SAS® on Red Hat Enterprise Linux 6 (RHEL6)
Tuning Guidelines

Last Updated: June 21, 2011

This paper provides a starting point for performance optimization from a system-wide perspective to create an enhanced environment for SAS® 9 on Red Hat Enterprise Linux (RHEL) 6.

Tuning Guidelines: The RHEL 6 system needs to have the following parameters changed from their default values:

- The LUNs and logical volumes, used by SAS file systems, should be tuned for increased read ahead support. Even though the RHEL 6 **tuned** tool (commands discussed below) as well as striped LVM volumes elevate these values, they are still not typically large enough for SAS. The best way to tune them is with the **blockdev** command.

  To view read-ahead for block device:

  ```
  # blockdev --getra path-to-block-device
  ```

  To set read-ahead for a block device:

  ```
  # blockdev --setra N path-to-block-device
  ```

  This is not persistent between boots, so we recommend creating a run level init.d service script to disable it ***after*** the **tuned** service is started during bootup. We have found that elevating the default value of N to either 8192 or 16384 on all the file systems used by SAS works very well. We also recommend the value of N to be power of 2.

- The default Red Hat Enterprise Linux I/O elevator is Completely Fair Queuing (CFQ). Testing has shown this mode is not optimal for SAS. Instead the **deadline** elevator or in some cases **noop** has shown to be better for SAS sequential workloads. You can switch the I/O elevator to **deadline** using the **tuned** [throughput and/or enterprise-storage profile] tool.

- I/O Barriers can be ***safely*** disabled with enterprise storage whose cache controller RAM is battery backed up. The **tuned** tool [enterprise-storage profile] remounts file systems with I/O barriers disabled.

- Transparent Huge Pages attempts to allocate 2MB pages vs 4KB pages for anonymous memory. Applications that use page cache for large file I/O may perform better with the feature disabled. This can be done with the following command:

  ```
  # echo never >  /sys/kernel/mm/redhat_transparent_hugepage/enabled
  ```

  Note: The **tuned** tool enables this feature by default for all profiles. Any time a **tuned** profile is enabled, whether at bootup via an init.d service, or manually at the command line, this feature gets re-enabled.

Important RHEL6 tuned Commands: Please read before using the **tuned** tool.

- **How to install** – The tuned tool does not get installed by default with RHEL 6.

  # yum install tuned*

- **Listing available tuned profiles** – There are several. throughput and enterprise-storage are the most commonly used

  # tuned-adm list

- **Viewing the presently active tuned profile**

  # tuned-adm active

- **Setting a specific tuned profile**

  # tuned-adm profile enterprise-storage

- **Removing tuned profile** (if you don't like the performance)

  # tuned-adm profile default

File System Preferences: Using either **xfs** or **ext4** is best for SAS file systems.

- **xfs** performs the best.
- **ext4** performs well, but in RHEL 6.0, its extra CPU requirements could reduce overall performance of SAS. RHEL 6.1 has enhancements to eradicate the CPU overhead.
- **ext3** is the legacy file system for RHEL. Not recommended for use with SAS (especially SAS WORK) because of the delete delays.

If you are looking to use SAS in a grid environment, a shared file system is required. Discussing which shared file system is best is beyond the scope of this paper. Please contact your SAS Account team for more information regarding running SAS in a grid environment.

Paging space: Configure the paging space to include at least the following suggestions:

- Place paging spaces on dedicated disks to eliminate I/O contention. Use multiple paging spaces that are spread across multiple disks. Make the primary paging space hd6 a little bigger than the secondary paging spaces.
- Ensure that the paging space is sufficient to support the number of concurrent SAS processes (because the number of SAS processes can be dynamic, depending on application workload).

Disk layout: Minimize disk contention between SAS temporary space and data spaces.

- Avoid disk contention by placing SAS temporary-space file systems and SAS data file systems on physically-separate disks.
- Use multiple storage-server controllers to further separate and isolate the I/O traffic between SAS temporary and data spaces.
- Use multiple mount points for SAS file systems. Place the system operating system, SAS, user, SAS temporary space, and SAS data file systems on separate physical disks.
- Consider creating multiple SAS WORK areas that can be used by groups of SAS users.
- Spread the I/O workload across many physical disk spindles rather than across fewer, larger-capacity disks. Determine the sizing based on the quantity of disks rather than disk capacity. Do not wrap logical unit numbers (LUNs) around the same spindle sets.
- Do not share disk spindles with a RDBMS.

Host bus adapters (HBAs): Use an adequate number of HBAs from storage to the host server to provide the required application bandwidth.

- Consider high-performance storage channels, such as Fibre Channel technology instead of slower mediums.
- If possible, use dynamic multipathing to spread the I/O load across multiple adapters.

Redundant Array of Independent Disks (RAID): Implement storage system RAID striping across multiple physical disks.

- Use RAID10 or RAID5, depending on the level of redundancy and total capacity instead of the usable capacity that is needed for each file-system type.
- Use Logical Volume Manager (LVM) striping instead of concatenation (default) across LUNs.

LVM striping: This is extremely important if multiple storage arrays are to work together. When choosing the disk stripe or segment size, or array stripe size, note that Linux file systems are aligned on a 16 KB boundary.

- The LVM stripe size of 64K or 128K, stripe sizes of 256K or 512K has shown better I/O performance in the case of SAS 9.2 workloads. A stripe size of 8K is too small for SAS.
- Synchronize SAS BUFSIZE with the storage-system stripe size and the LVM stripe size (if using LVM striping) and read-ahead increments.

- Synchronizing I/O sizes streamlines I/O processing and reduces the number of I/O requests to the storage subsystem.

Resources: For additional tuning information, refer to the following publications:

- Best Practices for Configuring your IO Subsystem for SAS9 Applications: http://support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf

- Frequently Asked Questions Regarding Storage Configurations: http://support.sas.com/resources/papers/proceedings10/FAQforStorageConfiguration.pdf

- How to Maintain Happy SAS Users: http://support.sas.com/resources/papers/happyIT.pdf

Trademarks and Special Notices: