AMD EPYC

TUNING GUIDE
AMD EPYC 9004

Artificial Intelligence Machine Learning (AI/ML)

Publication	58205
Revision	1.1
Issue Date	December, 2023

Date	Version	Changes
Mar, 2023	0.1	Initial NDA partner release
Jun, 2023	1.0	Initial public release
Dec, 2023	1.1	Updated AI/ML configuration settings

Audience

This document helps you tune systems powered by 4th Gen AMD EPYC™ processors for Artificial Intelligence/Machine Learning (AI/ML) workloads but is not an all-inclusive guide. Further, some items referred to herein may have different names across various OEM systems, such as OEM-specific BIOS settings. Every AI/ML workload also has unique performance characteristics. This document suggests items to focus on when performing additional, application-specific tuning as a starting point for performing your own performance testing and additional tuning for your specific workload. Performing effective AI/ML tuning requires familiarity with server configuration. You should also:

- Have admin access to the server's management interface (BMC) and be familiar with the server's management interface.
- Have admin OS access and be familiar with the OS-specific configuration, monitoring, and troubleshooting tools.

Authors

Mukund Kumar and Dinesh Chitlangia

Note: All of the settings described in this Tuning Guide apply to all AMD EPYC 9004 Series Processors of all core counts with or without AMD 3D V-Cache™ except where explicitly noted otherwise.

Note: AMD recommends installing all relevant BIOS, OS, application, and other updates/patches as they become available to help enhance performance and security.

Table of Contents

Chapter 1

Introduction

Artificial Intelligence (AI) and Machine Learning (ML) have become an integral part of many modern software applications, from chatbots to self-driving cars. The increase in complexity and size of AI/ML models makes optimizing the performance of the underlying hardware critical to ensure fast and efficient processing of model training or inferencing tasks. The 4th Gen AMD EPYC™ CPU architecture, NUMA layout, and thread counts, together with software optimizations play crucial roles when executing AI/ML algorithms Performance tuning can greatly improve the overall performance of an AI/ML application.

This Tuning Guide explores the key factors that affect CPU performance when running AI/ML workloads and provides practical tips and techniques for optimizing CPU performance. It includes topics such as the AMD EPUC CPU architecture, NUMA, BIOS, software-specific parameters, and Linux OS knobs, and explains how these factors impact AI/ML performance. This Tuning Guide also provides guidance on how to measure performance metrics such as latency and throughput and how to identify and resolve common performance bottlenecks.

Software developers, data scientists, and IT professionals can use this Tuning Guide to gain the knowledge and tools needed to optimize the performance of your AI/ML workloads on 4th Gen AMD EPYC CPUs, thereby achieving faster and more efficient processing.

This page intentionally left blank.

Chapter 2

AMD EPYC™ 9004 Series Processors

AMD EPYC™ 9004 Series Processors represent the fourth generation of AMD EPYC server-class processors. This generation of AMD EPYC processors feature AMD's latest "Zen 4" based compute cores, next-generation Infinity Fabric, next-generation memory & I/O technology, and use the new SP5 socket/packaging.

2.1 General Specifications

AMD EPYC 9004 Series Processors offer a variety of configurations with varying numbers of cores, Thermal Design Points (TDPs), frequencies, cache sizes, etc. that complement AMD's existing server portfolio with further improvements to performance, power efficiency, and value. Table 1-1 lists the features common to all AMD EPYC 9004 Series Processors.

Common Features of all AMD EPYC 9004 Series Processors	
Compute cores	Zen4-based
Core process technology	5nm
Maximum cores per Core Complex (CCX)	8
Max memory per socket	6 TB
Max # of memory channels	12 DDR5
Max memory speed	4800 MT/s DDR5
Max lanes Compute eXpress Links	64 lanes CXL 1.1+
Max lanes Peripheral Component Interconnect	128 lanes PCIe® Gen 5

Table 2-1: Common features of all AMD EPYC 9004 Series Processors

2.2 Model-Specific Features

Different models of 4th Gen AMD EPYC processors have different feature sets, as shown in Table 1-2.

AMD EPYC 9004 Series Processor (Family 19h) Features by Model		
Codename	"Genoa"*	"Bergamo"*
Model #	91xx-96xx	97xx
Max number of Core Complex Dies (CCDs)	12	8
Number of Core Complexes (CCXs) per CCD	1	2
Max number of cores (threads)	96 (192)	128 (256)
Max L3 cache size (per CCX)	1GB (96 MB)♦	256 MB (16 MB)
Max Processor Frequency	4.4 GHz♦♦	3.15 GHz
Includes ♦AMD 3D V-Cache (9xx4X) and ♦♦high-frequency (9xx4F) models.		
*GD-122: The information contained herein is for informational purposes only, and is subject to change without notice. Timelines, roadmaps, and/or product release dates shown herein and plans only and subject to change. "Genoa" and "Bergamo" are codenames for AMD architectures, and are not product names.		

Table 2-2: AMD EPYC 9004 Series Processors features by model

2.3 Operating Systems

AMD recommends using the latest available targeted OS version and updates. Please see AMD EPYC™ Processors Minimum Operating System (OS) Versions for detailed OS version information.

2.4 Processor Layout

AMD EPYC 9004 Series Processors incorporate compute cores, memory controllers, I/O controllers, RAS (Reliability, Availability, and Serviceability), and security features into an integrated System on a Chip (SoC). The AMD EPYC 9004 Series Processor retains the proven Multi-Chip Module (MCM) Chiplet architecture of prior successful AMD EPYC processors while making further improvements to the SoC components.

The SoC includes the Core Complex Dies (CCDs), which contain Core Complexes (CCXs), which contain the "Zen 4"-based cores. The CCDs surround the central high-speed I/O Die (and interconnect via the Infinity Fabric). The following sections describe each of these components.

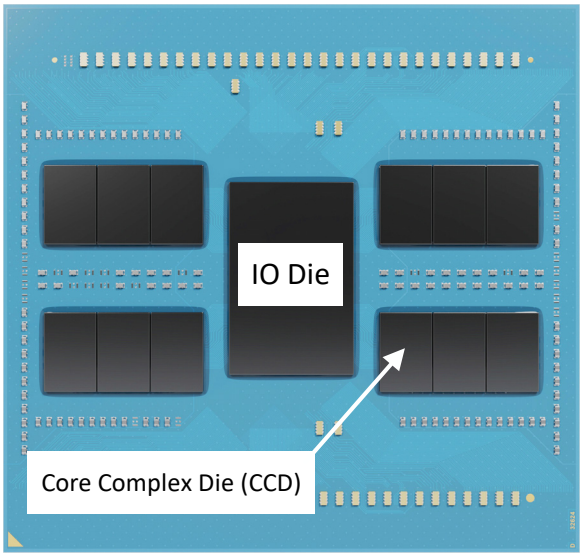

Figure 2-1: AMD EPYC 9004 configuration with 12 Core Complex Dies (CCD) surrounding a central I/O Die (IOD)

2.5 "Zen 4" Core

AMD EPYC 9004 Series Processors are based on the new "Zen 4" compute core. The "Zen 4" core is manufactured using a 5nm process and is designed to provide an Instructions per Cycle (IPC) uplift and frequency improvements over prior generation "Zen" cores. Each core has a larger L2 cache and improved cache effectiveness over the prior generation. Each "Zen 4" core includes:

- Up to 32 KB of 8-way L1 I-cache and 32 KB of 8-way of L1 D-cache
- Up to a 1MB private unified (Instruction/Data) L2 cache. All caches use a 64B cache line size.

Each core supports Simultaneous Multithreading (SMT), which allows 2 separate hardware threads to run independently, sharing the corresponding core's L2 cache.

2.6 Core Complex (CCX)

Figure 2-2 shows a Core Complex (CCX) where up to eight "Zen 4"-based cores share a L3 or Last Level Cache (LLC). Enabling Simultaneous Multithreading (SMT) allows a single CCX to support up to 16 concurrent hardware threads.

