Low Latency Real-Time Computing on Multiprocessor Systems Running Standard Linux

Mr. Dimitri Sivanich
SGI
Email: sivanich@sgi.com

Overview

The growing acceptance of Linux, along with the recognition by many of the importance of real-time applications, has lead to numerous improvements to Linux that reduce latencies and overhead experienced by application threads.

This presentation will describe and demonstrate how, through appropriate configuration and application structure, a standard Linux distribution can be used to run applications requiring very low latency response (10's of usec), high degrees of determinism, and high cpu availability on multiprocessor systems.

The methodology described in this presentation will not rely on any sort of preemption model. Instead it will focus on eliminating or avoiding sources of interference from real-time threads.

For many applications a non-preemptive thread isolation model such as this can yield very good results.

Configuration for Low Latency

Proper system configuration is essential to achieving very low response times (10's of usec) and a high degree of determinism on multiprocessor systems running standard distributions. Configuration can be accomplished through processor isolation (or processor shielding as it's sometimes known) and interrupt redirection.

Processor isolation refers to removal of load balancing effects in order to prevent unwanted threads from moving onto a specific processor, and to prevent the interference caused by other processors performing load balancing.

Interrupt redirection is used to eliminate the effect of interrupts that are non-critical to real-time threads, while directing time critical interrupts to processors that are configured to deliver a more deterministic response.

SGI uses this configuration methodology as the foundation of it's React real-time product line. React uses no additional kernel patches beyond what's in a standard Linux distribution. With this low latency foundation as a base, other real-time features can be added to assist developers in further optimizing their applications.

Application Structure

The use of features such as real-time priority and scheduling, memory locking, and timer interrupt suspension will be discussed, along with other strategies for minimizing interference. SGI's use of cpusets as a means to more easily facilitate thread placement and better enforce proper application behavior will also be covered.

Demonstration of Low Interrupt Latency

This presentation will demonstrate the benefits of proper configuration using interrupt latency and jitter measurements[1] to compare unconfigured and configured multiprocessor systems. A system running a preemptive model (the community linux preemption kernel) will also be compared. The data is meant to support the notion that low latency can be achieved on specially configured portions of an otherwise heavily loaded system.

1 Measurements done on an SGI Altix running a Novell SLES10 distribution and SGI React configuration. Your mileage may vary.