CST 8703 Lab 1 - First Real-Time Kernel Jitters

Kyle Chisholm

2022-05-01

Real-Time Systems and Embedded Programming

Submission Deadline: May 01, 2022

Realtime profiling tools will be used to characterize the latency and time variation (jitter) of the system, and demonstrate the utility of a preemptible kernel in a hard realtime operating system. A cyclic program with a fixed step timer will be created and timing analyzed.

Background

Hard real-time systems require operations to meet specified timing deadlines and a missed deadline is considered a system failure. For example, a digital control system such as a robotic manipulator may require a fixed interval timer. Filtering and closed-loop control systems are tuned based on a loop duration, also referred to as a "step size". If this duration deviates from a specified tolerance, the control system may go unstable and lead to catastrophic failure. If the control loop itself is robust and won't go unstable with variation in loop step size, a deviation from a robot trajectory, even over a short duration, may cause damage to surroundings or an individual interacting with the robot.

Linux was not initially designed as a real-time operating system and meeting strict deadlines was less of a priority compared to providing a solid desktop computing user experience. The Linux real-time project was established to incorporate changes into the mainline Linux kernel that allow the scheduler to be preempted by real-time processes. These changes (patches) can be enabled when configuring the Linux kernel. The following table provides links from the Linux Foundation with resources specific to the PREEMPT_RT kernel patch development:

Link	Description
Technical basics: Important aspects for real time	A must-read regarding basics of preemption and real-time Linux
HOWTO: RTOS and RT Applications	Getting started with real-time application development for Linux

Testing the embedded platform and the timing of your application is a necessary step in evaluating the capabilities of the software to meet hard timing deadlines. Hardware is typically supported by commercial RTOSes with Board Support Packages (BSPs) but with most free Linux distributions there are no officially supported real-time platforms. The Open Source Automation Development

Lab (OSADL) runs a quality assessment "farm", testing Linux configured with PREEMPT_RT on many platforms and distributions. Some of their boards under test are Raspberry Pis with latency plots and kernel configuration scripts made available online. The following plot is a sample latency histogram of a Raspberry Pi on the test rack:

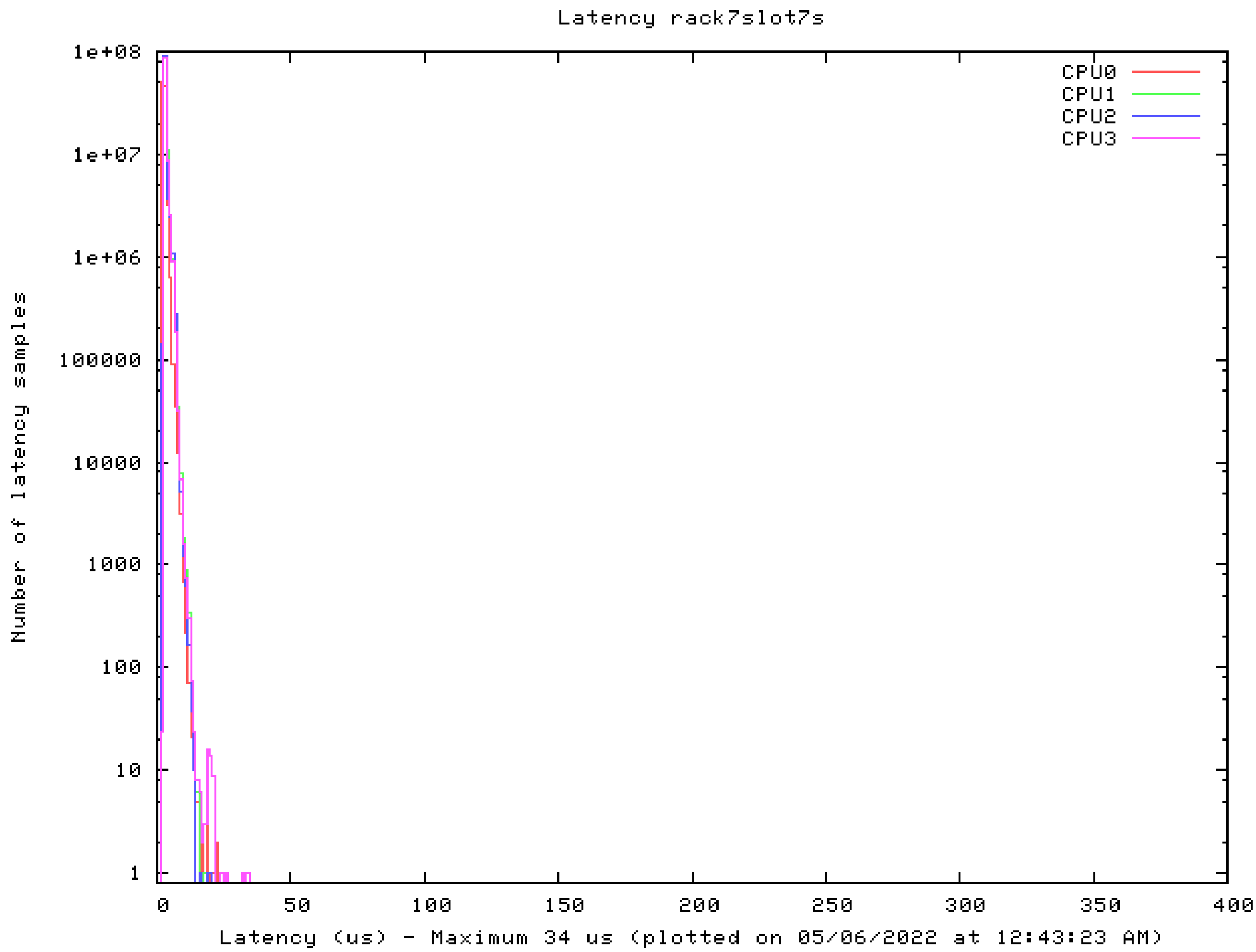

Figure 1: Latency plot of Raspeberry Pi on OSADL test rack

Latency is a key metric of realtime performance. It is the delay after a timer is triggered before processing begins. For a timed loop, the latency must be low enough to allow sufficient processing time such that it can complete before the next timer deadline occurs. Debugging errors caused by spikes in latency can be difficult to track down since source of latency can come from the OS performing other tasks, and may not be easily reproducible.

The program Cyclictest is a very useful tool for testing real-time performance (specifically timing latencies). In this lab, system latencies will be tested with `cyclictest`. Jitter is another metric commonly used in reference to network communications.

Materials

Methods

```
sudo apt install -y \
    python3-numpy \
    python3-matplotlib \
    rt-tests
```

Deliverables

Run

```
./scripts/run_submission.sh
```

Expected contents out output.txt:

References