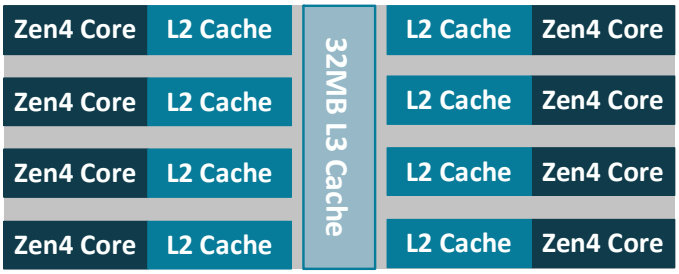

Figure 2-2: Top view of 8 compute cores sharing an L3 cache (91xx-96xx models)

2.7 Core Complex Dies (CCDs)

The Core Complex Die (CCD) in an AMD EPYC 9xx4 Series Processor may contain either one or two CCXs, depending on the processor (91xx-96xx "Genoa" vs. 97xx "Bergamo"), as shown in Figure 2-5.

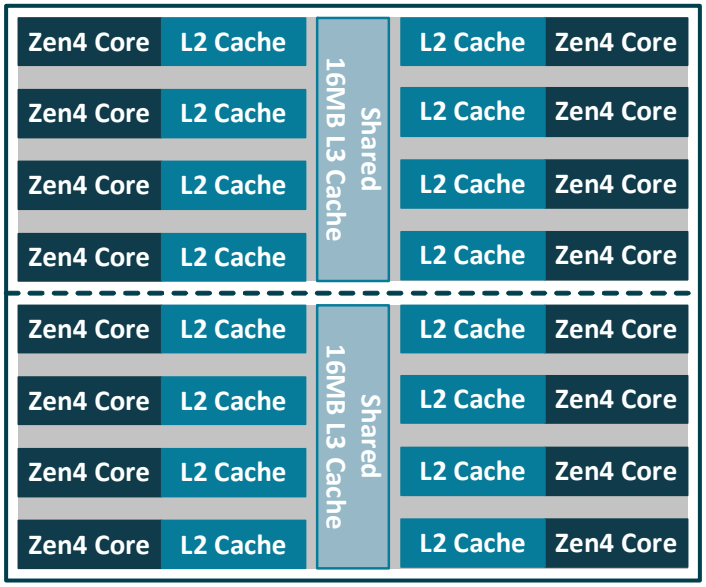

Figure 2-3: 2 CCXs in a single 4th Gen AMD EPYC 97xx CCD

Each of the Core Complex Dies (CCDs) in a 97xx model AMD EPYC 9004 Series Processor contains two CCXs (Figure 2-5):

AMD EPYC 9004 Series Processor	91xx-96xxq	97xx
# of CCXs within a CCD	1	2

Table 2-3: CCXs per CCD by AMD EPYC model

You can disable cores in BIOS using one or both of the following approaches:

- Reduce the cores per L3 from 8 down to 7,6,5,4,3,2, or 1 while keeping the number of CCDs constant. This approach increases the effective cache per core ratio but reduces the number of cores sharing the cache.
- Reduce the number of active CCDs while keeping the cores per CCD constant. This approach maintains the advantages of cache sharing between the cores while maintaining the same cache per core ratio.

2.8 AMD 3D V-Cache™ Technology

AMD EPYC 9xx4X Series Processors include AMD 3D V-Cache™ die stacking technology that enables 97xxr, more efficient chiplet integration. AMD 3D Chiplet architecture stacks L3 cache tiles vertically to provide up to 96MB of L3 cache per die (and up to 1 GB L3 Cache per socket) while still providing socket compatibility with all AMD EPYC™ 9004 Series Processor models.

AMD EPYC 9004 Series Processors with AMD 3D V-Cache technology employ industry-leading logic stacking based on copper-to-copper hybrid bonding "bumpless" chip-on-wafer process to enable over 200X the interconnect densities of current 2D technologies (and over 15X the interconnect densities of other 3D technologies using solder bumps), which translates to lower latency, higher bandwidth, and greater power and thermal efficiencies.

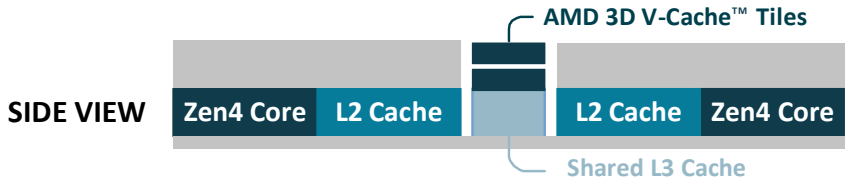

Figure 2-4: Side view of vertically-stacked central L3 SRAM tiles

AMD EPYC 9004 Series Processors	9xx4	9004X OPNs (with 3D V-Cache)
Max Shared L3 Cache (per CCD)	32 MB	96 MB

Table 2-4: L3 cache by processor model

Different OPNs also may have different numbers of cores within the CCX. However, for any given part, all CCxs will always contain the same number of cores.

2.9 I/O Die (Infinity Fabric™)

The CCDs connect to memory, I/O, and each other through an updated I/O Die (IOD). This central AMD Infinity Fabric™ provides the data path and control support to interconnect CCXs, memory, and I/O. Each CCD connects to the IOD via a dedicated high-speed Global Memory Interconnect (GMI) link. The IOD helps maintain cache coherency and additionally provides the interface to extend the data fabric to a potential second processor via its xGMI, or G-links. AMD EPYC 9004 Series Processors support up to 4 xGMI (or G-links) with speeds up to 32Gbps. The IOD exposes DDR5 memory channels, PCIe® Gen5, CXL 1.1+, and Infinity Fabric links.

All dies (chiplets) interconnect with each other via AMD Infinity Fabric technology. Figure 2-6 (which corresponds to Figure 2-2, above) shows the layout of a 96-core AMD EPYC 9654 processor. The AMD EPYC 9654 has 12 CCDs, with each CCD connecting to the IOD via its own GMI connection.

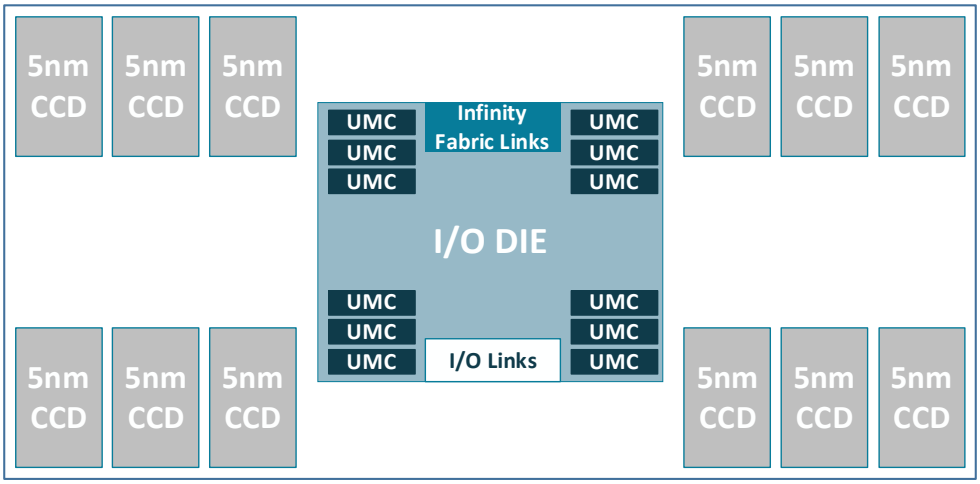

Figure 2-5: AMD EPYC 9654 processor internals interconnect via AMD Infinity Fabric (12 CCD processor shown)

AMD also provides "wide" OPNs (e.g. AMD EPYC 9334) where each CCD connects to two GMI3 interfaces, thereby allowing double the Core-to-I/O die bandwidth.

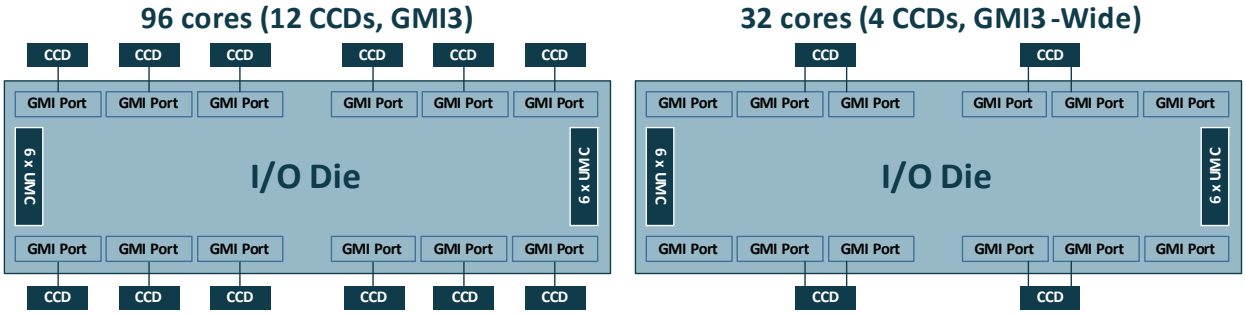

Figure 2-6: Standard vs. Wide GMI links

The IOD provides twelve unified memory controllers that support DDR5 memory. The IOD also presents 4 'P-links' that the system OEM/designer can configure to support various I/O interfaces, such as PCIe Gen5, and/or CXL 1.1+.

