AMD EPYC

TUNING GUIDE AMD EPYC 7003

Data Plane Development Kit

Publication	57083
Revision	3.1
Issue Date	July, 2022

Date	Version	Changes
Mar, 2021	2.0	Initial public release
Mar, 2022	3.0	Added AMD 3D V-Cache™ information
July, 2022	3.1	Updated installation/configuration information, minor errata corrections.

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 server management Interface (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

Vipin Varghese, Sivaprasad Tummala, and Keesang Song.

Note: All of the settings described in this Tuning Guide apply to all AMD EPYC 7003 Series Processors with or without AMD 3D V-Cache™ except where explicitly noted otherwise.

Table of Contents

Chapter 1 Introduction ---- 1

1.1 AMD EPYC 7003 Series Processors ... 2
1.2 Operating Systems ... 2
1.3 Data Plane Development Kit ... 3
1.3.1 DPDK Core Components ... 3

Chapter 2 System Configuration Best Practices 5

2.1 Memory Configuration ... 5
2.2 BIOS Performance Settings ... 6
2.2.1 Recommended Common BIOS Settings ... 6
2.2.2 Advanced BIOS Settings ... 8
2.3 NIC Specific Tunable Settings ... 10
2.3.1 NUMA Affinity ... 10
2.3.2 PCIe Relaxed Ordering ... 11

Chapter 3 Software Configuration Best Practices 13

3.1 Linux OS Configuration and Kernel parameters ... 13
3.1.1 Linux OS & Kernel Considerations ... 13
3.1.2 Linux OS Configuration and Kernel Parameters ... 13
3.2 DPDK Environment Setup ... 14
3.2.1 Prerequisites ... 14
3.2.2 Compilation ... 14
3.2.3 DPDK Environmental Abstraction Layer (EAL) Options ... 15
3.2.4 NIC Tunable Settings ... 16

Chapter 4 Resources ---- 17

Chapter 5 Glossary ---- 19

This page intentionally left blank.

Chapter 1 Introduction

This tuning guide describes parameters that can optimize DPDK application performance on servers built with both AMD EPYC™ 7003 Series Processors and a Network Interface Card (NIC). Default OEM system configurations and BIOS settings may not provide the best possible performance for all workloads on all operating system. This guide discusses best practices for:

- System configuration:
  - Memory configuration, PCIe® subsystem
  - BIOS settings that can impact performance
  - NIC-specific tunable settings
- Software configuration:
  - Linux OS configuration and kernel parameters
  - DPDK Environment setup and EAL (Environmental Abstraction Layer) command line 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.

1.1 AMD EPYC 7003 Series Processors

AMD EPYC 7003 Series Processors are built with the leading-edge "Zen 3" core and AMD Infinity Architecture. The AMD EPYC SoC offers a consistent set of features across 8 to 64 cores. Each 3rd Gen EPYC processor consists of up to eight Core Complex Dies (CCD) and an I/O Die (IOD). Each CCD contains one CCX, meaning that each CCD contains up to 8 "Zen 3" cores. The CCDs connect to the I/O Die (IOD) to access memory, I/O, and each other via AMD Infinity Fabric™ technology. 3rd Gen AMD EPYC processors support up to 8 memory channels, 4 TB of high-speed memory per socket, and 128 lanes of PCIe® Gen 4.

3rd Gen AMD EPYC Series processors are built with the following specifications:

3rd Gen AMD EPYC 7003 Series Processors	
Process technology	7nm
Max Processor speed	4.1 GHz
Max number of cores	64
Max memory speed	3200 MT/s
Max memory capacity	4 TB per socket
Peripheral Component Interconnect	128 lanes (max) PCIeGen4

Table 1-1: Table 1 AMD EPYC™ 7003 Series Processors

Some AMD EPYC™ 7003 Series Processors introduce AMD's new 3D V-Cache die stacking technology that enables denser, more efficient chiplet integration. AMD 3D Chiplet architecture stacks L3 cache tiles vertically to provide 768 MB of L3 cache per socket and up to 96MB of L3 cache per CCD, while still providing socket compatibility with existing AMD EPYC 7003 Series Processors. Applications that take advantage of AMD 3D V-cache can see significant performance gain and lower overall TCO.

