Virginia Tech ❖ Bradley Department of Electrical and Computer Engineering

ECE 4984 / 5984 Linux Kernel Programming
Spring 2017

Regular vs real-time Linux latency in virtualized environment

1 Introduction

The *latency* of a real-time system measures the amount of time the system takes to respond to a stimulus. More precisely, it's the amount of time between the reception by the system of an interrupt, and the start of the user level code supposed to be triggered by that interrupt. One of the main characteristics of real-time systems is their need for a quick reaction time. Thus, the latency should be as small as possible. Moreover, a second crucial requirement is the need for a deterministic behavior: the latency should be stable among several invocations of the stimulus.

The objective of this project is to compare and study the latency of a real-time-enabled Linux kernel (Concurrent *RedHawk* [5]) with the latency of a regular Linux kernel (CentOS), in a virtualized environment using the KVM hypervisor.

The following concepts from the course will be considered in this project:

- Interrupt management;
- Scheduling;
- Performance evaluation.

This is a group project and it should be done by groups of 2 to 3 students. Please use canvas to register your group.

2 Project description

It is needed to deploy the test environment (host operating system, hypervisor, and guest virtual machines) on your personal machine and write a latency measurement infrastructure. Then, define and run the tests, gathering the measurement data at different levels: (1) in the host and (2) in a guest. Finally, it is needed to investigate the difference that should be observed between the real-time enabled host/guests and the regular ones.

2.1 Project steps

1. **Install the host environment.** This project being primarily focused on performance evaluation, host operating systems (CentOS and RedHawk) must be installed on bare metal, e.g. on your personal machine. That environment cannot be packed inside a virtual machine (for example with VirtualBox) because of the performance impact that would be added by this additional layer of virtualization. You will get necessary OS and their update ISO images here:

   `http://bit.ly/2mSmQKa`

   Necessary installation instructions for RedHawk in *Redhawk Release Notes* [3].

2. **Install the guest environment.** Guest operating systems should be setup by KVM virtualization infrastructure. We are interested in the following host/guest OS types combinations:

   - CentOS host, CentOS guest;
   - RedHawk host, RedHawk guest;
   - CentOS host, RedHawk guest;
   - RedHawk host, CentOS guest.

The two first combinations will allow us to understand the real-time enabled kernel impact on the latency of applications in virtualized environments. Concerning the third setup, it will allow investigating the performance impact of a non-real-time host on a real-time guest. Finally, the fourth combination will help investigating potential benefits of having a real-time enabled host.

Instruction on how to create a guest using KVM in CentOS are available in [2]. These instruction should also be valid for RedHawk;

3. **Write and run the latency measurement framework.** The ***cyclictest*** Linux tool should be used to measure the latency. It comes pre-installed with CentOS as well as RedHawk. Extensive documentation about how *cyclictest* works and its usage is available in [6, 4].

   Cyclictest works as follows: it schedules a timer to expire at a time T in the future. When cyclictest wakes up due to the timer expiration, it computes the latency as the difference between the actual current time and T:

   $latency = (now - T)$

   Cyclictest can be configured to run for several iterations. You should measure the average latencies for different host/guest OS type combinations, both in host and guest VM.

4. **Investigate the results.** To investigate the observed latency difference, a suggested process is as follows:

   (a) Investigate and identify the sequence of steps executed from the actual expiration of the timer (determined in the system timer interrupt handler) and the moment cyclictest is scheduled back to run on a CPU;

   (b) For each of the important steps in that sequence, get a timestamp of the actual moment of execution of that step. To get a such timestamps without the need to recompile the entire kernel (i.e. in a module), the use of kernel probes (Jprobes) [1] is suggested;

   (c) Compare the difference between consecutive timestamps in different combination of host/guest OS to identify which steps introduces the largest differences;

   (d) Comparison should be made on average timing values of multiple iteration

   Scripts can be written to automate the measurement process to get time difference from gathered timestamps;

2.2 Project report

A report of 5 pages minimum should be produced as one of the results to be handed by the project deadline. It should contain:

1. A detailed description of the executed experiments concerning latency measurement with cyclictest. That section should also document the student personal machine hardware specifications (CPU model, amount of RAM) as well as the host / guest configuration (number of virtual CPUs and dedicated amount of RAM);
2. Measurement data presented in the form of graphs, with associated text commenting on notable results, in particular concerning the latency difference between real-time and regular environments;
3. The result of the investigation step: description of the investigation process, of the tracing infrastructure and potential additional software developed, presentation and interpretation of the obtained results.

3 Results to be handed - Deadline: 2017-04-07 11:59PM

The following is expected to be handed by 2017-04-07:

1. Project report in PDF version;
2. Sources of all developed software: latency measurement infrastructure, and tracing infrastructure used for the investigation step.

All of this should be contained in a *tarball*, with the following format: `group<group ID>.project5.tar.gz`. For example: `group1.project5.tar.gz`

References

[1] Kernel probes documentation. http://lxr.free-electrons.com/source/Documentation/kprobes.txt.

[2] Kvm and centos-6. https://wiki.centos.org/HowTos/KVM.

[3] Redhawk release notes. http://redhawk.ccur.com/docs/root/1Linux/1RedHawk/7.2/0898003-7.2.pdf.

[4] Finding realtime linux kernel latencies (for rt linux kernels >= 2.6.33), 2016. http://people.redhat.com/williams/latency-howto/rt-latency-howto.txt.

[5] Redhawk - concurrent, 2016. https://www.concurrent.com/linux/products/redhawk/.

[6] Frank Rowand. Using and understanding the real-time cyclictest benchmark, 2013. http://events.linuxfoundation.org/sites/events/files/slides/cyclictest.pdf.