

# **Determining File System Activity By System**

Bill Schoen  
wjs@us.ibm.com  
October 29, 2009

Last update: 3/17/2010

When using the z/OS UNIX shared file system support for read-write access, it is important to locate the applications using a file system on the system owning that file system. The shared file system support enables access to the data from all systems but uses a function shipping model to access the data from non-owning systems for file systems mounted read-write. XCF services are used as the network between the systems. Most observe the performance of this remote access to be far inferior to local access. This performance difference can become problematic as volume of access increases or throughput requirements increase.

One of the difficulties in managing this is determining which file systems are accessed from each system and by what applications. With that understanding, the next problem is to manage the location where the applications run and where the file system is owned. This is further complicated when systems are brought up and down and file system ownership changes. It is possible to manage some file system placement through the use of the SYSNAME parameter on the mount statements, prioritizing which systems should take ownership of which file systems.

In z/OS V1R11 it is possible to reduce these system management challenges. The zFS file system can be enabled to help with this by using the zFS `sysplex=on` parameter setting. In first half 2010 you will also be able to enable this function on an individual file system basis. Although this does not eliminate the need to balance your workload, zFS will monitor usage and dynamically move its local access to the system with the most access. On remote systems, zFS will perform aggressive caching which should improve performance over the prior functionality of zFS and HFS.

The SMF type 92, subtype 5 records contain file system read and write counts. If type 92 subtype 5 is active when a file system is mounted, this data will be accumulated and written to the record when the file system is unmounted. Accessing this information on each system should let you know to what degree the file system is used on each system.

An alternative to collecting and analyzing this SMF data is use of the `wjfsmon` tool. This tool uses some of the same counters used to collect this SMF data. If SMF is not active for type 92 subtype 5, the tool will enable tabulation of some of the same data while it is running but does not enable recording of the SMF data. You may be able to observe these counts in fields using the `w_getmntent` service or through the file system attribute display in ISHELL. The remainder of this document describes `wjfsmon` use and data interpretation.

## The wjfsmon utility

wjfsmon is a monitor facility that collects some file system usage data in intervals, aggregates the interval data, and is the view dialog for this data. The primary function of the viewer is to show which file systems have the greatest amount of remote access and from which systems. This tool should be able to run on z/OS 1.10 and above. It is intended to help you decide how zFS sysplex enablement could help and could provide additional insight into how you might place your file systems and UNIX workloads with or without zFS sysplex enablement.

### Start the monitor

In order to provide a view of file system access from all systems, it is necessary to start wjfsmon on all systems in the sysplex. This can be done in a variety of ways including from TSO, a z/OS UNIX shell, batch jobs, or from the operator console using SYSREXX. The tool must be installed in an appropriate place so that it may be run from the environment from which you want to use it. For example, to use in the shell place it in a directory from where you can run programs and have at least read and execute authority. From TSO, it is easiest if it is in a PDS in your SYSPROC or SYSEXEC concatenation. To use from SYSREXX it must be in a library defined by REXXLIB in your AXRxx parmlib member.

To start the monitor from TSO or the shell, run the command **wjfsmon -s**. This must be done on each system from a user that is either a superuser or is permitted to BPX.SUPERUSER.

From SYSREXX, use the operator command **F AXR,WJSFSMON,T=0 -SA**. This only needs to be done on one system and wjfsmon will route the command to each system, one at a time, to start the monitor. It determines the scope of your OMVS sharing group and if that is less than the entire sysplex it will only start the monitor on the systems in the sharing group. Rather than -SA, -S can be used to start the monitor only on that one system. In order to use SYSREXX for this command, you must be logged onto the console with a userid that is a superuser or permitted to BPX.SUPERUSER. Be sure to include the T=0 on the command line so that sysrexx does not time-out and cancel the tool. Note: prior to z/OS V1R11, APAR OA26802 is needed for SYSREXX to run REXX programs that use UNIX services.

