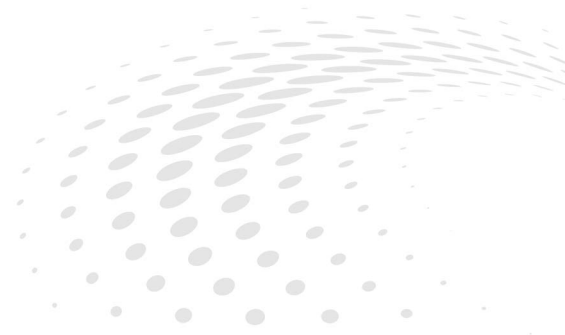


# TUNING GUIDE AMD EPYC 9004



## MongoDB®

Publication	58014
Revision	1.3
Issue Date	June, 2023

© 2023 Advanced Micro Devices, Inc. All rights reserved.

The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

**Trademarks**

AMD, the AMD Arrow logo, AMD EPYC, 3D V-Cache, and combinations thereof are trademarks of Advanced Micro Devices, Inc. MongoDB is a registered trademark of MongoDB, Inc. Other product names and links to external sites used in this publication are for identification purposes only and may be trademarks of their respective companies.

\* Links to third party sites are provided for convenience and unless explicitly stated, AMD is not responsible for the contents of such linked sites and no endorsement is implied.

Date	Version	Changes
July, 2022	0.1	Initial NDA partner release
Sep, 2022	0.2	Updated BIOS information
Nov, 2022	1.0	Initial public release
Dec, 2022	1.1	Minor errata corrections
Mar, 2023	1.2	Added 97xx OPN and AMD 3D V-Cache™ technology information
Jun, 2023	1.3	Second public release

**Audience**

This tuning guide describes best practices for optimizing performance when using MongoDB®. It is intended for a technical audience such as MongoDB 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.

**Authors**

Gnanakumar Rajaram and Sylvester Rajasekaran.

*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.*

# Table of Contents

<b>Chapter 1</b>	<b>Introduction</b>	<b>1</b>
1.1	MongoDB®	2
1.1.1	When to Use MongoDB	2
1.1.2	Recommended CPUs for MongoDB Servers	3
1.1.3	Recommended System Resources	4
<b>Chapter 2</b>	<b>AMD EPYC™ 9004 Series Processors</b>	<b>5</b>
2.1	General Specifications	5
2.2	Model-Specific Features	5
2.3	Operating Systems	6
2.4	Processor Layout	6
2.5	“Zen 4” Core	6
2.6	Core Complex (CCX)	7
2.7	Core Complex Dies (CCDs)	7
2.8	AMD 3D V-Cache™ Technology	8
2.9	I/O Die (Infinity Fabric™)	9
2.10	Memory and I/O	10
2.11	Visualizing AMD EPYC 9004 Series Processors (Family 19h)	11
2.11.1	Models 91xx-96xx (“Genoa”)	11
2.11.2	Models 97xx (“Bergamo”)	12
2.12	NUMA Topology	12
2.12.1	NUMA Settings	12
2.13	Dual-Socket Configurations	14
<b>Chapter 3</b>	<b>BIOS Defaults Summary</b>	<b>15</b>
3.1	Processor Core Settings	16
3.2	Power Efficiency Settings	18
3.3	NUMA and Memory Settings	19
3.4	Infinity Fabric Settings	20
3.5	PCIe, I/O, Security, and Virtualization Settings	21
3.6	Higher-Level Settings	22
<b>Chapter 4</b>	<b>Hardware Configuration Best Practices</b>	<b>23</b>
4.1	Memory Configuration	23
4.2	BIOS Settings	23
4.2.1	BIOS Settings for Maximizing Performance	24

<b>Chapter 5</b>	<b>Linux Optimizations</b> -----	<b>25</b>
5.1	Memory Subsystem .....	25
5.1.1	Swappiness .....	25
5.1.2	Disabling Transparent Huge Pages (THP) .....	25
5.1.3	NUMA Interleaving .....	26
5.2	Storage Subsystem .....	26
5.2.1	Filesystem Mount Options .....	27
5.2.2	I/O Scheduler .....	27
5.3	Network Subsystem .....	28
5.4	tuned-adm Profile .....	29
5.5	NTP Configuration using Chrony .....	29
5.6	Stopping Linux Firewall and SELinux (If Required) .....	30
5.7	Example RHEL Server Configuration Files .....	30
5.7.1	/etc/default/grub .....	30
5.7.2	/etc/rc.local .....	30
5.7.3	/etc/sysctl.conf .....	31
5.7.4	/etc/security/limits.conf .....	31
<b>Chapter 6</b>	<b>Performance Tuning MongoDB Settings</b> -----	<b>33</b>
6.1	MongoDB Architecture .....	33
6.2	MongoDB Tuning Best Practices .....	33
6.3	Locking Performance .....	35
6.4	Number of Connections .....	35
6.5	Fine-Grained Telemetry for MongoDB Performance Analysis .....	35
<b>Chapter 7</b>	<b>Resources</b> -----	<b>37</b>
<b>Chapter 8</b>	<b>Glossary</b> -----	<b>39</b>
<b>Chapter 9</b>	<b>Processor Identification</b> -----	<b>41</b>
9.1	CPUID Instruction .....	41
9.2	New Software-Visible Features .....	42
9.2.1	AVX-512 .....	42

**Chapter****1**

# Introduction

This tuning guide provides detailed descriptions of parameters that can optimize performance on servers built with AMD EPYC™ 9004 Series processors. Default OEM hardware and BIOS settings may not provide the best possible performance on all OS platforms and for all workloads. This guide helps you tune the following settings to optimize the performance of your specific workload:

- Hardware configuration
- BIOS
- Network settings
- OS kernel parameters
- MongoDB® performance tuning



## 1.1.2 Recommended CPUs for MongoDB Servers

MongoDB 1-2 includes basic AMD EPYC processor selection guidelines for general MongoDB use cases. Select the processor that best fits your needs.

System Type	Cores	CPUs <sup>1</sup>	Memory	Storage	Network	Min # of Shards	Min. Node # <sup>2</sup>
Small	16	1 x AMD EPYC™ 16-core	128 GB	1-4 SAS SSDs	10 Gbit	1 shard with 3 replica sets	3+2+1=6
Medium	24	1 x AMD EPYC™ 24-core	128 GB / 256 GB	1-4 SAS SSDs / NVMe SSDs	25 Gbit	1 shard with 3 replica sets	3+2+1=6
Large	32	1 x AMD EPYC™ 32-core	256 GB / 512 GB	1-4 NVMe SSDs	25 Gbit	1 shard with 3 replica sets	3+2+1=6
Cloud <sup>3</sup> : PaaS, IaaS	Please see <a href="https://www.mongodb.com/docs/atlas/cluster-config/multi-cloud-distribution/">https://www.mongodb.com/docs/atlas/cluster-config/multi-cloud-distribution/</a> for additional information.						
<p>1 - Please see <a href="#">AMD EPYC 9004 Series Processors</a> for more information.</p> <p>2 - Total number of nodes = Replica set + Config server + Mongos</p> <p>3 - Recommendations for Cloud Service Providers (CSP) / Private Cloud for Hypervisor configuration.</p>							

Table 1-1: Recommended AMD EPYC processors for MongoDB servers

AMD recommends using 8, 16, or 32 vCPU instances with 1:4 vCPU to memory ratio VMs with appropriate storage attached for cloud IaaS deployments.

### 1.1.3 Recommended System Resources

Resource requirements depend on the size and resource demands of your MongoDB deployment, but you can use the following general recommendations to get started:

Operating System	Recommended Specifications*
CPU*	<p>8-core processors are the minimum per node for production usage.</p> <p>MongoDB's WiredTiger storage engine architecture can efficiently use multiple CPU cores. Provision an adequate number of CPU cores in proportion to concurrent client connections. In MongoDB Atlas, the number of CPU cores and concurrent client connections is a function of your chosen cluster tier. CPU-heavy operations include (but are not limited to) Page Compression, Data Calculation, Aggregation Framework Operations, and Map Reduce.</p>
RAM*	<p>MongoDB performs best when the applications working set (indexes and most frequently accessed data) fits in memory. RAM size is the most important factor for instance sizing; other optimizations may not significantly improve database performance without sufficient RAM.</p> <p>MongoDB uses system memory for operations such as (but not limited to) Aggregation, Index Traversing, Write Operations, Query Engine, and Connections.</p>
Storage	<p>SATA, PCIe, and NVMe SSDs. Use SSDs for read-heavy applications if the working set no longer fits in memory.</p>
<p>*The smaller CPU and RAM configurations apply to virtualized environments or cloud deployments            See <a href="#">Performance Best Practices: Hardware and OS Configuration</a>* for additional information.</p>	

*Table 1-2: Recommended system resource requirements for MongoDB servers*

## 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)	1,152 MB (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.

