AMD EPYC[™] 9005

NGINX[®] Tuning Guide

AMD together we advance_data center computing

PID: 58489 v1.0 October 2024

© 2024 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, and combinations thereof are trademarks of Advanced Micro Devices, Inc. NGINX and the NGINX logo are trademarks of F5, Inc. in the US and other countries. 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
June, 2024	0.1	Initial NDA release
October, 2024	1.0	Initial public release

AUDIENCE

This document is intended for a technical audience such as NGINX[®] application architects, production deployment, and performance engineering teams with a server configuration background who have:

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



NGINX® TUNING GUIDE CONTENTS

CHAPTER 1 - INTRODUCTION	1
1.1 - About NGINX®	
1.2 - Important Reading	2
CHAPTER 3 - HARDWARE CONFIGURATIONS	v
3.1 - Memory Configurations	v
3.2 - Characteristics of AMD EPYC 9005 Platforms	V
3.3 - Recommended BIOS Settings	ν
CHAPTER 4 - NGINX ARCHITECTURE AND USE CASES	VII
4.1 - NGINX Architecture	vii
4.2 - NGINX as a Web Server	viii
CHAPTER 4 - NGINX TUNING METHODS AND RECOMMENDATIONS	ix
4.1 - Number of vCPU per NGINX Instance for Scale Up	x
4.2 - Scale out and Optimal Alignment with CCX/CCD and NUMA Nodes	xi
4.3 - Optimal CCX/CCD Alignment 4.3.1 - BIOS NPS Settings by NGINX Instance Size	
4.4 - Network Configuration	xiii
4.4.1 - Nic Interrupt Handling	
4.4.3 - Preferred I/O and PCIe Relaxed Ordering 4.4.4 - IOMMU Settings	XV
4.4.4 - IOMMO Settings	
4.5 - System Configurations	
4.5.2 - Max File Open	
4.5.3 - Configure Limits	xvi
4.5.4 - Sysctl Configuration	xvii

4.6 - NGINX Parameters	xvii
4.6.1 - Application Parameters Matching sysctl 4.6.2 - Worker_processes	xvii
4.6.2 - Worker_processes	xvii
4.6.3 - Worker connections	xvii
4.6.4 - Arcess Logging	xvii
4.6.5 - Sendfile and tcp_nopush	xviii
4.6.6 - Keepalive 4.6.7 - HTTP vs. HTTPS 4.6.8 - Caching and Compression	XVIII
4.b./ - HIIP vs. HIIPS	XVIII
	XVIII
4.7 - NGINX Build with the Latest GCC Compiler Using Znver5 Optimizations	
4.8 - Orchestration and Container Settings	xix
CHAPTER 5 - REFERENCES AND SETUP	XXI
5.1 - NGINX Setup for Cloud VMs	xxi
5.2 - NGINX Setup for Bare Metal	xxi
5.3 - NGINX Compilation with GCC	xxi
5.4 - Orchestration and Container Setup	xxii
Chapter 6 - Sample /etc/sysctl1.conf File	-xxIII
CHAPTER 7 - RESOURCES	- xxv



CHAPTER 1: INTRODUCTION

This tuning guide details parameters that can optimize performance on servers powered by AMD EPYC[™] 9005 Series Processors. OEM-default hardware and BIOS configurations are optimized for a wide range of workloads, and specific changes are often needed to provide optimal performance and scaling for specific workloads on all OS platforms. This tuning guide describes the following settings to optimize performance:

- Hardware configurations
- BIOS settings
- Network performance tuning
- OS kernel parameters
- NGINX performance tuning

1.1 - ABOUT NGINX®

NGINX[®] is a high-performance, open-source web server designed for both web content delivery and modern cloud architectures across a wide range of use cases such as HTTP, reverse proxy, mail, and a generic TCP/UDP proxy server. More than 400 million websites worldwide use NGINX web server and application delivery solutions to deliver their content quickly, reliably, and securely. NGINX's modular event-driven architecture can provide more predictable performance under high loads while maintaining low latencies and responsiveness. It handles requests using an asynchronous eventdriven approach instead of threads. NGINX can scale to hundreds of thousands of concurrent connections on modern hardware.

Simple packaging, configurability, and documents make NGINX easy to architect and deploy in on-premises, hybrid, or cloud environments using virtualization and containers with or without orchestration frameworks. Applying best practices for tuning the hardware platform, network configuration, operating system, and applications helps achieve optimal performance and full utilization. A balanced, fully-tuned platform should deliver superior performance compared to a reference system.

New platforms often feature changes to BIOS, cores, NUMA, memory population, network, OS, and even applications. The many changes can complicate identifying optimal configurations for a given platform. Performance benchmarks can help assess basic platform capabilities and help guide instance sizing, scaling up and scaling out, memory allocation, etc. One well-known test uses the wrk HTTP benchmarking tool along with the NGINX web server delivering web content. wrk data collection methodology and results allow users to understand all aspects of a given platform's web server performance. This tuning guide uses wrk and NGINX to validate most of the BIOS, OS, network, memory, NUMA placement and application-level parameter recommendations.

1.2 - IMPORTANT READING

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

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



CHAPTER 3: HARDWARE CONFIGURATIONS

3.1 - MEMORY CONFIGURATIONS

Proper memory subsystem configuration is important for achieving optimum performance testing. All I/O uses data transfers into or out of memory, and the I/O bandwidth can never exceed the memory subsystem limits. Achieving the maximum memory bandwidth on modern CPUs requires populating at least one DIMM in every DDR channel. Servers powered by AMD EPYC 9005 Series Processors include up to 12 DDR5 memory channels. AMD recommends populating all memory channels per socket, with every channel having the same speed and capacity DIMMs for optimal performance. Please see <u>Memory Population Guidelines for AMD Family 19h Models 10h–1F</u> for more details.

3.2 - CHARACTERISTICS OF AMD EPYC 9005 PLATFORMS

The CCDs connect to memory, I/O, and each other through the I/O Die (IOD). Each CCD connects to the IOD via a dedicated high-speed Global Memory Interconnect (GMI) link. The IOD also contains memory channels, PCIe[®] Gen5 lanes, and Infinity Fabric links. All dies (or chiplets) interconnect with each other via AMD's Infinity Fabric Technology.

3.3 - Recommended BIOS Settings

AMD provides default BIOS setting guidance to all OEMs and provides additional guidance for specific applications and workloads. Please see the BIOS & Workload Tuning Guide for AMD EPYC[™] 9005 Series Processor Based Servers</sup> (available from the <u>AMD Documentation Hub</u>) for additional tuning information.

THIS PAGE INTENTIONALLY LEFT BLANK.



CHAPTER 4: NGINX ARCHITECTURE AND USE CASES

AMD recommends deploying NGINX 1.22 or later for optimal performance, but some production deployments may use earlier versions for various reasons. AMD based the recommendations in this Tuning Guide on running NGINX as a web server, but most of them also apply when NGINX is deployed for reverse proxying, caching, load balancing, media streaming, etc.

4.1 - NGINX ARCHITECTURE

NGINX is a web server that offers high performance, scalability, and stability. NGINX can also function as a proxy server for email (IMAP, POP3, and SMTP) and as a reverse proxy and load balancer for HTTP, TCP, and UDP servers. NGINX handles multiple connections within a single process. It includes:

- A Primary process.
- Worker processes that handle the connections.
- Helper processes that manage the on-disk content cache.

NGINX can interleave multiple connections within a single Worker process and can switch from connection to connection almost instantaneously as new network traffic arrives, which allows handling heavy loads with minimal latency.

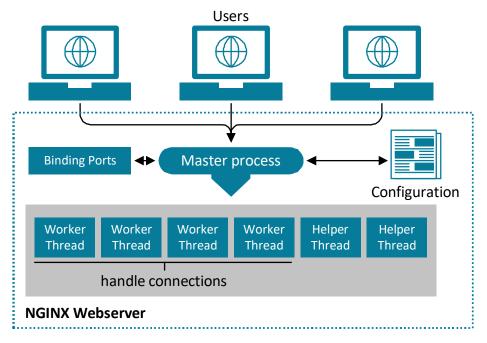


Figure 4-1: NGINX web server architecture

4.2 - NGINX AS A WEB SERVER

The wrk benchmark tool tests NGINX web server performance. wrk is lightweight and can easily handle multiple NGINX web server instances configured to deliver static or dynamic content ranging in size from ~1K bytes to many megabytes per request. It also provides an easy way to evaluate the impact of various deployment options such as compression, encryption, or local caching.

NGINX instance uses platform resources such as CPU cores and memory, cache, and network to respond to the client machine based on requests and assigned roles. This requires a balanced production environment. Here is a simple characterization of systems powered by AMD EPYC 9005 Series Processors based on findings gained by using wrk:

- Small (~1 KB) static and dynamic content requests require more compute and memory and scale well as more CPU cores are allocated to NGINX instance.
- Requests for large (>1 MB) static pages easily saturate a network bandwidth of ~100Gbps for most systems and will benefit from networkrelated optimizations and tunings.

Other important considerations that can significantly improve both throughput and response time include:

- Sizing NGINX instance for scale up and scale out.
- Platform-specific optimizations and tunings.



CHAPTER 4: NGINX TUNING METHODS AND RECOMMENDATIONS

This chapter describes the methodology and recommendations for tuning and optimizing NGINX instances for various deployments. A systematic approach follows these steps:

- 1. Number of vCPU cores per NGINX instance for scale up.
- 2. Scale out and optimal alignment with NUMA nodes.
- 3. Network configuration.
- 4. System configuration.
- 5. NGINX parameters.
- 6. NGINX builds using the latest GCC compiler.
- 7. Orchestration and container settings.

4.1 - NUMBER OF VCPU PER NGINX INSTANCE FOR SCALE UP

The web server's ability to handle requests depends on factors such as (but not limited to) the size of the content to be fetched, the processing capacity of the instance, and the number of incoming connections. You can adjust wrk to model production environment behavior and use those initial findings to determine both instance size and scale out behavior. 8 vCPU per instance delivers good performance, and 16 vCPU gives optimal performance when testing with wrk.

NGINX utilizes a hybrid disk-memory cache where all cached items are loaded into memory. For instances that require large caches, starting an instance with more cores (>8 vCPU up to 64 vCPU, depending on your specific AMD EPYC 9005 Series Processor) will open additional memory bandwidth to the NGINX instance.

