v1.6 | August 2022 | BP-2056

BEST PRACTICES

MySQL on Nutanix

NUTANIX
YOUR ENTERPRISE CLOUD

Copyright

Contents

1. Executive Summary

Nutanix software natively converges compute and storage to bring the benefits of web-scale infrastructure to all types and sizes of businesses. This document highlights why Nutanix is the ideal platform for virtualized instances of MySQL database systems. For business-critical transactional and analytical workloads, Nutanix delivers the performance, scalability, and availability that your IT staff (including basis and database administrators) requires. Nutanix systems offer:

- Localized I/O and flash for index and key database files, enabling low-latency operations.
- Infrastructure consolidation that lets you eliminate underused application silos and consolidate multiple workloads onto a single, dense platform, using up to 80 percent less space with 50 percent lower capex.
- Nondisruptive upgrades and scalability, including one-click node addition without system downtime.
- Data protection and disaster recovery to automate backups.
- Uncompromising simplicity and reduced risk from eliminating complicated configurations, manual provisioning, and mapping with disks, RAID, and LUNs.

In this best practice guide, you'll learn how Nutanix can lower the TCO of your MySQL environment while still delivering top-notch performance. We particularly focus on settings for MySQL Enterprise Edition and Percona Server for MySQL. We also provide recommendations for implementing MySQL Database with InnoDB Storage Engine on Nutanix.

2. Introduction

Audience

This best practice guide is part of the Nutanix Solutions Library. We wrote it for database solution architects, database administrators, storage architects, and system engineers responsible for designing, managing, and supporting Nutanix infrastructures running MySQL. Readers should already be familiar with MySQL database administration, operating system (OS) commands, and basic Nutanix design principles.

Purpose

We cover the following subject areas:

- Overview of the Nutanix solution for delivering MySQL on a virtualized platform.
- The benefits of MySQL on Nutanix.
- Choosing the right hardware for your MySQL deployment.
- Design and configuration considerations when architecting MySQL on AOS Storage.

Document Version History

Version Number	Published	Notes
1.0	April 2017	Original publication.
1.1	September 2017	Updated platform overview and logical volume size.
1.2	April 2018	Updated Nutanix overview.
1.3	April 2019	Updated Nutanix overview.

Version Number	Published	Notes
1.4	August 2020	Updated Nutanix overview.
1.5	February 2022	Refreshed content.
1.6	August 2022	Refreshed content.

Nutanix Database Service

Nutanix Database Service (NDB) simplifies database management across hybrid multicloud environments for database engines like PostgreSQL, MySQL, Microsoft SQL Server, and Oracle Database, with powerful automation for provisioning, scaling, patching, protection, and cloning of database instances. NDB helps customers deliver database as a service (DBaaS) and an easy-to-use self-service database experience on-premises and public cloud to their developers for both new and existing databases.

3. Solution Overview

Invisible Infrastructure

Invisible infrastructure means that enterprises can focus on solving business problems instead of on repetitive and tedious management and maintenance tasks that add no value. Prism, the Nutanix consumer-grade management interface, offers uncompromising simplicity, with one-click infrastructure management, remediation, and operational insights. Eliminating complexity frees your IT staff to innovate and create.

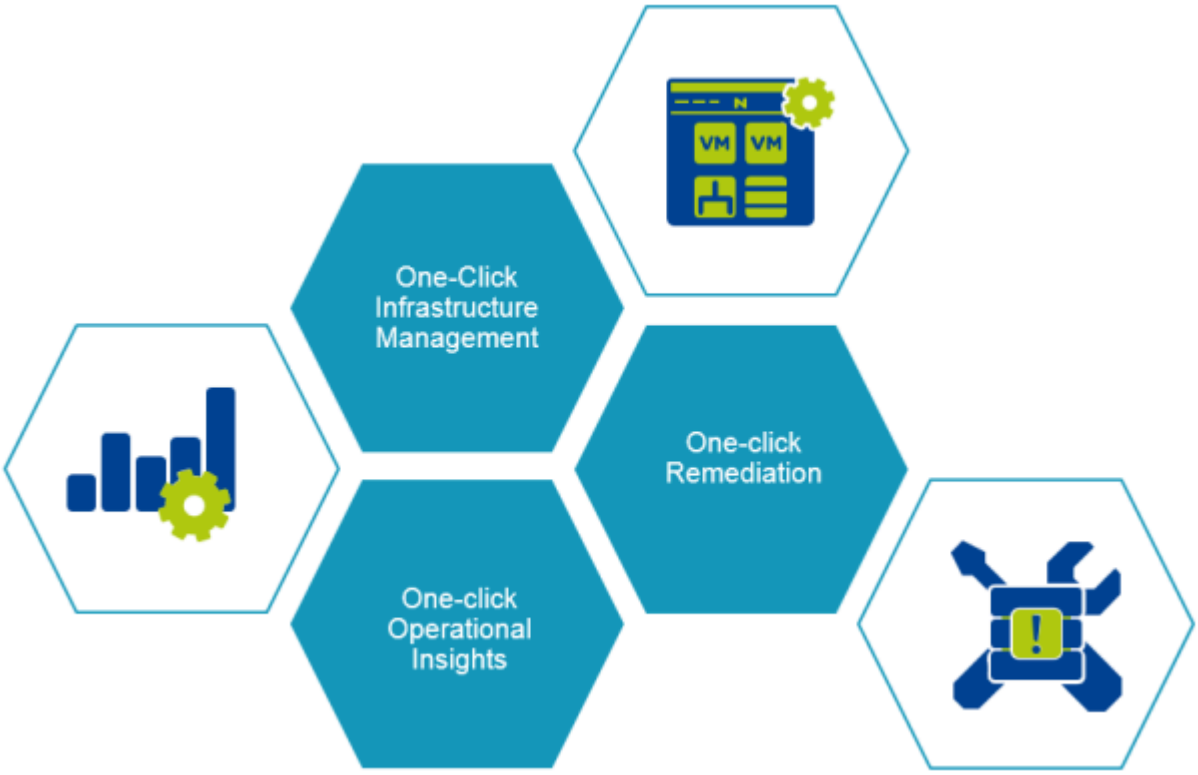

Figure 1: Nutanix Prism Overview

Deploy any application mix at any scale, all on a single platform. You can run both MySQL application servers and database VM workloads simultaneously, while isolating databases on dedicated hosts for licensing purposes. Nutanix offers high IOPS and low latency; this combination means that database computing and storage requirements drive deployment density, rather than concerns about I/O or resource bottlenecks. Our testing shows that it's better to increase the number of database VMs on the Nutanix platform to take full advantage of its performance capabilities than to scale large numbers of database instances or schemas in a single VM. From an I/O standpoint, Nutanix handles the throughput and transaction requirements of demanding transactional and analytical databases with AOS Storage.

