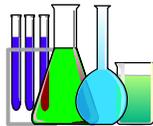


z/OS V1R2, z/OS V1R3, and z/OS V1R4

z/OS Distributed File Service (DFS) zSeries File System (zFS)



zFS File Systems



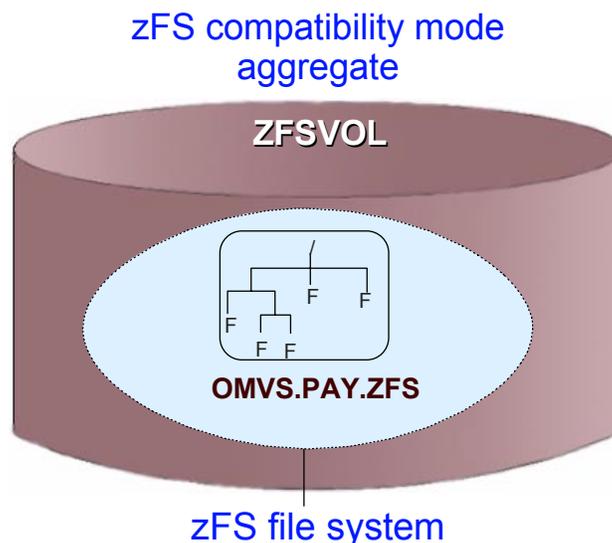
- ❑ **zFS is a "new" Unix File System for z/OS**
 - A component of the Distributed File Service since 1995
 - Does not replace HFS
- ❑ **Using zFS, you can**
 - Run applications just like HFS
 - Use zFS in addition to HFS or replace HFS
- ❑ **Advantages:**
 - Better performance
 - Enhanced administrative functions
 - Less loss of data on system failures

zFS Aggregates - (Terminology)



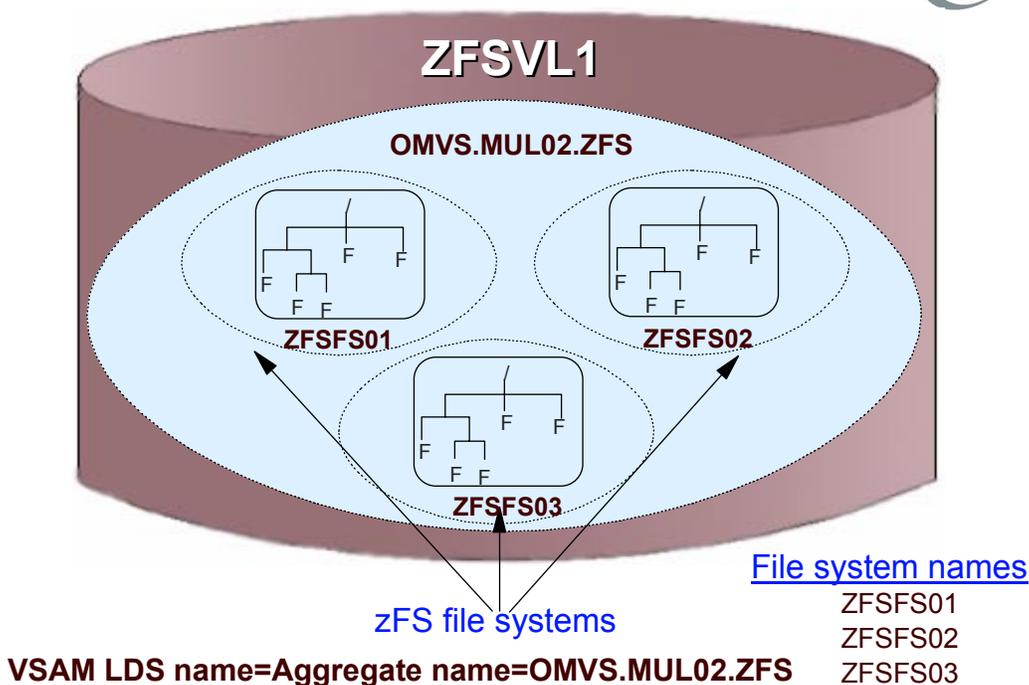
- ❑ An **aggregate** is a VSAM linear data set (LDS)
- ❑ An aggregate contains one or more zFS file systems
- ❑ Two types of aggregates:
 - HFS compatibility mode - contains 1 zFS file system
 - Multiple file system mode - contains 1 or more zFS file systems
 - Space sharing between file systems in same aggregate
- ❑ File system maximum size as a logical value
- ❑ File system **clone** (making a read/only copy)
- ❑ zFS File System
 - Can be mounted within the HFS root file system
 - Contains a logical maximum size known as its **quota**

