AMD EPYC

TUNING GUIDE
AMD EPYC 8004

Data Plane Development Kit (DPDK)

Publication	58310
Revision	1.0
Issue Date	September, 2023

Date	Version	Changes
Jul, 2023	0.1	Initial NDA release
Sep, 2023	1.0	Initial public release

Audience

This tuning guide describes best practices for optimizing performance using the Data Plane Development Kit (DPDK). It is intended for a technical audience such as DPDK application architects, production deployment, and performance engineering teams with:

- A background in configuring servers.
- Administrator-level access to both the BMC and the OS.
- Familiarity with both the BMC and OS-specific configuration, monitoring, and troubleshooting tools. See the DPDK Debug & Troubleshoot Guide* for additional information.

Authors

Manish Kumar, Vaibhav Patankar, and Keesang Song

Table of Contents

This page intentionally left blank.

Chapter 1

Introduction

This tuning guide provides various system configuration parameters that can optimize DPDK workload performance on servers based on AMD EPYC™ 8004 series processors. Default OEM system configurations may not provide the best possible performance for all DPDK workloads across different OS platforms. To optimize performance for a particular workload, this guide calls out:

- Hardware configuration (memory, PCIe) best practices
- BIOS settings that can impact performance
- Workload-specific settings in BIOS and OS parameters
- DPDK build optimization options
- Vendor-specific NIC configuration
- DPDK environment (EAL) options

Note: Do not use this tuning guide as a validation guide or a generic server optimization guide. It is only intended to help you optimize the performance of a specific AMD EPYC-based platform.

This page intentionally left blank.

Chapter 2 Resources

2.1 Essential Reading

From AMD EPYC Tuning Guides:

- *BIOS and Workload Tuning Guide for AMD EPYC™ 8004 Series Processors*
- *AMD EPYC™ 8004 Series Architecture Overview*

2.2 Other Resources

- DPDK Debugging & Troubleshooting Guide*
- Getting Started Guide for Linux*
- DPDK EAL Parameters*
- Memory Population Guidelines for AMD Family 19h Models A0h–AFh Socket SP6 Processors - Login required; please review the latest version if multiple versions are present.
- Socket SP5/SP6 Platform NUMA Topology for AMD Family 19h Models 10h–1Fh and Models A0h–AFh - Login required; please review the latest version if multiple versions are present.
- From AMD EPYC Tuning Guides:
  - *BIOS & Workload Tuning Guide for AMD EPYC™ 8004 Series Processors*
  - *Linux® Network Tuning Guide for AMD EPYC™ 8004 Series Processor Based Servers*
  - *Windows® Network Tuning Guide for AMD EPYC™ 8004 Series Processor Based Servers*
  - *VMware® Network Tuning Guide for AMD EPYC™ 8004 Series Processor Based Servers*
- NIC and ecosystem vendor specific tuning guide for AMD EPYC platform, such as:
  - NVIDIA*
  - Broadcom*
  - Red Hat
  - Canonical (Ubuntu)
  - SUSE

This page intentionally left blank.

Chapter 3 System Configuration

3.1 Recommended BIOS Settings

This section describes the recommended BIOS settings for optimal DPDK workload performance on AMD EPYC 8004 Series Processors.

Name	Recommended Value	Description
ACPI Auto Configuration	Disabled	Allows the DPDK library to handle power management.
Local APIC Mode	X2APIC	Set the local APIC mode to **x2APIC = Enabled** to allow the operating system to work with 128 threads and improve performance over legacy APIC
SMT Control	Enable	Enables Symmetric Multithreading (SMT), which allows one hardware thread per core. *Note: Disable SMT if you are running an operating system that does not support X2APIC.*
NUMA Node per Socket (NPS)	NPS [1|2]	• **NPS1:** Maximum memory bandwidth without NUMA affinity. This is the recommended setting for monolithic application with complete resource provisioning flexibility. • **NPS2:** For multi-tenant VNF/CNF workloads and resource (compute/memory and IO) partitioning.
L1 Stream HW Prefetcher	Enable	Enables the L1 Stream HW Prefetcher.
L2 Stream HW Prefetcher	Enable	Enables the L2 Stream HW Prefetcher.
IOMMU	Enable	IOMMU allows operating systems to provide additional protection for DMA capable I/O devices. If needed, you can disable IOMMU in BIOS and enable it via OS options (i.e., `amd_iommu=pt` in the `grub` configuration)
Determinism Control	Manual	Enable the **Determinism Slider** control.
Determinism Enable	Power	Ensure maximum performance levels for each CPU in a large population of identically-configured CPUs by only throttling CPUs when they reach the same cTDP. Please see Power/Performance Determinism for more details.

