SFS Performance Management Part I: Introduction

Version 4 – see https://www.vm.ibm.com/library/presentations/ for latest version.

Bill Bitner
z/VM Development Lab Client Focus & Care
bitnerb@us.ibm.com

Details are self-explanatory on the slide.

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

Db2*	FlashCopy*	IBM (logo)*	OMEGAMON*	z13*	z/Architecture*	zSeries*
DirMaint	FlashSystem	IBM Z*	PR/SM	z13s	zEnterprise*	z/VM*
DS8000*	GDPS*	LinuxONE*	RACF*	z14	z/OS*	z Systems*
ECKD	ibm.com	LinuxONE Emperor	System z10*	z10 BC	zSecure	
FICON*	IBM eServer	LinuxONE Rockhopper	XIV*	z10EC		

* Registered trademarks of IBM Corporation

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.
ITIL is a Registered Trade Mark of AXELOS Limited.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries 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, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
UNIX is a registered trademark of The Open Group in the United States and other countries.
VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.
Other product and service names might be trademarks of IBM or other 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.
This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Details are self-explanatory on the slide.

Abstract

This session provides an understanding of Shared File System (SFS) performance management. The presentation will cover performance tasks, such as preventing performance problems, monitoring performance, and solving performance problems. Tuning tips and a case study will be included. Attendees should have some familiarity with SFS, but they need not be experts.

Acknowledgements

- My thanks to various folks for helping pull this material together.
  - Charlie Bradley
  - Melissa Carlson
  - Wes Ernsberger
  - Sue Farrell
  - Bruce Hayden
  - Butch Terry

The speaker notes were never written with the intent of including them in handouts. So, if you are reading this, please keep in mind that I never took the time to do a quality job with the speaker notes.

Please excuse grammar and typos. However, any suggestions or corrections are appreciated.

Overview

- SFS Structure

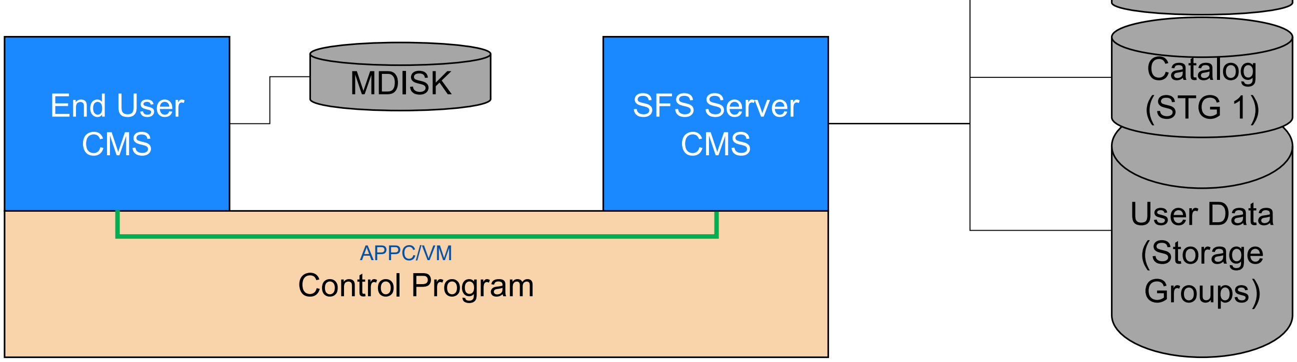
- SFS Performance Management
  - Preventing performance problems
  - Monitoring performance
  - Solving performance problems

- Case Study

This presentation will cover the tasks related to the performance of SFS file pool servers and is meant to take the mystery out of this. After an overview of the Shared File System structure, we'll look at three areas of performance management, and then wrap things up with a case study.

SFS Concepts

- Coexists with the minidisk file system
- Components in
  - CMS End User
  - SFS Server Virtual Machine



The presentation is really meant for those that know and understand at least the basics of SFS. A few charts here will review the basics and structure of SFS. SFS coexists with the current minidisk (EDF- Enhanced Data Format) file system. For our purposes SFS is made up of two parts: processing in the in the end user virtual machine (CMS nuc + CSL) and processing in the in the server virtual machine. It is important to note that communication is performed via APPC/VM with private protocol. The figure represents SFS (without data space exploitation).

When a user writes to a file, CMS in the user virtual machine sends the data to be written to the server virtual machine. That server, file pool server, writes the data in the file pool. For a user to have space in a file pool, it would first have to be enrolled in the file pool.

