

## 자원 재배치를 통한 처리율 기반 실시간 스케줄링 기법

정해건<sup>0</sup>, 안제민\*, 강경태\*

<sup>0</sup>한양대학교 컴퓨터공학과

e-mail: haegeonj@hanyang.ac.kr<sup>0</sup>, {ahnjemin, kt kang}@hanyang.ac.kr\*

## Throughput-aware real-time scheduling based on resource relocation

Haegon Jeong<sup>0</sup>, Jemin Ahn\*, Kyungtae Kang\*

<sup>0</sup>Dept. of Computer Science & Engineering, Hanyang University

### ● 요약 ●

본 논문에서는 개발자가 범용 리눅스 운영체제에서 유연하게 시스템의 실시간성 향상을 이뤄낼 수 있도록 자원재배치 모듈을 통한 스케줄링 기법을 제안한다. 제안하는 방법은 범용 리눅스 운영체제 기반이기 때문에 기존의 Cyber-Physical System (CPS)에 쉽게 적용이 가능하며, 실시간성을 필요로 하는 새로운 CPS 프로젝트의 개발 속도 향상시킬 것으로 예상된다. 또한, 실험결과 자원 재배치를 진행한 애플리케이션의 평균 처리율이 20%이상 증가한 것을 알 수 있다. 추후 메모리, I/O장치 등 다양한 자원에 대하여 모니터링을 진행하고 자원을 재배치 할 수 있는 플랫폼으로 확장할 예정이다.

**키워드:** 스케줄링(scheduling), 자원재배치(resource relocation), 실시간성(real-time)

### I. Introduction

Business Insider는 2016년부터 2021년까지 IoT 기기가 250억 개까지 늘어날 것이며, IoT 관련 사업에 총 5조 달러의 투자가 이루어질 것으로 예측했다.[1] 이러한 예측은 앞으로 1인당 사용하게 되는 IoT 기기 수가 증가될 것임을 암시한다. IoT 기기수의 증가는 단일 AP가 더 많은 수의 IoT 기기들을 감당해야 되며 이는 더 많은 작업을 처리해야함을 의미한다. AP는 일종의 Cyber-Physical System (CPS) 으로 한정된 자원을 이용하여 연결된 노드들로부터 요청된 작업을 처리하게 된다. 늘어나는 IoT 기기수와 요청되는 작업량을 감당하기 위해 AP를 비롯한 여러 CPS노드들은 지금보다 더 효율적으로 자원을 관리해야할 필요가 있다.

본 연구팀은 여러 서비스가 돌아가는 CPS 노드에서 일반적으로 사용되는 CFS 방식이 아닌 작업 처리율 기반으로 스케줄링 처리를 하면 시스템의 자원을 최적으로 사용할 수 있을 것으로 예상 하였다. 본 논문에서는 개발자가 범용 리눅스 운영체제에서 유연하게 시스템의 처리율 기반 실시간성 향상을 이뤄낼 수 있도록 자원재배치 모듈을 통한 스케줄링 기법을 제안한다. 제안하는 방법은 범용 리눅스 운영체제 기반이기 때문에 기존의 CPS에 쉽게 적용이 가능하며, 실시간성을 필요로 하는 새로운 CPS 프로젝트의 개발 속도 향상시킬 것으로 예상된다. 또한, 실험결과 자원 재배치를 진행한 애플리케이션의 평균 처리율이 20%이상 증가한 것을 알 수 있다.

### II. The Proposed Scheme

일반적인 리눅스 환경에서는 Completely Fair Scheduler (CFS)로 태스크를 관리한다. 이때 Figure 1과 같이 Red-black tree를 이용하여 모든 태스크가 공평하게 CPU자원을 사용하게 한다. 이는 프로세스 우선순위나 sleep 함수를 통해 자발적으로 CPU를 반환하는 경우를 고려하지 않는다면, CFS가 관리하는 모든 태스크는 동일한 시간동안 CPU를 점유하고 반환한다. 하지만 CFS는 우선순위나 nice값이 달라져도 처리율에는 큰 차이가 없다. 따라서 본 연구팀은 실시간성을 확보하기 위한 커널 패치를 진행했다. 커널 패치의 결과 Figure 1의 그림과 같이 CFS와 RT 두 가지의 스케줄링 정책을 사용할 수 있다. RT queue는 태스크가 먼저 수행되고 여분의 시간에 CFS가 수행되도록 하여 실시간성을 필요로 하는 애플리케이션의 CPU 자원 점유를 보장한다.