2.10 Memory and I/O

Each UMC can support up to 2 DIMMs per channel (DPC) for a maximum of 24 DIMMs per socket. OEM server configurations may allow either 1 DIMM per channel or 2 DIMMs per channel. 4th Gen AMD EPYC processors can support up to 6TB of DDR5 memory. Having additional and faster memory channels compared to previous generations of AMD EPYC processors provides additional memory bandwidth to feed high-core-count processors. Memory interleaving on 2, 4, 6, 8, 10, and 12 channels helps optimize for a variety of workloads and memory configurations.

Each processor may have a set of 4 P-links and 4 G-links. An OEM motherboard design can use a G-link to either connect to a second 4th Gen AMD EPYC processor or to provide additional PCIe Gen5 lanes. 4th Gen AMD EPYC processors support up to eight sets of x16-bit I/O lanes, that is, 128 lanes of high-speed PCIe Gen5 in single-socket platforms and up to 160 lanes in dual-socket platforms. Further, OEMs may either configure 32 of these 128 lanes as SATA lanes and/or configure 64 lanes as CXL 1.1+. In summary, these links can support:

- Up to 4 G-links of AMD Infinity Fabric connectivity for 2P designs.
- Up to 8 x16 bit or 128 lanes of PCIe Gen 5 connectivity to peripherals in 1P designs (and up to 160 lanes in 2-socket designs).
- Up to 64 lanes (4 P-links) that can be dedicated to Compute Express Link (CXL) 1.1+ connectivity to extended memory.
- Up to 32 I/O lanes that can be configured as SATA disk controllers.

2.11 Visualizing AMD EPYC 9004 Series Processors (Family 19h)

This section depicts AMD EPYC 9004 Series Processors that have been set up with four nodes per socket (NPS=4). Please see "NUMA Topology" on page 10 for more information about nodes.

2.11.1 Models 91xx-96xx ("Genoa")

4th Gen AMD EPYC 9004 processors with model numbers 91xx-96xx have up to 12 CCDs that each contain a single CCX, as shown below.

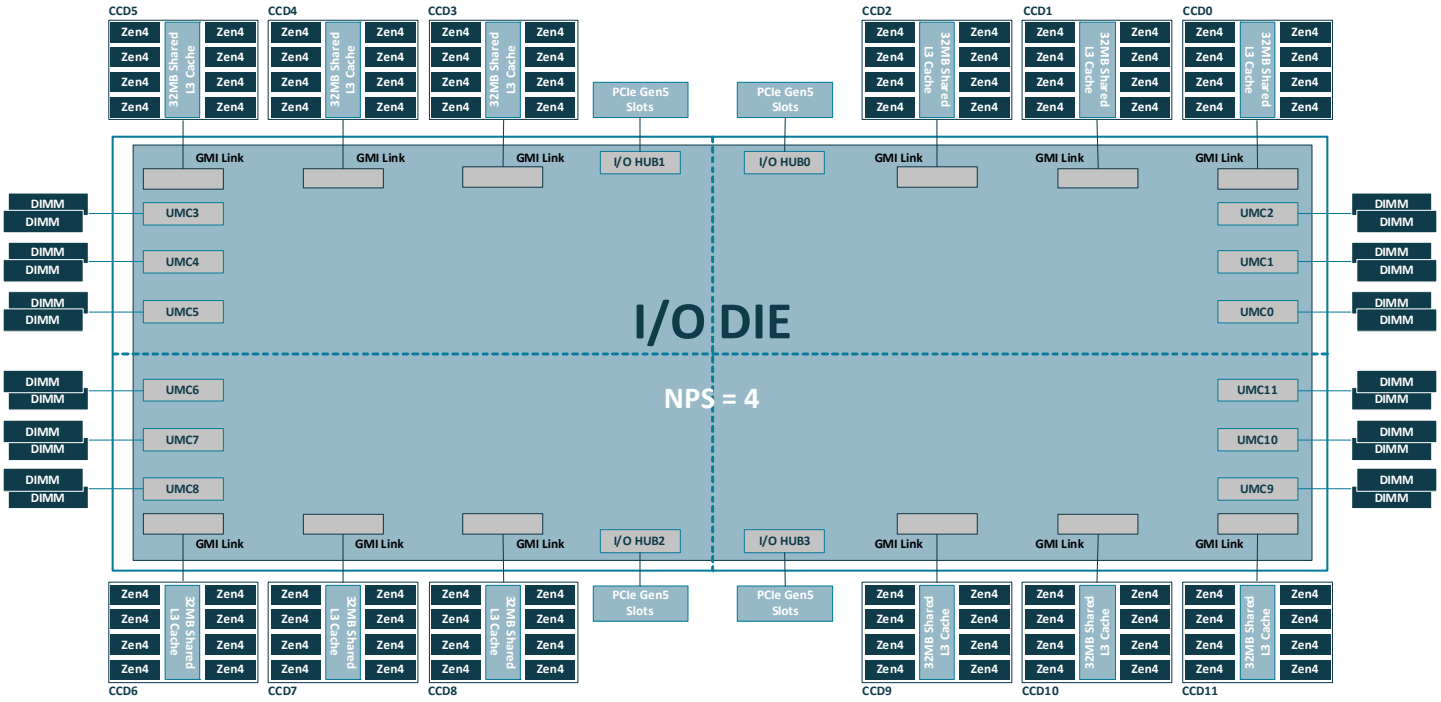

Figure 2-7: The AMD EPYC 9004 SoC consists of up to 12 CCDs and a central IOD for 91xx-96xx models, including "X" OPNs

2.11.2 Models 97xx ("Bergamo")

97xx 4th Gen AMD EPYC 9004 Series Processors with model numbers 97xx have up to 8 CCDs that each contain two CCXs, as shown below.