The file pool server has a pair of log disks for redundancy. It will have a control minidisk that contains meta data of the file pool, and then a set of storage groups. Storage group 1 is also known as the catalog. The other storage groups can be used to host user data. It is common for storage groups to be made up of multiple minidisks even though we only show one here.

SFS Structure – Server Data

- Control Data
  - **POOLDEF** file on Server A-disk
  - File pool control minidisk
  - Catalog Storage Group – also known as Storage Group 1

- Log Data
  - Log Minidisk 1
  - Log Minidisk 2

- User Data
  - Storage Group 2 minidisks
  - ...
  - Storage Group n minidisks

To level set on terminology, we split up the SFS Server structure into 3 parts:

•Control data is the management part of SFS. The Pooldef file describes the config/allocation of minidisks for various uses. The control minidisk is used to map out other disks used for real work. Storage Group 1 holds the catalog information. I'll try to refer to this as catalog so as not to confuse with other storage groups.

•Two log disks are provided to mirror each other for RAS reasons. Related to info about LUWs.

•User Data is the actual file data blocks (stuff inside file). The numbers start at 2 and go to "n".

SFS Performance Management

- Preventing performance problems
- Monitoring performance problems
- Solving performance problems

This section of the presentation is broken down into three pieces. I often use the lawn mower analogy. It is best to read the instructions when putting it together. Periodically check the fluids and replace spark plugs as necessary. When it is performing poorly, check various items and adjust as necessary.

Preventative Tuning

- CP tuning considerations
- CMS tuning considerations
- Disk placement
- VM Data Spaces
- Recovery
- Multiple file pools

The first task, preventing problems, we’ll refer to as preventative tuning. It involves a list of performance guidelines when you are defining a new file pool or modifying an existing one. These are the areas that we’ll discuss. If these guidelines are followed, you usually don't have any SFS performance problems.

CP Tuning Considerations

- **OPTION QUICKDSP**
  - Not really needed on current z/VM systems, but left as the default on many servers
- A higher Relative Share Setting
  - Default shown in books and shipped is Relative 1500
  - Adjust as necessary for loads and nature of workload
- Minidisk caching
  - Make logs ineligible (directory **MINIOPT NOMDC**)
  - Control minidisk not eligible
  - Other server minidisks may benefit greatly
  - Directory **OPTION NOMDCFS** statement to avoid limit on MDC insertions
- **CP SET RESERVED** as needed

The CP Tuning considerations are both the easiest and the most challenging of the different areas. A challenge because it depends on what else is going on in the z/VM system.

The first one is a bit of an artifact. QUICKDSP, which stands for quick dispatch, is left over from when z/VM had an “eligible list”, that is it had a sort of time out list it placed virtual machines to avoid thrashing on various resources. The QUICKDSP ON setting allowed excluded virtual machines from having to wait in the eligible list. So in many ways it is no longer needed. However, it also had other subtle effects to scheduling and therefore when the eligible list was dropped from z/VM virtual machines that previously had QUICKDSP retrained it. The CP Command and Utility Reference would have additional information.

The share setting is the biggest tuning knob in terms of controlling access to processor resources. The Relative value means relative to other virtual machines on the system. The default setting for a virtual machine is 100. The server supports multiple users such as 15 so we recommend 1500. This should be set inline with other server settings such as your external security manager and other work on the system.

Minidisk caching, or MDC, is a write-through cache managed by the control program to cache the data of minidisks.

- The logs will not benefit from MDC because I/O activity is write-mostly.
- Having a blocksize of 512 bytes, the control minidisk is not eligible. Even if it was eligible it would not benefit due to high write activity.
- The rest of the server minidisks are eligible and can be quite beneficial.
- The NOMDCFS option is for No MDC Fair Share limiting. Overrules CP's MDC processing that restricts updates for any given virtual machine. After all, SFS server is doing I/O on behalf of others.

SET RESERVED establishes the number of pages the virtual machine is entitled to always have resident in real memory. Use when server is serial page faulting. Remember that when the SFS file pool server waits, so do all the virtual machines with outstanding requests to the server.