When wjfsmon is started it runs as a background UNIX process. By default it will wake up once a minute and collect data and keeps a history of 60 intervals. Command line options are available to alter the default sample interval and history size. The data is kept in /var/fsmon/. This could be several megabytes of data. Ensure your /var file system has sufficient space on all systems. If it is too difficult to obtain space for /var you could create the directory /var/fsmon and mount a file system there or create /var/fsmon as a symlink to some other directory where you can get space.

## Aggregate the sampling data

The sampling data accumulated by the wjfsmon monitor should be aggregated and placed into a data set. As with starting the monitor, aggregating the data is done using wjfsmon from TSO, a UNIX shell, or sysrexx. You can let wjfsmon allocate the data set or allocate it yourself to control the allocation. Use dsorg=ps, recfm=vb, lrecl=252. If wjfsmon creates the data set it will allocate 10 cyl times the number of systems and 10 cyl as the secondary allocation using unit(sysallda). A similar amount of space will also be needed in /tmp. It will use directory /tmp/fsmon. As above, if /tmp space is a problem, you can create this directory and mount a file system there or make a symlink to another directory. From TSO or a shell, run

**wjfsmon -w *full.data.set.name***

To run from sysrexx issue the command

**F AXR,WJSFSMON,T=0 -W *FULL.DATA.SET.NAME***

Superuser authority is not needed to aggregate the data. If you wish to limit access to the sample files you must set appropriate access rights on the /var/fsmon directory. The aggregation process can take a minute or so to run. The time can be highly variable based on the history size, number of file systems, and number of systems in the sysplex.

## Stop the monitor

When you no longer want to monitor file system activity, wjfsmon can be stopped from TSO, the shell, or sysrexx. You must be a superuser or be permitted to BPX.SUPERUSER to perform this function. From TSO or a shell enter the command

