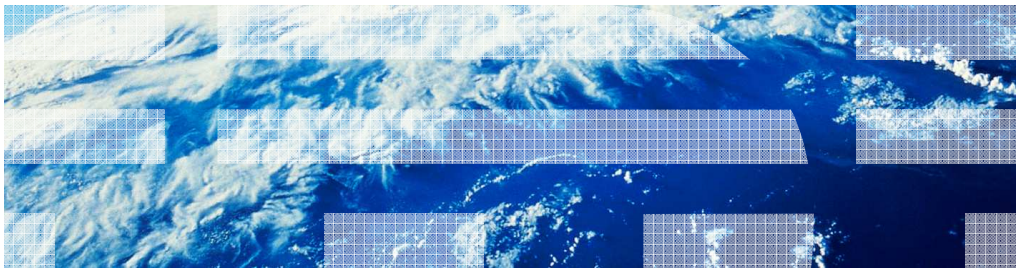
z/OS V1R13

zFS: Direct I/O



Session objectives

- zFS direct I/O
  - A performance improvement for the shared file system environment using zFS read-write sysplex-aware file systems

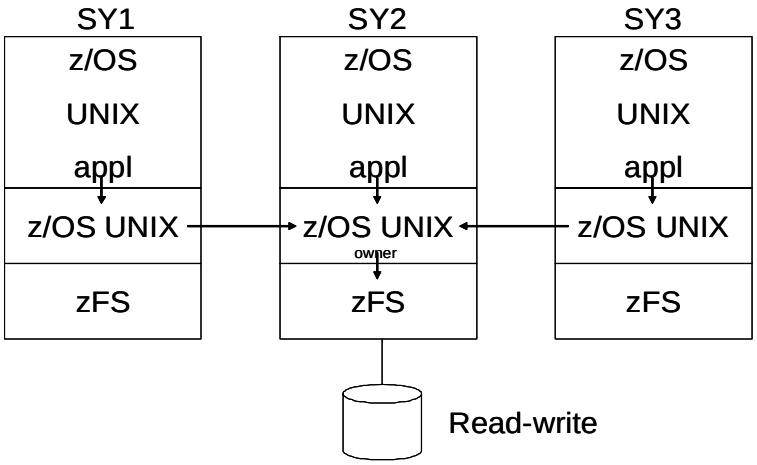
Overview

- zFS direct I/O
- Problem statement / need addressed
  - In a sysplex shared file system environment, it should not matter where an application runs, nor which system owns a file system.
    In current systems, location is important due to function shipping performance and file system ownership needs to be managed.
- Solution
  - Provide the ability to directly read and write zFS data from any system in a sysplex shared file system environment.
- Benefit / value
  - Much less concern about file system owner and where applications run in a sysplex shared file system environment.

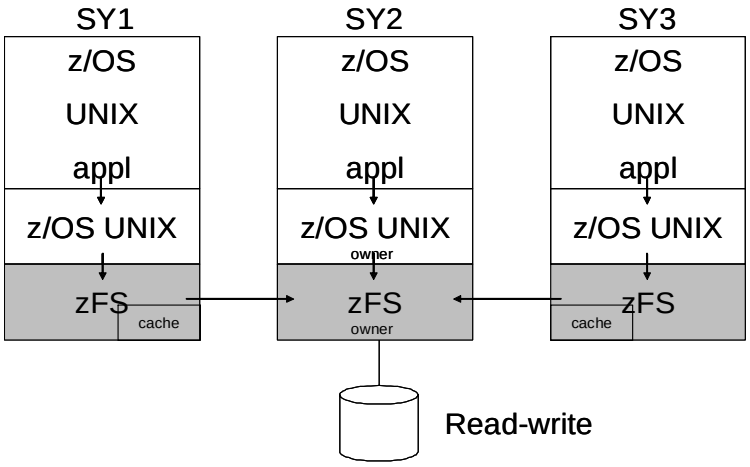
Usage and invocation

- zFS sysplex support (short history of recent enhancements)
  - zFS R11 introduced support for zFS read-write sysplex-aware file systems (sysplex=on). This provided a performance improvement especially in the area of read performance. It also made file system ownership less important.
  - ZFS R11 APAR OA25026 improved the granularity of using zFS read-write sysplex-aware file systems (sysplex=filesys). This allowed you to individually choose which zFS read-write file systems are sysplex-aware and which are not. **This is the preferred method of running in a shared file system environment (sysplex=filesys).**
  - zFS R13 direct I/O enhances support for zFS read-write sysplex-aware file systems. When all systems are at R13, zFS can directly read and write sysplex-aware file systems from all systems in the shared file system environment. This makes file system ownership much less important. **zFS R13 always runs sysplex=filesys in a shared file system environment.**