Table 3-1: Recommended common BIOS settings

ACPI SRAT L3 Cache as NUMA Domain	Disable	Do not report each CCX/L3 cache as a NUMA domain to OS.
APBDIS	1	Disables APB (Algorithm Performance Boost) and enables fixed Infinity Fabric P-state control. By default, the AMD Infinity Fabric selects between a full-power and low-power fabric clock and memory clock based on fabric and memory usage. This transition from low power to full power can increase latency in cases involving low bandwidth but latency-sensitive traffic (and memory latency checkers). Disabling APB by setting APBDIS to 1 and specifying a fixed Infinity Fabric SOC P-state of 0 forces the Infinity Fabric and memory controllers into full-power mode, thereby eliminating any latency jitter.
Power Profile Selection	High Performance Mode	DPDK applications can be heavy in I/O, memory, or compute. Allowing **High-Performance Mode** ensures that the CPU and its subsystem will not be throttled.
Global C-state Control	Disabled	Enables CPU C-States for power management.
DF C-states	Disabled	Disables Infinity Fabric C-States.
GMI encryption control	Disabled	Controls GMI link encryption
XGMI encryption control	Disabled	Controls XGMI link encryption
xGMI Force Link Width Control	Auto	Forces the **XGMI Link Width** control.
xGMI Force Link Width	Auto	Forces XGMI link width to the minimum width.
SME	Disable	Disables SME (Secure Memory Encryption). This feature provides hardware-based encryption of all data stored on system DIMMs at a slight increase in memory latency. AMD recommends disabling this feature for high-throughput applications with low security risks.
AVX512	Enabled	Allows DPDK to enable AVX512 ISA for optimized performance.
Monitor and MWAIT Disable	Disabled	This allows DPDK libraries to leverage MWAITX and MONITORX ISA.
Core Performance Boost	Auto (Enabled)	Turns the boost function ON or OFF on all cores. Example: On Linux operating systems, enable CPU frequency boost using the following command: `echo 1 > /sys/devices/system/cpu/cpufreq/boost`
PCIe Speed PMM Control	Static Target Link Speed (GEN 5) (2)	Enables the PCIe speed controller: • **0:** Enable the activity based PCIE PMM controller. • **1:** Limit to Gen4 (set the maximum idle link speed to 16 GT/s). • **2:** Limit to Gen5 (set the maximum idle link speed to 32 GT/s, thereby essentially disabling the feature)

Table 3-1: Recommended common BIOS settings (Continued)

Workload Profile		Depending on the workload, choose either: • **Compute** • **Memory Throughput Intensive** • **NIC Throughput Intensive** • **CPU Intensive**
PCIE Link Speed Capability	Auto	Gen5/Gen4.

Table 3-1: Recommended common BIOS settings (Continued)

3.2 Linux OS Recommended Settings

This section lists some parameters that you can configure for DPDK applications by editing the grub configuration (`/etc/default/grub`).

3.2.1 Huge Pages

The minimum recommended huge pages for DPDK applications running on a bare metal (host) OS are:

```
default_hugepagesz=1GB hugepagesz=1G hugepages=6
```

Note: The recommended single-socket setting is multiples of 6.

3.2.1.1 IOMMU

Place the IOMMU in passthrough mode (`iommu=pt`) to improve host performance by disabling the DMAR to the memory `iommu=pt amd_iommu=on`.

3.2.1.2 ISOLCPUs

Linux kernel parameters to isolate CPUs from the Kernel scheduler to avoid context switches by preventing non-DPDK workloads from running on reserved cores. You can also specify `nohz_full` to avoid unbound timer callbacks execution to outside the `nohz_full` range. Similarly, you can specify `rcu_nocbs` to avoid RCU processing on the specified CPU cores.