Compression

The Nutanix Capacity Optimization Engine (COE) transforms data to increase data efficiency on disk, using compression as one of its key techniques. AOS Storage provides both inline and post-process compression to suit the customer's needs and the types of data involved.

Inline compression condenses sequential streams of data or large I/O sizes in memory before writing them to disk, while post-process compression initially writes the data as usual (in an uncompressed state), then uses the Nutanix MapReduce framework to compress the data cluster-wide. When you use inline compression with random I/O, the system writes data to the oplog uncompressed, coalesces it, and compresses it in memory before writing it to the extent store. Nutanix uses LZ4 and LZ4HC for data compression. This method provides good compression ratios with minimal computational overhead and extremely fast compression and decompression rates.

The following figure shows an example of how inline compression interacts with the AOS Storage write I/O path.

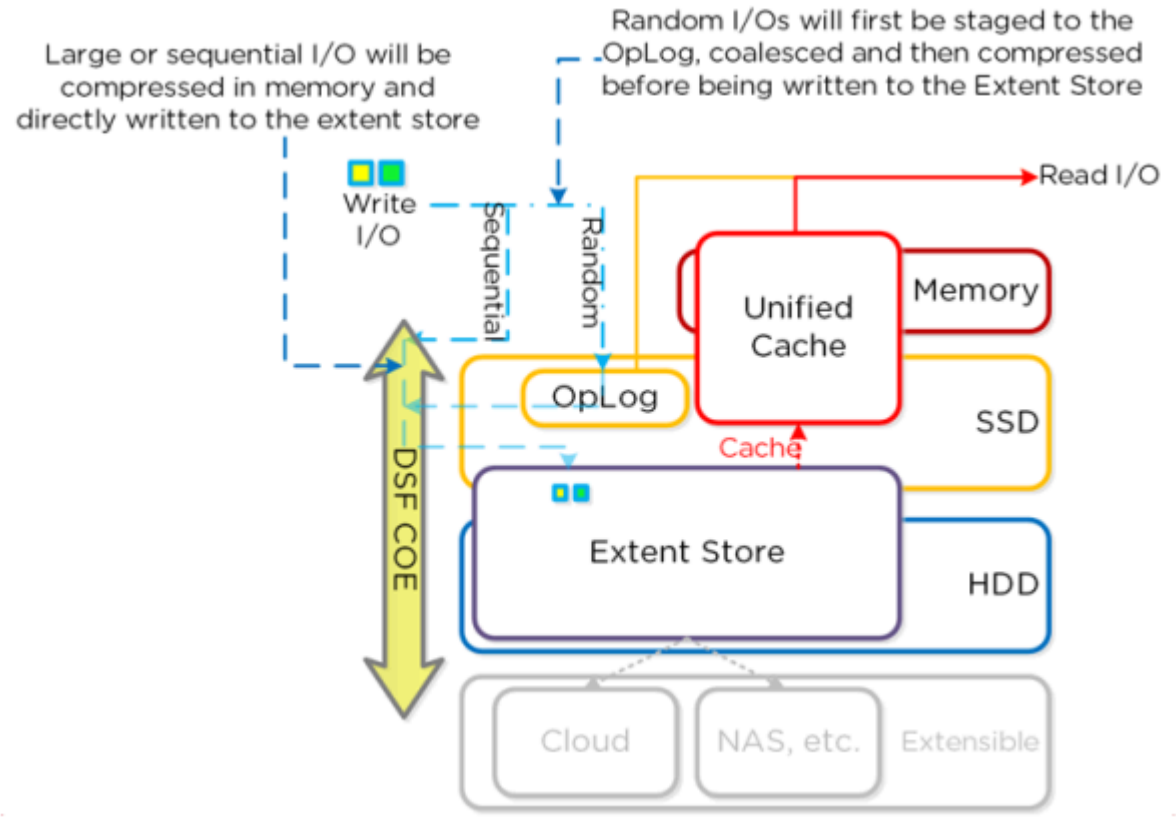

Figure 2: Information Life Cycle Management and Compression

4. MySQL on Nutanix Best Practices

* Always refer to the Nutanix website, the support portal, and our collection of solutions-specific documents for the most up-to-date information.

MySQL, now owned by Oracle Corporation, is the world's most popular open-source database for high-performance data services. Many business-critical, web-based, and cloud-enabled applications rely on MySQL to deliver high-performance transactional and ETL (extract, transform, load) workloads.

> Note: Nutanix is an Oracle Gold Partner.

In the MySQL on Nutanix best practices that follow, most configurations and optimizations occur at the database and OS levels.

Many database administrators still believe that virtualizing your database isn't a good idea. The countless tips, tricks, and white papers available on the subject only seem to add to the confusion. This guide is meant to make it as simple as possible to deploy MySQL workloads on Nutanix, while giving you deep insight into every aspect of the solution to ensure reliable, consistent, and fast performance for your database workloads. This guide also acts as a bridge between the various silos in IT organizations, where database, virtualization, and infrastructure administrators need visibility and harmony to design a high-performance application solution.

CPU

> Note: Work with your account team to choose the best CPU for your workloads.

Nutanix and its OEM vendors offer various CPU models designed for different purposes. Databases generally perform better with CPU types that have higher clock speeds (GHz) than core counts. With these CPU types, you get strong performance with fewer cores, saving on licensing costs. CPU types with higher clock speeds also achieve lower latency. If licenses aren't a constraint, there are CPU types that offer both higher clock frequencies and a greater number of

cores per socket. If you're using a socket-based license, you can choose sockets with many cores or select from the single-socket options available from our OEM vendors.

Currently, Intel CPUs have some of the best thread performance in the industry. Single-thread performance affects how well your applications can take advantage of Intel CPU offerings and is key to understanding the positive impact that moving workloads from Unix to x86 environments can have on your workload performance and stability. Modern Intel CPU families perform much better than most other CPU types out there and do so at a lower TCO.

For a clear example of the advantages that modern Intel CPUs can offer, see AnandTech's Intel Xeon E7-8800 V3 review. In general, even a midrange Intel E5 at 2.3 GHz performs better than a top-of-the-line IBM POWER 8 at 3.5 GHz overall. This difference is very noticeable in databases that migrate to Nutanix from Unix platforms. Combined with all-flash storage, cheap RAM means you can achieve fast performance at very low latencies.

Hyperthreading

Hyperthreading is available on Nutanix as a default, so it requires no administrative intervention.

NUMA

> Note: Size VMs to be within the boundary of a NUMA node whenever possible.

NUMA (nonuniform memory access) is a method of configuring a cluster of microprocessors in a multiprocessing system to share memory locally, which improves performance and the system's expandability.