See *Overview of AMD EPYC™ 7003 Series Processors Microarchitecture* (available from AMD EPYC Tuning Guides) to learn more about the AMD EPYC 7003 Series Processor microarchitecture.

1.2 Operating Systems

AMD recommends using the latest available OS version. See AMD EPYC™ 7003 Series Processors Minimum Operating System (OS) Versions for detailed OS version information.

1.3 Data Plane Development Kit

The Data Plane Development Kit (DPDK) is a set of open-source software libraries and drivers hosted by the Linux Foundation that provide a lightweight framework for getting packets directly to/from applications. These libraries accelerate packet-processing workloads running on all major CPU architectures by allowing incoming network packets to transition to user space with no overhead for memory copying. Bypassing the Linux kernel maximizes performance by eliminating throughput and latency expense of context switching between user space and kernel space. The DPDK does this by running a Poll-Mode Driver (PMD) in user space that continually checks incoming packet queues for new data. The DPDK libraries only provide minimal in-application packet operations but enable receiving and sending packets with a minimum number of CPU cycles. Certain use cases can see up to wire speed performance by reducing bulk depending on the processing depth. Please visit dpdk.org for additional information.

1.3.1 DPDK Core Components

The DPDK consists of several core components:

- **Memory Manager:** Allocates pools of objects in memory. A pool is created in hugepages memory space. This pool uses a ring to store free objects and provides an alignment helper to pad objects by spreading them equally across all DRAM channels.
- **Buffer Manager:** Significantly reduces the time the OS spends allocating and deallocating buffers by pre-allocating fixed-size buffers are stored in memory pools.
- **Queue Manager:** Implements safe lockless and fixed size queues (instead of spin-locks) that allow different software components to process packets while avoiding unnecessary wait times.
- **Flow Classification:** Greatly improves throughput by incorporating streaming SIMD Extensions to produce a hash based on tuple information to quickly place packets into flows for processing.
- **Poll Mode Drivers:** Speed up the packet pipeline by allocating a CPU core to constantly poll for new packets instead of relying on asynchronous, interrupt-based signaling mechanisms.
- **HugePages:** Boost performance by creating pre-allocated memory pages that eliminate memory replacement and reduce the size of the page table TLB and frequency of misses.

Platform optimizations include configuring NUMA memory and I/O (NIC Cards) to boost throughput and reduce latency using affinity. You can do this via BIOS settings, PCIe subsystem configuration, memory configuration, OS configuration, and kernel parameters.

This page intentionally left blank.

Chapter 2 System Configuration Best Practices

2.1 Memory Configuration

Proper memory subsystem configuration is crucial for all performance testing. I/O transfers data into and out of memory. I/O bandwidth therefore cannot exceed memory subsystem capabilities.

Consecutive memory blocks (often called cache lines) are read from the same memory bank. Software that reads consecutive memory must normally wait for a memory transfer to complete before starting the next memory access. CPUs increase available memory bandwidth by using memory interleaving to place consecutive memory blocks in different banks that collectively contribute to overall memory bandwidth, thus increasing throughput and lowering latency.

AMD EPYC processors include four integrated circuits (called dies) in one System on Chip (SoC) that occupies a single socket. Each die has two memory controllers, and each socket has eight memory controllers. AMD EPYC processors support multiple interleave methods that are only active when the memory controllers are controlling memory DIMMs. Many OEMs call this channel interleaving, which interleaves between the two memory controllers in a die. A single socket thus accommodates up to four interleave pairs. Installing eight DIMMs achieves optimal performance for many applications. AMD therefore recommends populating all eight memory channels per CPU socket with DIMMs of equal capacity, which allows eight-way interleaving for optimal performance in most cases.

