zFS file systems in sysplex sharing

#12 zNibbler

USS file systems cannot be mounted R/W on more than one system at the same time. Even the USS sysplex R/W sharing does not violate this rule and is implemented as a function within the logical file system interface. This is implemented within the same resource serialization complex (or sysplex) and is not a problem if set up correctly.

However, for systems that are completely independent and access a file system at the same time accidentally, this can cause problems and be harmful for the file system. Therefore, a physical file system may implement a function called “write protection” that prevents this situation. This is done by writing an identifier into the file system when accessing it that contains among other things the sysplex and system name and a time stamp.

When an HFS file system is mounted on system A, an identifier is written into the HFS data set that is checked every time data is physically put into the file system. If then system B also accesses the file system, it is not obvious that another system has mounted the file system R/W already and it overwrites the identifier of System A.

The next time System A wants to update the HFS data set, the problem is recognized and system A stops writing to the file system to avoid destroying the internal structure. This solution is not perfect because system A is the first one accessing the data set and still loses its R/W access.

Finally with z/OS V1R6, zFS has implemented a different technique to guarantee a correct state for zFS aggregates[1] that are accessed R/W in a system:

* The identifier is cleared when a compatibility mode aggregate is successfully unmounted or a multi-file system aggregate is detached.
* A daemon process updates the time stamp twice a minute.

These two actions are sufficient to ensure that a second system trying to get access in read/write mode can recognize whether there is another system writing to the aggregate already. This implementation with zFS assures that the first system always keeps its read/write access as desired.

When customers have a specific system, located in another sysplex, that is used for preparing and servicing UNIX System Services file systems that are type zFS, the first mount may take at least 65 seconds. This is caused by the write protection implementation in the following situation:

* The aggregate used in the service system is still mounted read-write after adding the service.
* You use ADRDSSU[2] to copy the aggregate into a new aggregate to be used in the production environment afterwards, or dump it first and restore it later for use.

As long as the aggregate is attached read-write, the identifier is active in block zero (0). As a result, on ADRDSSU COPY or DUMP processing, it is copied as well.

When customers mount or attach the aggregate in the target production system, zFS sees this identifier. Because it does not reflect the sysplex environment, zFS carefully waits 65 seconds for an update of the time stamp value within the identifier. If the aggregate is used read-write outside of the sysplex, then this time stamp should get updated every 30 seconds. If no update occurs, this shows that the identifier is there without a reason. In this case, after the 65 seconds, zFS attaches and mounts the aggregate.

Avoiding the unnecessary wait on the first access to the aggregate

To avoid the 65 seconds wait time on access to the aggregate in the production environment, customers can perform *one* of the following actions:

* Unmount or detach the source aggregate in the service system before using **ADRDSSU COPY** or **ADRDSSU DUMP**.
* Switch the source aggregate to read-only access, using the ISHELL modify interface from the mount table panel or the **/usr/sbin/chmount** shell command. Then copy the contentsof the aggregate. This switch removes the write protection identifier, as well.

[1] zFS aggregate is the data set that contains zFS file systems. A zFS aggregate can contain one or more zFS file systems. A zFS aggregate is a Virtual Storage Access Method Linear Data Set (VSAM LDS).

[2] ADRDSSU is a dfSMSdss utility used for various disk and dataset operations.