z/OS Basics: z/OS UNIX Shared File System environment and how it works

Jim Showalter
IBM

March 1, 2011
Session 9024

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

- DFS
- DFSMS
- DFSMSdss
- IBM
- MVS
- RACF
- RMF
- S/390
- z/OS
- zSeries

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

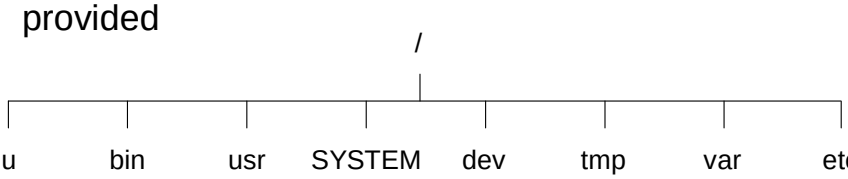
Agenda

- Single System z/OS UNIX
- Multiple Systems with DASD sharing
- The z/OS UNIX Shared File System Environment
- Multiple Systems in a shared file system environment

Single System z/OS UNIX

Introduction to z/OS UNIX

- z/OS UNIX is an element of z/OS that provides a UNIX environment consisting of:
  - an Application Programming Interface (API), and
  - an interactive shell interface
  - TSO/E commands for file system management
  - ISPF shell environment (ishell)
- As a part of z/OS UNIX, a hierarchical file system is provided



5

The application programming interface of z/OS UNIX supports C callable services (documented in the z/OS XL C/C++ Run-Time Library Reference) and Assembler callable services (documented in the z/OS UNIX System Services Programming: Assembler Callable Services Reference).

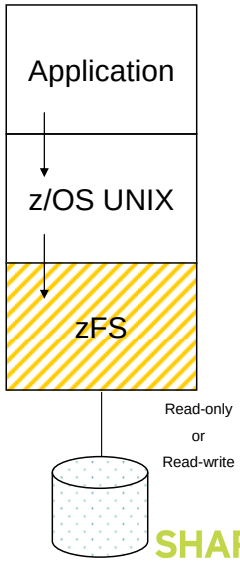
The interactive shell interface provides a UNIX command interface (commands are documented in the z/OS UNIX System Services Command Reference).

The TSO/E commands are also documented in the same book.

The ISPF shell is documented in the z/OS UNIX System Services User's Guide.

Introduction to z/OS UNIX …

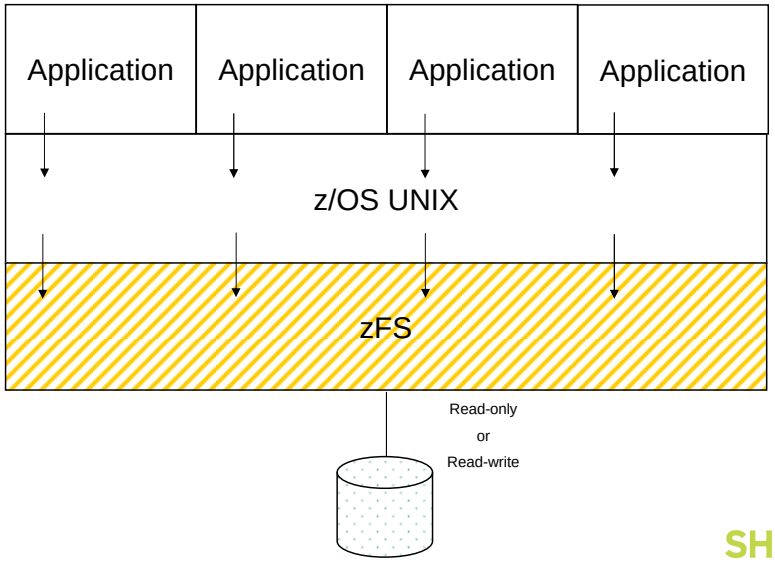
- In z/OS, applications run in their own address space
- When an application invokes a z/OS UNIX API, control is passed into the z/OS UNIX address space
- If the z/OS UNIX API is a file request, application control is passed into the z/OS UNIX address space and then from there into the Physical File System (PFS) address space
- The PFS address space (in this case, zFS) handles reading and writing data to the actual disk and presents the data as a hierarchical file system
- All users on the system can access the same data in the file system (based on their authorization to the data)



6

zFS runs in its own address space (sometimes referred to as a Colony Address Space). z/OS UNIX application requests program call into the z/OS UNIX address space and when z/OS UNIX determines that the request is for a zFS file system, z/OS UNIX program calls into the zFS address space. The zFS address space handles all I/O to the file system on DASD.

Introduction to z/OS UNIX …

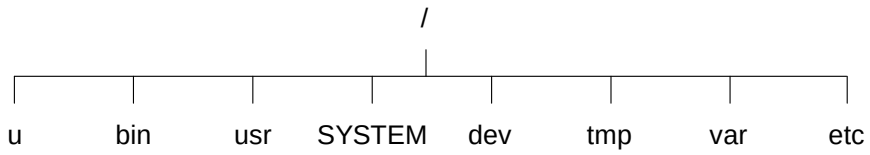


7

Of course, there can be multiple applications accessing z/OS UNIX and zFS at the same time, so this is a bit more accurate. For simplicity, I will show only a single application in the remaining pictures, but you should understand that there can be many applications.

