

# VxFS4.1 differences with VxFS3.5



|  |   |
|--|---|
| Introduction to VxFS4.1.....                   | 2 |
| Journaling.....                                | 2 |
| Scalability.....                               | 2 |
| Storage Checkpoints.....                       | 2 |
| New Features.....                              | 3 |
| <b>Multi-volume file systems (MVS)</b> .....   | 3 |
| <b>Quality of Storage Service (QoSS)</b> ..... | 3 |
| <b>Named Data Streams</b> .....                | 4 |
| <b>Cross-platform Data Sharing (CDS)</b> ..... | 4 |
| Disk Layout.....                               | 4 |
| Cluster File System.....                       | 4 |
| <b>Serviceguard integration</b> .....          | 4 |
| <b>Byte-range locking</b> .....                | 5 |
| <b>DMAPI</b> .....                             | 5 |
| <b>All new VxFS4.1 features</b> .....          | 5 |
| <b>I/O fencing</b> .....                       | 5 |
| For more information.....                      | 6 |
| Call to action.....                            | 6 |

## Introduction to VxFS4.1

HP-UX 11i version 2, released in September, 2003, was the newest version of HP-UX 11i for the Intel® Itanium® architecture. It was a full-fledged enterprise release of HP-UX 11i and contained the full range of HP systems management and availability software products. It was a major transition point for HP-UX 11i onto the Intel Itanium architecture, which offers large improvements in price/performance and performance scalability compared to current architectures.

In September, 2004 HP introduced an update to HP-UX 11i version 2 that is applicable to both the PA-RISC and Itanium architectures. This release is on a common DVD media for both the Itanium and PA-RISC architecture. Both HP-UX 11i version 2 and the September update provided VxFS3.5 as the default file system.

In June 2005, HP released VxFS4.1, the latest version of the VERITAS file system, for HP-UX 11i version 2 (Itanium and PA-RISC). It supports and contains a number of new features. We assume the reader is familiar with VxFS3.5, and we will be discussing the major differences between these two file systems.

### Journaling

VxFS provides fast recovery after system crashes through use of an intent log. It records pending changes to the file system structure in a circular intent log which is replayed during recovery to quickly ensure file system structural integrity. The default size of the intent log is based on the size of the file system. The maximum intent log size is 16MB with VxFS3.5. With VxFS4.1 and disk layout 6, the maximum has been increased to 256MB. The default maximum size for VxFS4.1 remains at 16MB. Increasing the size of the intent log can have a direct affect on the performance of the system. The larger the log, the fewer number of times the log need wrap around. However, the larger the log the longer it may take to replay on a recovery. With VxFS4.1 and disk layout 6, you can dynamically increase or decrease the size of the log (using the `fsadm log` option). It is often useful for improved performance to also configure the intent log on a different device than the file system. This minimizes head contention. QuickLog provided this capability for VxFS3.5. With VxFS4.1 and disk layout 6, QuickLog is not supported. The new way to accomplish this is with the multi-volume file system (MVS) feature (discussed later).

### Scalability

VxFS3.5 limits file system size to 32TB and file sizes to 2TB. HP-UX 11i v2 will support up to 2TB file sizes and 256TB file system sizes with VxFS4.1. File system sizes have been qualified at 32TB and will be increased on 11i v2, up to 256TB, based on customer needs. Larger file sizes will be supported with 11i v3.

### Storage Checkpoints

The Storage Checkpoint feature quickly creates a persistent point-in-time copy of the file system. You can roughly calculate the amount of storage required for the metadata based on the disk layout version. With disk layout 4 and 5, you calculated the minimum space required by multiplying the number of inodes in the file system by the size of an inode and then added 1-2MB. With disk layout 6, you simply take the number of inodes in the file system and add 1-2MB.

With VxFS3.5 checkpoints, the time required to create a checkpoint was proportional to the number of files in the file system. With 4.1, checkpoint creation is instantaneous and both the primary fileset and checkpoint are immediately available for use.

In addition, checkpoint rollback provides the ability to move the file system back to any previously taken checkpoint. The rolled back checkpoint becomes the new primary file system. This is an offline operation and requires the file system be unmounted.

Checkpoint quotas can be used to restrict the amount of space being used by checkpoints. Both soft and hard limits can be set.

## New Features

### Multi-volume file systems (MVS)

MVS provides flexibility to the customer to customize the mapping between their data requirements and the most appropriate choice of performance, availability, and cost available from their storage configurations. The granularity here is at the file system, directory, file, or individual storage checkpoint. The power of the flexibility is the clean separation of the storage allocation (i.e., where the file, directory, or storage checkpoint storage is allocated along with the underlying properties of that storage) and the file namespace (i.e., pathname). As an example, files contain different types of data. User data is pretty application specific; logs are typically large sequential writes; metadata is typically random small I/Os. Certain data might best be suited to striped mirrored storage; some data may best be suited for RAID-5; other files may only need lower cost, slower storage. With MVS, you can create a file system that spans volumes (up to 256 of them); dynamically add and remove volumes from a file system; separate the intent log, metadata, user data, and checkpoint data in any combination desired across the volumes; encapsulate a raw volume (e.g., used by a database) as a file in the file system so UNIX utilities like backup can work on it; and apply appropriate allocation policies to your data to match that data to the storage characteristics best suited for it. All of this is transparent to the application and the actual pathname of the file.