CMS Tuning Considerations

- Choose SFS startup **USERS** parameter value carefully
  - Best estimate of number of users at peak activity
  - Server optimizes its processing based on value
  - Better to over-estimate than under-estimate
- CRR (Coordinated Resource Recovery) Server
  - Should have one or performance degrades significantly
- CMS SFS file cache
  - Controls read ahead and write behind buffers
  - Defaults to 20KB for SFS files
  - If high paging rate, consider lowering
  - If low paging rate, performance benefit to increase
  - Controlled by **BUFFSIZE** parm in **DEFNUC** macro
  - Maximum is 96KB
- Saved Segments
  - **CMSVMLIB** on end user side (includes parts of SFS code)
  - **CMSFILES** on SFS server side (includes SFS and CRR server code)

The USERS starup value tells the server how much work it should configure itself to handle. If specified too large, may experience serial page faults problem in server and/or increased checkpoint duration (long blip). If specified too small, the server will not configure enough agents (tasking objects) to handle the incoming requests. This can cause an undesirable queueing effect. It is better to overestimate a little.

Coordinated Resource Recovery, or CRR, is a server machine that allows synchronization of changes across multiple resources, particularly file pools through use of a two phase commits, such that a rollback of work would roll back all the changes in a particular unit of work. This was originally implemented for use with the CICS for VM product. However, in normal SFS usage it would be rare to have a scenario where CRR is needed. But SFS never knows when a second resource will be introduced. So if you don’t have a CRR server running, SFS enters something called “limp mode” where it takes extra precautions (and extra overhead) to ensure it doesn’t get into a scenario where it could not rollback work. Bottom line is, just have a CRR server. Use QUERY FILEPOOL STATUS recovery: to determine if users connected.

This next tuning aspects is not as important in 2021 as it was 30 years ago, because memory is less expensive and I/O is faster. CMS has a read ahead and write behind buffer system to minimize I/O. This is true for both CMS minidisk and

SFS file systems. In the SFS case however, it has more to do with how much data at a time is transferred between the end user CMS virtual machine and the SFS file pool server.

The cache is specified for all users in their nonshared virtual storage. Some measurements indicate a value larger than 12k would benefit most environments. This cache is for the SFS file I/O and should not be confused with minidisk cache for read ahead, write behind. To change CMS file cache size, update BUFFSIZE parm in DEFNUC macro; assemble DMSNGP ASSEMBLE; rebuild CMS nucleus. Refer to Service Guide and CP Planning and Administration manuals for more details. Allowable range is 1 to 96KB.

Disk Placement

- Log disk placement considerations
  - Place on separate disks and channels[1]
  - Consider other activity on the disks and impact
- Catalog storage group (SG1)[2]
  - Spread across volumes to distribute I/O
- Guidelines for user data storage groups[2]
  - Spread across volumes to distribute I/O
  - Consider if non-SFS space activity is low or uniform
  - Same amount of space on each disk volume
  - Volumes should have similar performance characteristics
- For placement tips related to availability see the CMS Planning and Administration Guide, some involve trade-offs with performance.

[1] The default servers shipped with z/VM do not follow this practice, though in general they have very limited use.
[2] If the z/VM system is using HyperPAV, this can mitigate the impact from other I/O. However, the SFS server will not try to start I/Os to different minidisks on the same disk volume.

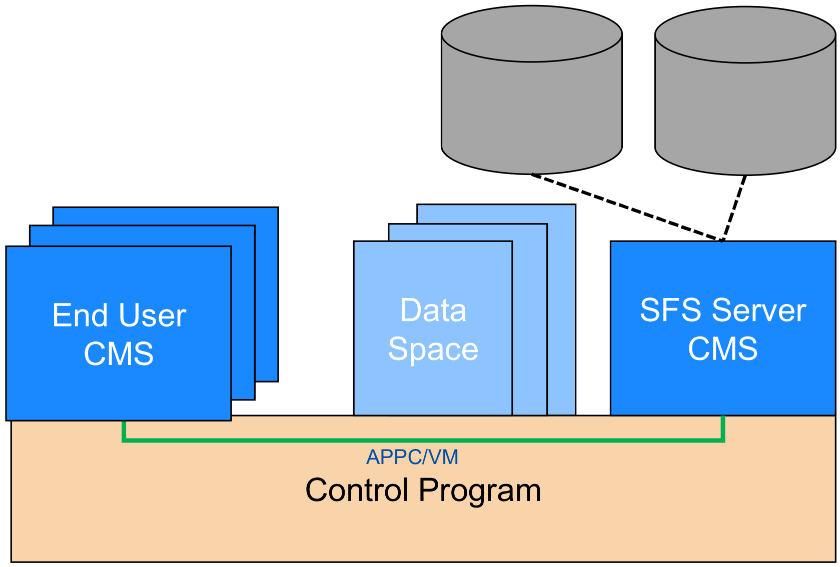
Because the SFS filepool server does I/O on behalf of all the SFS users, disk configuration and placement is important for not just good performance, but for consistent performance.

