A Concurrent Real-Time White Paper

2881 Gateway Drive
Pompano Beach, FL 33069
(954) 974-1700
www.concurrent-rt.com

Real-Time Performance of KVM using a RedHawk Linux PRT Host

By: Robert Davis

Software Engineer

March 2019

Abstract

This paper is for RedHawk Linux users who wish to run multiple copies of the RedHawk RTOS on a single server using Kernel Virtual Machine (KVM) while maintaining real-time performance. The RedHawk Linux PRT kernel combines traditional RedHawk real-time features with the open source community's PREEMPT_RT real-time patch. KVM with RedHawk Linux PRT as the host OS provides the best possible virtual real-time environment. This solution is known collectively as RedHawk KVM-RT

The real-time determinism of a RedHawk PRT host and three RedHawk Linux Virtual Machines (VMs) are evaluated while also running a copy of Windows 10 as an additional VM. Real-time performance is measured over a 24-hour period on all RedHawk instances while the host and guests are subjected to CPU, memory and I/O stresses.

A familiarity with RedHawk Linux PRT and KVM administration is assumed in this discussion.

Introduction

KVM is a Type 1 hypervisor and is part of the RedHawk PRT kernel. RedHawk KVM-RT leverages the unique features of RedHawk PRT host for memory management, process scheduling, device access and I/O.

KVM-RT creates individual threads for each virtual CPU allocated to a VM. Additional per-VM threads that are responsible for VNC/Spice and I/O interfaces are also created. Real-Time performance is guaranteed through RedHawk PRT's shielding and scheduling mechanisms.

I/O devices can be completely passed through to specific guests or para-virtualized to reduce the idle time of devices. Para-virtualized devices reduce the amount of necessary hardware without compromising throughput.

Setup

Two different system configurations are chosen to test the real-time performance of KVM-RT guests.

Hardware

- Artesyn ATCA-7480 blade
  Two 12-Core Xeon E5-2648L CPUs
  64 GB DDR4 (32GB per NUMA node)
- Supermicro X11DPH-i motherboard
  Two 10-Core Xeon Silver 4114 CPUs
  32 GB DDR4 (16GB per NUMA node)

Software

The host is 64-bit RedHawk PRT 7.5. The VMs are instances of 64-bit RedHawk 7.5, 64-bit RedHawk 6.5, 32-bit RedHawk 6.5 and Windows 10 Home edition.

We use ccur-rtbench package which provides the *cyclictest* to measure real-time response and *stress* to produce CPU, I/O and memory loads.

VM Initialization

The entire system configuration must be known in advance with a plan for mapping each VM to specific physical cores. Each real-time VM must be allocated more physical cores than virtual cores. For this paper, each RedHawk Linux VM is allocated one more physical core than virtual cores to accommodate the additional threads responsible for emulation and I/O.

The individual virtual core (vCPU) threads of a VM must be given proper affinity to a shielded physical core to allow for real-time. This is accomplished by using the *run* command.

Example: Here we see one VM assigned to 3 vCPUs.

host$ top -H –u qemu

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
22969	qemu	20	0	4769708	209756	19796	R	99.7	0.3	1:24.69	qemu-KVM-RT
22970	qemu	20	0	4769708	209756	19796	R	99.3	0.3	1:17.06	qemu-KVM-RT
22971	qemu	20	0	4769708	209756	19796	R	99.3	0.3	1:16.86	qemu-KVM-RT

The vCPUs of this VM are 22969, 22970 and 22971. Each of these threads must be assigned a shielded CPU and changed to run with real-time scheduling policy.

host$ shield –a 12-15

host$ run –s fifo –P 1 –b c13 –t 22969

host$ run –s fifo –P 1 –b c14 –t 22970

host$ run –s fifo –P 1 –b c15 –t 22971

Real-Time performance testing

The following table describes the configuration of the Host and VMs. Appropriate CPU shielding and VM initialization is performed as described in the section above to preserve the real-time performance of RedHawk VMs.

