

# TUNING GUIDE AMD EPYC 7003

## Couchbase

|             |           |
|-------------|-----------|
| Publication | 57071     |
| Revision    | 3.0       |
| Issue Date  | Mar, 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. Couchbase is a registered trademark of Couchbase, Inc. Other product names and links to external sites used in this publication are for identification purposes only and may be trademarks of their respective companies.

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

| Date      | Version | Changes                                |
|-----------|---------|--|
| Mar, 2021 | 2.0     | Initial public release                 |
| Mar, 2022 | 3.0     | Added AMD EPYC 3D V-Cache™ information |
|           |         |  |
|           |         |  |

**Audience**

This tuning guide describes best practices for optimizing performance using Couchbase®. It is intended for a technical audience such as Couchbase application architects, production deployment, and performance engineering teams with:

- A background in configuring servers.
- Administrator-level access to both the server management Interface (BMC) and the OS.
- Familiarity with both the BMC and OS-specific configuration, monitoring, and troubleshooting tools.

**Authors**

Gnanakumar Rajaram and Asheet Hakoo.

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

# Table of Contents

|                  |   |           |
|------------------|---|-----------|
| <b>Chapter 1</b> | <b>Introduction</b> .....   | <b>1</b>  |
| 1.1              | AMD EPYC 7003 Series Processors .....                                   | 1         |
| 1.2              | Operating Systems .....   | 1         |
| 1.3              | Couchbase® .....  | 2         |
| 1.3.1            | General Couchbase Tuning Guidelines .....                               | 2         |
| 1.3.2            | Licensing Costs .....   | 2         |
| 1.3.3            | Recommended CPUs for Couchbase Servers .....                            | 3         |
| 1.3.4            | Recommended System Resources .....                                      | 3         |
| <b>Chapter 2</b> | <b>Hardware Configuration Best Practices</b> .....                      | <b>5</b>  |
| 2.1              | Memory Configuration .....  | 5         |
| 2.2              | BIOS Settings .....   | 5         |
| 2.2.1            | BIOS Settings for Maximizing Performance .....                          | 6         |
| <b>Chapter 3</b> | <b>Linux Optimizations</b> .....  | <b>9</b>  |
| 3.1              | Memory Subsystem .....  | 9         |
| 3.1.1            | Swap .....  | 9         |
| 3.1.2            | Transparent Huge Pages (THP) .....                                      | 9         |
| 3.2              | Storage Subsystem .....   | 10        |
| 3.2.1            | Filesystem Mount Options .....  | 10        |
| 3.3              | Network Subsystem .....   | 12        |
| 3.4              | tuned-adm Profile .....   | 12        |
| 3.5              | NTP Configuration using Chrony .....                                    | 13        |
| 3.6              | Example RHEL Server Configuration Files .....                           | 13        |
| 3.6.1            | /etc/default/grub .....   | 13        |
| 3.6.2            | /etc/rc.local .....   | 13        |
| 3.6.3            | /etc/sysctl.conf .....  | 13        |
| 3.6.4            | /etc/security/limits.conf .....   | 14        |
| <b>Chapter 4</b> | <b>Couchbase Settings</b> .....   | <b>15</b> |
| 4.1              | Replication .....   | 15        |
| 4.2              | Disk Priority .....   | 15        |
| 4.3              | Data Ejection Watermarks .....  | 15        |
| 4.4              | Memory Optimized (MOI) vs Standard Global Secondary Indexes (GSI) ..... | 15        |
| 4.5              | Allocating Higher CPU Resources for Index Service .....                 | 16        |
| 4.6              | Fragmentation and Compaction .....                                      | 16        |
| 4.7              | REST Endpoints Available to View Settings .....                         | 16        |



**Chapter 5 Resources** ----- **17**

**Chapter 6 Glossary** ----- **19**

## Chapter

## 1

## Introduction

## 1.1 AMD EPYC 7003 Series Processors

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

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

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

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

Some AMD EPYC™ 7003 Series Processors introduce AMD’s new 3D V-Cache die stacking technology that enables denser, more efficient chiplet integration. AMD 3D Chiplet architecture stacks L3 cache tiles vertically to provide 768 MB of L3 cache per socket 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

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

## 1.3 Couchbase®

Couchbase® Server is an open source, distributed, JSON document database. It exposes a scale-out, key-value store with managed cache for sub-millisecond data operations, purpose-built indexers for efficient queries, and a powerful query engine for executing SQL-like queries. Couchbase implements a “memory- first architecture” that provide low-latency, large-scale data management for consistent high performance, availability, and scalability for enterprise web, mobile, and IoT applications.

