

멀티 혼합 중요도 시스템에서 태스크 마이그레이션의 스케줄가능성 분석

백전성^o, 강경태^{*}

한양대학교, 컴퓨터공학과^o

한양대학교, 컴퓨터공학과^{*}

e-mail: jsbaik@hanyang.ac.kr^o, ktkang@hanyang.ac.kr^{*}

Schedulability Analysis for Task Migration under Multiple Mixed-Criticality Systems

Jeanseong Baik^o, Kyungtae Kang^{*}

Dept. of Computer Science & Engineering, Hanyang University^o

Dept. of Computer Science & Engineering, Hanyang University^{*}

● 요약 ●

In this paper, we applied the migration technique to real-time tasks that have relatively low criticality but still important to be dropped by the mixed-criticality scheduling algorithms. The proposed drop and migrate algorithm analyzes the schedulability by calculating CPU utilization and response time of using task migration. We provide analysis to guarantee the deadline of LO-tasks, by transforming the response time equation specified with migration time. The transformed response time equation was able to analyze the migration schedulability. This algorithm can be used with various mixed-criticality schedulers as a supplementary method. We expect this algorithm will be used for scheduling LO-tasks such as communication task that requires safety guarantee especially in platooning and autonomous driving by utilizing the advantages of multiple node connectivities.

키워드: Task Migration, Mixed-Criticality Scheduling, Schedulability Analysis

I. Introduction

자율주행과 군집주행 등의 기술 도입에 앞서 고성능의 기능들을 상용화하기 위해서는, 수많은 기능이 수행되더라도 여전히 중요한 작업은 방해받지 않도록 하는 안전 보장이 필요하다. 특히 자율주행과 군집주행은 운전자나 책임자 안전 상황을 시스템에 맡기는 상황이므로 안전 보장에 대한 중요성은 기술력과 비례하여 증가하고 있다. 이러한 안전 중요도 시스템에는 모든 태스크들을 수행하지 못하는 임계 상황이 발생했을 때, 태스크의 중요성에 따라 중요도가 낮은 작업들을 스케줄링 대상에서 제외시키는 혼합 중요도 스케줄러를 사용한다. 하지만 스케줄러에 의해 제외당한 통신과 같은 낮은 중요도 태스크는 추적하는 차량들이 리드 차량에 운행을 위임하고 있는 군집주행에서 통신을 통해 안전을 보장해야 하는 경우가 존재한다.

본 논문에서는 혼합 중요도 스케줄링으로 인해 스케줄링 대상에서 제거된 중요도가 낮지만 여전히 스케줄 되어야 하는 태스크에 마이그레이션 기법을 적용하여 임무를 재개할 수 있도록 한다. 또한 마이그레이션으로 인해 발생하는 오버헤드를 CPU 이용률과 응답시간을 계산하여 스케줄 가능성을 검증할 수 있도록 해준다.

II. Preliminaries

1. Background

1.1 혼합 중요도 스케줄링

혼합 중요도 스케줄링에서는 각 작업이 다른 중요도를 가질 수 있음을 전제하며, 높은 중요도를 가진 태스크(HI-task)는 낮은 중요도를 가진 태스크(LO-task)보다 더 높은 보장을 해주는 알고리즘을 사용한다. 시스템 모드를 LO와 HI로 정의하여, 시스템이 HI-mode로 전환되면 HI-task들의 최악의 실행시간(WCET)은 이전 모드와 같거나 많은 WCET을 보장함과 동시에 LO-task들의 일부 혹은 전부를 버리는 Task Dropping 알고리즘을 사용한다[1,2].

1.2 스케줄 가능성 테스트

시스템을 HI-mode로 전환하면 LO-task들을 버리는 과정을 거치게 되는데 두 가지 테스트로 스케줄 가능성을 검증한다.

- *CPU utilization test:* CPU 이용률 초과 여부를 계산함으로써 모든 작업들의 수행여부 가능성을 판단하며, 이에 따라 중요도가 낮은 LO-task를 스케줄 대상에서 제외시킨다.
- *Response time test:* 개별 태스크에 대한 응답시간을 계산하여 마감기한과 비교함으로써 개별 태스크의 스케줄 가능성 여부를 검증한다.

III. Drop and Migrate Algorithm

본 알고리즘은 Task Dropping 알고리즘에 의해 제거된 LO-task 중에서도 마감기한을 반드시 지켜야 하는 태스크를 마이그레이션하여 임무 지속 가능하도록 한다. 이를 설명하기 위해 Fig. 1과 같은 태스크 예제를 사용한다. $C_i(LO)$ 과 $C_i(HI)$ 는 시스템이 각각 LO-mode와 HI-mode 상태일 때 WCET을 의미한다.