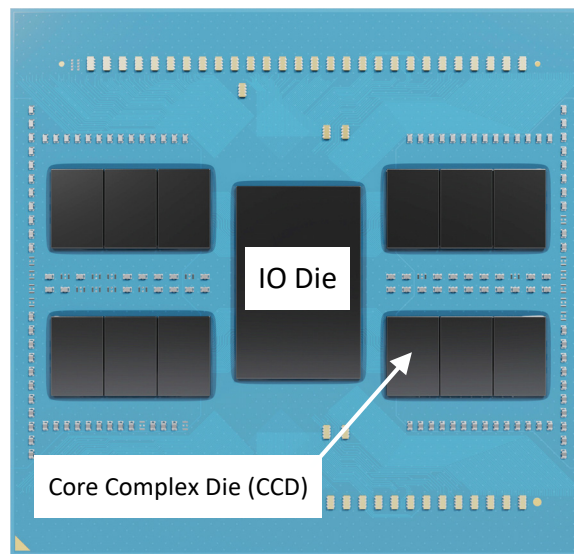


Figure 2-1: AMD EPYC 9004 configuration with 12 Core Complex Dies (CCD) surrounding a central I/O Die (IOD)

## 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 1 MB private unified (Instruction/Data) L2 cache.

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.

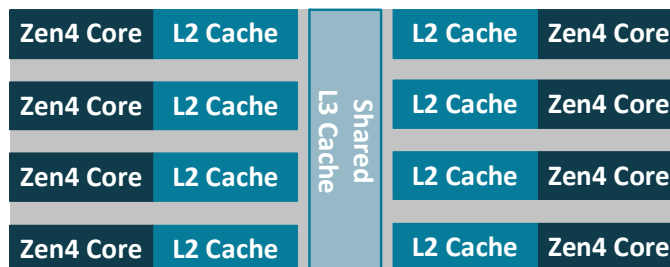


Figure 2-2: Top view of 8 compute cores sharing an L3 cache (91xx-96xx models)

## 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.

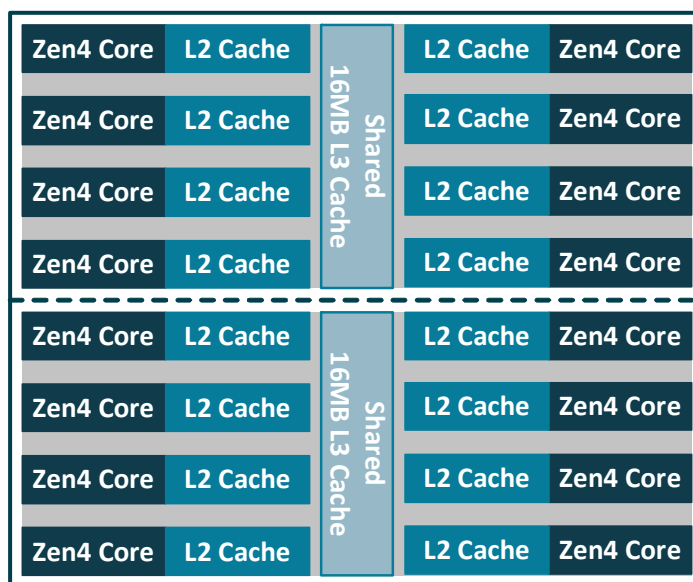


Figure 2-3: 2 CCXs in a single 4th Gen AMD EPYC 97xx CCD

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-96xx	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 97xx to achieve 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.

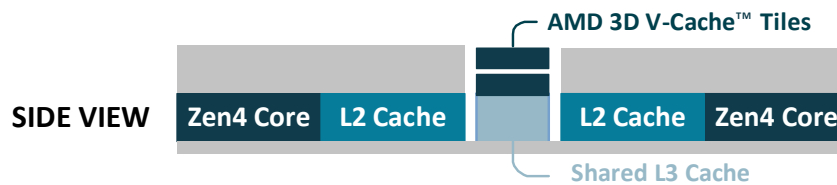


Figure 2-4: Side view of vertically-stacked central L3 SRAM tiles

AMD EPYC 9004 Series Processors	9xx4	9004X (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 (chipllets) 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.

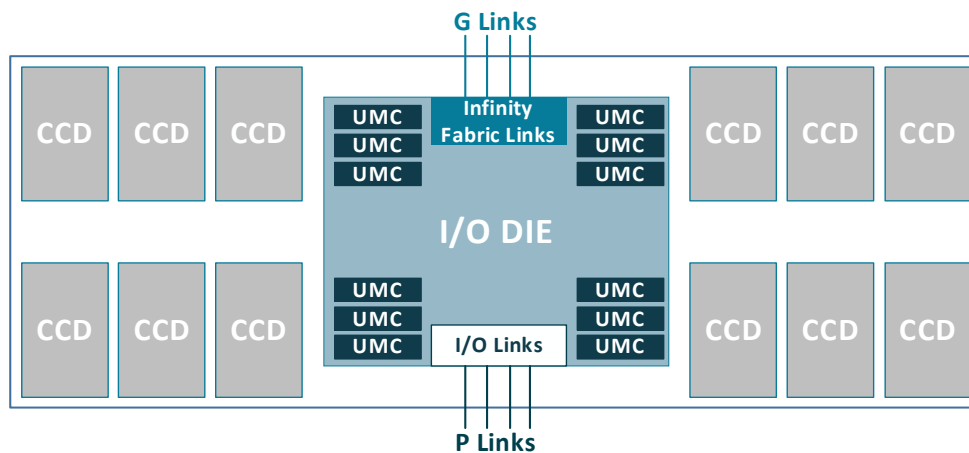


Figure 2-5: AMD EPYC 9654 processor internals interconnect via AMD Infinity Fabric (12 CCD processor shown)

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.

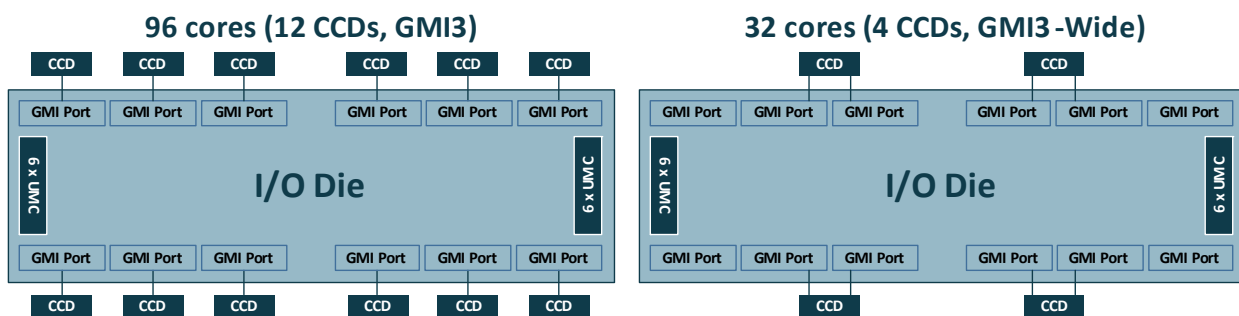


Figure 2-6: Standard vs. Wide GMI links

The IOD provides twelve Unified Memory Controllers (UMCs) 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 per socket. 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 12 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.