- **Single Socket:**
  - **SMT (ON):** `isolcpus=8-15,72-79 nohz_full=8-15,72-79 rcu_nocbs=8-15,72-79`
  - **SMT (OFF):** `isolcpus=8-15 nohz_full=8-15 rcu_nocbs=8-15`

3.2.1.3 IRQ Affinity

Linux kernel parameter to set the default IRQ affinity mask / set of CPUs (non DPDK cores) that should process interrupts. For a 64-core scenario:

- **SMT (ON):** `irqaffinity=0-7,16-63,64-71,80-127`
- **SMT (OFF):** `irqaffinity=0-7,16-63`

3.2.1.4 C-States

The CPU can idle in several Core-States or C-States:

- **C0:** Active. This is the active state while running an application.
- **C1:** Idle
- **C2:** Idle and power gated. This is a deeper sleep state and will have a greater latency when moving back to the C0 state relative to when coming out of C1.

The recommended settings for maximum power saving state when C-states is enabled in BIOS are:

- **Low latency:** `processor.max_cstate=0`
- **Power management:** `processor.max_cstate=1`

3.2.1.5 Additional Settings

- `nohz=on rcu_nocb_poll numa_balancing=disable transparent_hugepage=never nosoftlockup rcu_nocb_poll` will relieve each CPU from the responsibility of awakening their RCU offload threads.
- `nohz=on` can be used to configure full `dynticks`.

Note: Edit the distro-specific grub configuration file and then execute the `update-grub; reboot` *command.*

3.2.2 Linux Configuration

3.2.2.1 NUMA Awareness

- For optimal performance, AMD recommends placing logical cores, memory, and IO devices in the same NUMA node. Execute the `lstopo-no-graphics` command to determine the NPS setting and also to obtain the list of logical cores along with the IO devices within each NUMA node. For example:

```
    L3 L#7 (16MB)
      L2 L#56 (1024KB) + L1d L#56 (32KB) + L1i L#56 (32KB) + Core L#56
        PU L#112 (P#56)
        PU L#113 (P#120)
      L2 L#57 (1024KB) + L1d L#57 (32KB) + L1i L#57 (32KB) + Core L#57
        PU L#114 (P#57)
        PU L#115 (P#121)
      L2 L#58 (1024KB) + L1d L#58 (32KB) + L1i L#58 (32KB) + Core L#58
        PU L#116 (P#58)
        PU L#117 (P#122)
      L2 L#59 (1024KB) + L1d L#59 (32KB) + L1i L#59 (32KB) + Core L#59
        PU L#118 (P#59)
        PU L#119 (P#123)
      L2 L#60 (1024KB) + L1d L#60 (32KB) + L1i L#60 (32KB) + Core L#60
        PU L#120 (P#60)
        PU L#121 (P#124)
      L2 L#61 (1024KB) + L1d L#61 (32KB) + L1i L#61 (32KB) + Core L#61
        PU L#122 (P#61)
        PU L#123 (P#125)
      L2 L#62 (1024KB) + L1d L#62 (32KB) + L1i L#62 (32KB) + Core L#62
        PU L#124 (P#62)
        PU L#125 (P#126)
      L2 L#63 (1024KB) + L1d L#63 (32KB) + L1i L#63 (32KB) + Core L#63
```

```
        PU L#126 (P#63)
        PU L#127 (P#127)
    HostBridge
      PCIBridge
        PCI 06:00.0 (NVMExp)
          Block(Disk) "nvme0n1"
      PCIBridge
        2 x { PCI 08:00.0-1 (SATA) }
    HostBridge
      PCIBridge
        PCI 41:00.0 (Ethernet)
          Net "enp65s0f0np0"
          OpenFabrics "mlx5_0"
        PCI 41:00.1 (Ethernet)
          Net "enp65s0f1np1"
          OpenFabrics "mlx5_1"
      PCIBridge
        PCI 42:00.0 (Ethernet)
          Net "enp66s0f0np0"
          OpenFabrics "mlx5_2"
        PCI 42:00.1 (Ethernet)
          Net "enp66s0f1np1"
          OpenFabrics "mlx5_3"L3 L#7 (16MB)
```

- For PCIe devices, you can also probe `sysfs` to determine the NUMA node: `cat /sys/bus/pci/devices/0000\:xx\:00.x/numa_node`

Use `cat /sys/devices/system/node/node*/meminfo | grep HugePages` to determine the hugepage memory per NUMA. For example:

```
Node 0 HugePages_Total:     12
Node 0 HugePages_Free:      12
Node 0 HugePages_Free:       0
Node 1 HugePages_Total:     12
Node 1 HugePages_Free:      12
Node 1 HugePages_Surp:       0
```

You can also use the `dpdk` utility `dpdk-hugepages.py -s` command to determine this.