Log placement – Remember we have two logs disk for redundancy. Placing them on separate disks and channels maximizes the likelihood that the server can do I/O to logs in parallel thus reducing response time.

The Catalog (aka Storage Group 1) has a sizable portion of all I/Os and therefore it may be necessary to spread to prevent any one volume from becoming a bottleneck. A SG1 mdisk is relatively small, I/O intensive area. It will tend to have good storage server cache utilization. While z/VM implemented HyperPAV quite a while after SFS was introduced, the SFS server was never changed to start multiple I/Os to the same volume even if there are supported by HyperPAV.

Data Storage Group is where all the user data lives. When a storage group spans volumes, the server allocates space evenly across those volumes. This tends to spread the I/O demand across those volumes and makes it valuable to have equal performance and space characteristics.

VM Data Spaces



Let's talk a little about SFS use of VM data spaces. Remember that previously that the end user and the file pool server communicate with one another over APPC/VM. This includes passing data back and forth between the file pool server which is doing the I/O and the end user CMS virtual machine. That communication is still there in the data space scenario, but it’s more for sharing other information. The data, both file data and file meta data is going to be shared through the shared memory of the data space. Greater benefit comes from how the I/O for those data spaces is done

VM Data Spaces

- Usage considerations
  - Most benefit from highly used shared R/O or read-mostly data
  - Group updates to minimize multiple versions
  - End users should run in XC[1] mode virtual machines for most benefit
  - Consider using different file pools for R/O vs heavy R/W activity

- Performance advantages
  - Relative to minidisk file system
    - Performance like minidisk with minidisk cache
  - Relative to SFS without data spaces
    - End user retrieves data from shared virtual memory
    - Most communication overhead with SFS server eliminated
    - End users get data directly from data spaces
    - Control blocks describing files (FSTs) are shared in the data space

[1] With service to z/VM 7.2, z/CMS can run in an XC mode virtual machine. See VM66201 for details.

The server (logically) puts directory in VM data space, and user virtual machine takes from VM data space.

The benefit of data spaces is based on the degree of sharing. They provide a great benefit in user virtual storage as the FSTs are shared among accessed users and I/Os as the data is moved from the data space without a trip to the server. So not only does it cut down on the interaction between end user and SFS server virtual machine, it can save virtual memory across virtual machines.

Grouping updates will minimize the likelihood of having multiple versions in data spaces. (discuss ACCESS to RELEASE consistency here). Having users run in XC mode is how the previously stated benefits are achieved.

Separate servers for 1) less scheduled down time for R/O and 2) multiple user rules (discussed later) do not apply.

Performance is similar compared to read-mostly minidisks in minidisk cache. There are measurements that show both ends of the spectrum. It is dependent on workload and storage constraint.

File Pool Recovery

- To minimize time to restore control data
  - Keep file pool from growing too large (number of files, directories, aliases, etc.)
  - Do more frequent backups
  - Do backups to another file pool and get double buffering
  - Specify large **CATBUFFERS**

- To minimize time to restore user data
  - Limit storage group size to meet recovery time requirements
  - Specify large **CATBUFFERS** (~5000 for a 32MB virtual machine)

The following suggestions should minimize the amount of time required to restore the control data of a file pool. "too large" refers to number of objects (files, alias, directories, etc) and is relative to restore rate. Some measurements showed - restore rate = 22Mb/min or 49000 objects/min ; redo rate = 5.3 log blocks/min. The less file pool change activity since the last backup, the less time it will take to apply. SFS can do double buffering on restore when backup is from another file pool. For a 32mb machine try setting CATBUFFERS 5000 this will reduce time to reapply changes to catalog.

Multiple File Pools

- Maximum recommended enrolled users per file pool:

$$\frac{\texttt{Number of system defined users}}{\texttt{Number of system active users}} \times 300$$

  – Number of system defined users = defined in system directory that will use SFS
  – Number of system active users = actively using SFS over a 1-minute interval

  – Does not apply to R/O file pools
  – Assumes normal CMS interactive workload (or normal for those z/VM systems that still have a lot of CMS interactive processing.)