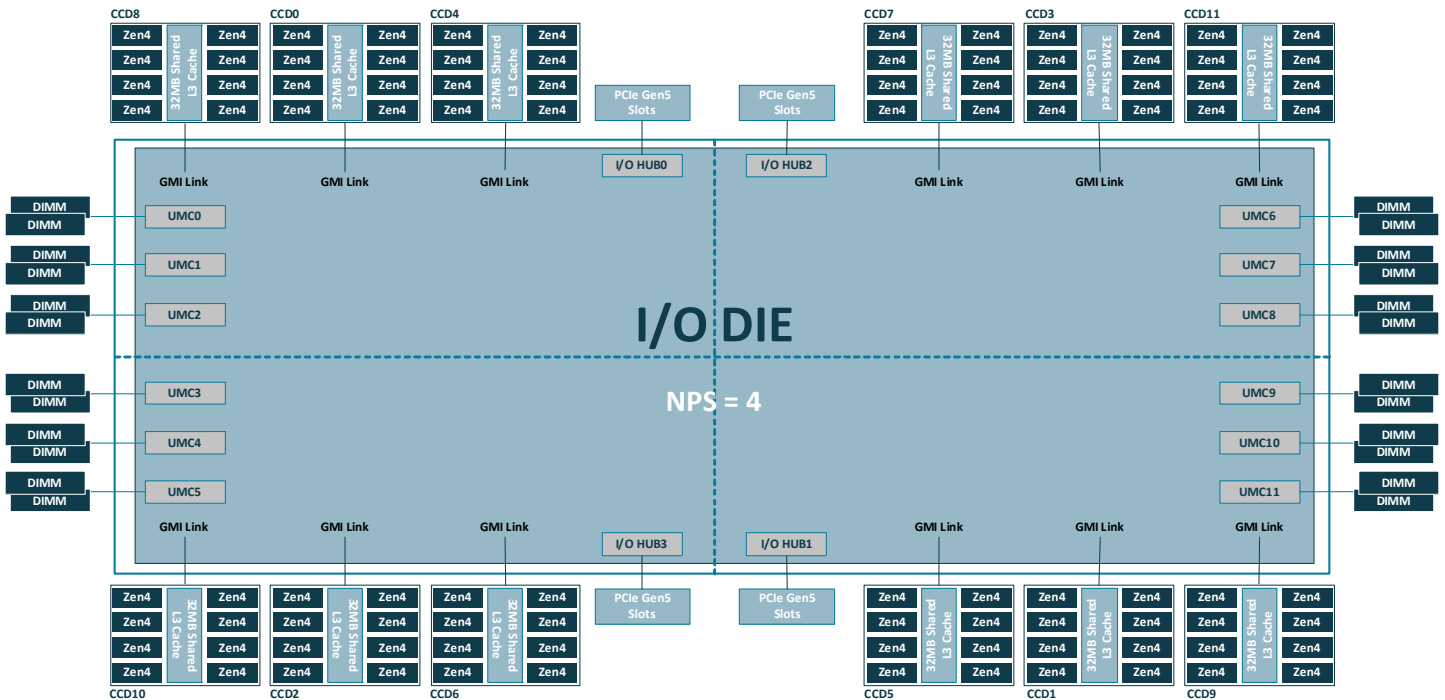


Figure 2-7: The AMD EPYC 9004 SoC consists of up to 12 CCDs and a central IOD for 91xx-96xx models, including “X” OPNs

## 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.

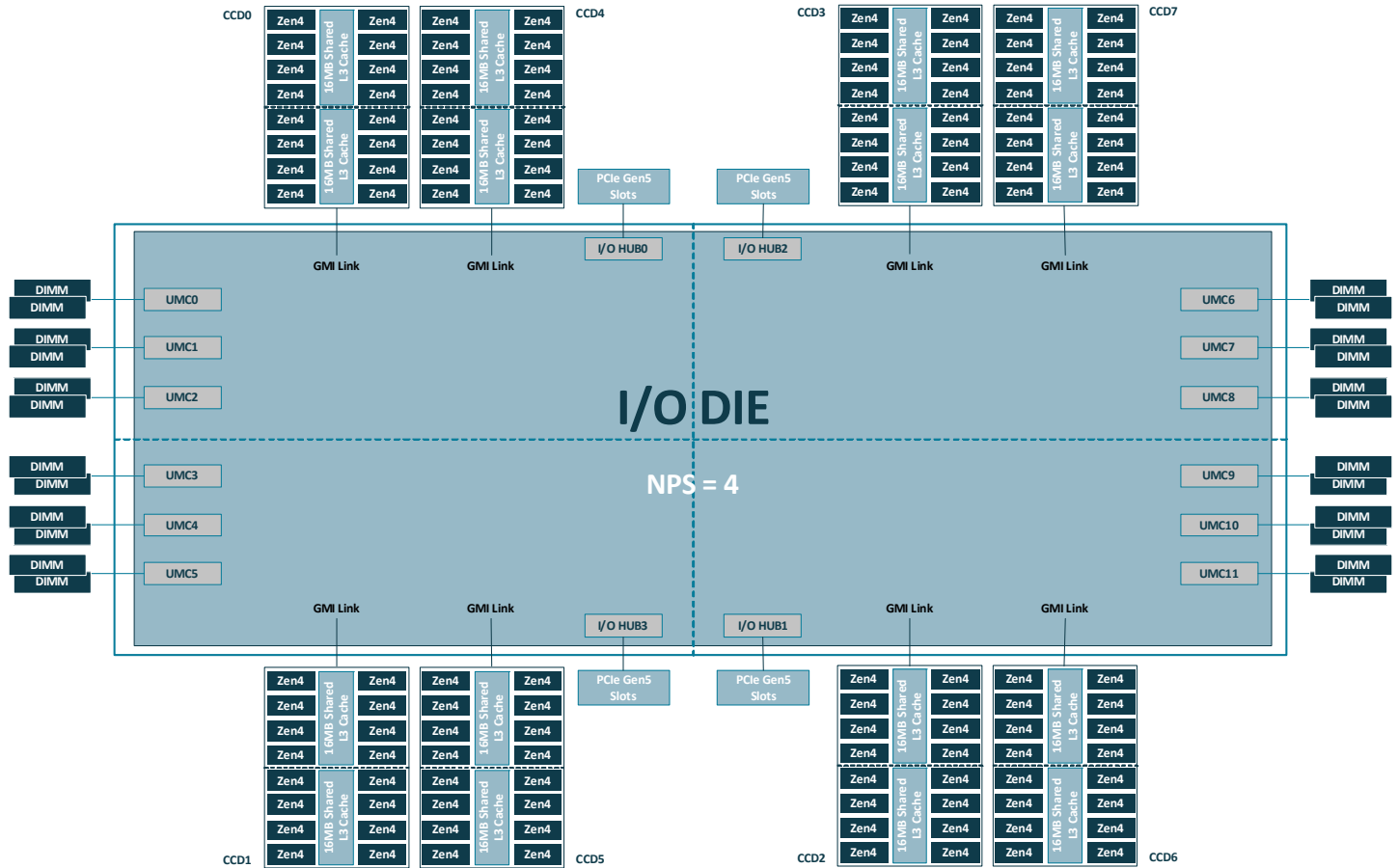


Figure 2-8: The AMD EPYC 9004 System on Chip (SoC) consists of up to 8 CCDs and a central IOD for 97xx models

## 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 10 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.

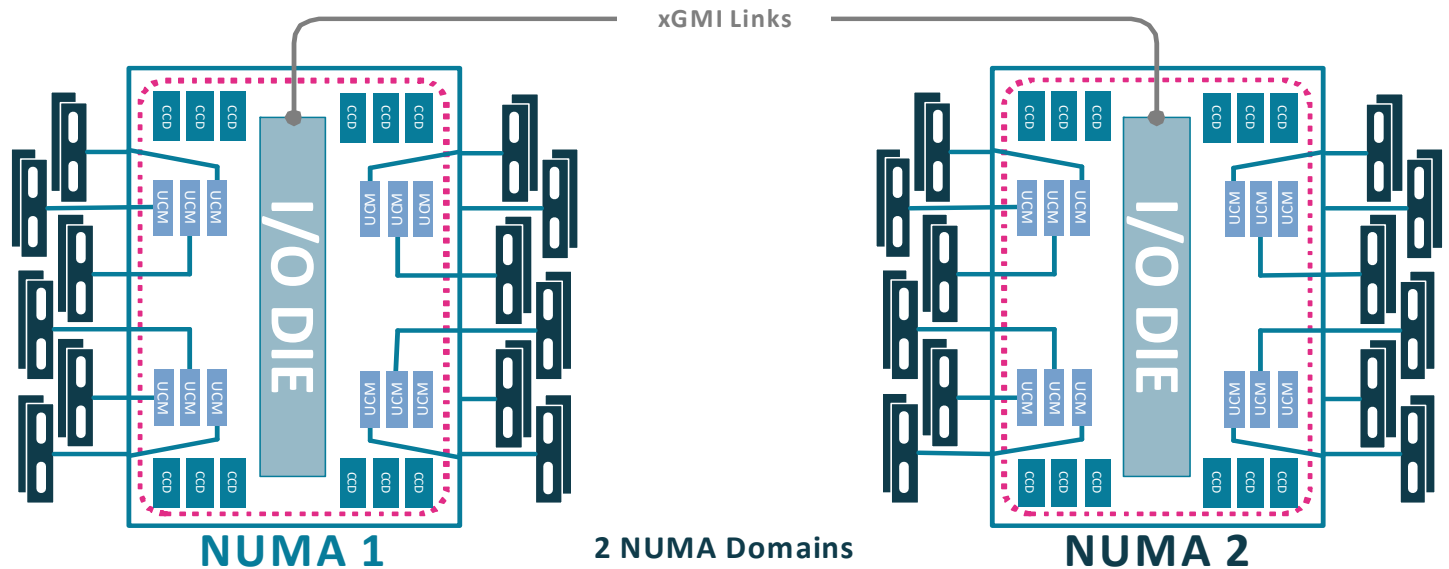


Figure 2-9: Two EPYC 9004 Processors connect through 4 xGMI links (NPS1)

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 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 workloads.