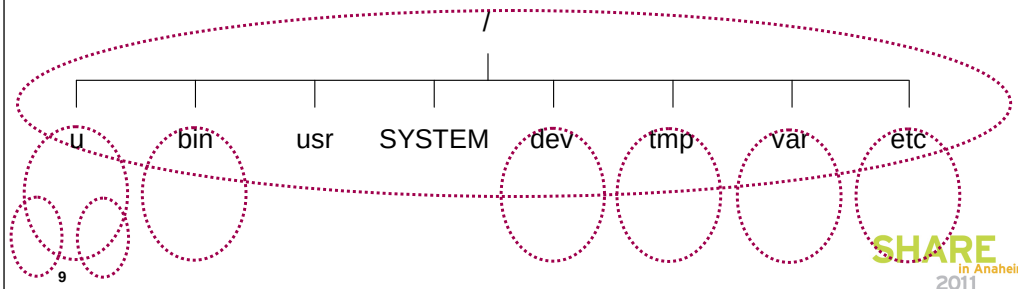
The z/OS UNIX file system

- This is how an end user views the z/OS UNIX hierarchical file system
- It looks like files and directories organized in a hierarchical tree



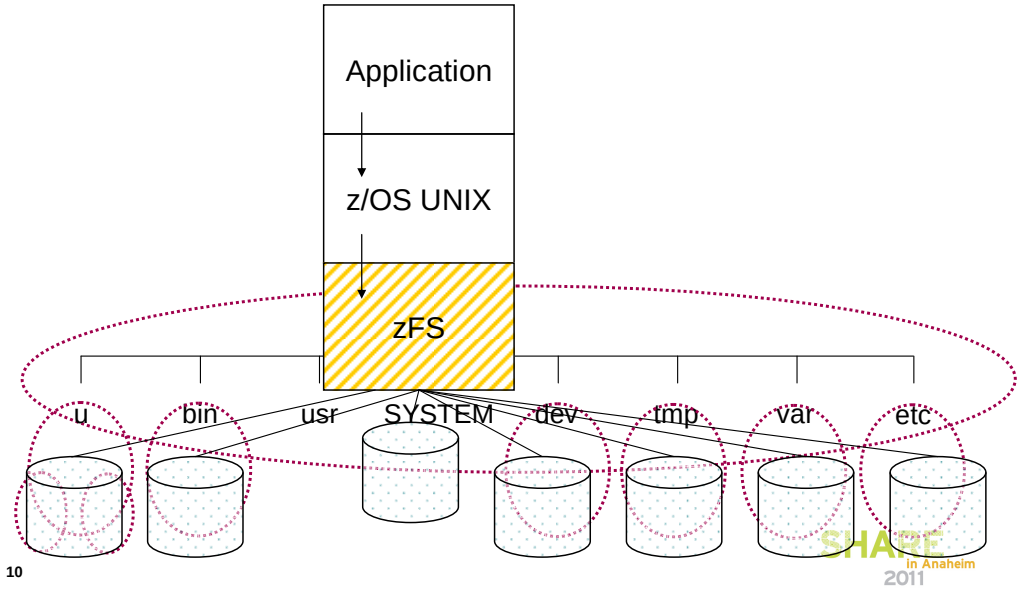
The z/OS UNIX file system …

- The system programmer has to know that the hierarchical file system actually consists of many individually mounted file systems
- Each file system is (usually) a z/OS data set
- The hierarchical file system consists of a root file system with other file systems mounted on directories in the root file system



9

The z/OS UNIX file system …



The z/OS UNIX file system …

- The z/OS UNIX file system is configured by specifications in the BPXPRMxx Parmlib members
  - BPXPRMxx contains
    - Mount statements for root and file systems below
  - BPXPRMxx list is specified in IEASYS00 as OMVS=(xx,yy,…)
  - When a file system is mounted, zFS allocates and opens the data set

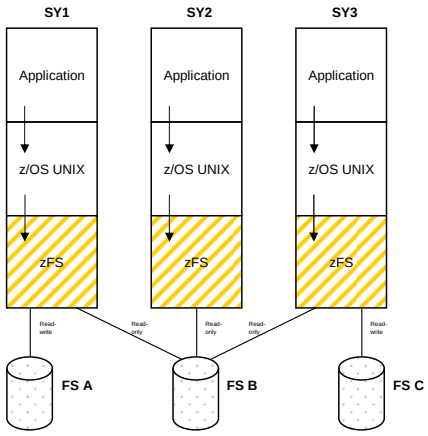
The logical parmlib concatenation is a set of up to ten partitioned data sets defined by PARMLIB statements in the LOADxx member of either SYSn.IPLPARM or SYS1.PARMLIB (quote from z/OS MVS Initialization and Tuning Reference).

BPXPRMxx also contains the FILESYSTYPE statement for zFS and other PFSes – defines the TYPE (for example, ZFS) and the load module that contains the zFS code.

Multiple Systems with DASD sharing

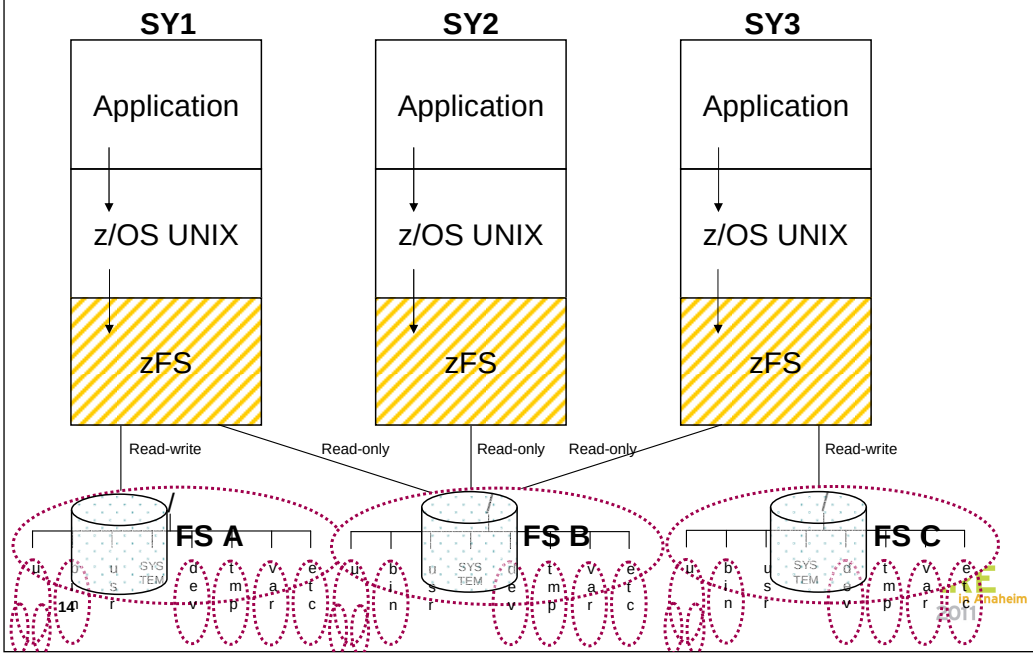
Multiple systems with DASD sharing

- When you are logged on to a particular system (for example, SY1), you can only access a file system that is mounted on that system
- A file system can be read-only mounted on all systems, so it can be accessible from all systems
- A file system can only be read-write mounted on one system, so it can only be accessed from the one system it is mounted on
- To update (or read from) a read-write mounted file system you must log on to the system that it is mounted on



If you try to mount a zFS read-write file system on multiple systems, zFS will deny the second mount.