기존의 실시간 리눅스와 다른 점은 자원 재배치 모듈을 통하여 동적으로 애플리케이션을 관리 할 수 있다는 점이다. Figure 1은 본 연구팀에서 설계 및 구현한 재배치 모듈의 동작과정을 나타낸 그림이다. 재배치 모듈은 캐릭터 디바이스 드라이버로 작성되어 있다. 모듈에 “boost 1000”을 쓰게 되면 PID 1000 태스크를 RT queue로 옮겨 CPU 자원의 할당을 더 많이 하게 된다. 이 외에도 모니터링 데이터를 기반으로 작업 처리율이 사용자가 정한 임계값을 넘어가면 자동으로 CPS에서 관리되던 task가 RT queue로 위치를 변경하여 애플리케이션 처리율을 높일 수 있다.

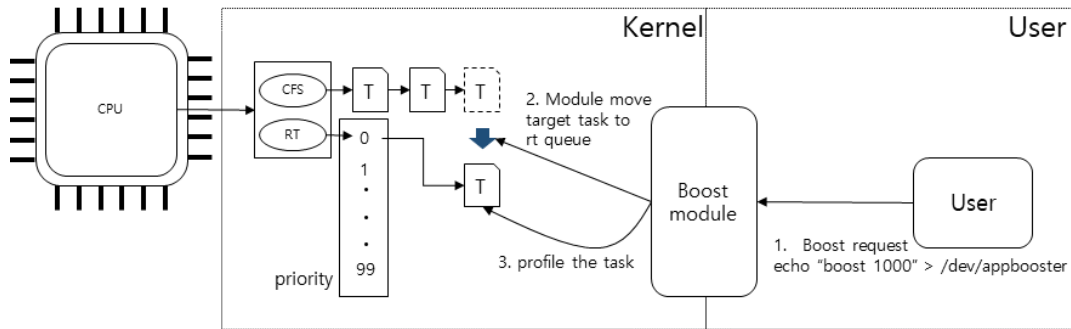


Fig. 1. Relocation Module Working Process

### III. Experiment

제공하는 재배치 모듈의 정상 작동을 확인하기 위하여 네트워크 애플리케이션을 가정한 실험 시나리오를 설계하였다.

1. APP1과 APP2 두 가지 애플리케이션이 존재하며 사용자에게 특정 서비스를 제공한다.
2. APP2에 사용자가 몰려 APP2에 더 많은 자원이 필요하게 되었다.
3. 자원 재배치 모듈을 통하여 APP2에 CPU자원을 더 많이 할당하여 같은 시간동안 더 많은 일을 처리할 수 있는지 확인한다.

Figure 2는 재배치 모듈을 통하여 앞서 설계한 실험의 결과로, 초당 CPU자원의 사용량을 그래프로 나타낸 것이다. 그래프에서 66초 적색선은 APP2에 자원 재배치를 시작한 지점이다. 자원 재배치 이전 두 애플리케이션의 CPU 점유율은 거의 비슷하나 재배치가 이루어진 후 APP2에 더 많은 CPU 자원이 할당된 것을 볼 수 있다. 이를 통해 제공하는 모듈이 자원을 적절히 배치하여, 평균적으로 단위시간당 처리율이 20%정도 상승했음을 볼 수 있다.

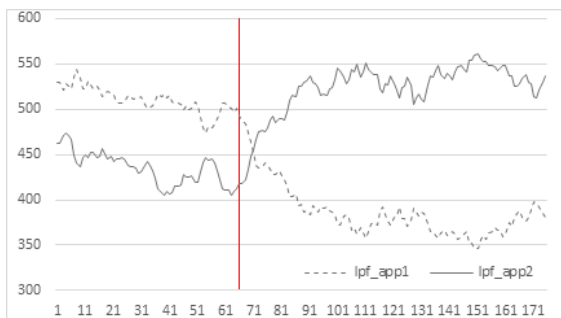


Fig. 2. CPU Execution per Second

### IV. Conclusions

본 연구팀은 커널패치 및 재배치 모듈을 통한 처리율 기반 실시간 스케줄링에 관하여 연구 및 개발을 진행하였다. 이를 통하여 높은 처리율을 필요로 하는 애플리케이션을 대상으로 실시간성을 부여하여 빠른 일처리가 가능하도록 플랫폼을 제공할 수 있었다. 추후 메모리, I/O장치 등 다양한 자원에 대하여 모니터링을 진행하고 자원을 재배치할 수 있는 플랫폼으로 확장할 예정이다.

### REFERENCES

- [1] John Greenough, July 18, 2016, business insider Inc. "How the 'Internet of Things' will impact consumers, businesses, and governments in 2016 and beyond," <http://www.businessinsider.com/how-the-internet-of-things-market-will-grow-2014-10>.