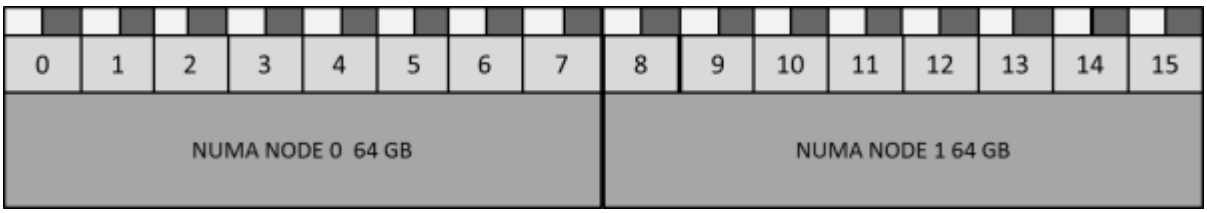

Figure 3: NUMA Node Boundaries

The previous diagram depicts two processors with eight cores each (logical CPU) and 128 GB of RAM total on a server. Each NUMA node (or socket) gets

64 GB of RAM locally. Local memory access times are very fast on NUMA-based systems because the memory controller connects directly to one processor. Remote memory access is many times slower than local.

The number of configured and available vSockets directly impacts how likely a given transaction is to require direct versus remote access. Modern hypervisors can provide vNUMA-aware scheduling, so VM configuration affects OS behavior.

As an example, if you use an Intel E5 2699 v4 CPU, which has 22 cores for each socket, size the VM at 22 vCPU or fewer to fit within the NUMA boundary. This limit enables the VM to schedule itself in the NUMA socket while also addressing locally available memory for faster access. If the VM is wider (in this example, more than 22 vCPU), it must access the memory of the other NUMA node remotely, which creates overhead.

Extra vCPUs

If workload monitoring shows that the database isn't using all the vCPUs, the extra vCPUs could cause scheduling constraints, especially under high workload.

> Note: Start with a lower vCPU count for your database and scale up in smaller increments if you witness performance issues. More isn't always better.

CPU Reservations

Generally, setting CPU reservations isn't critical. Tests on vCPU overcommitment show graceful degradation in performance, which you can overcome without any downtime by rebalancing the workloads using VM migration tools across a Nutanix cluster. This solution assumes spare CPU capacity in the cluster.

> Note: Avoid initial vCPU core oversubscription for tier-1 workloads.

CVM Utilization

> Note: Take Controller VM (CVM) usage into account.

The CVM is a key component of the IP storage framework in the Nutanix stack. It acts as a scale-out storage controller and manages performance for Nutanix systems. By default, the CVM uses 8 vCPU. This default number is generally acceptable for most database workloads and works well without adding CPU

overhead. However, for a VLDB (very large database), you may need a larger CVM to account for the additional IOPS it has to handle while maintaining low latency. Keep this factor in mind when deciding on the CPU and account for it in your sizing and design.

Hot-Adding vCPUs

> Note: Consider NUMA boundaries before you hot-add vCPUs. Hot-adding vCPUs can disable vNUMA in VMware vSphere. For more information, see VMware KB 2040375 (`https://kb.vmware.com/s/article/2040375`).

The effect of hot-adding vCPUs is particularly relevant to database VMs, which can be NUMA-wide. Because we generally size databases using vCPU types capable of handling peak workloads with an additional buffer, hot-add might not be an urgent use case. However, if you need to hot-add vCPUs beyond a NUMA boundary, what you lose in vNUMA benefits depends on the workload and on NUMA optimization algorithms specific to the database vendor and the version of VMware vSphere you use. To determine whether the performance tradeoff is warranted in your specific circumstances, VMware recommends determining the NUMA optimization benefits based on your own workload before setting the hot-add vCPU function.

The hot-add capability is available on all Nutanix-supported hypervisors and AOS versions.

Memory

Memory per vCPU

> Note: Assign a minimum of 8 GB per provisioned vCPU.

This guideline applies to both database and application servers, although some applications require less memory. The database server is likely to need more RAM, so determine its allocation based on requirements and testing. Databases love RAM because they usually cache data in memory before sending the I/O to storage, and RAM is faster than storage. The faster the database can access data from memory, the sooner it frees up CPU cycles for the next tasks, which

gives you better performance, lower latencies, and optimal use of your compute resources.

Memory Reservations

> Note: For production systems under strict performance SLAs, set memory reservations equal to the VM size.

Production databases should have memory 100 percent reserved, with no overcommitment allowed. We recommend that you use Intel-based memory because it's affordable.

Large Memory Pages

> Note: Use large memory pages for databases.

Large page support is enabled by default on VMware vSphere and supported in Linux and Windows. Using large pages can increase TLB (translation lookaside buffer) access efficiency and improve program performance.

> Note: Large pages can speed up memory allocation to a VM, as described in VMware KB 1021896 (`http://kb.vmware.com/kb/1021896`).

OS Swap Space

> Note: Configure the size of the OS swap space inside the VM to 16 GB.

This setting ensures that the swap space can easily handle memory dumps during a leak error, if necessary. Ideally, if configured and sized correctly, the database shouldn't be paging at all.

Virtualization Memory Overhead

> Note: Take virtualization memory overhead into account.

Because we don't recommend that you oversubscribe memory when you design and implement database workloads, a correct sizing must provide enough physical memory to support all VM memory requirements. Hypervisors also need physical memory to operate and thus create a small amount of memory overhead per VM, depending on the assigned resources. Take that overhead into account.

Storage

Storage Options

Nutanix has some of the best storage options available in the industry today. You can choose either our hybrid nodes or an all-flash array (AFA). Our hybrid nodes have both SSDs and HDDs. SSDs handle both reads (for active data) and writes. HDDs store cold data, which the system doesn't need for active operations, on more economical disks.

> Note: Select all-flash nodes whenever possible.

The cost differences between SSDs and HDDs have narrowed over the years, and AFAs provide far better performance and capacity than hybrid nodes at almost the same cost. These developments affect your database workloads positively. With AFAs, you get better IOPS overall, and thus lower CPU utilization.

> Note: Choose your hardware model based on compute, storage, and licensing requirements.

Some specific guidelines for selecting a hardware model include:

- Keep the working set in SSD and the total database size within node capacity when possible.
- Select a model that can fit all the database storage on a single node. For databases too large to fit on a single node, ensure that there is ample bandwidth between nodes.
- Use node models with more memory for I/O-heavy workloads.
- Use a node with a memory size twice that of the largest single VM.
- Use a node that fits your organization's licensing constraints.

Storage Protocols

Nutanix supports common storage protocols, such as NFS (using AOS Storage) and iSCSI (using Nutanix Volumes). Each of these storage protocols can achieve excellent performance.

