

TUNING GUIDE AMD EPYC 7003

Red Hat Enterprise Linux®

Publication	57088
Revision	3.0
Issue Date	March, 2022



© 2022 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, Infinity Guard, 3D V-Cache, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Linux is a registered trademark of Linus Torvalds. 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
Mar, 2021	2.0	Initial public release
Mar, 2022	3.0	Added AMD EPYC 3D V-Cache™ information

Audience

This document is intended for a technical audience such as Linux® 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.

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

Authors

Sandeep Gupta and Sylvester Rajasekaran

Table of Contents

Chapter 1	AMD EPYC™ 7003 Series Processors	1
1.1	AMD EPYC™ 7003 Series Processors	1
1.2	Operating Systems	1
1.3	Networking Support	2
Chapter 2	Common Linux Kernel Tools and Examples	3
2.1	lscpu	4
2.2	numactl	5
2.2.1	lscpu and numactl for AMD EPYC 7373X	6
2.3	lstopo	7
2.4	Portable Hardware Locality (hwloc) and hwloc-gui	8
2.4.1	hwloc-ls	8
2.4.1.1	hwloc-info	10
2.5	lstopo, hwloc-ls, hwloc-info, for AMD EPYC 7373X	11
2.5.1	lstopo	11
2.5.2	hwloc-info	12
2.5.3	hwloc-ls	12
2.6	cpupower	14
2.6.1	cpupower monitor	15
2.7	cpupower on AMD EPYC 7373X	15
2.7.1	3D V-Cache Disabled	15
2.7.2	3D V-Cache Enabled	17
2.8	CPU Governors	19
2.9	top	20
2.9.1	Example 1: Per-CPU Utilization Statistics	20
2.9.2	Example 2: Per-NUMA-Node Utilization Statistics	20
2.9.3	Example 3: Utilization Statistics Summary	21
2.10	tuned	21
2.11	tuna	21
Chapter 3	General Tuning Recommendations	23
3.1	LLC as NUMA Domain	23
3.2	AMD uProf	24
3.3	perf	24
3.3.1	perf list cpu	25
3.3.2	perf list cache	25
3.3.3	perf stat	26

Chapter 4	Virtualization	29
4.1	Secure Encrypted Virtualization (SEV)	29
4.1.1	SEV Prerequisites	30
4.2	AMD EPYC Virtualization Support	30
4.3	Resource Allocation and Host/VM Tuning	31
4.4	Tuning the Virtualization Host	31
4.5	Evaluating Workloads and VM Workloads	31
Chapter 5	Troubleshooting and Debugging Notes	33
5.1	Error Detection and Correction (EDAC)	33
5.1.1	Get the Memory Controller MCx Device Information	34
5.2	Error Injection	35
Chapter 6	AMD 3D V-Cache	41
6.1	BIOS Settings	41
6.2	lscpu	43
6.2.1	3D V-cache Disabled	43
6.2.2	3D V-cache Enabled	43
6.2.3	L3 Cache	44
6.3	Ishw -C memory Output for L3	44
6.3.1	3D V-Cache Disabled	44
6.3.2	3D V-Cache Enabled	45
6.3.2.1	L3 Cache	45
6.4	Ishw -C memory Output for L3	45
6.4.1	3D V-Cache Disabled	45
6.4.2	3D V-Cache Enabled	46
6.5	Cache Information using valgrind	46
6.5.1	3D V-Cache Disabled	46
6.5.2	3D V-Cache Enabled	47
Chapter 7	SPECpower and STEAM	49
7.1	STREAM using Spack	49
Chapter 8	Resources	51

Chapter

1

AMD EPYC™ 7003 Series Processors

This tuning guide describes parameters that can optimize performance of servers powered by AMD EPYC™ 7003 Series Processors running Red Hat Enterprise Linux (RHEL) Operating Systems.

Chapter 2 describes some of the available Linux tuning tools. Please also see [Red Hat Enterprise Linux 8 - Monitoring and Managing System Performance](#)* for more information on these toolkits and RHEL 8-specific monitoring, system performance, and tuning settings.

This tuning guide uses examples based on RHEL 8.3 running Linux 4.18.0-240.11.el8_3.x86_64.

1.1 AMD EPYC™ 7003 Series Processors

3rd Gen AMD EPYC™ 7003 Series Processors are socket compatible with previous 2nd Gen EPYC 7002 Series processors and may be supported in existing EPYC CPU-based system, depending on the specific platform. 3rd Gen AMD EPYC processors complement AMD's existing server portfolio by offering further improvements to performance and value in a number of configurations with varying core counts, thermal design points, frequencies, and other features. See [Table 1-1](#).

Component	Detail
Socket	SP3
Max Number of Cores	64
Core Process Technology	7 nm
Max Memory Speed	3200 MT/s
Max Memory Channels	8 per socket
Max Memory Capacity	4TB per socket
I/O Interconnect	128 lanes (max) PCIe® Gen4

Table 1-1: General AMD EPYC 7003™ processor specification

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

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

1.2 Operating Systems

Please see [AMD EPYC™ 7003 Series Processors Minimum Operating System \(OS\) Versions](#) for detailed OS requirements.

1.3 Networking Support

AMD tested several adapters at multiple speeds from 1Gbps to 200Gbps, using the recommendations and from the following tables in the latest version of *Linux Network Tuning Guide for AMD EPYC 7003 Series Processors* (available from [AMD EPYC Tuning Guides](#)), respectively:

- Network Tuning Recommendations
- Network Testing Results

Common Linux Kernel Tools and Examples

The Red Hat Enterprise Linux kernel has had NUMA-aware default memory and CPU scheduling policy since RHEL 6, and this support is being enhanced across all current Version 8 releases. See [Red Hat Enterprise Linux 7: Optimizing Memory System Performance](#)* for detailed information about RedHat Enterprise Linux NUMA architecture, manual NUMA binding using `numactl`, automatic NUMA binding using `numad`, kernel automatic NUMA balancing, and more.

NUMA (Non Uniform Memory Access) systems consists of CPU clusters or CPU groups. Each CPU group is called a NUMA node, and each NUMA node has its own CPUs, memory, and I/O devices. NUMA nodes connect to memory and I/O devices on remote CPUs via one or more buses (or interconnects). The term NUMA comes from the fact that it is faster to access local memory than memory associated with other NUMA nodes.

NUMA architecture introduces memory access latencies depending on the distance between the CPU and the memory location. System BIOS populates the System Locality Information Table (SLIT) and supplies it to the Linux kernel via the Advanced Configuration and Power Interface (ACPI) and provides the normalized distances between the different NUMA nodes. See [Socket SP3 Platform NUMA Topology for AMD Family 19h Models00h-0Fh](#) for additional information.

Red Hat created the optional user-level `numad` daemon that provides process management and placement advice for efficient CPU and memory usage on systems with NUMA topology.

The NUMA tools in Red Hat Enterprise Linux improve application performance on modern hardware systems. Additional tuning can improve performance even more.

Automatic NUMA balancing provides satisfactory performance in most cases, and the default performance is near optimal. This chapter lists some commonly available Linux management tools and provides some examples. See:

- [“lscpu” on page 4](#)
- [“numactl” on page 5](#)
- [“lstopo” on page 7](#)
- [“Portable Hardware Locality \(hwloc\) and hwloc-gui” on page 8](#)
- [“cpupower” on page 14](#)
- [“top” on page 20](#)
- [“tuned” on page 21](#)
- [“tuna” on page 21](#)

2.1 lscpu

`lscpu` gives a quick view of CPU topology with the following information:

- Number of sockets, nodes, cores, and threads present in the system.
- Caches and their sizes.
- NUMA nodes and CPU associations.

For example:

```
# lscpu
Architecture:          x86_64
CPU op-mode(s):       32-bit, 64-bit
Byte Order:           Little Endian
CPU(s):               256
On-line CPU(s) list: 0-255
Thread(s) per core:   1
Core(s) per socket:   64
Socket(s):            2
NUMA node(s):         2
Vendor ID:            AuthenticAMD
CPU family:           25
Model:                1
Model name:           AMD EPYC 7713 64-Core Processor
Stepping:             1
CPU MHz:              2000.000
CPU max MHz:          2000.0000
CPU min MHz:          1500.0000
BogoMIPS:             3992.79
Virtualization:       AMD-V
L1d cache:            32K
L1i cache:            32K
L2 cache:             512K
L3 cache:             32768K
NUMA node0            CPU(s):      0-63,128-191
NUMA node1            CPU(s):      64-127,192-255

Flags:                fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc art
rep_good nopl xtopology nonstop_tsc extd_apicid aperfmperf eagerfpu pni pclmulqdq monitor
ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c rdrand lahf_lm
cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt
tce topoext perfctr_core perfctr_nb bpext perfctr_l2 cpb cat_l3 cdp_l3 hw_pstate sme
retpoline_amd ssbd ibrs ibpb stibp vmmcall fsgsbase bmi1 avx2 smep bmi2 erms invpcid cqm
rdt_a rdseed adx smap clflushopt clwb sha ni xsaveopt xsavec xgetbv1 cqm_llc cqm_occup_llc
cqm_mbm_total cqm_mbm_local clzero irperf xsaveerptr arat npt lbrv svm_lock nrip_save
tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold v_vmsave_vmload
vgif umip pku ospke vaes vpclmulqdq overflow_recov succor smca
```

2.2 numactl

`numactl` can control the NUMA policy for processes or shared memory. `numactl` and `numad` help tune NUMA scheduling parameters and monitor both NUMA topology and resource utilization by automatically making affinity adjustments to locally optimize processes. You can use `numactl` to find:

- Number of NUMA nodes in the system.
- Distance between nodes.
- CPU and memory association with the node.
- Linux policy (default, bind, preferred, interleave) of the current process.

The second part of the `numactl -hardware` output gives the node distances in a matrix based on the System Locality Information Table in the Advanced Configuration and Power Interface (ACPI SLIT). These distances indicate the cost of accessing remote memory as the relative latency to access memory from a particular node, normalized to a base value of 10. Higher values indicate more overhead.

For example:

```
# yum install numactl
#numactl - hardware
available: 2 nodes (0-1)

node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
55 56 57 58 59 60 61 62 63 128 129 130 131 132 133 134 135 136 137 138 139 140 141
142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183
184 185 186 187 188 189 190 191

node 0 size: 64146 MB
node 0 free: 59835 MB

node 1 cpus: 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87
88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 192 193 194 195 196
197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217
218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238
239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255

node 1 size: 64414 MB
node 1 free: 63471 MB

node distances:
node    0  1
  0:   10 32
  1:   32 10
```

2.2.1 lscpu and numactl for AMD EPYC 7373X

```
# lscpu
Architecture:           x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                64
On-line CPU(s) list:  0-63
Thread(s) per core:    2
Core(s) per socket:    16
Socket(s):              2
NUMA node(s):          2
Vendor ID:              AuthenticAMD
BIOS Vendor ID:        Advanced Micro Devices, Inc.
CPU family:             25
Model:                  1
Model name:             AMD EPYC 7373X 16-Core Processor
BIOS Model name:       AMD EPYC 7373X 16-Core Processor
Stepping:               2
CPU MHz:                3050.000
CPU max MHz:           3830.3711
CPU min MHz:           1500.0000
BogoMIPS:               6088.78
Virtualization:        AMD-V
L1d cache:              32K
L1i cache:              32K
L2 cache:               512K
L3 cache:               98304K
NUMA node0 CPU(s):     0-15,32-47
NUMA node1 CPU(s):     16-31,48-63
Flags:                  fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
                        clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc
                        rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclmulqdq monitor ssse3 fma cx16
                        pcid sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand lahf_lm cmp_legacy svm extapic
                        cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt tce topoext perfctr_core
                        perfctr_nb bpext perfctr_llc mwaitx cpb cat_l3 cdp_l3 invpcid_single hw_pstate ssbd mba
                        ibrs ibpb stibp vmmcall fsgsbase bmi1 avx2 smep bmi2 invpcid cqm rdt_a rdseed adx smap
                        clflushopt clwb sha_ni xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total
                        cqm_mbm_local clzero irperf xsaveerptr wbnoinvd amd_ppin arat npt lbrv svm_lock nrip_save
                        tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold v_vmsave_vmload
                        vgif v_spec_ctrl umip pku ospke vaes vpclmulqdq rdpid overflow_recov succor smca sme sev
                        sev_es
# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 32 33 34 35 36 37 38 39 40 41 42 43 44
45 46 47
node 0 size: 515589 MB
node 0 free: 512828 MB
node 1 cpus: 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 48 49 50 51 52 53 54 55 56 57
58 59 60 61 62 63
node 1 size: 451572 MB
node 1 free: 449407 MB
node distances:
node   0   1
  0:  10  32
  1:  32  10
```

2.3 Istopo

The screenshot shows the Istopo tool interface for a 2P AMD EPYC 7713 system. The main display area is a large grid of memory modules, each represented by a small icon and a unique ID. The grid is organized into two main sections, each labeled with a system ID (U132A08 and U132A09). The modules are arranged in a regular pattern, with each cell containing a small icon and a unique ID. The AMD EPYC logo is prominently displayed in the center of the grid. The top of the interface shows the system configuration, including the number of processors (2P) and the total memory (256 GB). The bottom of the interface shows the system name (NU MANCOTE PH I (128EB)) and the system ID (U132A08).