  – Watch involuntary rollbacks and checkpoint processing as possible indicators of too many users.

There is a practical upper limit to the rate at which a server can process requests. This has been expressed in the following formula. System defined users are system CP directory entries for your system. Active users is the average # of users during peak hours who have interacted with the system during a one-minute interval. This can be found using monitor output such as is provided by Performance Toolkit. The gating factors for this calculation are 1) involuntary rollbacks; 2) checkpoint processing. Catalogs are shared, so even if unique data there are locks and potential for deadlocks. Multiple file pools doesn't mean duplicating data.

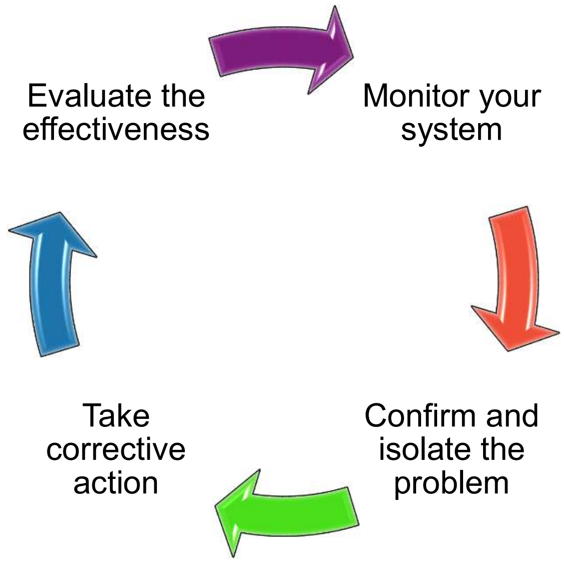
Monitoring Performance

- z/VM monitor data
  - Standard data for each virtual machine (USER domain)
  - SFS-contributed (APPLDATA domain)
- Performance Toolkit reports:
  - FCX116 SFS – Shared file system servers
  - FCX151 SFSIOLOG – Shared file system I/O activity log
  - FCX150 SFSLOG – General shared file system performance log
  - FCX152 SFSREQ – Shared file system requests log
- SFS administrator command **QUERY FILEPOOL STATUS** or **QUERY FILEPOOL REPORT**
  - SFS and CRR counters
  - File pool configuration and disk definitions
  - Agent information
  - Log information
  - Catalog space information
- SFS administrator commands: **QUERY DATASPACE** and **QUERY ACCESSORS** with **DATASPACE** option

Overall monitoring the performance of your system is unchanged if you use SFS. Still check overall system indicators and collect SFS data shown here. Use this data for performance problem determination.

Data for history/trend analysis can come from VM Monitor data. VMPRF uses some of the SFS supplied statistics and combines with other monitor data to produce 3 different reports. The QUERY FILEPOOL STATUS command can be used for immediate snapshot of SFS server. The same counters and timers are involved.

Solving Performance Problems



Most people understand the general performance analysis process. This shouldn't be new. SFS fits right in here, there is no need to really do anything drastically different.

Confirm and Isolate the Problem

- Is it a SFS or general system performance problem?
  - Are all users seeing it?
  - Are minidisk users also seeing it?
  - If processor time or elapsed time increase of application, how does that compare to increase in SFS file pool request processing?
- If its SFS related, look in the z/VM Performance book, Chapter 13. “SFS Tuning” for the Symptom/Causes table.

Table 5. Index of Server Performance Problems

Symptom	Possible Causes	See ...
High CPU time	Excessive remote usage.	“Excessive Remote Usage” on page 132
	Data spaces not used.	“Data Spaces Not Used” on page 132
	Need more processing capacity.	“Need More Processing Capacity” on page 133
High block I/O time	Not enough catalog buffers.	“Not Enough Catalog Buffers” on page 133
	Not enough control minidisk buffers.	“Not Enough Control Minidisk Buffers” on page 133
	SFS file cache is too small.	“SFS Cache is Too Small” on page 134
	Minidisk caching not used.	“Minidisk Caching Not Used” on page 135
	Data spaces not used.	“Data Spaces Not Used” on page 132
	I/O activity not balanced	“I/O Activity Not Balanced” on page 135
	Catalogs are fragmented	“Catalogs Are Fragmented” on page 136
	Logs not on separate paths	“Logs Not on Separate Paths” on page 136
	Need more channels or control units.	“Need More Channels or Control Units” on page 136
	Need more DASD actuators.	“Need More DASD Actuators” on page 136
High ESM time	Excessive external security manager delays.	“Excessive External Security Manager Delays” on page 137
High DFSMS time	Excessive DFSMS delays.	“Excessive DFSMS Delays” on page 137
High lock wait time	Excessive logical unit of work holding time. Logs not on separate paths.	“Logs Not on Separate Paths” on page 136