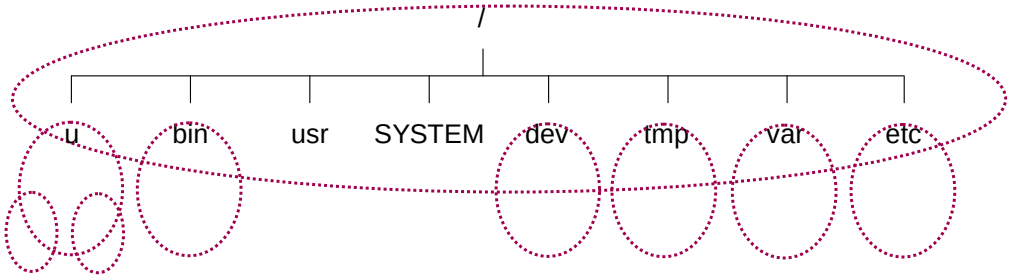
Multiple systems with DASD sharing …



The Shared File System Environment

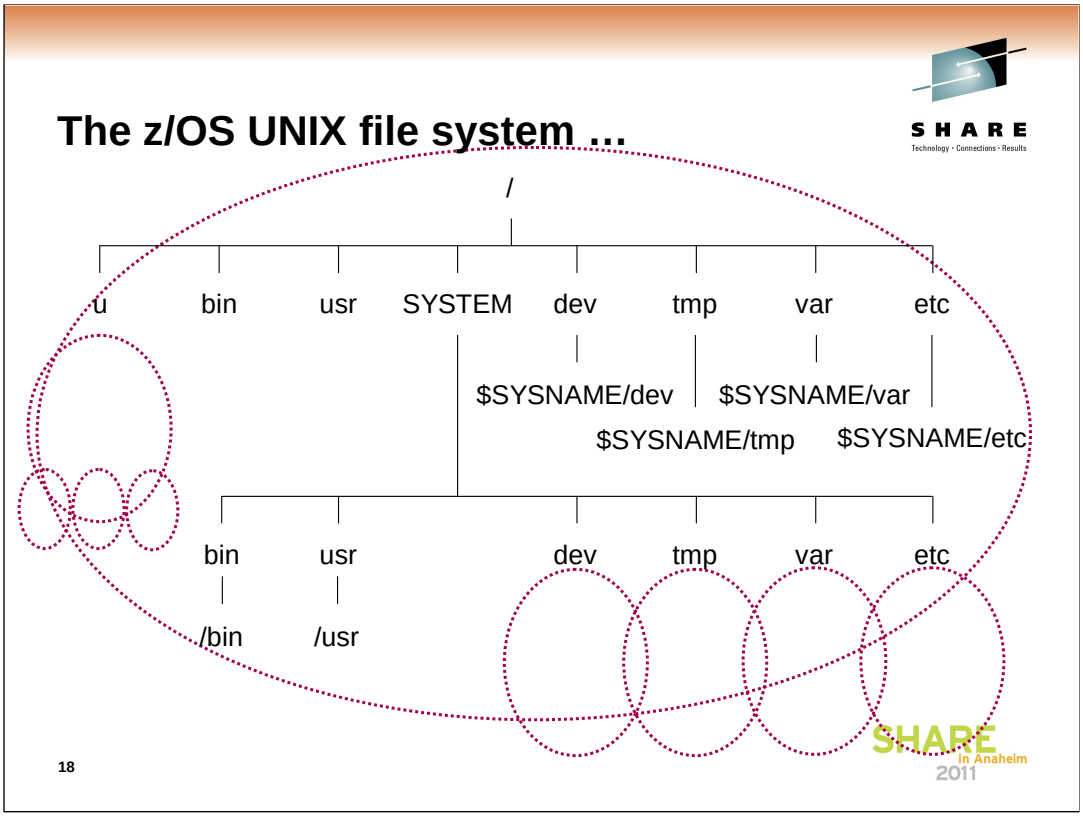
The z/OS UNIX file system

- Again, here is what the z/OS UNIX file system looks like to the end user (and showing the mounted file systems)