τ_i	L	$C_i(LO)$	$C_i(HI)$	D_i	T_i
τ_1	HI	1	2	4	4
τ_2	HI	1	1	5	5
τ_3	LO	5	-	10	10

Fig. 1. Mixed-criticality 태스크 예제

시스템의 초기 상태(LO-mode)에 대하여 CPU 이용률을 계산하면 다음과 같이 1 이하의 값이 나오며, 스케줄이 가능한 상태에서 스케줄링을 시작한다.

$$U_L = \sum_{\forall \tau_i \in \tau} \frac{C_i(LO)}{T_i} = 0.95 \leq 1$$

시스템이 HI-mode로 전환되면, 해당 모드에서의 각 WCET으로 CPU 이용률을 고려해야 한다. HI-task들의 WCET를 업데이트 하여 EDF(최단 마감기한 우선) 스케줄러를 사용하면 Fig. 2 과 같이 스케줄이 불가능한 태스크가 나타난다.

$$U_H = \sum_{\forall \tau_i \in \tau_H} \frac{C_i(HI)}{T_i} + \sum_{\forall \tau_i \in \tau_L} \frac{C_i(LO)}{T_i} = 1.2 > 1$$

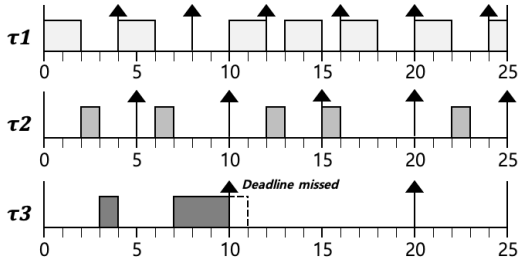


Fig. 2. HI-mode의 스케줄링 타임라인

본 알고리즘은 이를 해결하기 위하여 태스크 마이그레이션을 사용한 HI-task를 방해하지 않으면서 LO-task의 마감기한을 놓치지 않는 방법을 제안한다. 시스템이 HI-mode로 전환되면서 제거된 태스크를 다른 노드에 마이그레이션 시킨다. 마이그레이션을 수행한 이후에도 이전 노드에서의 마감기한을 보장하는 것이 알고리즘의 목표이다. 단일 프로세서에서 어떤 태스크에 대하여 다음과 같이 응답시간(R)을 계산함으로써 스케줄 가능성을 검증한다[1]. 여기서 LO-task인 τ_i 는 HI-task들에 의해 선점당할 것이기 때문에 모든 HI-task들의 실행시간을 고려하면서도 마감기한을 만족시키는지 검증해야 함을 나타낸다.

$$R_i = C_i + \sum_{\tau_j \in \tau_H} \left\lceil \frac{R_j}{T_j} \right\rceil C_j(HI) \leq D_i$$

응답시간 계산식을 마이그레이션을 고려한 변환된 응답시간식으로 업데이트한다. 제거된 태스크의 WCET는 마이그레이션 시간(M_i)으로 대체하고 마감기한은 다음 노드에서 WCET을 보장하기 위해 차감된 마감기한으로 대체한다.

$$R_i^{pre} = M_i + \sum_{\tau_j \in \tau_H} \left\lceil \frac{R_j}{T_j} \right\rceil C_j(HI) \leq D_i - C_i$$

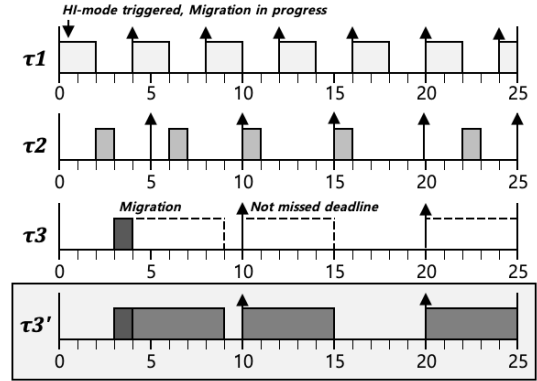


Fig. 3. Drop & Migrate 스케줄링 타임라인

그림 Fig. 3은 Fig. 1에 주어진 태스크 예제에 대하여 모드가 변환되는 시간을 0.5, 마이그레이션 수행 시간을 1이라고 했을 때, HI-mode에서 스케줄이 불가능했던 τ_3 를 다른 노드에 마이그레이션 하여 τ_3' 로써 태스크를 수행하여 기존 τ_3 에서 지정한 마감기한을 만족시키고 있다. 보정된 응답시간이 마감기한을 만족하는지 계산해 본 결과, 이를 만족하는 응답시간이 있음을 확인하였다.

$$R_3^{pre} = 1 + \left\lceil \frac{R_3}{4} \right\rceil 2 + \left\lceil \frac{R_3}{5} \right\rceil 1 = 4 \leq 5$$

IV. Conclusions

본 논문에서는 Drop and Migrate 알고리즘을 사용하여 중요한 역할을 하는 LO-task의 마감기한을 보장하기 위한 기법을 제공하였다. 또한 태스크 마이그레이션의 스케줄 가능성을 보장하기 위하여 보정된 응답시간을 사용하였고, 멀티 노드 환경에서 유용하게 적용될 만한 스케줄 가능성을 제공하였다. 군집주행 또는 자율주행 시나리오에서 중요도가 낮지만 안전 보장을 필요로 하는 통신 태스크 등에 대하여 멀티 노드의 이점을 활용한 안전 보장 기법으로 사용될 것으로 기대된다.

REFERENCES

- [1] S. Baruah, A. Burns, R. Davis, "Response-Time Analysis for Mixed Criticality Systems," in *proc. IEEE Real-Time Systems Symposium (RTSS)*, Vienna, Austria, May 2011.
- [2] J. Lee, H. Chwa, L. Phan, I. Shin, I. Lee, "MC-ADAPT: Adaptive Task Dropping in Mixed-Criticality Scheduling," in *proc. International Conference on Embedded Software (EMSOFT)*, Seoul, South Korea, Oct. 2017.