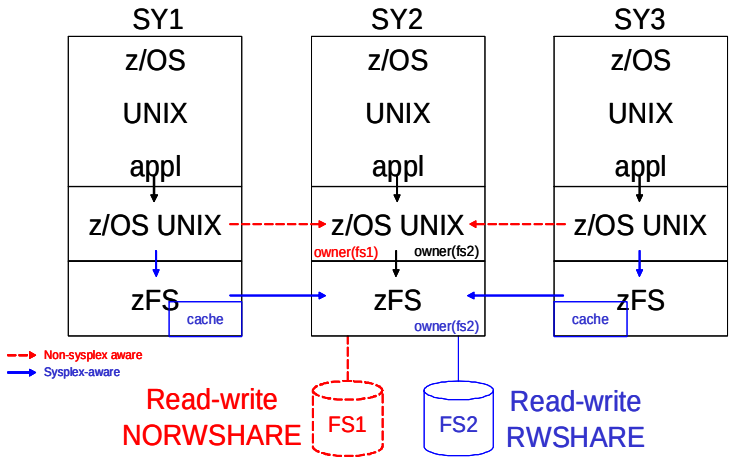
Usage and invocation: Non-sysplex aware read-write file system



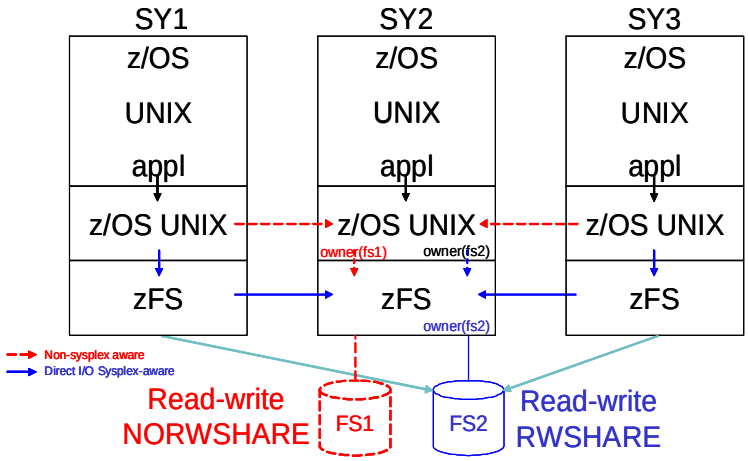
Usage and invocation: R11 sysplex-aware read-write file system



Usage and invocation: R11 OA29619 sysplex-aware read-write file system



Usage and invocation: R13 direct I/O sysplex-aware read-write file system



Interactions and dependencies

- Software dependencies
  - None
- Hardware dependencies
  - None
- Exploiters
  - Any users of zFS read-write sysplex-aware file systems in R13

Migration and coexistence considerations

- Migration to zFS R13 is a two step process
  - Install toleration APAR OA32925 (PTF UA55765) on all zFS R11 and R12 systems and make it active with a rolling IPL.
  - Change your zFS IOEFSPRM file to sysplex=filesys on all systems and make it active with a rolling IPL.
- At this point you can bring in z/OS® V1R13 (and zFS R13).
- Recognize that zFS R13 can use more DASD space than prior releases of zFS (zFS R13 does not use 1K fragments anymore).
- Also, default for aggrgrow changes from aggrgrow=off to aggrgrow=on.

Installation

- Changes in zFS installation
  - zFS load modules have moved from SIOELMOD (a PDS) to SIEALNKE (a PDSE).
  - The zfsadm command was a binary in the z/OS UNIX® file system. It is now a shell script that executes IOEZADM. This is an entry with the sticky bit – so the IOEZADM load module is executed.
- Changes in IOEFSPRM configuration options
  - dir_cache_size is ignored (it is not needed any longer)
  - nbs is ignored (nbs is always on with no performance penalty)
  - client_cache_size default changed from 128M to 32M (only needed for pre-R13 zFS owners of sysplex-aware file systems)
  - meta_cache_size defaults changed from 32M to 64M (storage that was used for dir_cache_size can be given to metadata cache)

Session summary

- zFS direct I/O
  - A performance improvement for the shared file system environment using zFS read-write sysplex-aware file systems

Appendix - References

- Publications
  - z/OS Distributed File Service zSeries® File System Administration (SC24-5989)
  - z/OS Distributed File Service Messages and Codes (SC24-5917)

Trademarks, disclaimer, and copyright information