*Note: The default setting names 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	<ul style="list-style-type: none"> <li><b>Enabled/Auto:</b> Two hardware threads per core.</li> <li><b>Disabled:</b> Single hardware thread per core.</li> </ul>
L1 Stream HW Prefetcher	Auto	<ul style="list-style-type: none"> <li><b>Enabled/Auto:</b> Enables the prefetcher.</li> <li><b>Disabled:</b> Disables the prefetcher.</li> </ul>
L1 Stride Prefetcher	Auto	<ul style="list-style-type: none"> <li><b>Enabled/Auto:</b> Enables the prefetcher.</li> <li><b>Disabled:</b> Disables the prefetcher.</li> </ul>
L1 Region Prefetcher	Auto	<ul style="list-style-type: none"> <li><b>Enabled/Auto:</b> Enables the prefetcher.</li> <li><b>Disabled:</b> Disables the prefetcher.</li> </ul>
L1 Burst Prefetch Mode	Auto	<ul style="list-style-type: none"> <li><b>Enabled/Auto:</b> Enables the prefetcher.</li> <li><b>Disabled:</b> Disables the prefetcher.</li> </ul>
L2 Stream HW Prefetcher	Auto	<ul style="list-style-type: none"> <li><b>Enabled/Auto:</b> Enables the prefetcher.</li> <li><b>Disabled:</b> Disables the prefetcher.</li> </ul>
L2 Up/Down Prefetcher	Auto	<ul style="list-style-type: none"> <li><b>Enabled/Auto:</b> Enables the prefetcher.</li> <li><b>Disabled:</b> Disables the prefetcher.</li> </ul>
Core Performance Boost	Auto	<ul style="list-style-type: none"> <li><b>Enabled/Auto:</b> Enables Core Performance Boost.</li> <li><b>Disabled:</b> Disables Core Performance Boost.</li> </ul>
BoostFmaxEn	Auto	<ul style="list-style-type: none"> <li><b>Auto:</b> Use the default Fmax</li> <li><b>Manual:</b> User can set the boost Fmax</li> </ul>
BoostFmax	Auto	Specify the boost Fmax frequency limit to apply to all cores (MHz in decimal)
Global C-State Control	Auto	<ul style="list-style-type: none"> <li><b>Enabled/Auto:</b> Controls IO based C-state generation and DF C-states, including core processor C-States</li> <li><b>Disabled:</b> AMD strongly recommends not disabling this option because this also disables core processor C-States.</li> </ul>

Table 3-1: Processor core BIOS settings

X3D	Auto	<p>Enables or disables AMD 3D V-Cache™ technology on Cache Optimized (9004X) processors.</p> <ul style="list-style-type: none"><li>• <b>Auto:</b> 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</li><li>• <b>Disabled:</b> Disabling this option reduces the L3 cache in the CCD to 32MB.</li></ul> <p><i>Note: This option only applies to AMD EPYC 9004 Series Processors with AMD 3D V-Cache technology.</i></p> <p><i>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.</i></p>
-----	------	---

Table 3-1: Processor core BIOS settings

## 3.2 Power Efficiency Settings

Name	Default	Description
Power Profile Selection	Auto	<ul style="list-style-type: none"> <li>• <b>Auto/0:</b> High-performance mode</li> <li>• <b>1:</b> Efficiency mode</li> <li>• <b>2:</b> Maximum I/O performance mode</li> </ul>
Determinism Control	Auto	<ul style="list-style-type: none"> <li>• <b>Auto:</b> Use default performance determinism settings.</li> <li>• <b>Manual:</b> Specify custom performance determinism settings.</li> </ul>
Determinism Enable	Auto	<ul style="list-style-type: none"> <li>• <b>Auto:</b> Performance.</li> <li>• <b>1:</b> Power.</li> </ul>
TDP Control	Auto	<ul style="list-style-type: none"> <li>• <b>Auto:</b> Use platform- and OPN-default TDP.</li> <li>• <b>Manual:</b> Set custom configurable TDP.</li> </ul>
TDP	OPN Max	<p>This option appears once the user sets the <b>TDP Control</b> to <b>Manual</b>.</p> <ul style="list-style-type: none"> <li>• <b>Values 85-400:</b> Set configurable TDP, in watts.</li> </ul>
PPT Control	Auto	<p>Enables or disables the <b>PPT</b> control.</p> <ul style="list-style-type: none"> <li>• <b>Auto:</b> Automatically set PPL in watts.</li> <li>• <b>Manual:</b> Specify a custom PPL.</li> </ul>
PPT	OPN Max	<p>This option appears once the user sets the <b>PPT Control</b> to <b>Manual</b>.</p> <ul style="list-style-type: none"> <li>• <b>Values 85-400:</b> Set configurable PPT, in watts.</li> </ul>
CPPC	Auto	<ul style="list-style-type: none"> <li>• <b>Enabled/Auto:</b> Allows the OS to make performance/power optimization requests using ACPI CPPC.</li> <li>• <b>Disabled:</b> Prevents the OS from making performance/power optimization requests using ACPI CPPC.</li> </ul>

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	<ul style="list-style-type: none"> <li><b>Disabled (recommended):</b> Both NUMA nodes (<code>cpubind</code>) and memory interleaving (<code>membind</code>) are determined by the NPS setting.</li> <li><b>Enabled:</b> Overrides the NPS setting for # of NUMA nodes by mapping each LLC as a NUMA node. This does not impact the memory interleaving</li> </ul>
Nodes Per Socket (NPS)	1	<p><b>Memory Interleaving:</b> The <b>NPS</b> setting always determines the memory interleaving regardless of whether <b>LLC as NUMA</b> is <b>Enabled</b> or <b>Disabled</b>.</p> <p># of NUMA nodes (if <b>LLC as NUMA Domain</b> is <b>Disabled</b>):</p> <ul style="list-style-type: none"> <li><b>NPS1/Auto:</b> One NUMA node per socket (Most cloud providers use this as it provides consistent average memory latency to all the accesses within a socket).</li> <li><b>NPS2:</b> Two NUMA nodes per socket.</li> <li><b>NPS4:</b> Four NUMA nodes per socket</li> <li><b>NPS0 (not recommended):</b> Only applicable for dual-socket systems. A single NUMA node is created for the whole two-socket platform.</li> </ul> <p>AMD recommends either NPS1 or NPS4 depending on your use case.</p> <p><b>Windows systems:</b> Make sure that the number of logical processors per NUMA node is <math>\leq 64</math>. You can do this by using NPS2 or NPS4 instead of the default NPS1.</p>
Memory Target Speed	Auto	<ul style="list-style-type: none"> <li><b>Auto:</b> Determine the maximum memory speed based on SPD information from populated DIMMs and platform memory speed support.</li> </ul> <p>Alternatively, you can select:</p> <ul style="list-style-type: none"> <li><b>Values 3200–5600 MT/s:</b> Run the DRAM memory target clock speed at the specified speed. The DRAM memory target is the DDR rate.</li> </ul> <p>Your OEM system default value may vary.</p>
Memory Interleaving	Auto	<ul style="list-style-type: none"> <li><b>Auto/Enable:</b> Enables memory interleaving.</li> <li><b>Disable:</b> Allows for disabling memory interleaving. The <b>NUMA Nodes per Socket</b> setting will be honored regardless of this setting. AMD strongly recommends not disabling this setting because most production deployments benefit from memory interleaving.</li> </ul>

Table 3-3: NUMA and memory BIOS settings

### 3.4 Infinity Fabric Settings

Name	Default	Description
3-4 xGMI Link Max Speed	Auto	<ul style="list-style-type: none"> <li>• 12 Gbps</li> <li>• 16 Gbps</li> <li>• 17 Gbps</li> <li>• 18 Gbps</li> <li>• 20 Gbps</li> <li>• 22 Gbps</li> <li>• 23 Gbps</li> <li>• 24 Gbps</li> <li>• 25 Gbps/Auto</li> <li>• 26 Gbps</li> <li>• 27 Gbps</li> <li>• 28 Gbps</li> <li>• 30 Gbps</li> <li>• 32 Gbps</li> </ul> <p>Your OEM system default value may vary.</p>
xGMI Link Width Control	Auto	<ul style="list-style-type: none"> <li>• <b>Auto:</b> Use the default xGMI link width controller settings.</li> <li>• <b>Manual:</b> Specify a custom xGMI link width controller setting.</li> </ul>
xGMI Force Link Width Control	Auto	<ul style="list-style-type: none"> <li>• <b>Unforce:</b> Do not force the xGMI to a fixed width.</li> <li>• <b>Force:</b> Use the xGMI link to the user-specified width.</li> </ul>
xGMI Force Link Width	Auto	<ul style="list-style-type: none"> <li>• <b>0:</b> Force xGMI link width to x4.</li> <li>• <b>1:</b> Force xGMI link width to x8.</li> <li>• <b>2:</b> Force xGMI link width to x16.</li> </ul>
xGMI Max Link Width Control	Auto	<ul style="list-style-type: none"> <li>• <b>Auto:</b> Use the default xGMI link width controller settings.</li> <li>• <b>Manual:</b> Specify a custom xGMI link with controller setting.</li> </ul>
xGMI Max Link Width	Auto	<ul style="list-style-type: none"> <li>• <b>0:</b> Set max xGMI link width to x8.</li> <li>• <b>1:</b> Set max xGMI link width to x16.</li> </ul>
APBDIS	Auto	<ul style="list-style-type: none"> <li>• <b>0/Auto:</b> Dynamically switch the Infinity Fabric P-state based on link usage.</li> <li>• <b>1:</b> Enabled fixed Infinity Fabric P-state control.</li> </ul>
DfPstate Range Support	Auto	<ul style="list-style-type: none"> <li>• <b>Auto:</b> If this feature is enabled, the range value setting should follow the rule that <math>MaxDfPstate \leq MinDfPstate</math>. Otherwise, it will not work.</li> <li>• <b>Enable:</b> Add the values <math>MaxDfPstate</math> &amp; <math>MinDfPstate</math>.</li> <li>• <b>Disable:</b> No <math>MaxDfPstate</math> &amp; <math>MinDfPstate</math> option.</li> </ul>