To make the determination whether it is an SFS or a general system problem, compare the percentage increase in average file pool request service time to the percentage increase in average response time. Average file pool request service time is displayed in the FCX116 SFS or FCX150 SFSLOG reports or can be calculated from the QUERY FILEPOOL STATUS output by dividing File Pool Request Service Time by Total File Pool Requests. If the file pool request time is much greater, then the server is probably contributing to the problem.

Confirm and Isolate the Problem

- Is it a SFS or general system performance problem?
  - Are all users seeing it?
  - Are minidisk users also seeing it?
  - If processor time or elapsed time increase of application, how does that compare to increase in SFS file pool request processing?
- If its SFS related, look in the z/VM Performance book, Chapter 13. “SFS Tuning” for the Symptom/Causes table.

Table 5. Index of Server Performance Problems

Symptom	Possible Causes	See ...
High CPU time	Excessive remote usage.	“Excessive Remote Usage” on page 132
	Data spaces not used.	“Data Spaces Not Used” on page 132
	Need more processing capacity.	“Need More Processing Capacity” on page 133
	Not enough catalog buffers.	“Not Enough Catalog Buffers” on page 133
	Not enough control minidisk buffers.	“Not Enough Control Minidisk Buffers” on page 133
	SFS file cache is too small.	“SFS Cache is Too Small” on page 134
	Minidisk caching not used.	“Minidisk Caching Not Used” on page 135
	Data spaces not used.	“Data Spaces Not Used” on page 132

Table 5. Index of Server Performance Problems

Symptom	Possible Causes	See ...
High CPU time	Excessive remote usage.	“Excessive Remote Usage” on page 132
	Data spaces not used.	“Data Spaces Not Used” on page 132
	Need more processing capacity.	“Need More Processing Capacity” on page 133

Let's just zoom in on the one category of the table.

Take Corrective Action

- Symptoms/Causes table will point to pages that describe possible corrective actions
- Example:

High system paging rate	Too many catalog buffers	“Too Many Catalog Buffers” on page 140
	SFS file cache is too large.	“SFS File Cache is Too Large” on page 141
	Server code not in a saved segment.	“Server Code Not in a Saved Segment” on page 139
	Excessive logical unit of work holding time.	“Users Not Running in XC Mode” on page 141
	Users not running in XC mode.	“Need More Real Storage” on page 141
	Need more real storage.	“Need More Real Storage” on page 141
High response times	ACCESS contention	“ACCESS Contention” on page 142

- Try **ONE** of the possible actions

Now for each problem type there are a series of reasons and possible corrective actions one can take. We’ll pick the one that is most logical in terms of likely to improve the problem and ease of implementation. We pick just one so that we don’t have a scenario of two having opposite effects and making it appear that neither is worth doing.

Evaluate for Effectiveness

- Review performance data
- Examine key performance indicators
- If not acceptable,
  - Correct actions taken?
  - Look at additional improvement options

After reading possible corrective actions, choose one (and only one at a time) and implement it.

An often-skipped step is the validation that the fix really worked. Now on to the case study...

Case Study – VMPRF Report (PRF006)

- Before

```
 RESPONSE_ALL_BY_TIME
   Transaction Response Time and Throughput for ALL Users

             <-----------Response Time---------------->
             <---Triv---> <--Non-Triv-->
From  To                                 Quick
Time  Time       UP      MP       UP      MP   Disp   Mean

09:24 09:54  0.163  0.000  69.095  0.000  9.158  38.635
```

Note: This is an old case study from a time when the Performance Reporting Facility (VMPRF) was an IBM performance product. It is no longer available, but the concepts illustrated would apply with today's Performance Toolkit.

This is an older case study so it was actually based on VMPRF (VM/ESA Performance Reporting Facility) which is no longer available. It was replaced by Performance Toolkit. But the concepts all still apply.

"BEFORE" here means before we get done fixing the system. Ideally we'd like a before the before picture where things are good, then we move to "bad". In this case, things are so bad it is obvious that there is a problem. Response time is horrible. We assume it is SFS since all users with SFS show problem.

