RED HAT® ENTERPRISE LINUX®

redhat.

HOW TO CHOOSE YOUR RED HAT ENTERPRISE LINUX FILESYSTEM

EXECUTIVE SUMMARY   Choosing the Red Hat Enterprise Linux filesystem that is appropriate for your application is often a non-trivial decision due to the large number of options available and the trade-offs involved. This white paper describes some of the filesystems that ship with Red Hat Enterprise Linux, and will provide historical background and recommendations on the right filesystem to suit your application.

TYPES OF FILESYSTEMS

Red Hat Enterprise Linux supports a variety of filesystems. Different types of filesystems solve different kinds of problems and their usage is very application-specific. At the most general level, filesystems available in Red Hat Enterprise Linux can be grouped into the following major categories:

- disk or local filesystem

- network or client/server filesystem

- shared storage or shared disk filesystem

- special filesystems

LOCAL FILESYSTEMS OVERVIEW

Local filesystems are filesystems that run on a single, local server and are directly attached to storage. For example, a local filesystem is the only choice for internal S-ATA or SAS disks and is used when your server has internal hardware RAID controllers with local drives. Local filesystems are also the most common filesystems used on SAN attached storage when the SAN's exported device is not shared.

The Red Hat Enterprise Linux 4 and Red Hat Enterprise Linux 5 platforms have traditionally provided two main filesystems for the Ext2 and Ext3 class of systems. In Red Hat Enterprise Linux 5.5 and future platforms, Red Hat also provides full support for the XFS filesystem as a layered, optional offering. The newest version of the Ext filesystem family, Ext4, is fully supported in Red Hat Enterprise Linux 6, as well as in Red Hat Enterprise Linux 5.6.

All of these filesystems are POSIX-compliant and are fully compatible regardless of the Red Hat Enterprise Linux release. POSIX-compliant filesystems provide support for a well-defined set of system calls, such as read(), write(), seek(), etc. From the application programmer's point of view, there are relatively few differences. The most notable differences from a user's perspective are scalability- and performance-related. When considering a filesystem choice, you need to ask yourself how large the filesystem needs to be, what unique features you need it to have, and how fast it will be.

THE EXT FILESYSTEM FAMILY

Ext3 Filesystem

Ext3 is Red Hat's long-standing default filesystem for Red Hat Enterprise Linux. Ext3 was originally developed by Red Hat as a disk-format-compatible upgrade for the previous Ext2 filesystem. The major benefit of Ext3 is its support for journal-based transactions, which eliminates the need to run the filesystem repair tool after an ungraceful shutdown. Ext3 filesystems can be remounted as Ext2 filesystems without any conversion. Ext3 is well-tuned for general-purpose workloads and has been in use since the introduction of Red Hat Enterprise Linux 2.1.

Ext3 has long been the most commonly used filesystem not only for enterprise distributions, but for the development of many enterprise applications. If Ext3 works for your applications today, you can continue to use Ext3 or you can seamlessly move to Ext4, which was made available in the release of Red Hat Enterprise Linux 6 and will be discussed later in this paper.

On high-end storage devices, Ext3 has some limitations in that it can only scale to a maximum of 16TB. On a 1TB S-ATA drive, the performance of the Ext3 filesystem repair utility (fsck), which is used to verify and repair the filesystem after a crash, is extremely long. For many users that require high availability, the Ext3 filesystem typically supports close to 2-4TB of storage.

Ext4 Filesystem

Ext4 is the fourth generation of the Ext filesystem family and is the default filesystem in Red Hat Enterprise Linux 6. Ext4 has been offered in test previews since Red Hat Enterprise Linux 5.4, giving customers confidence in its maturity. Ext4 can read and write to Ext2 or Ext3 filesystems, but is not forward-compatible with Ext2 or Ext3. However, Ext4 adds several new and improved features that are common with most modern filesystems, such as:

- extent-based metadata

- delayed allocation

- journal check-summing

- large storage support

A more compact and efficient way to track utilized space in a filesystem is the usage of extend-based metadata and the delayed allocation feature. These features improve filesystem performance and reduce the space consumed by metadata. Delayed allocation allows the filesystem to postpone selection of the permanent location for newly written user data until the data is flushed to disk. This enables higher performance since it can allow for larger, more contiguous allocations, enabling the filesystem to make decisions with much better information.

