

중단 호스트에서 QoS 보장을 위한 비례 분배 실시간 자원할당 기법

박정근*^o 유인수** 홍성수* 박선희***

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

**한양대학교 정보통신대학
msryu@redwood.snu.ac.kr

***서울대학교 자동화시스템공통연구소
shpark@redwood.snu.ac.kr

Fair Real-Time Resource Allocation for End System's QoS Support

Jungkeun Park*^o, Minsoo Ryu**, Seongsoo Hong*, Sunheui Park***

*School of Electrical Engineering & Computer Science
Seoul National University

**Collage of Information & Communications
Hanyang University

***Automation and Systems Research Institute (ASRI)
Seoul National University

요약

본 논문에서는 인터넷 중단 호스트에서 공유 자원의 대역폭 제약조건과 종료시한 제약조건 모두를 만족시킬 수 있는 자원 할당 구조를 제안한다. 제안된 구조는 두 단계로 구성된다. 상위 단계에서는 비례 분배 스케줄러 (proportional share scheduler)인 EFT-C/D (Earliest Finish Time Credit/Debit) 스케줄러가 수행된다. 이 스케줄러는 CPU와 같은 시분할 공유 자원을 하위 단계 스케줄러들에게 지정된 비율로 분배하는 역할을 한다. 그리고 하위 단계에서는 서로 다른 시간 제약조건이 부여된 태스크들을 스케줄링 하기 위해 다양한 실시간 스케줄러가 수행된다. 본 연구의 주요 성과는 두 가지로 요약된다. 첫째, 이상적인 GPS (Generalized Processor Sharing) 서버와 거의 동등한 수준으로 자원을 공정하게 분배하는 EFT-C/D 알고리즘을 개발하였다. 둘째, 하위 단계에서 수행되는 EDF 스케줄러에 대해 이용률(utilization)에 기반한 스케줄링 가능성 분석 방법을 개발하였다. 이 방법은 주어진 태스크 집합에 대해 단순히 이용률만을 계산하여 스케줄링 가능성을 판별할 수 있다. 따라서 새로운 태스크가 생성될 때 수락 여부를 시스템 수행 중에 제어할 수 있는 장점이 있다.

1. 서론

최근 프로세서와 네트워크 기술이 발달하면서 QoS 보장이 요구되는 다양한 응용들이 시스템 내에서 동시에 수행될 수 있게 되었다. 이러한 응용들은 각기 다른 성질의 시간 제약조건이 부여된다. 예를 들어, 인터넷 비디오와 같은 멀티미디어 응용에서는 화면을 부드럽게 보여주기 위해 대역폭 제약조건이 부여된다. 반면, 온라인 증권거래나 네트워크 게임 등에서는 종료시한 제약조건이 부여된다. 이는 시간 이내에 수행되지 않으면 금전적인 손실을 야기하거나 게임의 흐름이 끊어지는 문제점이 생기기 때문이다. 이런 각기 다른 성질의 시간 제약조건이 부여된 응용 프로그램을 동시에 수행하기 위해서는 중단 호스트에서의 효율적인 자원 관리 기법이 필요하다.

이를 해결하기 위해 본 논문에서는 그림 1과 같이 두 단계로 이루어진 자원 할당 구조를 제안한다. 상위 단계에서는 비례 분배 스케줄러가 CPU와 같은 시분할 공유 자원을 하위 단계 스케줄러들에게 각각 지정된 비율로 나누어 준다. 본 논문에서는 비례 분배 스케줄러 알고리즘으로 최소 종료시한 크레딧/데빗(Earliest Finish Time Credit/Debit, EFT-C/D)을 개발하였다. 이 알고리즘은 [1]에서 소개된 크레딧/데빗 알고리즘을 확장한 것이다. 하위 단계에서는 서로 다른 시간 제약조건이 부여된 태스크 집합을 스케줄링 하기 위한 다양한 스케줄러들이 수행된다. 예를 들어 종료시한 제약조건이 부여된 태스크들은 EDF(Earliest Deadline First) 스케줄러로 스케줄링되며, 대역폭 제약조건이 부여된 태스크들은 본 논문에서 개발한 EFT-C/D나 EEVDF(Earliest Eligible Virtual Deadline First) 스케줄러 [4]와 같은 비례 분배 스케줄러로 스케줄링 된다. 제안된 스케줄링 구

조는 이밖에도 RM(Rate Monotonic) 스케줄러와 같은 다양한 스케줄러를 지원할 수 있으나 본 논문에서는 상위 단계 스케줄러로 EFT-C/D, 하위 단계 스케줄러로 EDF 만을 선택하여 다룬다.

