = D_i - t$ 로 정의한다.

잉여 수행시간과 리드 시간에 의해 $\tau_i \in Q(t)$ 에 대한 이용률 요구 $U_{Q(t)}(\tau_i)$ 는 τ_i 가 종료시한 이전에 마치기 위해 필요한 프로세서 시간을 리드 시간으로 나눈 값으로 정의된다. 일반적인 경우에 τ_i 는 높은 우선순위 태스크가 수행을 마친 후에 수행될 수 있으므로 $U_{Q(t)}(\tau_i)$ 는 본자에는 τ_i 보다 높은 우선순위 태스크의 잉여시간을 포함한다. $Q(t, ed(t')) \subset Q(t)$ 를 절대 종료시한이 t' 보다 빠른 태스크들로 정의할 때 τ_i 에 대한 이용률 요구는 다음과 같다.

$$U_{Q(t)}(\tau_i) \stackrel{\text{def}}{=} \frac{\sum_{\tau_j \in Q(t, ed(D_j))} e_{j,t}}{d_{i,t}} \quad (1)$$

다음 정리는 비주기 태스크 집합에 대한 스케줄링 가능성의 필요충분 조건을 나타낸다.

정리 1: 비주기 태스크 집합 $Q(t)$ 는 임의의 $\tau_i \in Q(t)$ 에 대해

$$U_{Q(t)}(\tau_i) \leq 1 \quad (2)$$

가 만족될 때 필요충분 조건으로 EDF에 의해 스케줄 가능하다.

그러나 $\tau_i \in Q(t)$ 에 대한 스케줄링 가능성은 $\tau_i \in Q$ 에 대한 스케줄링 가능성과 다르다. 왜냐하면 $Q(t)$ 에는 t 이후에 수락될 태스크들이 포함되어 있지 않기 때문이다. 따라서 부등식 (2)에 의한 스케줄링 가능성은 다음 새로운 태스크가 도착할 때까지 유효하다. 따라서 매 비주기 태스크의 도착시점 A_i 마다 부등식 (2)에 의한 스케줄링 가능성 조사가 수행되어야 한다.

식 (1)에 의한 이용률 요구 정의는 미래의 태스크 도착이 예측 가능한 경우 이를 반영시키지 못한다. 이를 위해 본 논문에서는 이용률 요구를 특정 시간 구간에 대한 값으로 확장한다. $Q(t, t') \subset Q(t)$ 를 $Q(t)$ 와 시간 $[t, t']$ 에서 수락될 태스크들을 포함하는 집합으로 정의한다. 이 정의에 의해 $Q(t, \infty)$ 는 t 에 종료되지 않은 태스크들과 t 이후에 수락될 모든 태스크의 집합을 나타낸다. $Q((t, t'), ed(D_i)) \subset Q(t, t')$ 을 절대 종료시한이 D_i 보다 빠른 태스크 집합이라 할 때 $\tau_i \in Q(t, t')$ 에 대한 이용률 요구는 다음과 같이 확장 정의한다.

$$U_{Q(t,t')}(\tau_i) \stackrel{\text{def}}{=} \frac{\sum_{\tau_j \in Q((t,t'), ed(D_j))} e_{j,t}}{D_i - t} \quad (3)$$

위의 정의에 의해 더 일반적인 스케줄링 가능성 조건을 유도할 수 있다.

정리 2: 비주기 태스크 집합 Q 는 임의의 $\tau_i \in Q$ 에 대해

$$U_{Q(A_i, D_i)}(\tau_i) \leq 1 \quad (4)$$

가 만족될 때 충분 조건으로 EDF에 의해 스케줄 가능하다.