Figure 2-1: Istopo showing a 2P AMD EPYC 7713 system with 256 GB of memory

The graphical output consists of nested boxes representing the inclusion of objects in the resource hierarchies. A **Machine** box usually contains one or several **Package** boxes that contain multiple **Core** boxes that each have one or more **PUs**.

Caches appear in a slightly different manner because they do not actually include computing resources, such as cores. For instance, a L2 Cache shared by a pair of cores appears as a **Cache** box on top of two **Core** boxes (instead of having **Core** boxes inside the **Cache** box).

By default, NUMA nodes boxes appear on top of their local computing resources. For instance, a processor **Package** containing one NUMA node and four **Cores** appears as a **Package** box containing the NUMA node box above four **Core** boxes. If a NUMA node is local to the L3 Cache, then the NUMA node appears above that **Cache** box.

The PCI hierarchy does not appear as a set of included boxes; instead, it appears as a tree of bridges (that may actually be switches) with links between them. The tree starts with a small square on the left for the host bridge or root complex. It ends with PCI device boxes on the right. Intermediate PCI bridges/switches may appear as additional small squares in the middle.

Please see <https://github.com/open-mpi/hwloc/tree/master/utils/lstopo>* for additional information.

2.4 Portable Hardware Locality (hwloc) and hwloc-gui

`hwloc` and `hwloc-gui` help you visualize the system NUMA topology. `hwloc` includes other packages that help with viewing different aspects of NUMA management.

```
# yum install hwloc
# yum install hwloc-gui
```

2.4.1 hwloc-ls

`hwloc-ls` obtains CPU IDs when you need to know which cores to pin to your job to.

```
# hwloc-ls
Machine (251GB total) NUMANode L#0 (P#0 126GB)
  Package L#0
    L3 L#0 (32MB)
      L2 L#0 (512KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0
        PU L#0 (P#0)
        PU L#1 (P#128)
      L2 L#1 (512KB) + L1d L#1 (32KB) + L1i L#1 (32KB) + Core L#1
        PU L#2 (P#1)
        PU L#3 (P#129)
      L2 L#2 (512KB) + L1d L#2 (32KB) + L1i L#2 (32KB) + Core L#2
        PU L#4 (P#2)
        PU L#5 (P#130)
      L2 L#3 (512KB) + L1d L#3 (32KB) + L1i L#3 (32KB) + Core L#3
        PU L#6 (P#3)
        PU L#7 (P#131)
    .....
    .....
    .....
      L2 L#63 (512KB) + L1d L#63 (32KB) + L1i L#63 (32KB) + Core L#63
        PU L#126 (P#63)
        PU L#127 (P#191)
  HostBridge L#0
    PCIBridge
      PCI 15b3:1015
      Net L#0 "enp33s0f0"
      OpenFabrics L#1 "mlx5_0"
      PCI 15b3:1015
```

```
Net L#2 "enp33s0f1"
OpenFabrics L#3 "mlx5_1"
HostBridge L#2
  PCIBridge
    PCI 1022:7901
      Block(Disk) L#4 "sda"
HostBridge L#4
  PCIBridge
    PCI 1a03:2000
      GPU L#5 "controlD64"
      GPU L#6 "card0"
NUMANode L#1 (P#1 126GB) + Package L#1
  L3 L#8 (32MB)
    L2 L#64 (512KB) + L1d L#64 (32KB) + L1i L#64 (32KB) + Core L#64
      PU L#128 (P#64)
      PU L#129 (P#192)
.....
.....
    L2 L#77 (512KB) + L1d L#77 (32KB) + L1i L#77 (32KB) + Core L#77
      PU L#154 (P#77)
      PU L#155 (P#205)
.....
.....
    L2 L#127 (512KB) + L1d L#127 (32KB) + L1i L#127 (32KB) + Core L#127
      PU L#254 (P#127)
      PU L#255 (P#255)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)
```

2.4.1.1 hwloc-info

`hwloc-info` gets the system cache hierarchy.

```
# hwloc-info
depth 0:          1 Machine (type #1)
  depth 1:        2 NUMANode (type #2)
    depth 2:      2 Package (type #3)
      depth 3:    16 L3Cache (type #4)
        depth 4:  128 L2Cache (type #4)
          depth 5: 128 L1dCache (type #4)
            depth 6: 128 L1iCache (type #4)
              depth 7: 128 Core (type #5)
                depth 8: 256 PU (type #6)
Special depth -3: 7 Bridge (type #9)
Special depth -4: 4 PCI Device (type #10)
Special depth -5: 7 OS Device (type #11)
```

2.5 Istopo, hwloc-ls, hwloc-info, for AMD EPYC 7373X

2.5.1 Istopo

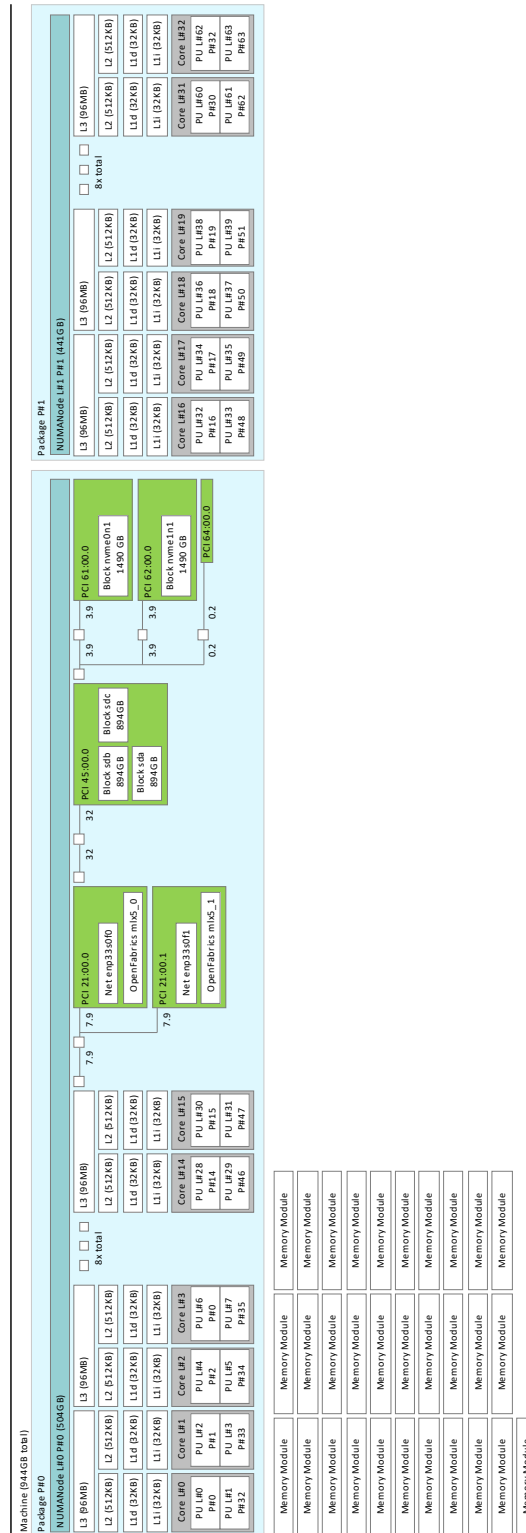


Figure 2-2: Istopo showing an AMD EPYC 7373X system

2.5.2 hwloc-info (For AMD EPYC 7373X)

```
# hwloc-info
depth 0:          1 Machine (type #0)
  depth 1:        2 Package (type #1)
    depth 2:      16 L3Cache (type #6)
      depth 3:    32 L2Cache (type #5)
        depth 4:  32 L1dCache (type #4)
          depth 5: 32 L1iCache (type #9)
            depth 6: 32 Core (type #2)
              depth 7: 64 PU (type #3)
Special depth -3: 2 NUMANode (type #13)
Special depth -4: 9 Bridge (type #14)
Special depth -5: 6 PCIDev (type #15)
Special depth -6: 9 OSDev (type #16)
Special depth -7: 32 Misc (type #17)
```

2.5.3 lwloc-ls (for AMD EPYC 7373X)

lwloc-ls shows a 96MB L3 cache size when AMD 3D V-Cache is enabled, as shown below.

```
]# hwloc-ls
Machine (944GB total)
  Package L#0
    NUMANode L#0 (P#0 504GB)
      L3 L#0 (96MB)
        L2 L#0 (512KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0
          PU L#0 (P#0)
          PU L#1 (P#32)
        L2 L#1 (512KB) + L1d L#1 (32KB) + L1i L#1 (32KB) + Core L#1
          PU L#2 (P#1)
          PU L#3 (P#33)
      L3 L#1 (96MB)
        L2 L#2 (512KB) + L1d L#2 (32KB) + L1i L#2 (32KB) + Core L#2
          PU L#4 (P#2)
          PU L#5 (P#34)
        L2 L#3 (512KB) + L1d L#3 (32KB) + L1i L#3 (32KB) + Core L#3
          PU L#6 (P#3)
          PU L#7 (P#35)
      L3 L#2 (96MB)
        L2 L#4 (512KB) + L1d L#4 (32KB) + L1i L#4 (32KB) + Core L#4
          PU L#8 (P#4)
          PU L#9 (P#36)
        L2 L#5 (512KB) + L1d L#5 (32KB) + L1i L#5 (32KB) + Core L#5
          PU L#10 (P#5)
          PU L#11 (P#37)
      L3 L#3 (96MB)
        L2 L#6 (512KB) + L1d L#6 (32KB) + L1i L#6 (32KB) + Core L#6
          PU L#12 (P#6)
          PU L#13 (P#38)
        L2 L#7 (512KB) + L1d L#7 (32KB) + L1i L#7 (32KB) + Core L#7
          PU L#14 (P#7)
          PU L#15 (P#39)
      L3 L#4 (96MB)
        L2 L#8 (512KB) + L1d L#8 (32KB) + L1i L#8 (32KB) + Core L#8
          PU L#16 (P#8)
          PU L#17 (P#40)
        L2 L#9 (512KB) + L1d L#9 (32KB) + L1i L#9 (32KB) + Core L#9
          PU L#18 (P#9)
          PU L#19 (P#41)
      L3 L#5 (96MB)
        L2 L#10 (512KB) + L1d L#10 (32KB) + L1i L#10 (32KB) + Core L#10
          PU L#20 (P#10)
```

```

    PU L#21 (P#42)
    L2 L#11 (512KB) + L1d L#11 (32KB) + L1i L#11 (32KB) + Core L#11
    PU L#22 (P#11)
    PU L#23 (P#43)
L3 L#6 (96MB)
    L2 L#12 (512KB) + L1d L#12 (32KB) + L1i L#12 (32KB) + Core L#12
    PU L#24 (P#12)
    PU L#25 (P#44)
    L2 L#13 (512KB) + L1d L#13 (32KB) + L1i L#13 (32KB) + Core L#13
    PU L#26 (P#13)
    PU L#27 (P#45)
L3 L#7 (96MB)
    L2 L#14 (512KB) + L1d L#14 (32KB) + L1i L#14 (32KB) + Core L#14
    PU L#28 (P#14)
    PU L#29 (P#46)
    L2 L#15 (512KB) + L1d L#15 (32KB) + L1i L#15 (32KB) + Core L#15
    PU L#30 (P#15)
    PU L#31 (P#47)
.....
.....
.....