We can look further into VMPRF reports at the SFS_BY_TIME report. It's worth spending some time here pointing out stuff. Notice that most of the categories from the symptoms and causes table map to the Time per file pool Request areas. We have 2 file pool servers. We mentioned "deadlocks w/ RB" before. point that out on last column.

Right off bat we know something is wrong since FPR total time is several seconds!! A large chunk of that is in Other. From there, we look at Utilization and see Page Read time is out of sight.

Case Study – VMPRF Report (PRF083)

SFS Activity by time

From Time	To Time	Userid	FPR Count	FPR Rate	<---Time Per File Pool Request---> Total	CPU	Lock	Block I/O	ESM	Other
09:24	09:54	RWSERV1	22545	12.540	3.443	0.004	0.140	1.740	0	1.559
09:24	09:54	RWSERV2	21470	11.942	4.205	0.004	0.190	1.986	0	2.027

	<----Server Utilization-------> Total	CPU	Page Read	Check point	QSAM	<----Agents-----> Active	Held	Deadlocks w/ RB
RWSERV1	75.29	5.47	60.38	9.44	0.00	43.2	152.6	0
RWSERV2	82.95	5.29	67.27	10.40	0.00	50.2	146.7	0

"BEFORE" here means before we get done fixing the system. Ideally, we'd like a before the before picture where things are good, then we move to "bad". In this case, things are so bad it is obvious that there is a problem. Response time is horrible. We assume it is SFS since all users with SFS show problem.

We can look further into VMPRF reports at the SFS_BY_TIME report. It's worth spending some time here pointing out stuff. Notice that most of the categories from the symptoms and causes table map to the Time per file pool Request areas. We have 2 file pool servers. We mentioned "deadlocks w/ RB" before. point that out on last column.

Right off bat we know something is wrong since FPR total time is several seconds!! A large chunk of that is in Other. From there, we look at Utilization and see Page Read time is out of sight.

Case Study – User of Symptoms and Causes Table

Symptom	Possible Causes	See ...
High other time	Insufficient real agents	"Insufficient Real Agents" on page 138
	Too much server paging.	"Too Much Server Paging" on page 138
	Server code not in a saved segment.	"Server Code Not in a Saved Segment" on page 139
	Server priority is too low.	"Server Priority is Too Low" on page 139
	QUICKDSP not specified	"Server Utilization is Too High" on page 140
	Server utilization is too high.	"Server Utilization is Too High" on page 140

- From the description, we can see possible corrective actions such as:
  – Reserve pages
  – Change Share setting
  – Use saved segments for SFS code

So, now we go to our Symptoms and Causes Table and look in the High Other Time symptom.

Case Study – VMPRF Report (PRF008)

```
 USER_RESOURCE_UTIL
   Resource Utilization by User

              Est
Userid ...... WSS  Resid  .....

RWSERV1      1163   1142
RWSERV2      1225   1217
```

- SET RESERVED RWSERV1 1300
- SET RESERVED RWSERV2 1300

Can go back to symptom and cause table then to pointer about "too much server paging". SET RESERVED with WSS .

We can get the value for WSS from VMPRF or INDICATE USER. And issue the above commands.

Case Study – VMPRF Report (PRF006)

- Before

```
Transaction Response Time and Throughput for ALL Users

             <-----------Response Time---------------->
             <---Triv---> <--Non-Triv-->
From  To                                    Quick
Time  Time       UP      MP       UP      MP   Disp    Mean
09:24 09:54  0.163  0.000  69.095  0.000  9.158  38.635
```

- After

```
Transaction Response Time and Throughput for ALL Users

             <-----------Response Time---------------->
             <---Triv---> <--Non-Triv-->
From  To                                    Quick
Time  Time       UP      MP       UP      MP   Disp    Mean
09:52 10:22  0.072  0.000   0.866  0.000  7.396   0.579
```


Being good little performance managers, we look at the after case. The response time is much more acceptable.

We need to go a step further and see if the change in Resp Time is really from what we did. In the after picture, things are much better. We see FPR total time is subsecond, where it should be.

Also notice that the FPR rate has increased. Not only are we getting better response time, but better throughput as well. The Deadlocks w/RB are still zero which is good. You can see that the number of active agents and held agents also decreased. This is all part of the change to avoid serialization from page faults.

This case study was a gross problem, but is sufficient to show the methodology.