Achieving maximum memory bandwidth on modern CPUs requires populating at least one DIMM in every DDR channel. Servers based on AMD EPYC™ 7003 Series Processors include eight DDR4 memory channels per CPU socket. Therefore:

- Populate all eight memory channels with DIMMs of equal sizes and speeds for 1 DIMMs per Channel (DPC) configuration on a single-socket system. This runs the memory at 3200 MHz.
- Populate all 16 memory channels with DIMMs of equal sizes and speeds for 2 DPC configuration on a dual-socket system. This runs the memory at 3200 MHz.

Either configuration achieves maximum memory bandwidth and lowest latency in SoC P0 Power State mode with 1600 MHz MEMCLK and 1600 MHz FCLK.

Please see the latest version of Memory Population Guidelines for AMD EPYC 7003 Series Processors for additional information.

OEM servers supporting AMD EPYC 7003 series processors are built to either support previous generation of AMD EPYC 7002 series processors or are specifically designed for the AMD EPYC 7003 series processors. Contact your OEM vendor to determine the characteristics of your servers.

2.2 BIOS Performance Settings

The application profile influences platform configuration, but you can tune several areas of the platform BIOS to boost better overall performance. Table 2 describes the BIOS options that most impact the DPDK PMD application. See the latest version of Workload Tuning Guide for AMD EPYC™ 7003 Series Processors for additional BIOS settings.

2.2.1 Recommended Common BIOS Settings

Name	Recommended Value	Description
Local APIC Mode	X2APIC	Scales beyond 255 hardware threads but is not supported by some legacy OS versions; verify support before enabling this option. Recommended even for configurations with fewer than 256 logical cores.
SMT Control	Disable	Disables Symmetric Multithreading (SMT), which allows one hardware thread per core. X2APIC must be enabled to support more than 255 threads.
NUMA Node per Socket (NPS)	NPS [1|2|4]	This setting manages a tradeoff between minimizing local memory latency for NUMA-aware or highly parallelizable workloads vs. maximizing per-core memory bandwidth for non-NUMA-friendly workloads. The NPS setting determines the number of NUMA nodes to split the memory channels between. Higher number indicates fewer memory channels per NUMA node, which lowering both memory throughput for a particular NUMA node and memory latency. The available NPS settings are: • **NPS1:** Maximum per-core memory bandwidth without NUMA affinity. • **NPS2:** Compromise between memory latency and memory bandwidth for when using 100/200+ Gbps network adapters. • **NPS4:** If the memory has sufficient bandwidth, then this is the recommended setting for optimum latency and throughput, especially for the maximum aggregated throughput from 3+ multiple high-speed(100GbE+) networking cards. Begin with NPS=1 for any network application, and then fine tune with NPS=2 or NPS-4 for latency-sensitive workloads. This setting is independent of ACPI SRAT L3 Cache as NUMA Domain. When ACPI SRAT L3 Cache as NUMA Domain is enabled, this setting now determines the memory interleaving granularity, as follows: • **NPS1:** All eight memory channels are interleaved. • **NPS2:** Every four channels are interleaved with each other. • **NPS4:** every pair of channels is interleaved.

Table 2-1: Recommended common BIOS settings

IOMMU	Enable	Enables the IOMMU (Input–Output Memory Management Unit) that connects a direct-memory-access–capable (DMA-capable) I/O bus to the main memory. An IOMMU creates a virtual address space for the device, where each I/O Virtual Address (IOVA) may translate to different addresses in the physical system memory. When the translation is completed, the devices are connected to a different address within the physical system's memory. Without an IOMMU, all devices have a shared, flat view of the physical memory because they lack memory address translation. With an IOMMU, devices receive the IOVA space as a new address space, which is useful for device assignment. Enabling the IOMMU allows devices to present separate IRQs for each attached device instead of one IRQ for the subsystem. IOMMU also 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 (e.g., `amd_iommu=pt` in the grub file on the Linux® kernel command line). When in pass-through mode, the adapter does not need to use DMA translation to the memory, which improves performance
Determinism Control	Manual	Enable the **Determinism Slider** control.
Determinism Slider	Power	Ensure maximum performance levels for each CPU in a large population of identically-configured CPUs by throttling CPUs only when they reach the same cTDP. See Power/Performance Determinism for more details.
ACPI SRAT L3 Cache as NUMA Domain	Disable	Do not report each L3 cache as a NUMA domain to the OS This setting controls automatic or manual generation of distance information in the ACPI System Locality Information Table (SLIT) and NUMA proximity domains in the System Resource Affinity Table (SRAT).