zFS Compatibility Mode Aggregate



VSAM LDS name=Aggregate name=File system name=OMVS.PAY.ZFS

Multiple File System Aggregate

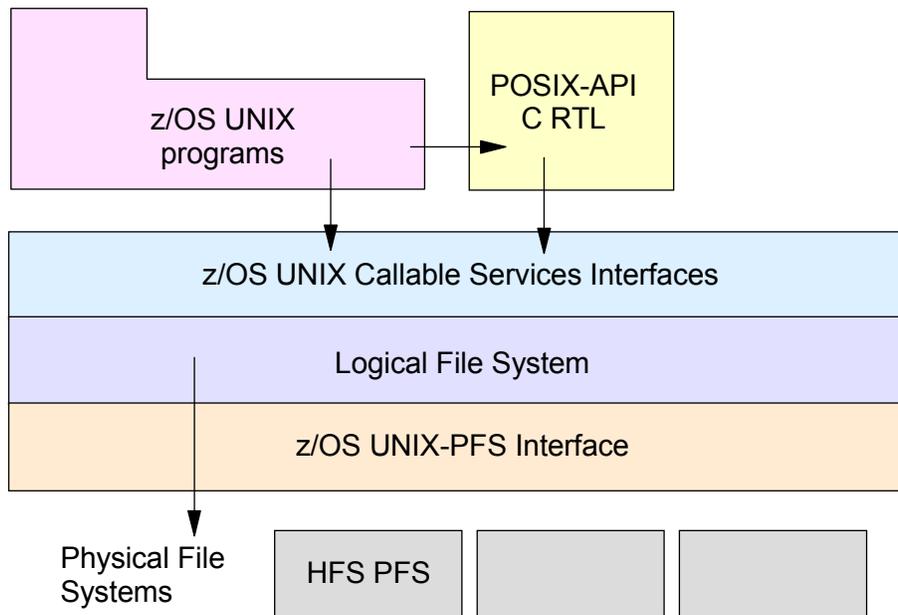


Installation and Documentation

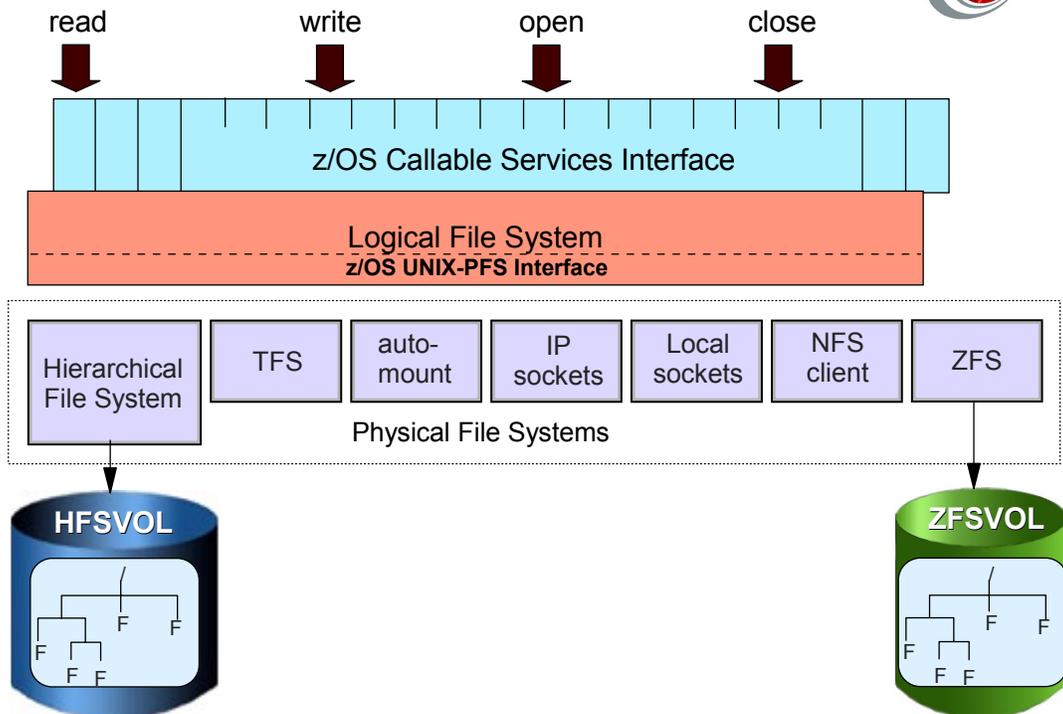


- ❑ **Installing in z/OS V1R2**
 - Enabling APAR/PTF - OW50850 / UW82925
 - Coding APAR/PTF - OW51563 / UW83377
- ❑ **The support for OS/390 V2R10 and z/OS V1R1 is provided by APAR OW51780**
- ❑ **z/OS Distributed File Service zSeries Administration, SC24-5989**
- ❑ **z/OS Distributed File Service zSeries File System Implementation, SG24-6580 - (z/OS V1R3)**

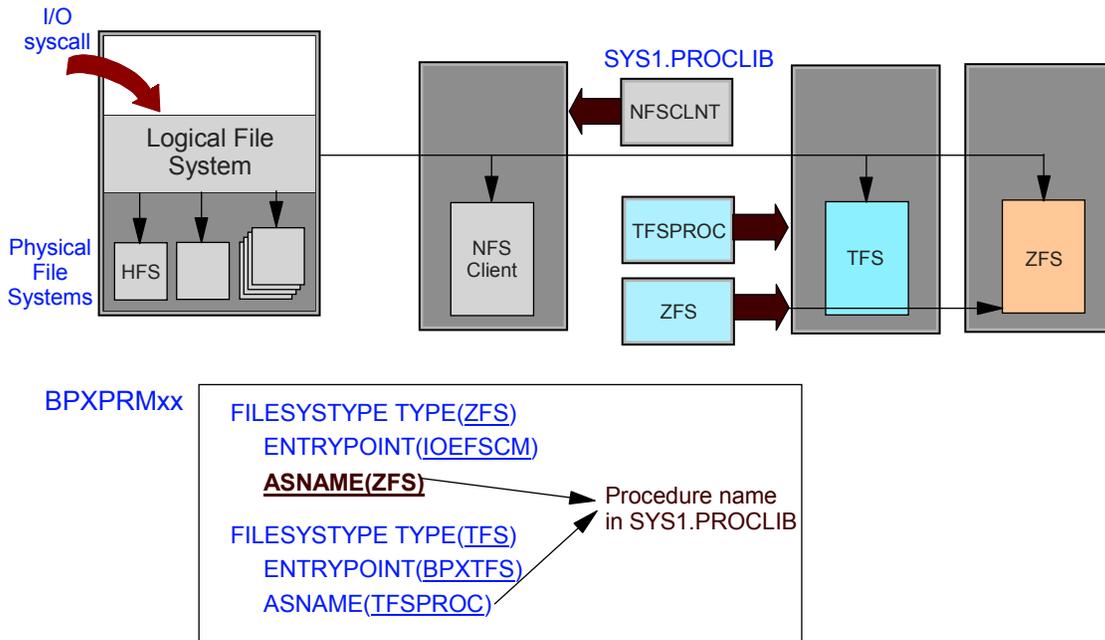
UNIX System Services



Physical File Systems



zFS Colony Address Space



New Functions in z/OS V1R3



- ❑ zFS file systems support UNIX ACLs
- ❑ MOUNT statements for zFS file systems can now be placed in BPXPRMxx
- ❑ Use the UNIXPRIV class profile, SUPERUSER.FILESYS.PFSCTL, to protect the use of zfsadm commands
- ❑ Use dataspace for user data cache

z/OS V1R4 zFS Enhancements



- ❑ Dynamic configuration of IOEFSPRM member
- ❑ Dynamic extension of aggregates and file systems
- ❑ -grow option when formatting an aggregate
- ❑ Duplicate file system names in different aggregates
- ❑ System symbols in IOEFSPRM
- ❑ Metadata backing cache
- ❑ Log file cache
- ❑ zFS mounts