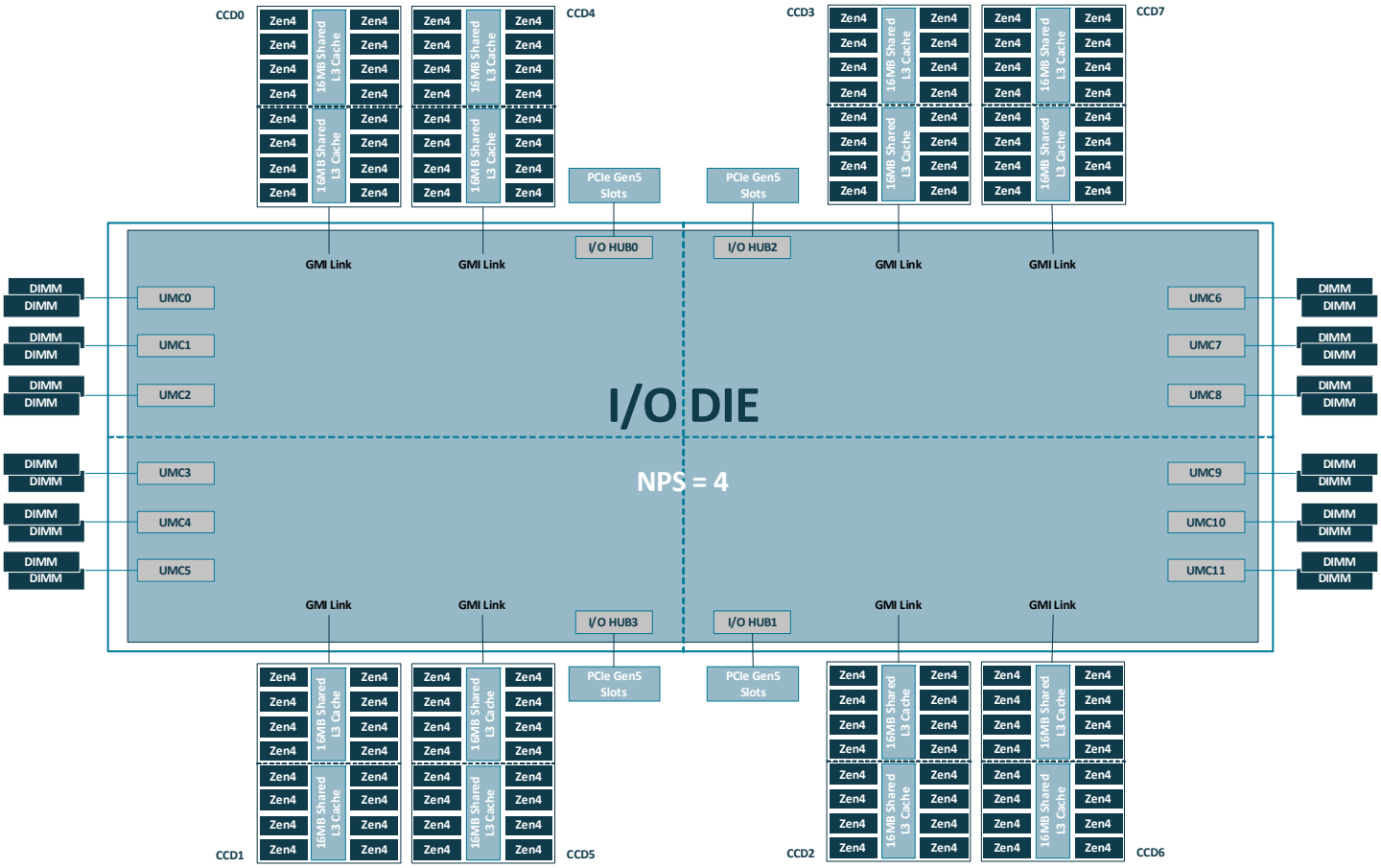

Figure 2-8: The AMD EPYC 9004 System on Chip (SoC) consists of up to 8 CCDs and a central IOD for 97xx models

2.12 NUMA Topology

AMD EPYC 9004 Series Processors use a Non-Uniform Memory Access (NUMA) architecture where different latencies may exist depending on the proximity of a processor core to memory and I/O controllers. Using resources within the same NUMA node provides uniform good performance, while using resources in differing nodes increases latencies.

2.12.1 NUMA Settings

A user can adjust the system **NUMA Nodes Per Socket** (NPS) BIOS setting to optimize this NUMA topology for their specific operating environment and workload. For example, setting NPS=4 as shown in "Memory and I/O" on page 8 divides the processor into quadrants, where each quadrant has 3 CCDs, 3 UMCs, and 1 I/O Hub. The closest processor-memory I/O distance is between the cores, memory, and I/O peripherals within the same quadrant. The furthest distance is between a core and memory controller or IO hub in cross- diagonal quadrants (or the other processor in a 2P configuration). The locality of cores, memory, and IO hub/devices in a NUMA-based system is an important factor when tuning for performance.

The NPS setting also controls the interleave pattern of the memory channels within the NUMA Node. Each memory channel within a given NUMA node is interleaved. The number of channels interleaved decreases as the NPS setting gets more granular. For example:

- A setting of NPS=4 partitions the processor into four NUMA nodes per socket with each logical quadrant configured as its own NUMA domain. Memory is interleaved across the memory channels associated with each quadrant. PCIe devices will be local to one of the four processor NUMA domains, depending on the IOD quadrant that has the corresponding PCIe root complex for that device.

- A setting of NPS=2 configures each processor into two NUMA domains that groups half of the cores and half of the memory channels into one NUMA domain, and the remaining cores and memory channels into a second NUMA domain. Memory is interleaved across the six memory channels in each NUMA domain. PCIe devices will be local to one of the two NUMA nodes depending on the half that has the PCIe root complex for that device.

- A setting of NPS=1 indicates a single NUMA node per socket. This setting configures all memory channels on the processor into a single NUMA node. All processor cores, all attached memory, and all PCIe devices connected to the SoC are in that one NUMA node. Memory is interleaved across all memory channels on the processor into a single address space.

- A setting of NPS=0 indicates a single NUMA domain of the entire system (across both sockets in a two-socket configuration). This setting configures all memory channels on the system into a single NUMA node. Memory is interleaved across all memory channels on the system into a single address space. All processor cores across all sockets, all attached memory, and all PCIe devices connected to either processor are in that single NUMA domain.

You may also be able to further improve the performance of certain environments by using the **LLC (L3 Cache) as NUMA** BIOS setting to associate workloads to compute cores that all share a single LLC. Enabling this setting equates each shared L3 or CCX to a separate NUMA node, as a unique L3 cache per CCD. A single AMD EPYC 9004 Series Processor with 12 CCDs can have up to 12 NUMA nodes when this setting is enabled.

Thus, a single EPYC 9004 Series Processor may support a variety of NUMA configurations ranging from one to twelve NUMA nodes per socket.

Note: If software needs to understand NUMA topology or core enumeration, it is imperative to use documented Operating System (OS) APIs, well-defined interfaces, and commands. Do not rely on past assumptions about settings such as APICID or CCX ordering.

2.13 Dual-Socket Configurations

AMD EPYC 9004 Series Processors support single- or dual-socket system configurations. Processors with a 'P' suffix in their name are optimized for single-socket configurations (see the "Processor Identification" chapter) only. Dual-socket configurations require both processors to be identical. You cannot use two different processor Ordering Part Numbers (OPNs) in a single dual-socket system.

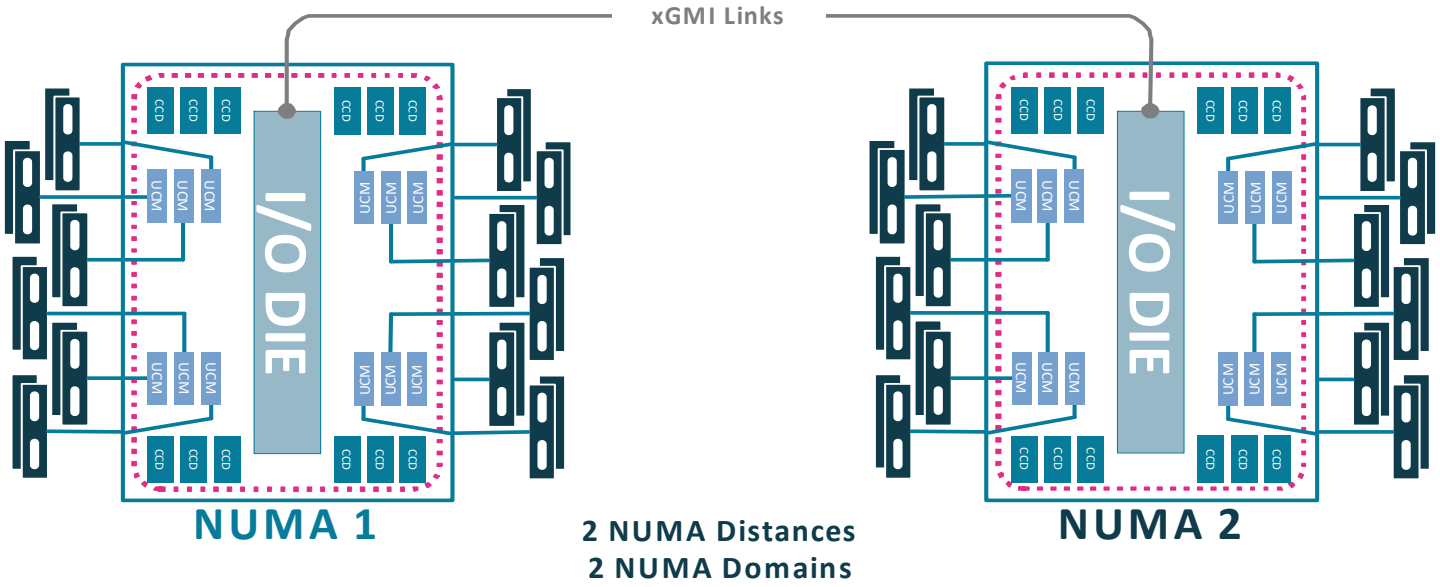

Figure 2-9: Two EPYC 9004 Processors connect through 4 xGMI links (NPS1)

In dual-socket systems, two identical EPYC 9004 series SoCs are connected via their corresponding External Global Memory Interconnect [xGMI] links. This creates a high bandwidth, low latency interconnect between the two processors. System manufacturers can elect to use either 3 or 4 of these Infinity Fabric links depending upon I/O and bandwidth system design objectives.