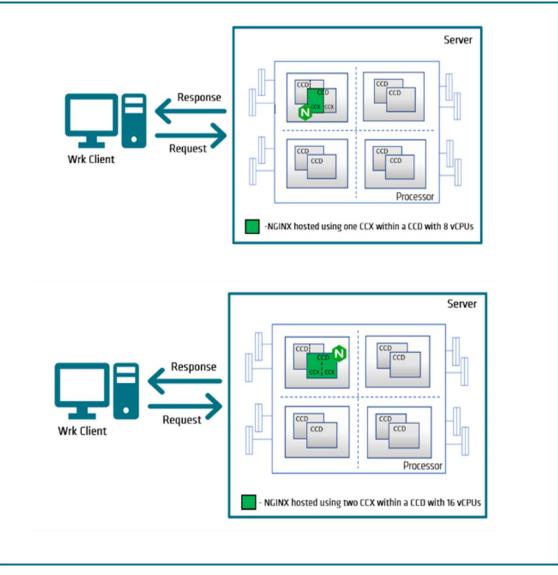


Figure 4-1: NGINX Scale-up solution examples of alignment with AMD EPYC 9005 Series Processors

Instances that are either cloud-based or that use VMs on-premises scale up well from 8vCPUs to 16 vCPUs and 32 vCPUs unless network bandwidth becomes a bottleneck. See <u>"Network Configuration" on page xili</u> for important network tuning information.

Requests with small or dynamic content scale well even for larger instances with greater than 64 vCPUs if network bandwidth is not a bottleneck. IOMMU must be set correctly because it plays a very important role in scaling beyond 32 vCPUs. See <u>"IOMMU Settings" on page xvi</u> for more details.

4.2 - SCALE OUT AND OPTIMAL ALIGNMENT WITH CCX/CCD AND NUMA NODES

Scale out strategy is the next step after determining the optimal number of vCPUs per NGINX instance. This involves a rough approximation of the overall scale out performance, deciding to align and pin instances to CCD and NUMA nodes, network, system, and NGINX parameters settings, and monitoring both CPU and network resources for bottlenecks.

Scale-out performance should be close to linear for cloud deployments where each additional NGINX instance will likely use a different node. Onpremise deployments should also scale well unless network bandwidth or other shared resources become a bottleneck. Again, monitor and evaluate CPU and network utilization.

The ability to pin NGINX instances to a given NUMA node and aligning within CCD boundaries becomes critical for consistent performance as additional NGINX instances are deployed. Failing to do this allows the vCPUs of a VM to be allocated to any CPU cores, which often lowers throughput and increases response time. Worst- case scenarios saw up to 20% performance difference when comparing 32 vCPU NGINX instances that perfectly aligned and pinned versus instances split evenly between two sockets with 16 vCPUs per socket. AMD observed this performance difference in both cloud and bare metal VMs and recommends adopting allocation policies that avoid splitting a single instance across the sockets.

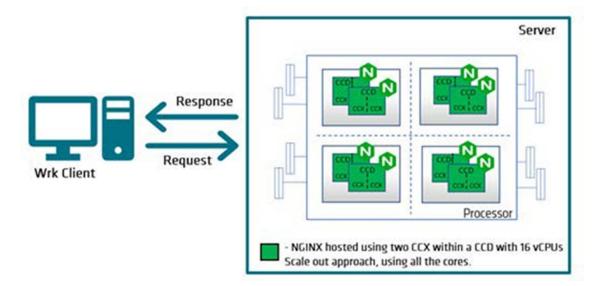


Figure 4-2: Sample NGINX scale out examples of alignment with AMD EPYC 9005 Series Processor CCDs

4.3 - OPTIMAL CCX/CCD ALIGNMENT

The AMD EPYC 9005 Series Processor architecture has multiple CCDs in each socket (1P), and each CCD has a single CCX unit. Each CCX has up to 8C/ 16T (Zen5, or "Classic") or 16C/32T (Zen5c, or "Dense") with each core having its own L1i, L1d, and L2 (1MB) caches. Each CCX also includes an L3 cache.

L3 cache is shared across all cores of a CCX. A thread on one CCD runs optimally when it accesses memory from the memory channels connected to the same CCD, and it therefore best to ensure that (for example) each NGINX instance runs on a single CCD with memory allocations/accesses occurring from the channels connected to that CCD.

Using the AMD EPYC 9005 Series Processor architecture (2P, 192C/384T) as an example, having either one NGINX instance per CCD (8C/16T) or 2 CCDs in the same quadrant(16C/32T) will perform better than having one NGINX instance with 8C/16C split between 2 CCDs from different quadrants or sockets. AMD further recommends keeping the vCPU size of NGINX VMs within one NUMA node.

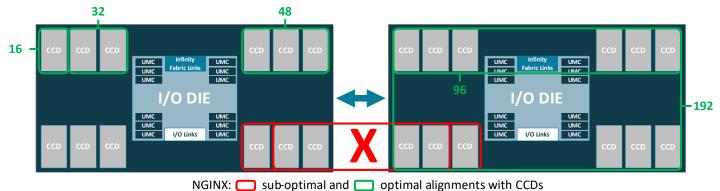


Figure 4-3: Example of pinning different # of vCPUs per NGINX aligned to CCX/CCD/NUMA nodes

AMD recommends pinning instances within a NUMA node while hosting/creating instances but does not recommend doing this via the applicationlevel worker_cpu_affinity option. The following sample configuration lowers performance by increasing the time a process spends waiting for a free CPU.

worker processes	4;			
worker_cpu_affinnity	0001	0010	0100	1000;

Note: AMD recommends maximizing performance by pinning NGINX instances or VMs or containers deploying NGINX to NUMA nodes at minimum, and also aligning them to CCD boundaries.

4.3.1 - BIOS NPS Settings by NGINX Instance Size

AMD EPYC 9005 Series Processors include Node Per Socket (NPS) settings of NPS1, NPS2, and NPS4 which create 1, 2, and 4 NUMA nodes per socket, respectively. Table 4-1 shows the recommended NPS settings for a 96-core processor based on NGINX instance sizes:

vCPUs per NGINX instance	NPS setting in BIOS
32 or less	NPS4
96	NPS2
192 or more	NPS1

Table 4-1: Recommended optimal NPS settings based on vCPUs per NGINX instance

Determine the optimal NPS setting for AMD EPYC 9005 Series Processors with fewer than 192 vCPUs based on the number of vCPUs per NGINX instance matching as closely as possible to the number of vCPUs in a NUMA node or that best fit within a NUMA node.

Note: If deployment restrictions prevent you from pinning VMs or NGINX instances, then NPS1 will deliver the most consistent performance because memory accesses within each socket will all be uniform. This is the best trade-off for this situation.

4.4 - NETWORK CONFIGURATION

Network configuration is one of the most important components in any NGINX deployment. Begin network configuration by applying the relevant recommendations from the *Linux® Network Tuning Guide for AMD EPYC 9005 Series Processors* (available from <u>AMD EPYC Tuning Guides</u>), which also includes additional recommendations for deployments that use an orchestration framework and containers. AMD recommends continually monitoring and recording network and disk I/O in both test and production environments.

4.4.1 - NIC Interrupt Handling

Having a number of interrupt queues that are equal to or less than the CPU cores per NUMA node optimizes performance. Assigning multiple queues to a single core risks thrashing the interrupt handler by swapping between queues. AMD recommends using a single queue per core.

There are three types of interrupt queues:

- Receive (RX)
- Transmit (TX)
- Combined. This uses a single queue to handle both RX and TX interrupts.

Note: Some vendors have separate RX and TX queues while others only implement combined queues.

Use lscpu output to determine the number of cores per NUMA node:

Architecture: CPU op-mode(s):	x86_64 32-bit, 64-bit
Address sizes:	52 bits physical, 57 bits virtual
Byte Order:	Little Endian
CPU(s):	384
On-line CPU(s) list:	0-383
Vendor ID:	AuthenticAMD
Model name:	AMD EPYC 9654 96-Core Processor
CPU family:	25

