A Concurrent Real-Time White Paper

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

Comparison of Real-time Network Performance of RedHawk™ Linux® 7.5.2 and Red Hat® Operating Systems

By: David Baston

Software Engineer

July 2019

Abstract

Concurrent Real-Time's RedHawk Linux operating system is designed specifically for a variety of applications that require time-critical or hard real-time environments. Its user-level commands are fully compatible with Red Hat Enterprise Linux and CentOS. [1] This paper will analyze the influence of these kernels on network latency by studying the average round-trip time divided by 2 (RTT/2) for TCP/IP and UDP/IP protocols over several network interface cards (NICs). This paper will also show that in the presence of a load on a system, Concurrent's RedHawk kernel consistently outperforms the Red Hat and CentOS kernels in terms of latency and jitter without additional procedures or user configurations in place. Additional user-level commands from RedHawk (specifically the shield and run commands) will show significant performance improvement comparable to, if not superior to, similar command-line configuration mechanisms inherent in the Linux kernel.

Procedure

The network performance procedures identified in this paper were performed on two identical test machines in a laboratory setting running Intel® Xeon® 3.0 GHz quad-core processors. A dedicated server-client relationship was established between the test machines on a defined subnet with no additional network traffic between them. The NIC cards between test machines were directly linked via a Mellanox passive copper InfiniBand QSFP 2-meter cable with 40 Gb/s capability.

All testing was performed and repeated using the following kernels with each NIC:

Kernels:

- RedHawk 7.5.2
- RHEL 7 RT with PREEMPT_RT
- CentOS 7

NIC:

- Solarflare SFN8542 Dual-Port 40 Gigabit QSFP+ with OpenOnload
- Mellanox ConnectX-5 CX555A VPI/FDR/EDR InfiniBand 100 Gigabit Dual-Port QSFP28 PCI Express 3.0 x16
- Intel Omni-Path HFI Silicon 100 Series

Latency testing, measured in terms of average round-trip time (RTT/2), was performed and compared between each kernel for the three different NICs using the TCP/IP and UDP transfer protocols. For each kernel, latency was tested with and without an additional load on the system. Testing *without a load* refers to performing the test with no other user processes running. Additional load on the system, in this study, will refer to a continuous building of a kernel as a series of background processes.

Latency is further tested under the condition of a load placed on the system with and without CPU *shielding*. Shielding in this study will refer to implementing the RedHawk kernel CPU shielding capability (using the shield command), or with Linux's command-line restriction isolcpus, with a single CPU isolated for testing purposes. These functions are both intended to restrict CPU usage by the kernel to tasks specifically purposed to them with a user-level command. In this case, a load was placed on both test systems while CPU 1 was shielded and assigned to all networking tasks. All network cards were given interrupt affinity to CPU 1 as well. The specificity given to performing all network tasks on a single CPU is done for benchmarking purposes and not as a means for optimizing performance in this study.

Test machines were identically prepared for benchmarking with the following commands as recommended in the Solarflare Onload latency benchmarking guide:

- `systemctl stop cpupower`
- `systemctl stop irqbalance`
- `systemctl stop firewalld`
- `ethtool –C` ***device*** `rx-usecs 0 adaptive-rx off`
- SMP affinity should be set to CPU 1 for all network card interrupts
- `tuned-adm profile network-latency`
- `sysctl net.core.busy_poll=50`
- `sysctl net.core.busy_read=50`

Latency data was collected using the sfnt-pingpong utility from sfnettest-1.5.0, found at openonload.org. RTT/2 and jitter are reported in nanoseconds and converted to microseconds in this paper. The sfnt-pingpong utility reports RTT/2 and jitter for multiple packet sizes from 1 byte to 65536 bytes. Data was further compared between TCP and UDP transfer protocols. Comparisons between kernels are standardized with the 32-byte package size.

Network Interface Cards:

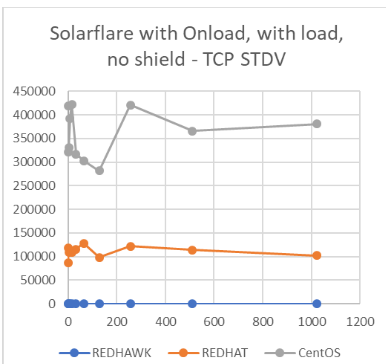
The **Solarflare Dual-Port 40 Gigabit QSFP+ Network Adapter** combines precision time synchronization and hardware timestamping with 40GB/10GB Ethernet. [2] Adapters also support Onload which provides a *kernel bypass* accelerated user-level TCP/IP network stack, providing extremely low latency. [2] The Onload stack includes transmit and receive buffers, open connections and the associated port numbers and stack options. Onload reduces CPU overhead by eliminating kernel-to-user interrupts, allowing applications to remain in user-space while the network adapter directly handles network receive calls, taking those duties away from the kernel.