Artesyn Blade

Host/Guest	# Cores	Memory (GB)	OS	NUMA node
Host	24	64	RedHawk PRT 7.5	-
VM 1	6	16	Windows 10	0
VM 2	4 (3 vCPUs)	10	RedHawk 7.5	1
VM 3	4 (3 vCPUs)	4	RedHawk 6.5 (32-bit)	1
VM 4	4 (3 vCPUs)	4	RedHawk 6.5 (64-bit)	1

Supermicro X11DPH-i

Host/Guest	# Cores	Memory (GB)	OS	NUMA node
Host	20	32	RedHawk PRT 7.5	-
VM 1	6	8	Windows 10	0,1
VM 2	4 (3 vCPUs)	4	RedHawk 7.5	0
VM 3	4 (3 vCPUs)	4	RedHawk 6.5 (32-bit)	1
VM 4	4 (3 vCPUs)	4	RedHawk 6.5 (64-bit)	1

RedHawk host and RedHawk VMs run *stress*[*] and *cyclictest*[+] with the following options

$ stress –cpu 3 –io 3 –vm 1

$ cyclictest –a 1 –p 95 –m

Results

Artesyn Blade

After 24 hours, cyclictest measures a worst-case response time of 24 microseconds

Artesyn	Min (us)	Avg (us)	Max (us)
RedHawk PRT 7.5 (Host)	2	3	12
RedHawk 7.5	6	6	22
RedHawk 6.5 (32-bit)	7	7	24
RedHawk 6.5 (64-bit)	7	8	24

On Artesyn blade, the native real-time performance of RedHawk PRT 7.5 without any VMs running has a worst-case response time of 6 microseconds.

Supermicro X11DPH-i

After 24 hours, cyclictest measures a worst-case response time of 70 microseconds

X11DPH-i	Min (us)	Avg (us)	Max (us)
RedHawk PRT 7.5 (Host)	8	9	37
RedHawk 7.5	5	5	59
RedHawk 6.5 (32-bit)	4	6	70
RedHawk 6.5 (64-bit)	4	6	68

On Supermicro X11DPH, the native real-time performance of RedHawk PRT 7.5 without any VMs running has a worst-case time of 31 microseconds.

*** Stress can be downloaded from https://people.seas.harvard.edu/~apw/stress/**

+ Cyclictest can be downloaded from https://git.kernel.org/pub/scm/utils/rt-tests/rt-tests.git

PCIe Pass-Through

KVM-RT supports PCIe pass-through to the VMs. Devices such as NICs and USB cards can be assigned to the VMs easily. NVIDIA GPUs can be assigned to VMs and are available only if the VM is booted using UEFI and if the GPU supports UEFI. This feature is beyond the scope of this paper.

KVM-RT performs PCIe pass-through by mapping a dummy driver to the PCIe device on the host. The dummy driver is responsible for translating memory mappings. Any communication from the VM to the PCIe devices needs to go through the PCIe device driver within the VM, the dummy driver on the host and the hypervisor. Latency overhead is dependent upon the PCIe device being used and needs to be individually evaluated.

Some PCIe devices support VFIO drivers for lower latency between VMs and PCIe devices.

Conclusions

RedHawk KVM-RT can support multiple real-time guests while maintaining real-time performance on the host as well, while having non real-time guests like Windows 10 running alongside. The guests can be either 32-bit or 64-bit Linux or Windows.

Both RedHawk KVM-RT and Xen were investigated for this paper, however Xen real-time performance is not as good as KVM-RT. Xen provides soft real-time on multicore while providing hard real-time only for single-core use. Xen is also not supported on RHEL or CentOS.

It is to be noted that real-time performance is also hardware dependent. Windows 10 performance can be enhanced by adding dedicated hardware acceleration for its VM. If more than two virtual machines are required to support graphics of any type, it is recommended that additional GPU hardware is allocated to the VMs.

About Concurrent Real-Time

Concurrent Real-Time is the industry's foremost provider of high-performance real-time Linux® computer systems, solutions and software for commercial and government markets. The Company focuses on hardware-in-the-loop and man-in-the-loop simulation, data acquisition, and industrial systems, and serves industries that include aerospace and defense, automotive, energy and financial. Concurrent Real-Time is located in Pompano Beach, Florida with offices through North America, Europe and Asia.

Concurrent Real-Time's RedHawk Linux is an industry standard, real-time version of the open source Linux operating system for Intel x86 and ARM64 platforms. RedHawk Linux provides the guaranteed performance needed in time-critical and hard real-time environments. RedHawk is optimized for a broad range of server and embedded applications such as modeling, simulation, data acquisition, process control and imaging systems.

For more information, please visit Concurrent Real-Time at www.concurrent-rt.com.