

Xen-ARM에서 물리시간 기반 실시간 운영체제 스케줄링

정재우^o 유시환 유혁
고려대학교

jwjang@os.korea.ac.kr, shyoo@os.korea.ac.kr, hxy@os.korea.ac.kr

Scheduling RT-guest OS with Physical Time in Xen-ARM

Jae Woo Jung^o See Hwan Yoo Chuck Yoo
Korean University

가상화 시스템은 초기의 네트워크 서버와 같은 서버 환경에서 벗어나 임베디드나 클라우드 컴퓨팅과 같은 다양한 시스템에서 활용되고 있다. 가상화는 초기의 다중 사용자를 하나의 물리 머신에서 수용하려는 목적에서 넘어서 신뢰성, 확장성, 유연성을 제공하는 수단으로 활용되고 있으며, 특히 임베디드 시스템에서는 다중 운영체제를 활용한 다양한 사용자 경험을 제공하기도 하며, 사용자 개인정보를 신뢰성있게 다룰 수 있는 근본적인 구조로서 활용될 수 있다.

하지만, 실제 임베디드 시스템의 가상화는 실시간성에 대한 요구사항을 충분히 만족하지 못하는 이유로 널리 활용되지 못하고 있다. 특히, 스마트폰이나 차량용 소프트웨어의 경우, 실시간 요구사항은 사용자의 만족도를 크게 떨어뜨리거나 이용자의 안전과 직접적으로 연관되어있어 매우 엄격하다고 할 수 있다. 따라서 임베디드 시스템 가상화 환경에서는 기존 가상화 환경에서 다루어져 왔던 범용 운영체제들과 달리 높은 실시간성을 가지는 운영체제에 대한 필요성이 있다. 이 논문에서는 임베디드 시스템의 가상화에 있어 핵심적인 요구사항인 실시간성과 관련된 문제를 다룬다.

특히, 이 논문에서는 게스트 운영체제 내부의 스케줄링과 관련한 문제를 다루고 있다. 게스트 운영체제의 스케줄링은 일반적으로 가상시간을 기준으로 이루어진다. 대개의 범용 운영체제들이 사용하는 공평성을 강조하는 스케줄러(fair scheduler)들의 경우, 실제 CPU 사용시간을 기준으로 스케줄링을 수행하기 때문에 가상머신이 실제 실행한 시간에만 진행되는 가상시간을 기준으로 스케줄링 하도록 한다.

하지만, 실시간 운영체제들은 프로세스의 사용시간에 대한 정확한 측정을 하는 것보다는 프로세스의 우선순위에 기반한 스케줄링을 수행한다. 이 때 우선 순위는 CPU 사용 시간이 아닌 물리적 임계시간이나 태스크에 따라 고정적으로 주어진다. 따라서, 가상시간을 기반으로 동작하는 경우, 심각한 문제가 발생한다.

특히, 물리 머신 위에서 사용하는 물리시간과 달리 가상시간은 가상머신이 실행될 때만 진행되기 때문에, 물리시간과 가상시간은 차이가 생기게 되며, 시간이 지날수록 가상시간과 물리시간의 차이는 점점 커지게 되어, 결국 가상시간을 기반으로 동작하는 실시간 게스트 운영체제는 물리적 임계시간 내에 작업을 마칠 수 없게 된다. 본 논문에서는 이러한 실시간 시스템의 실패를 정형적 분석과 실제 예를 통해 보여준다.

그림 1은 실제 Xen-arm 가상머신 모니터 상에 실시간 게스트 운영체제를 동작하여, 물리 임계시간 내에 실행을 완료할 수 없음을 보인다.

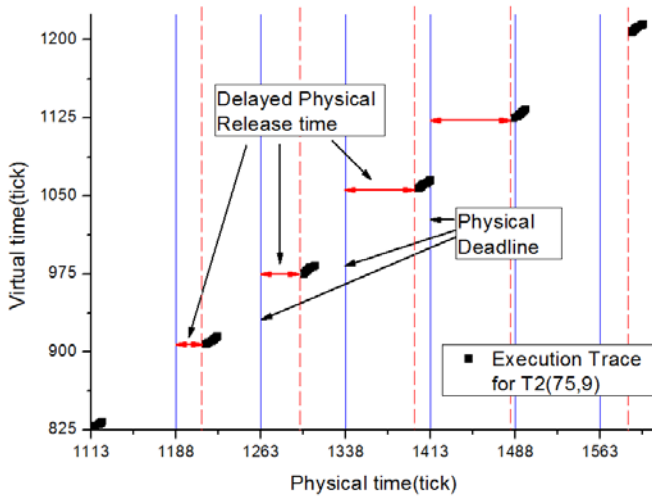


그림 1 가상시간을 이용한 실시간 게스트 운영체제의 스케줄링

그림 2는 그림 1을 물리 시간 1488에서 조금 더 자세히 본 그림이다. 1488에서 delayed release time이 증가하면서 주기의 시작시간에서 실행시간을 더한 값이 주기의 물리적 임계시간을 넘어서는 경우가 보인다. 따라서 그림 3은 1488에서 임계시간을 넘어 실행이 끝나는 것을 보여주고 있다.

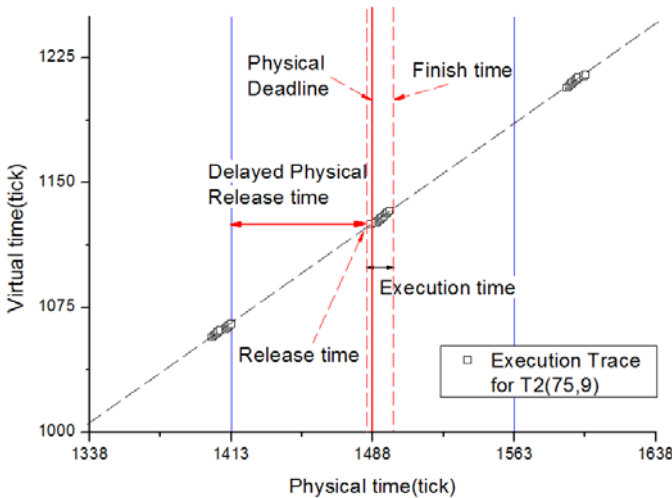


그림 2 가상시간 기반 실시간 스케줄링의 실패

논문의 구성은 다음과 같다. 2장에서는 가상머신의 스케줄링과 가상시간에 관련한 여러 가지 연구들을 소개하며, 3장에서는 가상시간을 기반으로 한 실시간 게스트 운영체제의 실패를 보여준다. 4장에서는 본 연구 내용의 결론을 맺고 향후 연구 방향에 대해 제시한다.