Performance Results

zFS vs HFS Performance



HFS vs zFS Comparison



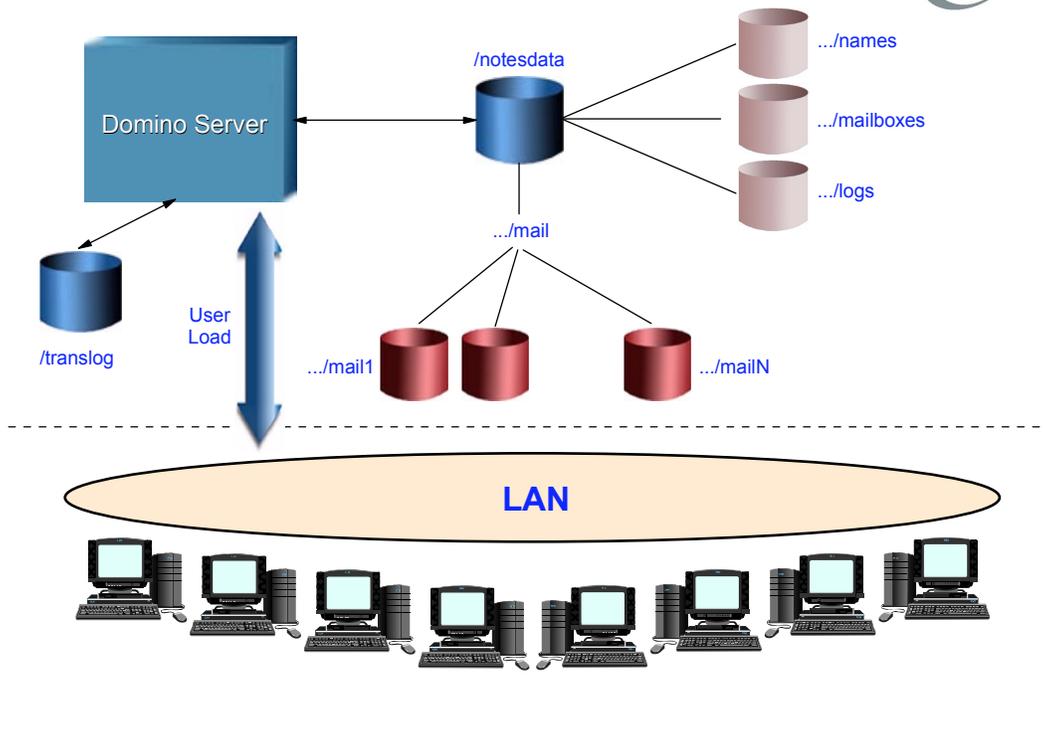
- Choose either the zFS or the HFS file system
- Define the number of processes which run in parallel to read or write 1 MB blocks of data.
- Specify the number of I/Os that each process performs
- Select the percentage of reads among all I/Os. R70 means that 70% of the I/Os are reads and 30% are writes
- Provide a seed value to create predictable random numbers for offsets into the large file when doing an I/O access

HFS vs zFS Comparison Results



	Processes	AVG secs	MIN secs	MAX secs
HFS	10	512.72	507.39	516.31
ZFS1	10	261.64	249.47	268.75
ZFS2	10	230.73	212.76	240.99
HFS	20	992.99	972.35	1002.94
ZFS1	20	478.90	455.40	496.29
ZFS2	20	470.53	438.43	487.82
HFS	40	1964.11	1869.39	1998.09
ZFS1	40	944.53	894.27	979.17
ZFS2	40	961.50	869.07	994.06

Domino File System Structure



Database used in the Study



- ❑ IBM US Directory
 - Compacted to On Disk Structure 41 format
 - 1.24 GB is size
 - 191,000 person documents
 - 1850+ groups
 - 1900+ connection documents

Domino Study Conclusions



- ❑ Domino servers perform better with zFS in all operations
 - Client experiences higher throughput and lower elapsed times for:
 - Notes client access
 - Access via web browser
- ❑ zFS caching appears to be the key for performance
- ❑ zFS drives only about 1/3 of Read I/O to DASD