```

Package L#1

```

NUMANode L#1 (P#1 441GB)
L3 L#8 (96MB)
    L2 L#16 (512KB) + L1d L#16 (32KB) + L1i L#16 (32KB) + Core L#16
    PU L#32 (P#16)
    PU L#33 (P#48)
    L2 L#17 (512KB) + L1d L#17 (32KB) + L1i L#17 (32KB) + Core L#17
    PU L#34 (P#17)
    PU L#35 (P#49)
L3 L#9 (96MB)
    L2 L#18 (512KB) + L1d L#18 (32KB) + L1i L#18 (32KB) + Core L#18
    PU L#36 (P#18)
    PU L#37 (P#50)
    L2 L#19 (512KB) + L1d L#19 (32KB) + L1i L#19 (32KB) + Core L#19
    PU L#38 (P#19)
    PU L#39 (P#51)
L3 L#10 (96MB)
    L2 L#20 (512KB) + L1d L#20 (32KB) + L1i L#20 (32KB) + Core L#20
    PU L#40 (P#20)
    PU L#41 (P#52)
    L2 L#21 (512KB) + L1d L#21 (32KB) + L1i L#21 (32KB) + Core L#21
    PU L#42 (P#21)
    PU L#43 (P#53)
L3 L#11 (96MB)
    L2 L#22 (512KB) + L1d L#22 (32KB) + L1i L#22 (32KB) + Core L#22
    PU L#44 (P#22)
    PU L#45 (P#54)
    L2 L#23 (512KB) + L1d L#23 (32KB) + L1i L#23 (32KB) + Core L#23
    PU L#46 (P#23)
    PU L#47 (P#55)
L3 L#12 (96MB)
    L2 L#24 (512KB) + L1d L#24 (32KB) + L1i L#24 (32KB) + Core L#24
    PU L#48 (P#24)
    PU L#49 (P#56)
    L2 L#25 (512KB) + L1d L#25 (32KB) + L1i L#25 (32KB) + Core L#25
    PU L#50 (P#25)
    PU L#51 (P#57)
L3 L#13 (96MB)
    L2 L#26 (512KB) + L1d L#26 (32KB) + L1i L#26 (32KB) + Core L#26
    PU L#52 (P#26)
    PU L#53 (P#58)
    L2 L#27 (512KB) + L1d L#27 (32KB) + L1i L#27 (32KB) + Core L#27
    PU L#54 (P#27)
    PU L#55 (P#59)

```

```

L3 L#14 (96MB)
  L2 L#28 (512KB) + L1d L#28 (32KB) + L1i L#28 (32KB) + Core L#28
    PU L#56 (P#28)
    PU L#57 (P#60)
  L2 L#29 (512KB) + L1d L#29 (32KB) + L1i L#29 (32KB) + Core L#29
    PU L#58 (P#29)
    PU L#59 (P#61)
L3 L#15 (96MB)
  L2 L#30 (512KB) + L1d L#30 (32KB) + L1i L#30 (32KB) + Core L#30
    PU L#60 (P#30)
    PU L#61 (P#62)
  L2 L#31 (512KB) + L1d L#31 (32KB) + L1i L#31 (32KB) + Core L#31
    PU L#62 (P#31)
    PU L#63 (P#63)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)
Misc(MemoryModule)

```

2.6 cpupower

`cpupower` is a collection of tools that examine and tune processor power-saving features, such as checking the CPU governor. The `cpupower monitor` samples the current CPU frequency in real time. From a Linux perspective, the CPU frequency scaling subsystem can modulating the CPU frequency based on workload feedback. Setting the governor to `performance` reduces OS frequency toggling. See [CPU Performance Scaling*](#) for additional information.

```

# cpupower frequency-info
analyzing CPU 0:
  driver: acpi-cpufreq
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency:  Cannot determine or is not supported.
  hardware limits: 1.50 GHz - 2.00 GHz
  available frequency steps: 2.00 GHz, 1.70 GHz, 1.50 GHz
  available cpufreq governors: conservative ondemand userspace powersave performance
  current policy: frequency should be within 1.50 GHz and 2.00 GHz.
                   The governor "performance" may decide which speed to use
                   within this range.
  current CPU frequency: 2.00 GHz (asserted by call to hardware)
  boost state support:
    Supported: yes
    Active: no
    Boost States: 0
    Total States: 3
    Pstate-P0: 2600MHz
    Pstate-P1: 2700MHz
    Pstate-P2: 2100MHz

```

Execute `cat /sys/devices/system/cpu/cpufreq/boost` to check whether boost is ON (1) or OFF (0).

```

# cat /sys/devices/system/cpu/cpufreq/boost
1

```

2.6.1 cpupower monitor

Check core frequencies and idle states.

```
[root@localhost ~]# cpupower monitor
```

PKG	CORE	CPU	Mperf			Idle States		
			CO	Cx	Freq	POLL	C1	C2
0	0	0	0.07	98.93	1948	0.00	0.00	100.8
0	0	128	0.45	98.93	1948	0.00	0.00	99.10
0	1	1	0.98	98.93	1948	0.00	0.00	100.0
0	1	129	0.99	98.93	1948	0.00	0.00	99.11
0	2	2	0.97	98.93	1948	0.00	0.00	100.0
0	2	130	0.99	98.93	1948	0.00	0.00	99.11
0	3	3	0.98	98.93	1948	0.00	0.00	100.0
0	3	131	0.99	98.93	1948	0.00	0.00	99.12
0	4	4	0.99	98.93	1948	0.00	0.00	100.0
0	4	132	1.00	98.93	1948	0.00	0.00	99.12
0	5	5	0.99	98.93	1948	0.00	0.00	100.0
0	5	133	0.99	98.93	1948	0.00	0.00	99.13
0	6	6	0.97	98.93	1948	0.00	0.00	100.0
0	6	134	1.00	98.93	1948	0.00	0.00	99.13
0	7	7	1.01	98.93	1948	0.00	0.00	100.0
0	7	135	1.02	98.93	1948	0.00	0.00	99.14
0	8	8	1.30	98.93	1948	0.00	0.00	100.0
0	8	136	1.29	98.93	1948	0.00	0.00	99.13
0	9	9	1.27	98.93	1948	0.00	0.00	100.8
0	9	137	1.28	98.93	1948	0.00	0.00	99.13
0	10	10	1.32	98.93	1948	0.00	0.00	100.8
0	10	138	1.32	98.93	1948	0.00	0.00	99.13
0	11	11	1.32	98.93	1948	0.00	0.00	100.8
0	11	139	1.33	98.93	1948	0.00	0.00	99.14
0	12	12	1.36	98.93	1948	0.00	0.00	100.8
0	12	140	1.36	98.93	1948	0.00	0.00	99.14
0	13	13	1.34	98.93	1948	0.00	0.00	100.8
0	13	141	1.34	98.93	1948	0.00	0.00	99.15
0	14	14	1.35	98.93	1948	0.00	0.00	100.8
0	14	142	1.37	98.93	1948	0.00	0.00	99.14
0	15	15	1.39	98.93	1948	0.00	0.00	100.8
0	15	143	1.40	98.93	1948	0.00	0.00	99.15
0	16	16	1.08	98.93	1948	0.00	0.00	100.8
0	16	144	1.09	98.93	1948	0.00	0.00	99.17

2.7 cpupower on AMD EPYC 7373X

2.7.1 3D V-Cache Disabled

```
# cat /sys/devices/system/cpu/cpufreq/boost
1

# cpupower frequency-info

analyzing CPU 0:
  driver: acpi-cpufreq
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: Cannot determine or is not supported.
  hardware limits: 1.50 GHz - 3.83 GHz
  available frequency steps: 3.05 GHz, 2.40 GHz, 1.50 GHz
```

```

available cpufreq governors: conservative ondemand userspace powersave performance
schedutil
current policy: frequency should be within 1.50 GHz and 3.05 GHz.
                    The governor "performance" may decide which speed to use
                    within this range.
current CPU frequency: 3.05 GHz (asserted by call to hardware)
boost state support:
  Supported: yes
  Active: yes
  Boost States: 0
  Total States: 3
  Pstate-P0: 3050MHz
  Pstate-P1: 2400MHz
  Pstate-P2: 1500MHz

```

```
# cpupower monitor
```

PKG	CORE	CPU	Mperf		Freq	Idle_Stats		
			C0	Cx		POLL	C1	C2
0	0	0	0.03	99.97	1796	0.00	0.00	99.97
0	0	32	0.01	99.99	2414	0.00	0.00	100.0
0	1	1	0.00	100.00	1794	0.00	0.00	100.0
0	1	33	0.00	100.00	2733	0.00	0.00	100.0
0	2	2	0.01	99.99	2259	0.00	0.00	100.0
0	2	34	0.03	99.97	2568	0.00	0.00	99.94
0	3	3	0.00	100.00	2543	0.00	0.00	100.1
0	3	35	0.00	100.00	2751	0.00	0.00	100.0
0	4	4	0.01	99.99	2669	0.00	0.00	100.1
0	4	36	0.00	100.00	2606	0.00	0.00	100.0
0	5	5	0.00	100.00	2939	0.00	0.00	100.1
0	5	37	0.02	99.98	2160	0.00	0.00	99.97
0	6	6	0.01	99.99	1796	0.00	0.00	100.1
0	6	38	0.00	100.00	2397	0.00	0.00	100.0
0	7	7	0.01	99.99	1796	0.00	0.00	100.0
0	7	39	0.01	99.99	2216	0.00	0.00	100.0
0	8	8	0.01	99.99	1795	0.00	0.00	100.0
0	8	40	0.01	99.99	2713	0.00	0.00	100.0
0	9	9	0.00	100.00	1795	0.00	0.00	100.1
0	9	41	0.00	100.00	3801	0.00	0.00	100.0
0	10	10	0.01	99.99	1796	0.00	0.00	100.0
0	10	42	0.01	99.99	2596	0.00	0.00	100.0
0	11	11	0.01	99.99	1796	0.00	0.00	100.0
0	11	43	0.01	99.99	2784	0.00	0.00	100.0
0	12	12	0.01	99.99	1795	0.00	0.00	100.0
0	12	44	0.02	99.98	2127	0.00	0.00	100.0
0	13	13	0.08	99.92	1784	0.00	0.00	100.00
0	13	45	0.00	100.00	3781	0.00	0.00	100.0
0	14	14	0.02	99.98	2609	0.00	0.00	100.0
0	14	46	0.02	99.98	2405	0.00	0.00	99.80
0	15	15	0.00	100.00	2623	0.00	0.00	100.0
0	15	47	0.00	100.00	3076	0.00	0.00	100.0
1	0	16	0.01	99.99	1768	0.00	0.00	100.0
1	0	48	0.01	99.99	1888	0.00	0.00	100.0
1	1	17	0.00	100.00	1795	0.00	0.00	100.0
1	1	49	0.01	99.99	2513	0.00	0.00	100.0
1	2	18	0.01	99.99	1796	0.00	0.00	100.0
1	2	50	0.01	99.99	2939	0.00	0.00	100.0
1	3	19	0.00	100.00	1796	0.00	0.00	100.0
1	3	51	0.00	100.00	3816	0.00	0.00	100.0
1	4	20	0.01	99.99	1795	0.00	0.00	100.0
1	4	52	0.01	99.99	2926	0.00	0.00	100.0
1	5	21	0.00	100.00	1796	0.00	0.00	100.0
1	5	53	0.01	99.99	2697	0.00	0.00	100.0
1	6	22	0.01	99.99	1795	0.00	0.00	100.0

```

1| 6| 54| 0.00|100.00| 2262|| 0.00| 0.00| 100.0
1| 7| 23| 0.00|100.00| 1796|| 0.00| 0.00| 100.0
1| 7| 55| 0.00|100.00| 2190|| 0.00| 0.00| 100.0
1| 8| 24| 0.01| 99.99| 1795|| 0.00| 0.00| 100.0
1| 8| 56| 0.00|100.00| 2274|| 0.00| 0.00| 100.0
1| 9| 25| 0.03| 99.97| 1773|| 0.00| 0.86| 99.15
1| 9| 57| 0.00|100.00| 2176|| 0.00| 0.00| 100.0
1| 10| 26| 0.01| 99.99| 1775|| 0.00| 0.00| 100.0
1| 10| 58| 0.00|100.00| 3815|| 0.00| 0.00| 100.0
1| 11| 27| 0.01| 99.99| 1796|| 0.00| 0.00| 100.0
1| 11| 59| 0.00|100.00| 3250|| 0.00| 0.00| 100.0
1| 12| 28| 0.01| 99.99| 1795|| 0.00| 0.00| 100.0
1| 12| 60| 0.00|100.00| 2286|| 0.00| 0.00| 100.0
1| 13| 29| 0.07| 99.93| 1795|| 0.00| 0.00| 99.96
1| 13| 61| 0.00|100.00| 2185|| 0.00| 0.00| 100.0
1| 14| 30| 0.01| 99.99| 2033|| 0.00| 0.00| 100.0
1| 14| 62| 0.00|100.00| 2266|| 0.00| 0.00|100.00
1| 15| 31| 0.00|100.00| 2139|| 0.00| 0.00| 100.0
1| 15| 63| 0.25| 99.75| 3694|| 0.00| 0.00| 99.76

```

2.7.2 3D V-Cache Enabled

```

# cat /sys/devices/system/cpu/cpufreq/boost
1

# cpupower frequency-info

analyzing CPU 0:
driver: acpi-cpufreq
CPUs which run at the same hardware frequency: 0
CPUs which need to have their frequency coordinated by software: 0
maximum transition latency: Cannot determine or is not supported.
hardware limits: 1.50 GHz - 3.83 GHz
available frequency steps: 3.05 GHz, 2.40 GHz, 1.50 GHz
available cpufreq governors: conservative ondemand userspace powersave performance
schedutil
current policy: frequency should be within 1.50 GHz and 3.05 GHz.
                    The governor "performance" may decide which speed to use
                    within this range.
current CPU frequency: 3.05 GHz (asserted by call to hardware)
boost state support:
  Supported: yes
  Active: yes
  Boost States: 0
  Total States: 3
  Pstate-P0: 3050MHz
  Pstate-P1: 2400MHz
  Pstate-P2: 1500MHz

# cpupower monitor

          | Mperf          || Idle_Stats
PKG|CORE| CPU| C0  | Cx  | Freq  || POLL | C1  | C2
0| 0| 0| 0.01| 99.99| 3249|| 0.00| 0.00| 99.98
0| 0| 32| 0.02| 99.98| 1904|| 0.00| 0.00| 99.98
0| 1| 1| 0.01| 99.99| 2039|| 0.00| 0.00| 100.0
0| 1| 33| 0.00|100.00| 2729|| 0.00| 0.00| 100.0
0| 2| 2| 0.01| 99.99| 2034|| 0.00| 0.00| 100.1
0| 2| 34| 0.00|100.00| 2703|| 0.00| 0.00| 100.0
0| 3| 3| 0.02| 99.98| 1966|| 0.00| 0.00| 100.1
0| 3| 35| 0.00|100.00| 2439|| 0.00| 0.00| 100.0
0| 4| 4| 0.01| 99.99| 1795|| 0.00| 0.00| 100.0
0| 4| 36| 0.00|100.00| 2822|| 0.00| 0.00| 100.0