Additionally, filesystem repair time (fsck) in Ext4 is much faster than in Ext2 and Ext3. Some filesystem repairs have demonstrated up to a six-fold increase in performance.

Currently, Red Hat's maximum supported size for Ext4 is 16TB in both Red Hat Enterprise Linux 5 and Red Hat Enterprise Linux 6. Application performance depends on many variables; in addition to the actual filesystem chosen, the application performance also depends on the specific I/O pattern the application generates and the type of server and storage hardware used.

THE XFS FILESYSTEM

XFS is a robust and mature 64-bit journaling filesystem that supports very large files and filesystems on a single host. As mentioned earlier in this paper, journaling ensures filesystem integrity after system crashes, for example, due to power outages, by keeping a record of filesystem operations that can be replayed when the system is restarted and the filesystem remounted. XFS was originally developed in the early 1990s by SGI and has a long history of running on extremely large servers and storage arrays. XFS supports a wealth of features including, but not limited to:

- delayed allocation

- dynamically allocated inodes

- b-tree indexing for scalability of free space management

- ability to support a large number of concurrent operations

- extensive run-time metadata consistency checking

- sophisticated metadata read-ahead algorithms

- tightly integrated backup and restore utilities

> "Red Hat also provides full support for the XFS filesystem as a layered, optional offering. The newest version of the Ext filesystem family, Ext4, is fully supported in Red Hat Enterprise Linux 6, as well as in Red Hat Enterprise Linux 5.6.

- online defragmentation

- online filesystem growing

- comprehensive diagnostics capabilities

- scalable and fast repair utilities

- optimizations for streaming video workloads

While XFS scales to exabytes, Red Hat's maximum supported XFS filesystem image is 100TB.

Given its long history in environments that require high performance and scalability, it is not surprising that XFS is routinely measured as one of the highest performing filesystems on large systems with enterprise workloads. For instance, a large system would be one with a relatively high number of CPUs, multiple HBAs, and connections to external disk arrays. XFS also performs well on smaller systems that have a multi-threaded, parallel I/O workload. XFS has relatively poor performance for single threaded, metadata-intensive workloads, for example, a workload that creates or deletes large numbers of small files in a single thread.

CHOOSING A LOCAL FILESYSTEM

How should you go about choosing a filesystem that meets your application requirements? The first step is to understand the target system on which you are going to deploy the filesystem. Ask some simple questions:

- Do you have a large server?

- Do you have large storage requirements or have a local, slow S-ATA drive?

- What kind of I/O workload do you expect your application to present?

- What are your throughput and latency requirements?

- How stable is your server and storage hardware?

- What is the typical size of your files and data set?

- If the system fails, how much down time can you suffer?

- Is your data integrity critical?

- Can you easily regenerate the data on a system crash?

Depending on the answers to some of the above questions, your choice can be obvious. If both your server and your storage device are large, XFS is likely to be the best choice. Even with smaller storage arrays, XFS performs very well when the average file sizes are large, for example hundreds of megabytes in size.

If your existing workload has performed well with Ext3, staying with Ext3 on Red Hat Enterprise Linux 5 or migrating to Ext4 on Red Hat Enterprise Linux 6 should provide you and your applications with a very familiar environment. Two key advantages of Ext4 over Ext3 on the same storage include faster filesystem check and repair times and higher streaming read and write performance on high-speed devices.

Another way to characterize this is that the Ext4 filesystem variants tend to perform better on systems that have limited I/O capability. Ext3 and Ext4 perform better on limited bandwidth (< 200MB/s) and up to ~1,000 iops capability. For anything with higher capability, XFS tends to be faster. XFS also consumes about twice the CPU-per-metadata operation compared to Ext3 and Ext4, so if you have a CPU-bound workload with little concurrency, then the Ext3 or Ext4

variants will be faster. In general Ext3 or Ext4 is better if an application uses a single read/write thread and small files, while XFS shines when an application uses multiple read/write threads and bigger files.

We recommend that you measure the performance of your specific application on your target server and storage system to make sure you choose the appropriate type of filesystem.

Red Hat Enterprise Linux 6 has new filesystem capabilities and performance characteristics. Key features that have been introduced in Red Hat Enterprise Linux 6 include support for the SSD "trim" command, support for thinly provisioned storage, and automated detection and alignment of new filesystems on many types of storage devices.

