

QoS 스케줄링 알고리즘을 위한 지분 분포의 효과 분석*

김아름⁰¹ 정진만² 홍지만¹

¹숭실대학교 컴퓨터학부

²서울대학교 컴퓨터공학부

k.irene2407@gmail.com, jmjung@os.snu.ac.kr, jiman@ssu.ac.kr

Impact of the Weight Distribution on the QoS scheduling algorithms

AhReum Kim⁰¹ Jinman Jung² Jiman Hong¹

¹School of Computing, Soongsil University

²School of Computer Science and Engineering, Seoul National University

요 약

응용 프로그램의 QoS를 보장하기 위하여 태스크들에게 지분을 부여하고 단위 시간동안 각 태스크에게 주어진 지분만큼의 CPU 자원을 할당하는 비례 지분 알고리즘들이 많이 연구되었다. 대부분 비례 지분 알고리즘들은 태스크 수, 지분 분포 등의 다양한 환경에 따라 동작 방식이 다르기 때문에 이러한 인자들은 시스템의 공정한 자원 할당에 많은 영향을 준다. 본 논문에서는 비례 지분 알고리즘을 동작 방식에 따라서 분류한 후, 태스크의 수와 더불어 지분의 편향성, 즉 지분 분포에 따라 QoS 공정성 결과를 분석한다. 분석 결과, 비례 지분 알고리즘에서는 태스크의 수 뿐만 아니라 지분의 분포 역시 QoS 공정성에 많은 영향을 주며, 각 환경에 따라 효율적인 최적의 알고리즘들이 존재함을 확인하였다. 제안된 분석 결과는 QoS를 지원하는 다양한 시스템에서 효율적인 비례 지분 알고리즘을 선택하여 시스템의 신뢰성을 향상시킬 수 있다.

1. 서 론

응용 프로그램의 QoS를 보장하기 위하여 태스크들에게 지분을 부여하고 해당 단위 시간동안 각 태스크에게 주어진 지분만큼의 CPU 자원을 할당하는 비례 지분 알고리즘들이 많이 연구되었다. 대부분의 비례 지분 알고리즘 연구에서는 QoS를 평가하기 위하여 실제 서비스 받은 시간과 예약된 자원 사용율의 차이를 나타내는 공정성오차(Fairness Error)를 사용한다. 하지만 이 공정성오차는 다양한 환경에 의하여 영향을 받는다. 특히 태스크의 수 뿐만 아니라 태스크들에게 할당되는 지분은 공정성의 결과를 다르게 하는 주요한 원인이 될 수 있다. 기존 연구에서는 태스크의 수에 따른 공정성의 변화에 초점을 맞췄다면, 본 논문에서는 지분의 분포가 공정성에 미치는 영향을 살펴본다.

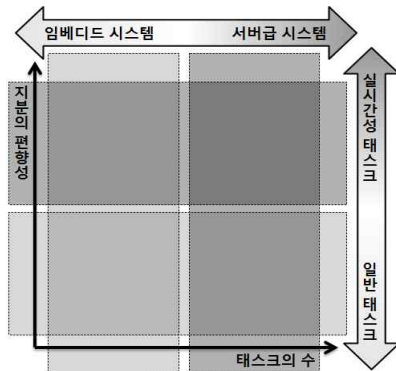


그림 1 지분 분포와 태스크의 수에 따른 다양한 시스템 환경

태스크에게 할당되는 지분은 시스템의 성격에 따라 다르게 분포될 수 있다. 그림 1은 태스크의 수와 지분 분포에 따른 시스템의 다양한 환경을 나타낸다. 태스크의 수가 작은 임베디드 시스템부터 태스크 수가 많고, 고신뢰성을 요구하는 서버급 시스템까지 다양하다.

본 논문의 구성은 다음과 같다. 2장에서는 비례 지분 스케줄링 알고리즘들을 동작 방식을 기반으로 분류하여 설명하고, 3장에서는 태스크 수와 지분의 분포에 따른 각 기법의 공정성과 오버헤드를 분석한다. 4장에서는 제안된 공정성 분석을 통해 각 환경에서 최적의 비례 지분 스케줄링 알고리즘을 선택하고 마지막으로 5장에서는 결론을 맺는다.

2. 비례 지분 스케줄링 알고리즘

각 태스크가 가지고 있는 지분에 근거하여 자원을 할당하는 비례 지분 알고리즘은 동작 방식에 따라 크게 라운드로빈(Round Robin) 기반 기법, 타임스탬프(Time Stamp) 기반 기법 및 혼합 기반 기법으로 나눌 수 있다.

먼저, 라운드로빈(Round Robin) 기반 기법은 일정한 시간 단위씩 태스크에게 CPU 자원을 할당해주는 알고리즘이며, 예로는 WRR(Weighted Round Robin)[1], DRR(Deficit Round Robin)[2], VTRR(Virtual-Time Round Robin)[3] 등이 있다. 다음으로 라운드로빈 기반 기법에 비해 공정성오차는 적지만 스케줄링 오버헤드가 $O(n)$ 으로 비교적 큰 값을 갖는 타임스탬프(Time Stamp) 기반 기법이 있다. 그 예로는 Stride[4], WF2Q(Worst-case Fair Weighted Fair Queueing)[5] 등이 있다. 마지막으로 기존 기법들의 장점을 최대한 취하여 공정성오차와 스케줄링 오버헤드를 최소화하고자 고안된 혼합 기반 기법이 있으며, 대표적인 예는 GR³(Group Ratio Round Robin)[6] 이다.

* 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2011-0027454).

본 논문에서는 라운드로빈 기반 기법의 DRR, VTRR 스케줄링 알고리즘, 타임스탬프 기반 기법의 Stride 스케줄링 알고리즘 그리고 대표적 혼합 기반 기법 스케줄링 알고리즘인 GR³에 대한 시뮬레이션을 통하여 지분 분포의 효과를 분석하도록 한다.

3. 지분 분포와 공정성 분석

기존 비례 지분 알고리즘들의 연구에서는 공정성오차(Fairness Error)를 식 (1)과 같이 정의하고, 비례 지분 스케줄링의 공정성 분석을 위해 사용하였다[3-6]. 자원에 대한 태스크 T_i 의 지분 할당량이 ω_i 이라 할 때, 이 자원에 대한

T_i 의 예약 비율은 $\frac{\omega_i}{\sum_j w_j}$ 이다. 임의의 시간 구간 $[t_1, t_2]$ 에

서 기대하는 예상 수행 시간은 $(t_2 - t_1) \frac{\omega_i}{\sum_j w_j}$ 이고, 이는 실제

T_i 가 임의의 시간 구간 $[t_1, t_2]$ 에서 수행된 시간과의 차이를 공정성오차라고 한다. 이러한 공정성오차 값이 0에 가까울수록 스케줄링 알고리즘의 정확도가 크다는 것을 의미한다.

$$E_i(t_1, t_2) = S_i(t_1, t_2) - (t_2 - t_1) \frac{\omega_i}{\sum_j w_j} \quad (1)$$

한편, 비례 지분 알고리즘은 태스크의 수가 증가할수록 공정성오차가 커지는 경향이 있으므로, 대부분의 비례 지분 스케줄링 알고리즘은 태스크 수가 증가하더라도 일정한 공정성을 보이면서 스케줄링 오버헤드를 최소화하는 것에 초점이 맞춰져 있다. 하지만 본 논문에서는 태스크 수와 지분 분포를 동시에 고려하여 공정성을 분석하였다. 지분 분포란, 전체 지분을 태스크들이 각자 어떻게 나눠가지고 있는지에 대한 정도를 뜻한다. 즉, 지분의 편향성이 크다는 것은 태스크들 간의 지분 분포가 고르지 않고 어느 한쪽으로 심하게 치우쳐져 있음을 의미한다.

3.1. 라운드로빈 기반 스케줄링 알고리즘 공정성 분석

DRR 스케줄링 알고리즘의 공정성오차 시뮬레이션 결과를 나타낸 그래프인 그림 2에서 보는 바와 같이 DRR은 지분 편향성이 커지면 커질수록 공정성오차가 크게 나타나며, 음의 공정성오차보다는 양의 공정성오차의 변화량이 훨씬 크다.

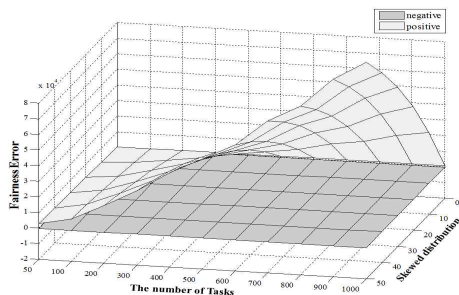


그림 2. DRR 스케줄링 알고리즘의 공정성오차

DRR은 라운드로빈으로 수행하며 결손(Decifit)이 1이 되었을 때 해당 태스크를 한번 더 수행시키므로 음의 공정성

은 보정되지만 양의 공정성은 태스크 수와 지분 분포가 커질수록 증가한다. 또한 그림 6에서 확인할 수 있듯이 DRR의 스케줄링 오버헤드 값은 항상 일정하다. 이러한 DRR의 특성은 DRR기법이 지분 분포의 편향성이 없는 환경에 적합하며 태스크 수의 증가량에 유연함을 의미한다.