3.2 주기 태스크에 대한 이용률 요구 분석

주기 태스크 집합 $P = \{\tau_1, \tau_2, \dots, \tau_N\}$ 에 대해 각 주기 태스

크의 인스턴스는 비주기 태스크로 볼 수 있다. 즉 어떤 주기 태스크의 인스턴스 $\tau_{i,j}$ 는 도착시간이 $A_k = \overline{A}_i + (j-1)\overline{T}_i$ 이고 상대 종료시한이 $d_k = \overline{T}_i$ 인 비주기 태스크 τ_k 로 볼 수 있다. 위와 같이 주기 태스크 집합 P 에 대해 가상 비주기 태스크 집합(pseudo-a-periodic set) Q_p 를 연관시킬 수 있다. 이때 P 는 Q_p 가 스케줄 가능할 때만 필요충분 조건으로 스케줄 가능하다. 따라서 주기 태스크 집합에 대해서도 비주기 태스크에 대한 스케줄링 가능성 검사와 유사한 방법으로 스케줄링 가능성 검사를 할 수 있다.

Q_p 는 t 이후의 태스크 도착에 대한 정보를 알 수 있고 t 이후에 도착한 태스크에 대해서도 종료시한이 보장되어야 한다는 점에서 일반 비주기 태스크 집합 Q 와 다르다. 다음 정리는 식 (3)의 확장 이용률 요구 정의를 이용하여 주기 태스크에 대한 이용률 요구의 특성을 보인다.

정리 3: 주기 태스크 집합 $P = \{\tau_1, \tau_2, \dots, \tau_N\}$ 에 대한 이용률

이 $U_p = \sum e_i / \overline{T}_i$ 일 때, 집합 P 가 EDF에 의해 스케줄링 가능하면 임의의 $\tau_i \in Q_p$ 와 $t \geq 0, t' \geq 0$ 에 대해서 다음 부등식이 성립한다.

$$U_{Q_p(t,t')}(\tau_i) \leq U_p \quad (5)$$

정리 3은 식 (3)에 의한 이용률 요구 분석이 U_p 보다 더 정확한 이용률 값을 반영한다는 것을 보여준다.

3.3 혼합 태스크 집합에 대한 스케줄 가능성 분석

집합 S 를 일반 비주기 태스크 집합 Q 와 주기 태스크 집합 P 에 대한 가상 비주기 태스크 집합 Q_p 의 합집합이라 하면 $S = Q \cup Q_p$ 이다. 정리 2에 의해 임의의 $\tau_i \in S(t, \infty)$ 에 대해 $U_{S(t, \infty)}(\tau_i) \leq 1$ 의 조건을 만족시키면 $S(t, \infty)$ 는 스케줄링 가능하다. 그러나 이는 무한개의 태스크를 검사해야 하므로 구현할 수 없다.

본 논문에서는 주기와 비주기 태스크가 혼합되어 스케줄링 될 때 비주기 태스크가 스케줄링 가능하기 위한 이용률 요구의 상한을 구하고 이용률 요구의 상한이 재귀적(recursive) 방법에 의해 계산될 수 있다는 것을 보인다. 그리고 모든 비주기 태스크의 이용률 요구가 스케줄링 가능한 상한을 만족하면 주기 태스크들도 스케줄링 가능하다는 것을 보인다. 제안된 방법은 새로 도착한 태스크 τ_i 보다 우선 순위가 낮은 비주기 태스크들에 대해서만 이용률 요구분석을 재귀적으로 수행하므로 실시간 수락제어에 사용 가능하다.

비주기 태스크들의 이용률 요구를 재귀적으로 계산하기 위해 태스크 $\tau_i \in Q$ 의 이용률 요구에 의한 가상 종료시각 f_i 를 정의한다. $\tau_i \in Q$ 에 대한 이용률 요구 $U_{S(A_i, D_i)}(\tau_i)$ 를 간단히 $U(\tau_i)$ 로 표시할 때 가상 종료시각 $f_i = A_i + U(\tau_i) \cdot d_i$ 이다. 즉, 태스크 τ_i 의 가상 종료시각은 구간 $[A_i, D_i]$ 에서 τ_i 보다 높은 우선순위의 주기와 비주기 태스크의 수행시간을 모두 합한 값이다. 만약 τ_i 가 스케줄링 가능하면 가상 종료시각은 절대 종료

시한 D_i 보다 작다.