Note: MVS requires disk layout 6 and VxVM4.1; enabled via bundle.

### Quality of Storage Service (QoSS)

With MVS you can control where storage is allocated for a given file, directory, or checkpoint. However, the value of that data can change over time. With QoSS, built on top of MVS, you can further configure relocation policies to ensure these files are stored on the storage most appropriate to matching their characteristics at a given time. For example, today's application log may be accessed via large sequential writes but tomorrow it will be rarely accessed and only via reads. One might want it on a striped mirrored volume today but a lower cost, slower, non-mirrored storage tomorrow. With QoSS, you can set rules that can use combinations of size, location, name pattern, and other attributes of the file to determine relocation. You can specify multiple rules, applied in order until one matches. That rule's relocation action is then applied. You can also do "what if" scenarios to understand and analyze impact before setting policy. The file system remains mounted, applications can still be accessing the files while the relocations are taking place.

These relocation policies can be applied on a checkpoint, file system, directory (with or without inheritance), or file granularity. The policies determine what should be relocated and are driven by two utilities that are usually set up in **cron** for automated periodic application.

Note: QoSS requires MVS; enabled via bundle.

## Named Data Streams

Named data streams provides a means for an application to store persistent application specific data (e.g., attributes) with a file. Traditional UNIX files have an inode identifier and a single stream of file data. Using named data streams, the inode is retained, but can now be associated with multiple data streams. In VxFS4.1, the original data stream is accessed in the same way as in previous releases, but other data streams are referenced through a new directory inode associated with the file. The directory inode points to the new inodes, one per stream. The directory containing the named streams is not directly visible to the user.

A programmatic interface is provided for the feature.

Note: Named data streams requires disk layout 6; enabled in VxFS.

## Cross-platform Data Sharing (CDS)

CDS provides a means for the serial sharing of a VxFS file system across heterogeneous platforms that have direct access to the physical devices that contain the data. This may be useful for migrating from one platform to another (e.g., Solaris or AIX to HP-UX) or for the serial processing of data across multiple platforms (e.g., HP-UX and Linux). Optionally, you can run a validation utility to check limits – e.g., the destination platform supports file and file system sizes needed, uid ranges etc. The general process is to unmount the file system; Deport the disks; Make the storage accessible to the destination (import the disks); Mount it and use it. NOTE: application data is left as an application issue. If you are moving to/from a different Endian platform – e.g., to/from Linux, you will need to convert the file system structural information to that Endianess by running a utility.

CDS is also referred to as Portable Data Containers (PDC).

Note: CDS requires VxVM4.1 and disk layout 6; enabled via bundle.

## Disk Layout

VxFS4.1 supports disk layouts 4, 5, and 6 (new and default). Disk layout 3 is no longer supported. Many of the new VxFS4.1 features and future scalability beyond a 32TB file system size require disk layout 6. With disk layout 6, any application that uses the statvfsdev(3C) family of interfaces (statvfsdev, fstatvfsdev, statvfsdev64, fstatvfsdev64, statfsdev, fstatfsdev) must relink with these routines. These routines have been updated with knowledge of Disk Layout 6 and the old routines would fail to recognize the presence of Disk Layout 6 on the disk. Also, the default mkfs/newfs/mount option for VxFS4.1 disk layout 6 is “largefiles.” The default for disk layout 4 and 5 remain as “nolargefiles.” This means that files, as a default, can grow beyond 2GB on disk layout 6. You may want to specify “nolargefiles” when you mount file systems on which you plan on running 32-bit applications or stay with disk layout 5 if you do not need to use the new functionality introduced with disk layout 6.

## Cluster File System

Through use of the cluster file system, you can concurrently share file systems and files between nodes in the cluster. There are several new extensions to CFS.

## Serviceguard integration

VERITAS CFS has been integrated into Serviceguard for high-availability cluster support. Serviceguard version 11.17 is required. A mixed environment of applications that use CFS and don't is supported. Simple changes to the Serviceguard package control script will



allow an existing application that does not currently use CFS to start using it. Oracle single instance, Oracle RAC, and non-Oracle HA environments that want to transparently share files across nodes in the cluster are supported.

### **Byte-range locking**

Version 3.5 of CFS locked access at file granularity. Version 4.1 locks access at byte range granularity. This means that application access to non-overlapping blocks in the same file from different nodes will not be serialized.

### **DMAPI**

DMAPI (Data Management API) was supported only locally in version 3.5. With version 4.1, DMAPI is supported across the cluster.

### **All new VxFS4.1 features**

Essentially, all VxFS3.5 and VxFS4.1 features are supported across the cluster.

### **I/O fencing**

This capability insures that there is no data corruption in a “split brain” (network partition) situation.

CFS is enabled via bundle.

## For more information

[www.hp.com/go/somewhere](http://www.hp.com/go/somewhere)

## Call to action

[www.hp.com/go/somewhere](http://www.hp.com/go/somewhere)