Model: 17 Thread(s) per core: 2 Core(s) per socket: 36 Setepping: 1 Frequency houst: subled CFU max Mix: 1000.000 CFU min Mix: The set set set of the set set ons rpae mcc ox8 apic sep mirr pie mcc amov pat pael6 Cifies min mix:		
Core (a) per socket: 96 Socket (a) per socket: 2 Stepping: 1 EVEN each photon: 2000 000 CPU min Mis: 1000.0000 ENGOMETRS: 4000.15 Flags: Fur work of a pass sear ht syncell nut maneaut face opt perhaps rates pinct age not conver a pass of CPU min Mis: 1000.0000 CPU min Mis: 1000000 CPU min Mis: 10000000000000	Model:	17
Socker(a): 2 Socker(a): 2 Socker(a): 2 Socker(a): 2 Steppin; 1 Frequency boost: enabled CC max ME: 1400.000 CC max ME: 1400.010 CC max ME: 1400.01	Thread(s) per core:	2
Stepping: 1 Frequency boost: enabled CPU mix MH2: 2400.000 CPU mix MH2: 1500.000 Plags: Tpu wm de pse tso mar pae mce x8 apic sep mtrr pge mcs cmov pat psel6 Ciflush mm fixer see sex save avx fåc rdrand lahf ln cmp legacy sum setapic cr0 legacy atm sse4a missilgnase 3dnowprefetch osw ibs skinit wit tce topeart perform core perform ab pext perform ab performance about the performance about the performance about performance about the performance about performance about the performa		96
Frequency Boost: enabled C2U mix MH2: 1500.000 http://fix.d00.0000 http://fix.d00.0000 http://fix.d00.0000 http://fix.d00	Socket(s):	2
CrU max MHz: 2400.000 CrU min MHz: 4500.000 BogodHTS: 4800.13 Flags: Lpu vmc de pse tao mar pae moe ox8 apic sep mtr pge moa omov pat pse36 Flags: Cpuid extd apicid apic Theorer api pi Columbudg monitors uses fina oxf pped sed 1 asi-2 motatop_tes cpuid extd apicid apic theorer api pi Columbudg monitors uses fina oxf pped sed 1 asi-2 motatop_tes cpuid extd apicid apic theorer apic pi Columbudg monitors uses fina oxf pped sed 1 asi-2 motatop_tes cpuid extd apicid apic theorer apic apic apic apic apic apic apic apic	Stepping:	1
CFU min MH:: 1500.0000 BogoMIPS: 1800.15 Flags: Fury ome de pae tac mar pae mee cas par see mir r pae mea cave pat paes6 ciflush mam fxar sse sse2 ht syscall nx mmxext fxar opt pdpeldp fdtacp im constant tac rep good nop1 nonstop tac cpuid extd apicid aperfmperf rap1 pai pfinulded monitor ssee5 fina cxif poid saf4 i sse1_2 nonstop tac cpuid extd apicid aperfmperf rap1 pai pfinulded monitor ssee5 fina cxif poid saf4 i sse1_2 misalinnss Schowpricterh osw ibs skinit wit toe toppest perfort core perfort noper perfort part part perfort core perfort core man to soll car any soll car	Frequency boost:	enabled
PeopdNTS:4400.15Flags:Fpu vme de pse tsc msr pae mcc cX8 apic sep mtrr pge mcc mcv pat pse36ciflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr opt pdpelgb rdtscp lm constant tsc rep_good noplnonstop tsc opuid ext daplcid aperfmperf rapp in piDmuldq monitor sses fna cxt6 poid sse4x2apic movbe popert ass xaave avx fl&c rdrand lahf in cmp legacy svm extapic cr8 legacy abm äse4amisalignsse admonpreteth oswn bis sknir ukt toe topoext perforr no bpext perforr. Dipext perforrwisalignsse admonpreteth oswn bis sknir ukt toe topoext perforr no bpext perforr. Dipext perforrwisalignsse admonpreteth oswn bis sknir ukt toe topoext perforr. Dipext perforrwisizidnsse admonpreteth oswn bis sknir ukt toe topoext perforr. Dipext perforr. Di	CPU max MHz:	2400.0000
Flágs:	CPU min MHz:	1500.0000
clflush mmx fxer see see2 ht syscall nx mmxekt fxer opt bdpeldp rdtscp im constant iso rep_good nopl nonstop tsc opuid extd apticid aperimperf rangt pni pClmuldqd monitor ssee3 fma cxtl iso fpoid sse42 x3apic movbe popert ase xsave avx fl6c rdrand lahf lm cmp_legacy sum extapic cr0 legacy sum iseda cms invoide cgm rdt avx512d avx512d rdraded adx smap avx512th profect and lahv 12bw avx512d vdraded avx512 avx512d rdraded adx smap avx512 min avx512bw avx512d rdraded avx512 vms12bw iso rmbn dots represent seave expetively taseves cgm llc cgm occup llc cgm mbm total cgm mbm local avx512 vms12 min avx512 vms12 vms12 vms12bw impacts iso vmcb clean avx512 vms2 vmcb rade for flush lld vwerflow recov succor sma farm flush lld Virtualifation: AMD-V Lld cache: AMD V Lld cache: AMD V Lld cache: AMD V Lld cache: AMD V NUMA node(s): 24 NUMA node(s): 32-33, 224-231 NUMA node(s): 40-47, 232-239 NUMA node(s): 40-47, 232-239 NUMA node(s) CPU(s): 40-47, 232-239 NUMA node	BogoMIPS:	4800.15
nonstop is capuid extd apicid apertmperf rapl pni p2lmulddg monitor seeses fram cxt6 pcid see4 1 see4 2 Xapic moves popent as Xaave avx file ordenal lah j mem piegacy was mextapic cr8 legacy abm See4 misalignese 3dnowprefetch osw ibs skinit wit toe toppert perfort ore perfort nb bpext perfort ll mwaitx erms invpcid cqn rdt a avx512f avx512d rdseed adx smap avx512ifma clflushopt clwb avx512d sha ni avx512 erms invpcid cqn rdt a avx512f avx512d rdseed adx smap avx512ifma clflushopt clwb avx512d sha ni avx512 erms invpcid cqn rdt a avx512f avx512d rdseed adx smap avx512ifma clflushopt clwb avx512d sha ni avx512 erms invpcid ecodeasits pausefilter pfthreshold avic v unsave wind hock nrip avv tso cale wncb/clean flushoptaid decodeasits pausefilter pfthreshold avic v unsave wind vgff v spec ctfl avx512 bffd clzero inperf saveept rdpru whonivd and ppin arit npt lbru svm lock nrip avv tso cale wncb/clean flushoptaid decodeasits pausefilter pfthreshold avic v unsave wind vgff v spec ctfl avx512 bffd verflow recov succor succ firm flush ld verflow recov succor succ firm flush verflow recov succor suck firm flush verflow recover succor suck form flush ver	Flags:	fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
x2apic movbe popent see save six fl&c rdrahd lahf_lm cmp_legacy sum extapic cfB legacy abm Sse4a — misalgnases Ganowprefactch osv ubs skinit wit tee topoext perfetr norm popet perfetr lle wwitx cpb cat_l3 cdp_l3 invpcid single hw pstate ssbd mba ibrs ipb stDp vmclal f5gsbase bml avx2 smep bml2 erms invpcid cdm rdt, avx512d rdraws12dg rdraws12d mpc lbrs vm Lock nrip save tos cale vmcb-lean flushbyssid decodeassists pausefilter pfthreshold avic v unsave vmload vgif v spec_ctl avx512/bf16 clzero invpcid varverpt rdpru vholinvd amd ppin ark npt_brv vm_Lock nrip save tos cale vmcb-lean flushbyssid decodeassists pausefilter pfthreshold avic v unsave vmload vgif v spec_ctl avx512/bf16 clzero invpcid for avis vmclmidqd avx312_vmn avx512_bf1ad avx312_vmcdat varb tos flushbyssid decodeassists pausefilter pfthreshold avic v unsave vmload vgif v spec_ctl avx512/bf16 clzero invpcid for avis vmclmidqd avx312_vmn avx512_bf1ad avx312_vmcdat varb tos flu cache: 6 MiB (192 instances) Lli cache: 192 MiB (192 instances) Ll cache: 768 MiB (24 instances) L3 cache: 768 MiB (24 instances) NUMA node(s): 24 NUMA node(s): 24 NUMA node(cF0(s): 0-7, 192-199 NUMA node(cF0(s): 0-7, 192-192 NUMA node(cF0(s): 0-7, 192-193 NUMA node(cF0(s): 0-7, 222-239 NUMA node(cF0(s): 0-6-63, 244-271 NUMA node(cF0(s): 0-6-63, 244-271 NUMA node(cF0(s): 0-6-63, 244-271 NUMA node(cF0(s): 0-6-63, 242-231 NUMA node(cF0(s): 0-6-103, 200-215 NUMA node(cF0(s): 0-6-103, 200-217 NUMA node(cF0(s): 0-6-103, 200-215 NUMA node(cF0(s): 0-6-103, 200-215 NUMA node(cF0(s): 0-6-103, 200-215 NUMA node(cF0(s): 0-6-103	clflush mmx fxsr sse sse2 ht sys	call nx mmxext fxsr opt pdpe1gb rdtscp lm constant tsc rep good nopl
<pre>misalignsse 3dnowprefetch oww ibs skinit wdt tee Topoext perfetr core perfetr nb Topskt perfetr lle mwaitx gp cat 12 cdg 13 invpcid sign hw pstate ssdd mab libs iopb stDp vmccll fögsbase hmil avx25120 wrs120t avwept xsdvec xgstbv1 saveec gn lic cgn occup llc cgn mbm total cgn mbm local avx5121 may 1 avx2 methy the savee transmitter avx121 may 1 avx2 methy the savee transmitter avx121 may 1 avx12 methy 1 avx12 meth</pre>		
cpb cai_l3 cdp_l3 invpcid_single hw pstate sabd mba ibrs ibpb stiDp vmCall f5gsbase bmil avx2=mep bmi2 erms invpcid com rdz avx512d vx512dg rdseed adx smap avx512/mm ciflushopt club avx512ds that avx512b avx512vl xsaveopt xsavec xgetbvl xsaves com lic com coup lic com mom total com mbm local avx512 bfi6 claro irperf xsaveeptr rdpru vbolnvd amd ppin aris npt lbrs vsm_lock nrip_save tss cased vmcb_lean flushbyasid decodeassists pausefilter pfthreshold avic v_unsave vmload vgif v_spec_ctll avx512vbmi umip pku ospke avx512 vbmi2 gfni vaes vpclmilqdq avx512_vniavx512_bitalg avx512_vpopcntdq la57 rdpid overflow recov succor smca fsrm flush lld Virtualization: AMD-V Lld cache: 6 MiB (192 instances) L2 cache: 192 MiB (192 instances) L2 cache: 192 MiB (192 instances) L2 cache: 768 MiB (192 instances) L2 cache: 192 MiB (192 instances) NOMA node(8): 24 NOMA node(8): 24 NOMA node(9): 24-11/21-223 NOMA node CFU(0): 44-55/240-247 NOMA node CFU(0): 42-55/240-247 NOMA node CFU(0): 56-63,248-255 NOMA node CFU(0): 56-63,248-255 NOMA node CFU(0): 66-71/22-279 NOMA node CFU(0): 68-95/240-287 NOMA node(2 CFU(0): 68-95/240-287 NOMA node(2 CFU(0): 64-71/256-263 NOMA node(2 CFU(0): 1623,208-295 NOMA node(1 CFU(0): 104-11/256-263 NOMA node(1 CFU(0): 104-11/256-263 NOMA node(1 CFU(0): 114-151,336-343 NOMA node(1 CFU(0): 116-133,208-295 NOMA node(1 CFU(0): 116-13,280-387 NOMA node(1 CFU(0): 116-13,280-387 NOMA node(1 CFU(0): 116-13,280-387 NOMA node(1 CFU(0): 116-11,296-303 NOMA node(1 CFU(0): 116-143,280-33 NOMA node(1 CF	x2apic movbe popcnt aes xsave av	x fl6c rdrand lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a
erms invpcid cgm rat, a vx512f avx512f avx512f avx512f avx512f avx512f avx512f avx512f avx512f avx512f savecpt såvec getbyt savec gm llc cgm llc cgm mbm total cgm nbm local avx512 bf16 clsero irperf xsavecptr rdpru whnolnvd amd ppin arat npt lbrv svm lock nrip save tsc scale vmcb clean flushbyssid decodessits pauses pilter pfthreshold avid v umsave vmload vgif 0° spec cfr1 avx512/bm1 umjp ku ospke avx512 vbm2 gfni vaes vpclmulqd avx512 vnni avx512 bital avx512 vpopcntdq la57 rdpid voverflow recov succor smca farm lld cache: AMD-V Lld cache: AMD-V Lld cache: Cache:		