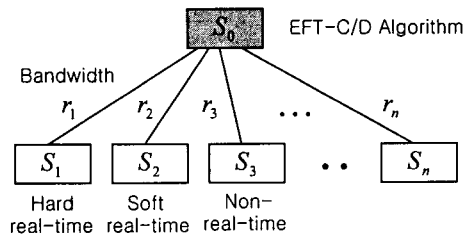


그림 1: 두 단계 스케줄링 구조

본 연구의 성과는 크게 두 가지로 요약된다. 첫째, 이상적인 GPS(Generalized Processor Sharing) 서버 [2]와 거의 동등한 수준으로 자원을 공정하게 분배하는 EFT-C/D 알고리즘을 개발하였다. EFT-C/D는 GPS와 비교하여 태스크에 제공되는 서비스 시간의 차이가 스케줄링 퀀텀(quantum)보다 길지 않다. EFT-C/D의 수행시간은 $O(n)$ 으로 태스크의 개수에 비례하여 증가한다. 이 단점에도 불구하고 EFT-C/D는 가상시간(virtual time) [2]을 유지할 필요가 없어 구현이 매우 쉬우며 런타임 오버헤드도 적다. 둘째, 하위 단계에서 수행되는 EDF 스케줄러에 대해 이용률에 기반한 스케줄링 가능성 분석 방법을 개발하였다. 이 방법을 이용하면 주어진 태스크 집합에 대하여 단순히 자원 이용률만을 계산하여 스케줄링 가능성을 판별할 수 있다. 따라서 새로운 태스크가 생성될 때 수락 여부를 시스템 수행 중에 제어할 수 있

는 장점이 있다.

계층적 구조를 가지는 스케줄링 방법은 몇몇 논문의 연구결과로 알려져 있다. Liu와 Deng은 [6]에서 EDF 스케줄러를 상위 단계 스케줄러로 하고 CBS(Constant Utilization Server)에 기반한 하위 단계 스케줄러에서 응용에 맞는 실시간 스케줄링을 수행한다. 이 방법은 종료시한을 가지는 실시간 태스크와 비실시간 태스크를 모두 지원하지만 대역폭 제약조건이 부여된 태스크를 스케줄링 하지 못한다.

2. 모델과 가정

경성 실시간 태스크 τ_i 는 상대 종료시한 d_i 와 최악 수행시간 e_i 로 표현된다. 태스크가 주기적일 경우 태스크는 $\tilde{\tau}_i$ 로 표현되고 주기는 T_i 로 표현된다.

제한된 스케줄링 구조의 하위 단계에서는 특히 다음 두가지 스케줄러가 고려된다. 경성 실시간 스케줄러 S_1 은 상위 단계 스케줄러 S_0 로부터 r_1 의 상대 비율로 CPU를 할당 받아 EDF 알고리즘에 의해 경성 실시간 태스크를 스케줄링 한다. 연성 실시간 스케줄러 S_2 는 r_2 의 비율로 EFT-C/D나 EEVDF[4]를 사용하여 연성 실시간 태스크들을 스케줄링 한다.

3. 프로세서 대역폭의 비례할당

3.1 크레딧/데빗 알고리즘

크레딧/데빗 알고리즘은 CPU 시간과 같은 시분할 자원을 공유하기 위한 비례 할당 알고리즘의 한 종류이다. CPU는 일정 시간단위로 할당되는데 스케줄링 시간단위를 시간퀀텀(Δ)로 표현한다. 이 알고리즘은 하위 단계 스케줄러(S_j)에 대해 크레딧 값(C_i)을 유지한다. C_i 는 상위 단계 스케줄러가 시간퀀텀을 할당하는데 사용되는 일종의 가상시간으로 볼 수 있다. 알고리즘은 다음과 같다. 상위 단계 스케줄러는 C_i 를 0으로 초기화한다. 매 퀀텀이 시작될 때 스케줄러는 각 C_i 를 $(r_i/\sum_j r_j)\Delta$ 만큼 증가시키고 가장 큰 C_i 를 가진 하위 단계 스케줄러를 선택하여 수행시킨다. 같은 최대 크레딧 값이 두 개 이상 있으면 임의로 선택한다. 선택된 스케줄러에 대해서는 C_i 에서 할당된 퀀텀 만큼을 뺀다.