Table 2-1: Recommended common BIOS settings

Please see the latest versions of the OS network tuning guides.

2.2.2 Advanced BIOS Settings

Default BIOS options generally to produce the best overall performance for typical generic workloads. Individual workloads can have different requirements, and the BIOS defaults may therefore nor be optimal for your needs.

Name	Recommended Value	Description
PCIe ARI Support [SRIOV]	Enable	Enables ARI (Alternative Routing ID) interpretation support for SR-IOV SR-IOV requires enabling PCIe® ARI on both root complexes and endpoints. ARI devices interpret the PCI address as an 8-bit function number instead of a 3-bit function number, and the device number is implied to be 0.
PCIe Ten Bit Tag	Enable	Enables PCIe 10-bit tags for supported devices. An adapter must support 10-bit extended tags to achieve full bandwidth on PCIe Gen 4, This allows a 3x increase over the previous number of non-posted requests and allows the adapter to achieve higher performance. PCIe Gen 4 devices that do not support 10-bit extended tags can cause issues during boot. Disabling this feature allows the server to boot if the adapter is having issues.
Preferred I/O	Disable	Disables Preferred I/O. If you enable this option, then the **Preferred I/O Bus** setting must specify the bus number. This setting allows the system to prioritize traffic for one PCIe I/O device per system and is available on both single- and dual-socket systems.
Enhanced Preferred I/O	Enable	Enables enhanced Preferred I/O mode for the bus number specified by Preferred I/O bus, which gives priority to the I/O devices attached to the PCIe slot(s) associated with only one enabled Root Complex for I/O transactions. The NBIO local clock frequency (LCLK) for that Root Complex will be set to a fixed frequency and will not be affected by global Dynamic Power Management (DPM), thus further reducing PCIe latency. This option may help latency-sensitive workloads achieve higher bandwidth and reduce packet loss during a zero packet loss test, such as RFC2544. By default, PCIe uses ordering semantics to maintain coherence by enforcing strict data packet ordering, but there are two other options: • Relaxed Ordering (RO) • ID-Based Ordering (IDO)

Table 2-2: Advanced BIOS settings

APB Disable (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.
Fixed SOC P-State	P0	• **P0:** Highest-performing Infinity Fabric P-state • **P1:** Next-highest-performing Infinity Fabric P-state • **P2:** Next highest-performing Infinity Fabric P-state • **P3:** Minimum Infinity Fabric Power P-state
LCLK setting	593 MHz	Sets RC (Root Complex) LCLK frequency. This is an internal clock within the AMD EPYC 7003 Series Processor that controls the communication frequency between the Northbridge I/O (NBIO) logic and the rest of the I/O die (IOD). The PCIe root complex (RC) is part of the NBIO, and each RC has its own independently-controlled LCLK that controls the speed of upstream and downstream communication.
L1/L2 Stream HW Prefetcher	Enable	Enables the L1/L2 Stream HW Prefetcher.
Core C-States	Enable(C0/C1)	Enables CPU C-States.
DF C-States	Disable	Disables DF (Data Fabric) C-states.
xGMI Force Link Width Control	Forced	Forces the XGMI link width control.
xGMI Force Link Width	x 2	Forces the XGMI link width to the minimum width.
TSME	Disable	Disables TSME (Transparent Secure Memory Encryption). This feature provides hardware-based encryption of all data stored on system DIMMs at a small increase in memory latency. This encryption is invisible to the OS.