The Infinity Fabric links utilize the same physical connections as the PCIe lanes on the system. Each link uses up to 16 PCIe lanes. A typical dual socket system will reconfigure 64 PCIe lanes (4 links) from each socket for Infinity Fabric connections. This leaves each socket with 64 remaining PCIe lanes, meaning that the system has a total of 128 PCIe lanes. In some cases, a system designer may want to expose more PCIe lanes for the system by reducing the number of Infinity Fabric G-Links to from 4 to 3. In these cases, the designer may allocate up to 160 lanes for PCIe (80 per socket) by utilizing only 48 lanes per socket for Infinity Fabric links instead of 64.

A dual-socket system has a total of 24 memory channels, or 12 per socket. Different OPNs can be configured to support a variety of NUMA domains.

Chapter

3

BIOS Defaults Summary

This chapter provides high-level lists of the default AMD EPYC 9004 BIOS settings and their default values. Please see Chapter 4 of the *BIOS & Workload Tuning Guide for AMD EPYC™ 9004 Series Processors* (available from AMD EPYC Tuning Guides) for detailed descriptions. Later chapters in this Tuning Guide discuss the BIOS options as they relate to a specific workload or set of workload(s).

Note: The default setting name and values described in this chapter are the AMD default names and values that serve as recommendations for OEMs. End users must confirm their OEM BIOS setting availability and options.

AMD strongly recommends that customers download and install the latest BIOS update for your AMD EPYC 9004 Series Processor-based server from your platform vendor. BIOS updates often help customers by providing new and updated features, bug fixes, enhancements, security features, and other improvements. These improvements can help your system software stability and dependency modules (such as hardware, firmware, drivers, and software) by giving you a more robust environment to run your applications.

3.1 Processor Core Settings

Name	Default	Description
SMT Control	Auto	• **Enabled/Auto:** Two hardware threads per core. • **Disabled:** Single hardware thread per core.
L1 Stream HW Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L1 Stride Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L1 Region Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L1 Burst Prefetch Mode	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L2 Stream HW Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
L2 Up/Down Prefetcher	Auto	• **Enabled/Auto:** Enables the prefetcher. • **Disabled:** Disables the prefetcher.
Core Performance Boost	Auto	• **Enabled/Auto:** Enables Core Performance Boost. • **Disabled:** Disables Core Performance Boost.
BoostFmaxEn	Auto	• **Auto:** Use the default Fmax • **Manual:** User can set the boost Fmax
BoostFmax	Auto	Specify the boost Fmax frequency limit to apply to all cores (MHz in decimal)
Global C-State Control	Auto	• **Enabled/Auto:** Controls IO based C-state generation and DF C-states, including core processor C-States • **Disabled:** AMD strongly recommends not disabling this option because this also disables core processor C-States.

Table 3-1: Processor core BIOS settings

X3D	Auto	Enables or disables AMD 3D V-Cache™ technology on Cache Optimized (9004X) processors. • **Auto:** Enabled on an AMD EPYC 9004 Series processor with AMD 3D V-Cache™ technology, enabling this option enables the AMD 3D V-Cache module in the CCD to increase the total size of the L3 cache memory to 96MB • **Disabled:** Disabling this option reduces the L3 cache in the CCD to 32MB. *Note: This option only applies to AMD EPYC 9004 Series Processors with AMD 3D V-Cache technology.* *Note: AMD engineers performed extensive internal testing and validation for various applications using the X3D BIOS option found in AMD EPYC 9xx4X processors with AMD 3D V-Cache technology. This testing and validation cannot cover all applications or use cases. Testing has shown AMD 3D V-Cache to be beneficial for most workloads, however AMD recommends that you test and evaluate the benefits of enabling or disabling the X3D BIOS option for your application workloads in your environment and proceeding based on those results.*

Table 3-1: Processor core BIOS settings (Continued)

3.2 Power Efficiency Settings

Name	Default	Description
Power Profile Selection	Auto	• **Auto/0:** High-performance mode • **1:** Efficiency mode • **2:** Maximum I/O performance mode
Determinism Control	Auto	• **Auto:** Use default performance determinism settings. • **Manual:** Specify custom performance determinism settings.
Determinism Enable	Auto	• **Auto:** Performance. • **1:** Power.
TDP Control	Auto	• **Auto:** Use platform- and OPN-default TDP. • **Manual:** Set custom configurable TDP.
TDP	OPN Max	This option appears once the user sets the **TDP Control** to **Manual**. • **Values 85-400:** Set configurable TDP, in watts.
PPT Control	Auto	Enables or disables the **PPT** control. • **Auto:** Automatically set PPL in watts. • **Manual:** Specify a custom PPL.
PPT	OPN Max	This option appears once the user sets the **PPT Control** to **Manual**. • **Values 85-400:** Set configurable PPT, in watts.
CPPC	Auto	• **Enabled/Auto:** Allows the OS to make performance/power optimization requests using ACPI CPPC. • **Disabled:** Prevents the OS from making performance/power optimization requests using ACPI CPPC.

Table 3-2: Power efficiency BIOS settings

3.3 NUMA and Memory Settings

Name	Default	Description
LLC as NUMA Domain (ACPI SRAT L3 Cache as NUMA Domain)	Disabled	• **Disabled (recommended):** Both NUMA nodes (`cpubind`) and memory interleaving (`membind`) are determined by the NPS setting. • **Enabled:** Overrides the NPS setting for # of NUMA nodes by mapping each LLC as a NUMA node. This does not impact the memory interleaving
Nodes Per Socket (NPS)	1	**Memory Interleaving:** The **NPS** setting always determines the memory interleaving regardless of whether **LLC as NUMA** is **Enabled** or **Disabled**. # of NUMA nodes (if **LLC as NUMA Domain** is **Disabled**): • **NPS1:** One NUMA node per socket (Most cloud providers use this as it provides consistent average memory latency to all the accesses within a socket). • **NPS2:** Two NUMA nodes per socket. • **NPS4:** Four NUMA nodes per socket • **NPS0 (not recommended):** Only applicable for dual-socket systems. A single NUMA node is created for the whole two-socket platform. AMD recommends either NPS1 or NPS4 depending on your use case. **Windows systems:** Make sure that the number of logical processors per NUMA node is <=64. You can do this by using NPS2 or NPS4 instead of the default NPS1.
Memory Target Speed	Auto	• **Auto:** Determine the maximum memory speed based on SPD information from populated DIMMs and platform memory speed support. Alternatively, you can select: • **Values 3200–5600 MT/s:** Run the DRAM memory target clock speed at the specified speed. The DRAM memory target is the DDR rate. Your OEM system default value may vary.
Memory Interleaving	Auto	• **Auto/Enable:** Enables memory interleaving. • **Disable:** Allows for disabling memory interleaving. The N**UMA Nodes per Socket** setting will be honored regardless of this setting. AMD strongly recommends not disabling this setting because most production deployments benefit from memory interleaving.

Table 3-3: NUMA and memory BIOS settings

3.4 Infinity Fabric Settings

Name	Default	Description
3-4 xGMI Link Max Speed	Auto	• 12 Gbps • 16 Gbps • 17 Gbps • 18 Gbps • 20 Gbps • 22 Gbps • 23 Gbps • 24 Gbps • 25 Gbps/Auto • 26 Gbps • 27 Gbps • 28 Gbps • 30 Gbps • 32 Gbps Your OEM system default value may vary.
xGMI Link Width Control	Auto	• **Auto:** Use the default xGMI link width controller settings. • **Manual:** Specify a custom xGMI link width controller setting.
xGMI Force Link Width Control	Auto	• **Unforce:** Do not force the xGMI to a fixed width. • **Force:** Use the xGMI link to the user-specified width.
xGMI Force Link Width	Auto	• **0:** Force xGMI link width to x4. • **1:** Force xGMI link width to x8. • **2:** Force xGMI link width to x16.
xGMI Max Link Width Control	Auto	• **Auto:** Use the default xGMI link width controller settings. • **Manual:** Specify a custom xGMI link with controller setting.
xGMI Max Link Width	Auto	• **0:** Set max xGMI link width to x8. • **1:** Set max xGMI link width to x16.
APBDIS	Auto	• **0/Auto:** Dynamically switch the Infinity Fabric P-state based on link usage. • **1:** Enabled fixed Infinity Fabric P-state control.
DfPstate Range Support	Auto	• **Auto:** If this feature is enabled, the range value setting should follow the rule that MaxDfPstate<=MinDfPstate. Otherwise, it will not work. • **Enable:** Add the values MaxDfPstate & MinDfPstate. • **Disable:** No MaxDfPstate & MinDfPstate option.

