SUSE® Linux Enterprise Tuning and Configuration for SAS 9

Guide

www.suse.com

Guide

SUSE Linux Enterprise Server

SUSE

SUSE® Linux Enterprise Server 12 Tuning and Configuration

Achieving excellent performance with SAS 9 on SUSE® Linux Enterprise 12 SP3 is enabled by following optimal configuration and tuning practices. Those practices are summarized in this guide. For additional information or help, please contact your SUSE Linux Sales Representative.

File System Choice and Configuration

File system choice and logical layout is crucial for excellent IO performance on SUSE Linux Enterprise 12.

- **File System Type:** We recommend using XFS as the primary data file system for SAS Grid.
- **Logical Volume Layout:** We recommend creating smaller logical volumes to avoid potential bottlenecks from the use of one large volume. For maximum performance, we encourage you limit the size of each volume to 10–20 TB, where possible.
- **XFS File System Considerations:**
  If configuring XFS with RAID volumes, pay attention to the following:
  - For hardware-based RAID, set the stripe size to 64K for large block, read/write to match that of your SAS application. Any smaller and multiple IO requests will be issued; any larger and adjacent 64K reads/writes might contend, causing delays.
  - For RAID-5 or a similar parity, pay attention to the ratio of data to parity disks, to prevent the parity disk from becoming a bottleneck. The decisions here depend entirely on whether performance or redundancy is favored.

  Pay careful attention to the location of the XFS log. In multi-disk storage configurations, it is typically recommended that the XFS log be placed on a dedicated fast partition to avoid the log being a bottleneck.

Kernel Performance Tuning

Implementing tuning changes via a tuned profile is important. The following section reviews the tuned settings used for this testing and how to enable them. If you would like more information about tuning options, please refer to the *SUSE System Analysis and Tuning Guide*.

Creating a Tuned Profile

1. Create a tuned profile for the SAS workload named *sas-performance*:

```
#  mkdir /usr/lib/tuned/sas-performance
```

2. Create `/usr/lib/tuned/sas-performance/tuned.conf` and edit it to include the following:

```
[main]
include=throughput-performance
[cpu]
force_latency=1
governor=performance
energy_perf_bias=performance
min_perf_pct=100
[vm]
transparent_hugepages=never
[disk]
devices=!dm-*
```

Enabling Tuned

Run the following commands as sudo or root:

```
#  systemctl enable tuned
#  systemctl start tuned
#  tuned-adm profile sas-performance
```

Host Kernel Tuning for IO Performance

Host Kernel Tuning for IO performance for SAS large block workloads is crucial. The following multipath, scheduler, and elevator settings are very important for SAS IO performance.

Enabling Multipath

Multipath enables a server to communicate with the same physical or logical block storage device across multiple physical paths between the host bus adapters in the server and the storage controllers for the device, typically in FC or iSCSI SAN environments. It provides connection fault tolerance and load balancing across the active connections.

Enable it with the following commands:

```
#  systemctl enable multipathd
#  systemctl start  multipathd
```

I/O SCHEDULERS

The default I/O scheduler is chosen for each device based on whether that device reports as a non-rotational disk or not. For non-rotational disks, the `DEADLINE` I/O scheduler is picked. Other devices default to CFQ (Completely Fair Queuing). To change this default, use the following boot parameter:

```
#  elevator=OPTION
```

Replace *OPTION* with one of the values cfq, noop, or deadline. See Section 12.2, "Available I/O Elevators," in the *SUSE System Analysis and Tuning Guide* for details.

If the array does its own scheduling, it might be best to use the noop scheduler, to avoid both the OS and array scheduling independently. Work with your Storage Administrator or vendor to determine the best scheduling options for your architecture.

I/O ELEVATORS

The following I/O elevators are available on SLES® 12. Each elevator has a set of tunable parameters, which can be set with the following command:

```
#  echo VALUE > /sys/block/DEVICE/queue/iosched/
   TUNABLE
```

where *VALUE* is the desired value for *TUNABLE* and *DEVICE* is the block device.

For example, to find out which elevator is currently selected for the *sda* device, run the following command. Replace sda with the device in question. The currently selected scheduler/elevator is listed in brackets.

```
#  cat /sys/block/sda/queue/scheduler
   noop deadline [cfq]
```

This file can also contain the string none meaning that I/O scheduling is not enabled for this device. This typically happens when a device is using a multi-queue queueing mechanism *(refer to blk-mq below)*.

BLOCK MULTIQUEUE (BLK-MQ)

Blk-mq reduces lock contention by using per-CPU software queues to queue I/O requests. In particular, blk-mq improves performance for devices that support a higher number of input/output operations per second (IOPS), such as SSDs.

Contact us at:
www.suse.com

If you have fast SCSI devices (SSDs) instead of hard disks attached to your system, consider using blk-mq for SCSI. To enable this, use the following command:

```
#  scsi_mod.use_blk_mq=1
```

Disabling PTI and Spectre

If you ran the most recent updates and security patches on SUSE Linux Enterprise Server 12 SP3, it will likely include the Spectre/Meltdown updates. Unfortunately, these updates may cause a performance impact across all operating systems industry-wide. If gaining better performance is a higher priority than these security fixes and updates, this feature can be disabled. **However, please note that this is dangerous for security reasons and should not be conducted unknowingly if the proper alternate security measurements are not in place**. **For more information, please visit SUSE Support information for Meltdown and Spectre.** To disable this, add the following command to the kernel boot parameters:

```
#  pti=off spectre_v2=off
```

The default parameters were used for the storage configuration on the dynamic provisioning pool. The workload was evenly distributed on front-end ports to allow for required bandwidth to handle the large sequential workloads.