3.2.2.2 Additional Tuning Options

You can also exercise the following tuning knobs to avoid kernel noise. These configurations indirectly affect DPDK workloads because isolated CPUs only prevent user applications from using the dedicated cores, but all kernel and interrupt processing can be triggered on the DPDK dedicated cores. These additional settings allow further fine tuning for asynchronous events. You will need `sudo` privileges to run the following commands:

- `swapoff -a; ufw disable`
- `echo 0 | tee /proc/sys/vm/zone_reclaim_mode`
- `echo 1 | /proc/sys/vm/drop_caches`
- `systemctl disable irqbalance`
- `sysctl -w kernel.sched_migration_cost_ns=5000000`
- `sysctl -w kernel.sched_min_granularity_ns=10000000`

- `echo 0 > /sys/kernel/mm/ksm/run`
- `echo "never"> /sys/kernel/mm/transparent_hugepage/enabled`
- `echo "never"> /sys/kernel/mm/transparent_hugepage/defrag`
- `echo 0 > /sys/kernel/mm/transparent_hugepage/khugepaged/defrag`
- `echo -1 > /proc/sys/kernel/sched_rt_period_us`
- `echo -1 > /proc/sys/kernel/sched_rt_runtime_us`
- `echo 10 > /proc/sys/vm/stat_interval`

Along with this, disable the watchdog to reduce overhead:

- `echo 0 > /proc/sys/kernel/watchdog_thresh`
- `echo 0 > /proc/sys/kernel/watchdog`
- `echo 0 > /proc/sys/kernel/nmi_watchdog`

You can also perform the following system cleanup:

- `journalctl --vacuum-time=1d`
- `apt clean; apt autoremove` (For Ubuntu)
- `yum clean all` (For RHEL)

3.2.2.3 Disable Services

You may stop the following optional services if they are not needed:

- `service cryptdisks stop`
- `service cups stop`
- `service mdadm stop`
- `service whoopsie stop`
- `service ufw stop`
- `service speech-dispatcher stop`
- `service ModemManager stop`
- `service lightdm stop`
- `service gdm3 stop`
- `systemctl stop irqbalance`
- `systemctl disable irqbalance`
- `systemctl –w vm.zone reclaim`

This page intentionally left blank.

Chapter 4 DPDK

This chapter explains how to configure DPDK on servers powered by AMD EPYC 8004 Series Processors.

4.1 Prerequisites

Please see Getting Started Guide for Linux* for information on building DPDK libraries along with the Linux prerequisites. Install all the required packages to compile DPDK.

4.2 Compilation

- **Library mode:** static (recommended for best performance)
- **For a native build:**
```
CC=gcc meson --default-library=static amd_siena_linuxapp_gcc -Dc_args="-march=znver4 -
Ofast"; ninja -C amd_siena_linuxapp_gcc install; ldconfig
```

Note: AMD EPYC 8004 Series Processors will need GCC version 12.3 or onwards and clang 16 or onwards to fully leverage the micro-architecture features and optimizations.

- **For building applications with DPDK libraries:**
  - **Static Library mode:**
    ```
    gcc test.c $(pkg-config --static --cflags --libs libdpdk) -o test.exe
    ```
  - **Shared Library mode:**
    ```
    gcc test.c $(pkg-config --cflags --libs libdpdk) -o test.exe
    ```

4.3 Environmental Abstraction Layer (EAL) Options

Select following EAL parameters should be selected based on AMD EPYC platform. See EAL Parameters* for more information.