Table 2-2: Advanced BIOS settings (Continued)

2.3 NIC Specific Tunable Settings

2.3.1 NUMA Affinity

I/O intensive workloads can perform better when placed on the same NUMA node that connects to the I/O device used. For example, place a networking-intensive operation on the socket that the NIC connects to. Tools such as the Linux `lstopo(hwloc)` command can help determine the connectivity between PCI devices and sockets by displaying how the OS sees NUMA nodes, caches, cores, and PCI devices. Figure 2-1 shows `lstopo(hwloc)` on a dual-socket system with 64 cores per socket configured with NPS=4 and 'ACPI SRAT L3 Cache as NUMA Domain enabled:

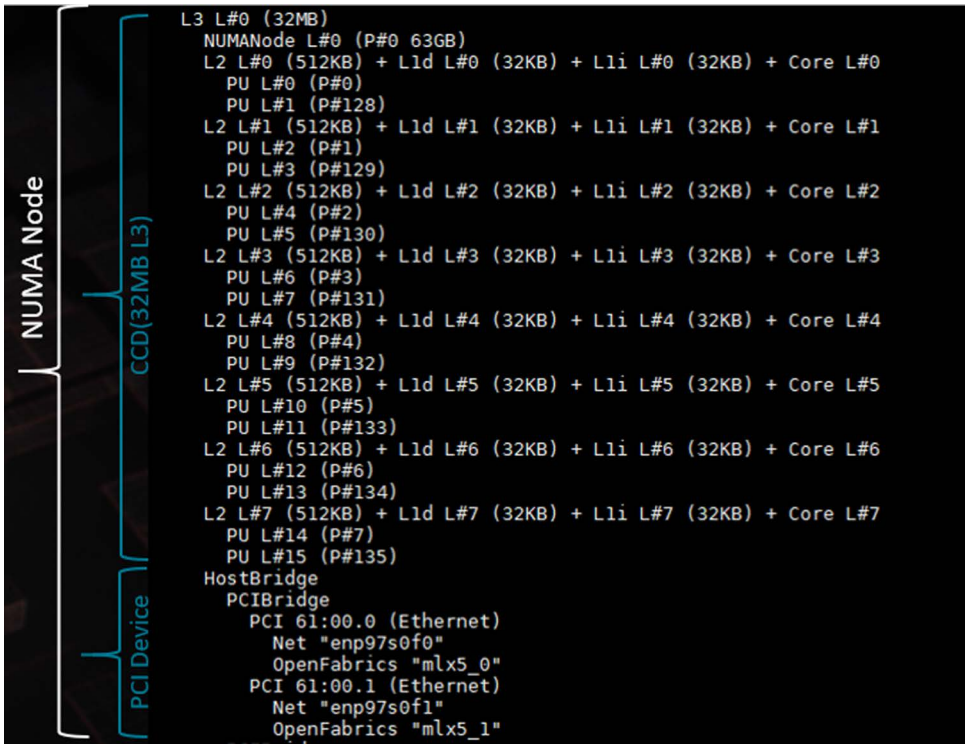

Figure 2-1: Sample lstopo(hwloc) output on a dual-socket system

In Figure 2-1:

- 8 cores per L3 cache are grouped together along with the CPU IDs associated with each 32 MB L3 Cache, - 0,1,2,3. Each group of 8 cores per L3 represent a single CCX/CCD.
- The 2 CCXs/CCDs are logically grouped under each NUMA node.
- Each NUMA node consists of 128GB of memory.

Note: SMT is not enabled on this system.

2.3.2 PCIe Relaxed Ordering

Relaxed PCIe ordering helps maximum throughput performance, especially with high-speed PCIe 4 NICs when targeting memory attached to AMD EPYC 7003 Series Processors. You can do this by either:

- Using NIC vendor-provided communication libraries that enable relaxed ordering using a new API. This is the preferred option.
- Forcing all traffic to use relaxed ordering. This can break some optimized communications that rely on memory being written and visible in a given order in memory.

Note: If you plan to use single root I/O virtualization (SR-IOV) for RHEL7, then please see the RedHat Network Guide* *for additional information.*

This page intentionally left blank.

Chapter 3 Software Configuration Best Practices

This chapter outlines software configuration options that affect application performance, including tuning Linux operating system and kernel parameters for optimal DPDK application performance.

3.1 Linux OS Configuration and Kernel parameters

Using DPDK to optimize data plane performance for network applications requires understanding:

- How DPDK uses the compute node hardware such as the CPU, NUMA nodes, memory, and NICs, as well as considerations for determining the Linux OS configuration.
- Underlying kernel parameters based on allocated compute resources and targeted performance emphasis.

3.1.1 Linux OS & Kernel Considerations

Several AMD EPYC-related Linux patches have been released. You can apply these patches manually or deploy at least the minimum-supported OS version:

- **Red Hat®/ CentOS:** v 8.3 with Kernel 3.10.0-1062.15.1.el7 or later.
- **Canonical®:** Ubuntu 18.04.5 or Ubuntu 20.04 with default Kernel version 5.4 or later.
- **SUSE®:** SLES 12 SP5 1/SLES 15 SP2 with Kernel 4.12.14-122.17 or later.

3.1.2 Linux OS Configuration and Kernel Parameters

Add the following lines to `/etc/default/grub`, and then use the `isolcpus` Linux kernel parameter to isolate them from the Linux scheduler to reduce context switches by preventing non-DPDK workloads from running on reserved cores. See "NUMA Affinity" on page 10 for instructions on selecting the correct CPU cores for affinity. You should also use IOMMU in pass through (`iommu=pt`) mode to improve host performance by disabling the DMAR to the memory. The total number of available hugepages per NUMA socket depends on the BIOS NPS setting.

```
processor.max_cstate=1"
```

1. Disable the interrupt load balance daemon if necessary. Move all IRQs to the far NUMA node, and disable IRQ balance.

```
"IRQBALANCE_BANNED_CPUS=$LOCAL_NUMA_CPUMAP irqbalance --oneshot" systemctl stop irqbalance
systemctl disable irqbalance
sysctl -w vm.zone_reclaim_mode=0; sysctl -w vm.swappiness=0
```

2. Update GRUB.

```
update-grub
```

3. Reboot the host to apply the changes.

```
reboot
```

4. Verify that 64x 1GB Hugepages are evenly distributed across all NUMA nodes.

```
cat /sys/devices/system/node/node$N/hugepages/hugepages-1048576kB/nr_hugepages
cat /sys/devices/system/node/node*/meminfo |grep HugePages_Free
```

5. Check the PCI device related NUMA node id.

```
cat /sys/bus/pci/devices/0000\:xx\:00.x/numa_node
```

3.2 DPDK Environment Setup

This section explains how to configure DPDK an AMD EPYC 7003 platform.

3.2.1 Prerequisites

Note: Certain NICs and other devices do not implement pure poll-mode drivers. If you have one of these devices, then you will need to install the vendor-recommended library packages with their recommended drivers and firmware.

3.2.2 Compilation

The recommended configurations are

- **Library Mode:** Static (for best performance)
- **DPDK threads:**
  - 128 for single-socket systems.
  - 256 for dual-socket systems.
- **Compiler flags:** Use `znver3` if supported by your version of GCC. If your version of GCC does not support `znver3`, then omit the `c_args` from the meson command line.