Replication Factor

Nutanix relies on a replication factor for data protection and availability. This method provides the highest degree of availability because it doesn't require reading from more than one storage location or data recomputation on failure. However, this advantage comes at the cost of storage resources, as it requires full copies.

> Note: We recommend a replication factor of 2 for most database workloads.

Environments with higher protection requirements can use replication factor 3 but doing so requires more nodes and storage capacity.

vDisks

- Use standard VMDK vDisks on a standard container or datastore.
- Create multiple vDisks for your database.
- Separate disk groups for logs and data.

> Note: We recommend that you assign the database log vDisk to a separate PVSCSI adapter, especially when you use VMware vSphere. Spread the database files across virtual SCSI controllers. This distribution maximizes parallel I/O processing in the guest OS. In AHV, the design is much simpler, with a single VSCSI adapter.

- Use paravirtualized SCSI adapters for database data and log virtual disks wherever applicable (for example, in VMware vSphere).

Containers

> Note: Create only the containers you need.

There's no technical advantage to using multiple containers, but customers can choose to have more than one for organizational and operational purposes like replication. When you use vSphere, you might need to create two containers to gain maximum performance, especially in an all-flash node, because of the performance of a single NFS network thread. Nutanix AHV doesn't require more than one container for performance benefits.

Thin Disks

> Note: Use thin disks.

There's no performance advantage in a Nutanix environment to using lazy-zero or eager-zeroed thick disks over thin disks.

Data Efficiency

- For containers created on Nutanix, enable inline compression (delay=0) for improved performance and space savings. With inline compression, customers gain much more usable disk capacity with no performance impact—in certain cases, performance even improves.
- Disable deduplication for database environments.
- Use erasure coding for archival workloads. Don't use erasure coding for database workloads, unless Nutanix Support advises you to enable it.
- Increase the queue depth to increase performance for I/O-intensive database workloads. See VMware KB article 2053145 on large-scale workloads with intensive I/O patterns for more information.
- In VMware vSphere, disable Storage I/O Control and Storage DRS, as they offer no benefit in a Nutanix environment. Nutanix has built-in noisy neighbor mitigation because of the platform's web-scale design and the storage controller on each Nutanix node.
- Use Nutanix snapshots to create crash-consistent, point-in-time copies of systems as needed to avoid VM stun or pauses in the hypervisor. Snapshots require Nutanix Guest Tools (NGT).

> Note: A local snapshot isn't a backup.

Adding Storage Capacity

> Note: Add storage capacity with storage nodes.

Storage nodes provide additional storage capacity but don't run VMs, so they need no additional hypervisor licenses. The system can use this added capacity for supplementary snapshots or other storage requirements.

Snapshot Behavior

When you delete a VMware snapshot (for example, when you back up a VM running a database in a three-tier setup), there might be a period during which the VM is stunned. See VMware KB article 1002836 for more information. The stun can cause application servers to disconnect unless they're configured to automatically reconnect.

> Note: Don't delete a VMware snapshot while a batch job is running. This operation could cause the batch job to cancel. This limitation only applies to snapshots with VMware.

> Note: We recommend that you use Nutanix snapshots wherever possible.

Networking

- Use hypervisor network control mechanisms (for example, VMware NIOC).
- Use the VMXNET family of paravirtualized network adapters. These adapters implement an optimized network interface that passes network traffic between the VM and the physical NICs with minimal overhead.
- Use low-latency switches with at least 10 GbE connectivity.
- Use redundant 10 GbE uplinks from each Nutanix node.
- Don't use network fabric extenders for your core storage network. Network fabric extenders adversely impact the storage subsystem's performance, latency, and reliability.

Sizing and Architecture

It's better to have multiple smaller application servers than one large application server for your databases. This best practice reduces CPU context switching between processes and generates less overhead. We also recommend that you have one VM per application server to manage workload distribution and resiliency.

Working set size (WSS)

As noted previously, size for your hot tier (SSD). The simplest way to estimate your active WSS is to take the maximum delta of your monthly backup and multiply that number by three.

For example, the following table shows six months of backup deltas for a database that initially went live with 100 GB. The delta is the difference in size between the previous database size and the next available size in the table.

Table: Sample Database Information

Month	Database Backup Size	Delta
Month 1	110 GB	10 GB
Month 2	125 GB	15 GB
Month 3	150 GB	25 GB
Month 4	180 GB	30 GB
Month 5	200 GB	20 GB
Month 6	230 GB	30 GB

In this case, the largest delta available is 30 GB, so the WSS for this database is 30 GB x 3 = 90 GB. Combine the WSS for all your databases to determine the minimum SSD size required as a starting point.

Note: The extent store (not the advertised size of the SSD) is directly proportional to the WSS.

This method for calculating WSS doesn't apply to every scenario, because certain database systems have far more read-heavy or write-heavy requirements based on the nature of the business the system supports. Only use the delta calculation described above when no other performance data is available.

In certain scenarios, a quick estimate makes sense when sizing data isn't available. Remember that with Nutanix, you can start small and scale quickly. If no information is available, use this capability to your advantage: start small, monitor your databases, and adjust as needed.

Hardware (Node) Selection

Although Nutanix has certified and supports all available node types for running database workloads, certain node types are better suited for the different server types (VMs) that constitute a database implementation than others. We cover the lines of NX nodes offered directly through Nutanix. For information on node types available from our OEM vendors, please contact them directly.

As a guideline, we recommend the following approach: for application servers, choose nodes from the NX-3000 and NX-8000 series based on your performance requirements and the sizing done by your database team. For small and medium database servers, use nodes from the same series (whichever series you choose). For large database servers, we strongly recommend nodes from the NX-8000 series because of their greatly enhanced cold tier performance and their overall optimization for database workloads.

We also strongly recommend using all-flash nodes (SSDs) whenever possible. AFAs are available for all Nutanix node models for databases that require very high performance.

Sizing Support

To ensure alignment with your requirements, work with the Nutanix sales team to create a definitive design for your Nutanix cluster and choose the nodes for that design. The database specialists in the vBCA group of Nutanix Engineering's Solutions and Performance group (`vbca@nutanix.com`) can support these sizing exercises.

Migration Support

Migrating databases from older hardware or different operating systems can become critical as you move to Nutanix. We recommend that you engage Nutanix Xpert Services to help you migrate your existing workloads from physical or virtual systems to Nutanix, while observing tested and proven best practices for your selected configuration. With their guidance, you can move your database workloads onto Nutanix without compromising the performance, reliability, or stability of your applications.

Hypervisor

Nutanix supports three different hypervisors for use on the enterprise cloud:

- Nutanix AHV
- VMware ESXi
- Microsoft Hyper-V

All three of these hypervisors can run your database environments. It's up to the respective OEM (Dell, Lenovo, and so on) to support the different hypervisors for database virtualization. Please make sure to check with your specific OEM about their current hypervisor support situation.