The z/OS UNIX file system …

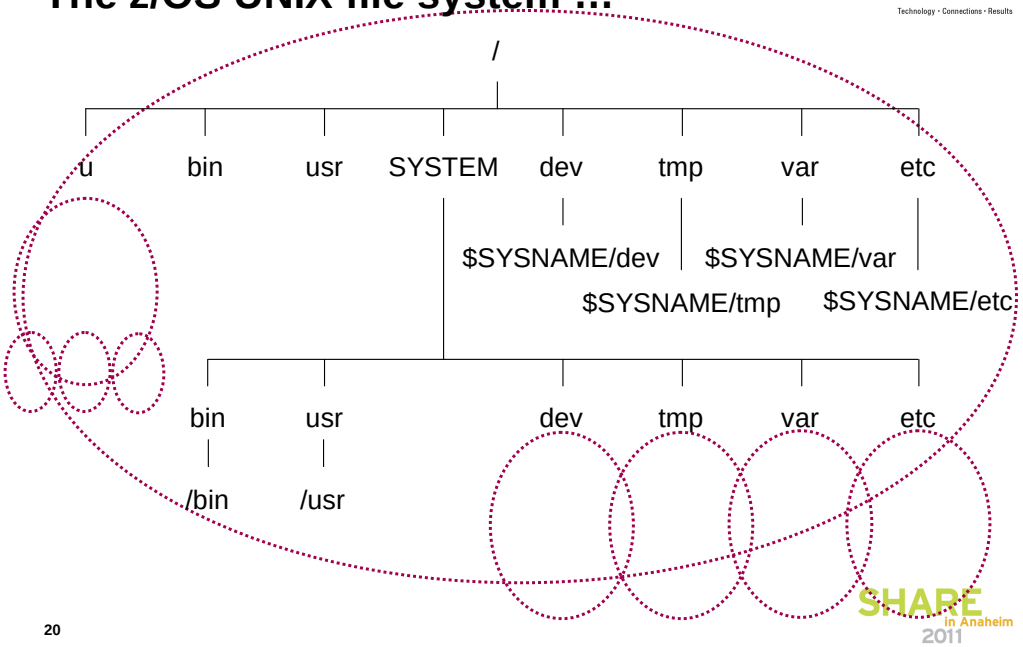
- The z/OS hierarchical file system is actually a bit more involved than the previous slide shows
- The sysplex shared file system environment needs to support multiple concurrent z/OS releases and even multiple concurrent service levels for different LPARs in a single file system hierarchy
- We also want to support system specific file systems
- The sysplex shared file system environment uses special mount points and symbolic links with system symbols to provide this
- Even a single system uses symbolic links to allow easy transition to the shared file system environment
- But, the end user view of the z/OS UNIX hierarchical file system does not change whether they are in a sysplex environment or not



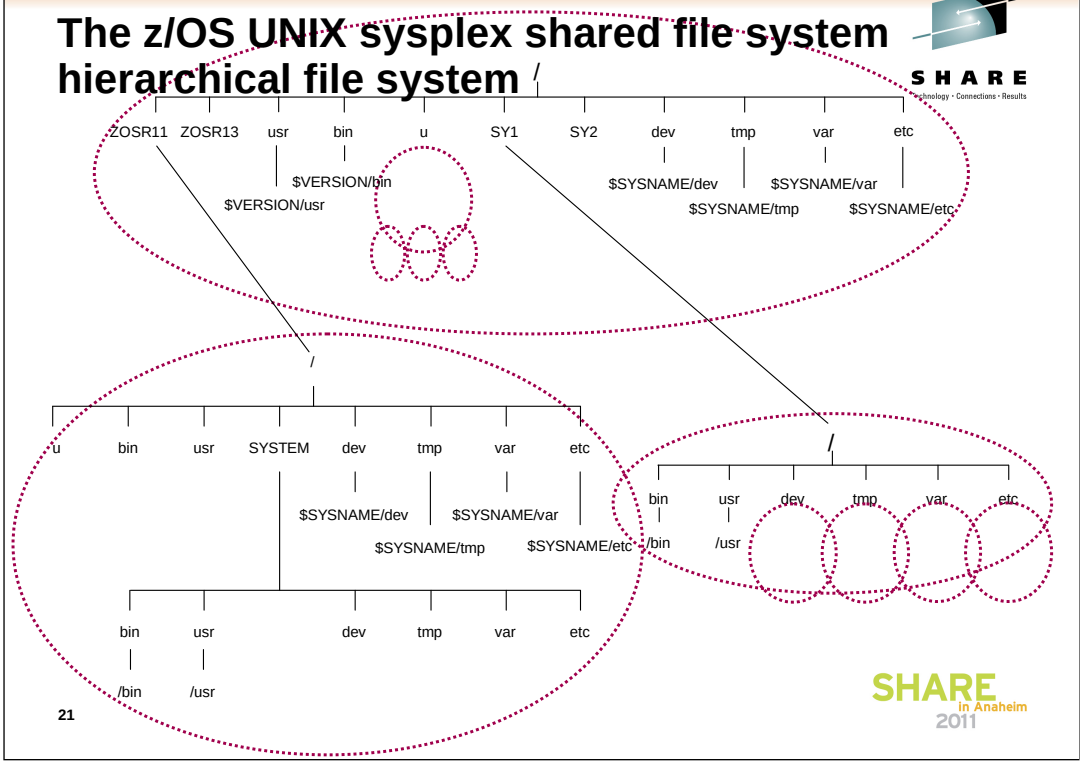
If the content of the symbolic link begins with $SYSNAME and SYSPLEX is specified NO, then $SYSNAME is replaced with /SYSTEM when the symbolic link is resolved.