For a native build:

- **Single socket:**
  ```
  CC=gcc meson –default-library=static amd_linuxapp_gcc -Dmax_lcores=128 -Dc_args="–
  march=znver3 -mtune=znver3"; ninja -C amd_linuxapp_gcc install; ldconfig
  ```
- **Dual-socket:**
  ```
  CC=gcc meson –default-library=static amd_linuxapp_gcc -Dmax_lcores=256 -Dc_args="–
  march=znver3 -mtune=znver3"; ninja -C amd_linuxapp_gcc install; ldconfig
  ```

Applications built with DPDK libraries can leverage `$(pkg-config –static –libs –cflags libdpdk)`.

You can make the `LCORE` option permanent by either;:

- Building with `-Dmax_lcore=256`.
- Changing `max_lcores` to 256 in `meson_options.txt`, as follows:

diff meson_options.txt_org meson_options.txt

```
< option('max_lcores', type: 'string', value: 'default', description:
---
> option('max_lcores', type: 'string', value: '256', description:
```

Please see the Getting Started Guide for Linux* for installation instructions.

Be sure to build with -Dmax_lcore=256 or change ‘max_lcores” to 256 in meson_options.txt, as follows:

```
# diff meson_options.txt_org meson_options.txt
< option('max_lcores', type: 'string', value: 'default', description:
---
> option('max_lcores', type: 'string', value: '256', description:
```

3.2.3 DPDK Environmental Abstraction Layer (EAL) Options

Review the required command line options for the DPDK EAL environment and use optimal configuration to run a DPDK application on an AMD EPYC-based system.

Function	Command	Description
Choose logical cores	`-l <core list>` or `-c <core mask>`	List of cores to run on Or Hexadecimal bitmask selection of logical cores to run on.
Number of memory channels	`-n <number of channels>`	Force the user-desired number of memory channels.
Master logical core number	`--master-lcore`	Initialize EAL and load environment parameters.
Force maximum SIMD bitwidth	`--force-max-simd-bitwidth=256`	Limits which vector paths, if any, are taken, because any paths taken must use a bitwidth below the max bitwidth limit.

Table 3-1: EAL CLI options

3.2.4 NIC Tunable Settings

Please refer to your NIC vendor performance reports for accurate driver and firmware information, plus any settings required for optimal performance.

- **Broadcom P2100:** Achieves 200Gbps with 512B. AMD recommends using SMT logical cores with 2RX-TX for 64B.
- **Intel E810:** AMD EPYC 7003 Series Processors support AVX2. AMD recommends using SMT logical cores with 2 RX-TX for 64B.
- **Mellanox CX-6:** Uses a port representative. Disable `autoneg`, set to 100G speed, and disable pause via `ethtool`. Use `mlxconfig` to set CQE compression and PCI relaxed ordering. This NIC can achieve:
  - 90 Mpps for 64B with 1 logical core with 4 RX-TX queues.
  - 145Mpps with 8 logical cores with 8RX-TX queues.
  - 149Mpps with 10 logical cores with 13 RX-TX queues.

Note: Mellanox NICs support up to 16 hardware RX-TX queues. Do not set more than 16 RX-TX queues, because this will cause software multiplexing that will reduce NIC performance.

Chapter 4 Resources

- DPDK Debug & Troubleshoot Guide*
- Getting Started Guide for Linux*
- DPDK EAL Parameters*
- Memory Population Guidelines for AMD EPYC 7003 Series Processors – Login required.
- Socket SP3 Platform NUMA Topology for AMD Family 19h Models 00h–0Fh - Log in required.
- From AMD EPYC Tuning Guides:
  - *Workload Tuning Guide for AMD EPYC™ 7003 Series Processors*
  - *Linux® Network Tuning Guide for AMD EPYC™ 7003 Series Processor Based Servers*
  - *Windows® Network Tuning Guide for AMD EPYC™ 7003 Series Processor Based Servers*
  - *VMware® Network Tuning Guide for AMD EPYC™ 7003 Series Processor Based Servers*
- NIC vendor-specific tuning guide for AMD EPYC platforms.

This page intentionally left blank.

Chapter 5 Glossary

- **ACPI** - Advanced Configuration and Power Interface
- **BIOS** - Basic Input/Output System
- **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
- **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.