Table 3-4: Infinity Fabric BIOS settings

DF C-States	Auto	Controls DF C-states. • **Disabled:** Prevents the AMD Infinity Fabric from entering a low-power state. • **Enabled/Auto:** Allows the AMD Infinity Fabric to enter a low-power state.

Table 3-4: Infinity Fabric BIOS settings (Continued)

3.5 PCIe, I/O, Security, and Virtualization Settings

Name	Default	Description
Local APIC Mode	Auto(0x02)	• **xAPIC:** Use xAPIC, supports up to 255 cores. • **x2APIC:** Supports more than 255 cores. • **Auto:** The system will choose the mode that best fits the number of active cores in the system. • **Compatibility:** Threads below 255 run in xAPIC with xAPIC ACPI structures, and threads 255 and above run in x2 mode with x2 ACPI structures. • **XApicMode (0x01):** Forces legacy xAPIC mode. • **X2ApicMode (0x02):** Forces x2APIC mode independent of thread count.
PCIe Speed PMM Control	Auto	• **0:** Dynamic link speed determined by power management functionality. • **1:** Static Target Link Speed (Gen4); sets the maximum idle link speed to 16 GT/s. • **Auto/2:** Static Target Link Speed (Gen5); sets the maximum idle link speed to 32 GT/s, thereby disabling the feature).
PCIe ARI Support (SRIOV)	Auto	• **Enabled/Auto:** Enables Alternative Routing ID interpretation. • **Disabled:** Disables Alternative Routing ID interpretation.
PCIe Ten Bit Tag Support	Auto	• **Enabled/Auto:** Enables PCIe 10-bit tags for supported devices. • **Disabled:** Disables PCIe 10-bit tags for all devices.
IOMMU	Auto	• **Enabled/Auto:** Enables IOMMU. AMD recommends setting this to `pt:pass-through` in the Linux kernel settings. • **Disabled:** Disables IOMMU.
AVIC	Disabled	Advanced Virtual Interrupt Controller. • **Disabled:** Disables AVIC. • **Enabled:** Enables AVIC.
x2AVIC	Disabled	x2AVIC is an extension of the advanced virtual interrupt controller. This feature currently requires a custom AMD Linux kernel. • **Disabled:** Disables x2AVIC. • **Enabled:** Enables x2AVIC.

Table 3-5: PCIe, I/O, security, and virtualization BIOS settings

TSME	Auto	• **Auto/Disabled:** Disables transparent secure memory encryption. • **Enabled:** Enables transparent secure memory encryption.
SEV	Disabled	In a multi-tenant environment (such as a cloud), Secure Encrypted Virtualization (SEV) mode isolates virtual machines from each other and from the hypervisor. • **Disabled:** SEV is disabled. • **Enabled:** SEV is enabled.
SEV-ES	Disabled	Secure Encrypted Virtualization-Encrypted State (SEV-ES) mode extends SEV protection to the contents of the CPU registers by encrypting them when a virtual machine stops running. Combining SEV and SEV-ES can reduce the attack surface of a VM by helping protect the confidentiality of data in memory. • **Disabled:** SEV-ES is disabled. • **Enabled:** SEV-ES is enabled.
SEV-SNP	Disabled	Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP) mode builds on SEV and SEV-ES by adding strong memory integrity protection to create an isolated execution environment that helps prevent malicious hypervisor-based attacks such as data replay and memory re-mapping. SEV-SNP also introduces several additional optional security enhancements that support additional VM use models, offer stronger protection around interrupt behavior, and increase protection against recently-disclosed side channel attacks. • **Disabled:** SEV-SNP is disabled. • **Enabled:** SEV-SNP is enabled.

Table 3-5: PCIe, I/O, security, and virtualization BIOS settings (Continued)

3.6 Higher-Level Settings

The system powers on to an initial state, after which succeeding software layers may affect system settings:

1. System firmware validates basic hardware functionality and allows users to change various settings via the BIOS Setup menus.
2. UEFI provides a shell environment that allows users to further interact with the system.
3. The operating system or hypervisor is the next software layer that provides control over system hardware.
4. Lastly, certain applications can also affect underlying hardware.

Each of the lines above may alter settings made by prior line, and some user changes require a reboot to take effect.

Please refer to your OEM documentation and/or applicable AMD Tuning Guide(s) for further guidance.

Chapter 4 Hardware Configuration

4.1 CPU

Some of the key factors which to keep in mind while training/inferencing AI/ML benchmarks include:

- Number of processors.
- Number of cores per processor
- Threads per core
- NUMA nodes per socket

Observation shows that some end-to-end AI/ML workloads that encompass the complete process of generating the data, preprocessing, training the model, and serving the model may not have greater scalability unless the workload is being implemented using some distributed framework (for example TPCx-AI single-node). These types of workloads require considering the number of processors and cores per processors. They may also show comparable performance with lower core density systems.

However, concurrent executions while inferencing with workloads such as BERT-Large, ResNet50, and DLRM can show improved performance. This requires special consideration during performance tuning to obtain optimal performance from a machine with Non-Uniform Memory Access (NUMA) enabled.

4.2 Memory

Use all memory channels evenly when populating memory by placing the same number and size of memory DIMMs in each memory channel to keep the NUMA nodes balanced.

All AI/ML workloads, be they training or inferencing, are very sensitive to memory capacity, latency, and bandwidth. Properly placing memory with memory DIMMs evenly installed across the sockets and setting the memory operating speed to a recommended high value (1DPC) can help improve performance. AMD recommends populating all 12 memory channels per CPU socket with DIMMs of equal capacity to allow the memory subsystem to get the best possible performance.

Please see the latest version of Memory Population Guidelines for AMD Family 19h Models 10h–1Fh (login required) for additional guidance.

4.3 Network and I/O

AI/ML workloads for inferencing (such as BERT-Large, ResNet50, or DLRM) are compute and memory intensive workloads. Networking or I/O therefore do not play any significant role. However, end-to-end AL/ML workloads, such as TPCx-AI, have networking and I/O requirements that are covered in "Network Configuration" on page 25. Please also see the following Tuning Guides, which are available from AMD EPYC™ Server Performance Tuning Guides:

- *Microsoft® Windows® Network Tuning Guide for AMD EPYC 9004 Series Processors*
- *Linux® Network Tuning Guide for AMD EPYC 9004 Series Processors*

Chapter 5 BIOS Tuning Recommendations

The BIOS settings described in this chapter relate to a specific AI/ML workload or set of workloads. "BIOS Defaults Summary" on page 13 contains a summary of AMD BIOS settings and their defaults. You can also see the *BIOS & Workload Tuning Guide for AMD EPYC™ 9004 Series Processors*, which is available from AMD EPYC™ Server Performance Tuning Guides, for detailed explanations of all AMD BIOS settings.

Name	Value	Description
SMT Control	Enabled	Enables Symmetric Multi-Threading (SMT), which creates two computing threads per core.
NUMA Node per socket (NPS)	NPS4	The default value is NPS1, but there are times when setting this to NPS4 boosts performance when doing concurrent executions during inferencing with multiple instances that are each running in different CCDs, thus maximizing throughput. Generally speaking, all deep learning workloads, training, or inference get better performance without accessing hardware resources across NUMA nodes.
ACPI SRAT L3 Cache as NUMA Domain	Disabled	This setting is disabled by default. In some cases, enabling this feature will improve L3 cache hits from the processor, thereby improving memory latency and overall performance.

Table 5-1: Recommended BIOS settings for AI/ML workloads

This page intentionally left blank.