Couchbase is straightforward to deploy and manage. Each Couchbase Server node consists of identical software, which simplifies automation. A single administrator console manages the entire cluster and offers single-click cluster expansion and rebalancing. Couchbase Server replicates data across multiple nodes to support failover. It also provides a comprehensive management UI to visualize, monitor, and manage both individual nodes in the cluster and overall cluster status and statistics.

### 1.3.1 General Couchbase Tuning Guidelines

Out-of-the-box Couchbase is very fast because it buffers significant I/O in memory and auto-tunes itself by allocating threads based on the number of CPU cores or the amount of available disk space. Couchbase manages memory and disk resources and prefers less OS interference with those tasks. Even so, the Linux kernel contains several tunable sysctl parameters at multiple layers for I/O, CPU, memory, storage, and networking that can help optimize Couchbase Server performance.

### 1.3.2 Licensing Costs

Applications are generally licensed by either the number of processor sockets or by the total core count. To reduce both licensing costs and Total Cost of Ownership (TCO):

- Core-based application licenses benefit from low-core, high frequency processors.
- Socket-based application licenses benefit from high-core processors.

See [Couchbase Pricing](#)\* for additional information.

The size of the Couchbase Server cluster is crucial for overall stability and performance. Sizing is beyond the scope of this tuning guide. Make sure to evaluate the overall performance and capacity requirements for your workload and dataset, and then divide that into your available hardware and resources. Couchbase performs best when the majority of reads to come out of the cache and from having enough I/O capacity to handle the writes. Please see the [sizing guidelines](#) for help determining the best system specifications for your Couchbase Server deployment.

### 1.3.3 Recommended CPUs for Couchbase Servers

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

| System Type                     | Cores | CPUs <sup>1</sup>                        | Memory          | Storage                    | Network  | Min. Node #                |
|---------------------------------|-------|--|-----------------|----------------------------|----------|----------------------------|
| Small                           | 16    | 1 x AMD EPYC™ 7313P                      | 128 GB          | 1-4 SAS SSDs               | 10 Gbit  | 3                          |
| Medium                          | 24    | 1 x AMD EPYC™ 7443P                      | 128 GB / 256 GB | 1-4 SAS SSDs / NVMe SSDs   | 25 Gbit  | 3                          |
| Large                           | 32    | 1 x AMD EPYC™ 7543                       | 256 GB / 512 GB | 1-4 NVMe SSDs              | 25 Gbit  | 3                          |
| Cloud <sup>2</sup> : PaaS, IaaS | 2x64  | 2 x AMD EPYC™ 7763<br>2 x AMD EPYC™ 7713 | 1 TB            | Size based on requirements | 100 Gbit | Size based on requirements |

1 - Please see [AMD EPYC 7003 Series Processors](#) for more information.  
2 - Recommendations for Cloud Service Providers (CSP) / private cloud for hypervisor configuration.

Table 1-2: Recommended AMD EPYC processors for Couchbase Server

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

### 1.3.4 Recommended System Resources

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

| Operating System | Recommended Specifications*  |
|------------------|--|
| CPU**            | <ul style="list-style-type: none"> <li>Quad-core x86_64 and above.</li> <li>Six-core x86_64 when using Cross Datacenter Replication (XDCR) and Views.</li> </ul> |
| RAM*             | 16 GB (physical) and above.  |
| Storage          | 16 GB and above (block-based; HDD, SSD, EBS, iSCSI)<br>Network filesystems such as CIFS and NFS are not supported.   |

\*The Recommended Specifications do not consider your intended workload. Follow the [Couchbase System Resource Requirements](#)\* when determining system specifications for your Couchbase Server deployment.  
\*\*Virtual or cloud environments use the smaller minimum CPU and RAM requirements.

Table 1-3: Recommended Couchbase Server system resource requirements

*This page intentionally left blank.*

# Hardware Configuration Best Practices

## 2.1 Memory Configuration

Proper memory subsystem configuration is crucial for optimum performance. I/O transfers data into or out of memory, so I/O bandwidth can never exceed memory subsystem capabilities. Modern CPUs require populating at least one DIMM for every DDR channel for maximum memory bandwidth. Systems powered by AMD EPYC 7003 Series Processors include eight DDR4 memory channels on each CPU socket. Thus:

- A single-socket system requires populating all eight memory channels with 3200 MHz DIMMs.
- A dual-socket system requires populating 16 memory channels with 2933 MHz DIMMs.