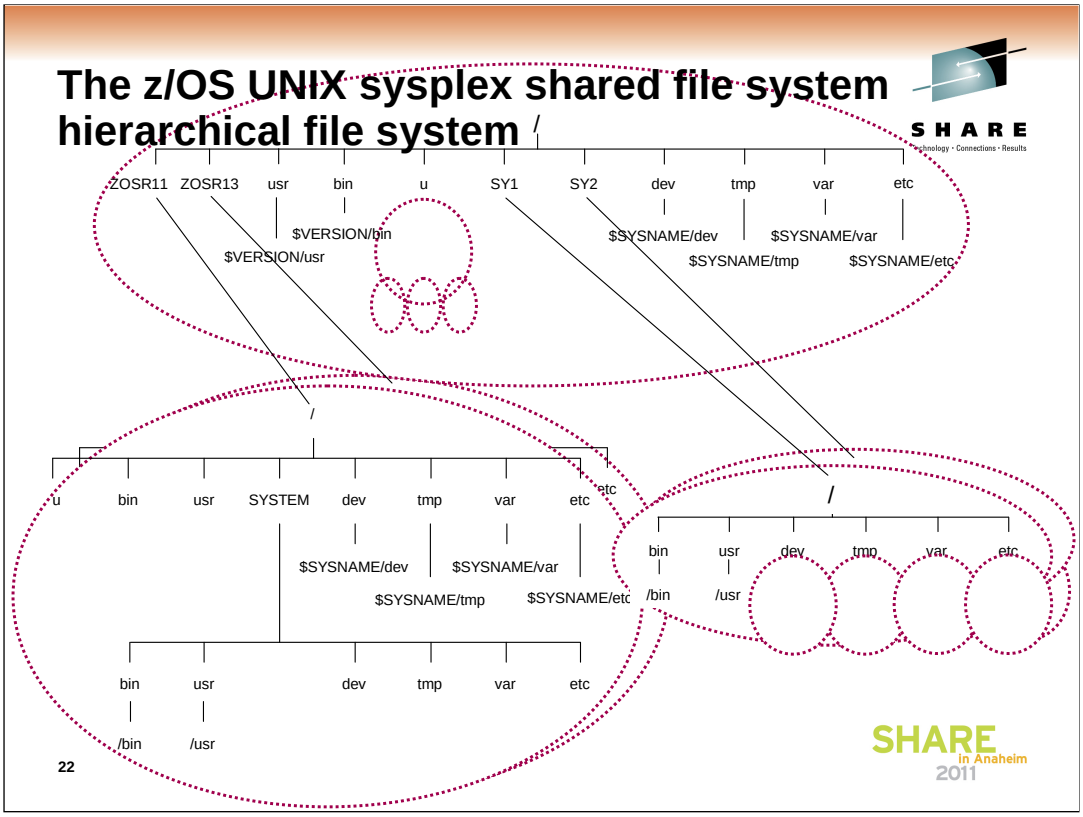

SHARE
Technology • Connections • Results
Multiple Systems
in a
Shared File System
Environment
19
SHARE
in Anaheim
2011

The z/OS UNIX file system …



The z/OS UNIX sysplex shared file system hierarchical file system





From SY1 (running z/OS V1R11)

df /bin

Mounted on Filesystem Avail/Total Files Status

/ZOSR11 (OMVS.MNT.ZOSR11.ZD1111.ZFS) 26236/4579200 4294950685 Available

From SY2 (running z/OS V1R13)

df /bin

Mounted on Filesystem Avail/Total Files Status

/ZOSR13 (OMVS.MNT.ZOSR13.ZD1131.ZFS) 1113638/5760000 4294951449 Available

Anyone can access a different system's /etc by using a full pathname such as /SY1/etc

Anyone can access a different release of /bin by using a full pathname such as /ZOSR13/bin

The z/OS UNIX file system in a shared file system environment

- The z/OS UNIX file system is configured by specifications in the BPXPRMxx Parmlib members
  - xx list is specified in IEASYS00 as OMVS=(xx,yy)
  - BPXPRMxx contains
    - Mount statements for root and lower file systems
      (At IPL, if a file system is already mounted, this is accepted silently)
    - SYSPLEX(YES) specifies shared file system environment
    - In this case a z/OS UNIX CDS (Couple Data Set) is required
    - VERSION('ZOSR13') specifies the value of $VERSION
      (When SYSPLEX(YES) is specified, you must specify VERSION)
  - IEASYMxx contains
    - SYSDEF SYSNAME(SY1) specifies the value of $SYSNAME
    - If the content of the symbolic link begins with $SYSNAME and SYSPLEX is specified NO, then $SYSNAME is replaced with /SYSTEM when the symbolic link is resolved.
  - When a file system is mounted, zFS allocates and opens the data set

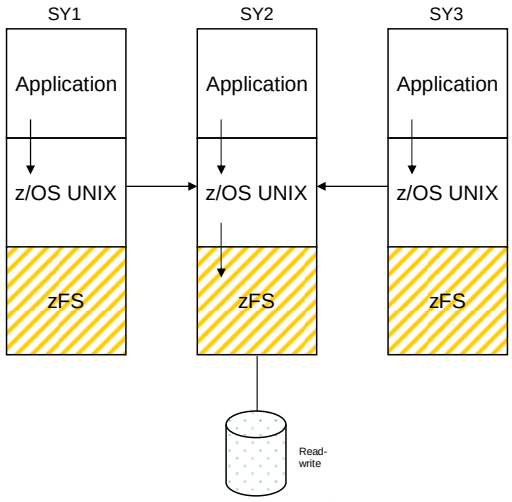
23

You can use System Symbols in Parmlib members. For example,

In IEASYS00, you can say OMVS=(&SYSCLONE.,01)

The z/OS UNIX shared file system environment

- In a parallel sysplex environment, z/OS UNIX can provide access to the entire file system hierarchy for all users from all LPARs in the sysplex
- This environment is called a **shared file system environment**
- z/OS UNIX provides this support by forwarding file requests from other LPARs (SY1 or SY3) to the LPAR designated as the z/OS UNIX file system owning LPAR (SY2 in this case)
- When you mount a file system, an owning LPAR is designated and the file system is mounted and available to all LPARs in the shared file system environment



Benefits of the shared file system environment

- **System independence**
  All z/OS UNIX data can be accessed from any system in the sysplex
- **Availability**
  If a system is shutdown or if it abnormally goes down, file system ownership movement occurs automatically and file systems remain accessible from the other systems (although temporary failures may be visible to applications during abnormal shutdown)
- **Flexibility**
  General users and applications automatically access the correct (system specific and/or version/release) file systems while administrators can access any file system
- **Transparency**
  Users and applications do not need to change to run in a shared file system environment (except to possibly handle and recover from the temporary failures)