**wjfsmon -e**

From sysrexx enter the operator command

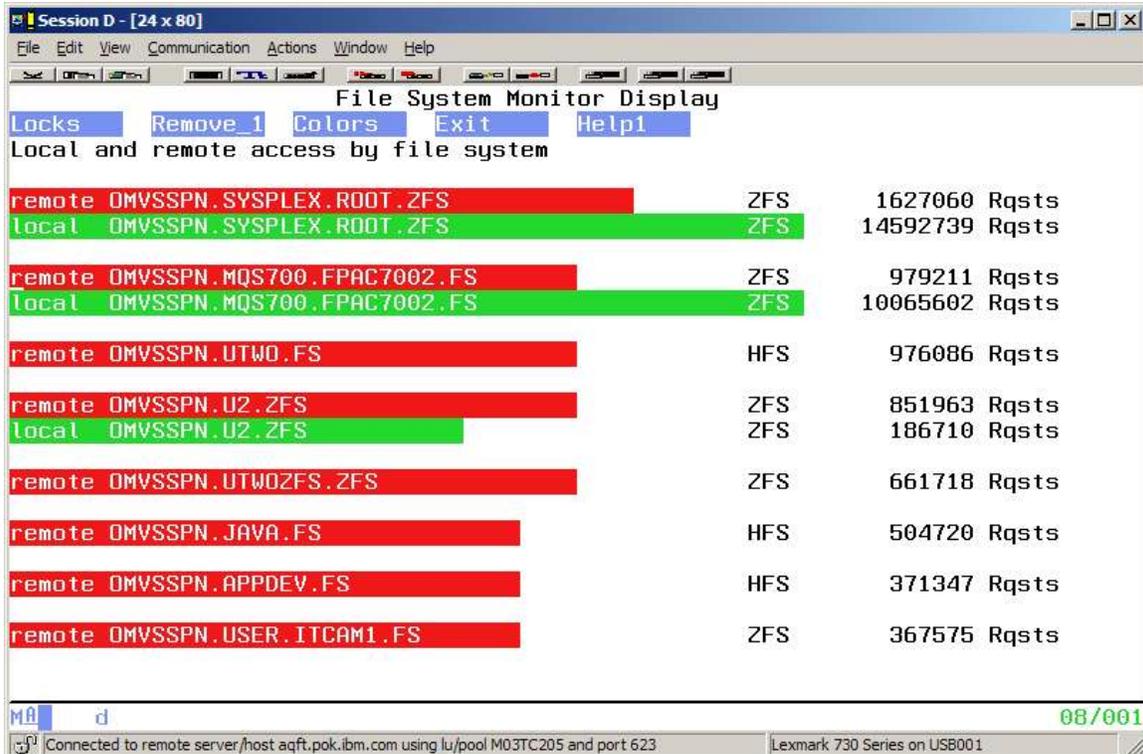
**F AXR,WJSFSMON,T=0 -E**

You only need to run this on one system. This command ends the monitor and cleans up its residual files in /var/fsmon/ on all systems.

## View and interpret the data

wjfsmon will only present the data in an ISPF session. A seven color terminal with extended highlighting is recommended. Run the command **wjfsmon -r *full.data.set.name***

The first data screen you see shows file systems ordered by volume of remote access, showing both remote and local access. This is an example of what might be shown.



The screenshot shows a terminal window titled "Session D - [24 x 80]" with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main display is titled "File System Monitor Display" and contains a menu (Locks, Remove\_1, Colors, Exit, Help1) and the text "Local and remote access by file system". The data is presented as a list of file systems with horizontal bars representing access volume. The bars are color-coded: red for remote access and green for local access. The data is as follows:

Access Type	File System	File System Type	Remote Access (Rqsts)	Local Access (Rqsts)
remote	OMVSSPN.SYSPLEX.ROOT.ZFS	ZFS	1627060	
local	OMVSSPN.SYSPLEX.ROOT.ZFS	ZFS		14592739
remote	OMVSSPN.MQS700.FPAC7002.FS	ZFS	979211	
local	OMVSSPN.MQS700.FPAC7002.FS	ZFS		10065602
remote	OMVSSPN.UTW0.FS	HFS	976086	
remote	OMVSSPN.U2.ZFS	ZFS	851963	
local	OMVSSPN.U2.ZFS	ZFS		186710
remote	OMVSSPN.UTW0ZFS.ZFS	ZFS	661718	
remote	OMVSSPN.JAVA.FS	HFS	504720	
remote	OMVSSPN.APPDEV.FS	HFS	371347	
remote	OMVSSPN.USER.ITCAM1.FS	ZFS	367575	

The bottom of the window shows a status bar with "MA d", "08/001", and connection information: "Connected to remote server/host aqft.pok.ibm.com using lu/pool M03TC205 and port 623" and "Lexmark 730 Series on USB001".

The information on this screen shows remote and local access for file systems. The label **Rqsts** is a count of the number of accesses to the file system. With APAR OA29712 on release 11, this label will show **I/O** which is the total read and write count for the file system. The number of requests only loosely relates to the amount of cross system messaging but is still a good indication of the need for remote access.

Be careful when interpreting local access on the displays. Remote access drives requests to the owning system which show as local access on subsequent displays. On this first display which shows local and remote by file system, remote requests are deducted from local requests but due to differences in sampling times on each system it is possible local counts can be off by a couple sampling intervals. Identifying remote access rather than local access is key in determining how much value zFS sysplex support might benefit.

The first file system in this list is the sysplex root. When the sysplex root is mounted in read-write mode, volume of cross system access may be quite high, but the data can be misleading prior to OA29712 due to the name caching in the logical file system. This file system is likely to perform best when mounted read-only.

The other file systems on this display are interesting. The second file system in the list has a significant volume of remote access. The raw numbers indicate remote access is about 10% of local access. This indicates the file system is probably mounted on the best system. Move the cursor to that file system and press Enter and you see a breakdown of access by system.



From this display you can see almost all of the access is on only two systems. Prior to the sharing by file system support, the fast and slow counts indicate general activity. The count of slow represents the amount of contention on the file system. Generally you should expect very little contention. With the sysplex sharing support, the columns will be counts of reads and writes.

The first character on each line indicates what type of access is represented by that line. **L** represents local access, **R** represents remote access, and if you have zFS sysplex sharing enabled, **Z** represents remote access through a shared zFS.