Growing Aggregates

Understanding zFS Format of Large Aggregates



Allocate zFS Multi-File Aggregates



- ❑ Aggregate can contain one or more zFS file systems

- JCL utility

```
//RFRZAL JOB (999,POK),'R F',CLASS=A,MSGCLASS=U,NOTIFY=&SYSUID,
//      REGION=0M
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DISK DD DISP=OLD,UNIT=3390,VOL=SER=ZFSVOL
//SYSIN DD *
        DEFINE CLUSTER -
            (NAME (OMVS.MUL01.ZFS) VOL(ZFSVL1,ZFSVL2,ZFSVL3) -
             LINEAR CYL(200 0) SHAREOPTIONS(2))
```

z/OS V1R3 Define Aggregate



- ❑ New zfsadm command - zfsadm define

```
$> zfsadm define -a OMVS.CMP01.ZFS -volumes totzf1 -cylinders 10 1
IOEZ00248E VSAM linear dataset OMVS.CMP01.ZFS successfully created.
$> su
#> zfsadm format -a OMVS.CMP01.ZFS -compat -owner 316 -p o755
IOEZ00077I HFS-compatibility aggregate OMVS.CMP01.ZFS has been
successfully created
#> /usr/sbin/mount -f OMVS.CMP01.ZFS -t ZFS /u/zfs/cmp01
```

- ❑ Display quota

```
$> zfsadm lsquota -filesystem OMVS.CMP01.ZFS
Filesys Name      Quota    Used  Percent Used  Aggregate
OMVS.CMP01.ZFS   6335      9      0      2 = 154/6480 (zFS)
```

Format the zFS Aggregate



- ❑ Stand alone utility to format: IOEAGFMT format utility
 - zFS multiple file aggregates and Compatibility mode aggregates
 - Formatting default is the primary allocation and 1 secondary allocation
- ❑ ZFSADM command to format
 - `zfsadm format -aggregate name [-initialempty blocks] [-size blocks] [-logsize blocks] [-overwrite] [-compact] [-owner {uid | name}] [-group {group_id | name}] [-perms decimal | octal | hex_number] [-level] [-help]`
- ❑ Does not need zFS colony address space to be active

Grow an Aggregate Prior to z/OS 1.4



- ❑ Grow the size of an aggregate
 - Size 0 - Indicates to use secondary extent allocation
 - Specify size in KB

```
ROGERS @ SC43: />zfsadm aggrinfo omvs.mul01.zfs
OMVS.MUL02.ZFS (R/W MULT): 50806 K free out of total 51272 (2000 reserved)
```

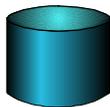
```
#> zfsadm grow -aggregate OMVS.MUL01.ZFS -size 0
IOEZ00173I Aggregate OMVS.MUL02.ZFS successfully grown
OMVS.MUL02.ZFS (R/W MULT): 61598 K free out of total 62072 (2000 reserved)
```

New -grow Option - z/OS V1R4

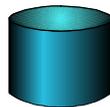


- ❑ When an aggregate is initially formatted using IOEAGFMT or zfsadm format command
 - Size, -size specified must be able to be allocated in the primary allocation and one extension
- ❑ If you wanted to format a three volume aggregate with IOEAGFMT - not possible
 - Requires a primary and at least two extensions
- ❑ Need to use **zfsadm format** command to specify:
 - zfsadm format OMVS.MUL01.ZFS -size 900720 -grow 300240

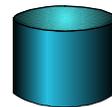
300240 8K blocks



300240 8K blocks



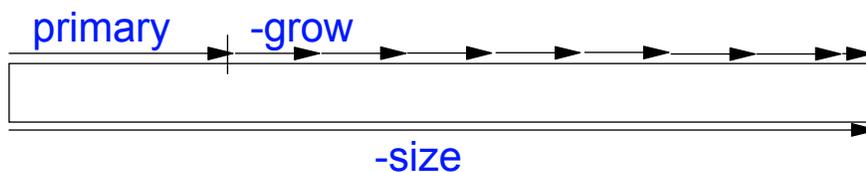
300240 8K blocks



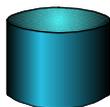
New -grow Option - z/OS V1R4



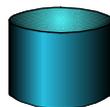
- ❑ Now, IOEAGFMT and **zfsadm format** provide the -grow option
 - Specifies the increment that will be used for extension when -size is larger than the primary allocation
 - Extends by -grow amount until -size is satisfied