avx512vl saveopt sävec sqetbu xaves cgm llc cgm occup llc cgm mot tol cgm mbm local avx512 bf16 claero irperf xavecerptr rdpru whoinvd amd ppin arit npt lbrv svm lock nrip.save tsc scale vmcb Clean flushbyasid decodeassists pausefilter pfthreshold avic v vmsave vmload vyif v spec ctl avx512vbmi umjp pku ospke avx512 vhniu gfni vaes vpclmulqq avx512_nni avx512_bild avx512_vppcntdq la57 rdpid overflow recov succor smca farm flush lld virtualization: AMD-V Lld cache: 6 M1B (192 instances) L1 cache: 192 M1B (192 instances) L2 cache: 192 M1B (192 instances) L3 cache: 768 M1B (24 instances) MUMA node(8): 24 NUMA node(8): 24 NUMA node(9): 24-31,216-221 NUMA node(CPU(8): 24-31,216-221 NUMA node(CPU(8): 24-31,216-221 NUMA node(CPU(8): 24-32,224-231 NUMA node(CPU(8): 32-39,224-231 NUMA node(CPU(8): 36-63,242-255 NUMA node(CPU(8): 36-63,242-255 NUMA node(CPU(8): 36-63,242-255 NUMA node(CPU(8): 36-87,22-279 NUMA node(CPU(8): 36-87,22-279 NUMA node(CPU(8): 36-87,22-273 NUMA node(CPU(8): 36-87,22-273 NUMA node(CPU(8): 36-87,22-235 NUMA node(CPU(8): 36-13,288-295 NUMA node(CPU(8): 164-11,256-263 NUMA node(CPU(8): 120-127,312-319 NUMA node(C		
claency irperf xaveerptr rdpru whnoinvd and_ppin ardt npt_lbrv svm lock nrip_save tic scale vmcb_clean flushysavd decodeasists pauses filter pfthreshold avit v musey vmload vgif v gsec ctrl avx512/bital avx512/bit		
flushbysid decodeasists pipuefilter pfthreshold avic v vmaave vmload vgif v spec_trl avx512/bmi umip pku ospke avx512 vmi2 gfni vess vpclmulqd avx512_vnni avx512_bialg avx512_vpopentdg la57 rdpid overflow_recov succor smc fsm flush lld Virtualization: AMP-V Lld cache: 6 MiB (192 instances) Ll cache: 192 MiB (192 instances) L2 cache: 768 MiB (24 instances) NUMA node(s): 24 NUMA node(s): 24 NUMA node(cPU(s): 24-31,216-223 NUMA node2 CPU(s): 72-79,264-271 NUMA node2 CPU(s): 8-15,200-207 NUMA node2 CPU(s): 8-23,224-231 NUMA node2 CPU(s): 8-56-33,224-235 NUMA node2 CPU(s): 8-64-7,272-279 NUMA node2 CPU(s): 8-64-7,272-279 NUMA node2 CPU(s): 8-64-7,272-279 NUMA node2 CPU(s): 8-64-7,232-230 NUMA node2 CPU(s): 8-64-7,232-230 NUMA node2 CPU(s): 8-72-8,208-215 NUMA node2 CPU(s): 9-6103,288-295 NUMA node1 CPU(s): 106-17,302-310 NUMA node1 CPU(s): 104-11,256-263 NUMA node1 CPU(s): 104-11,256-263 NUMA node1 CPU(s): 104-11,256-263 NUMA node1 CPU(s): 104-11,256-303 NUMA node1 CPU(s): 104-11,256-303 NUMA node1 CPU(s): 104-11,266-303 NUMA node1 CPU(s): 104-11,286-303 NUMA node21 CPU(s): 104-11,286-305 NUMA node21 CPU(s): 104-11,286-305 NUMA node21 CPU(s): 104-11,286-305 NUMA node		
pku ospike avx512 vbmi2 gfni væs vpclmulqdq avx512_vni āvx512_Ditalg avx512_vpopcātdq la57 rdpid verflow recov sūcor smoa fsrm flush lld Virtualization: AMD-V L1d cache: 6 MiB (192 instances) L2 cache: 192 MiB (192 instances) L3 cache: 768 MiB (192 instances) L3 cache: 768 MiB (192 instances) L3 cache: 768 MiB (192 instances) MUMA node(s): 24 NUMA node(c): 24 NUMA node: 201(s): A=55,240-247 NUMA node: 201(s): NUMA node: 201(s): S2-302-2423 NUMA node: 201(s): NUMA node: 201(s): S2-302-2423 NUMA node: 201(s): NUMA node: 201(s): S2-302-2423 NUMA node: 201(s): NUMA node: 201(s):<		
overflow recov sūcor sīca fsrm flush 11d AMD-7 Virtualization: AMD-7 L1d cache: 6 MiB (192 instances) L1 cache: 192 MiB (192 instances) L2 cache: 192 MiB (192 instances) L3 cache: 768 MiB (24 instances) NUMA node(s): 24 NUMA node(s): 24 NUMA node(s): 24 NUMA node(CPU(s): 0-7,192-199 NUMA node2 CPU(s): 24-31,216-223 NUMA node2 CPU(s): 8-15,200-207 NUMA node3 CPU(s): 8-15,200-207 NUMA node5 CPU(s): 8-56,220-217 NUMA node5 CPU(s): 8-57,220-217 NUMA node5 CPU(s): 16-23,208-215 NUMA node2 CPU(s): 10-47,232-239 NUMA node1 CPU(s): 10-47,232-239 NUMA node1 CPU(s): 12-56-263 NUMA node1 CPU(s): 12-73,268-295 NUMA node1 CPU(s): 128-135,320-327 NUMA node1 CPU(s): 128-135,320-327 NUMA node1 CPU(s): 128-135,320-327 NUMA node1 CPU(s): 128-135,320-327 NUMA		
VirtualiZation: AMD-V L1d cache: 6 MiB (192 instances) L1 cache: 6 MiB (192 instances) L2 cache: 192 MiB (192 instances) L3 cache: 768 MiB (24 instances) NUMA node(s): 24 NUMA node(s): 24 NUMA node CPU(s): 24-11/216-223 NUMA node CPU(s): 24-31/216-223 NUMA node CPU(s): 24-31/216-223 NUMA node CPU(s): 64-57/20-247 NUMA node CPU(s): 64-5/200-207 NUMA node CPU(s): 66-63/248-255 NUMA node CPU(s): 64-7/272-279 NUMA node CPU(s): 64-7/232-239 NUMA node10 CPU(s): 64-7/1,256-263 NUMA node11 CPU(s): 64-11/2,256-263 NUMA node12 CPU(s): 124-137,312-319 NUMA node12 CPU(s): 124-137,332-319 NUMA node12 CPU(s): 124-137,332-343 NUMA node12 CPU(s): 124-137,332-319 NUMA node12 CPU(s): 124-137,332-343 NUMA node21 CPU(s): 124-137,332-343 NUMA node12 CPU(s): 124-137,320-327		
Lld cache: 6 MiB (192 instances) Ll cache: 192 MiB (192 instances) L2 cache: 192 MiB (192 instances) NUMA node(5): 24 NUMA node(2): 0-7,192-199 NUMA node2 CPU(s): 24-31,216-223 NUMA node2 CPU(s): 24-31,216-223 NUMA node2 CPU(s): 24-31,216-223 NUMA node2 CPU(s): 84-55,240-247 NUMA node6 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 56-63,248-255 NUMA node6 CPU(s): 60-63,248-255 NUMA node6 CPU(s): 60-61,228-239 NUMA node6 CPU(s): 64-71,256-263 NUMA node1 CPU(s): 68-95,280-287 NUMA node1 CPU(s): 120-127,312-319 NUMA node1 CPU(s): 120-127,312-319 NUMA node16 CPU(s): 104-11,296-303 NUMA node16 CPU(s): 104-11,296-305 NUMA node16 CPU(s): 128-13,320-337 NUMA node20 CPU(s): 120-127,312-319 NUMA node18 CPU(s): 126-13,328-375 NUMA node20 CPU(s): 126-13,328-335 NUMA node20 CPU(s): 160-167,322-359 NUMA node20 CPU(s): 160-167,322-3		
Lli cache: 6 MiB (192 instances) L2 cache: 192 MiB (192 instances) L3 cache: 768 MiB (24 instances) NUMA node(CFU(s): 0-7,192-199 NUMA node2 CFU(s): 24-31,216-223 NUMA node2 CFU(s): 24-31,216-223 NUMA node2 CFU(s): 72-79,264-271 NUMA node2 CFU(s): 72-79,264-271 NUMA node5 CFU(s): 32-39,224-231 NUMA node6 CFU(s): 32-39,224-231 NUMA node5 CFU(s): 32-39,224-231 NUMA node5 CFU(s): 86-15,240-255 NUMA node5 CFU(s): 16-23,208-215 NUMA node1 CFU(s): 86-95,240-215 NUMA node1 CFU(s): 64-71,232-239 NUMA node1 CFU(s): 64-71,236-263 NUMA node1 CFU(s): 16-21,288-295 NUMA node1 CFU(s): 168-175,360-367 NUMA node1 CFU(s): 168-175,360-367 NUMA node1 CFU(s): 168-175,360-367 NUMA node1 CFU(s): 128-135,320-327 NUMA node1 CFU(s): 128-135,320-335 NUMA node2 CFU(s): 128-135,320-335 NUMA node2 CFU(s): 128-135,320-335 NUMA node2 CFU(s): 128-135,320-335 NUMA node2 CFU(s): 136-143,328-335 NUMA node2 CFU(s): 140-11,375-335 NUMA node2 CFU(s): 140-11,375-335 NUMA node2 CFU(s): 140-141,375-335 NUMA node2 CFU(s): 141,376-335 NUMA node2 CFU(s): 141,376-335 NUMA node2 CFU(s): 144-141,376-335 NUMA node2 CFU(s): 144-141,376-345 YUHerability Mds: Not affected YUHerability Mds: Not affected YUHerability		
L2 cache: 192 MiB (192 instances) L3 cache: 768 MiB (24 instances) NUMA node(s): 24 NUMA node(S): 24 NUMA node1 CPU(s): 24-31,216-23 NUMA node2 CPU(s): 24-31,216-23 NUMA node3 CPU(s): 72-79,264-271 NUMA node5 CPU(s): 72-79,264-271 NUMA node5 CPU(s): 8-15,200-207 NUMA node5 CPU(s): 8-15,200-207 NUMA node6 CPU(s): 8-15,200-207 NUMA node6 CPU(s): 8-15,200-207 NUMA node6 CPU(s): 8-23,208-235 NUMA node7 CPU(s): 80-87,272-279 NUMA node6 CPU(s): 16-23,208-215 NUMA node7 CPU(s): 80-87,272-279 NUMA node10 CPU(s): 64-71,226-263 NUMA node10 CPU(s): 98-95,280-287 NUMA node11 CPU(s): 120-127,312-319 NUMA node16 CPU(s): 120-127,312-319 NUMA node16 CPU(s): 168-175,360-367 NUMA node16 CPU(s): 128-135,320-367 NUMA node16 CPU(s): 128-135,320-327 NUMA node18 CPU(s): 128-135,320-327 NUMA node18 CPU(s): 128-135,320-327 NUMA node20 CPU(s): 136-147,322-335 NUMA node20 CPU(s): 136-147,322-335 NUMA node20 CPU(s): 136-147,320-335 NUMA node20 CPU(s): 136-147,320-335 NUMA node20 CPU(s): 136-147,320-335 NUMA node20 CPU(s): 142-119,304-311 NUMA node20 CPU(s): 136-141,328-335 NUMA node20 CPU(s): 142-119,304-311 NUMA node20 CPU(s): 142-119,304-315 NUMA node20 CPU(s): 142-119,304-315 NUMA node20 CPU(s): 142-119,304-315 NUMA node20 CPU(s): 141,376-383 Vulnerability Meltdwm: Not affected Vulnerability Meltdwm: Not affected Vulnerability Meltdwm: Not affected Vulnerability Meltdwm: Not affected Vulnerability Spectre v1: Mitigationr sercopy/swages bariers and _us		
L3 cache: 768 MiB (24 instances) NUMA node(s): 24 NUMA node(s): 24 NUMA node(s): 24-31,216-223 NUMA node2 CPU(s): 44-55,240-247 NUMA node3 CPU(s): 72-79,264-271 NUMA node4 CPU(s): 8-15,200-207 NUMA node5 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 80-87,272-279 NUMA node8 CPU(s): 16-23,208-215 NUMA node1 CPU(s): 80-87,272-279 NUMA node1 CPU(s): 160-17,282-235 NUMA node1 CPU(s): 80-95,280-287 NUMA node10 CPU(s): 80-95,280-287 NUMA node11 CPU(s): 80-87,312-319 NUMA node12 CPU(s): 120-127,312-319 NUMA node13 CPU(s): 120-127,312-319 NUMA node15 CPU(s): 168-175,360-367 NUMA node16 CPU(s): 128-135,320-327 NUMA node16 CPU(s): 128-135,320-327 NUMA node16 CPU(s): 128-135,320-327 NUMA node16 CPU(s): 128-135,320-327 NUMA node20 CPU(s): 128-137,320-327 NUMA node20 CPU(s): 136-143,328-335 NUMA node20 CPU(s): 136-143,328-335 NUMA node20 CPU(s): 1419,1376-383 Vulnerability Hib multihit: Not affected Vulnerability Mdet Not Affected Vulnerability Mdet Not affected Vulnerability Metledwn: Not affected Vulnerability Metledwn: Not affected Vulnerability Spectre V1: Mitigation, Speculative Store Bypass disabled via prc1 and seccomp Vulnerability Spectre V2: Mitigation, Speculative Store Bypass disabled via prc1 and seccomp Vulnerability Spectre V1: Mitigation, Speculative Store Bypass disabled via prc1 and seccomp Vulnerability Spec		
NUMA node(s): 24 NUMA node0 CPU(s): 0-7,192-199 NUMA node1 CPU(s): 24-31,216-223 NUMA node3 CPU(s): 48-55,240-247 NUMA node3 CPU(s): 72-79,264-271 NUMA node6 CPU(s): 8-15,200-207 NUMA node6 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 56-63,248-255 NUMA node0 CPU(s): 16-23,208-215 NUMA node10 CPU(s): 64-71,256-263 NUMA node12 CPU(s): 64-71,256-263 NUMA node12 CPU(s): 9-103,288-295 NUMA node12 CPU(s): 120-127,312-319 NUMA node13 CPU(s): 168-175,360-367 NUMA node16 CPU(s): 104-11,296-303 NUMA node17 CPU(s): 152-159,344-351 NUMA node20 CPU(s): 162-133,388-375 NUMA node21 CPU(s): 116-143,328-335 NUMA node22 CPU(s): 160-167,352-359 NUMA node23 CPU(s): 184-191,376-383 Vulnerability Lth multihit: Not affected Vulnerability Mds: Not affected Vulnerability Mds Not affected Vulnerability Spectre V1: Mitigation; Secong/swages disabled via prct1 and seccomp		