From these two displays you can narrow additional investigation. Enabling zFS sysplex=on or enabling sharing for this file system are easy choices that may help

performance. Another choice is to look at what applications are running on system JB0 and evaluate whether they can be moved to JC0. This would keep almost all access local.

You can repeat this assessment for each file system on the first display. Best performance is likely when there is little remote access to any file system. That is not always practical.

If you must have a significant number of file systems with remote access or you observe you have a configuration with a significant amount of sharing and analysis and the management effort is too labor intensive, enabling zFS sysplex=on will probably help with overall performance including a likely reduction in XCF traffic.

If you only have a few file systems that are actively shared as read-write, enabling the zFS sharing by file system, when available, for those specific file systems is likely to be the best choice.

The colored bars drawn in the examples above are not proportional. All sizing bars displayed by wjfsmon are based on a log 2 scale.

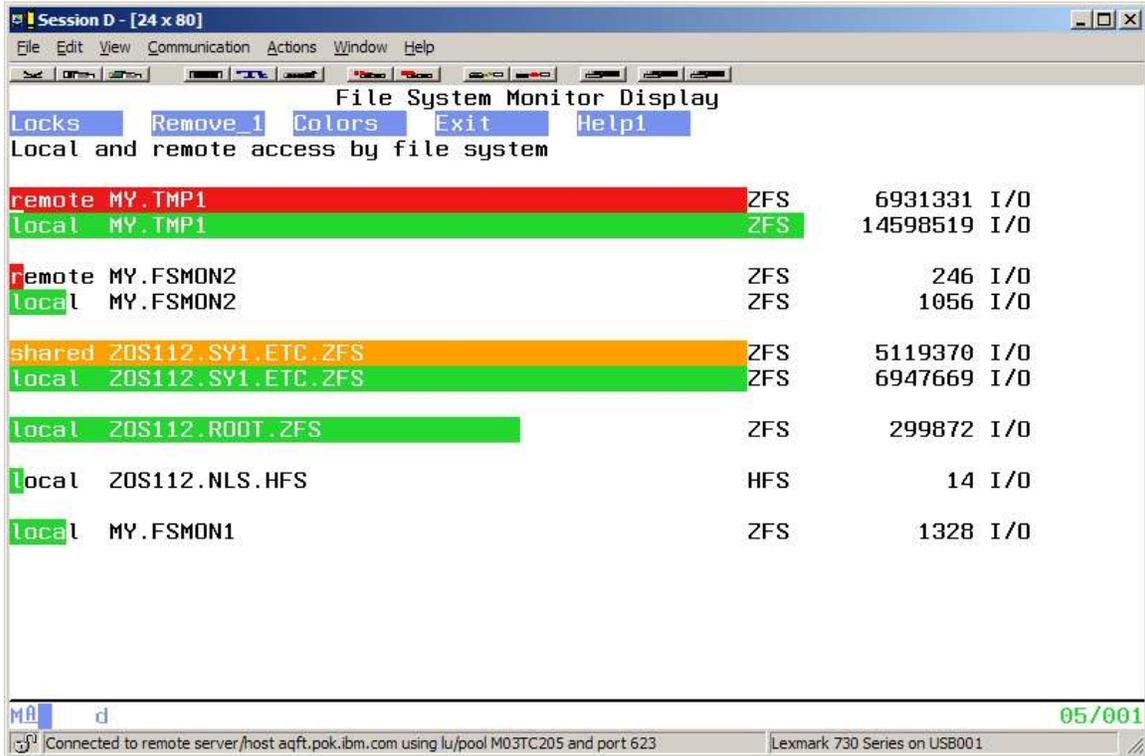
The wjfsmon viewer has a number of other displays that may provide some insights into how your file systems are used. However, some of the functionality will only function as of release 11 with APAR OA29712.