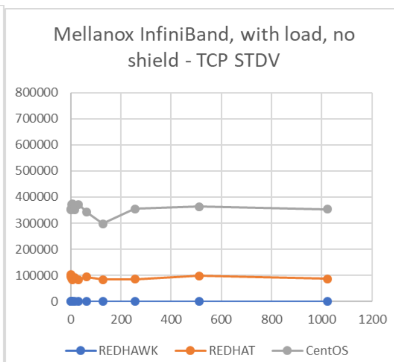
The **Mellanox ConnectX-5 Ethernet Adapter Card** promotes new acceleration engines for maximizing High Performance, Web 2.0, Cloud, Data Analytics and Storage platforms. [3] The technology is meant to increase transfer rates to improve Cloud and NFV platform efficiency through offloading of virtual switches and routers. [3]

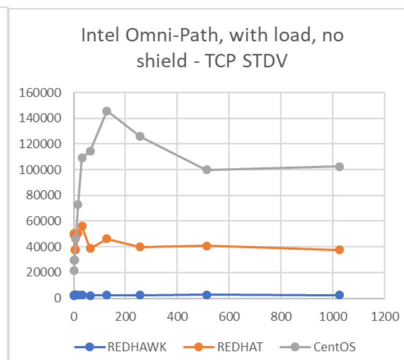
The **Intel Omni-Path Host Fabric Interface Silicon 100 Series Adapter** promotes efficiency with large packet transfer support for reduced per-packet processing overheads and multi-core scaling. [4] The Host Fabric Interface adapters mitigate traffic control for Link Transfer Packets (LTPs) for efficient wire transfer while the Omni-Path Architecture (OPA) is meant to make switching decisions to optimize latency and throughput. [4]

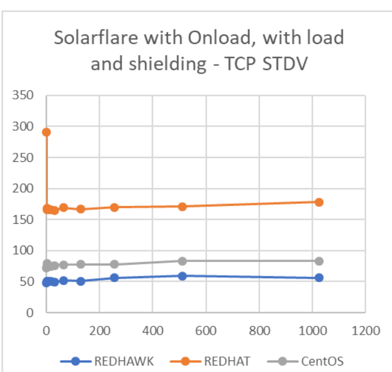
Comparing Operating Systems to a NIC:

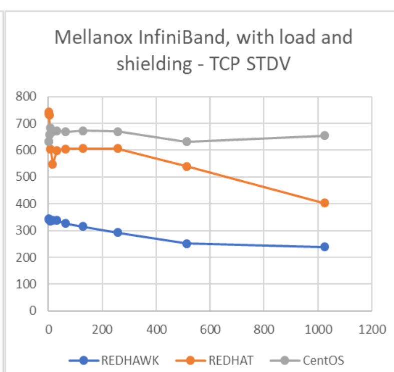
The data provided below shows that RedHawk's kernel interacts with consistently less jitter than the other operating systems, most notably when a load is placed on the system. The following graphs display transfer packet size in bytes vs. standard deviation time in nanoseconds. With competitive RTT/2 times for all packet sizes, significantly decreased jitter translates to a greater confidence in consistency with a focus on real-time applications. The comparisons of latency standard deviations presented below show that when CPU shielding methods are implemented, RedHawk tests consistently demonstrate nearly 50% less jitter than Red Hat and CentOS tests for all packet sizes.