Table 3-4: Infinity Fabric BIOS settings

DF C-States	Auto	<p>Controls DF C-states.</p> <ul style="list-style-type: none"> <li>• <b>Disabled:</b> Prevents the AMD Infinity Fabric from entering a low-power state.</li> <li>• <b>Enabled/Auto:</b> Allows the AMD Infinity Fabric to enter a low-power state.</li> </ul>
-------------	------	--

Table 3-4: Infinity Fabric BIOS settings

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

Name	Default	Description
Local APIC Mode	Auto(0x02)	<ul style="list-style-type: none"> <li>• <b>xAPIC:</b> Use xAPIC, supports up to 255 cores.</li> <li>• <b>x2APIC:</b> Supports more than 255 cores.</li> <li>• <b>Auto:</b> The system will choose the mode that best fits the number of active cores in the system.</li> <li>• <b>Compatibility:</b> Threads below 255 run in xAPIC with xAPIC ACPI structures, and threads 255 and above run in x2 mode with x2 ACPI structures.</li> <li>• <b>XApicMode (0x01):</b> Forces legacy xAPIC mode.</li> <li>• <b>X2ApicMode (0x02):</b> Forces x2APIC mode independent of thread count.</li> </ul>
PCIe Speed PMM Control	Auto	<ul style="list-style-type: none"> <li>• <b>0:</b> Dynamic link speed determined by power management functionality.</li> <li>• <b>1:</b> Static Target Link Speed (Gen4); sets the maximum idle link speed to 16 GT/s.</li> <li>• <b>Auto/2:</b> Static Target Link Speed (Gen5); sets the maximum idle link speed to 32 GT/s, thereby disabling the feature).</li> </ul>
PCIe ARI Support (SRIOV)	Auto	<ul style="list-style-type: none"> <li>• <b>Enabled/Auto:</b> Enables Alternative Routing ID interpretation.</li> <li>• <b>Disabled:</b> Disables Alternative Routing ID interpretation.</li> </ul>
PCIe Ten Bit Tag Support	Auto	<ul style="list-style-type: none"> <li>• <b>Enabled/Auto:</b> Enables PCIe 10-bit tags for supported devices.</li> <li>• <b>Disabled:</b> Disables PCIe 10-bit tags for all devices.</li> </ul>
IOMMU	Auto	<ul style="list-style-type: none"> <li>• <b>Enabled/Auto:</b> Enables IOMMU. AMD recommends setting this to <code>pt:pass-through</code> in the Linux kernel settings.</li> <li>• <b>Disabled:</b> Disables IOMMU.</li> </ul>
AVIC	Disabled	<p>Advanced Virtual Interrupt Controller.</p> <ul style="list-style-type: none"> <li>• <b>Disabled:</b> Disables AVIC.</li> <li>• <b>Enabled:</b> Enables AVIC.</li> </ul>
x2AVIC	Disabled	<p>x2AVIC is an extension of the advanced virtual interrupt controller. This feature currently requires a custom AMD Linux kernel.</p> <ul style="list-style-type: none"> <li>• <b>Disabled:</b> Disables x2AVIC.</li> <li>• <b>Enabled:</b> Enables x2AVIC.</li> </ul>

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

TSME	Auto	<ul style="list-style-type: none"> <li>• <b>Auto/Disabled:</b> Disables transparent secure memory encryption.</li> <li>• <b>Enabled:</b> Enables transparent secure memory encryption.</li> </ul>
SEV	Disabled	<p>In a multi-tenant environment (such as a cloud), Secure Encrypted Virtualization (SEV) mode isolates virtual machines from each other and from the hypervisor.</p> <ul style="list-style-type: none"> <li>• <b>Disabled:</b> SEV is disabled.</li> <li>• <b>Enabled:</b> SEV is enabled.</li> </ul>
SEV-ES	Disabled	<p>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.</p> <ul style="list-style-type: none"> <li>• <b>Disabled:</b> SEV-ES is disabled.</li> <li>• <b>Enabled:</b> SEV-ES is enabled.</li> </ul>
SEV-SNP	Disabled	<p>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.</p> <ul style="list-style-type: none"> <li>• <b>Disabled:</b> SEV-SNP is disabled.</li> <li>• <b>Enabled:</b> SEV-SNP is enabled.</li> </ul>

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

## 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.

# Hardware Configuration Best Practices

## 4.1 Memory Configuration

Proper memory subsystem configuration is crucial for optimum performance. I/O transfers data into or out of memory, so I/O bandwidth can never exceed memory subsystem capabilities. AMD recommends using a symmetric memory population for optimal system performance. Please see the latest version of [Memory Population Guidelines for AMD Family 19h Models 10h-1Fh](#) (login required) for additional details.

## 4.2 BIOS Settings

Tuning BIOS settings can improve performance for specific workloads. Evaluate all of the options discussed in this section to determine their impact on your workload.

Table 4-1 describes the BIOS options that most impact MongoDB performance using the BIOS parameters and settings found on AMD Customer Reference Boards (CRB), and OEM settings may vary. Please see your OEM BIOS documentation for platform-specific BIOS information. Please also see the *BIOS & Workload Tuning Guide for AMD EPYC™ 9004 Series Processors* (available from [AMD EPYC Tuning Guides](#)) for additional BIOS tuning information.

## 4.2.1 BIOS Settings for Maximizing Performance

Name	Value	Description
SMT Control	Enabled	Enabling Symmetric Multithreading (SMT) allows two hardware threads per core. You must enable AMD x2APIC with support more than 255 threads if you are using a system with dual 64-core AMD EPYC 9004 Series Processors. If you are running dual 64-core processors and your OS does not support AMD x2APIC, then you must disable SMT.
NUMA Node per Socket (NPS)	NPS1	This setting enables 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 by determining the number of NUMA nodes to split the memory channels between. Higher settings reduce memory channels per NUMA node, which lowers both throughput and latency for that NUMA node.
IOMMU	Enabled	Enabling IOMMU allows devices (such as the AMD EPYC processor-integrated SATA controller) to present separate IRQs for each attached device instead of one IRQ for the subsystem. Enabling IOMMU also allows operating systems to provide additional protection for DMA capable I/O devices. If you believe IOMMU is impeding performance, then enable it in BIOS and disable it via OS options, such as <code>iommu=pt</code> on the Linux kernel command line. Add <code>iommu=pt</code> (passthrough) to the grub file for optimal performance. In passthrough mode, the adapters need not use DMA translation to the memory, which improves performance.
Determinism Control	Power	This setting disables performance determinism and sets the determinism mode to <b>Power</b> .
ACPI SRAT L3 Cache as NUMA Domain	Disable	Controls automatic or manual generation of distance information in the ACPI System Locality Information MongoDB (SLIT) and NUMA proximity domains in the System Resource Affinity MongoDB (SRAT). Disabling this option disables reporting each L3 cache as a NUMA domain to the OS.
X3D	Enabled	This option only applies to AMD EPYC 9004 Series Processors 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. Disabling this option reduces the L3 cache in the CCD to 32MB. <i>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. Testing has shown AMD 3D V-Cache to be beneficial in most application workloads; however, AMD recommends testing and evaluating the benefits of this option for your application workload in your environment and using it accordingly.</i>

Table 4-1: Recommended BIOS settings

## Chapter

## 5

# Linux Optimizations

Tuning the Linux operating system's Memory and storage subsystems positively impacts MongoDB performance.

## 5.1 Memory Subsystem

MongoDB performs best when the application's working set (indexes and most frequently accessed data) fits in memory. RAM size is the most important factor for instance sizing; other optimizations may not significantly improve the performance of the database if there is insufficient RAM.

### 5.1.1 Swappiness

Change the swappiness setting to "1" for MongoDB.

