



IBM Enterprise Content Management for Linux on z Systems Scale-Out Case Study (Part 1): Single ECM Node with XFS and IBM Spectrum Scale 4.2



IBM Enterprise Content Management for Linux on z Systems Scale-Out Case Study (Part 1): Single ECM Node with XFS and IBM Spectrum Scale 4.2

Before using this information and the product it supports, read the information in “Notices” on page 59.

Edition notices

© Copyright International Business Machines Corporation 2016. All rights reserved.

U.S. Government Users Restricted Rights — Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
--------------------------	----------

Tables	vii
-------------------------	------------

About this publication	ix
Notational conventions.	ix

Chapter 1. Introduction	1
Objectives of this white paper	1
IBM Enterprise Content Management	1
IBM FileNet P8 Content Platform Engine	2
IBM Content Navigator.	2
IBM Spectrum Scale Version 4.2	3
XFS filesystem.	4

Chapter 2. Summary	7
-------------------------------------	----------

Chapter 3. Description of the SUT used with Part 1	9
---	----------

Chapter 4. Hardware and software configuration of the SUT	13
IBM z Systems server	13
Networking setup	13
IBM storage subsystem setup	13
Linux virtual machines	14
IBM ECM server.	14
IBM HTTP Server	14
IBM DB2 ECM database server	15
WAS Deployment Manager	15
IBM Tivoli Directory Server	16
Workload generator for ICN.	16

Chapter 5. Setting up and tuning the SUT	17
Setting up z/VM	17
Setting up the virtual network	17
Setting up HyperPAV	17
Setting up the Linux operating system	18
Setting up the network	18
Setting up the disk I/O	18
Other Linux operating system adaptations.	23
Tuning the ECM gateway.	24
Maximum number of network clients for IBM HTTP Server	24
Enterprise Content Management application tuning	24
Common WebSphere Application Server tuning	24
WebSphere Application Server Java virtual machine	26

Enterprise Content Management database server	27
DB2 workload type.	27
Results of Enterprise Content Management tuning	30

Chapter 6. Enterprise Content Management workload	33
FileNet Advanced File Storage Area	34
Advanced File Storage Area on an XFS filesystem	35
Advanced File Storage Area on Spectrum Scale	35

Chapter 7. IBM Spectrum Scale tuning parameters	37
Linux system clock synchronization	37
IBM Spectrum Scale cache usage	37
Setting the maxFilesToCache parameter	37
Setting the pagepool size	39
Relation between pagepool size and level of maxFilesToCache.	40
Setting the value of maxStatCache	42
Memory consideration for IBM Spectrum Scale cache usage	42
IBM Spectrum Scale filesystem parameters	43
Setting the value of blocksize	43
Setting a value for the number of inodes	43

Chapter 8. Single Enterprise Content Management (ECM) node results	45
Enterprise Content Management user scaling for XFS	46
Transaction throughput and z/VM total CPU load	46
Average ICN transaction response times.	47
ECM user scaling for IBM Spectrum Scale single node cluster	48
Average ICN transaction response times.	49
FileNet file storage area disk I/O	50
z/VM total CPU load for the ECM system	52

Chapter 9. Multiple ECM node setup with Spectrum Scale 4.2 (Part 2 white paper)	55
--	-----------

References	57
-----------------------------	-----------

Notices	59
Trademarks	61
Terms and conditions	61

Figures

1. IBM Spectrum Scale at a glance	4	9. XFS versus Spectrum Scale (different pagepool sizes and mFTC values)	41
2. How the SUT was configured for the XFS FNP8 CPE File Storage Area	10	10. Spectrum Scale maxFilesToCache metadata for open files	42
3. How the system under test (SUT) was configured for the Spectrum FileNet File Storage Area	11	11. Normalized ICN page throughput and z/VM total CPU load	46
4. CPU loads on ECM database server with and without the additional database index applied .	30	12. Average ICN transaction response times	48
5. Summarized results of ECM application tuning	31	13. Average ICN transaction response times for XFS and Spectrum Scale	49
6. Virtual user activity flow and workload levels	34	14. Spectrum Scale and XFS - FileNet File Storage Area disk IO rates	51
7. Spectrum Scale maxFilesToCache	38	15. Spectrum Scale and XFS - z/VM total CPU loads and ICN page throughput	52
8. Spectrum Scale startIO calls for different pagepool sizes	40		