Hypervisor-related best practices include:

- If you use AHV, review the AHV best practices guide.
- If you use vSphere, review Performance Best Practices for VMware vSphere.
- Size Nutanix clusters for a minimum of n + 1 redundancy.
- Don't use resource pools unless you absolutely must. If you do use resource pools, ensure that the shares are sized correctly.
- Always use the latest Nutanix AOS version. You can update AOS without any downtime or impact to your production workloads.
- We recommend that you use the latest hypervisor upgrades, subject to testing and validation by your IT team. Using Nutanix Prism to upgrade your hypervisor generally doesn't require downtime.
- Use hypervisor high availability wherever possible. High availability is a very effective method of protecting your database VMs against hardware component failures. In most SLA scenarios, hypervisor high availability is more than sufficient to meet availability requirements, as long as the design accommodates reasonable failure scenarios.

Guest Operating Systems (Linux, Windows)

Although almost all Linux distributions support MySQL, most customers deploying MySQL tend to use Oracle Linux (OEL), Red Hat Linux (RHEL), CentOS, SUSE Linux (SLES), or Ubuntu. For MS Windows, we recommend Windows 2008 R2 and later.

Specific tips for guest operating systems include:

- If required for your hypervisor, install the latest version of guest tools in the guest OS. For AHV, you must install NGT, primarily used for Windows guest operating systems and some Linux distros. For VMware ESXi, install VMware Tools. Hyper-V doesn't require guest tools.
- Turn off all OS services that your organization doesn't need. Refer to individual OS hardening guides for more information on securing and optimizing your OS.
- Refer to VMware KB article 1006427 for guidance on how to minimize VM time drift.

Linux 2.6

From Linux 2.6 onward, set the Linux kernel I/O scheduler to NOOP. We recommend NOOP for VMs and SSDs, as it tries not to interfere with I/O processes and uses simple FIFO. ESXi uses an asynchronous intelligent I/O scheduler; virtual guests should perform better when ESXi handles I/O scheduling.

- To set NOOP at boot time, add the elevator option at the kernel line in the /etc/grub.conf file:

```
elevator=noop
```

- While it's possible to set the elevator option for a disk manually, we don't recommend doing so, because you must set it manually every time you add a new disk or device to the OS.
- Also add the following line to the boot loader grub.conf:

```
iommu=soft elevator=noop apm=off transparent_hugepage=never powersaved=off
```


For these settings in Linux versions prior to 2.6, refer to your specific Linux guide. Some Linux distributions, such as SLES 12, don't have grub.conf anymore, so this setting may not be applicable.

VM Swap and Page Parameters

Note: Set VM swappiness to 0 in /etc/sysctl.conf.

The swappiness control defines how aggressively the kernel swaps memory pages. Higher values increase aggressiveness; lower values decrease the amount of swap. A value of 0 instructs the kernel not to initiate swap until the number of free and file-backed pages is less than the high-water mark in a zone.

```
# sysctl -w vm.swappiness=0
```

Administrators can tune the following advanced parameters depending on requirements:

- Allow the Linux kernel to better handle memory overcommit:

  ```
  vm.overcommit_memory = 1
  ```

- The hugepages mechanism allows the Linux kernel to use the multiple page size capabilities of modern hardware architectures:

  ```
  vm.nr_hugepages=Same as your DB buffer memory, maximum to VM memory size
  ```

- Define the number of persistent hugepages configured for use in the VM or host:

  ```
  vm.dirty_background_ratio = 5
  ```

- Contains, as a percentage of total available memory containing free pages and reclaimable pages, the number of pages at which a process that generates disk writes starts to write out dirty data itself:

  ```
  vm.dirty_ratio = 15
  ```

- This tunable defines when dirty data is old enough to be eligible for the kernel flusher threads to write it out:

  ```
  vm.dirty_expire_centisecs = 500
  ```

- The kernel flusher threads periodically wake up and write old data out to disk:

  ```
  vm.dirty_writeback_centisecs=100
  ```


Linux Maximum I/O Size

> Note: The Linux maximum I/O size mostly applies to kernel 4.x and later but check this in your kernel.

The default maximum I/O size set by Max_Sectors_KB restricts the largest I/O size that the OS issues to a block device. Whether you issue I/O at this size depends on the elevator (scheduler) in use, the driver, and the type of I/O your applications issue. However, large reads and writes are often at the maximum I/O size.

UDEV (user space /dev) can ensure that all block devices connected to your VM, even if they're hot plugged, have the same, consistent maximum I/O size applied. Simply create a file 71-block-max-sectors.rules under /etc/udev/rules.d/ with the following line:

```
ACTION=="add|change", SUBSYSTEM=="block", RUN+="/bin/sh -c '/bin/echo 1024 > /sys
%p/queue/max_sectors_kb'"
```

If you don't have UDEV in your distribution, you can use rc.local as an alternative and essentially apply the same command. As an example (reboot required):

```
echo 1024 > /sys/block/sd?/queue/max_sectors_kb
```

> Note: Without UDEV, you must change max_sectors_kb on every single disk device in your OS individually.

Additional Guest OS Configuration Tips

On high-performance networks, increase network receive and transmit queues. Add these to rc.local:

```
/sbin/ethtool -G ethX rx 4096 tx 4096
```

For very high-performance database systems, add the following options to the boot loader (grub) configuration in addition to those listed above (PVSCSI required):

```
vmw_pvscsi.cmd_per_lun=254 vmw_pvscsi.ring_pages=32
```

On Windows, use the following command to adjust OS queue:

```
REG ADD HKLM\SYSTEM\CurrentControlSet\services\pvscsi\Parameters\Device
/v DriverParameter /t REG_SZ /d "RequestRingPages=32,MaxQueueDepth=254"
```


Add options = -x to /etc/sysconfig/ntpd in the time server option:

```
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"
```

Ensure that your firewall policy allows database ports to communicate. This policy setting includes the modifications required in SELinux.

File System

To maximize your database's performance and scalability in the Linux OS, we recommend that you use LVM (Logical Volume Manager). LVM creates a logical volume that aggregates multiple disks to stripe reads and writes and also serves as a scalable volume for maintenance.

> Note: When you use LVM, we recommend that you stripe volumes (don't concatenate) over multiple disks. Use the lvs --segments command to check whether striping or concatenation is in use.