The `zone_reclaim_mode` parameter allows you to set aggressive approaches to reclaim memory when a zone runs out of memory. Setting this to zero disables zone reclaim. You must disable `vm.zone_reclaim_mode` when NUMA is enabled.

The `dirty_ratio` is the percentage of total system memory that can hold dirty pages. Most Linux systems default to between 20-30%. Exceeding this limit commits the dirty pages to disk and creates a small pause. You can avoid this hard pause using the `dirty_background_ratio` that defaults 10-15% and tells the kernel to start flushing dirty pages to disk in the background without pausing.

On large-memory (64GB+) database servers, set the ratios in `/etc/sysctl.conf` as follows:

- `vm.dirty_ratio = 15`
- `vm.dirty_background_ratio = 5`, or possibly less.

```
vm.swappiness=1
vm.zone_reclaim_mode=0
vm.dirty_ratio = 15
vm.dirty_background_ratio = 5
```

### 5.1.2 Disabling Transparent Huge Pages (THP)

Linux distributions enable Transparent Huge Pages (THP) by default, which makes the kernel try to allocate memory in large chunks (usually 2MB) instead of 4K. THP is a Linux memory management system that reduces the overhead of Translation Lookaside Buffer (TLB) lookups on machines with large amounts of memory by using larger memory pages. However, database workloads often perform poorly with THP enabled because they tend to have sparse rather than contiguous memory access patterns. Disable THP when running MongoDB on Linux for best performance. Follow the procedures described in the [MongoDB Disable Transparent Huge Pages \(THP\)](#)\* documentation to disable THP by creating a service file.

### 5.1.3 NUMA Interleaving

Running MongoDB on a system with Non-Uniform Memory Access (NUMA) can cause a number of operational problems, including slow performance for periods of time, inability to use all available RAM, and high system process usage.

Use the `numactl --interleave` command to configure a memory interleave policy when running MongoDB servers and clients on NUMA hardware. Start the `mongod` process with `numactl --interleave=all`:

```
# numactl --interleave=all /usr/bin/mongod -f /etc/mongod.conf
```

If `systemd` is in use, then edit `/etc/systemd/system/multi-user.target.wants/mongod.service` so that the existing `ExecStart` statement reads as follows:

```
# ExecStart=/usr/bin/numactl --interleave=all /usr/bin/mongod --config /etc/mongod.conf
```

Restart any running `mongod` instances:

```
# sudo systemctl daemon-reload
# sudo systemctl stop mongod
# sudo systemctl start mongod
```

## 5.2 Storage Subsystem

MongoDB performs all read and write operations through in-memory data structures. Data is persisted to disk, and queries on data not already in RAM trigger a read from disk. Storage performance is therefore critical.

Most MongoDB disk access patterns do not have sequential properties, and using SSDs may deliver substantial performance gains. Use SSDs for read-heavy applications if the working set no longer fits in memory. AMD EPYC 9004 Series Processors support PCIe® Gen 5 connections that offer twice the I/O bandwidth of PCIe 4.0.

AMD recommends storing MongoDB journal files on a separate disk partition. Most MongoDB deployments should use RAID-10 storage configurations.

MongoDB supports a range of compression options for both documents and indexes. The default snappy compression typically reduces the storage footprint by 50% or more and enables higher IOPs as fewer bits are read from disk. All compression algorithms trade storage efficiency for CPU overhead, and the AMD EPYC SoC is capable of handling the extra load needed to handle compression. Indexes use prefix compression by default, which reduces the in-memory index storage footprint, thereby freeing up more RAM for frequently-accessed documents.

Most Linux systems use the ext4 virtual filesystem by default. AMD recommends using XFS to avoid performance issues observed when using EXT4 with WiredTiger.

## 5.2.1 Filesystem Mount Options

Most Linux system use the virtual EXT4 filesystem by default, but XFS is better for MongoDB because it offers slightly better performance for append-only workloads.

Name	Description
noatime	Disables updating the metadata associated with files in the filesystem with an updated access time. This tracking is superfluous because databases maintain their own accesses in their logs.
nobarrier	Disables the filesystem write barrier. Using a write barrier degrades I/O performance by requiring more frequent data flushes.

Table 5-1: XFS file system mount options

Use the following `xfs` filesystem mount options in `/etc/fstab`:

```
/dev/nvme0n1p1 /commitlogdir xfs noatime,nobarrier 0 0
```

Set the readahead setting between 8 and 32.

```
# blockdev --setra 32 <storage device path>
```

## 5.2.2 I/O Scheduler

Choosing the right disk I/O scheduler algorithm greatly improves performance.

- `deadline scheduler` is essentially a FIFO but does basic reordering and merging within the scheduler queue and guarantees a maximum latency for any given write in its queue.
- `noop` is a less-frequently-used option that works for MongoDB.
- Either `deadline` or `noop` will perform better than CFQ in bare metal installations.

```
# echo deadline > /sys/block/<dev>/queue/scheduler
```

`nr_requests`: The I/O request queue also offers opportunities for boosting performance. This queue determines how many objects can be essentially reordered before being flushed to disk. Longer queues mean better write ordering and fewer head movements a spinning disk will encounter when it starts writing to disk. For NVMe drives, set this to `1024` for `/sys/block/<dev>/queue/nr_requests`.

```
# echo 128 > /sys/block/<dev>/queue/nr_requests
```

Align filesystem I/O down to the physical devices. Unaligned I/O can multiply the number of on-disk writes incurred by any given logical write to your filesystem, which impedes I/O performance. Partition alignment is more critical when using SSD and NVMe drives.

Disk topography determines optimum alignment to a multiple of the physical block size so as to guarantee optimal performance. This example shows creating two partitions using `parted -a optimal` to create partitions that align to a multiple of the physical block size in a way that guarantees optimal performance:

```
# fdisk -l /dev/nvme0n1
# parted /dev/nvme0n1 mklabel gpt
# parted -a optimal /dev/nvme0n1 mkpart primary 0% 50%
# parted -a optimal /dev/nvme0n1 mkpart primary 51% 100%
```

The commit log and data dirs (`ssMongoDBs`) should be on different disks. If they both are on the same disk, then place them on separate partitions.

Set the following system limits in `/etc/security/limits.d/99-mongodb-nproc.conf`:

- -f (file size): unlimited
- -t (cpu time): unlimited
- -v (virtual memory): unlimited
- -l (locked-in-memory size): unlimited
- -n (open files): 64000
- -m (memory size): unlimited
- -u (processes/threads): 64000

## 5.3 Network Subsystem

Start tuning the adapter TCP/IP stack in Linux by making sure to use the maximum number of both transmit (TX) and receive (RX) ring buffers. Setting initial TX and RX network buffer sizes reduces the amount of post-boot time required for the network to reach an optimal performance state.

See the *Linux® Network Tuning Guide for AMD EPYC™ 9004 Series Processors* (available from [AMD EPYC Tuning Guides](#)) for information on configuring the network for systems in a MongoDB cluster and follow the guidelines therein to:

- Tune the TX and RX ring sizes.
- Change the number of interrupts queues to match the cores on the NUMA node which the NIC is collocated and pin those interrupts to the correct processor cores.

You can use the `iperf` utility to stress test the network infrastructure to verify proper setup. Be sure to properly set the OS IOMMU because this has significant performance impact on system performance. This is normally done by setting the IOMMU to pass-through mode by adding the kernel parameter `iommu=pt` on the kernel boot line. If you are using RHEL 8.x, then modify `/etc/default/grub` and run the `grub2-mkconfig` utility.

If you are using Windows, then please see the *Windows® Network Tuning Guide for AMD EPYC™ 9004 Series Processors* (available from [AMD EPYC Tuning Guides](#)) for network tuning information.

MongoDB is a distributed database that relies on efficient network transport during query routing and inter- node replication. The snappy compression algorithm compresses MongoDB network traffic across a cluster by up to 80% for significant performance boosts in bandwidth-constrained environments and reducing networking costs.

Enable compression by adding the `compressors` parameter to the connection string:

```
//localhost/?compressors=snappy
```

A distributed data store loads the network with read/write requests and replicating data across nodes. The network can become a bottleneck if the cluster includes enough high-performance NVMe drives for storage. The minimum required bandwidth is as low as 1000 Mb/s, but you can ensure sufficient network capacity by using 25GbE or higher Ethernet cards.