NETWORK FILESYSTEMS

Network filesystems, also referred to as client/server filesystems, will allow client machines to access files that are stored on a shared server. This makes it possible for multiple users on multiple machines to share files and storage resources. Such filesystems are built from one or more servers that export to one or more clients a set of filesystems. The client nodes do not have access to the underlying block storage, but rather interact with the storage using a protocol that allows for better access control. Historically, these systems have used L2 networking technologies like Gigabit Ethernet to provide a reasonably good performance for a set of clients.

The most common client/server filesystem for Red Hat Enterprise Linux customers is the NFS filesystem. Red Hat Enterprise Linux provides both an NFS server component that is used to export a local filesystem over the network, and an NFS client that can be used to import these filesystems.

Red Hat Enterprise Linux also includes a CIFS client that supports the popular Microsoft SMB file servers for Windows interoperability. To provide Windows clients with a Microsoft SMB service from a Red Hat Enterprise Linux server, Red Hat Enterprise Linux provides the user space Samba server.

SHARED STORAGE FILESYSTEMS

Shared storage filesystems, sometimes referred to as cluster filesystems, give each server in the cluster direct access to a shared block device over a local storage area network (SAN). Like client/server filesystems mentioned above, shared storage filesystems work on a set of servers that are all members of a cluster. Unlike NFS, no single server provides access to data or metadata to other members: each member of the cluster has direct access to the same storage device (the "shared storage") and all cluster member nodes access the same set of files.

Cache coherency is paramount in a clustered filesystem to insure data consistency and integrity. There must be a single version of all files in a cluster visible to all nodes within a cluster. In order to prevent members of the cluster from updating the same storage block at the same time that causes data corruption, shared storage filesystems use a cluster wide-locking mechanism to arbitrate access to the storage as a concurrency control mechanism. For example, before creating a new file or writing to a file that is opened on multiple servers, the filesystem component on the server must obtain the correct lock.

The requirement of cluster filesystems is to provide a highly available service like an Apache web server. Any member of the cluster will see a fully coherent view of the data stored in his/her shared disk filesystem and all updates will be arbitrated correctly by the locking mechanisms. Performance of shared disk filesystems is normally less than that of a local filesystem running on the same system since it has to account for the cost of the locking overhead.

Shared disk filesystems perform well with workloads where each node writes almost exclusively to a particular set of files that are not shared with other nodes, or where a set of files is to be shared in an almost exclusively read-only manner across a set of nodes. This results in a minimum of cross-node cache invalidation and can maximize performance. Setting up a shared disk filesystem is complex and tuning an application to perform well on a shared disk filesystem can be challenging.

For Red Hat Enterprise Linux customers, Red Hat provides the GFS1 filesystem in Red Hat Enterprise Linux 4 and Red Hat Enterprise Linux 5, and the GFS2 filesystem in Red Hat Enterprise Linux 5.4 and Red Hat Enterprise Linux 6. GFS1 is not supported in Red Hat Enterprise Linux 6, so customers will have to migrate their data to GFS2 before upgrading. GFS2 comes tightly integrated with the Red Hat Enterprise Linux High Availability Add-On.

Note that Red Hat Enterprise Linux supports both GFS1 and GFS2 on clusters that range in size from 2-16 nodes.

CHOOSING BETWEEN NETWORK AND SHARED STORAGE FILESYSTEMS

NFS-based network filesystems are an extremely common and popular choice for environments that provide NFS servers. Note that network filesystems can be deployed using very high-performance networking technologies like Infiniband or 10 Gigabit Ethernet. This means that users should not turn to shared storage filesystems just to get raw bandwidth to their storage. If speed of access is of prime importance, then use NFS to export a local filesystem like Ext4.

Shared storage filesystems are not easy to set up or to maintain, so users should deploy them only when they cannot provide their required availability with either local or network filesystems. Additionally a shared storage filesystem in a clustered environment helps reduce downtime by eliminating the steps needed for un-mounting and mounting that need to be done during a typical fail-over scenario that involves relocation of an HA service. We recommend the use of shared storage filesystems primarily for deployments that need to provide high-availability services with minimum downtime and have stringent service-level requirements.

CONCLUSION

Choosing the Red Hat Enterprise Linux filesystem that satisfies your specific application needs requires consultation of various parameters. This document was intended to outline the tradeoffs of various filesystem options and help users to make the decision around the right filesystem for your application environment. For additional information about filesystems for your IT environment, please contact Red Hat Support.