```

0	5	5	0.00	100.00	1795		0.00	0.00	100.0
0	5	37	0.00	100.00	2702		0.00	0.00	100.0
0	6	6	0.01	99.99	2705		0.00	0.00	100.0
0	6	38	0.01	99.99	2632		0.00	0.00	100.0
0	7	7	0.04	99.96	2040		0.00	0.00	99.73
0	7	39	0.00	100.00	3554		0.00	0.00	100.0
0	8	8	0.01	99.99	1796		0.00	0.00	100.0
0	8	40	0.01	99.99	2017		0.00	0.00	100.0
0	9	9	0.00	100.00	1795		0.00	0.00	100.0
0	9	41	0.00	100.00	2085		0.00	0.00	100.0
0	10	10	0.01	99.99	1795		0.00	0.00	100.0
0	10	42	0.09	99.91	1887		0.00	0.36	99.46
0	11	11	0.00	100.00	1796		0.00	0.00	100.0
0	11	43	0.00	100.00	3620		0.00	0.00	100.0
0	12	12	0.01	99.99	1705		0.00	0.00	100.0
0	12	44	0.00	100.00	3577		0.00	0.00	100.0
0	13	13	0.43	99.57	1973		0.00	0.00	99.61
0	13	45	0.01	99.99	3141		0.00	0.00	100.00
0	14	14	0.01	99.99	1796		0.00	0.00	100.0
0	14	46	0.00	100.00	3778		0.00	0.00	100.0
0	15	15	0.02	99.98	1745		0.00	0.00	100.0
0	15	47	0.00	100.00	3567		0.00	0.00	100.00
1	0	16	0.02	99.98	1796		0.00	0.00	100.0
1	0	48	0.01	99.99	2928		0.00	0.00	100.00
1	1	17	0.01	99.99	1795		0.00	0.00	100.0
1	1	49	0.00	100.00	2795		0.00	0.00	100.0
1	2	18	0.01	99.99	1795		0.00	0.00	100.0
1	2	50	0.01	99.99	1698		0.00	0.00	100.0
1	3	19	0.01	99.99	1796		0.00	0.00	100.0
1	3	51	0.01	99.99	1676		0.00	0.00	100.0
1	4	20	0.01	99.99	1795		0.00	0.00	100.0
1	4	52	0.00	100.00	3224		0.00	0.00	100.0
1	5	21	0.00	100.00	1796		0.00	0.00	100.0
1	5	53	0.00	100.00	3326		0.00	0.00	100.0
1	6	22	0.01	99.99	1796		0.00	0.00	100.0
1	6	54	0.01	99.99	1985		0.00	0.00	99.99
1	7	23	0.01	99.99	1795		0.00	0.00	100.0
1	7	55	0.01	99.99	2395		0.00	0.00	99.99
1	8	24	0.01	99.99	1795		0.00	0.00	100.0
1	8	56	0.00	100.00	3606		0.00	0.00	100.0
1	9	25	0.01	99.99	1793		0.00	0.00	100.0
1	9	57	0.01	99.99	2502		0.00	0.00	100.00
1	10	26	0.01	99.99	1706		0.00	0.00	100.0
1	10	58	0.00	100.00	3548		0.00	0.00	100.0
1	11	27	0.01	99.99	1584		0.00	0.00	100.0
1	11	59	0.00	100.00	3816		0.00	0.00	100.0
1	12	28	0.01	99.99	1579		0.00	0.00	100.00
1	12	60	0.00	100.00	3815		0.00	0.00	100.0
1	13	29	0.01	99.99	1669		0.00	0.00	100.00
1	13	61	0.00	100.00	3585		0.00	0.00	100.0
1	14	30	0.01	99.99	1576		0.00	0.00	99.99
1	14	62	0.00	100.00	3553		0.00	0.00	100.00
1	15	31	0.01	99.99	1668		0.00	0.00	99.99
1	15	63	0.17	99.83	3706		0.00	0.00	99.84

2.8 CPU Governors

AMD EPYC supports several CPU governors. Different governors can be applied to different cores. For example, the performance governor is often used in a High-Performance Computing environment.

- **performance:** Sets the core frequency to the highest available frequency within P0.
- **Boost=OFF:** The CPU will operate at the base frequency, e.g., 2.25GHz on an AMD EPYC 7742 CPU.
- **Boost=ON:** The CPU will attempt to boost the frequency up to the Max Boost frequency of 3.4GHz. While operating at the boosted frequencies, this still represents the P0 P-state.
- **ondemand:** Sets the core frequency depending on the trailing load. This favors a rapid ramp to the highest operating frequency with a subsequent slow step down to P2 when idle. This could penalize short-lived threads.
- **conservative:** Similar to `ondemand` but favors a more graceful ramp to highest frequency and a rapid return to P2 at idle.
- **powersave:** Sets the lowest-supported core frequency, locking it to P2.

Administrators can execute the `cpupower` command to set the CPU governor. For example, to set the CPU governor to Performance:

```
cpupower frequency-set -g performance
```

Please see [Linux CPUFreq Governors](#)* for a more extensive discussion and explanation of Linux CPU governors.

Here are some examples of using `cpupower` to query and set a range of conditions on the CPU:

```
cpupower -c 0-15 monitor
```

Displays the frequencies on cores 0 to 15. Useful if a user needs to observe the changes while turning Boost ON and OFF.

```
cpupower frequency-info
```

Lists the boost state, CPU governor, and other useful information about the CPU configuration.

```
cpupower frequency-set -g performance
```

Changes the CPU governor to `performance`.

```
cpupower -c 0-15 idle-set -d 2
```

Disables the C2 idle state on CPUs 0 to 15.

2.9 top

`top` provides a dynamic view of the resources being consumed by various processes. While running `top`, you can

- Press [1] to view a per-CPU breakdown of utilization statistics.
- Press [2] to view per-NUMA-node utilization statistics.
- Press [3] to select and highlight a NUMA node and view summary information.

2.9.1 Example 1: Per-CPU Utilization Statistics

```
top - 13:52:14 up 8 days, 14:11, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 2303 total, 1 running, 2302 sleeping, 0 stopped, 0 zombie
%Cpu0  :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1  :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu5  :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu6  :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu7  :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu8  :  0.0 us,  0.3 sy,  0.0 no, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu9  :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu10 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu11 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu12 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu14 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu15 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu16 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu17 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu18 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu19 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu20 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu21 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu22 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu23 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu24 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu25 :  0.0 us,  0.0 sy,  0.0 no,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

2.9.2 Example 2: Per-NUMA-Node Utilization Statistics

```
top - 13:54:20 up 8 days, 14:13, 2 users, load average: 0.06, 0.02, 0.00
Tasks: 2303 total, 1 running, 2302 sleeping, 0 stopped, 0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Node0  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Node1  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 128720.3 total, 122010.6 free,  4436.8 used,  2272.9 buff/cache
MiB Swap: 10240.0 total, 10240.0 free,    0.0 used. 123220.1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 167565 root      20   0   67988   7652  4304 R   0.7   0.0   0:00.17 top
    1 root      20   0 246364 15072  9248 S   0.0   0.0   0:09.28 systemd
    2 root      20   0     0     0     0 S   0.0   0.0   0:00.28 kthreadd
    3 root       0 -20     0     0     0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20     0     0     0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root       0 -20     0     0     0 I   0.0   0.0   0:00.00 kworker/0:0H-
events_highpri
    8 root      20   0     0     0     0 I   0.0   0.0   0:00.00 kworker/u512:0-edac-
poller
   10 root       0 -20     0     0     0 I   0.0   0.0   0:00.00 mm_percpu_wq
```

```

11 root      20    0    0    0    0 S  0.0  0.0  0:00.01 ksoftirqd/0
12 root      20    0    0    0    0 I  0.0  0.0  3:03.83 rcu_sched
13 root      rt    0    0    0    0 S  0.0  0.0  0:00.00 migration/0
14 root      rt    0    0    0    0 S  0.0  0.0  0:00.28 watchdog/0
15 root      20    0    0    0    0 S  0.0  0.0  0:00.00 cpuhp/0
16 root      20    0    0    0    0 S  0.0  0.0  0:00.00 cpuhp/1
17 root      rt    0    0    0    0 S  0.0  0.0  0:00.54 watchdog/1
18 root      rt    0    0    0    0 S  0.0  0.0  0:00.00 migration/1
19 root      20    0    0    0    0 S  0.0  0.0  0:00.00 ksoftirqd/1
21 root      0   -20   0    0    0 I  0.0  0.0  0:00.00 kworker/1:0H
22 root      20    0    0    0    0 S  0.0  0.0  0:00.00 cpuhp/2
.....

```

2.9.3 Example 3: Utilization Statistics Summary

```

top - 13:56:59 up 8 days, 14:15,  2 users, load average: 0.02, 0.02, 0.00
Tasks: 2303 total,  1 running, 2302 sleeping,  0 stopped,  0 zombie
%Node1 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu64 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu65 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu66 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu67 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu68 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu69 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu70 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu71 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu72 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu73 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu74 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu75 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu76 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu77 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu78 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu79 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu80 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu81 :  0.0 us,  0.0 sy,  0.0 ni, 100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
.....
.....

```

2.10 tuned

`tuned` is a daemon that uses `udev` to monitor connected devices and statically and dynamically tunes system settings according to a selected profile. Tuned is distributed with a number of predefined profiles for common use cases such as high throughput, low latency, or powersave. You can modify the defined rules for each profile and customize how to tune a particular device. See [Performance Tuning Guidelines for Low Latency Response on AMD EPYC 7002™ Series Processor Based Servers](#) for information on tuning Linux for low latency.

2.11 tuna

`tuna` simplifies adjusting tunable scheduler parameters such as thread priority and IRQ handlers, and can also isolate CPU cores and sockets. After installation, execute the `tuna` command without any arguments to start the Tuna graphical user interface (GUI). You can also execute the `tuna -h` command to view available Command Line Interface (CLI) options. See [Red Hat Enterprise Linux 8 - Monitoring and Managing System Status and Performance*](#) for complete tuned and tuna instructions.

This page intentionally left blank.

General Tuning Recommendations

3.1 LLC as NUMA Domain

Certain datacenter applications that use a remote job scheduler to manage workloads can benefit from pinning execution to a single NUMA node and (preferably) to share a single Last-Level Cache (LLC or L3 cache) within that node. This can be done if the system BIOS includes the L3AsNumaNode setting, which creates a NUMA node for each system CCX (L3 cache) when enabled.

Enabling this setting can improve performance for highly NUMA-optimized workloads if either workloads or components of workloads can:

- Be pinned to cores in a CCX.
- Benefit from sharing an L3 cache.

Please see https://devhub.amd.com/wp-content/uploads/Docs/56795_1.10.pdf (login required) for additional information about NUMA architecture and settings for AMD EPYC 7003 Series Processors.

1P System AMD EPYC 7003 Series Processor with 8 CCDs	# NUMA Nodes with LLCasNUMA is		Memory Interleaving when LLCasNUMA is (Enabled or Disabled)
	Enabled	Disabled	
NPS1	8	1	Across all eight channels in a socket
NPS2	8	2	Across groups of four channels (ABCD and EFGH) in a socket
NPS4	8	4	Across pairs of two channels (AB, CD, EF and GH) in a socket

Table 3-1: #NUMA nodes and memory interleaving for 1P AMD EPYC 7003 Series Processor systems with 8 CCDs

2P System AMD EPYC 7003 Series Processor with 8 CCDs	# NUMA Nodes with LLCasNUMA is		Memory Interleaving when LLCasNUMA is (Enabled or Disabled)
	Enabled	Disabled	
NPS0	16	1	Across all channels in both sockets
NPS1	16	2	Across all eight channels in each socket
NPS2	16	4	Across groups of four channels (ABCD and EFGH) in each socket
NPS4	16	8	Across pairs of two channels (AB, CD, EF and GH) in each socket

Table 3-2: #NUMA nodes and memory interleaving for 2P AMD EPYC 7003 Series Processor systems with 8 CCDs

Please see [Memory Population Guidelines for AMD EPYC 7003 Series Processors](#) for additional information.

Note: Note: All AMD EPYC 7003C Series Processors have 8 CCDs.

3.2 AMD uProf

AMD uProf is a performance analysis tool for applications running on Windows and Linux. Developers can use this tool to better understand application runtime performance and identify ways to optimize performance. AMD uProf offers:

- **Performance Analysis:** The CPU profiling utility identifies application runtime performance bottlenecks.
- **System Analysis:** The Performance Counter Monitor utility monitors system performance metrics.
- **Power Profiling:** System-wide power profiling monitors system thermal and power characteristics.
- **Energy Analysis:** The Power Application Analysis utility identifies energy hotspots in Windows applications.

Please see <https://developer.amd.com/amd-uprof/> for more information about AMD uProf.

3.3 perf

`perf` is a powerful tool that helps monitor various OS subsystems at the server, process, or process subset level to detect and identify performance bottlenecks and possibly tune the OS. Here are two examples of `perf` functionality and usage:

- `perf record` samples the function calls executed by a process or processes and writes output to `perf.data`.
- `perf report` reads the `perf.data` file and prints a human-readable report of top function calls grouped by function calls and ordered by count.

For example:

- `perf record -a -e cycles sleep 30` captures 30 seconds of data for the entire system.