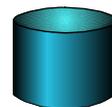
300240 8K blocks



300240 8K blocks



300240 8K blocks



Dynamic Aggregate Extension - V1R4



- ❑ Pre **z/OS V1R4**, aggregates/quotas must be grown via:
 - **zfsadm grow** command
 - File system quota must increase by **zfsadm setquota**
- ❑ With **z/OS V1R4**, aggregates and file system quotas can be dynamically increased
 - VSAM LDS must have a secondary allocation + space

Dynamic Aggregate Extension - V1R4



- ❑ Ways to specify aggregate extension
 - **zfsadm config** command - aggrgrow on | off
 - New option in IOEFSPRM file - **aggrgrow=on | off**
 - mount command - parm('aggrgrow')
 - mount filesystem('omvs.test.zfs') mountpoint('/tmp/test') type(zfs) mode(rdwr) parm('aggrgrow')
- ❑ Multi-file mode aggregate dynamic extension
 - Attach the aggregate using IOEFSPRM file
 - define_aggr R/W attach aggrgrow cluster(OMVS.TEST.ZFS) **aggrgrow | noaggrgrow**
 - **zfsadm attach** command - -aggrgrow or -noaggrgrow
`zfsadm attach -aggregate OMVS.TEST.ZFS -aggrgrow`

Dynamic Aggregate Extension - V1R4



- ❑ When the aggregate fills and an option is specified
 - Aggregate is extended using a secondary allocation
 - Secondary allocation is formatted
 - Becomes available to application

```
IOEZ00312I Dynamic growth of aggregate OMVS.TEST.ZFS in progress, (by
user JANE) .
IOEZ00329I Attempting to extend OMVS.TEST.ZFS by a secondary extent.
IOEZ00324I Formatting to 8K block number 360 for secondary extents of
OMVS.TEST.ZFS
IOEZ00309I Aggregate OMVS.TEST.ZFS successfully dynamically grown (by
user JANE) .
```

Allocate a zFS Aggregate - z/OS V1R4



- ❑ Allocate an aggregate from the ISHELL
 - Option 4 New ZFS ...

```
File Directory Special_file Tools File_systems Options Setup Help
Command ==> UNIX System Serv - 1. Mount table...
Enter a pathname and do one of these: 2. New HFS...
3. Mount(0)...
4. New ZFS...
- Press Enter.
- Select an action bar choice.
- Specify an action code or command on the command line.
Return to this panel to work with a different pathname. More: +
/u/harry
EUID=0
```

Allocate an Aggregatez/OS V1R4



New panel to allocate the zFS aggregate

```
File Directory Special_file Tools File_systems Options Setup Help
-
C
E
E
R
E
Create a zFS File System
Enter the fields as required then press Enter.
File system name . . . . . OMVS.TEST.ZFS
Owning User . . . . . ROGERS (Number or user name)
Owning Group . . . . . SYS1 (Number or group name)
Permissions . . . . . 700 (3 digits, each 0-7)
Primary cylinders . . . . . 10
Secondary cylinders . . . . . 5
Storage class . . . . .
Management class . . . . .
Data class . . . . .
Volume names . . . . .
F1=Help F3=Exit F6=Keyshelp F12=Cancel
```

DFSMSdss Dump and Restore



DFSMS new function APARs for z/OS V1R4

- OW57046 - Catalog flag for zFS aggregate
- OW57015 - DFSMS catalog support
- OW57141 - DSS support
- OW57017 - Listcat support
- OA02713 - HSM support

- APARS allow DFSMS to automatically quiesce the aggregate
- Concurrent Copy can be used

Current method for backup - restore

- 1. Quiesce the aggregate (this drains any activity and suspends any new requests)
- 2. Backup the aggregate (and all the file systems)
- 3. Unquiesce the aggregate (allowing zFS activity to continue)



**z/OS Distributed File Service
zSeries File System
Implementation**

--- SG24-6580 ---

Redbook at z/OS V1R3

Published: September 2002

--- SG24-6581 ---

zFS Chapter at z/OS V1R4