Tables

1. Notation conventions	ix	9. Dmgr software	16
2. ECM virtual machine resources	14	10. ITDS virtual machine	16
3. ECM virtual machine software	14	11. ITDS software.	16
4. IHS virtual machine resources	14	12. x86_64 hardware.	16
5. IHS software	15	13. Software running on x86_64 machine	16
6. ECM database server virtual machine resources	15	14. Initial content population for the FileNet repositories	34
7. ECM database server software	15	15. Scale-out case study SUT load level definitions	45
8. Dmgr virtual machine resources	15		

About this publication

This white paper is the first of two closely-related white papers that describe *IBM® Enterprise Content Management (ECM)* scale-out case studies for Linux on z Systems. This white paper examines the setup, tuning, and performance aspects of an ECM system that is implemented as a virtual environment under z/VM®. The ECM system is based on the *IBM FileNet® P8 5.2.1* and *IBM Content Navigator 2.0.3* applications.

The two related white papers can be summarized as follows:

- *Scale-Out Case Study Part 1 - Single ECM Node with XFS and IBM Spectrum Scale 4.2* (this white paper), which focuses on a single ECM node with an XFS file system or IBM Spectrum Scale™ as FileNet File Storage Area. The setup, tuning and performance analysis of the ECM components and IBM Spectrum Scale are described for the “system under test” (SUT).
- *Scale-Out Case Study Part 2 - Multiple ECM Nodes with IBM WebSphere Application Server Cluster and Spectrum Scale 4.2* (the second white paper), which focuses on the scale-out to multiple ECM nodes in a WebSphere® Application Server (WAS) Cluster, and the effectiveness of IBM Spectrum Scale as a distributed parallel filesystem. In addition, the Spectrum Scale cluster configurations *Shared Disk (SD)* and *Network Shared Disk (NSD)* are tested.

Authors

IBM Linux end-to-end Performance team: Thomas Weber, Dr. Juergen Doelle

IBM Spectrum Scale team: Tomer Perry

IBM ECM team: Michael V. Bordash, Nhan Hoang, Dave Royer

Remarks

The web-links referred in this paper are up-to-date as of June, 2016.

Notational conventions

The notational conventions used throughout this white paper are described here.

Table 1. Notation conventions

Symbol	Full name	Derivation
KiB	kibibyte	2 ** 10 byte == 1024 byte
MiB	mebibyte	2 ** 20 byte == 1048576 byte
GiB	gibibyte	2 ** 30 byte == 1073741824 byte
KiB/s	kibibyte per second	2 ** 10 byte / second
MiB/s	mebibyte per second	2 ** 20 byte / second
GiB/s	gibibyte per second	2 ** 30 byte / second

Chapter 1. Introduction

Introductory information is provided in this topic about the setup, tuning, and performance aspects of an ECM system scale-out study for Linux on z Systems. The ECM system is implemented as a virtual environment under z/VM and is based on the *IBM FileNet P8 5.2* (FNP8 5.2) and *Content Navigator 2.0* (ICN 2.0) applications.

Objectives of this white paper

IBM ECM solutions use analytics to provide access to content for the right people at the right time to help make better decisions. ECM solutions with optimized performance can better deal with the large amounts of content data that is typically involved.

The ECM solution implemented for this case study runs in a virtualized environment on the IBM z Systems™ platform with several layers of performance tuning.

Starting from the z Systems™ hardware, z/VM hypervisor, Linux operating system to ECM applications. Especially the multi-node ECM solution with a shared FileNet P8 File Storage Area requires a reliable parallel filesystem with good performance like IBM Spectrum Scale.

One of the major objectives for this case study is to demonstrate the strength of IBM ECM and Spectrum Scale as a winning solution for Linux on z Systems.

IBM Enterprise Content Management

IBM Enterprise Content Management (ECM) high-value solutions can help companies transform the way they do business by enabling companies to put content in motion by capturing, activating, socializing, analyzing, and governing it throughout the entire lifecycle.

ECM solutions help companies and government agencies reduce costs and make quick, smart, and cost-effective decisions. ECM solutions can be organized into these areas:

Essential ECM

Manages essential content anywhere. Production imaging and capture, enterprise report management, office document management (ODM), and standardization and consolidation are content management foundations for reducing costs and improving efficiencies.

Advanced case management

Takes workflow and business process management to the next level. It brings people, processes, and information together to drive better case outcomes.

Information lifecycle governance

Expands the value of repositories. It can reduce costs and risks while improving compliance posture.

Trusted content analytics

Derives unexpected content insights.

ECM solutions take care of content, security, storage, workflows, decision-making, and productivity. They also support rapid application development, using Web 2.0 technology in user-oriented application environments. That accelerates the creation of solutions and reduces dependence on IT resources.