다음으로 VTRR 스케줄링 알고리즘은 DRR 스케줄링 알고리즘과 마찬가지로 태스크의 수 보다는 지분 분포 편향성에 영향을 많이 받으나, 그림 3에서 보는 바와 같이 DRR과는 반대로 음의 공정성오차 값이 양의 공정성오차 값의 변화량보다 크게 나타난다. VTRR은 QVT라는 시스템 전체의 평균 서비스 시간과 비교해 한 태스크가 이 QVT를 넘지 않도록 하기 때문에 양의 공정성오차는 크게 증가하지 않는다. [3]

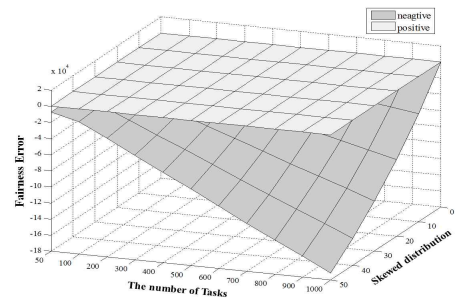


그림 3. VTRR 스케줄링 알고리즘의 공정성오차

3.2. 타임스탬프 기반 스케줄링 알고리즘 공정성 분석

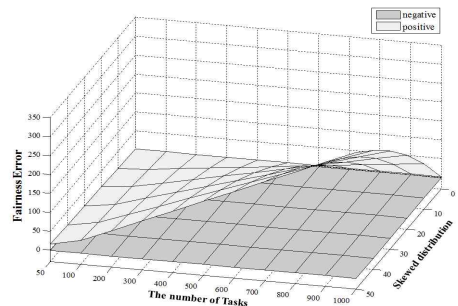


그림 4. Stride 스케줄링 알고리즘의 공정성오차

그림 4의 Stride 스케줄링 알고리즘은 양의 공정성 오차의 변화량이 음의 공정성 오차의 변화량보다 매우 크다는 점에서는 DRR 스케줄링 알고리즘과 유사하게 보인다. 그러나 Stride는 음·양의 공정성오차 모두 DRR보다 매우 낮으며 그림 6에서 볼 수 있듯이 스케줄링 오버헤드는 태스크의 수에 따라 $O(n)$ 으로 기하급수적으로 증가한다. 따라서 타임스탬프 기반의 기법은 태스크 수가 작은 환경에서 효율적임을 알 수 있다.

3.3. 혼합 기반 스케줄링 알고리즘 공정성 분석

혼합 기반의 GR³ 스케줄링 알고리즘은 그림 5에서 보는 것과 같이 기존의 알고리즘들에 비하여 매우 적은 공정성오차 값을 갖는다. 특히 지분 편향이 없는 경우, Stride의 공정성과 유사한 성능을 보이지만 스케줄링 오버헤드 관점에서는 태스크 수가 많아질수록 Stride에 비해 GR³가 효율적이다.

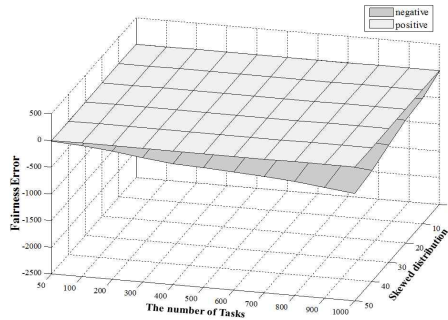


그림 5. GR3 스케줄링 알고리즘의 공정성오차

3.4. 각 기법의 스케줄링 오버헤드 분석

스케줄링 오버헤드 분석을 위해 각 기법을 한국전자통신연구원(ETRI)에서 개발 중인 AVOS라는 실시간 운영체제에서 구현하고 성능을 평가하였다. 스케줄링 오버헤드는 각 스케줄링 기법이 다음 태스크를 선정하는데 걸리는 시간을 측정한 결과로 단위 시간은 us를 사용한다.

그림 6은 지분 편향성이 50% 이상인 경우의 각 스케줄링 기법들의 스케줄링 오버헤드를 나타내는 그래프이다. 앞서 각 기법에 대한 공정성오차 시뮬레이션 결과 타임스탬프 기반의 Stride 스케줄링 알고리즘이 매우 높은 공정성을 보이는 것을 확인하였다. 그러나 그래프에서 보이는 바와 같이 Stride 스케줄링 알고리즘은 매우 높은 공정성을 보였으나 스케줄링 오버헤드가 최소 $O(n)$ 으로 매우 비효율적이고 느리게 동작함을 알 수 있다.