Publications

- z/OS UNIX System Services Planning (GA22-7800)
  General Administration of z/OS UNIX file systems
- z/OS UNIX Command Reference (SA22-7802)
  confighfs command for HFS
- z/OS MVS System Messages Volume 9 (IGF-IWM) (SA22-7639)
  IGWxxxt messages for HFS
- z/OS UNIX System Services Messages and Codes (SA22-7807)
  z/OS UNIX return codes, z/OS UNIX reason codes, X'5Bxxrrrr' reason codes for HFS
- z/OS Distributed File Service zSeries File System Administration (SC24-5989) – **was refreshed in April 2010**
  zFS Concepts and zfsadm command for zFS
- z/OS Distributed File Services Messages and Codes (SC24-5917)
  IOEZxxxt messages and X'EFxxrrrr' reason codes for zFS
- z/OS Distributed File Service zSeries File System Implementation (SG24-6580)
  - Redbook available (updated February 2010 to include z/OS V1R11)
  - http://www.redbooks.ibm.com/abstracts/sg246580.html?Open
- z/OS Version 1 Release 8 Implementation (SG24-7265)
  - Redbook available (contains zFS updates for z/OS V1R8)
  - http://www.redbooks.ibm.com/abstracts/sg247265.html?Open
- z/OS DFSMS™ Access Method Services for Catalogs (SC26-7394)
  IDCAMS utility
- z/OS DFSMS™ Storage Administration Reference (SC26-7402)
  ADRDSSU utility for backup

Other related SHARE presentations

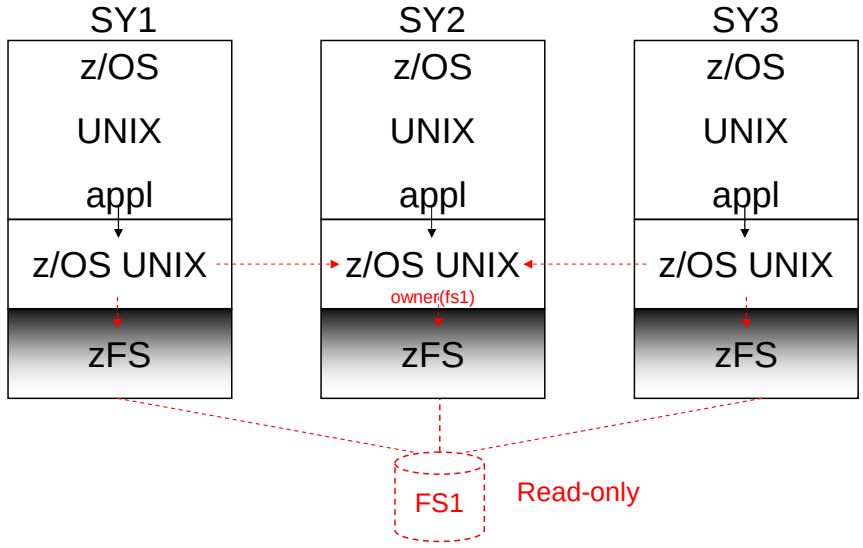
- Session 9041: z/OS UNIX Shared File System Configuration – Overview and New Features
  Wednesday, March 2, 2011 4:30-5:30 PM
- Session 9037: z/OS Basics: Migrating from HFS to zFS and Things to watch out for
  Thursday, March 3, 2010 9:30-10:30 AM
- Session 9040: z/OS UNIX File System Administration
  Friday, March 4, 2011 9:30-10:30 AM


SHARE
Technology • Connections • Results

Backup


SHARE
in Anaheim
2011

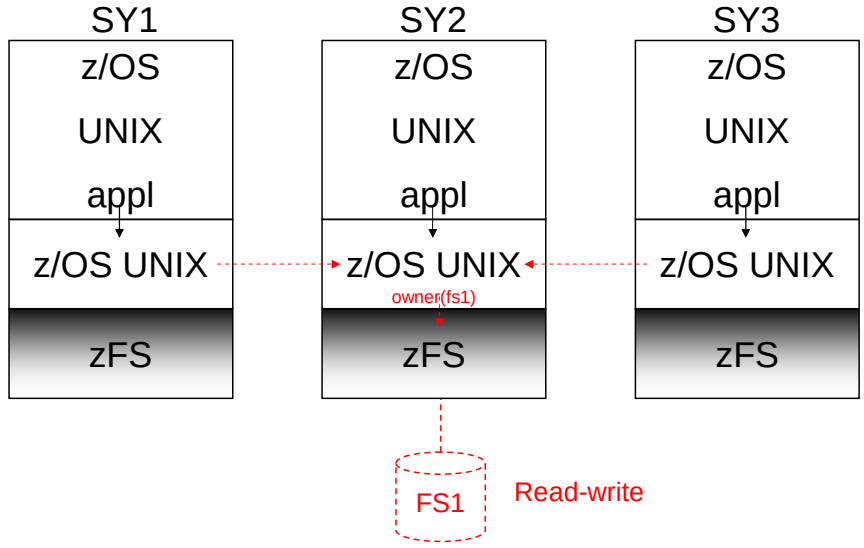
Read-only mounted file system in a shared file system environment



29

This shows a shared file system environment in a sysplex (z/OS UNIX BPXPRMxx specifies SYSPLEX(YES)). When a zFS file system is mounted read-only, it is locally mounted on each system. Read-only mounted file systems are always sysplex-aware. This has always been the case for shared file system environments. There is no communications between sysplex members to access a read-only mounted file system.

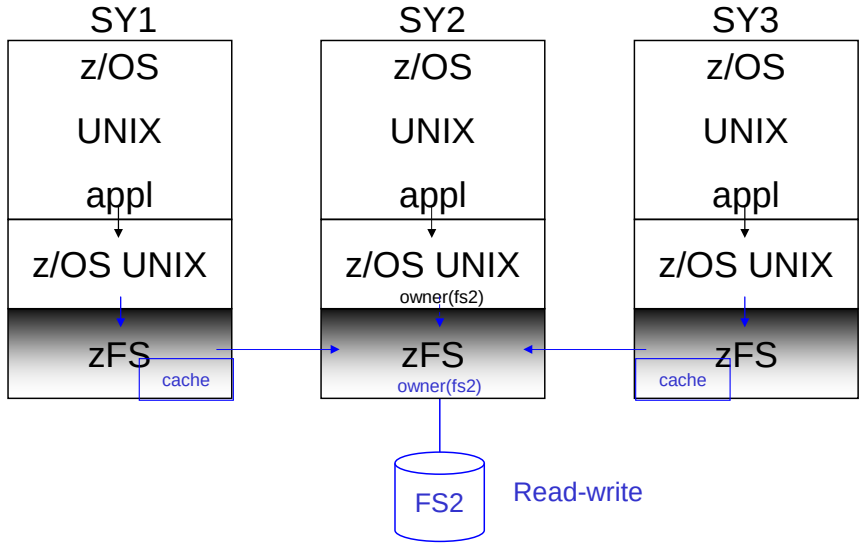
Read-write mounted file system in a shared file system environment