AMD recommends populating all eight memory channels per socket with every channel having the same capacity and speed of DIMMs for optimal performance. See [Memory Population Guidelines for AMD EPYC 7003 Series Processors](#) for additional guidance.

OEM servers support either:

- Both 2nd Gen and 3rd Gen AMD EPYC processors.
- 3rd Gen AMD EPYC processors only.

Contact your OEM vendor for information about the CPUs supported by your servers.

## 2.2 BIOS Settings

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

Table 4 describes the BIOS options that most impact Couchbase Server performance using the BIOS parameters and settings found on AMD Customer Reference Boards (CRB), and OEM settings may vary. Please see your OEM BIOS documentation for platform-specific BIOS information. Please also see the latest version of *Workload Tuning Guide for AMD EPYC™ 7003 Series Processors* (available from [AMD EPYC Tuning Guides](#)) for additional BIOS tuning information.

## 2.2.1 BIOS Settings for Maximizing Performance

| Name                              | Value   | Description  |
|-----------------------------------|---------|--|
| SMT Control                       | Enabled | Enabling Symmetric Multithreading (SMT) allows two hardware threads per core.<br>You must enable AMD x2APIC with support more than 255 threads if you are using a system with dual 64-core AMD EPYC 7003 Series Processors. If you are running dual 64-core processors and your OS does not support AMD x2APIC, then you must disable SMT.   |
| NUMA Node per Socket (NPS)        | NPS1    | This setting enables a tradeoff between minimizing local memory latency for NUMA-aware or highly parallelizable workloads vs. maximizing per-core memory bandwidth for non-NUMA-friendly workloads by determining the number of NUMA nodes to split the memory channels between. Higher settings reduce memory channels per NUMA node, which lowers both throughput and latency for that NUMA node.  |
| IOMMU                             | Enabled | Enabling IOMMU allows devices (such as the AMD EPYC processor-integrated SATA controller) to present separate IRQs for each attached device instead of one IRQ for the subsystem.<br>Enabling IOMMU also allows operating systems to provide additional protection for DMA capable I/O devices. If you believe IOMMU is impeding performance, then enable it in BIOS and disable it via OS options, such as <code>iommu=pt</code> on the Linux kernel command line.<br>Add <code>iommu=pt</code> (passthrough) to the grub file for optimal performance. In passthrough mode, the adapters need not use DMA translation to the memory, which improves performance. |
| Determinism Control               | Manual  | Enables the <b>Determinism Slider</b> control.   |
| Determinism Slider                | Power   | Ensures maximum performance for each CPU in a large population of identically-configured CPUs by only throttling CPUs when they reach the same cTDP. See <a href="#">Power/Performance Determinism</a> for more information.   |
| cTDP Control                      | Manual  | Setting <b>Configurable Thermal Design Power</b> (cTDP) to <b>Manual</b> allows you to modify the platform CPU cooling limit.  |
| cTDP                              | OPN Max | Set TDP in watts.  |
| Package Power Limit Control       | Manual  | Setting <b>Package Power Limit</b> (PPL) to <b>Manual</b> allows you to modify the CPU Power Dissipation Limit.  |
| Package Power Limit               | OPN Max | Set PPL in watts.  |
| ACPI SRAT L3 Cache as NUMA Domain | Disable | Controls automatic or manual generation of distance information in the ACPI System Locality Information Table (SLIT) and NUMA proximity domains in the System Resource Affinity Table (SRAT). Disabling this option disables reporting each L3 cache as a NUMA domain to the OS.   |

Table 2-1: Recommended BIOS settings

|     |         |   |
|-----|---------|---|
| X3D | Enabled | <p>On an AMD EPYC 7003 processor with AMD 3D V-Cache technology, enabling this option enables the AMD 3D V-Cache module in the CCD to increase the total size of the L3 cache memory to 96GB. Disabling this option reduces the L3 cache in the CCD to 32MB.</p> <p>This option is only available on AMD EPYC 7003 Series Processors with AMD 3D V-Cache.</p> |
|-----|---------|---|

*Table 2-1: Recommended BIOS settings (Continued)*

*This page intentionally left blank.*

## Chapter

## 3

## Linux Optimizations

Couchbase performance is very sensitive to cache hit ratios because data passes through a caching layer. The [Couchbase documentation](#)\* suggests that the cache miss ratio is a key performance monitoring variable. Tuning the Linux OS memory and storage configuration optimizes Couchbase performance.