Set the following network kernel settings to prevent connections between nodes from timing out:

```
net.core.somaxconn = 4096
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 120
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_keepalive_probes = 6
```

Use the following settings to handle the thousands of concurrent connections used by the MongoDB database:

```
net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.core.rmem_default=16777216
net.core.wmem_default=16777216
net.core.optmem_max=40960
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 65536 16777216
```

## 5.4 tuned-adm Profile

The `tuned-adm` profile `throughput-performance` normally generates the best performance. This profile set up the overall system I/O and memory throughput by configuring the CPU governor, kernel scheduler granularity, disk read ahead, swappiness behavior, and dirty cache write back settings. See `/usr/lib/tuned/throughput-performance/tuned.conf` for these settings.

```
# yum install tuned -y
# systemctl start tuned
# tuned-adm profile throughput-performance
# systemctl enable tuned
# cpupower -c all frequency-set -g performance
```

## 5.5 NTP Configuration using Chrony

MongoDB requires an accurate system clock. Chrony is better at keeping the clocks in all MongoDB nodes in the cluster synchronized with the Network Time Protocol than `ntpd` for most networks than `ntpd` because:

- It is much faster than NTP at synchronizing to the time server. It can also compensate for fluctuating clock frequencies, such as when a host hibernates or enters sleep mode, or when the clock speed varies due to frequency stepping that slows clock speeds when loads are low.
- It handles intermittent network connections and bandwidth saturation and adjusts for network delays and latency.
- It never stops the clock after the initial update, which ensures stable and consistent time intervals for system services and applications. Please see [Using the Chrony suite to configure NTP](#) for detailed information.

```
# timedatectl list-timezones
# timedatectl set-timezone America/Los_Angeles; date # Now setup the Automatic NTP Timing
through Chrony
# systemctl status chronyd; systemctl enable chronyd; systemctl start
chronyd; chronyc tracking; hwclock -w; hwclock; date; ps -ef | grep [ch]rony
```

## 5.6 Stopping Linux Firewall and SELinux (If Required)

Disable the firewall on servers and SELinux if not required for your testing environment does not need them. IT policies may require firewalls in production environments.

```
# systemctl stop firewalld; systemctl disable firewalld; systemctl list- unit-files | grep
-i fire
# cp -p /etc/selinux/config /etc/selinux/config.ORIG; sed -i 's/SELINUX=enforcing/
SELINUX=disabled/g' /etc/selinux/config; grep -iw "SELINUX=disabled" /etc/selinux/config
# getenforce; sestatus; setenforce 0; sestatus; getenforce # reboot (This will make the
SELinux Disabled permanently)
```

## 5.7 Example RHEL Server Configuration Files

### 5.7.1 /etc/default/grub

```
# cat /etc/default/grub
...
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb iommu=pt
quiet"
...
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

### 5.7.2 /etc/rc.local

```
# cat /etc/rc.local
touch /var/lock/subsys/local

cpupower -c all idle-set -d 2
ethtool -G p2p1 rx 4096 tx 4096
/usr/sbin/set_irq_affinity_cpulist.sh 1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61 p2p1

echo "deadline" > /sys/block/nvme0n1/queue/scheduler
echo 1024 > /sys/block/nvme0n1/queue/nr_requests

echo "never" > /sys/kernel/mm/transparent_hugepage/enabled
echo "never" > /sys/kernel/mm/transparent_hugepage/defrag

echo 0 > /sys/class/block/sda/queue/rotational
echo 8 > /sys/class/block/sda/queue/read_ahead_kb
```

### 5.7.3 /etc/sysctl.conf

```
# cat /etc/sysctl.conf
vm.swappiness=1
vm.zone_reclaim_mode=0 vm.dirty_ratio = 15
vm.dirty_background_ratio = 5
vm.max_map_count=128000

net.core.somaxconn = 4096
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_time = 120
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_keepalive_probes = 6

net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.core.rmem_default=16777216
net.core.wmem_default=16777216
net.core.optmem_max=40960
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 65536 16777216
```

### 5.7.4 /etc/security/limits.conf

Increase the process & file limits only for MongoDB user, hardcoded system-wide:

```
# cat /etc/security/limits.d/99-mongodb-nproc.conf
mongod - fsize unlimited
mongod - cpu unlimited
mongod - as unlimited
mongod - nofile 64000
mongod - rss unlimited
mongod - nproc 64000
mongod - memlock unlimited
```



*This page intentionally left blank.*

# Chapter 6

# Performance Tuning MongoDB Settings

## 6.1 MongoDB Architecture

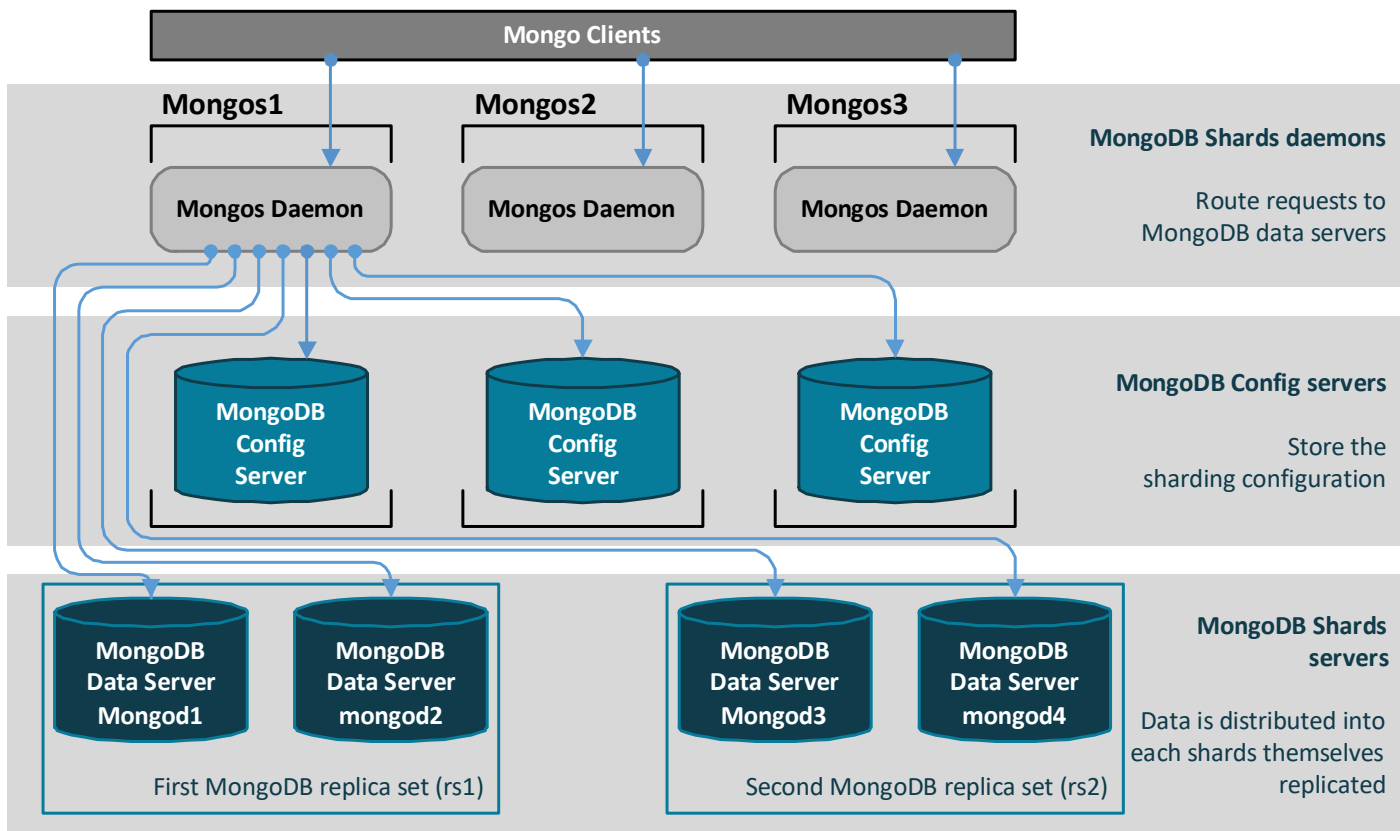


Figure 6-1: MongoDB architecture with Shards and ReplicaSet

## 6.2 MongoDB Tuning Best Practices

You must consider MongoDB best practices for the following in order to achieving optimal performance at scale:

- Data modeling and sizing memory (the working set)
- Query patterns and profiling
- Indexing
- Sharding

- Transactions and read/write concerns
- Hardware and OS configuration
- Benchmarking

MongoDB allows flexible schema designs. If you are planning to use sharded clusters for horizontal scaling, then your schema must include a shard key. The shard key affects read and write performance by determining how MongoDB partitions data. Verify that your shard key distributes the load evenly across your shards.

MongoDB prefers indexes. Fields in a document play the role of columns in a SQL database. Like columns, they can be indexed to increase search performance. A missing index forces searching every document within the collection to select the documents requested by the query, which can increase read times.

Working set sizing is another major performance optimization consideration. MongoDB performs best when the application's working set, indexes, and most-frequently-accessed data fits in memory. RAM size is the most important factor for instance sizing; other optimizations may not significantly improve database performance. If price/performance is more of a priority over performance alone, then you can use fast SSDs to complement smaller amounts of RAM. Be sure to test the optimum balance for your workload and SLAs.

If your working set exceeds the RAM of your chosen instance size or server, then consider either moving to a larger instance with more memory or partitioning (sharding) your database across multiple servers. MongoDB uses sharding to support deployments with very large data sets and high throughput operations.

There are two ways to optimize performance when the working set size grows:

- **Vertical Scaling:** Increasing the capacity of a single server, such as using a more powerful CPU, adding more RAM, or increasing the amount of storage space.
- **Horizontal Scaling:** Divides the system dataset and load over multiple servers, adding additional servers to increase capacity as required.

To shard a populated collection, the collection must have an index that starts with the shard key. The choice of shard key affects the performance, efficiency, and scalability of a sharded cluster. Selecting the wrong shard key can impede performance on a cluster with the best possible hardware and infrastructure.

MongoDB has built-in replication with auto-elections, which allows setting up a secondary database that can be auto-elected if the primary database becomes unavailable. However, MongoDB requires some setup (and possibly some support) to do replication. MongoDB has replica sets where one member is the primary and all others have a secondary role. The reads and writes are first committed to the primary replica and then replicated to the secondary replicas. MongoDB has a single master. The auto-elect process can take 10 to 40 seconds, during which you cannot write to the replica set.

Creating long-running transactions or attempting to perform an excessive number of operations in a single ACID transaction can pressure the WiredTiger storage engine cache. Consider the following to maintain predictable database performance levels:

- MongoDB allows you to specify the durability guarantee level (write concern) when issuing writes to the database. Write concerns can apply to any operation that executes against the database, regardless of whether it is a regular operation against a single document or wrapped in a multi-document transaction. You can configure write concerns can be configured on a per-connection, per database, per collection, or even per operation basis.
- Read concerns can apply to any query that executes against the database, regardless of whether it is a regular read against a single or set of documents or wrapped in a multi-document read transaction. The read concern

configuration can have a significant impact on latency. Supply a `maxTimeMS` value to timeout long-running operations.

## 6.3 Locking Performance

MongoDB uses a locking system to ensure data set consistency. If certain operations are long-running or a queue forms, then performance will degrade as requests and operations wait for the lock. Lock-related slowdowns can be intermittent. See the `locks` and `globalLock` sections of the `serverStatus` output to determine whether the lock is affecting performance. Long queries can be caused by:

- Ineffective index use.
- Suboptimal schema design.
- Poor query structure.
- System architecture issues.
- Insufficient RAM causing disk reads.

## 6.4 Number of Connections

The number of connections between the applications and the database can sometimes overwhelm the server's ability to handle requests. See the `locks` and `connections` sections of the `serverStatus` output. If there are numerous concurrent application requests, then the database may have trouble keeping up with demand. If this is the case, then increase the capacity of your deployment.

- **For read-heavy applications:** Increase the size of your replica set and distribute read operations to secondary members.
- **For write-heavy applications:** Deploy sharding and add one or more shards to a sharded cluster to distribute the load among MongoDB instances.

## 6.5 Fine-Grained Telemetry for MongoDB Performance Analysis

Tools like the explain plan, MongoDB Atlas Data Explorer, and MongoDB Query Profiler provide fine-grained telemetry and visibility across all database cluster components.

You can use the MongoDB [PY-TPCC](#)\* adaption of the TPC-C benchmark for MongoDB (implemented in Python) to evaluate different Atlas tiers or hardware configurations.

MongoDB's `explain ()` method allows you to test queries from your application and shows information about how a query will be, or was, resolved, including:

- The indexes that were used.
- Whether or not the index covered the query.
- Whether an in-memory sort was performed, which indicates an index would be beneficial.
- Number of index entries scanned.
- Number of documents returned, and the number read.

- How long the query took to resolve in milliseconds.
- Which alternative query plans were rejected (when using the `allPlansExecution` mode).

Use MongoDB Query Profiler to expose performance issues by displaying slow-running queries and their key performance statistics directly in the Atlas UI. This utility collects detailed information about operations and commands executed against a running MongoDB instance. All data collected by the profiler is written to the `system.profile` collection. This capped collection resides in the admin database and can be queried for insights. Logging levels can be configured based on the granularity of the data you want to analyze.

MongoDB Atlas features charts, custom dashboards, and automated alerting. It tracks 100+ key database and systems metrics including operations counters, memory, and CPU utilization, replication status, open connections, queues, and any node status.

`mongotop` lets you track how long a MongoDB instance `mongod` takes when reading and writing data. `mongotop` provides statistics on a per-collection level. By default, `mongotop` returns values every second.

The `mongostat` utility provides a quick status overview of a currently-running `mongod` or `mongos` instance using functionally similar to the UNIX/Linux `vmstat` filesystem utility.

## Chapter

## 7

## Resources

- [Memory Population Guidelines for AMD Family 19h Models 10h-1Fh](#) - Login required; please review the latest version if multiple versions are present.
- [Socket SP5 Platform NUMA Topology for AMD Family 19h Models 10h-1Fh](#) - Login required; please review the latest version if multiple versions are present.
- [AMD EPYC™ Processor Minimum Operating System \(OS\) Versions](#)
- From [AMD EPYC Tuning Guides](#):
  - *Windows® Network Tuning Guide for AMD EPYC™ 9004 Series Processors*
  - *Linux® Network Tuning Guide for AMD EPYC™ 9004 Series Processors*
  - *BIOS & Workload Tuning Guide for AMD EPYC™ 9004 Series Processors*
- [MongoDB Supported Platforms\\*](#)
- [MongoDB Production Notes\\*](#)
- [Performance Best Practices: MongoDB Data Modeling and Memory Sizing\\*](#)
- [mongotop\\*](#)
- [MongoDB Operations Checklist\\*](#)
- [Performance Best Practices: Transactions and Read / Write Concerns\\*](#)
- [TPC-C in Python for MongoDB\\*](#)
- [Manage NTP with Chrony\\*](#)
- [Documentation for /proc/sys/vm/\\*\(kernel version 2.6.29\)\\*](#)

*This page intentionally left blank.*

## Chapter

## 8

## 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
- **MDADM** - Multiple Disk and Device Administration
- **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
- **VM** - Virtual Machine

*This page intentionally left blank.*

# Chapter

# 9

# Processor Identification

Figure 9-1 shows the processor naming convention for AMD EPYC 9004 Series Processors and how to use this convention to identify particular processors models:

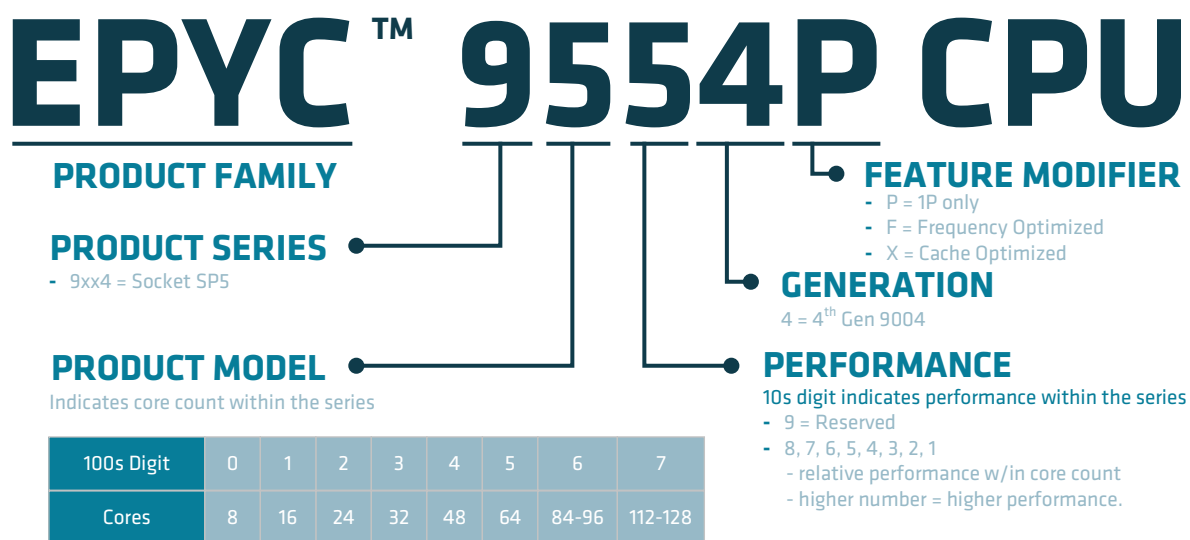


Figure 9-1: AMD EPYC SoC naming convention

## 9.1 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 Family 19h, Model 10h corresponds to an “A” part “Zen 4” CPU.
  - **91xx-96xx (including “X” OPNs):** Family 19h, Model 10-1F
  - **97xx:** Family 19h, Model 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.

## 9.2 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*.

### 9.2.1 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.