크레딧 값 $C_i(t)$ 는 이상적인 GPS에서 수행되었을 경우와 비교하여 S_j 가 시간 t 에 얼마나 더 혹은 덜 수행되었는가를 나타낸다. 선택된 스케줄러의 C_i 가 0이면 이상적인 GPS서버와 동일한 시점에 스케줄링 된 것을 의미하고 0보다 크면 GPS서버보다 늦게 서비스 된 것을 의미한다.

3.2 최소 종료 시간 크레딧/데빗 알고리즘

본 논문에서는 패킷 페어 큐잉 알고리즘인 WF²Q를 에뮬레이트 하도록 기본 크레딧/데빗 알고리즘을 수정한다. 여러 세션의 패킷을 멀티플렉싱 하는 스위치에서 만약 패킷이 일정 크기로 도착하고 각 패킷을 처리하는 시간이 Δ 로 일정하면 패킷 큐잉 문제는 본 논문의 프로세서 스케줄링 문제로 바뀔 수 있다. WF²Q는 GPS에서 현재 서비스를 받기 시작한 패킷 중에 가장

빨리 서비스를 마치고 떠날 패킷을 먼저 스케줄링 한다. WF²Q를 에뮬레이트 하기위해 해당 GPS 서버를 우선 고려한다. $q_{i,j}$ 를 S_j 가 GPS상에서 j 번째로 받은 퀀텀이라 하고 $f_{i,j}$ 를 S_j 가 $q_{i,j}$ 의 사용을 마치는 시간이라고 하자. 본 논문에서는 크레딧/데빗 알고리즘을 양의 크레딧 값을 가지는 하위 단계 스케줄러 S_j 중 가장 작은 $f_{i,j}$ 를 가지는 스케줄러를 선택하도록 수정하였다. 수정된 알고리즘은 다음 조건에 의해 하위 단계 스케줄러를 선택한다.

$$\min \left\{ \frac{\Delta - C_i}{r_i} \right\} \text{ and } C_i > 0 \quad (1)$$

수정된 알고리즘은 최소 종료 시간 크레딧/데빗(Earliest Finish Time Credit/Debit, EFT-C/D)로 부르기로 한다.

S_j 가 EFT-C/D에 의해 $[0, t]$ 에서 서비스 받은 시간을 $W_i^{CD}(0, t)$ 로 표시하고 GPS에 의해 서비스 받은 시간을 $W_i^{GPS}(0, t)$ 로 하면 서비스 지연 시간은 다음과 같다.

$$E_i(t) = W_i^{GPS}(0, t) - W_i^{CD}(0, t) \quad (2)$$

서비스 지연 시간은 EFT-C/D가 얼마나 이상적인 경우와 유사하게 자원을 분배하였는지를 나타낸다. 다음 정리는 EFT-C/D에 의한 서비스 지연이 퀀텀 크기인 Δ 이하로 제한되는 것을 보여준다.

정리 1: EFT-C/D의 서비스 지연시간은 다음 조건을 만족시킨다.

$$|E_i(t)| = |W_i^{GPS}(0, t) - W_i^{CD}(0, t)| \leq \Delta \quad (3)$$

증명: [3]의 정리 1 참조.

