AMD EPYC

TUNING GUIDE

AMD EPYC™ 9005

Linux® Network Tuning Guide

AMD
together we advance_data center computing

PID: 58472
v1.1
October 2024

Steve Rochefort

DATE	VERSION	CHANGES
June, 2024	0.1	Initial NDA release
October, 2024	1.0	Initial public release
October, 2024	1.1	Corrected list of tested NICs and added cable details

AUDIENCE

This document is intended for a technical audience with a server configuration background who have:

- Admin access to the server's management interface (BMC).
- Familiarity with the server's management interface.
- Admin OS access.
- Familiarity with the OS-specific configuration, monitoring, and troubleshooting tools.

LINUX® NETWORK TUNING GUIDE

CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 1: Introduction

This Tuning Guide provides an overview of steps needed to tune your chosen network adapters for optimal performance in a platform powered by AMD EPYC™ 9005 Series Processors running Linux®, including the steps taken by AMD engineers to prepare the reference platform for maximum performance. If you are testing a system powered by AMD EPYC 9005 Series Processors that was designed by another company, then be sure to also review the vendor product documentation.

There is no single golden rule for tuning a network interface card (NIC) for all conditions. Different adapter models have different parameters that should be changed. Operating systems also have settings that can be modified to help with overall network performance. Depending on the exact hardware topology, you may have to make different adjustments to optimize for a specific workload. With Ethernet speeds going higher, up to 400 Gbps, and the number of ports being installed in servers growing, these tuning guidelines become even more important to achieve the best performance possible.

This guide does not provide exact settings for modifying every scenario. Rather, it includes parameters to check and modify for a given configuration. In this guide, the steps are focused on TCP/IP network performance. Table 3-1 provides tables of recommended tuning parameters used. Review the block diagrams and descriptions of the AMD EPYC™ 9005 processor NUMA architecture in the following sections in the *AMD EPYC™ 9005 Series Architecture Overview* (available from the AMD Documentation Hub) before you begin tuning:

- Memory and I/O
- NUMA Topology
- Dual-Socket Configurations

All I/O uses data transfers into or out of memory, hence the I/O bandwidth can never exceed the capabilities of the memory subsystem. Therefore, before you start, verify that your memory subsystem is properly configured for maximum frequency. To reach maximum memory bandwidth on modern CPUs, you must populate one DIMM in every DDR channel. For AMD EPYC™ 9005 Series Processor-based servers, there are twelve DDR5 channels on each CPU socket. For a single-socket platform, populate all twelve memory channels. Likewise, on a dual-socket platform, you should populate twenty-four memory channels.

In addition, AMD recommends consulting the tuning guide available from your NIC vendor. Each vendor decides which standard commands they support and may have also created their own value-added commands to support. As examples: A vendor may support interrupt coalescing or not. Another vendor may support relaxed ordering of PCI transactions while another does not.

1.1 - Important Reading

Please be sure to read the following guides (available from the AMD Documentation Hub), which contain important foundational information about 5th Gen AMD EPYC processors:

- *AMD EPYC™ 9005 Processor Architecture Overview*
- *BIOS & Workload Tuning Guide for AMD EPYC™ 9005 Series Processors*
- *Memory Population Guidelines for AMD EPYC™ 9005 Series Processors*

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 2: TCP Performance Tuning

This chapter addresses test configuration, BIOS tuning, network adapter tuning and OS tuning.

2.1 - Test Configuration

Several of the network adapters evaluated have very high-bandwidth ports that are faster than the switches commonly available in engineering labs. AMD therefore adopted a test method that eliminates the switch by passing data directly between two systems. AMD engineers connected two identical AMD EPYC 9005 systems and passed data directly between them, as shown in Figure 2-1.

Figure 2-1: Direct connections between the AMD reference systems used when creating this Tuning Guide

Note: If your system ships with multiple network adapters installed, it is a good practice to first test with one adapter and demonstrate line rate as discussed in this Tuning Guide before you try to scale bandwidth by increasing the number of adapters.

Note: It is important to use two identically-configured systems.

2.2 - Single- and Dual-Socket Systems

AMD measures traffic passing between two identical network adapters plugged into PCIe slots in reference boards. The network adapter should only use local resources regardless of whether the processor it is connected to is installed in a single- or dual-socket system.

Determining which socket the adapter is connected to and which NUMA node the adapter is in is a standard first step when preparing to run tests. This ensures that the adapter is only using local cores and memory when passing traffic. Performance will be sub-par if the adapter uses cores and/or memory on the distant socket.

Dual-socket systems use xGMI links between the sockets. You can dynamically reduce the xGMI width to save power. It is a good practice to force the XGMI link to the maximum width supported by your board, but the most important safeguard is to verify that your test script is NUMA aware and uses local resources.

