Karan Sarkar

Intro to Enterprise Computing

Paper: Supporting Time-Sensitive Applications on a CommodityOS

Authors: Ashvin Goel,Luca Abeni,Charles Krasic,Jim Snow, Jonathan Walpole

Summary:

More and more often, commodity operating systems are being used for time-sensitive applications. These applications require low latency responses from various system level features specifically resource allocation. In order to have fast resource allocation, one needs high precision timing facility, well-designed pre-emptible kernel and appropriate scheduling techniques. Paper provides three techiques: firm timers - an efficient high-resolution timer mechanism, fine-grained kernel preemptibility and priority and reservation-based CPU scheduling mechanisms

Strengths:

1. In the past, commodity operating systems have implemented timing mechanisms using periodic interrupts. The total latency can be reduced by increasing the frequency of interrupts. However, then the sheer number of interrupts can also increase latency. This paper proposes to only create interrupts as an as needed basis. This improves latency.
2. Traditional commodity kernels disable preemption for the entire period of time when a thread is in the kernel. However, this limits when preemptions can occur and results in a high preemption latency. This paper suggests to release and require spinlocks at designed places within the kernel. This improves latency.
3. In Proportion-Period CPU scheduling, every task is given a certain proportion of the CPU power. This ensures that no processes starve. The paper uses a form of priority based CPU scheduling where high priority tasks get more CPU power. This reduces latency.

Weaknesses:

1. Unlike periodic timers, one shot timers have to be continuously reprogrammed for each timer event generating some overhead.
2. Priority based CPU scheduling may be greater affected by having high priority programs become unresponsive.

Reflections.

I think the paper is very impressive. It lays out a number of ways to reduce latency. It drastically reduces the worst case scenario compared to Linux. However, it may be a bit worse on the average case.