NUMA node0 CPU(s): 0-7,192-199 NUMA node1 CPU(s): 24-31,216-223 NUMA node2 CPU(s): 48-55,240-247 NUMA node3 CPU(s): 72-79,264-271 NUMA node6 CPU(s): 8-15,200-207 NUMA node6 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 56-63,248-255 NUMA node6 CPU(s): 64-71,256-263 NUMA node6 CPU(s): 64-71,256-263 NUMA node10 CPU(s): 80-87,272-279 NUMA node10 CPU(s): 64-71,256-263 NUMA node10 CPU(s): 80-87,272-279 NUMA node10 CPU(s): 80-87,272-279 NUMA node10 CPU(s): 16-71,256-263 NUMA node10 CPU(s): 120-127,312-319 NUMA node11 CPU(s): 120-127,312-319 NUMA node12 CPU(s): 104-111,296-303 NUMA node13 CPU(s): 104-111,296-303 NUMA node16 CPU(s): 112-115,320-327 NUMA node17 CPU(s): 112-115,320-327 NUMA node20 CPU(s): 112-113,304-311 NUMA node20 CPU(s): 112-113,304-311 NUMA node20 CPU(s): 106-143,328-335 NUMA node20 CPU(s): 160-143,328-355 NUMA node20 CPU		
NUMA nodel CPU(s): 24-31,216-223 NUMA node2 CPU(s): 72-79,264-271 NUMA node3 CPU(s): 8-15,200-207 NUMA node6 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 56-63,248-255 NUMA node7 CPU(s): 80-67,272-279 NUMA node0 CPU(s): 40-47,232-239 NUMA node1 CPU(s): 64-71,256-263 NUMA node1 CPU(s): 96-103,288-285 NUMA node1 CPU(s): 120-127,312-319 NUMA node12 CPU(s): 120-127,312-319 NUMA node14 CPU(s): 144-151,336-343 NUMA node16 CPU(s): 104-111,296-303 NUMA node16 CPU(s): 104-111,296-303 NUMA node17 CPU(s): 128-135,320-327 NUMA node18 CPU(s): 128-135,320-327 NUMA node18 CPU(s): 128-133,320-327 NUMA node19 CPU(s): 126-167,322-359 NUMA node20 CPU(s): 116-167,322-359 NUMA node20 CPU(s): 164-191,376-383 Vulnerability Itb multihit: Not affected Vulnerability Mis: Not affected Vulnerability Mis Not affected Vulnerability Mis Stale data: Not affected		
NUMA node2 CPU(s): 48-55,240-247 NUMA node3 CPU(s): 72-79,264-271 NUMA node3 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 56-63,248-255 NUMA node6 CPU(s): 80-87,272-279 NUMA node8 CPU(s): 64-71,256-263 NUMA node10 CPU(s): 64-71,256-263 NUMA node10 CPU(s): 88-95,280-287 NUMA node11 CPU(s): 88-95,280-287 NUMA node13 CPU(s): 164-17,256-263 NUMA node14 CPU(s): 164-17,256-263 NUMA node13 CPU(s): 164-17,256-263 NUMA node10 CPU(s): 164-17,256-303 NUMA node13 CPU(s): 124-151,336-343 NUMA node14 CPU(s): 104-111,296-303 NUMA node15 CPU(s): 122-159,344-351 NUMA node18 CPU(s): 122-159,344-351 NUMA node20 CPU(s): 112-119,304-311 NUMA node20 CPU(s): 126-167,352-359 NUMA node20 CPU(s): 184-191,376-383 Vulnerability Ridtown: Not affected Vulnerability Midtown: Not affected Vulnerability Midtown: Not affected Vuln		
NUMA node3 CPU(s): 72-79,264-271 NUMA node5 CPU(s): 8-15,200-207 NUMA node5 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 56-63,248-255 NUMA node7 CPU(s): 80-87,272-279 NUMA node10 CPU(s): 16-23,208-215 NUMA node10 CPU(s): 64-71,225-239 NUMA node11 CPU(s): 64-71,256-263 NUMA node12 CPU(s): 96-103,288-295 NUMA node13 CPU(s): 120-127,312-319 NUMA node15 CPU(s): 168-175,360-367 NUMA node16 CPU(s): 164-175,360-367 NUMA node17 CPU(s): 122-139,344-351 NUMA node18 CPU(s): 152-159,344-351 NUMA node19 CPU(s): 126-143,328-335 NUMA node20 CPU(s): 160-167,352-359 NUMA node20 CPU(s): 160-167,352-359 NUMA node23 CPU(s): 184-911,376-383 Vulnerability TLH5 Not affected Vulnerability Meltdown: Not affected Vulnerability Meltdown: Not affected Vulnerability Spectre V1: Mitigation; Speculative Store Bypass disabled via prc1 and seccomp Vulnerability Spectre V2:		
NUMA node4 CPU(s): 8-15,200-207 NUMA node5 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 66-63,248-255 NUMA node8 CPU(s): 60-87,272-279 NUMA node8 CPU(s): 61-23,208-215 NUMA node10 CPU(s): 64-71,252-239 NUMA node10 CPU(s): 88-95,280-287 NUMA node11 CPU(s): 88-95,280-287 NUMA node12 CPU(s): 120-127,312-319 NUMA node15 CPU(s): 120-127,312-319 NUMA node15 CPU(s): 1244-151,336-343 NUMA node15 CPU(s): 104-111,296-303 NUMA node17 CPU(s): 122-139,344-351 NUMA node20 CPU(s): 112-119,304-311 NUMA node20 CPU(s): 126-163,288-375 NUMA node20 CPU(s): 126-1143,328-335 NUMA node20 CPU(s): 126-119,304-311 NUMA node20 CPU(s): 136-143,328-335 NUMA node20 CPU(s): 160-167,352-359 NUMA node22 CPU(s): 166-167,352-359 NUMA node22 CPU(s): 164-191,376-383 Vulnerability Hthe multihit: Not affected Vulnerability Mds: Not affected Vulnerability Meltdown: Not affected		
NUMA node5 CPU(s): 32-39,224-231 NUMA node6 CPU(s): 56-63,248-255 NUMA node8 CPU(s): 16-23,208-215 NUMA node9 CPU(s): 40-47,232-239 NUMA node10 CPU(s): 64-71,256-263 NUMA node12 CPU(s): 96-103,288-295 NUMA node14 CPU(s): 120-127,312-319 NUMA node15 CPU(s): 168-175,360-367 NUMA node16 CPU(s): 128-135,320-327 NUMA node17 CPU(s): 128-135,320-327 NUMA node18 CPU(s): 128-135,320-327 NUMA node19 CPU(s): 160-167,352-359 NUMA node20 CPU(s): 160-167,352-359 NUMA node21 CPU(s): 184-191,376-383 Vulnerability ILft Not affected Vulnerability Maio stale data: Not affected Vulnerability Metbed: Not affected Vulnerability Spectre v1: Mitigation; speculative Store Bypass disabled via prct1 and seccomp Vulnerability Spectre v2: Mitidgation; ketpolines, IBPB conditional, TBRS_FW, STIBP		
NUMA node6 CPU(s): 56-63,248-255 NUMA node7 CPU(s): 80-87,272-279 NUMA node8 CPU(s): 16-23,208-215 NUMA node10 CPU(s): 40-47,232-239 NUMA node10 CPU(s): 88-95,280-287 NUMA node13 CPU(s): 120-127,312-319 NUMA node14 CPU(s): 120-127,312-319 NUMA node15 CPU(s): 120-127,312-319 NUMA node16 CPU(s): 164-11,296-303 NUMA node16 CPU(s): 104-111,296-303 NUMA node17 CPU(s): 128-135,320-327 NUMA node18 CPU(s): 152-159,344-351 NUMA node20 CPU(s): 136-143,328-335 NUMA node21 CPU(s): 136-143,328-335 NUMA node22 CPU(s): 160-167,352-359 NUMA node23 CPU(s): 184-191,376-383 Vulnerability Mds: Not affected Vulnerability Mds: Not affected Vulnerability Metldown: Not affected Vulnerability Metldewn: Not affected Vulnerability Spectre v1: Mitigation; Speculative Store Bypass disabled via prct1 and seccomp Vulnerability Spectre v2: Mitigation; Retpolines, IBPB conditional, TBRS_FW, STIBP always-on, RSB </td <td></td> <td></td>		
NUMA node7 CPU(s): 80-87,272-279 NUMA node8 CPU(s): 16-23,208-215 NUMA node9 CPU(s): 64-71,256-263 NUMA node11 CPU(s): 88-95,280-287 NUMA node12 CPU(s): 96-103,288-295 NUMA node14 CPU(s): 120-127,312-319 NUMA node15 CPU(s): 120-127,312-319 NUMA node14 CPU(s): 168-175,360-367 NUMA node16 CPU(s): 104-111,296-303 NUMA node17 CPU(s): 128-135,320-327 NUMA node20 CPU(s): 176-183,368-375 NUMA node20 CPU(s): 112-119,304-311 NUMA node22 CPU(s): 166-167,352-359 NUMA node22 CPU(s): 184-191,376-383 Vulnerability Ttlb multihit: Not affected Vulnerability Mds: Not affected Vulnerability Mds: Not affected Vulnerability Retbleed: Not affected Vulnerability Spectre v1: Mitigation; Speculative Store Bypass disabled via prct1 and seccomp Vulnerability Spectre v2: Mitigation; Retpolines, IBPE conditional, TBRS_FW, STIBP always-on, RSB filling, PBRSB-eIBRS Not affected Vulnerability Stods:		
NUMA node8 CPU(s): 16-23,208-215 NUMA node10 CPU(s): 40-47,232-239 NUMA node11 CPU(s): 64-71,256-263 NUMA node12 CPU(s): 88-95,280-287 NUMA node13 CPU(s): 120-127,312-319 NUMA node15 CPU(s): 120-127,312-319 NUMA node16 CPU(s): 144-151,336-343 NUMA node16 CPU(s): 168-175,360-367 NUMA node16 CPU(s): 104-111,296-303 NUMA node17 CPU(s): 128-135,320-327 NUMA node19 CPU(s): 152-159,344-351 NUMA node20 CPU(s): 176-183,386-375 NUMA node20 CPU(s): 136-143,328-335 NUMA node22 CPU(s): 160-167,352-359 NUMA node23 CPU(s): 164-191,376-383 Vulnerability Itlb multihit: Not affected Vulnerability Mds: Not affected Vulnerability Mdst Not affected Vulnerability Retbleed: Not affected Vulnerability Spectre v1: Mitigation; Speculative Store Bypass disabled via prc1 and seccomp Vulnerability Spectre v2: Mitigation; Retpolines, IBPB conditional, TBRS_FW, STIBP always-on, RSB filling, PBRSB-eIBRS Not affected Vulnerability Stbds:		
NUMA node9 CPU(s):40-47,232-239NUMA node10 CPU(s):64-71,256-263NUMA node11 CPU(s):88-95,280-287NUMA node12 CPU(s):96-103,288-295NUMA node13 CPU(s):120-127,312-319NUMA node14 CPU(s):144-151,336-343NUMA node15 CPU(s):168-175,360-367NUMA node16 CPU(s):104-111,296-303NUMA node18 CPU(s):122-159,344-351NUMA node19 CPU(s):152-159,344-351NUMA node20 CPU(s):176-183,368-375NUMA node21 CPU(s):160-167,352-359NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability IthtNot affectedVulnerability Mds:Not affectedVulnerability Retbleed:Not affectedVulnerability Spectre v1:Mitigation; Speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v2:Mitigation; Retpolines, IBPE conditional, TBRS_FW, STIBP always-on, RSBfilling, PBRSB-EIBRS Not affectedVulnerability Srbds:Vulnerability Srbds:Not affected		
NUMA nodel0 CPU(s):64-71,256-263NUMA nodel1 CPU(s):88-95,280-287NUMA nodel2 CPU(s):96-103,288-295NUMA nodel3 CPU(s):120-127,312-319NUMA nodel5 CPU(s):144-151,336-343NUMA nodel5 CPU(s):104-111,296-303NUMA nodel6 CPU(s):104-111,296-303NUMA nodel7 CPU(s):128-135,320-327NUMA nodel8 CPU(s):152-159,344-351NUMA nodel8 CPU(s):176-183,368-375NUMA node20 CPU(s):112-119,304-311NUMA node21 CPU(s):160-167,352-359NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability L1tf:Not affectedVulnerability Mds:Not affectedVulnerability Mds:Not affectedVulnerability Spectre v1:Mitigation; Speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v1:Mitigation; Netpolines, IBPB conditional, TBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedVulnerability Srbds:Vulnerability Srbds:Not affected		
NUMA nodel1 CPU(s):88-95,280-287NUMA nodel2 CPU(s):96-103,288-295NUMA nodel3 CPU(s):120-127,312-319NUMA nodel4 CPU(s):144-151,336-343NUMA nodel5 CPU(s):168-175,360-367NUMA nodel6 CPU(s):104-11,296-303NUMA nodel7 CPU(s):128-135,320-327NUMA nodel9 CPU(s):152-159,344-351NUMA node20 CPU(s):176-183,368-375NUMA node21 CPU(s):136-143,328-335NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):164-191,376-383Vulnerability L1tf:Not affectedVulnerability Mds:Not affectedVulnerability Meltdown:Not affectedVulnerability Spectre v1:Mitigation; Speculative Store Bypass disabled via prcl1 and seccompVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIEP always-on, RSBfilling, PBRSB-eIBRS Not affectedVulnerability Stds:Vulnerability Stds:Not affected		
NUMA nodel2 CPU(s):96-103,288-295NUMA nodel3 CPU(s):120-127,312-319NUMA nodel4 CPU(s):144-151,336-343NUMA nodel5 CPU(s):168-175,360-367NUMA nodel6 CPU(s):104-111,296-303NUMA nodel7 CPU(s):128-135,320-327NUMA nodel9 CPU(s):152-159,344-351NUMA node20 CPU(s):176-183,368-375NUMA node21 CPU(s):136-143,328-335NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability It1b multihit:Not affectedVulnerability Mds:Not affectedVulnerability Meltdown:Not affectedVulnerability Spectre v1:Not affectedVulnerability Spectre v1:Mitigation; speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedVulnerability Srbds:Vulnerability Srbds:Not affected		