```
# perf record -a -e cycles sleep 30
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 2.154 MB perf.data (13632 samples) ]
```

- `perf record -e cycles <command>` gathers profile information for a given workload.

```
# perf record -e cycles ls -R > /dev/null
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.346 MB perf.data (8679 samples) ]
```

You can also use trace points or create probe points using either `perf` or `trace-cmd` to gather specific information on the OS. Describing these analyses is beyond the scope of this tuning guide.

Here are a few invocation examples of `perf` commands.

3.3.1 perf list cpu

The `perf list cpu` command displays the symbolic events that you can select in the various `perf` commands using the `-e` option.

```
# perf list cpu
cpu-cycles OR cycles [Hardware event]

cpu-clock [Software event]
cpu-migrations OR migrations [Software event]

branch-instructions OR cpu/branch-instructions/ [Kernel PMU event]
branch-misses OR cpu/branch-misses/ [Kernel PMU event]
cache-misses OR cpu/cache-misses/ [Kernel PMU event]
cache-references OR cpu/cache-references/ [Kernel PMU event]
cpu-cycles OR cpu/cpu-cycles/ [Kernel PMU event]
instructions OR cpu/instructions/ [Kernel PMU event]
stalled-cycles-backend OR cpu/stalled-cycles-backend/ [Kernel PMU event]
stalled-cycles-frontend OR cpu/stalled-cycles-frontend/ [Kernel PMU event]
cpuhp:cpuhp_enter [Tracepoint event]
cpuhp:cpuhp_exit [Tracepoint event]
cpuhp:cpuhp_multi_enter [Tracepoint event]
kmem:mm_page_pcpu_drain [Tracepoint event]
kvm:kvm_cpuid [Tracepoint event]
kvm:kvm_vcpu_wakeup [Tracepoint event]
kvm:vcpu_match_mmio [Tracepoint event]
percpu:percpu_alloc_percpu [Tracepoint event]
percpu:percpu_alloc_percpu_fail [Tracepoint event]
percpu:percpu_create_chunk [Tracepoint event]
percpu:percpu_destroy_chunk [Tracepoint event]
percpu:percpu_free_percpu [Tracepoint event]
power:cpu_frequency [Tracepoint event]
power:cpu_frequency_limits [Tracepoint event]
power:cpu_idle [Tracepoint event]
syscalls:sys_enter_getcpu [Tracepoint event]
syscalls:sys_exit_getcpu [Tracepoint event]
xdp:xdp_cpumap_enqueue [Tracepoint event]
xdp:xdp_cpumap_kthread [Tracepoint event]
xen:xen_cpu_load_idt [Tracepoint event]
xen:xen_cpu_set_ldt [Tracepoint event]
xen:xen_cpu_write_gdt_entry [Tracepoint event]
xen:xen_cpu_write_idt_entry [Tracepoint event]
xen:xen_cpu_write_ldt_entry [Tracepoint event]
```

3.3.2 perf list cache

The `perf list cache` command displays the pre-defined events that you can select in the various `perf` commands using the `-e` option.

```
# perf list cache
L1-dcache-load-misses L1-dcache-loads [Hardware cache event]
L1-dcache-prefetches [Hardware cache event]
L1-icache-load-misses [Hardware cache event]
L1-icache-loads [Hardware cache event]
branch-load-misses [Hardware cache event]
branch-loads [Hardware cache event]
dTLB-load-misses [Hardware cache event]
dTLB-loads [Hardware cache event]
iTLB-load-misses [Hardware cache event]
iTLB-loads [Hardware cache event]
```

3.3.3 perf stat

`perf-stat` executes a command and gathers performance counter statistics using the following syntax:

```
perf stat [-e <EVENT> | --event=EVENT] [-a] <command>perf stat [-e <EVENT> | --
event=EVENT] [-a] - <command> [<options>]perf stat [-e <EVENT> | --event=EVENT] [-a] record
[- o file] - <command> [<options>]perf stat report [-i file]
```

Here is a `perf` example:

```
# perf stat -e L1-dcache-loads,L1-dcache-load-misses,L1-dcache-prefetches ls -R >
/dev/null

Performance counter stats for 'ls -R':
 2,872,769,836      L1-dcache-loads
 48,549,215        L1-dcache-load-misses      #    1.69% of all L1-dcache hits 14,454,120
 14,454,120        L1-dcache-prefetches

 2.121991076 seconds time elapsed
 0.933120000 seconds user
 1.177917000 seconds sys

perf stat -e dTLB-loads,dTLB-load-misses ls -R > /dev/null

Performance counter stats for 'ls -R':
 13,965,275        dTLB-loads
 726,322           dTLB-load-misses      #    5.20% of all dTLB cache hits

 2.185590119 seconds time elapsed
 0.912492000 seconds user
 1.262729000 seconds sys

# perf stat -e branch-loads,branch-load-misses ls -R > /dev/null

Performance counter stats for 'ls -R':
 1,749,235,099      branch-loads
 9,749,633          branch-load-misses

 2.080123904 seconds time elapsed
 0.904804000 seconds user
```

Here is another `perf` example:

```
# perf stat -e L1-dcache-loads,L1-dcache-load-misses,L1-dcache-prefetches ls -R > /dev/
null

Performance counter stats for 'ls -R':
 1,011,851         L1-dcache-loads
 36,970           L1-dcache-load-misses      #    3.65% of all L1-dcache accesses
 11,107           L1-dcache-prefetches

 0.001932625 seconds time elapsed
 0.000000000 seconds user
 0.002084000 seconds sys

# perf stat -e dTLB-loads,dTLB-load-misses ls -R > /dev/null
```

```

Performance counter stats for 'ls -R':
          6,600      dTLB-loads
          884      dTLB-load-misses          #   13.39% of all dTLB cache accesses

0.001716656 seconds time elapsed

0.001849000 seconds user
0.000000000 seconds sys

# perf stat -e branch-loads,branch-load-misses ls -R > /dev/null

Performance counter stats for 'ls -R':
          511,661      branch-loads
          18,989      branch-load-misses

0.001418633 seconds time elapsed

0.001490000 seconds user
0.000000000 seconds sys

```

The core AMD PMU has a 4-bit-wide per-cycle increment for each performance monitor counter. This works for most counters, but AMD EPYC Family 17h and above processors can have more than 15 events occur in a cycle. These events are called “Large Increment per Cycle” events. One example is the number of SSE/AVX FLOPs retired (event code `0x003`). To count these events, two adjacent hardware PMCs have their count signals merged to form a total of 8 bits per cycle. The `PERF_CTR` count registers also merge so as to count up to 64 bits.

Normally, events such as instructions retired get programmed on a single counter. For example:

```

PERF_CTL0 (MSR 0xc0010200) 0x0000000000053ff0c # event 0x0c, umask 0xff
PERF_CTR0 (MSR 0xc0010201) 0x00008000000000001 # r/w 48-bit count

```

The next counter at MSRs `0xc0010202-3` either remains unused or can be used independently to count something else.

When counting Large Increment per Cycle events, such as FLOPs, we have to reserve the next counter and program the `PERF_CTL (config)` register with the Merge event (`0xFFFF`). For example:

```

PERF_CTL0 (msr 0xc0010200) 0x0000000000053ff03 # FLOPs event, umask 0xff
PERF_CTR0 (msr 0xc0010201) 0x00008000000000001 # read 64-bit count, wr low 48b
PERF_CTL1 (msr 0xc0010202) 0x00000000f004000ff # Merge event, enable bit
PERF_CTR1 (msr 0xc0010203) 0x00000000000000000 # write higher 16-bits of count

```

The count widens from the normal 48 bits to 64 bits by having the second counter carry the higher 16 bits of the count in the lower 16 bits of its counter register. This version does not support mixed 48-bit and 64-bit counts. Here is an example for an AMD EPYC 7003 processor:

```
# perf stat -e cpu/instructions/,cpu/event=0x03,umask=0xff/ ls -R > /dev/null
Performance counter stats for 'ls -R':

   8,541,788,345      cpu/instructions/
     244,511         cpu/event=0x03, umask=0xff/

   2.103605730 seconds time elapsed
   0.909099000 seconds user
   1.184830000 seconds sys
```

Here is the same example for an AMD EPYC processor with AMD 3D V-Cache technology:

```
# perf stat -e cpu/instructions/,cpu/event=0x03,umask=0xff/ ls -R > /dev/null
Performance counter stats for 'ls -R':

     2,585,402      cpu/instructions/
           5        cpu/event=0x03,umask=0xff/

   0.001151648 seconds time elapsed
   0.000000000 seconds user
   0.001248000 seconds sys
```

Chapter

4

Virtualization

RHEL 8 includes virtualization functionality that allows a machine running RHEL 8 to host multiple virtual machines (VMs), also referred to as guests. VMs use the host's physical hardware and computing resources to run a separate, virtualized operating system (guest OS) as a user-space process on the host OS.

4.1 Secure Encrypted Virtualization (SEV)

AMD Secure Encrypted Virtualization (SEV) protects the data in DRAM while in use by a running VM instance. SEV encrypts the memory of each instance with a unique key. Executing the following command tells you whether the deployment is SEV-capable:

```
# lscpu | grep -i sev

Flags:                fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm
constant_tsc rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclmulqdq
monitor ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c
rdrand lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch
osvw ibs skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb
cat_l3 cdp_l3 invpcid_single hw_pstate sme ssbd mba sev ibrs ibpb stibp vmmcall
fsgsbase bmi1 avx2 smep bmi2 erms invpcid cqm rdt_a rdseed adx smap clflushopt clwb
sha_ni xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total
cqm_mbm_local clzero irperf xsaveerptr wbnoinvd amd_ppin arat npt lbrv svm_lock
nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold
v_vmsave_vmload vgif umip pku ospke vaes vpclmulqdq rdpid overflow_recov succor smca
```

If enabled on a virtual machine (VM), then SEV encrypts VM memory, which prevents the host from accessing data on the VM. This increases VM security if the host is successfully breached. The host hardware version determines how many VMs can use this feature simultaneously on a single host.

```
# dmesg | grep -i sev
[    0.938380] AMD Memory Encryption Features active: SEV SEV-ES
```

Enabling SEV requires all DMA operations inside the guest to use shared memory. SEV makes this transparent to the guest by using the SWIOTLB Linux kernel pool, which has a default size of 64MB. A guest panic will occur if the Linux kernel exhausts the SWIOTLB pool. The number of devices used by the guest and the utilization of these devices directly impacts the amount of SWIOTLB required. AMD recommends increasing the SWIOTLB pool that the Linux kernel allocates for SEV guests, with 512MB as the recommended starting size.

4.1.1 SEV Prerequisites

To ensure SEV support,

- The deployment must include a compute node that runs on SEV-capable AMD hardware, such as an AMD EPYC CPU.
- The deployment must include `libvirt` 4.5 or later, which includes SEV support.
- The operating system running in an encrypted instance must support SEV. For example:

```
# dmesg | grep -i sev
[ 8.723924] ccp 0000:22:00.1: sev enabled
[ 8.820716] ccp 0000:22:00.1: SEV API:1.13 build:13
```

Please see the [AMD Secure Encrypted Virtualization \(SEV\) documentation](#) for more information.

4.2 AMD EPYC Virtualization Support

By default, the virtualization packages are not installed, as shown below. Be sure to install the virtualization packages.

```
# yum install libvirt
Updating Subscription Management repositories
Last metadata expiration check 0:00:11 ago on Sat 16 Jan 2021 10:49:54 AM PST
Dependencies resolved
=====
Package                               Architecture Version                               Repository                               Size
=====
Installing:
libvirt                                x86_64      6.0.0-28.module+el8.3.0+7827+5e65edd7  rhel8-for-x86_64-appstream-rpms        53 k
Installing dependencies:
autogen-libopts                        x86_64      5.18.12-8.el8                          rhel8-for-x86_64-appstream-rpms        75 k
gnutls-dane                            x86_64      3.6.14-7.el8_3                          rhel8-for-x86_64-appstream-rpms        51 k
gnutls-utils                           x86_64      3.6.14-7.el8_3                          rhel8-for-x86_64-appstream-rpms       347 k
libvirt-bash-completion                x86_64      6.0.0-28.module+el8.3.0+7827+5e65edd7  rhel8-for-x86_64-appstream-rpms        54 k
libvirt-client                         x86_64      6.0.0-28.module+el8.3.0+7827+5e65edd7  rhel8-for-x86_64-appstream-rpms       361 k
libvirt-daemon-config-nwfilter         x86_64      6.0.0-28.module+el8.3.0+7827+5e65edd7  rhel8-for-x86_64-appstream-rpms        59 k
Transaction Summary
=====
Install 7 Packages
```

Validate that the virtualization host and packages are installed by executing the command `virt-host-validate`, and then verifying that all of the validations show `PASS`. If not, then adjust the required parameters as recommended in the `virt-host-validate` output.