가상 종료시각에 의한 이용률 요구는 다음과 같이 계산된다. 주어진 주기와 비주기 태스크의 혼합 집합 $S(t, \infty) = Q(t) \cup Q_p(t, \infty)$ 에 대하여 $Q(t) = \{\tau_1, \dots, \tau_i, \dots, \tau_n\}$ 가 우선순위로 정렬되어 있다고 가정하자. 즉, τ_{i-1} 의 우선순위가 τ_i 의 우선순위보다 높다. $\tau_i \in Q$ 에 대한 이용률 요구 $U(\tau_i)$ 를 계산하기 위해서는 구간 $[A_i, D_i]$ 에서 τ_i 이외의 태스크들의 프로세스 시간 요구량을 구해야 한다. $\tau_{i-1} \in Q$ 의 가상 종료시각을 f_{i-1} 라 하고 τ_{i-1} 보다 우선순위가 같거나 높은 태스크들이 모두 스케줄링 가능하다고 하면 $f_{i-1} < D_{i-1}$ 이다. 구간 $[A_i, D_i]$ 사이에서의 프로세스 시간 요구량은 구간 $[A_i, D_{i-1}]$ 와 구간 $[D_{i-1}, D_i]$ 로 나누어 생각할 수 있다. 본 논문에서는 구간 $[A_i, D_{i-1}]$ 사이에서의 프로세스 시간 요구량이 $\max\{f_{i-1} - A_i, 0\}$ 보다 작고 구간 $[D_{i-1}, D_i]$ 에서의 프로세스 시간 요구량이 $U_p \cdot (D_i - D_{i-1})$ 보다 작다는 것을 보인다. 이를 이용하면 $\tau_i \in Q$ 의 이용률 요구 $U(\tau_i)$ 는 다음 부등식을 만족한다.

$$U(\tau_i) \leq \frac{\max\{f_{i-1} - A_i, 0\} + e_i + U_p \cdot (D_i - D_{i-1})}{D_i - A_i} \quad (6)$$

여기에서 $f_{i-1} = A_{i-1} + U(\tau_{i-1}) \cdot d_{i-1}$ 이므로 $U(\tau_{i-1})$ 값으로부터 $U(\tau_i)$ 의 최대값을 재귀적으로 계산해 나갈 수 있다. 다음은 주기와 비주기 혼합 태스크 집합에서 비주기 태스크의 스케줄링 가능성 분석에 대한 정리이다.

정리 4: 혼합 태스크 집합 $S(t, \infty) = Q(t) \cup Q_p(t, \infty)$ 에 대하여 $Q(t) = \{\tau_1, \dots, \tau_i, \dots, \tau_n\}$ 가 우선 순위에 의해 정렬되어 τ_{i-1} 의 우선순위가 τ_i 의 우선순위보다 높을 때, 비주기 태스크 $\tau_i \in Q$ 는 다음 조건을 만족시킬 때 스케줄링 가능하다.

$$\frac{\max\{f_{i-1} - A_i, 0\} + e_i + U_p \cdot (D_i - D_{i-1})}{D_i - A_i} \leq 1 \quad (7)$$

여기에서 $f_{i-1} = A_{i-1} + U(\tau_{i-1}) \cdot d_{i-1}$ 이다.

정리 4에 의해 비주기 태스크의 스케줄링 가능성 분석을 상수 시간에 수행할 수 있다. 이를 위해 각 태스크들은 이용률 요구 $U(\tau_i)$ 와 가상 종료시각 f_i 를 유지한다. 새로운 비주기 태스크 τ_i 가 도착하면 τ_i 보다 우선순위가 높은 태스크 중 절대 종료시한이 D_i 에 가장 가까운 태스크 τ_{i-1} 을 찾아 정리 4에 의해 스케줄링 가능성 분석을 수행한다.