NUMA nodel3 CPU(s):120-127,312-319NUMA nodel4 CPU(s):144-151,336-343NUMA nodel5 CPU(s):168-175,360-367NUMA nodel6 CPU(s):104-111,296-303NUMA nodel7 CPU(s):128-135,320-327NUMA nodel9 CPU(s):152-159,344-351NUMA node20 CPU(s):176-183,368-375NUMA node20 CPU(s):112-119,304-311NUMA node22 CPU(s):160-167,352-359NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability Itlb multihit:Not affectedVulnerability Mds:Not affectedVulnerability Meltdown:Not affectedVulnerability Spectre v1:Mitigation; Speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, TERS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedVulnerability Srbds:Not affectedVulnerability Srbds:Not affectedVulnerability Srbds:		
NUMA nodel4 CPU(s):144-151,336-343NUMA nodel5 CPU(s):168-175,360-367NUMA node16 CPU(s):104-111,296-303NUMA node17 CPU(s):128-135,320-327NUMA nodel9 CPU(s):152-159,344-351NUMA node20 CPU(s):176-183,368-375NUMA node21 CPU(s):112-119,304-311NUMA node22 CPU(s):160-167,352-359NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability Itlb multihit:Not affectedVulnerability Mds:Not affectedVulnerability Mds:Not affectedVulnerability Spectre v1:Mitigation; Speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedVul affectedVulnerability Srbds:Not affectedVulnerability Srbds:Not affected		
NUMA nodel5 CPU(s):168-175,360-367NUMA nodel6 CPU(s):104-111,296-303NUMA node17 CPU(s):128-135,320-327NUMA nodel8 CPU(s):152-159,344-351NUMA node20 CPU(s):176-183,368-375NUMA node20 CPU(s):136-143,328-335NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability Lltf:Not affectedVulnerability Mds:Not affectedVulnerability Meltdown:Not affectedVulnerability Spectre v1:Mitigation; Speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v1:Mitigation; Retpolines, IBPB conditional, TBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedVulnerability Srbds:Vulnerability Srbds:Not affectedVulnerability StateNot affectedVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, TBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedNot affectedVulnerability Srbds:Not affected		
NUMA nodel6 CPU(s):104-111,296-303NUMA nodel7 CPU(s):128-135,320-327NUMA node18 CPU(s):152-159,344-351NUMA node19 CPU(s):1176-183,368-375NUMA node20 CPU(s):112-119,304-311NUMA node21 CPU(s):136-143,328-335NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability Itlb multihit:Not affectedVulnerability Mds:Not affectedVulnerability Meltdown:Not affectedVulnerability Retbleed:Not affectedVulnerability Spectre v1:Mitigation; Speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, TBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedVulnerability Srbds:Vulnerability Srbds:Not affected		
NUMA node17 CPU(s):128-135,320-327NUMA node18 CPU(s):152-159,344-351NUMA node19 CPU(s):176-183,368-375NUMA node20 CPU(s):112-119,304-311NUMA node21 CPU(s):136-143,328-335NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability Itlb multihit:Not affectedVulnerability Mds:Not affectedVulnerability Mds:Not affectedVulnerability Retbleed:Not affectedVulnerability Spectre v1:Mitigation; Speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedNot affectedVulnerability Srbds:Not affected		
NUMA nodel8 CPU(s):152-159,344-351NUMA node19 CPU(s):176-183,368-375NUMA node20 CPU(s):112-119,304-311NUMA node21 CPU(s):136-143,328-335NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability Itlb multihit:Not affectedVulnerability Mds:Not affectedVulnerability Mds:Not affectedVulnerability Metdown:Not affectedVulnerability Retbleed:Not affectedVulnerability Spec store bypass:Mitigation; Speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v1:Mitigation; usercopy/swapgs barriers and user pointer sanitizationVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, TBRS_FW, STIBP always-on, RSBfilling, PBRSB-EIBRS Not affectedNot affectedVulnerability Srbds:Not affected		
NUMA node19 CPU(s):176-183,368-375NUMA node20 CPU(s):112-119,304-311NUMA node21 CPU(s):136-143,328-335NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability Itlb multihit:Not affectedVulnerability Mds:Not affectedVulnerability Meltdown:Not affectedVulnerability Retbleed:Not affectedVulnerability Spec store bypass:Mitigation; Speculative Store Bypass disabled via prct1 and seccompVulnerability Spectre v1:Mitigation; Retpolines, IBPB conditional, TBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedNot affectedVulnerability Srbds:Not affected		
NUMA node20 CPU(s):112-119,304-311NUMA node21 CPU(s):136-143,328-335NUMA node22 CPU(s):160-167,352-359NUMA node23 CPU(s):184-191,376-383Vulnerability Itlb multihit:Not affectedVulnerability Meltdown:Not affectedVulnerability Meltdown:Not affectedVulnerability Retbleed:Not affectedVulnerability Spec store bypass:Mitigation; Speculative Store Bypass disabled via prctl and seccompVulnerability Spectre v1:Mitigation; usercopy/swapgs barriers anduser pointer sanitizationVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, TBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedNot affectedVulnerability Srbds:Not affected		
<pre>NUMA node21 CPU(s): 136-143,328-335 NUMA node22 CPU(s): 160-167,352-359 NUMA node23 CPU(s): 184-191,376-383 Vulnerability Itlb multihit: Not affected Vulnerability L1tf: Not affected Vulnerability Meltdown: Not affected Vulnerability Meltdown: Not affected Vulnerability Retbleed: Not affected Vulnerability Retbleed: Not affected Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp Vulnerability Spectre v1: Mitigation; usercopy/swapgs barriers anduser pointer sanitization Vulnerability Spectre v2: Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSB filling, PBRSB-eIBRS Not affected Vulnerability Srbds: Not affected</pre>		
<pre>NUMA node22 CPU(s): 160-167,352-359 NUMA node23 CPU(s): 184-191,376-383 Vulnerability Itlb multihit: Not affected Vulnerability L1ff: Not affected Vulnerability Mds: Not affected Vulnerability Mmio stale data: Not affected Vulnerability Retbleed: Not affected Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp Vulnerability Spectre v1: Mitigation; Speculative Store Bypass disabled via prctl and seccomp Vulnerability Spectre v2: Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSB filling, PBRSB-eIBRS Not affected Vulnerability Srbds: Not affected</pre>		
<pre>NUMA node23 CPU(s): 184-191,376-383 Vulnerability Itlb multihit: Not affected Vulnerability Itlf: Not affected Vulnerability Mds: Not affected Vulnerability Meltdown: Not affected Vulnerability Meltdown: Not affected Vulnerability Retbleed: Not affected Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp Vulnerability Spectre v1: Mitigation; usercopy/swapgs barriers and user pointer sanitization Vulnerability Spectre v2: Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSB filling, PBRSB-eIBRS Not affected Vulnerability Srbds: Not affected</pre>		
<pre>Vulnerability Itlb multihit: Not affected Vulnerability L1tf: Not affected Vulnerability Mds: Not affected Vulnerability Meltdown: Not affected Vulnerability Mmio stale data: Not affected Vulnerability Retbleed: Not affected Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp Vulnerability Spectre v1: Mitigation; usercopy/swapgs barriers anduser pointer sanitization Vulnerability Spectre v2: Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSB filling, PBRSB-eIBRS Not affected Vulnerability Srbds: Not affected</pre>		
Vulnerability L1tf:Not affectedVulnerability Mds:Not affectedVulnerability Meltdown:Not affectedVulnerability Mmio stale data:Not affectedVulnerability Retbleed:Not affectedVulnerability Spec store bypass:Mitigation; Speculative Store Bypass disabled via prctl and seccompVulnerability Spectre v1:Mitigation; usercopy/swapgs barriers anduser pointer sanitizationVulnerability Spectre v2:Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSBfilling, PBRSB-eIBRS Not affectedNot affectedVulnerability Srbds:Not affected	Vulnerability Itlb multihit:	Not affected
VulnerabilityMeltdown:Not affectedVulnerabilityMmio stale data:Not affectedVulnerabilityRetbleed:Not affectedVulnerabilitySpec store bypass:Mitigation;VulnerabilitySpectre v1:Mitigation;VulnerabilitySpectre v2:Mitigation;VulnerabilitySpectre v2:Mitigation;Filling,PBRSB-eIBRSNot affectedVulnerabilitySrbds:Not affected		Not affected
Vulnerability Mmio stale data: Not affected Vulnerability Retbleed: Not affected Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp Vulnerability Spectre v1: Mitigation; usercopy/swapgs barriers anduser pointer sanitization Vulnerability Spectre v2: Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSB filling, PBRSB-eIBRS Not affected Vulnerability Srbds: Not affected		Not affected
VulnerabilityRetbleed:Not affectedVulnerabilitySpec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccompVulnerabilitySpectre v1:Mitigation; usercopy/swapgs barriers and user pointer sanitizationVulnerabilitySpectre v2:Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSBfilling,PBRSB-eIBRS Not affectedVulnerabilitySrbds:Not affected		Not affected
<pre>Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp Vulnerability Spectre v1: Mitigation; usercopy/swapgs barriers anduser pointer sanitization Vulnerability Spectre v2: Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSB filling, PBRSB-eIBRS Not affected Vulnerability Srbds: Not affected</pre>	Vulnerability Mmio stale data:	Not affected
VulnerabilitySpectre v1:Mitigation; usercopy/swapgs barriers and user pointer sanitizationVulnerabilitySpectre v2:Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSBfilling,PBRSB-eIBRS Not affectedVulnerabilitySrbds:Not affected	Vulnerability Retbleed:	Not affected
Vulnerability Spectre v2: Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSB filling, PBRSB-eIBRS Not affected Vulnerability Srbds: Not affected	Vulnerability Spec store bypass:	
filling, PBRSB-eIBRS Not affected Vulnerability Srbds: Not affected		Mitigation; usercopy/swapgs barriers anduser pointer sanitization
Vulnerability Srbds: Not affected		
Vulnerability Tsx async abort: Not affected		
	vulnerability Tsx async abort:	Not allected