2.3 - BIOS Tuning

It is a good practice to start fresh by loading the optimized default BIOS settings before beginning the tuning process. This is especially true if you are sharing a system with other users. Resetting the BIOS to default can also be faster than manually changing BIOS settings, especially if you are not certain what those defaults are. It can also be a time saver to manually set BIOS settings if you are not sure what the default settings are. For example, your BIOS may default to an **Auto** memory speed instead of the maximum available speed.

Note: This section presents BIOS settings as they appear in the default AMD BIOS. Different Original Equipment Manufacturers (OEMs) may modify the names and/or locations of these settings.

2.3.1 - Numa Nodes Per Socket (NPS)

The BIOS **NPS** setting allows you to make a trade-off between minimizing local memory latency for NUMA-aware or highly parallel workloads versus maximizing per-core memory bandwidth for non-NUMA friendly workloads. Setting NPS=1 interleaves all 12 memory channels on a socket.

AMD standard BIOS defaults to NPS=1 with **LLC as NUMA** disabled. This setting reports one NUMA node per socket to the operating system. However, using NPS=1 with LLC as NUMA enabled means the OS will see one NUMA node per L3 cache and still use 12-channel interleaving. The combination of NPS=1 and **LLC as NUMA** enabled is usually the best combination for NIC tuning. If you are concerned about latency, then set NPS=2 to interleave 6 memory channels.

Advanced > AMD CBS > DF Common Options > Memory Addressing > NUMA nodes per socket > NPS1

2.3.2 - Last Level Cache (LLC) as NUMA Domain

AMD EPYC processors use multiple Last Level Caches (LLCs, or L3 caches). Operating systems can handle multiple LLCs and schedule jobs accordingly; however, the AMD BIOS **LLC as NUMA** setting allows creating a single NUMA domain per LLC. This can help the operating system schedulers maintain locality to the LLC without causing unnecessary cache-to-cache transactions. Please see the latest version of the Socket SP5/SP6 Platform NUMA Topology for AMD Family 1Ah Models 00h–0Fh and Models 10h–1Fh (login required; please review the latest version if multiple versions are present) and the *BIOS & Workload Tuning Guide for AMD EPYC™ 9005 Series Processors* (available from the AMD Documentation Hub), for additional information.

Advanced > AMD CBS > DF Common Options > ACPI > ACPI SRAT L3 Cache as NUMA Domain > Enable

2.3.3 - SMT

Symmetric Multithreading (SMT) is enabled by default. To take measurements with SMT disabled:

Advanced > AMD CBS > CPU Common Options > Performance > SMT Control > Disable

2.3.4 - X2APIC

AMD EPYC 9005 Series Processors include an x2APIC controller. This has two benefits:

- Allows operating systems to work with the 384 CPU threads now available on AMD platforms.
- Provides improved performance over the legacy APIC AMD recommends without requiring you to enable the x2APIC mode in BIOS, even for lower core counts.

The X2APIC option should be selected by default; however, if needed, you can set it manually:

Advanced -> AMD CBS -> CPU Common Options -> Local APIC -> x2APIC

2.3.5 - Determinism Control and Slider

The **Determinism** BIOS setting can affect throughput,

Advanced > AMD CBS > NBIO Common Options > SMU Common Options > Determinism Control > Manual > Determinism Enable > Enable Performance

2.3.6 - 10-Bit Tag

The **PCIE Ten Bit Tag Support** setting increases the maximum number of non-posted requests from 256 to 768 and sometimes helps with high bandwidth port throughput. AMD CRBs implement the 10-bit tag by default.

Advanced > AMD CBS > NBIO Common Options > PCIE > Ten Bit Tag Support > Enabled

2.3.7 - Memory Clock Speed

Setting the memory clock speed to **Auto** should select the maximum memory speed using default BIOS settings; however, you are welcome to manually set the speed in BIOS. Either way, be sure to check the memory speed from the operating system before starting performance tests. The `dmidecode` and `lsmem` commands are particularly useful.

Advanced > AMD CBS > UMC Common Options > DDR Timing Configuration > Accept > Memory Target Speed > 6000

2.3.8 - Slot Bifurcation

The Intel® E810-2CQDA2 requires slot bifurcation for full performance. This is the only card tested that has this requirement. AMD Chalupa systems have all four PCIE slots connected to Socket 0. A "P" link is routed to each PCIe slot. Slot 1 is Link 0. You can change Slot 1 from 16x1 to 8x2 as follows:

Advanced > AMD CBS > NBIO Common> Enable Port Bifurcation > Socket0 Slot Info Override > Socket P0 Override > 2 ports of x8 (instead of **Auto**)

You can change the slot bifurcation for any slot you use by overriding the default setting for the correct Px link.