Case Study – VMPRF Report (PRF083)

- Before

```
                                 <---Time Per File Pool Request--->
From  To             FPR   FPR                         Block
Time  Time  Userid   Count Rate   Total  CPU   Lock   I/O ESM Other
09:24 09:54 RWSERV1 22545 12.540 3.443 0.004 0.140 1.740   0 1.559

       <----Server Utilization-------> <----Agents----->
       Page  Check                      Deadlocks
       Total   CPU   Read  point  QSAM  Active Held   w/ RB
RWSERV1 75.29  5.47  60.38   9.44  0.00  43.2  152.6    0
```

- After

```
From  To             FPR   FPR                         Block
Time  Time  Userid   Count Rate   Total  CPU   Lock   I/O ESM Other
09:52 10:22 RWSERV1 63617 35.343 0.158 0.003 0.002 0.051   0 0.103

        <----Server Utilization-------> <----Agents----->
                      Page  Check                        Deadlocks
        Total   CPU   Read  point  QSAM  Active Held    w/ RB
RWSERV1 39.51 11.64  15.44  12.43  0.00   5.6    9.5     0
```


Being good little performance managers, we look at the after case. The response time is much more acceptable.

We need to go a step further and see if the change in Resp Time is really from what we did. In the after picture, things are much better. We see FPR total time is subsecond, where it should be.

Also notice that the FPR rate has increased. Not only are we getting better response time, but better throughput as well. The Deadlocks w/RB are still zero which is good. You can see that the number of active agents and held agents also decreased. This is all part of the change to avoid serialization from page faults.

This case study was a gross problem but is sufficient to show the methodology.

Some Application Performance Tips

- See *CMS Application Development Guide* book for more details
- Use direct reference vs. ACCESS command
  - If very few file operations, then use direct reference
  - Lots of file operations, then do access first
- Use hierarchical directories to minimize the number files accessed
- Use **DMSEXIFI** instead of **DMSEXIST** when applicable
- Replace the file directly instead of create temporary file / erase / rename

As users become more comfortable with SFS they will write or use applications that exploit SFS. It is good to understand the performance impacts.

•A trade off between reference methods exists. If there are only a few operations use direct referencing, but we many ACCESS the directory. Per request 'direct referencing' is slightly more expensive.

•To save virtual storage references and search overhead, minimize the number of files accessed by utilizing tree structure.

•DMSEXIFI allows us to use info cached in end user and therefore avoid some server requests.

Understanding Application Performance

- Create a test file pool server or get dedicated or low activity time
- Create wrapper exec
- Collect data and do a little math

```
/* Measure the specified function */
ARG function
"QUERY FILEPOOL STATUS"
time=TIME('R')

/* Put for application or function here */
time=TIME('R')

SAY "Elapsed time is" time "seconds"
"QUERY FILEPOOL STATUS"
```

```
Elapsed time is 1.3 seconds
Q FILEPOOL STATUS (selected values)
Initial    Final    Delta     Counter Name
-------    -----    -----     ------------
     14       15        1     Refresh Directory Requests
  13018    13136      118     File Pool Request Time (msec)
  23795    23904      109     Total BIO Request Time (msec)
   1726     1745       19     Total I/O Requests
```


At times you want to evaluate an application of your own or to be added to system. Foil describes method. Note in this example, the sfs time (118 milliseconds) is a small part of application time (1.3 seconds).

References

- Primary Sources in z/VM Library - https://www.vm.ibm.com/library/
  – CMS File Pool Planning, Administration, and Operation
  – Performance
  – CMS Application Development Guide

- Others
  – CP Planning and Administration
  – CP Command and Utility Reference
  – Performance Toolkit Reference
  – CMS Planning and Administration
  – CMS Callable Services Reference

Summary

- Consider performance when creating a file pool
- Follow normal performance methodology
- SFS provides great performance information
  - Realtime via **QUERY FILEPOOL REPORT**
  - z/VM Monitor data stream for real time or post processing
- Read the books. A lot of background in them

When performance is considered upfront, there should be no performance problems. SFS performance doesn't need constant attention, but periodically check it out.

Bottom line is VM tried to make SFS performance management as painless as possible. Both by automating and by documentation. If you find this not to be the case, we need to know. We can't fix what we don't know about.

Do you want to learn even more about SFS performance management? Then check out SFS Performance Management Part II: Mission Possible. You can find that on the https://www.vm.ibm.com/library/