30

In contrast to a read-only mounted file system, this is a read-write mounted file system. One system is the owning system and the other systems are “clients”. The file system is only locally mounted on the owning system. (It is still externally mounted and available on all systems.) Applications running on the owning system (SY2) access the file system locally (without any XCF communications). But applications that run on the other systems (SY1 and SY3), access the file system through the use of z/OS UNIX function shipping (using XCF communications). That is, the request is forwarded by the z/OS UNIX on that system (SY1 or SY3) to the z/OS UNIX running on the owning system (SY2) which then program calls the local zFS. The response goes back along the same path. So, access to the file system from systems other than the owner (that is, from the client systems) involves XCF communications. This makes it important to have the z/OS UNIX owning system, be the system that is doing the most accesses.

Sysplex-aware read-write mounted file system in a shared file system environment



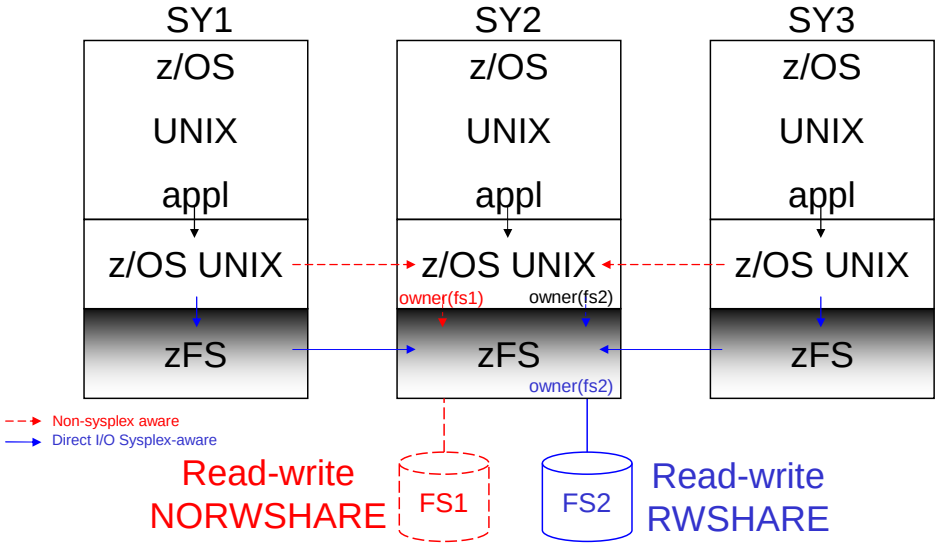
31

Here is a picture of the new R11 zFS sysplex-aware for read-write support. When zFS runs sysplex-aware (for read-write) on all systems, a read-write mounted sysplex-aware file system is locally mounted on all systems. There is still a z/OS UNIX owning system but there is no z/OS UNIX function shipping to the owner. Rather, requests from applications on any system are sent directly to the local zFS on each system. This means it is now the responsibility of the local zFS to determine how to access the file system. One of the systems is known as the zFS owning system. This is the system where all I/O to the file system is done. zFS uses function shipping to the zFS owning system to access the file system. (If this was all that zFS did, it would be essentially the same as the z/OS UNIX function shipping as shown in the previous slide.) However, each zFS client system has a local cache where it keeps the most recently read file system information. So, in many cases (when the data is still in the cache), zFS can avoid the zFS function shipping (and the XCF communications) and satisfy the request locally.

Sysplex-aware read-write mounted file system in a shared file system environment

- Local mount
  This means that the mount request was sent by z/OS UNIX to the local zFS. The local zFS has the data set open. A **df -v** command for the file system would show Client=N. A file system is always locally mounted on the z/OS UNIX owning system.
- Global mount
  This means that the file system is available from any system in the shared file system environment. The file system is locally mounted on the z/OS UNIX owning system and may or may not be locally mounted on the other systems. A file system mounted in a shared file system environment is always globally mounted.

Sysplex-aware and non-sysplex aware read-write mounted file systems in a shared file system environment (R11 zFS APAR OA29619)