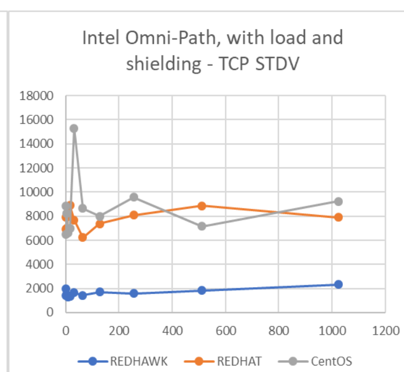


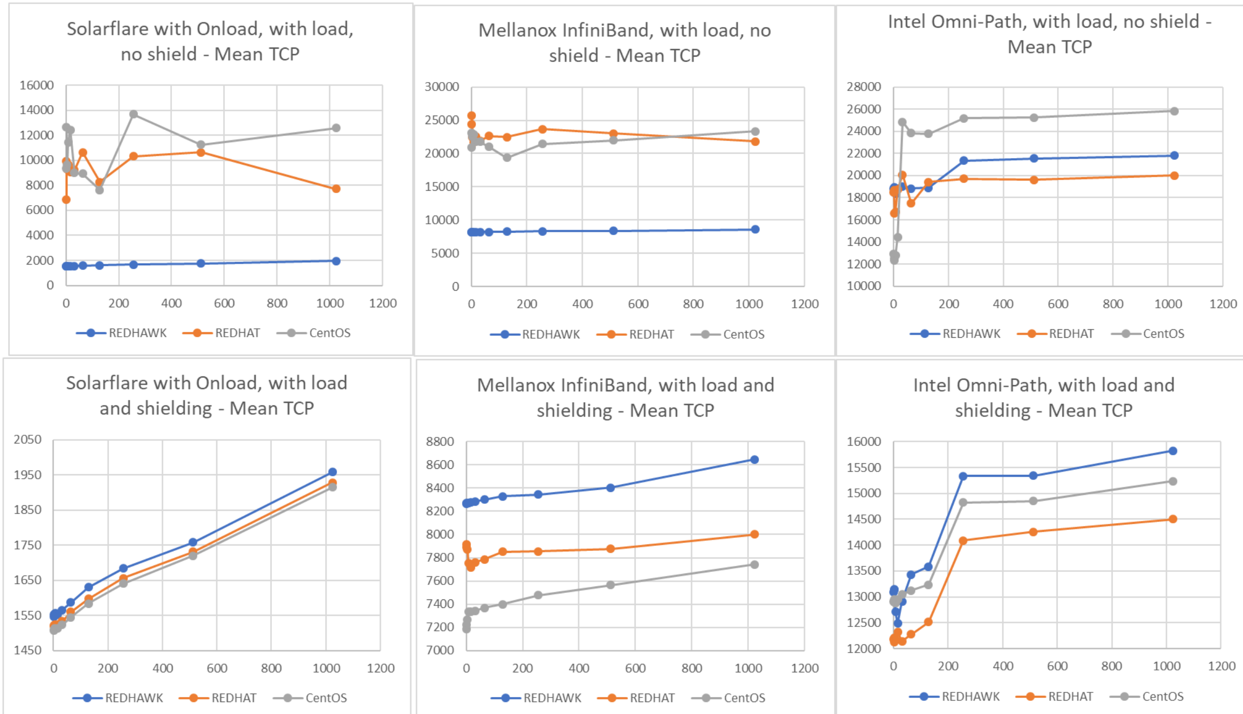






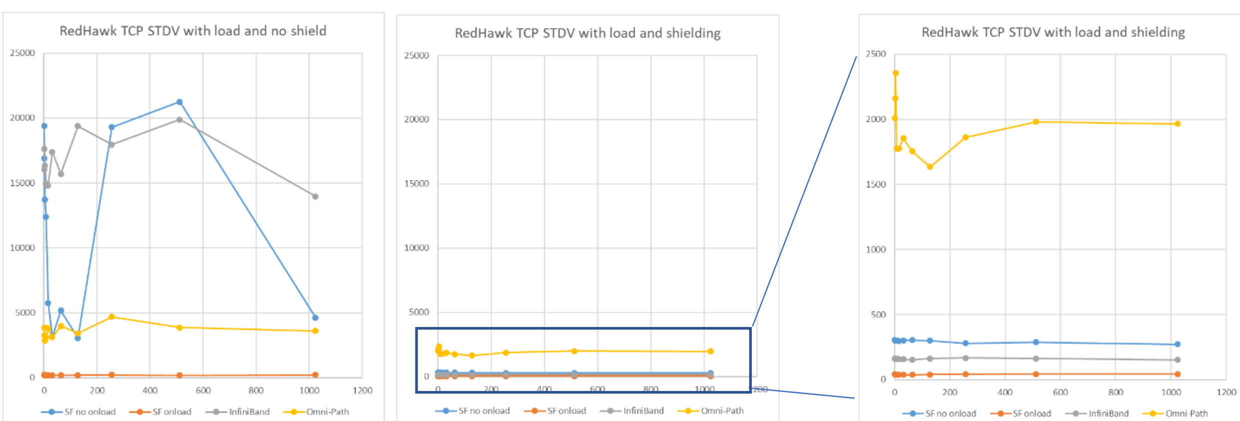






The graphs above display packet size in bytes vs. latency RTT/2 time in nanoseconds. In instances with a load present on the system and no shielding, RedHawk demonstrates comparatively lower latency times when implementing Solarflare and Mellanox InfiniBand adapters. When both shielding and a load are present, all three kernels tend to perform within one microsecond of each other.