## 3.1 Memory Subsystem

Providing sufficient memory for Couchbase based on the sizing guidelines for the Couchbase cluster allows Couchbase to move data between memory and disk to keep from running out of memory.

### 3.1.1 Swap

Swapping to disk hurts Couchbase performance. Set `vm.swappiness` to 1 to reduce the likelihood of swapping as much as possible by adding the following lines to `/etc/sysctl.conf`:

```
vm.swappiness=1
vm.zone_reclaim_mode=0
fs.file-max = 500000
```

### 3.1.2 Transparent Huge Pages (THP)

Transparent huge pages are enabled by default in most Linux operating systems. This can delay allocating new system memory because of reshuffling pages in the background into large pages. Couchbase recommends disabling THP. Use the following `init` script:

```
# cat disable-thp #
#!/bin/bash
### BEGIN INIT INFO
# Provides:          disable-thp
# Required-Start:    $local_fs
# Required-Stop:
# X-Start-Before:    couchbase-server
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Disable THP
# Description:       disables Transparent Huge Pages (THP) on boot
### END INIT INFO
case $1 in start)
if [ -d /sys/kernel/mm/transparent_hugepage ]; then
echo 'never' > /sys/kernel/mm/transparent_hugepage/enabled echo 'never' > /sys/kernel/mm/
transparent_hugepage/defrag
elif [ -d /sys/kernel/mm/redhat_transparent_hugepage ]; then
echo 'never' > /sys/kernel/mm/redhat_transparent_hugepage/enabled echo 'never' > /sys/
kernel/mm/redhat_transparent_hugepage/defrag
else
return 0
fi
;;
```

```
esac
```

```
# cp disable-thp /etc/init.d/disable-thp # chmod 755 /etc/init.d/disable-thp
# service disable-thp start # chkconfig disable-thp on
```

Couchbase does not currently implement any NUMA related optimization and recommends either disabling NUMA in BIOS or for Couchbase.

To enable NUMA interleaving, modify the Couchbase-server startup script by prepending the daemon variable with `numactl -interleave all`, as shown below. When starting it with `numactl --interleave all`, Couchbase will run with its memory interleaved on all NUMA nodes in round-robin.

```
# yum install numactl
# cat /usr/lib/systemd/system/couchbase-server.service
...
...
[Service]
SyslogIdentifier=couchbase
User=couchbase
Type=simple
WorkingDirectory=/opt/couchbase/var/lib/couchbase
LimitNOFILE=70000
LimitMEMLOCK=infinity
ExecStart=numactl -interleave all /opt/couchbase/bin/couchbase-server -- -
noinput
ExecStop=/opt/couchbase/bin/couchbase-server -k
...
...
```

## 3.2 Storage Subsystem

AMD EPYC 7003 Series Processors support PCIe 4.0 connections that offer twice the I/O bandwidth of PCIe 3.0. Use SSD for Couchbase storage. If you cannot do this, then use high-RPM local storage. If you cannot use SSD for everything, then leverage Couchbase’s ability to segregate data storage from index storage by deploying the data on the HDDs and the indexes on the SSDs.

*Note: Provisioning separate data and index disks generally yields better results.*

### 3.2.1 Filesystem Mount Options

Most Linux system use the virtual ext4 filesystem by default, but xfs is better for Couchbase because it offers slightly better performance for append-only workloads.

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

Table 3-1: XFS file system mount options

Use the following ext4 and xfs filesystem mount options in `/etc/fstab`:

```
/dev/nvme0n1p1    /datadir    xfs    noatime,nobarrier    0 0
/dev/nvme0n1p2    /indexdir   ext4
```

Couchbase takes a single directory name for storing data and indexes. Hardware configurations with multiple drives can use RAID0 striping via Multiple Disk and Device Administration `mdadm(1)` or other tools to improve storage performance.

The following settings allow you to tune the amount of dirty memory and frequency (interval) at which the background kernel flusher threads will start writeback into the disks.

To set the dirty bytes limit in the page cache:

```
sysctl -w vm.dirty_background_bytes=104857600
sysctl -w vm.dirty_bytes=209715200
```

To set the max time for dirty pages:

```
# sysctl -w vm.dirty_writeback_centisecs=100
# sysctl -w vm.dirty_expire_centisecs=200
```

To tune VFS cache reclaim:

```
# sysctl -w vm.vfs_cache_pressure=50
```

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

- `deadline scheduler` is essentially a FIFO but does basic reordering and merging within the scheduler queue and guarantees a maximum latency for any given write in its queue.
- `noop` is a less-frequently-used option that works for Couchbase.