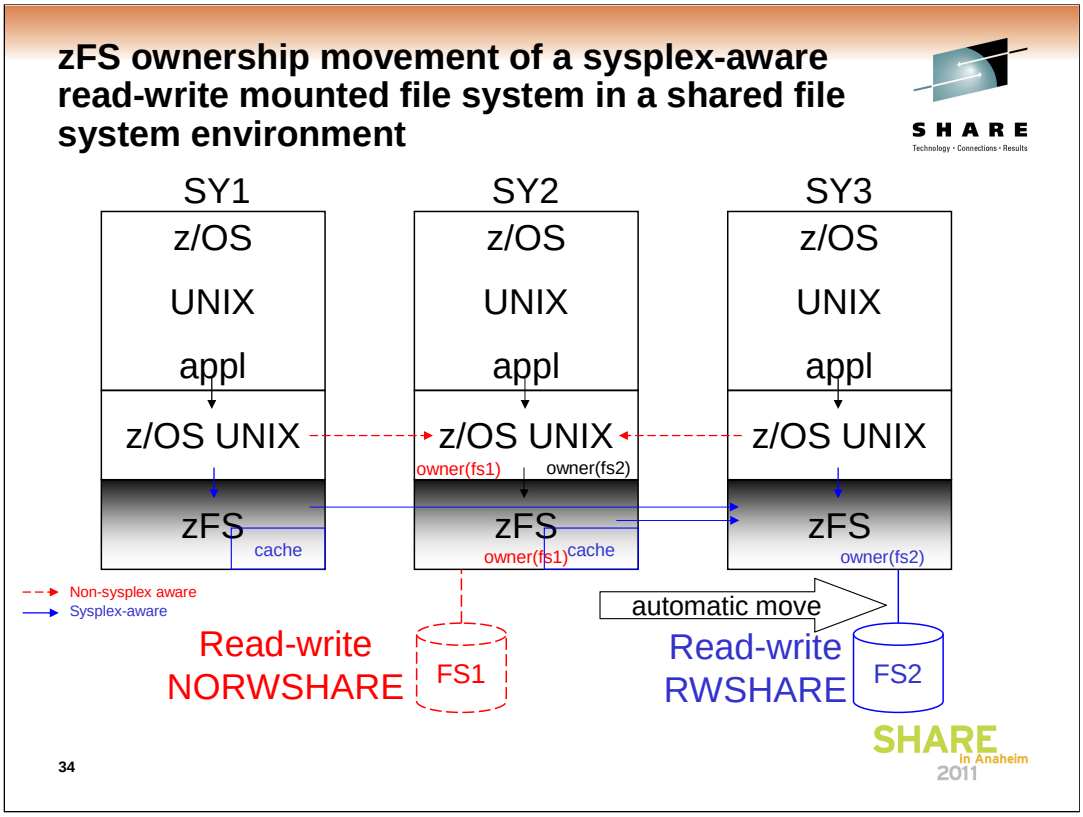
33

SHARE in Anaheim 2011

Here are two zFS read-write file systems on a sysplex running zFS sysplex-aware on file system basis on all members. One file system (FS1) is mounted NORWSHARE and the other (FS2) is mounted RWSHARE. They are both z/OS UNIX owned on SY2. The NORWSHARE file system (FS1) acts like a non-sysplex aware file system (it is only locally mounted on the z/OS UNIX owner (SY2) and requests from z/OS UNIX clients (SY1 and SY2) are function shipped to the z/OS UNIX owner (SY2) by z/OS UNIX). A df –v command for the NORWSHARE file system (FS1) from SY1 would display Client=Y.

The other file system (FS2) is mounted RWSHARE. It acts like a sysplex-aware file system (it is locally mounted on all systems and z/OS UNIX never function ships requests to the z/OS UNIX owner (SY2)). A df –v command for the RWSHARE file system (FS2) from SY1 would display Client=N.

When you run zFS sysplex-aware on a file system basis on all your members, the zFS Physical File System initializes as sysplex-aware but it can individually determine which file system is sysplex-aware and which is not based on the RWSHARE/NORWSHARE mount parm.

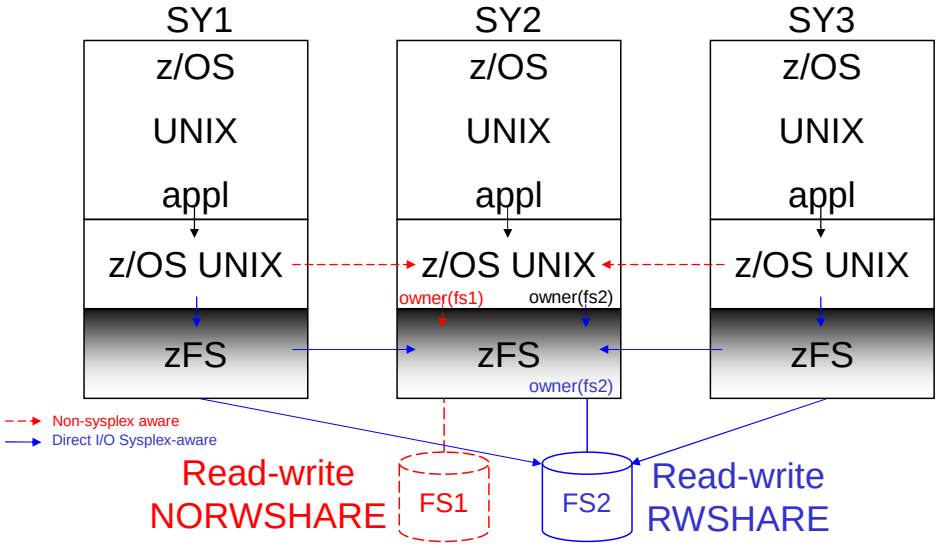


In addition, zFS can move its ownership of zFS read-write sysplex-aware file systems dynamically based on system usage among zFS systems where the file systems are locally mounted. If one sysplex member is a better fit for ownership (because more requests to the file system are being done there than any other sysplex member), then zFS will dynamically move zFS ownership to that system. zFS will also move ownership when the owning system is shutdown, or an abnormal outage occurs on the zFS owning system. Thus the z/OS UNIX owner and the zFS owner can be two entirely different systems depending on the sequence of events. This is normal, and for a sysplex that is using the zFS sysplex=filesys support on all sysplex members this should have no negative consequences.

zFS does not have any external commands to move zFS ownership between systems. It does provide query commands to show the zFS owner, such as **zfsadm lsaggr**.

Note that zFS ownership movement based on performance criteria is non-disruptive.

Direct I/O sysplex-aware read-write mounted file system in a shared file system environment



35

In zFS R13, zFS supports direct I/O for sysplex-aware read-write file systems. This means that when all systems are running z/OS V1R13, zFS can directly read and write user data to a sysplex-aware (RWSHARE) read-write file system (for example, FS2). This generally improves the performance of client system access to the files (from SY1 and SY3) since the data does not need to pass through SY2 with XCF communications. Only metadata updates are sent to the zFS owning system.

Adding a new system or version root

- In general, the contents of the sysplex root should only change when you need a new version root or system-specific root file system directory for your shared file system configuration. When a system is IPLed (initialized), the mount processing for the sysplex root file system will include defining the appropriate $SYSNAME or $VERSION directory in the sysplex root file system if the sysplex root is mounted as read/write. Assuming that you have the sysplex root file system mounted as read-only, the procedure to use to create a new version root or system-specific file system directory is as follows:
  - Remount the sysplex root file system to read-write
  - IPL the new system
  - Remount the sysplex root file system to read-only