Chapter 6 Linux Optimizations

6.1 Network Configuration

Some end-to-end AI/ML workloads may require setting up a Hadoop cluster. These workloads require high-bandwidth 25 GbE NICs. Please visit AMD EPYC™ Server Performance Tuning Guides to review the following Tuning Guides:

- *Microsoft® Windows® Network Tuning Guide for AMD EPYC 9004 Series Processors*
- *Linux® Network Tuning Guide for AMD EPYC 9004 Series Processors*

Per these guides, you may need to:

- Tune the sizes of the TX and RX rings.
- Change the number of interrupt queues to match the cores on the NUMA node on which the NIC is collocated, and then pin those interrupts to the correct CPU cores.
- Use the `iperf` utility to stress test the network infrastructure and verify proper configuration.

6.2 Disk Configuration

The general guidelines in the following table regarding the storage device to CPU core ratios will help you determine the minimum number of devices needed to establish ideal I/O throughput.

Drive Type	Drive to Core Ratio
10K RPM SATA Hard Drives	1 drive for every core
SATA Solid State Disks	1 drive for every 2 to 4 cores
PCIe Generation 3 NVMe Drive	1 drive for every 4 to 8 cores
PCIe Generation 4 NVMe Drive	1 drive for every 8 to 16 cores
PCIe Generation 5 NVMe Drive	1 drive for every 16 to 20 cores

Table 6-1: Drive to core recommendations

End-to-end AI/ML workloads requiring data generation and processing may encounter the `java.io.FileNotFoundException` (too many open files) error, and jobs will fail if the default number of open files is too small. You can avoid this by executing the `ulimit` command as the root user to increase the number of open files to 32768 or even 65536, such as `ulimit -n 65536`. Add the following lines to `/etc/security/limits.conf`:

```
-nofile 65536
-nrpoc 65536
```

Please refer to the disk configuration section of the Hadoop® Tuning Guide for AMD EPYC™ 9004 Series Processors (available from AMD EPYC™ Server Performance Tuning Guides) for detail disk performance optimizations suitable for end-to-end AI/ML workloads involving distributed training and inferencing.

6.3 tuned-adm profile

Check the performance governor on all processor cores. The tuned-adm throughput-performance profile sets the governor to performance, which generally works best for AI/ML workloads.

6.4 Transparent Huge Pages

Transparent Huge Pages (THPs) are a Linux kernel feature for memory management to improve application performance by efficiently using the processor's memory-mapping hardware. THP should reduce the overhead of the Translation Lookaside Buffer. User must login as a sudo user to enable or disable THP settings. It operates mainly in two modes:

- **always:** The system kernel tries to assign huge pages to the processes running on the system.
  `echo always > /sys/kernel/mm/transparent_hugepage/enabled`
- **madvise:** The system kernel only assigns huge pages to the memory areas of individual processes.
  `echo madvise > /sys/kernel/mm/transparent_hugepage/enabled`

AMD recommends the following THP setting for better performance:

- **CNN models:** `always`
- **NLP models:** `madvise`

Chapter

7

Hadoop Settings

End-to-end AI/ML workloads involving distributed training may require Hadoop Distributed File System (HDFS), Yet Another Resource Negotiator (YARN), and SPARK as the main components that form the data management layer for Hadoop, where:

- HDFS is the storage management framework.
- YARN is the resource management framework.
- SPARK is the processing framework.

Please refer Hadoop setting section from Hadoop® Tuning Guide for AMD EPYC™ 9004 Series Processors for end-to-end AI/ML workloads requiring distributed trainings and inferencing.

This page intentionally left blank.

Chapter 8 ZenDNN 4.0 Libraries

8.1 TensorFlow

ZenDNN 4.0 supports:

- TensorFlow v2.10, which is expected to deliver similar or better performance compared to TensorFlow v2.9.
- AMD Unified Inference Frontend (UIF) optimized models. Please see the AMD UIF documentation* for details.
- TensorFlow v2.10 wheel file, which is compiled with GCC v9.3.1

You can use the TensorFlow-ZenDNN User Guide to set up and tune your AI/ML workloads, such as BERT-large and ResNet50, for optimal performance on AMD EPYC 9004 Series Processors with AVX-512 support.

8.2 PyTorch

ZenDNN 4.0 supports:

- PyTorch v1.12. This is a baseline release for PyTorch v1.12 and is expected to deliver similar or better performance as compared to PyTorch v1.11.
- AMD Unified Inference Frontend (UIF) optimized models. Please see the AMD UIF documentation* for details.
- PyTorch v1.12 wheel file, which is compiled with GCC v9.3.1.

You can use the PyTorch-ZenDNN User Guide to set up and tune AI/ML workloads, such as DLRM, for optimal performance on AMD EPYC 9004 Series Processors with AVX-512 support.

8.3 OpenMP Options/Thread Affinity

You can use OpenMP to improve the performance of parallel computation tasks. `OMP_NUM_THREADS` is the easiest switch you can use to accelerate computations. It determines the number of threads used for OpenMP computations.

You can use the `OMP_NUM_THREADS`, `OMP_WAIT_POLICY`, `OMP_PROC_BIND`, and `GOMP_CPU_AFFINITY` enviornment variables to tune performance. For optimal performance, the Batch Size must be a multiple of the total number of cores (used by the threads). On a 4th Gen AMD EPYC server (configuration: AMD EPYC 9654P 96-Core, 2P, and SMT=ON) with the above environment variable values, `OMP_NUM_THREADS=96` and `GOMP_CPU_AFFINITY=0-95` yield the best throughput numbers.

The `OMP_WAIT_POLICY` environment variable provides options to the OpenMP runtime library based on the expected behavior of the waiting threads. This variable can take the abstract values `PASSIVE` and `ACTIVE` (default). Setting `OMP_WAIT_POLICY` is set to `PASSIVE` means that the waiting threads will be passive and will not consume processor cycles. Conversely, setting it to `ACTIVE` will consume processor cycles. For the ZenDNN stack, setting `OMP_WAIT_POLICY` to `ACTIVE` may give better performance.

You can improve ZenDNN performance by precisely guarding the behavior of the OpenMP thread via thread affinity settings. You cannot modify or change a thread affinity defined at startup during application runtime. You can use the GOMP_CPU_AFFINITY environment variable to bind threads to the physical CPUs. For example:

```
export GOMP_CPU_AFFINITY=0-95
```

This setting will yield optimal performance on a 4th Gen AMD EPYC server (configuration: AMD EPYC 9654P 96-Core, 2P, and SMT=ON).

Chapter 9 AI/ML-Specific Configuration Settings

9.1 BERT-Large

The Bidirectional Encoder Representations for Transformers (BERT) is a deep learning model used for various natural language processing tasks. This model has been pre-trained on Wikipedia and BooksCorpus and requires additional tuning for specific tasks.

The following tunings are with ZenDNN4.0 for inferencing task using the BERT-L wwm_uncased_L-24_H-1024_A-16 (FP32) model on a 4th Gen AMD EPYC system (configuration: AMD EPYC 9654P 96-Core, 2P, and SMT=OFF, NPS4).

Optimal settings for ZenDNN4.0:

```
export ZENDNN_GEMM_ALGO=4
export ZENDNN_ENABLE_MEMPOOL=1
export ZENDNN_CONV_ALGO=1
export ZENDNN_TF_VERSION=2.10
export OMP_NUM_THREADS=96
export GOMP_CPU_AFFINITY="0-95"
export ZENDNN_TENSOR_POOL_LIMIT=1024
export ZENDNN_TF_CONV_ADD_FUSION_SAFE=1

sudo echo madvise  > /sys/kernel/mm/transparent_hugepage/enabled
```

Optimal settings for ZenDNN4.1:

```
export ZENDNN_GEMM_ALGO=4
export ZENDNN_ENABLE_MEMPOOL=2
export ZENDNN_CONV_ALGO=4
export ZENDNN_TF_VERSION=2.12
export OMP_NUM_THREADS=96
export GOMP_CPU_AFFINITY="0-95"
export ZENDNN_TENSOR_POOL_LIMIT=1024
export ZENDNN_TF_CONV_ADD_FUSION_SAFE=1

sudo echo madvise  > /sys/kernel/mm/transparent_hugepage/enabled
```

You can use `numactl` to trigger the workload to avoid cross-socket memory access to reduce memory access overhead:

```
numactl --cpunodebind=0-3 -membind=0-3 python <bert-script>
```