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

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

```
# echo 1024 > /sys/block/<dev>/queue/nr_requests
```

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

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

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

Set the file descriptor and core file size limits in `/etc/security/limits.conf`:

```
couchbase hard nfile 40960
couchbase hard core unlimited
couchbase soft memlock unlimited
couchbase hard memlock unlimited
```

### 3.3 Network Subsystem

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

See the latest version of *Linux® Network Tuning Guide for AMD EPYC™ 7003 Series Processors* (available from [AMD EPYC Tuning Guides](#)) for information on configuring the network for systems in a Couchbase cluster and follow the guidelines therein to:

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

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

If you are using Windows, then please see the latest version of *Windows® Network Tuning Guide for AMD EPYC™ 7003 Series Processors* (available from [AMD EPYC Tuning Guides](#)) for network tuning information.

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

### 3.4 tuned-adm Profile

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

```
# yum install tuned -y
# systemctl start tuned
# tuned-adm profile throughput-performance
```

```
# systemctl enable tuned
```

## 3.5 NTP Configuration using Chrony

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

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

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

## 3.6 Example RHEL Server Configuration Files

### 3.6.1 /etc/default/grub

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

### 3.6.2 /etc/rc.local

```
# cat /etc/rc.local
cpupower -c all idle-set -d 2 ethtool -G p2p1 rx 4096 tx 4096
/usr/sbin/set_irq_affinity_cpulist.sh 1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61 p2p1
echo "deadline" > /sys/block/nvme0n1/queue/scheduler echo 1024 > /sys/block/nvme0n1/queue/
nr_requests
```

### 3.6.3 /etc/sysctl.conf

```
# cat /etc/sysctl.conf
vm.zone_reclaim_mode=0
vm.swappiness=1
vm.dirty_background_bytes=209715200
vm.dirty_bytes=104857600
vm.dirty_writeback_centisecs=100
vm.dirty_expire_centisecs=200
vm.vfs_cache_pressure=50
vm.overcommit_memory=0
```

```
...
...
```

```
net.core.somaxconn=16384
net.core.rmem_max=33554432
net.core.wmem_max=33554432
net.core.rmem_default=33554432
net.core.wmem_default=33554432
net.core.optmem_max=33554432
net.core.netdev_max_backlog=250000
net.ipv4.tcp_rmem=4096 87380 33554432
net.ipv4.tcp_wmem=4096 65536 33554432
net.ipv4.tcp_timestamps=0
net.ipv4.tcp_sack=1
net.ipv4.tcp_low_latency=1
net.ipv4.tcp_adv_win_scale=1
net.ipv4.tcp_retries2=5
net.ipv4.conf.all.arp_filter=1
```

### 3.6.4 /etc/security/limits.conf

```
# cat /etc/security/limits.conf
...
...
couchbase hard nofile 40960
couchbase hard core unlimited
couchbase soft memlock unlimited
couchbase hard memlock unlimited
```

## Chapter

## 4

# Couchbase Settings

Allocating enough memory to Couchbase prevents disk I/O bottlenecks, making this the single most important parameter for tuning Couchbase.

## 4.1 Replication

Couchbase distributes data replication throughout the cluster to prevent a single point of failure. You can configure data replication on a bucket-level and node-basis. Couchbase supports up to 3 replicas, which means up to 4 copies of the data (three backups and one set of active data). This is different than Hadoop, where setting replication to 3 means 3 copies of the data.

## 4.2 Disk Priority

Set disk priority to high to allocate more disk I/O resources to buckets that require higher disk I/O access:

```
# curl -v -X POST -u Administrator:password \ http://10.1.1.101:8091/pools/default/
buckets/usertable -d
'threadsNumber=8'
```

## 4.3 Data Ejection Watermarks

Couchbase uses as much allocated memory as possible to cache data in memory. If data exceeds allocated memory, then Couchbase will eject data from memory so it only exists on disk. A series of watermarks govern when the ejection occurs. The default high watermark is 85%. To change this to 90%:

```
# /opt/couchbase/bin/cbepctl 10.1.1.104:11210 -b usertable -p password \
set flush_param mem_high_wat 90%
```

## 4.4 Memory Optimized (MOI) vs Standard Global Secondary Indexes (GSI)

Couchbase creates indexes to lower query latencies. Keeping indexes in memory significantly reduces latency, and both initial and ongoing indexing times are faster compared to standard GSI

- MOI is optimal for lower latency and highest throughput needs and requires machines with larger memory to keep the index in RAM. A high-performance I/O subsystem is not required.
- Standard GSI can spill to disk when memory runs out. I/O subsystem performance is therefore important for optimal standard GSI performance.