2.4 - Network Adapter Tuning

AMD strongly recommends that you disable firewalls and install a fresh copy of the operating system on your EPYC platform, being sure to install the latest NIC vendor firmware and drivers before proceeding. Be sure to review the installation for errors. Be sure that the OS has access to the network during installation and can download anything needed. Lastly, if you are adopting a script that someone else wrote or that worked on another CPU platform or with another NIC vendor's product, then be prepared to debug it. These are common mistakes.

2.4.1 - Local NUMA Node Usage

Ensure maximum performance by using cores and memory that are in the same NUMA node as your network adapter. There are different ways to check to see the core and NUMA node assignments, such as by executing the `lscpu` command. Other useful commands are:

- `cat /sys/class/net/(ethernet interface)/device/numa_node`
- `cat /sys/class/net/(ethernet interface)/device/local_cpulist`

2.4.2 - Controlling IRQ

Controlling the number of interrupt queues used within a NUMA node is a key element in network performance because you do not want to have more interrupt queues than you have CPU cores. Assigning multiple queues to a single core introduces the risk that the interrupt handler repeatedly swaps between queues, reducing efficiency (thrashing). It is more efficient to have a single queue per core to eliminate that thrashing.

Before making assignments, stop and then disable `irqbalance` altogether.

There are three types of interrupt queues:

- Receive (RX)
- Transmit (TX)
- Combined. This uses a single queue to handle both RX and TX interrupts. Some vendors still have separate RX and TX queues while others implement only combined queues. For example, if you have a one-port NIC that combines the RX and TX interrupts, then you could execute the following command:

  ```
  ethtool -L enp33s0f0 combined 16 tx 0 rx 0
  ```

Make sure to review the default settings before you begin making changes. In the above example, you can read the default setting by executing the `ethtool -I enp33s0f0` command.

2.4.3 - TX/RX Flow Steering

Advanced Receive Flow Steering (aRFS) is an effective way to direct incoming packets to the cores where the application is running. Use this If supported by your NIC.

2.4.4 - TX/RX Queue Size

The NIC Queue size or ring size represents the number of buffers that a NIC uses to DMA data into system memory. Increasing the TX and RX queue size will help prevent dropped packets. The `ethtool` utility allows you to find the current ring size and maximum allowed ring size. For example:

```
root@testsystem:~# ethtool -g enp33s0f0
Ring parameters for enp33s0f0:
Pre-set maximums:
RX:             8192
RX Mini:        0
RX Jumbo:       0
TX:             8192
Current hardware settings:
RX:             512
RX Mini:        0
RX Jumbo:       0
TX:             512
Ethtool -G (device interface) tx 2047 rx 2047
```

You can then set the buffer sizes to the maximum value, a value recommended by your NIC vendor, or one that you find works best. For example:

```
ethtool -G (device interface) tx 2047 rx 2047
```

After issuing the `ethtool` command with a `–G` to set ring size, you should reissue the command with `–g` to verify that the change was accepted.

2.4.5 - Relaxed Ordering

3rd Gen and prior AMD EPYC processors included the **Preferred I/O** and **Relaxed Ordering** settings that helped optimize network and disk I/O performance. 5th Gen AMD EPYC processors (9xx5 models) include architectural enhancements that deliver optimal network and disk I/O performance by default without the need for either of these features.

2.4.6 - LRO

Many NIC vendors have tuning guides to help end users optimize for specific use cases. Those use cases usually involve optimizing for either the highest possible throughput or the lowest possible latency possible but can only rarely achieve both at the same time. Enabling Large Receive Offload (LRO) is a common way to improve network throughput performance. Consult your NIC documentation for information on enabling LRO on your adapter. Some providers use standard commands available via `ethtool`, while others might have extra parameters that are needed to fully enable LRO. Be aware that enabling LRO helps improve your throughput but could negatively impact network latency. Make sure to tune as appropriate for your workload.

2.4.7 - TX-No Cache Copy

Check the default value for your device driver's setting or proactively shut it off as shown here:

```
Ethtool -K $[local interface name] tx-nocache-copy off
```

2.5 - OS Tuning

AMD recommends that you perform testing using a fresh copy of an operating system that understands and support the AMD EPYC 9005 Series Processor.

2.5.1 - IOMMU Settings

The Linux kernel is constantly updated. The official mainline releases from The Linux Foundation are available via http://kernel.org*. However, common enterprise level Linux distributions rarely use a mainline kernel.

AMD has contributed code into the Linux kernel for years. The most recent focus has been enabling the "Zen," "Zen 2," "Zen 3," and "Zen 4" architectures contained in AMD EPYC processors. One area of code contribution focuses on optimizing the input-output memory management unit (IOMMU) code for AMD EPYC processors. These IOMMU patches can have a direct impact on TCP/IP performance, even in a bare metal (non-virtualized) environment.