4. 하위 단계 EDF 스케줄러의 종료시한 보장

4.1 이용률 요구량(utilization demand) 분석

우선순위에 기반하여 스케줄링 되는 비주기 태스크의 집합 $Q = \{\tau_1, \tau_2, \dots, \tau_j, \dots\}$ 의 각 태스크 $\tau_i \in Q(t)$ 에 대해서 최대 잉여 수행시간(maximum residual execution time) $e_{i,t}^{res}$ 과 리드 시간(lead time) $d_{i,t}^{res}$ 을 생각해 보자. 시간 t 에서 $e_{i,t}^{res}$ 는 τ_i 가 종료할 때까지 남은 최대 수행시간이다. $d_{i,t}^{res}$ 는 τ_i 의 절대 종료시한 D_i 와 현재시간 t 의 차이, 즉 $D_i - t$ 이다. 이를 이용하여 $\tau_i \in Q(t)$ 의 이용률 요구량은 현재 태스크가 종료시한 전에 수행을 마치기 위해 남은 수행시간을 리드 시간으로 나눈 값이다. τ_i 보다 높은 우선순위의 태스크 집합을 $Q(t, hp(\tau_i)) \subset Q(t)$ 라 할때 τ_i 의 이용률 요구량은 다음과 같이 정의된다.

$$u_{Q(t)}(\tau_i) = \frac{\sum_{\tau_j \in Q(t, hp(\tau_i))} e_{j,t}^{res} + e_{i,t}^{res}}{d_{i,t}^{res}} \quad (4)$$

또한 최대 이용률 요구량 $U_{Q(t)}$ 는 다음과 같이 정의된다.

$$U_{Q(t)} = \max[u_{Q(t)}(\tau_i)] \quad (5)$$

이 때 주기 태스크와 비주기 태스크가 혼합된 집합에 대한 스케줄 가능성의 충분조건은 다음 정리와 같다.

정리 2: 주기 태스크 집합 P 와 비주기 태스크 집합 $Q(t)$ 에 대해 주기 태스크의 이용율을 U_p 라 할 때 $U_p + U_{Q(t)} \leq 1$ 이면 $P \cup Q(t)$ 는 EDF에 의해 스케줄 가능하다.

증명: [5] 참조

4.2 스케줄링 가능성 분석

그림 2에서 하위 단계 스케줄러 S_i 에 대한 쿼터 할당을 살펴 보자. 정리 1에 의해 S_i 에 할당된 쿼터 간의 최대 간격은 $(\sum_{j=1}^n r_j / r_i) \cdot 2\Delta - \Delta$ 이다. 따라서 하위 단계 스케줄러 S_i 는 $[t, t + (\sum_{j=1}^n r_j / r_i) \cdot 2\Delta]$ 시간동안 적어도 Δ 만큼의 시간을 할당 받는다. 일반적으로 m 개의 연속적인 쿼터 할당을 고려할 때 스케줄러 S_i 는 $[t, t + (\sum_{j=1}^n r_j / r_i) \cdot m\Delta]$ 시간 동안 $m-1$ 개의 쿼터를 보장 받는다. 따라서 하위 단계 스케줄러는 $[t, t + (\sum_{j=1}^n r_j / r_i) \cdot m\Delta]$ 기간동안

$$U_i^{lower} = \frac{(m-1)\Delta}{(\sum_{j=1}^n r_j / r_i) \cdot m\Delta} = \frac{m-1}{m} \frac{r_i}{\sum_{j=1}^n r_j} \quad (6)$$

만큼의 이용율을 보장 받는다. 만약 $(\sum_{j=1}^n r_j / r_i) \cdot 2\Delta$ 보다 짧은 구간을 선택하면 이 구간 내에서 하위 단계 스케줄러가 서비스 받는 것을 보장할 수 없다. 따라서 태스크의 종료시한이 $(\sum_{j=1}^n r_j / r_i) \cdot 2\Delta$ 보다 짧으면 이를 보장할 수 없다.

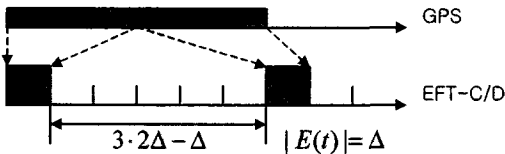


그림 2: 1/3로 수행될 때 연속된 쿼터의 최대 간격

하위 단계 EDF스케줄러에 의해 스케줄링 되는 주기 태스크와 비주기 태스크의 혼합 집합 $G(t) = P \cup Q(t)$ 에 대해 최소 종료시한을 $D = \min\{d_k | \tau_k \in G\}$ 와 같이 정의하자. 본 논문에서는 앞서 설명한 것과 같이 $D \geq (\sum_{j=1}^n r_j / r_i) \cdot 2\Delta$ 인 경우만을 고려한다. 이를 이용하여 최악 이용율에 기반한 스케줄링 가능성 테스트를 다음 정리와 같이 할 수 있다.