```
# virt-host-validate
QEMU: Checking for hardware virtualization           : PASS
QEMU: Checking if device /dev/kvm exists             : PASS
QEMU: Checking if device /dev/kvm is accessible     : PASS
QEMU: Checking if device /dev/vhost-net exists       : PASS
QEMU: Checking if device /dev/net/tun exists         : PASS
QEMU: Checking for cgroup 'cpu' controller support   : PASS
QEMU: Checking for cgroup 'cpuacct' controller support : PASS
QEMU: Checking for cgroup 'cpuset' controller support : PASS
QEMU: Checking for cgroup 'memory' controller support : PASS
QEMU: Checking for cgroup 'devices' controller support : PASS
QEMU: Checking for cgroup 'blkio' controller support : PASS
QEMU: Checking for device assignment IOMMU support   : PASS
QEMU: Checking if IOMMU is enabled by kernel         : PASS
QEMU: Checking for secure guest support              : WARN
```

(AMD Secure Encrypted Virtualization appears to be disabled in kernel. Add `kvm_amd.sev=1` to the kernel cmdline arguments)

The `qemu-kvm` command allows you to view and validate AMD EPYC processor support.

```
# /usr/libexec/qemu-kvm -cpu help | grep -i amd
x86 EPYC-Rome-v1          x86 AMD EPYC-Rome Processor
EPYC-v1                  AMD EPYC Processor
x86 EPYC-v2              AMD EPYC Processor (with IBPB)
x86 Opteron_G1-v1        AMD Opteron 240 (Gen 1 Class Opteron)
x86 Opteron_G2-v1        AMD Opteron 22xx (Gen 2 Class Opteron)
x86 Opteron_G3-v1        AMD Opteron 23xx (Gen 3 Class Opteron)
x86 Opteron_G4-v1        AMD Opteron 62xx class CPU
x86 Opteron_G5-v1        AMD Opteron 63xx class CPU
x86 phenom-v1            AMD Phenom(tm) 9550 Quad-Core Processor
3dnow 3dnowext 3dnowprefetch abm ace2 ace2-en acpi adx aes amd-no-ssb
amd-ssbd amd-stibp apic arat arch-capabilities avx avx2 avx512-4fmaps
```

The following command validates AMD EPYC support:

```
# /usr/libexec/qemu-kvm -cpu help | grep -i epyc
x86 EPYC                  (alias configured by machine type)
x86 EPYC-IBPB             (alias of EPYC-v2)
x86 EPYC-Rome             (alias configured by machine type)
x86 EPYC-Rome-v1         AMD EPYC-Rome Processor
x86 EPYC-v1               AMD EPYC Processor
x86 EPYC-v2               AMD EPYC Processor (with IBPB)
```

4.3 Resource Allocation and Host/VM Tuning

Please see the Red Hat Enterprise Linux documentation at [Red Hat Enterprise Linux 8 - Configuring and Managing Virtualization](#)* for instructions on setting up a virtualization host, creating and administering VMs, and understanding the virtualization features in Red Hat Enterprise Linux

4.4 Tuning the Virtualization Host

Please see <https://access.redhat.com/solutions/5427>* for suggested I/O schedulers to improve disk performance when using Red Hat Enterprise Linux with virtualization.

4.5 Evaluating Workloads and VM Workloads

NUMA-aware or highly parallelizable workloads can take maximum advantage of AMD EPYC 7003 Series Processor architecture for performance tuning and gains. Memory-bound, NUMA friendly workloads can be parallelized to have each thread run independently. I/O-bound workloads can support multiple devices such that each device can remain connected to the original task owner. For example:

- The original STREAM benchmark can be parallelized with OpenMP and extended to measure NUMA-aware workload performance.
- The OpenMP and MPI Versions of the NASA Parallel Benchmarks (NPB) can be another good example test workload.

Both examples make very effective test VM workloads.

This page intentionally left blank.

Troubleshooting and Debugging Notes

This section describes the following troubleshooting and debugging tools and procedures:

- [“Error Detection and Correction \(EDAC\)” on page 33](#)
- [“Error Injection” on page 35](#)

5.1 Error Detection and Correction (EDAC)

Red Hat Enterprise Linux for AMD EPYC CPUs includes two AMD-compatible Error Detection and Correction (EDAC) modules for diagnosing memory errors:

Linux has two EDAC modules:

- `amd64_edac_mod` provides information and DRAM ECC-specific decoding. It loads automatically on supported systems if not blacklisted/disabled.
- `edac_mce_amd` provides more detailed MCA error decoding and is loaded by `amd64_edac_mod`.

```
# lsmod | grep -i edac
amd64_edac_mod      40960      0
edac_mce_amd        32768      1 amd64_edac_mod
```

System administrators monitoring ECC should monitor system health by continually recording both the number and rate of change of correctable and uncorrectable errors. The EDAC driver provides details about the number of memory controllers, memory controller characteristics, and physical DIMM characteristics (number of chip-select rows [`csrows`], and the channel tables).

The `/sys` filesystem for each `csrow` contain a number of entries with detailed information about the specific DIMM:

```
/sys/devices/system/edac/mc/mc0/csrow0
/sys/devices/system/edac/mc/mc0/ce_count          # correctable error count
for this mem controller
/sys/devices/system/edac/mc/mc0/ce_noinfo_count
/sys/devices/system/edac/mc/mc0/mc_name
/sys/devices/system/edac/mc/mc0/reset_counters
/sys/devices/system/edac/mc/mc0/sdram_scrub_rate
/sys/devices/system/edac/mc/mc0/seconds_since_reset
/sys/devices/system/edac/mc/mc0/size_mb
/sys/devices/system/edac/mc/mc0/ue_count          #un-correctable errors count
for this mem controller
/sys/devices/system/edac/mc/mc0/ue_noinfo_count
```

5.1.1 Get the Memory Controller MCx Device Information

EDAC driver messages help identify the DIMMs.

```
# dmesg | grep -i edac
[ 1.269512] EDAC MC: Ver: 3.0.0
[ 10.945243] EDAC amd64: F19h detected (node 0).
[ 10.945492] EDAC MC: UMC0 chip selects:
[ 10.945494] EDAC amd64: MC: 0:      OMB 1:      OMB
[ 10.945495] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945498] EDAC MC: UMC1 chip selects:
[ 10.945499] EDAC amd64: MC: 0: 16384MB 1:      OMB
[ 10.945499] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945502] EDAC MC: UMC2 chip selects:
[ 10.945503] EDAC amd64: MC: 0: 16384MB 1:      OMB
[ 10.945503] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945506] EDAC MC: UMC3 chip selects:
[ 10.945507] EDAC amd64: MC: 0:      OMB 1:      OMB
[ 10.945508] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945511] EDAC MC: UMC4 chip selects:
[ 10.945511] EDAC amd64: MC: 0: 16384MB 1:      OMB
[ 10.945512] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945515] EDAC MC: UMC5 chip selects:
[ 10.945516] EDAC amd64: MC: 0:      OMB 1:      OMB
[ 10.945516] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945519] EDAC MC: UMC6 chip selects:
[ 10.945520] EDAC amd64: MC: 0:      OMB 1:      OMB
[ 10.945521] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945524] EDAC MC: UMC7 chip selects:
[ 10.945524] EDAC amd64: MC: 0: 16384MB 1:      OMB
[ 10.945525] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945525] EDAC amd64: using x16 syndromes.
[ 10.945542] EDAC amd64: Node 0: DRAM ECC enabled.
[ 10.945542] EDAC amd64: MCT channel count: 4
[ 10.945663] EDAC MC0: Giving out device to module amd64_edac controller
F19h: DEV 0000:00:18.3 (INTERRUPT)
[ 10.945665] EDAC amd64: F19h detected (node 1).
[ 10.945885] EDAC MC: UMC0 chip selects:
[ 10.945886] EDAC amd64: MC: 0:      OMB 1:      OMB
[ 10.945887] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945890] EDAC MC: UMC1 chip selects:
[ 10.945891] EDAC amd64: MC: 0: 16384MB 1:      OMB
[ 10.945891] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945895] EDAC MC: UMC2 chip selects:
[ 10.945895] EDAC amd64: MC: 0: 16384MB 1:      OMB
[ 10.945896] EDAC amd64: MC: 2:      OMB 3:      OMB
[ 10.945899] EDAC MC: UMC3 chip selects:
```

```
[ 10.945900] EDAC amd64: MC: 0:    OMB 1:    OMB
[ 10.945900] EDAC amd64: MC: 2:    OMB 3:    OMB
[ 10.945904] EDAC MC: UMC4 chip selects:
[ 10.945904] EDAC amd64: MC: 0: 16384MB 1:    OMB
[ 10.945905] EDAC amd64: MC: 2:    OMB 3:    OMB
[ 10.945908] EDAC MC: UMC5 chip selects:
[ 10.945909] EDAC amd64: MC: 0:    OMB 1:    OMB
[ 10.945910] EDAC amd64: MC: 2:    OMB 3:    OMB
[ 10.945913] EDAC MC: UMC6 chip selects:
[ 10.945913] EDAC amd64: MC: 0:    OMB 1:    OMB
[ 10.945914] EDAC amd64: MC: 2:    OMB 3:    OMB
[ 10.945917] EDAC MC: UMC7 chip selects:
[ 10.945918] EDAC amd64: MC: 0: 16384MB 1:    OMB
[ 10.945919] EDAC amd64: MC: 2:    OMB 3:    OMB
[ 10.945919] EDAC amd64: using x16 syndromes.
[ 10.945937] EDAC amd64: Node 1: DRAM ECC enabled.
[ 10.945938] EDAC amd64: MCT channel count: 4
[ 10.950756] EDAC MC1: Giving out device to module amd64_edac controller
F19h: DEV 0000:00:19.3 (INTERRUPT)
[ 10.950796] EDAC PCI0: Giving out device to module amd64_edac controller
EDAC PCI controller: DEV 0000:00:18.0 (POLLED)
[ 10.950797] AMD64 EDAC driver v3.5.0
```

5.2 Error Injection

The `mce-inject` tool provides valuable ECC diagnostics and debugging by simulating machine check errors. See [How to Install the mce-inject Tool](#)* for additional information. To use `mce-inject`:

1. Load the kernel module.

```
#modprobe mce-inject
[774741.462842] Machine check injector initialized
```

2. Create sample files that simulate the desired errors. `mce-inject` uses these sample files to simulate the errors.

```
#cat correct1
CPU 1 BANK 2
STATUS corrected
RIP 0xa3450
```

3. Run the tool.

```
# ./mce-inject correct1

Message from syslogd@localhost at Nov 29 22:54:15 ... kernel:[Hardware Error]: Corrected
error, no action required.

Message from syslogd@localhost at Nov 29 22:54:15 ...
kernel:[Hardware Error]: CPU:1 (19:1:0) MC2_STATUS[-|CE|-|-|-|-|-|-]: 0x9000000000000000

Message from syslogd@localhost at Nov 29 22:54:15 ... kernel:[Hardware Error]: IPID:
0x0000000000000000

Message from syslogd@localhost at Nov 29 22:54:15 ... kernel:[Hardware Error]: L2 Cache
Extended Error Code: 0

Message from syslogd@localhost at Nov 29 22:54:15 ...
kernel:[Hardware Error]: L2 Cache Error: L2M tag multi-way-hit error.
```

```
Message from syslogd@localhost at Nov 29 22:54:15 ... kernel:[Hardware Error]: cache level:
RESV, tx: INSN
```

Messages can also contain information about DRAM ECC errors:

```
# cat correct2
CPU 1 BANK 1
STATUS corrected
RIP 0xa3450
```

```
# ./mce-inject correct2

Message from syslogd@localhost at Nov 29 22:57:59 ... kernel:[Hardware Error]: Corrected
error, no action required.

Message from syslogd@localhost at Nov 29 22:57:59 ...
kernel:[Hardware Error]: CPU:1 (19:1:0) MC1_STATUS[-|CE|-|-|-|-|-|-|]: 0x9000000000000000

Message from syslogd@localhost at Nov 29 22:57:59 ... kernel:[Hardware Error]: IPID:
0x0000000000000000

Message from syslogd@localhost at Nov 29 22:57:59 ...
kernel:[Hardware Error]: Instruction Fetch Unit ExtendedError Code: 0

Message from syslogd@localhost at Nov 29 22:57:59 ...
kernel:[Hardware Error]: Instruction Fetch Unit Error: microtag probe port parity error.