On the second display which breaks down the I/O by system, you can move the cursor to one of the bars and press Enter. This will breakdown those requests by time interval. You may also be able to view what jobs and users that were accessing that file system if a connection to the file system existed when wjfsmon was taking a sample. This information might be used to help evaluate what users or applications might perform better if moved to another system.

The Locks button on the upper left of the first data display can be selected to show file system contention. Amount of contention is indicated by the number of slow latch obtains. Fast obtains indicate no contention. A very low percentage contention should normally be expected, however, contention is likely during file system management operations, such as moving the owner of a file system. If you see contention that cannot be explained by file system management operations, enabling zFS sysplex support for those file system could help.

The rightmost button on the top of each display is a help button. Select that button for information on what the display can be used for and navigation on that display.

Below is an example of the first display with sysplex sharing support enabled:



Note that the sysplex root is not in the display. A properly configured sysplex root will only have activity due to name resolution and no file reads or writes. You will be able to observe activity on this file systems such as this by selecting the Locks button as described above.

The third file system shows read and write requests from non-owning systems and local reads and writes. The scale of access from the owner and other systems is comparable.

Selecting that third file system, you can see the breakdown of reads and writes by system. With a similar balance of activity from two systems, enabling the zFS sysplex support for this file system was a good choice.

The screenshot shows a terminal window titled "Session D - [24 x 80]" with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main display area is titled "File System Monitor Display" and contains a table of file system activity. At the top, there are four buttons: "Back", "Remove", "Exit", and "Help2". Below these buttons, the text "ZOS112.SY1.ETC.ZFS" is displayed. The table has three columns: the first column lists file systems, the second column is labeled "Reads", and the third column is labeled "Writes". The first row, "L SY2", is highlighted in green and shows 39602 reads and 6908067 writes. The second row, "Z SY1>SY2", is highlighted in orange and shows 30333 reads and 5089037 writes. At the bottom right of the window, the text "02/001" is visible. The status bar at the very bottom shows "Connected to remote server/host aqft.pok.ibm.com using lu/pool M03TC205 and port 623" and "Lexmark 730 Series on USB001".

	Reads	Writes
L SY2	39602	6908067
Z SY1>SY2	30333	5089037

## Use of sample interval and history size

When the monitor is started, a sampling interval and history size can be specified. Use **-i** to specify the interval in seconds and **-h** to specify history in number of intervals. These numbers are approximate. The monitor should be started with the same parameters on each system. The default is **-i 60 -h 60**

The default will give you a fairly detailed view for up to the last hour prior to capturing an aggregation of the data.

If 60 samples per system is producing very large sets of data, consider changing **-h** to a lower value. The interval can be increased to give you the same period or a larger period for analysis. For example, **-i 3600 -h 24** will sample once an hour and give you a view for the past 24 hours with a smaller set of data. If your workload varies during the day, this will give a broader picture.

User and process data is only collected when a sample is taken. If you need finer grained information about the users or jobs using file systems you can increase the sampling frequency. For example, **-i 6 -h 30** would sample on 6 second intervals and keep a three minute history.

The wjfsmon data viewer has the capability of aggregating and viewing the live system in addition to viewing from a data set. Keeping the history small will minimize the time it takes to collect the data. This can make interactive use on a live system practical. If the interval is small and you already have the monitor running, you might be able to use wjfsmon to view file system activity when trying to trouble shoot sudden performance issues that may relate to file system access.

## Command format

Start the monitor:

```
wjfsmon -s[a] [-i interval] [-h history]
```

Query monitor status:

```
wjfsmon -q
```

Aggregate data and save to a data set:

```
wjfsmon -w full.data.set.name
```

View monitor data:

```
wjfsmon -r [full.data.set.name | active]
```

Stop the monitor:

```
wjfsmon -e
```

## Option flags

-s	start the monitor
-sa	start the monitor on all systems (sysrexx only)
-i	set the monitor sampling interval in seconds (default=60)
-h	set the number of intervals the monitor keeps (default=60)
-q	query monitor status on all systems
-w	write aggregated intervals to the specified cataloged data set
-r	read and display the aggregated data from a data set or the system
-e	stop the monitor on all systems