The preceding shows the physical core numbers followed by the logical core numbers. For instance, the highlighted row for NUMA node 2 shows eight physical cores 48–55 and logical cores 240–247. In this example, eight total interrupt queues is the best setting. If then NIC combines the RX and TX interrupts, then execute the following command:

root@testsystem:~# ethtool -L enp33s0f0 combined 8

Evaluate the throughput vs. latency trade-off for dual-socket systems powered by AMD EPYC 9005 Series Processors based on your production requirements. Splitting the interrupts across NUMA-nodes will maximize throughput, while limiting the interrupts to a single node will reduce latency.

4.4.2 - RX and TX Ring Sizes

The NIC ring values represent the number of buffers that a NIC uses to DMA data into system memory. Having more buffers available queues more work for processing during a single interrupt. The ethtool utility allows you to check the current and maximum NIC ring sizes:

```
root@testsystem:~ethtool -g enp33s0f0
Ring parameters for enp33s0f0:
Pre-set maximums:
RX:
              8192
RX Mini:
              0
RX Jumbo:
              0
TX:
              8192
Current hardware settings:
RX:
              512
              0
RX Mini:
RX Jumbo:
              0
TX:
              512
```

The silicon and driver in the adapter shown in this example allow a maximum ring size of 8192, but it is currently set to 512, or only 1/16th of the maximum allowed. Use ethtool to boost network performance by setting the maximum allowable ring size:

root@testsystem:~# ethtool -G enp33s0f0 rx 8192 tx 8192

Older kernels or drivers do not support byte queue limits, and bigger values might bloat the transmit (TX) interrupt queue buffers. AMD recommends using the latest kernel and driver versions for optimal performance.

Enabling interrupt coalescing will avoid interrupt throttling, and enabling it may add latency. If you need higher throughput vs. latency, then enable interrupt coalescing. Keep the default setting for latency-sensitive applications.

4.4.3 - Preferred I/O and PCIe Relaxed Ordering

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

4.4.4 - IOMMU Settings

The Linux kernel is constantly being updated. You can obtain official mainline releases that typically include the latest features from The Linux Foundation at http://kernel.org*. Common enterprise-grade Linux distributions rarely use the latest mainline kernel.

```
# vi /etc/default/grub
add "iommu=pt" to the GRUB_CMDLINE_LINUX_DEFAULT line
GRUB_CMDLINE_LINUX_DEFAULT="splash=silent resume=/dev/disk/by-path/pci- 0000:03:00.0-scsi-0:0:0:0-part4
mitigations=auto iommu=pt quiet"
# grub2-mkconfig -o /boot/grub2/grub.cfg # reboot (required to take effect)
```

AMD testing observed a scaling bottleneck when deploying NGINX instances with more than 32 vCPUs within a single host OS Ubuntu 18.04 image with kernel 4.20 and wrk requesting static 1 KB content. Updating to kernel 5.5 that includes IOMMU-optimized patches allowed NGINX instances up to 256 vCPUs to exhibit expected scaling performance. You may also try deploying a guest OS image for each scale out 32 vCPUs NGINX instance. AMD strongly recommends using Linux kernel 5.20 or newer with IOMMU optimized patches.

4.5 - System Configurations

4.5.1 - Simultaneous Multi-Threading (SMT)

Enabling SMT depends on your application. AMD testing saw improved performance with SMT=ON for static-content requests of ~1 KB when using NGINX as a web server. AMD recommends enabling SMT in BIOS.

4.5.2 - Max File Open

NGINX allows users between 1024 (soft limit) and 4096 (hard limit) open files by default. You can increase this to 65535 for a NGINX worker. Make this change in nginx.service by adding the LimitNOFILE option with a value.

```
#nproc - number of processes
#nofile - number of file descriptors
      soft
               nproc
                             32768
                             65535
      hard
                 nproc
                             32768
      soft
                 nofile
                             65535
      hard
                 nofile
root
      soft
                 nproc
                             32768
     hard
                             65535
root
                 nproc
      soft
                 nofile
                             32768
root
                             65535
                 nofile
root
      hard
```

4.5.3 - Configure Limits

/etc/security/limits.conf contains the system limits for various resources. Setting these values is important to make sysctl .conf
parameters work for the specified values.

4.5.4 - Sysctl Configuration

Sysctl modifies kernel parameters at runtime which helps boost performance at the OS level. <u>"Sample /etc/sysctl1.conf File" on page xxiii</u> has a sample /etc/sysctl.conf that was tested to scale up to 64 vCPUs in VM environments and up to 384 vCPUs in bare metal environments.

4.6 - NGINX PARAMETERS

NGINX documentation inclOudes many parameters that can significantly impact throughput, response time, and system resource utilization. AMD validated most of the recommendations in this section for NGINX as a web server for static and dynamic content using wrk as a load generator. Be sure to verify the impact of these parameters in your production environment.

The easiest way to do this is to create test scripts that are optimized for performance so as to obtain stable and predictable results and separate test scripts that match production requirements. For example, the keepalive tuning, disable access log, and other settings are idea for test scripts. They will not provide absolute performance figures that apply directly to a production environment, but they indicate whether the setting is likely to have a significant impact.

4.6.1 - Application Parameters Matching sysctl

Configuring all OS-level tunings, and then make sure that the application is aware of resource availability by enabling application-level parameters such as worker_rlimit_nofile which is equal to the number of file descriptors, open_file_cache, open_file_cache_valid, open_file_cache_min_uses, open_file_cache_errors and keepalive_requests.