Message from syslogd@localhost at Nov 29 22:57:59 ... kernel:[Hardware Error]: cache level:
RESV, tx: INSN
```

Here is an example from an AMD EPYC 7373X processor:

```
# dmesg | grep -i edac
[ 0.676232] EDAC MC: Ver: 3.0.0
[ 6.300433] EDAC amd64: F19h detected (node 0).
[ 6.300605] EDAC amd64: Node 0: DRAM ECC enabled.
[ 6.300606] EDAC amd64: MCT channel count: 8
[ 6.412203] EDAC MC0: Giving out device to module amd64_edac controller F19h: DEV
0000:00:18.3 (INTERRUPT)
[ 6.412207] EDAC MC: UMC0 chip selects:
[ 6.412208] EDAC amd64: MC: 0: OMB 1: OMB
[ 6.412209] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.412211] EDAC MC: UMC1 chip selects:
[ 6.412212] EDAC amd64: MC: 0: OMB 1: OMB
[ 6.412212] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.412215] EDAC MC: UMC2 chip selects:
[ 6.412215] EDAC amd64: MC: 0: OMB 1: OMB
[ 6.412216] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.412218] EDAC MC: UMC3 chip selects:
[ 6.412218] EDAC amd64: MC: 0: OMB 1: OMB
[ 6.412219] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.412221] EDAC MC: UMC4 chip selects:
[ 6.412222] EDAC amd64: MC: 0: OMB 1: OMB
[ 6.412222] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.412225] EDAC MC: UMC5 chip selects:
[ 6.412225] EDAC amd64: MC: 0: OMB 1: OMB
[ 6.412226] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.412228] EDAC MC: UMC6 chip selects:
[ 6.412229] EDAC amd64: MC: 0: OMB 1: OMB
[ 6.412229] EDAC amd64: MC: 2: 32768MB 3: 32768MB
```

```

[ 6.412232] EDAC MC: UMC7 chip selects:
[ 6.412233] EDAC amd64: MC: 0:      0MB 1:      0MB
[ 6.412233] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.412234] EDAC amd64: using x16 syndromes.
[ 6.412235] EDAC amd64: F19h detected (node 1).
[ 6.412465] EDAC amd64: Node 1: DRAM ECC enabled.
[ 6.412466] EDAC amd64: MCT channel count: 7
[ 6.508844] EDAC MC1: Giving out device to module amd64_edac controller F19h: DEV
0000:00:19.3 (INTERRUPT)
[ 6.508848] EDAC MC: UMC0 chip selects:
[ 6.508849] EDAC amd64: MC: 0:      0MB 1:      0MB
[ 6.508850] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.508852] EDAC MC: UMC1 chip selects:
[ 6.508853] EDAC amd64: MC: 0:      0MB 1:      0MB
[ 6.508853] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.508856] EDAC MC: UMC2 chip selects:
[ 6.508857] EDAC amd64: MC: 0:      0MB 1:      0MB
[ 6.508857] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.508860] EDAC MC: UMC3 chip selects:
[ 6.508861] EDAC amd64: MC: 0:      0MB 1:      0MB
[ 6.508861] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.508864] EDAC MC: UMC4 chip selects:
[ 6.508864] EDAC amd64: MC: 0:      0MB 1:      0MB
[ 6.508865] EDAC amd64: MC: 2:      0MB 3:      0MB
[ 6.508868] EDAC MC: UMC5 chip selects:
[ 6.508868] EDAC amd64: MC: 0:      0MB 1:      0MB
[ 6.508869] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.508872] EDAC MC: UMC6 chip selects:
[ 6.508872] EDAC amd64: MC: 0:      0MB 1:      0MB
[ 6.508872] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.508877] EDAC MC: UMC7 chip selects:
[ 6.508878] EDAC amd64: MC: 0:      0MB 1:      0MB
[ 6.508878] EDAC amd64: MC: 2: 32768MB 3: 32768MB
[ 6.508878] EDAC amd64: using x16 syndromes.
[ 6.509997] EDAC PCI0: Giving out device to module amd64_edac controller EDAC PCI
controller: DEV 0000:00:18.0 (POLLED)
[ 6.509998] AMD64 EDAC driver v3.5.0

```

```

# cat correct1
CPU 1 BANK 2
STATUS corrected R
RIP 0xa3450

```

```

[root@liunxtunel791-os mce-inject]# cat correct2
CPU 1 BANK 1
STATUS corrected
RIP 0xa3450

```

```
# mce-inject ./correct1
```

```

Message from syslogd@liunxtunel791-os at Feb 11 21:48:19 ...
kernel:[Hardware Error]: Corrected error, no action required.

```

```

Message from syslogd@liunxtunel791-os at Feb 11 21:48:19 ...
kernel:[Hardware Error]: CPU:1 (19:1:2) MC2_STATUS[-|CE|-|-|-|-|-|-|-]:
0x9000000000000000

```

```

Message from syslogd@liunxtunel791-os at Feb 11 21:48:19 ...
kernel:[Hardware Error]: PPIN: 0x02b5fd03fabbc02c

```

```

Message from syslogd@liunxtunel791-os at Feb 11 21:48:19 ...
kernel:[Hardware Error]: IPID: 0x0000000000000000

```

```

Message from syslogd@liunxtune1791-os at Feb 11 21:48:19 ...
kernel:[Hardware Error]: L2 Cache Ext. Error Code: 0, L2M Tag Multiple-Way-Hit error.

Message from syslogd@liunxtune1791-os at Feb 11 21:48:19 ...
kernel:[Hardware Error]: cache level: RESV, tx: INSN

[root@liunxtune1791-os mce-inject]# mce-inject ./correct2

Message from syslogd@liunxtune1791-os at Feb 11 21:48:32 ...
kernel:[Hardware Error]: Corrected error, no action required.

Message from syslogd@liunxtune1791-os at Feb 11 21:48:32 ...
kernel:[Hardware Error]: CPU:1 (19:1:2) MC1_STATUS[-|CE|-|-|-|-|-|-|]:
0x9000000000000000

Message from syslogd@liunxtune1791-os at Feb 11 21:48:32 ...
kernel:[Hardware Error]: PPIN: 0x02b5fd03fabbc02c

Message from syslogd@liunxtune1791-os at Feb 11 21:48:32 ...
kernel:[Hardware Error]: IPID: 0x0000000000000000

Message from syslogd@liunxtune1791-os at Feb 11 21:48:32 ...
kernel:[Hardware Error]: Instruction Fetch Unit Ext. Error Code: 0, Op Cache Microtag
Probe Port Parity Error.

Message from syslogd@liunxtune1791-os at Feb 11 21:48:32 ...
kernel:[Hardware Error]: cache level: RESV, tx: INSN

# mce-inject test/corrected

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: Corrected error, no action required.

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: CPU:0 (19:1:2) MC1_STATUS[-|CE|-|AddrV|-|-|-|-|-|]:
0x9400000000000000

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: Error Addr: 0x000000000000abcd

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: PPIN: 0x02b5fd03fabbc02c

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: IPID: 0x0000000000000000

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: Instruction Fetch Unit Ext. Error Code: 0, Op Cache Microtag
Probe Port Parity Error.

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: cache level: RESV, tx: INSN

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: Corrected error, no action required.

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: CPU:1 (19:1:2) MC2_STATUS[-|CE|-|AddrV|-|-|-|-|-|]:
0x9400000000000000

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: Error Addr: 0x0000000000001234

```

```
Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: PPIN: 0x02b5fd03fabbc02c

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: IPID: 0x0000000000000000

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: L2 Cache Ext. Error Code: 0, L2M Tag Multiple-Way-Hit error.

Message from syslogd@liunxtune1791-os at Feb 11 21:46:48 ...
kernel:[Hardware Error]: cache level: RESV, tx: INSN
```

This page intentionally left blank.

Chapter

6

AMD 3D V-Cache

6.1 BIOS Settings

To enable AMD 3D-Cache in the BIOS:

1. Access your system BIOS.
2. Select the **Advanced** tab.

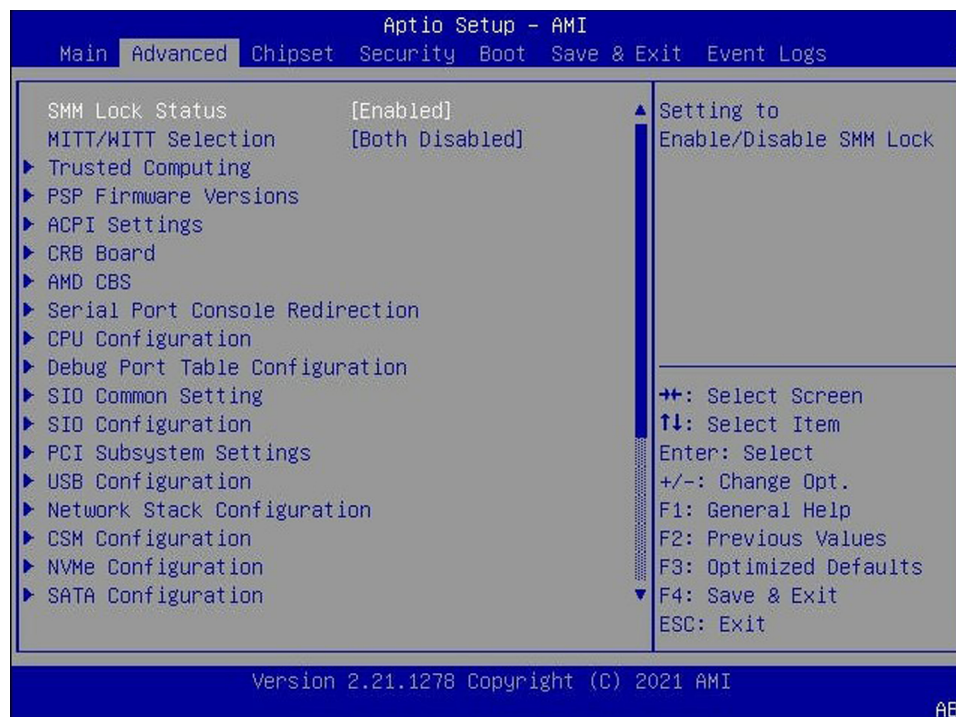


Figure 6-1: BIOS Advanced tab

3. Select **AMD CBS**.

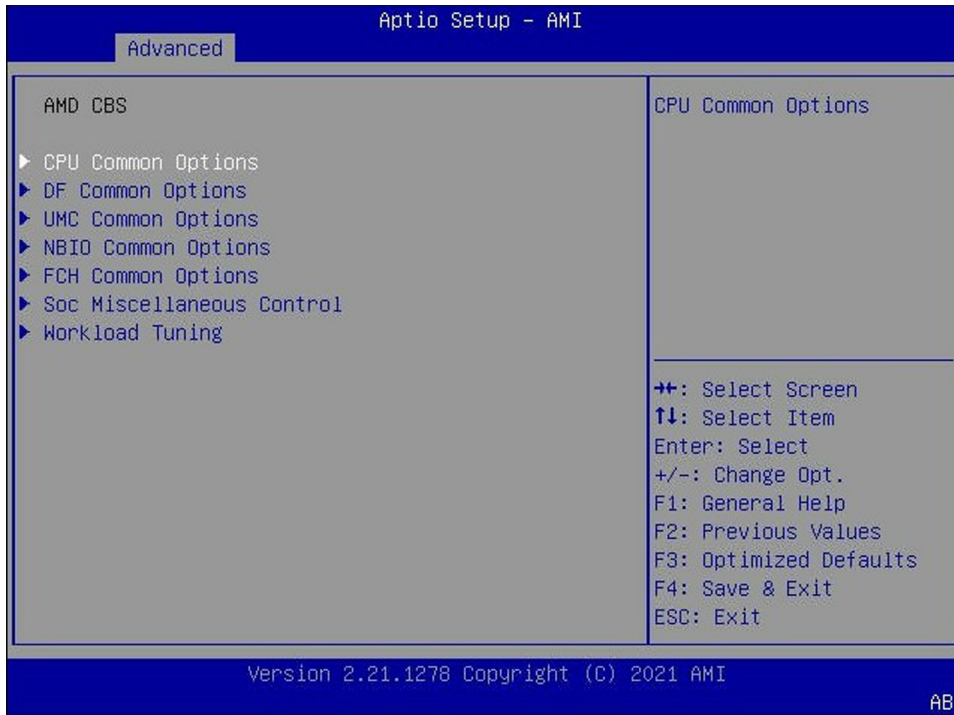


Figure 6-2: BIOS AMD CBS options

4. Select **CPU Common Options**, then select **3D V-Cache**, and then set it to **Auto**.

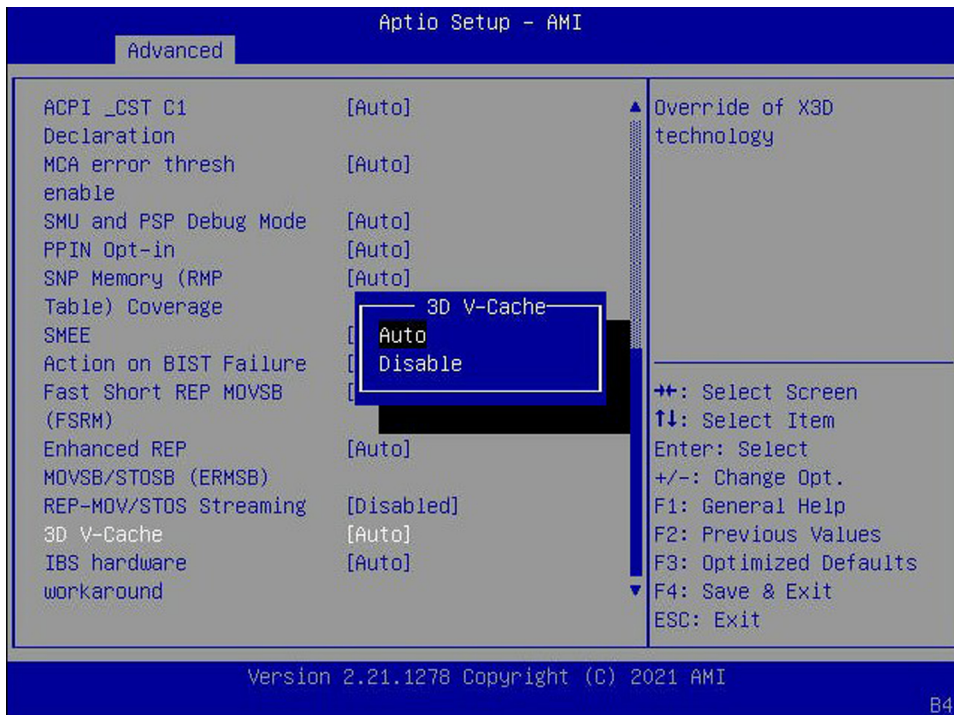


Figure 6-3: Setting 3D V-Cache to Auto

Note: Different OEM BIOS may have different navigation than described above.

6.2 Iscpu

6.2.1 3D V-cache Disabled