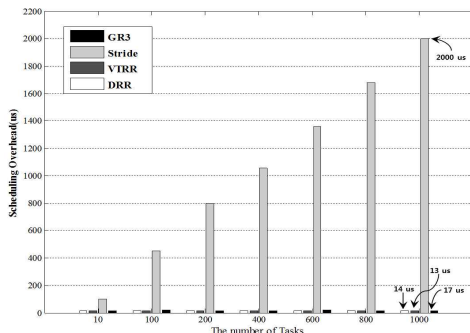


그림 6. 지분 편향성이 50% 이상인 경우의 스케줄링 오버헤드

반면, DRR, VTRR, GR³ 스케줄링 알고리즘의 경우, 지분의 편향성이나 태스크 수의 증가에 크게 영향을 받지 않고 비교적 일정한 오버헤드 값을 갖는다.

4. 공정성 분석을 통한 최적의 스케줄링 알고리즘 결정

앞서 실험한 공정성오차 및 스케줄링 오버헤드 분석 결과에 의하면, 지분 편향성에 영향을 받는 태스크들이 적당하게 존재하면서 태스크의 수가 적은 임베디드 시스템 환경에서는 Stride와 같은 타임스탬프 기반 기법을 사용하는 것이 유리하다.

한편, 태스크의 수가 많은 서버급 시스템 환경이면서 태스크들 간의 지분 편향성이 크지 않은 경우 즉, 일반 태스크가 주로 존재하는 시스템에서는 태스크의 수 보다는 지분 편향성에 대하여 공정성오차 값의 변화량이 크게 나타나는 DRR, VTRR과 같은 라운드로빈 기반 스케줄링 알고리즘이나 GR³ 같은 혼합 기반 스케줄링 알고리즘을 사용하는 것이 유리하다.

그러나 실험 결과, 태스크의 수와 지분 편향성에 대하여 모두 일정한 공정성오차를 보이는 스케줄링 알고리즘이 없으므로 지분 편향성이 큰 실시간성 태스크들이 많이 존재하는 서버급 시스템을 위한 비례 지분 스케줄링 알고리즘에 대한 연구가 필요할 것으로 판단된다.

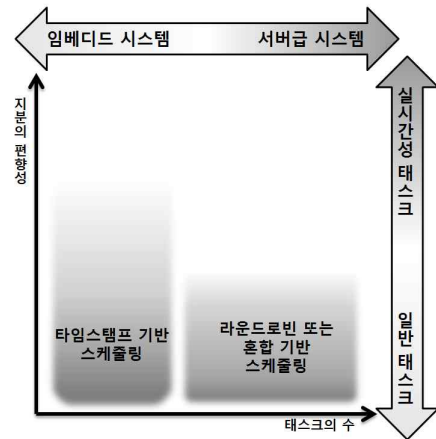


그림 7. 각 시스템 환경에 적합한 스케줄링 알고리즘

5. 향후 연구 및 결론

본 논문에서는 지분 분포가 QoS 보장을 위한 비례 지분 스케줄링 알고리즘에 미치는 영향을 분석하였다. 또한 분석한 결과를 토대로 각 시스템 환경에 적합한 최적의 스케줄링 알고리즘 결정에 대하여 연구하였다.

향후 연구로는 태스크의 수가 많고 지분의 편향성이 큰 경우, 즉 실시간성 태스크가 다수 존재하는 서버급 시스템에 적합한 새로운 스케줄링 알고리즘을 연구할 계획이다.

참고문헌

- [1] Stoica, I., Abdel-Wahaby H. and Jeffayz, K., "A Proportional Share Resource Allocation Algorithm for Real-Time", Time-Shared Systems, IEEE Real-Time Systems Symposium, 1996.
- [2] Shreedhar M. and Varghese, G., "Efficient Fair Queueing Using Deficit Round Robin", ACM SIGCOMM Computer Communication Review Vol. 25, No. 4, pp. 231, 1995.
- [3] Nieh, J., Vaill, C. and Zhong, H., "Virtual-Time Round-Robin: An $O(1)$ Proportional Share Scheduler", Proc. USENIX Annual Technical Conference, June. 2001.
- [4] Waldspurger, C. A., "Lottery and stride scheduling : flexible proportional-share resource management", PhD thesis, MIT, September 1995.
- [5] J. Bennett and H. Zhang, "WF2Q: Worst-case Fair Weighted Fair Queueing," in Proceedings of INFOCOM, 1996
- [6] B. Caprita, W. Chun Chan, J. Nieh, C.Stein, and H.Zheng, "Group Ratio Round-Robin : $O(1)$ Proportional Share Scheduling for Uniprocessor and Multiprocessor Systems", In Proceeding of the USENIX Annual Technical Conference, pp. 337-352, 2006