Function	Command	Description
Logical core number	`-l <core list>` (or) `-c <core mask>`	List of cores to run on. (or) Set the hexadecimal bitmask of the cores to run on.
Number of memory channels	`-n <number of channels>`	Set the number of memory channels to use.
Master logical core number	`--main-lcore`	Initialize EAL and load environment parameters.
Socket NUMA Huge page memory allocation	`--socket-mem`	Allows fine grain control of huge page allocation for a given NUMA node.
Force maximum SIMD bitwidth	`--force-max-simd-bitwidth=512`	The internal default setting for libraries and PMD is to use 256b SIMD operation. Force 512bit mode improves the application performance by making use of AVX512 optimization in both the libraries and PMDs

Table 4-1: EAL parameter options

4.4 NIC-Specific Tunable Settings

Please see the DPDK NIC performance reports from your NIC vendor(s) for firmware and driver version requirements and for recommended settings. Please also refer to the vendor-specific DPDK Driver* documentation for specific functionalities (e.g., SR-IOV).

4.4.1 Broadcom P2100G

None. Please see BNXT PMD* for more details.

4.4.2 Intel E810

Per the DPDK Intel performance reports, AMD recommends building the DPDK PMD with 16B descriptors for optimal performance by passing the `-DRTE_LIBRTE_ICE_16BYTE_RX_DESC` option. Also, force the SIMD bit width to 512 using the EAL option `--force-max-simd-bitwidth`. You can optimize the following PMD options for:

- Low RX latency: `rx_low_latency=1`
- Normal operation: none

4.4.3 Mellanox Cx-7:

The Mellanox PMD makes use of `libibverbs` for device control and configuration via PMD. Thus, the device need not be bound to `vfio-pci` or `igb_uio`. Linux control tools control and maintain the network Interface. To use the network interface for DPDK applications, enable specific configurations for high-throughput packet transfer, as described in NVIDIA MLX5*.

For 64B throughput (max MPPs) using `testpmd` in IO forward mode with MLX5 PMD in vector mode (200Gbps):

- 68 MPPS can be achieved with 1 logical core and 1 RX/TX queue without SMT.
- 82 MPPS can be achieved with 1 logical core and 1 RX/TX queue with SMT enabled.

This page intentionally left blank.