```
# lscpu
Architecture:          x86_64
CPU op-mode(s):       32-bit, 64-bit
Byte Order:           Little Endian
CPU(s):               64
On-line CPU(s) list:  0-63
Thread(s) per core:   2
Core(s) per socket:   16
Socket(s):            2
NUMA node(s):         2
Vendor ID:            AuthenticAMD
BIOS Vendor ID:      Advanced Micro Devices, Inc.
CPU family:           25
Model:                1
Model name:           AMD EPYC 7373X 16-Core Processor
BIOS Model name:     AMD EPYC 7373X 16-Core Processor
Stepping:             2
CPU MHz:              3050.000
CPU max MHz:          3830.3711
CPU min MHz:          1500.0000
BogoMIPS:             6088.94
Virtualization:       AMD-V
L1d cache:            32K
L1i cache:            32K
L2 cache:             512K
L3 cache:             32768K
NUMA node0 CPU(s):   0-15,32-47
NUMA node1 CPU(s):   16-31,48-63
Flags:                fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc
rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclmulqdq monitor ssse3 fma cx16
pcid sse4_1 sse4_2 movbe popcnt aes_xsave avx f16c rdrand lahf_lm cmp_legacy svm extapic
cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt tce topoext perfctr_core
perfctr_nb bpext perfctr_llc mwaitx cpb cat_l3 cdp_l3 invpcid_single hw_pstate ssbd mba
ibrs ibpb stibp vmmcall fsgsbase bmi1 avx2 smep bmi2 invpcid cqm rdt_a rdseed adx smap
clflushopt clwb sha_ni xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total
cqm_mbm_local clzero irperf xsaveerptr wbnoinvd amd_ppin arat npt lbrv svm_lock nrip_save
tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold v_vmsave_vmload
vgif v_spec_ctrl umip pku ospke vaes vpclmulqdq rdpid overflow_recov succor smca sme sev
sev_es
```

6.2.2 3D V-cache Enabled

```
# lscpu
Architecture:          x86_64
CPU op-mode(s):       32-bit, 64-bit
Byte Order:           Little Endian
CPU(s):               64
On-line CPU(s) list:  0-63
Thread(s) per core:   2
Core(s) per socket:   16
Socket(s):            2
NUMA node(s):         2
Vendor ID:            AuthenticAMD
```

```

BIOS Vendor ID:      Advanced Micro Devices, Inc.
CPU family:         25
Model:             1
Model name:        AMD EPYC 7373X 16-Core Processor
BIOS Model name:   AMD EPYC 7373X 16-Core Processor
Stepping:         2
CPU MHz:          3050.000
CPU max MHz:      3830.3711
CPU min MHz:      1500.0000
BogoMIPS:         6088.93
Virtualization:   AMD-V
L1d cache:       32K
L1i cache:       32K
L2 cache:        512K
L3 cache:        98304K
NUMA node0 CPU(s): 0-15,32-47
NUMA node1 CPU(s): 16-31,48-63
Flags:           fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpelgb rdtscp lm constant_tsc
rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclmulqdq monitor ssse3 fma cx16
pcid sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand lahf_lm cmp_legacy svm extapic
cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt tce topoext perfctr_core
perfctr_nb bpext perfctr_llc mwaitx cpb cat_l3 cdp_l3 invpcid_single hw_pstate ssbd mba
ibrs ibpb stibp vmmcall fsgsbase bmi1 avx2 smep bmi2 invpcid cqm rdt_a rdseed adx smap
clflushopt clwb sha_ni xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total
cqm_mbm_local clzero irperf xsaveerptr wbnoinvd amd_ppin arat npt lbrv svm_lock nrip_save
tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold v_vmsave_vmload
vgif v_spec_ctrl umip pku ospke vaes vpclmulqdq rdpid overflow_recov succor smca sme sev
sev_es

```

6.2.3 L3 Cache

In these examples, the L3 caches is:

- **With 3D V-Cache enabled:** 98304K
- **With 3D V-Cache disabled:** 32768K

6.3 lshw -C memory Output for L3

6.3.1 3D V-Cache Disabled

```

.....
.....
*-cache:2
  description: L3 cache
  physical id: 31
  slot: L3 - Cache
  size: 256MiB
  capacity: 256MiB
  clock: 1GHz (1.0ns)
  capabilities: pipeline-burst internal write-back unified
  configuration: level=3
.....
.....
*-cache:5
  description: L3 cache
  physical id: 5d
  slot: L3 - Cache

```

```

size: 256MiB
capacity: 256MiB
clock: 1GHz (1.0ns)
capabilities: pipeline-burst internal write-back unified
configuration: level=3

```

6.3.2 3D V-Cache Enabled

```

.....
.....
*-cache:2
  description: L3 cache
  physical id: 31
  slot: L3 - Cache
  size: 768MiB
  capacity: 768MiB
  clock: 1GHz (1.0ns)
  capabilities: pipeline-burst internal write-back unified
  configuration: level=3
.....
.....
*-cache:5
  description: L3 cache
  physical id: 5d
  slot: L3 - Cache
  size: 768MiB
  capacity: 768MiB
  clock: 1GHz (1.0ns)
  capabilities: pipeline-burst internal write-back unified
  configuration: level=3

```

6.3.2.1 L3 Cache

In these examples, the L3 caches is:

- **With 3D V-Cache enabled:** 98304K
- **With 3D V-Cache disabled:** 32768K

6.4 Cache Information using valgrind

`cachegrind` simulates how your program interacts with a machine's cache hierarchy and (optionally) branch predictor. It simulates a machine with independent first-level instruction and data caches (I1 and D1), backed by a unified second-level cache (L2). This exactly matches the configuration of many modern machines.

6.4.1 3D V-Cache Disabled

```

# valgrind --tool=cachegrind nginx

==8149== Cachegrind, a cache and branch-prediction profiler
==8149== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==8149== Using Valgrind-3.17.0 and LibVEX; rerun with -h for copyright info
==8149== Command: nginx
==8149==
--8149-- warning: L3 cache found, using its data for the LL simulation.
==8149==
==8149== I   refs:      35,543,031
==8149== I1  misses:    450,983

```

```

==8149== LLi misses:          12,380
==8149== I1 miss rate:       1.27%
==8149== LLi miss rate:     0.03%
==8149==
==8149== D   refs:          14,454,381 (9,634,307 rd + 4,820,074 wr)
==8149== D1 misses:         355,112 ( 307,266 rd + 47,846 wr)
==8149== LLd misses:        55,037 ( 28,447 rd + 26,590 wr)
==8149== D1 miss rate:      2.5% ( 3.2% + 1.0% )
==8149== LLd miss rate:    0.4% ( 0.3% + 0.6% )
==8149==
==8149== LL refs:           806,095 ( 758,249 rd + 47,846 wr)
==8149== LL misses:        67,417 ( 40,827 rd + 26,590 wr)
==8149== LL miss rate:     0.1% ( 0.1% + 0.6% )

```

6.4.2 3D V-Cache Enabled

```

# valgrind --tool=cachegrind nginx

==4976== Cachegrind, a cache and branch-prediction profiler
==4976== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==4976== Using Valgrind-3.17.0 and LibVEX; rerun with -h for copyright info
==4976== Command: nginx
==4976==
--4976-- warning: L3 cache found, using its data for the LL simulation.
--4976-- warning: specified LL cache: line_size 64  assoc 1  total_size 805,306,368
--4976-- warning: simulated LL cache: line_size 64  assoc 2  total_size 1,073,741,824
==4976==
==4976== I   refs:          35,538,920
==4976== I1 misses:         450,941
==4976== LLi misses:        12,319
==4976== I1 miss rate:      1.27%
==4976== LLi miss rate:     0.03%
==4976==
==4976== D   refs:          14,452,405 (9,633,011 rd + 4,819,394 wr)
==4976== D1 misses:         354,928 ( 307,077 rd + 47,851 wr)
==4976== LLd misses:        55,042 ( 28,452 rd + 26,590 wr)
==4976== D1 miss rate:      2.5% ( 3.2% + 1.0% )
==4976== LLd miss rate:    0.4% ( 0.3% + 0.6% )
==4976==
==4976== LL refs:           805,869 ( 758,018 rd + 47,851 wr)
==4976== LL misses:        67,361 ( 40,771 rd + 26,590 wr)
==4976== LL miss rate:     0.1% ( 0.1% + 0.6% )

```

Chapter

7

SPECpower and STREAM

The following SPECpower_ssj[®]2008 result represents a single measurement across a random pair of AMD EPYC 7773X samples on the Ethanol-X reference platform running Microsoft[®] Windows[®] Server 2019 and JDK-11.0.4, leveraging a Yokogawa WT310e power analyzer connected to a controller PC (server) via Serial-to-USB cable. These results are presented for illustrative purposes only. No performance claim is intended or implied; your results may vary significantly from those shown below.

Benchmark Results Summary					
Performance			Power	Performance to Power Ratio	
Target Load	Actual Load	ssj_ops	Average Active Power (W)		
100%	99.6%	12,514,926	517	24,208	
90%	88.7%	11,155,957	475	23,463	
80%	79.8%	10,035,417	432	23,219	
70%	69.7%	8,755,382	398	21,983	
60%	60.0%	7,542,115	347	21,705	
50%	50.0%	6,287,074	310	20,295	
40%	40.0%	5,026,105	280	17,923	
30%	30.0%	3,771,561	252	14,940	
20%	20.0%	2,515,272	226	11,116	
10%	10.0%	1,257,702	197	6,387	
	Active Idle	0	89.0	0	
sum of ssj_ops /			sum of power =		19,354

Please refer to the following sections in [AMD Family 19h Models 00h-0Fh SP3 Power and Performance Optimization Guide](#) (login required):

- Appendix B: Optimizations and Tuning
- Appendix D: AMD EPYC 7003 Series Processors with AMD 3D V-Cache
- Appendix D.2: Optimizations and Tuning Summary for 3D V-Cache Models
- Appendix I: Data Summary Tables for AMD EPYC 7003 Processors with 3D V-Cache

7.1 STREAM using Spack

STREAM tests the maximum memory bandwidth of a core or entire CPU. Please see the *High Performance Computing (HPC) Tuning Guide for AMD EPYC™ 7003 Series Processors* (available from [AMD EPYC Tuning Guides](#)) for instructions on how to build the STREAM benchmark using Spack.

This page intentionally left blank.

Chapter

8

Resources

- [Memory Population Guidelines for AMD EPYC 7003 Series Processors](#)– Login required.
- [Socket SP3 Platform NUMA Topology for AMD Family 19h Models 00h–0Fh](#) - Log in required.
- [Add support for Large Increment per Cycle Events](#)*
- [Red Hat Enterprise Linux 8 - Configuring and Managing Virtualization](#)*
- [Red Hat Enterprise Linux 8.3 Release Notes](#)*
- [Red Hat Enterprise Linux 7: Optimizing Memory System Performance](#)*
- [Enabling AMD Secure Encrypted Virtualization in RHEL 8](#)*
- [AMD Family 19h Models 00h–0Fh SP3 Power and Performance Optimization Guide \(login required\)](#)
- *High Performance Computing (HPC) Tuning Guide for AMD EPYC™ 7003 Series Processors* (available from [AMD EPYC Tuning Guides](#))

This page intentionally left blank.