그러나 정리 4는 주기 태스크의 스케줄링 가능성을 제시하지는 못하고 있다. 다음 정리는 주기 태스크의 스케줄링 가능성을 보장하기 위한 조건을 제시한다.

정리 5: 혼합 태스크 집합 $S(t, \infty) = Q(t) \cup Q_p(t, \infty)$ 에 대하여

$Q(t) = \{\tau_1, \dots, \tau_i, \dots, \tau_n\}$ 가 우선 순위에 의해 정렬되어 τ_{i-1} 의 우선순위가 τ_i 의 우선순위보다 높을 때, 모든 주기 태스크 $\tau_p \in Q_p(t, \infty)$ 는 모든 비주기 태스크 $\tau_i \in Q(t)$ 가 다음 조건을 만족시킬 때 스케줄링 가능하다.

$$\frac{\max\{f_{i-1} - A_i, 0\} + e_i + U_p \cdot (D_i - D_{i-1})}{D_i - A_i} \leq 1 \quad (8)$$

여기에서 $f_{i-1} = A_{i-1} + U(\tau_{i-1}) \cdot d_{i-1}$ 이다.

정리 5에 의해 주기와 비주기 태스크가 혼합된 태스크 집합에서 비주기 태스크에 대한 수락제어 조건을 구할 수 있다. 새로운 비주기 태스크 τ_i 가 수락되기 위해서는 τ_i 가 스케줄링 가능해야 할 뿐만 아니라 이미 수락되어 수행되고 있는 τ_i 보다 낮은 우선순위 태스크의 스케줄링 가능성도 해치지 않아야 한다. 따라서 식 (6)에 의해 낮은 우선순위 태스크들의 이용률 요구와 가상 종료시한을 재귀적으로 계산하고 정리 5에 의해 스케줄링 가능성 분석을 수행한다. 주기 태스크의 영향은 식 (8)에서 U_p 에 의해 모두 표현되므로 주기 태스크에 대한 스케줄링 분석은 별도로 수행하지 않아도 된다. 위의 수락제어 기법은 현재 $Q(t)$ 의 비주기 태스크의 개수를 n 이라 할 때 $O(n)$ 의 복잡성을 가진다.

4. 결론

본 논문에서는 이용률 요구 분석을 이용한 주기와 비주기 태스크의 통합 수락제어 방법을 제안하였다. 제안된 수락제어 방법은 주기와 비주기 태스크가 혼합된 태스크 집합에 대해 비주기 태스크의 이용률 요구를 재귀적으로 계산하여 스케줄링 가능성을 조사한다. 또한 제안된 수락제어 방법은 매우 적은 오버헤드로 구현될 수 있어 시스템 수행 중에 새로운 태스크의 수락여부를 효과적으로 계산할 수 있다.

본 논문에서는 태스크들이 EDF 의해 스케줄링 되는 경우만을 살펴보았다. 그러나 본 논문에서 제시된 이용률 요구분석과 수락제어 방법은 일반적인 고정 우선순위 기반 스케줄링에 대해서도 쉽게 확장 가능하다.

참고문헌

- [1] C. Liu and J. Layland. Scheduling algorithm for multiprogramming in a hard real-time environment. Journal of the ACM, 20(1):46-61, 1973.
- [2] N. Audsley, A. Burns, M. Richardson, and A. Wellings. Hard real-time scheduling: The deadline-monotonic approach. In Proceeding of IEEE Workshop on Real-Time Operating Systems and Software, pp. 133-137, 1991.
- [3] J. P. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. In Proceedings of IEEE Real-Time Systems Symposium, pp. 166-171, 1989.
- [4] T. F. Abdelzäher, V. Sharma, and C. Lu. A utilization bound for aperiodic tasks and priority driven scheduling. IEEE Trans. on Computers, 53(3):334-350, March 2004.
- [5] M. Spuri and G. Buttazzo. Scheduling aperiodic tasks in dynamic priority systems. Real-Time Systems, 10(2):179-210, 1996.