## 4.5 Allocating Higher CPU Resources for Index Service

Multi-Dimensional-Scaling (MDS) offers independent scalability for best computational capacity per service. You can run all Couchbase services (e.g. data, query, index, search etc.) on all nodes, but Couchbase having individual workloads run on their own set of nodes to allow workload isolation and independent scaling. You can also allocate hardware as needed for your specific workload. For example, indexes are generally memory intensive while queries are CPU intensive.

If all services are run on the same server, and To allocate more CPU resources to Index service when all services are running on the same server:

```
# curl -X POST -u 'Administrator:password' http://10.1.1.101:9102/settings \
-d '{"indexer.settings.max_cpu_percent":1600}'
```

## 4.6 Fragmentation and Compaction

Couchbase uses an append-only file structure. Cleaning up dead bytes requires compacting the disk data structures, which impacts Disk I/O. Increasing the fragmentation threshold percentage boosts performance by performing compaction less frequently.

To specify the disk fragmentation percentage that triggers bucket compaction:

```
# couchbase-cli setting-cluster -c 192.168.0.1:8091 -u Administrator
-p password \
--compaction-db-percentage=70
```

To enable auto compaction starting at 2:00 AM (during off-peak period):

```
# couchbase-cli setting-cluster -c 192.168.0.1:8091 -u Administrator
-p password \
--compaction-period-from=2:00
```

## 4.7 REST Endpoints Available to View Settings

Bucket:

```
# curl http://10.1.1.101:8093/admin/settings -u 'Administrator:password' | jq
```

Index:

```
# curl http://10.1.1.101:9102/settings -u 'Administrator:password' | jq
```

To collect all Couchbase logs as a .zip file for troubleshooting by Couchbase experts:

```
# /opt/couchbase/bin/cbcollect_info -v name_of_file.zip
```

**Chapter****5****Resources**

- [Memory Population Guidelines for AMD EPYC 7003 Series Processors](#)– Login required.
- [Socket SP3 Platform NUMA Topology for AMD Family 19h Models 00h–0Fh](#) - Login required.
- [https://devhub.amd.com/wp-content/uploads/Docs/56795\\_1.10.pdf](https://devhub.amd.com/wp-content/uploads/Docs/56795_1.10.pdf) - Login required.
- *Workload Tuning Guide for AMD EPYC™ 7003 Series Processors* (available from [AMD EPYC Tuning Guides](#))
- [Couchbase Supported Platforms](#)\*
- [Couchbase Pricing](#)\*
- [Couchbase System Resource Requirements](#)\*
- [Couchbase Overview](#)\*
- [Couchbase Sizing Guidelines](#)\*
- [Couchbase Database Indexing Best Practices](#)\*
- [Blog: Top 10 things an Ops / Sys Admin Must Know about Couchbase](#)\*
- [Documentation for /proc/sys/vm/\\*\(kernel version 2.6.29\)](#)\*

*This page intentionally left blank.*

## Chapter

## 6

## Glossary

- **ACPI** - Advanced Configuration and Power Interface
- **BIOS** - Basic Input/Output System
- **CCD** - Core Complex Die
- **CCX** - Core Complexes
- **cTDP** - Configurable Thermal Design Power
- **DIMM** - Dual In-line Memory Module
- **DPC** - DIMMs Per Channel
- **DRAM** - Dynamic Random-Access Memory
- **LLC** - Last Level Cache
- **MDADM** - Multiple Disk and Device Administration
- **NIC** - Network Interface Card
- **NUMA** - Non-Uniform Memory Access
- **PPL** - Package Power Limit
- **OPN** - Orderable Part Number
- **OS** - Operating System
- **SLIT** - System Locality Information Table
- **SMT** - Symmetric Multithreading
- **SRAT** - System Resource Affinity Table
- **TCO** - Total Cost of Ownership
- **TDP** - Thermal Design Power
- **VM** - Virtual Machine

*This page intentionally left blank.*