Chapter 5

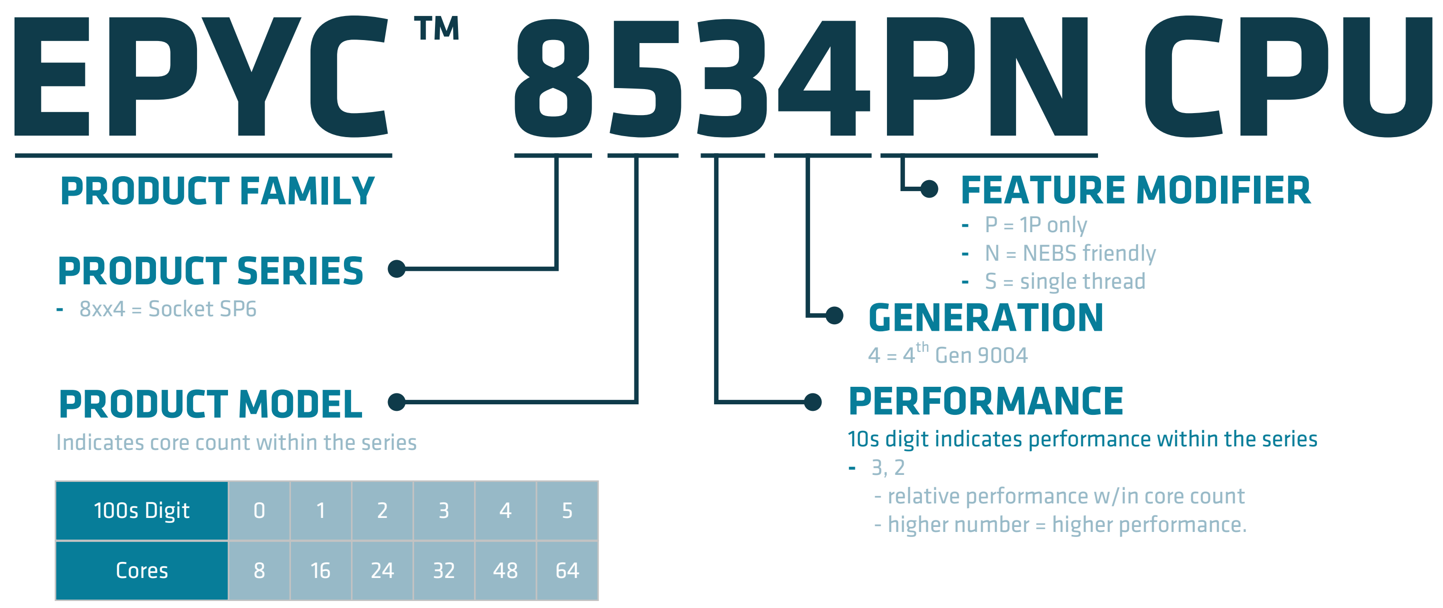
Glossary

- **ACPI** - Advanced Configuration and Power Interface
- **BIOS** - Basic Input/Output System
- **BMC -** Baseboard Management Controller
- **CCD** – Core Complex Die
- **CCX** – Core Complexes
- **cTDP** - Configurable Thermal Design Power
- **DIMM** – Dual In-line Memory Module
- **DPC** – DIMMs Per Channel
- **DRAM** – Dynamic Random-Access Memory
- **LLC** - Last Level Cache
- **NIC** – Network Interface Card
- **NUMA** - Non-Uniform Memory Access
- **PMD -** Poll Mode Driver
- **PPL** - Package Power Limit
- **OPN** – Orderable Part Number
- **OS** - Operating System
- **SLIT** - System Locality Information Table
- **SMT** - Symmetric Multithreading
- **SRAT** - System Resource Affinity Table
- **TCO** – Total Cost of Ownership
- **TDP** - Thermal Design Power

This page intentionally left blank.

Chapter 6 Processor Identification

Figure 6-1 shows the processor naming convention for AMD EPYC 8004 Series Processors and how to use this convention to identify particular processors models:



100s Digit	0	1	2	3	4	5
Cores	8	16	24	32	48	64

Figure 6-1: AMD EPYC SoC naming convention

6.1 CPUID Instruction

Software uses the `CPUID` instruction (`Fn0000_0001_EAX`) to identify the processor and will return the following values:

- **Family:** 19h identifies the “Zen 4” architecture
- **Model:** Varies with product. For example, EPYC Model 10h corresponds to an “A” part “Zen 4” CPU.
  - **8xx4:** Models A0h–AFh
- **Stepping:** May be used to further identify minor design changes

For example, `CPUID` values for Family, Model, and Stepping (decimal) of 25, 17, 1 correspond to a “B1” part “Zen 4” CPU.

6.2 New Software-Visible Features

AMD EPYC 8004 Series Processors introduce several new features that enhance performance, ISA updates, provide additional security features, and improve system reliability and availability. Some of the new features include:

- 5-level Paging
- AVX-512 instructions on a 256-bit datapath, including BFLOAT16 and VNNI support.
- Fast Short Rep STOSB and Rep CMPSB

Not all operating systems or hypervisors support all features. Please refer to your OS or hypervisor documentation for specific releases to identify support for these features.

Please also see the latest version of the *AMD64 Architecture Programmer's Manuals* or *Processor Programming Reference (PPR) for AMD Family 19h*.

6.2.1 AVX-512

AVX-512 is a set of individual instructions supporting 512-bit register-width data (i.e., single instruction, multiple data [SIMD]) operations. AMD EPYC 8004 Series Processors implement AVX 512 by "double-pumping" 256-bit-wide registers. AMD's AVX-512 design uses the same 256-bit data path that exists throughout the Zen4 core and enables the two parts to execute on sequential clock cycles. This means that running AVX-512 instructions on AMD EPYC 8004 Series will cause neither drops on effective frequencies nor increased power consumption. On the contrary, many workloads run more energy-efficiently on AVX-512 than on AVX-256P.

Other AVX-512 support includes:

- Vectorized Neural Network Instruction (VNNI) instructions that are used in deep learning models and accelerate neural network inferences by providing hardware support for convolution operations.
- Brain Floating Point 16-bit (BFLOAT16) numeric format. This format is used in Machine Learning applications that require high performance but must also conserve memory and bandwidth. BFLOAT16 support doubles the number of SIMD operands over 32-bit single precision FP, allowing twice the amount of data to be processed using the same memory bandwidth. BFLOAT16 values mantissa dynamic range at the expense of one radix point.