Keep the logical volumes (LVs) and physical volumes (PVs) for data files separate from LVs and PVs used for redo logs and archive logs. For example, assume you created four vDisks and presented them to the OS. The Linux OS recognizes them as sdb, sdc, sdd, and sde. Issue this command to identify the device names:

```
fdisk -l
```

Now create the physical volume for all these devices:

```
pvcreate /dev/sdb /dev/sdc /dev/sdd /dev/sde
```

Create a volume group (VG) out of these physical volumes. In this case, we've provided the name mySQLDataVG:

```
vgcreate mysqlDataVG /dev/sdb /dev/sdd /dev/sdc /dev/sde
```

Finally, create the LV for this VG. In this case, we're using the entire space in the VG for the LV and naming the LV mysqlDataLV:

```
lvcreate -l 100%FREE -i4 -I1M -n mysqlDataLV mysqlDataVG
```

> Note: In the previous command line, change -i4 to the number of disks in your VG. If you have eight disks, then this argument is -i8. This setting stripes the LV across all the disks in the VG.

The previous process creates an LV for the MySQL data volume. Some additional LV tips include:

- We recommend that you have at least two LVs: one for MySQL data and one for MySQL logs. If you have a high-performance OLAP database, you may need a third LV for MySQL undo logs.
- We recommend that you have four vDisks in each LV. The size of the vDisks depends on your database size and design. As a default, start with 50 GB vDisks for data disks and 10 GB vDisks for logs.
- Turn off read-ahead for each LV (following the example at the beginning of this section):

```
lvchange -r 0 /dev/mysqlDataVG/mysqlDataLV
```

- Create file system:

```
mkfs.ext4 /dev/mysqlDataVG/mysqlDataLV
```

Note: XFS can provide significant performance improvement, especially for OLAP databases. Most customers still prefer to use EXT4 because the file system has proven stable and reliable over the years and is backward-compatible with a lot of older Linux versions as well. If you decide to use XFS as the file system, use the mount option for it described in the following section.

- Mount options for the LV: Assuming you created a directory in the path /mysql/data in your OS, use the following options to mount the LV to the directory (ensure that they are in /etc/fstab for permanent effect).
- For EXT4:

```
mount -o noatime,barrier=0 /dev/mysqlDataVG/mysqlDataLV /mysql/data
```

- For XFS:

```
mount -o inode64,nobarrier,noatime,logbufs=8 /dev/mysqlDataVG/mysqlDataLV /mysql/
data
```

Parameter Key:

barrier=0 / nobarrier

Don't use barriers to pause and receive assurance when writing (trust the hardware).

noatime

Don't update access times (atime) metadata on files after reading or writing them.

inode64

Use 64-bit inode numbering. (This setting is the default in the most recent kernel trees.)

logbufs=8

Number of in-memory log buffers (between 2 and 8, inclusive).

- In Windows, when you format a disk, use 4,096 as the allocation unit size for MySQL data and log disks. Using mount points is preferable to using drive letters in Windows.
- The Windows equivalent of LVs is storage spaces. For larger databases, you can explore using storage spaces in Windows.
- Ensure that all disks associated with multiple LVs are spread over multiple PVSCSI adapters. (This action isn't required for AHV.) The following figure depicts an example with two LV groups. This design can easily cater to a medium to large database's performance requirements.

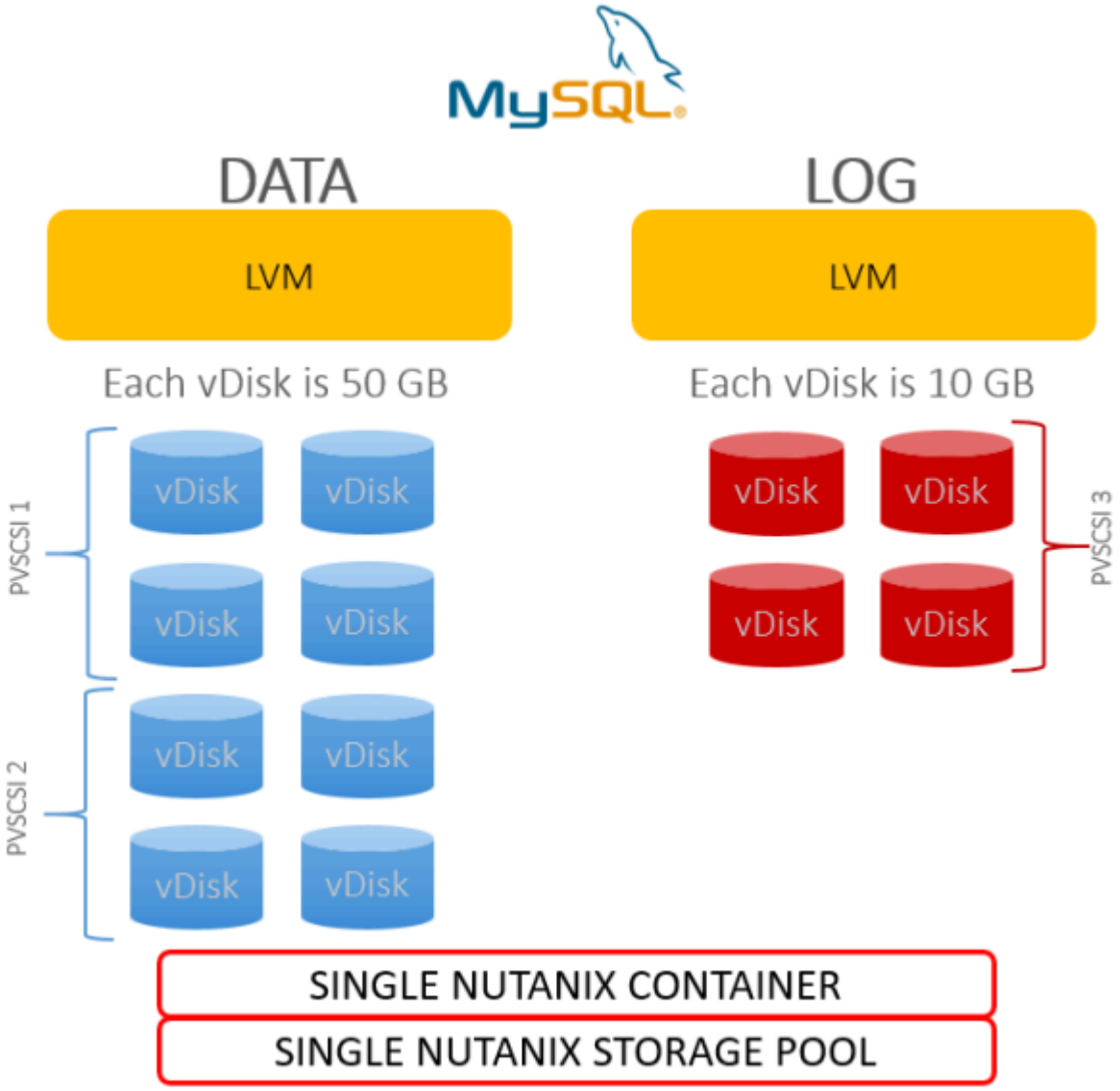

Figure 4: LVM Disks Spread over Multiple PVSCSI Adapters

MySQL Database

- We recommend that you use the InnoDB storage engine for your database. Although MyISAM can be faster in certain cases, it doesn't have any support for referential integrity checks (so it isn't a relational database management system) and it performs table-level locks (not row-level locks). InnoDB is a fully ACID-compliant engine.