Comparing NICs for an Operating System:



While RedHawk shows better latency standard deviation values compared to other operating systems on the same NIC, Solarflare with Onload improves jitter the most over the other network adapters.

Table 1 – `snft-pingpong` latency values in microseconds, broken down by NIC, shielding and OS

NIC	Back-ground Load	CPU Shielding	OS	`snft-pingpong` (**units in microseconds**)			
				tcp		udp	
				RTT/2	jitter stdev	RTT/2	jitter stdev
Solarflare without onload	No	N/A	RedHawk 7.5.2	6.997	0.463	6.802	0.234
			RHEL 7 RT	8.452	0.255	7.662	0.242
			CentOS 7	7.014	0.231	6.088	0.207
	Yes	No	RedHawk 7.5.2	7.636	0.29	6.345	0.511
			RHEL 7 RT	114.756	313.876	48.68	195.245
			CentOS 7	21.744	335.4	16.944	284.923
	Yes	Yes	RedHawk 7.5.2	7.844	0.237	6.26	0.465
			RHEL 7 RT	8.475	0.252	7.724	0.321
			CentOS 7	7.122	0.154	6.167	0.135
Solarflare with onload	No	N/A	RedHawk 7.5.2	1.538	0.123	1.431	0.094
			RHEL 7 RT	1.513	0.089	1.416	0.075
			CentOS 7	1.51	0.101	1.412	0.099
	Yes	No	RedHawk 7.5.2	1.545	0.142	1.444	0.118
			RHEL 7 RT	9.219	115.35	3.736	54.921
			CentOS 7	8.982	317.102	3.699	150.493
	Yes	Yes	RedHawk 7.5.2	1.565	0.049	1.439	0.044
			RHEL 7 RT	1.533	0.165	1.427	0.153
			CentOS 7	1.523	0.076	1.423	0.074
Mellanox InfiniBand	No	N/A	RedHawk 7.5.2	8.006	0.219	7.512	0.381
			RHEL 7 RT	8.371	0.879	7.446	0.939
			CentOS 7	7.148	0.656	6.074	0.483
	Yes	No	RedHawk 7.5.2	8.2	0.275	7.686	0.421
			RHEL 7 RT	21.916	84.718	13.975	65.658
			CentOS 7	21.787	371.32	21.427	678.71
	Yes	Yes	RedHawk 7.5.2	8.284	0.338	7.735	0.341
			RHEL 7 RT	7.762	0.599	7.517	0.834
			CentOS 7	7.342	0.672	6.278	0.573
Omni-Path	No	N/A	RedHawk 7.5.2	12.208	0.395	11.341	0.539
			RHEL 7 RT	10.998	9.249	9.637	5.036
			CentOS 7	11.59	0.317	10.193	0.302
	Yes	No	RedHawk 7.5.2	19.028	2.382	16.397	2.973
			RHEL 7 RT	20.081	56.205	12.826	26.895
			CentOS 7	24.837	109.508	19.374	96.61
	Yes	Yes	RedHawk 7.5.2	12.918	1.64	11.779	1.406
			RHEL 7 RT	12.148	7.671	10.404	7.312
			CentOS 7	13.047	15.309	11.26	7.571

Conclusion:

The table above highlights the performance advantages that the RedHawk kernel provides when a system is under load. While average round-trip latency times are similar under shielded conditions, RedHawk additionally provides a reduction in standard deviation of nearly 50% compared to the Red Hat real-time kernel. This enhancement of consistency directly translates to the heightened confidence necessary to ensure real-time application performance.

The use of Onload consistently improves RTT/2 and jitter over TCP and UDP protocols across the board.

Additional Resources:

[1] http://concurrent-rt.com/solutions/linux

[2] http://solarflare.com

[3] Mellanox.com documentation - https://docs.mellanox.com/display/MLNXOFEDv461000/MLNX_OFED+Documentation+Rev+4.6-1.0.1.1

[4] Intel.com Omni-Path documentation - https://software.intel.com/en-us/articles/using-intel-omni-path-architecture

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.