정리 3: $G = P \cup Q(t)$ 는 주기 태스크와 비주기 태스크를 모두 포함하는 집합이고 M 이 $(\sum_{j=1}^n r_j / r_i) \cdot m\Delta \leq D$ 를 만족시키는 최대 m 이라고 하자. G 가 EDF 스케줄링에 의해 속도가

$\frac{M-1}{M} \frac{r_i}{\sum_{j=1}^n r_j}$ 로 느린 프로세서에서 스케줄링 될 때 만약

$U_{G(t)}$ 가 $\frac{M-1}{M} \frac{r_i}{\sum_{j=1}^n r_j}$ 보다 작으면 G 는 EFT-C/D에 의해

스케줄링 되는 하위 단계 EDF 스케줄러에서도 스케줄링 가능하다.

증명: τ_k 를 $G(t)$ 상의 임의의 비주기 태스크 또는 임의의 주기 태스크의 한 인스턴스라 하고 도착시간을 a_k 라고 하자. 그러면 하위 단계 EDF 스케줄러가 $[a_k, a_k + d_k]$ 동안 τ_k 가 수행을 마치기 위한 충분한 시간을 할당받는다는 것을 보이면 된다. 정의에 의해 τ_k 가 종료시한 전에 수행을 마치기 위해 필요한 시간은 $d_k \cdot u_{G(t)}(\tau_k)$ 이다. 식 (6)으로부터 하위 단계 EDF 스케줄러 S_i 는 $d_k \frac{M-1}{M} \frac{r_i}{\sum_{j=1}^n r_j}$ 이상의 시간을 보장받는다. 한편 가정

에 의해 $u_{G(t)}(\tau_k) \leq U_{G(t)} \leq \frac{M-1}{M} \frac{r_i}{\sum_{j=1}^n r_j}$ 이므로 시간요구

량은 $d_k \cdot u_{G(t)}(\tau_k) \leq d_k \frac{M-1}{M} \frac{r_i}{\sum_{j=1}^n r_j}$ 을 만족한다.

5. 결론

본 논문에서는 중단 시스템에서 대역폭과 종료시한 요구조건을 모두 다룰 수 있는 계층적 스케줄러를 제안하였다. 제안된 스케줄러는 하위 단계 스케줄러에게 지정된 비율로 자원을 분배하는 상위 단계 EFT-C/D 스케줄러를 기반으로 하고 있다. EFT-C/D 스케줄러는 WF²Q를 에뮬레이트면서 가상시간을 사용하지 않고도 공정성을 보장해 준다. 또한 본 논문에서는 하위 단계 EDF 스케줄러에 대해 이용율에 기반한 스케줄링 가능성 분석 방법을 제시하였다. 제시된 방법은 CPU 자원 이용율만을 계산하여 시스템 수행 중에 태스크의 수락제어를 할 수 있다.

참고문헌

- [1] H. H. Chu and K. Nahrstedt. CPU service classes for multimedia applications. In Proceedings of IEEE International Conference on Multimedia Computing and Systems, 1999.
- [2] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. IEEE/ACM Trans. on Networking, 1(3):344-357, 1993.
- [3] J. Bennett and H. Zhang. Wf2q: Worst-case fair weighted fair queueing. In Proceedings of IEEE INFOCOM, 1996.
- [4] I. Stoica, H. Abdel-Wahab, K. Jeffay, S. K. Baruah, J. E. Gehrke, and C. G. Plaxton. A proportional share resource allocation algorithm for real-time, time-shared systems. In Proceedings of IEEE Real-Time Systems Symposium, December 1996.
- [5] M. Ryu and S.-J. Kim. Dynamic and statistical deadline guarantees for a mixed set of periodic and aperiodic tasks. In Proceedings of International Conference on Real-Time and Embedded Computing Systems and Applications, 2003.
- [6] Z. Deng and J. W. Liu. Scheduling real-time applications in an open environment. In Proceedings of the Real-Time Systems Symposium, 1997.