Note: The `numactl` *interleave option only works when the number of nodes allocated for a particular application is more than one.* `cpunodebind` *and* `physcpubind` *behave the same way for the ZenDNN stack, whereas interleave memory allocation performs better than* `membind`*.*

You can obtain maximum throughput by running multiple instances of the script, where each instance runs on one socket. For example, the following commands launch two instances of the BERT_L inferencing script and place each instance on separate 4th Gen AMD EPYC system (configuration: AMD EPYC 9654P 96-Core, 2P, and SMT=OFF, NPS4) sockets.

```
GOMP_CPU_AFFINITY=0-95 numactl --cpunodebind=0-3 -membind=0-3 python <bert-script>
GOMP_CPU_AFFINITY=96-191 numactl --cpunodebind=4-7 -membind=4-7 python <bert-script>
```

9.2 ResNet50

Residual Networks (ResNet) is a Convolutional Neural Network (CNN) used for computer vision. The ResNet-50 CNN is 50 layers deep and is commonly used for image classification and training using an image dataset such as ImageNet before the trained model can be used for inference. The following tunings are for ZenDNN4.0 running an inferencing task using the `resnet50_fp32_pretrained_model.pb` (FP32) on a 4th Gen AMD EPYC system (configuration: AMD EPYC 9654P 96-Core, 2P, and SMT=OFF, NPS4):

Optimal settings for ZenDNN4.0/ZenDNN4.1:

```
export ZENDNN_GEMM_ALGO=3
export ZENDNN_ENABLE_MEMPOOL=1
export ZENDNN_CONV_ALGO=4
export ZENDNN_TF_VERSION=2.10
export OMP_NUM_THREADS=96
export GOMP_CPU_AFFINITY="0-95"
export ZENDNN_TF_CONV_ADD_FUSION_SAFE=1
export ZENDNN_TENSOR_POOL_LIMIT=512
```

For latency:

```
sudo echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

For throughout:

```
sudo echo always > /sys/kernel/mm/transparent_hugepage/enabled
```

You can avoid cross-socket memory access to reduce memory access overhead by triggering the workload using `numactl`:

```
numactl --cpunodebind=0-3 -membind=0-3 python <resnet50-script>
```

You can also run multiple instances of the script for maximum throughput where each instance runs on one socket. For example, the following commands launch two instances of a ResNet50 inferencing script where each instance is placed on a separate socket of a 4th Gen AMD EPYC system (configuration: AMD EPYC 9654P 96-Core, 2P, and SMT=OFF, NPS4):

```
GOMP_CPU_AFFINITY=0-95 numactl --cpunodebind=0-3 -membind=0-3
python <resnet50-script>
GOMP_CPU_AFFINITY=96-191 numactl --cpunodebind=4-7 -membind=4-7
python <resnet50-script>
```

9.3 DLRM

The Deep Learning Recommendation Model (DLRM) is a recommendation model designed to make use of both categorical and numerical inputs. The DLRM model handles continuous (dense) and categorical (sparse) features that describe users and products. It exercises a wide range of hardware and system components, such as memory capacity and bandwidth, as well as communication and compute resources.

The following tunings use ZenDNN4.0 for an inferencing task using the `tb00_40M.pt` (90GB FP32) model with MLPerf scripts in offline mode on a 4th Gen AMD EPYC system (configuration: AMD EPYC 9654P 96-Core, 2P, and SMT=OFF, NPS1).

```
export ZENDNN_EMBAG_MAX_THREADS=2
export ZENDNN_GEMM_ALGO=3
export OMP_NUM_THREADS=2
export BLIS_NUM_THREADS=2
```

For DLRM with TF stack, Optimal settings for ZenDNN4.0

```
export ZENDNN_GEMM_ALGO=4
export ZENDNN_ENABLE_MEMPOOL=1
export ZENDNN_CONV_ALGO=1
export ZENDNN_TF_VERSION=2.10
export OMP_NUM_THREADS=96
export GOMP_CPU_AFFINITY="0-95"
export ZENDNN_TENSOR_POOL_LIMIT=1024
export ZENDNN_TF_CONV_ADD_FUSION_SAFE=1

sudo echo madvise  > /sys/kernel/mm/transparent_hugepage/enabled
```

For DLRM with TF stack, Optimal settings for ZenDNN4.1

```
export ZENDNN_GEMM_ALGO=4
export ZENDNN_ENABLE_MEMPOOL=2
export ZENDNN_CONV_ALGO=4
export ZENDNN_TF_VERSION=2.12
export OMP_NUM_THREADS=96
export GOMP_CPU_AFFINITY="0-95"
export ZENDNN_TENSOR_POOL_LIMIT=1024
export ZENDNN_TF_CONV_ADD_FUSION_SAFE=1

sudo echo madvise  > /sys/kernel/mm/transparent_hugepage/enabled
```

This page intentionally left blank.

Chapter 10 Additional Information

- Memory Population Guidelines for AMD EPYC 9004 Series Processors (login required)
- Socket SP5 Platform NUMA Topology for AMD Family 19h Models 10h–1Fh (login required)
- Please see AMD EPYC™ Server Performance Tuning Guides for the following titles:
  - *Linux® Network Tuning Guide for AMD EPYC™ 9004 Series Processors*
  - *Windows® Network Tuning Guide for AMD EPYC™ 9004 Series Processors*
- Tuning YARN*
- TensorFlow-ZenDNN User Guide
- PyTorch-ZenDNN User Guide
- https://github.com/google-research/bert*
- https://github.com/facebookresearch/dlrm*
- https://mlcommons.org/en/*
- https://github.com/mlcommons/inference*

This page intentionally left blank.

Chapter 11

Processor Identification

Figure 11-1 shows the processor naming convention for AMD EPYC 9004 Series Processors and how to use this convention to identify particular processors models:

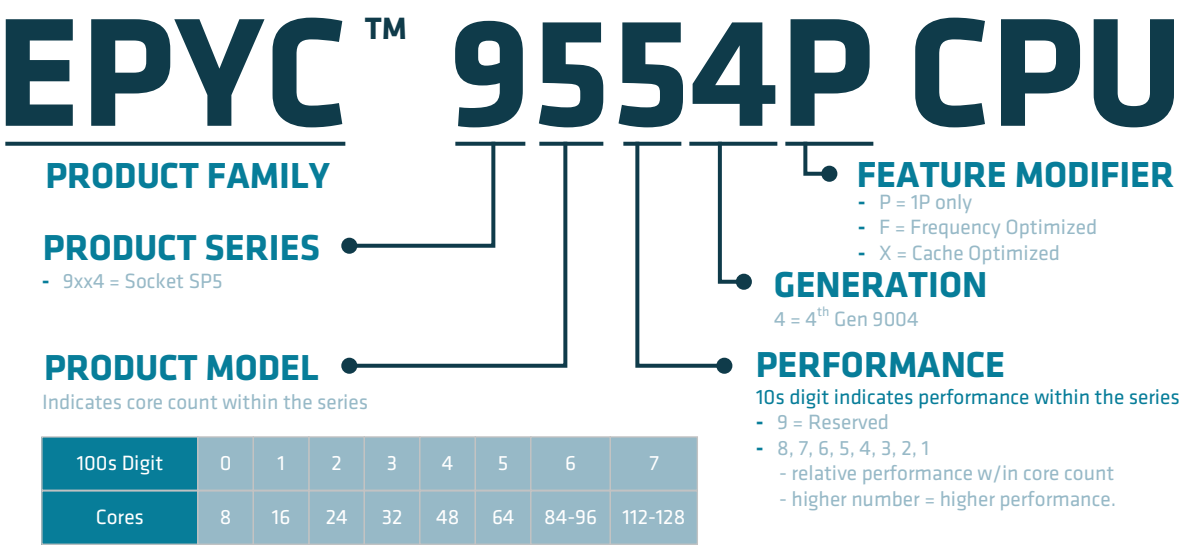

Figure 11-1: AMD EPYC SoC naming convention

11.1 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.
  - **91xx-96xx (including "X" OPNs):** Family 19h 10-1F
  - **97xx:** Family 19h A0-AF
- **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.

11.2 New Software-Visible Features

AMD EPYC 9004 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-byte 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*.

11.2.1 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 9004 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 9004 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.