4.6.2 - Worker_processes

It is important to adjust worker_processes based on the use case when assigning more vCPUs to a NGINX instance. It is common practice to run 1 worker process per vCPU. There are times when you may want to increase this number, such as when the worker processes must do a lot of disk I/O.

worker_processes number | auto; Default: worker_processes 1;

4.6.3 - Worker_connections

Each worker process can handle 512 simultaneous processes by default, but most systems have enough resources to support a larger number. The appropriate setting depends on the use case, content size, etc., and should be evaluated via testing.

worker connections number; Default: worker connections 512;

4.6.4 - Access Logging

NGINX writes client request information in the access log right after processing the request. Enabling default access logging causes very poor scaling beyond 4 VCPUs NGINX instance because multiple threads are attempt to write to the same log file, which results in <code>osq_lock()</code> dominating the profile. You can also try enabling conditional access logging to reduce the locking issue. AMD recommends disabling access logging to evaluate the best possible performance. Enable it later to test with a logging infrastructure that can support both scale up and scale out NGINX instances.

```
access_log path [format [buffer=size] [gzip[=level]] [flush=time] [if=condition]];
access_log off;
```

Default: access log logs/access.log combined;

4.6.5 - Sendfile and tcp_nopush

Enabling the sendfile() directive eliminates the step of copying the data into the buffer and enables direct copying data from one file descriptor to another. Enabling sendfile() improves performance for request with large content size (>1MB) and results in a small performance loss for requests with small content sizes. AMD recommends setting sendfile_max_chunk to equal to the request content size. If requests could be a mix of various content sizes, then set it to the typical average request size.

Use the tcp_nopush directive together with the sendfile on; directive to enable NGINX to send HTTP response headers in one packet right after sendfile () obtains the chunk of data.

```
sendfile on | off; Default: sendfile off;
sendfile_max_chunk size; Default: sendfile_max_chunk 0;
tcp_nopush on | off; Default: tcp_nopush off;
```

4.6.6 - Keepalive

NGINX enables keepalive by default. Keepalive connections can have a major impact on performance by reducing the CPU and network overhead needed to open and close connections. Production environment needs may restrict keeping connections alive. AMD recommends exploring solutions that can enhance security and performance.

```
keepalive_requests number; Default: keepalive_requests 100;
keepalive_timeout timeout [header_timeout]; Default: keepalive_timeout 75s;
keepalive_disable none | browser ...; Default: keepalive_disable msie6;
```

4.6.7 - HTTP vs. HTTPS

NGINX web server setups evaluating the impact of HTTP vs. HTTPS traffic found minimal performance and latency impact when comparing HTTP and HTTPS requests being sent by the wrk tool.

4.6.8 - Caching and Compression

Enabling caching on NGINX web server will boost performance. Most of the time, serving stale content while re-validating in the background will enhance user performance and engagement. Set the caching limit and duration by analyzing the load on the server. Make sure that there is sufficient memory available to store the hot cached content in the OS page cache when caching is enabled.

NGINX supports Gzip compression. Compression decreases file sizes and accelerates the transfer rate from the server to the client, which can boost application performance.

4.7 - NGINX Build with the Latest GCC Compiler Using Znver5 Optimizations

AMD used znver5 to enable "Zen 5" support in GNU Compiler Collection (GCC) versions 13 and later.

4.8 - ORCHESTRATION AND CONTAINER SETTINGS

Production NGINX deployments often use orchestration on all types of infrastructure such as bare- metal, containers, VMs, and public, private clouds, and hybrid clouds. Orchestration frameworks such as Kubernetes are most prevalent for automating deployment, scaling, and managing containerized applications.

AMD used a test setup created using Kubernetes for orchestration and Docker NGINX containers. NGINX web server often has suboptimal performance settings related to container overlay, networking, file systems, and health monitoring. Here are some references for overlay network configurations and performance comparison among the popular ones in use:

- <u>Cluster Networking</u>*
- <u>Networking Overview</u>*

Here is a sample deployment file for NGINX on K8S with no CNI:

```
kind: DaemonSet
metadata:
 name: web-server-payload
 namespace: nginx-ingress
spec:
 selector:
   matchLabels:
     app: web-server-payload
 template:
   metadata:
      labels:
       app: web-server-payload
spec:
 hostNetwork: true
 containers:
    - name: web-server-payload
    image: mynginx:latest
    imagePullPolicy: IfNotPresent
    ports:
      - containerPort: 80
     _
         containerPort: 443
 nodeSelector:
   name: webserver
```

Note: Developers and administrators should be aware there is a known issue with NGINX ignoring the cgroup resource requests and limits created by Kubernetes and will use all CPUs on the host by default. A possible workaround is to set the number of worker processes parameter to the desired number of CPUs in the NGINIX configuration file.

Pinning containerd to certain cores can reduce overhead. Docker, K8S, and other container platforms use containerd to abstract away syscalls or OS-specific functionality to run containers on Linux, Windows, and other operating systems. For example:

\$ sudo vim /usr/lib/systemd/system/containerd.service

[Service]

CPUAffinity=24

THIS PAGE INTENTIONALLY LEFT BLANK.



CHAPTER 5: REFERENCES AND SETUP

This tuning guide focuses on NGINX-related optimizations and tunings, but a complete system requires tuning all modules. See the <u>AMD</u> <u>Documentation Hub</u> for the complete list of available tuning guides, and be sure to use the latest version applicable to your AMD EPYC processor model. Some specific tuning guides you may want to look at (available from the <u>AMD Documentation Hub</u>) include:

- Linux[®] Network Tuning Guide for AMD EPYC[™] 9005 Series Processor Based Servers
- Microsoft[®] Windows[®] Network Tuning Guide for AMD EPYC[™] 9005 Series Processor Based Servers
- BIOS & Workload Tuning Guide for AMD EPYC™ 9005 Series Processor Based Servers

5.1 - NGINX SETUP FOR CLOUD VMs

See the following resources for cloud VM instances based on AMD EPYC 9005 Series Processors with Ubuntu 20.04:

- wrk: <u>https://github.com/wg/wrk</u>*
- NGINX: <u>http://nginx.org/en/download.html</u>*

5.2 - NGINX SETUP FOR BARE METAL

See the following resources for bare metal VM instances:

- wrk: <u>https://github.com/wg/wrk</u>*
- NGINX: http://nginx.org/en/download.html
- **OS:** Ubuntu 20.04 with kernel 5.13

5.3 - NGINX COMPILATION WITH GCC

See the following resources for NGINX compiled using the latest GCC 12:

- NGINX: <u>http://nginx.org/en/download.html</u>*
- Compiler: Latest GCC 12 to build NGINX 1.22 version with Znver4, O3 optimization flags

5.4 - ORCHESTRATION AND CONTAINER SETUP

See the following resources for NGINX using Kubernetes to orchestrate Docker containers:

- K8S env: OS: 5.3.0-64-generic #58-Ubuntu SMP Fri Jul 10 19:33:51 UTC 2020 x86_64 kubeadm version: &version.Info{Major:"1", Minor:"19", GitVersion:"v1.19.0"
- Docker version: 19.03.12, build 48a66213fe
- Official NGINX container image pulled from Docker Hub: 1.19.2

XXII



CHAPTER 6: SAMPLE /ETC/SYSCTL1.CONF FILE

This example presents a sample /etc/sysctl.conf that scales well up to 64 vCPUs in VM environments and up to 256 vCPUs in bare metal environments:

```
# /etc/sysctl.conf
######### Kernel#####
                      #########
 Controls the System Request debugging functionality of the kernel kernel.sysrq = 0
# Controls whether core dumps will append the PID to the core #filename.
Useful for debugging multi-threaded applications kernel.core_uses_pid = 1
# increase system file descriptor limit fs.file-max = 65535
#Allow for more PIDs kernel.pid max = 65536
#Enable ExecShield protection kernel.exec-shield = 1
kernel.randomize va space = 1
kernel.printk = 4 4 1 7
kernel.panic = 10
kernel.shmmax = 4294967296
kernel.shmall = 4194304
kernel.msgmnb = 65536
kernel.msgmax = 65536
vm.swappiness = 10
vm.vfs_cache_pressure = 50
vm.dirty ratio = 80
########## IPv4 networking ######
# Controls IP packet forwarding net.ipv4.ip forward = 0
# Controls source route verification net.ipv4.conf.default.rp filter = 1
# Do not accept source routing net.ipv4.conf.default.accept source route = 0
# Send redirects, if router, but this is just server net.ipv4.conf.all.send redirects = 0
net.ipv4.conf.default.send_redirects = 0
# Accept packets with SRR option? No net.ipv4.conf.all.accept source route = 0
# Accept Redirects? No, this is not router net.ipv4.conf.all.accept redirects = 0
net.ipv4.conf.all.secure_redirects = 0
# Log packets with impossible addresses to kernel log? yes net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log martians = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure redirects = 0
# Ignore all ICMP ECHO and TIMESTAMP requests sent to it via broadcast/multicast
net.ipv4.icmp_echo_ignore_broadcasts = 1
# Turn on protection for bad icmp error messages net.ipv4.icmp ignore bogus error responses = 1
```

Prevent against the common 'syn flood attack' net.ipv4.tcp syncookies = 1 # Controls the use of TCP syncookies net.ipv4.tcp synack retries = 2 # Enable source validation by reversed path, as specified in RFC1812 net.ipv4.conf.all.rp filter = 1 net.ipv4.conf.default.rp filter = 1 # TCP and memory optimization # increase TCP max buffer size setable using setsockopt() net.ipv4.tcp rmem = 4096 87380 8388608 net.ipv4.tcp_wmem = 4096 87380 8388608 # increase Linux auto tuning TCP buffer limits net.core.rmem max = 8388608 net.core.wmem max = 8388608 net.core.netdev max backlog = 5000 net.ipv4.tcp window scaling = 1 #Increase system IP port limits net.ipv4.ip local port range = 2000 65000 net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 0 net.ipv4.tcp_fin_timeout = 60 net.ipv4.tcp_slow_start after idle = 0 net.ipv4.tcp_mtu probing = 1 # TCP Fast Open net.ipv4.tcp fastopen = 3 net.ipv4.tcp congestion control = htcp Number of Router Solicitations to send until assuming no routers are present. # This is host and not router net.ipv6.conf.default.router solicitations = 0 # Accept Router Preference in RA? net.ipv6.conf.default.accept ra rtr pref = 0 # Learn Prefix Information in Router Advertisement net.ipv6.conf.default.accept ra pinfo = 0 # Setting controls whether the system will accept Hop Limit settings # from a router advertisement net.ipv6.conf.default.accept_ra_defrtr = 0 #router advertisements can cause the system to assign a global unicast #address to an interface net.ipv6.conf.default.autoconf = 0 #how many neighbor solicitations to send out per address? net.ipv6.conf.default.dad transmits = 0

How many global unicast IPv6 addresses can be assigned to each interface? net.ipv6.conf.default.max_addresses = 1



CHAPTER 7: RESOURCES

- NGINX on AMD powered Amazon C5a instances
- NGINX on AWS powered Amazon M5a instances
- NGINX on AMD powered Azure Eav4 instances
- NGINX on AMD powered Google Cloud N2D confidential VMs
- <u>NGINX on AMD powered GCP N2D standard VMs</u>

NGINX[®] Tuning Guide for AMD EPYC[™] 9005 Processors

PID: 58489