- Most of the MySQL parameters are typically stored in a file called /etc/my.conf. The following table contains the output of an optimized my.conf file. Use this my.conf as guidance and read through each configuration to assess whether it applies to your database scenario and what the user-defined values can do.

Table: Optimized my.conf File Output

# Optimized my.conf file for MySQL on Nutanix	**Notes**
[mysqld]	
# MyISAM directories	
datadir=/mysql/data	# Change to user-defined path for MySQL for MyISAM.
# InnoDB Data & Log directories	
innodb_data_home_dir = /mysql/data	# User defined. Separate VG. Drive path for Windows.
innodb_log_group_home_dir = /mysql/dblog	# User defined. Separate VG. Drive path for Windows.
innodb_file_per_table	# To ensure every table has a separate file.
tmpdir=/mysql/tmpdir	# To ensure that /tmp is not used for temp ops.
innodb_undo_directory = /mysql/undo	# User defined. Separate VG.
# Recommended in standard MySQL setup	
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES	
#InnoDB storage engine defaults	
default-storage-engine = innodb	
default_tmp_storage_engine = innodb	
# InnoDB logs section	
innodb_log_files_in_group = 4	# User defined.

# Optimized my.conf file for MySQL on Nutanix	Notes
innodb_log_file_size = 16G	# User defined. Higher is better but takes longer to recover.
# InnoDB log buffer	
innodb_log_buffer_size = 8M	# User defined for OLTP performance.
# InnoDB Memory and Buffer section	
innodb_buffer_pool_size = 32G	# Change to 80% of vRAM assigned to VM for dedicated DB VM.
large-pages	# Enable large pages in MySQL.
innodb_buffer_pool_instances = 64	# Tune for concurrency.
innodb_flush_method=O_DIRECT	# Recommended. Avoids double buffering.
innodb_flush_neighbors=0	
innodb_flush_log_at_trx_commit=1	# User defined. Can be set to 2, which has better performance but with a data integrity penalty.
innodb_buffer_pool_dump_at_shutdown=1	# Optional.
innodb_buffer_pool_load_at_startup=1	# Optional.
bulk_insert_buffer_size = 256	# User defined. Tune for bulk inserts.
innodb_thread_concurrency = 16	# Typically 2x vCPU. Can be more.
# Undo tablespace	
innodb_undo_tablespaces = 5	# User defined.
innodb_undo_logs = 20	# User defined.
# Networking	
wait_timeout=57600	
max_allowed_packet=1G	# User defined.
socket=/var/lib/mysql/mysql.sock	

# Optimized my.conf file for MySQL on Nutanix	Notes
skip-name-resolve	# Recommended to avoid name resolution overhead.
bind=0.0.0.0	# Bind MySQL to specific IP if required.
port=3306	# 3306 default. User defined.
max_connections=1000	# No. of connections allowed to DB. User defined.
# Advanced tuning (test first)	
query_cache_type = 1	
innodb_io_capacity = 200	
thread_cache_size = 32	

- You can install MySQL binaries from Oracle's website, Percona, or MariaDB. For any other variants, refer to the respective setup manual and any additional parameters you may need to observe.
- From MySQL 5.6 onward, queries can slow down significantly on large tables. This slowdown is especially the case with ICP (Index Condition Pushdown) optimization. If you experience this issue, refer to the index condition pushdown optimization section of the MySQL manual.
- memcached is the distributed in-memory object-caching daemon in MySQL. This daemon helps speed up reads significantly for small chunks of data from active tables in the database. If you plan to implement memcached, do so on Nutanix nodes that have a larger memory footprint for scalability, such as nodes in the NX-8000 series.

High Availability

- Use on-hypervisor high availability whenever possible.
- Use MySQL clustering if required.
- Leave VMware DRS automation at the default level (3).
- Let the hypervisor and DRS manage noisy neighbors and workload spikes.

- Use at least n + 1 for high availability configuration.
- Use hypervisor antiaffinity rules to separate clustered nodes, application VMs, and database VMs.
- Use a percentage-based admission control policy for VMware environments.
- Review MySQL VM restart priorities.

5. Monitoring

Prism

Nutanix Prism is an end-to-end, consumer-grade management solution for virtualized datacenter environments that combines several aspects of administration and reporting to offer unprecedented simplicity. Powered by advanced machine learning technology, Prism can mine large volumes of system data to automate common tasks and generate actionable insights for optimizing virtualization, infrastructure management, and everyday operations.

We designed Prism from the ground up to deliver a rich yet uncluttered experience. It provides an intuitive user interface to simplify and streamline routine datacenter workflows, eliminating the need to have disparate management solutions for different tasks.

Figure 5: Prism Dashboard

Prism is included with all Nutanix licenses; for operations like managing multiple clusters or AI-powered capacity planning, you may want to investigate

our expanded product offerings such as Prism Central and Prism Pro. The Prism console offers a wealth of information, including storage performance, cluster performance, alerts management for any critical events or warnings, compression and deduplication ratios, and overall resource health. For more information on Prism, Prism Central, and Prism Pro, please refer to the Prism product page.

MySQL Workbench

MySQL Workbench, available as open-source software, is one of the leading MySQL monitoring and administration tools. Almost all DBAs use MySQL Workbench as a go-to tool to manage MySQL workloads, and it can help you understand and configure some of the advanced features in MySQL with tips and dependency checks.