Either disabling the IOMMU or setting it to pass through mode (which disables DMAR to memory) can sometimes benefit the highest bandwidth adapters, such as 200 or 400 Gbps Ethernet adapters. To set the IOMMU to pass-through mode, the following kernel parameter must be passed in during boot time in the `grub` command line:

```
iommu=pt
```

After booting the system, check the setting by executing the following command:

```
Cat /proc/cmdline | grep -I iommu=pt
```

2.5.2 - Nohz

Nohz is another boot time parameter that can be included in the `grub` command line to disable `dyntick idlemode`.

```
nohz=off
```

2.5.3 - IRQ Balancing

CPUs generally share interrupts automatically, but this can cause delayed interrupt processing. To disable this:

```
systemctl disable irqbalance
```

2.5.4 - TCP Memory Configuration

Increasing the memory buffer size for TCP sockets can help eliminate transmission gaps when lots of data is in flight and device buffers are full. The following examples define minimum, nominal, and maximum TCP socket buffer values:

```
echo "4096 131072 268435456" > /proc/sys/net/core/tcp_rmax
echo "4096 131072 268435456" > /proc/sys/net/core/tcp_wmax
```

2.5.5 - Scaling Governor

Set the CPU scaling governor to Performance mode by executing the following command:

```
Echo performance  | sudo tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

THIS PAGE INTENTIONALLY LEFT BLANK.

Chapter 3: Additional Information

3.1 - Recommendations and Results

Table 3-1 provides recommended values for each of the options described in this Tuning Guide. As shown, not all adapters require modifying the default BIOS, OS, or adapter settings. You may also find that some drivers already enable a feature (such as relaxed ordering) by default. You should verify all the settings listed before making any assumptions about default settings.

	Dual Port 25 Gbps Ethernet	Single Port 100 Gbps Ethernet	Dual Port EDR InfiniBand	Dual Port 100 Gbps Ethernet	Single Port NDR InfiniBand
BIOS Options					
Local APIC Mode	default	x2apic	x2apic	x2apic	x2apic
Determinism mode	default	performance	performance	performance	performance
LLC as NUMA	default	enabled	enabled	enabled	enabled
10-bit tag	default	enabled	enabled	enabled	enabled
Adapter Options					
Relaxed Ordering	default	enabled	enabled	enabled	enabled
OS Options					
Ring Buffers	default	maximum	maximum	maximum	maximum
Large Receive Offload (lro)	default	enabled	enabled	enabled	enabled
Interrupts	default	combined 16	combined 16	combined 16	combined 16
MTU (default 1500)	default	default	default	default	default

Table 3-1: NIC configuration recommendations

Table 3-2 lists several adapters that have all been tested with RHEL 9.4. Following the guidelines contained in this Tuning Guide should yield near-line rate performance with any NIC you choose. AMD will continue testing and updating results as drivers are released for RHEL 9.5.

Note: All adapters were tested using the AMD "Chalupa" reference design with BIOS version RCHT1000F.

Tested Adapter	Port Speed	Product Description	Cable Used
Broadcom BCM957414A4142CC	25 Gbps	Dual-Port 25 Gbps Network Interface Card	Mellanox MCP2M00
Broadcom BCM957508-P2100G	100 Gbps	Dual-Port 100 Gbps Network Interface Card	Mellanox MCP1600
Broadcom BCM957508-P2200G	200 Gbps	Dual-Port Network Interface Card (one port x 200Gbps or two ports x 100Gbps)	Mellanox MCP1650
Broadcom BCM957608-P1400G	400 Gbps	Single-Port 400 Gbps Network Interface Card	FS Q112-PC015
Broadcom BCM957608-P2200G	200 Gbps	Dual-Port 200 Gbps Network Interface Card	FS Q112-PC015
Cornelis Networks 100HFA016LS	100 Gbps	Single-Port 100 Gbps Omni-Path Host Fabric Adapter	Intel MM#952163
Intel X710-DA2	10 Gbps	Dual-Port 10 Gbps Network Interface Card	Mellanox MCP2M00
Intel E810-2CQDA2	100 Gbps	Intel Dual-Port 100 Gbps Ethernet Adapter	Mellanox MCP2M00
NVIDIA MCX653106A-HDAT	200 Gbps	ConnectX-6 VPI Dual Port InfiniBand & Ethernet Adapter	Mellanox MCP1650
NVIDIA MCX75310AAS-NEAT	400 Gbps	ConnextX-7 VPI Single Port InfiniBand & Ethernet Adapter	Mellanox MCP4Y10
AMD X2541	40 Gbps	Single Port 40 Gbps Ultra Low Latency Ethernet Adapter	Mellanox MCP1600

Table 3-2: Tested network adapters

THIS PAGE INTENTIONALLY LEFT BLANK.

Linux® Network Tuning Guide for AMD EPYC™ 9005 Processors

PID: 58472

Steve Rochefort