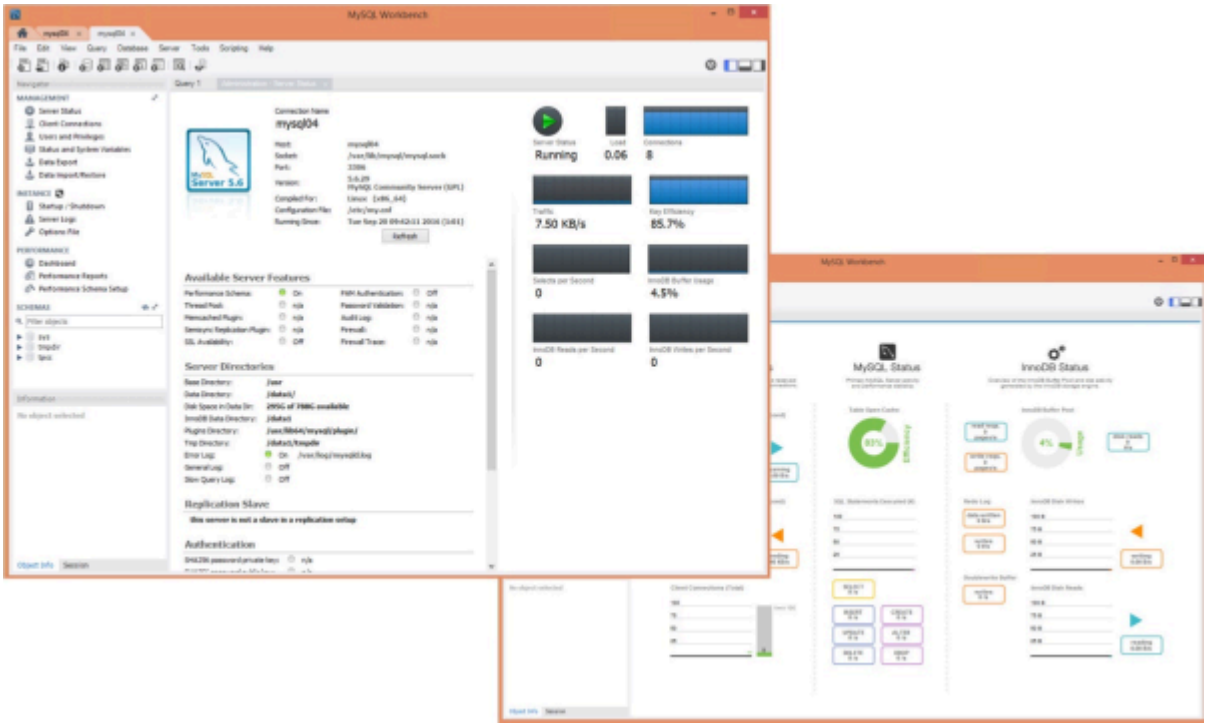

Figure 6: MySQL Workbench

For more monitoring, auditing, and enterprise tools from Oracle for MySQL databases, please refer to the MySQL datasheet.

Glances

Glances is a Python-based Linux OS monitoring tool that gives you insight into your database OS and allows you to correlate the data between performance and resource utilization.

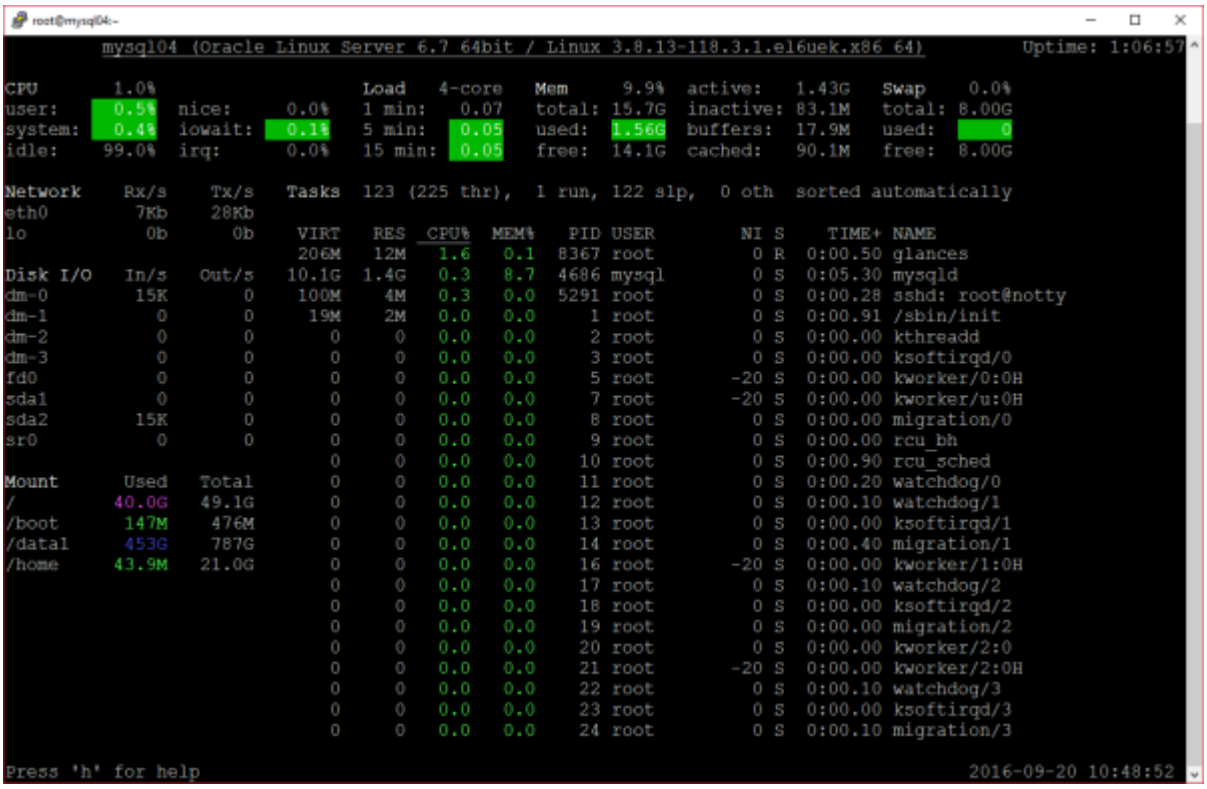

Figure 7: Glances

6. Conclusion

Most of the time, you can run workloads in Nutanix almost immediately—especially DevOps and critical workloads. However, when deploying high-performance, critical workloads on Nutanix, sometimes customers can benefit from some additional guidance—which is where this document comes in.

In this guide, we covered general strategies and tips for designing your MySQL database to run on Nutanix with uncompromised performance. Larger implementations also need a proper methodology and migration plan—and we can help. Please contact your Nutanix representative early when planning large-scale database projects, so you can move to Nutanix without impacting your IT or your business.

7. Appendix

References

- MySQL
- Percona (Database Performance)
- Nutanix AHV Best Practices
- Best Practices for Performance Tuning of Latency-Sensitive Workloads in vSphere VMs
- VMware KB 2053145: Large-scale workloads with intensive I/O patterns might require queue depths significantly greater than Paravirtual SCSI default values
- VMware KB 1002836: Snapshot removal stops a virtual machine for long time
- VMware KB 1006427: Timekeeping best practices for Linux guests
- MySQL 5.6 Reference Manual: Index Condition Pushdown Optimization

About the Author

Bruno Sousa is a Technical Director for Solutions Engineering at Nutanix. He is also NPX 15. Follow him on Twitter @bsousapt.

About Nutanix

Nutanix is a global leader in cloud software and a pioneer in hyperconverged infrastructure solutions, making clouds invisible and freeing customers to focus on their business outcomes. Organizations around the world use Nutanix software to leverage a single platform to manage any app at any location for their hybrid multicloud environments. Learn more at www.nutanix.com or follow us on Twitter @nutanix.

List of Figures