IBM FileNet P8 Content Platform Engine

IBM FileNet P8 (FNP8) Content Platform Engine (CPE) is a component that is designed to handle the heavy demands of a large enterprise.

FNP8 CPE can manage enterprise-wide workflow objects, custom objects, and documents by offering powerful and easy-to-use administration tools. Using the tools, an administrator can create and manage the classes, properties, storage, and metadata that form the foundation of an ECM system.

For more information on the Content Platform Engine, proceed to the FileNet P8 Platform 5.2.1 documentation at the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8toc.doc/welcome_p8.htm

Next, select **System overview > FileNet P8 architecture > Content Platform Engine**.

IBM Content Navigator

IBM Content Navigator (ICN) is a Web client that provides users with a console for working with content from multiple content servers.

ICN also enables users to create custom views of the content on the Web client by creating teamspaces, which provide a focused view of the relevant documents, folders, and searches that a team needs to complete their tasks.

ICN also includes a powerful API toolkit that you can use to extend the Web client and build custom applications. ICN enables users to search for and work with documents that are stored in content servers that are located around the world from a Web browser.

ICN can be used to access multiple content management repositories. You can also configure the same instance of ICN to access multiple types of content management servers, depending on your business needs. You can use ICN to connect to:

- IBM Content Manager Enterprise Edition repositories
- IBM Content Manager OnDemand repositories
- IBM FileNet P8 repositories
- OASIS Content Management Interoperability Services (CMIS) repositories

For more information on the IBM Content Navigator, proceed to the Content Navigator 2.0.3 documentation at the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSEUEX_2.0.3/contentnavigator_2.0.3.htm

Then select **IBM Content Navigator overview**.

IBM Spectrum Scale Version 4.2

IBM Spectrum Scale Version 4.2 is a proven, scalable, high-performance data and file management solution that is based upon the *IBM General Parallel File System (GPFS™)*.

Spectrum Scale provides comprehensive storage management with a high degree of scalability, flash-accelerated performance, and automatic policy-based storage. Furthermore, it can considerably reduce storage costs while improving security and management efficiency in cloud, big data, and analytics environments.

IBM Spectrum Scale 4.2 includes various integrated *Information Lifecycle Management (ILM)* tools that allow you to automatically move data based on *policies*. This can significantly reduce operational costs, since fewer system administrators can manage larger storage infrastructures.

Note: To see which Spectrum Scale functions are supported for your Spectrum Scale version or operating system, refer to the IBM Spectrum Scale FAQs.

Software Defined Storage lets you build your infrastructure your way:

- Easy to scale with relatively inexpensive commodity hardware while maintaining world class storage management capabilities.
- Supports any combination of flash, spinning disk, and tape.
- Supports various cluster configurations that include storage area networks (SANs), Network Shared disk, and Shared Nothing clusters.
- Supports addition of more storage capacity without affecting the application to greatly simplify administration.
- Spectrum Scale is available for IBM z Systems.

Spectrum Scale 4.2:

- Provides global data access across geographic distances and unreliable WAN connections.
- Provides proven reliability with use in the most demanding commercial applications.
- Protects data from most security breaches, unauthorized access, or being lost, stolen, or improperly discarded with native file encryption for data at rest and secure erase.
- Provides multi-site support, connecting local Spectrum Scale cluster to remote clusters to provide Disaster Recovery configurations.

IBM Spectrum Scale provides additional protocol access methods in the Standard and Advanced editions of the product. Providing these additional file and object access methods and integrating them with Spectrum Scale offers several benefits:

- It enables users to consolidate various sources of data efficiently in one global name space.
- It provides a unified data management solution and enables not just efficient space utilization but also avoids having to make unnecessary data moves just because access methods may be different.

The additional protocol access methods integrated with Spectrum Scale are file access using Network File System (NFS) and Server Message Block (SMB) and object access using OpenStack Swift. While each of these server functions (NFS,

SMB, and Object) uses open source technologies, this integration adds value by providing the ability to scale and by providing high availability using the clustering technology in Spectrum Scale.

- IBM Spectrum Scale provides policy-driven compression to reduce the size of data at rest.
- IBM Spectrum Scale provides block storage for OpenStack deployments via its cinder driver support.
- IBM Spectrum Scale is a POSIX-compliant file system that offers an enterprise-class alternative to HDFS and is a preferred platform for data analytics where it's Hadoop compatibility extensions help replace HDFS in a Hadoop ecosystem, with no changes required to Hadoop applications.

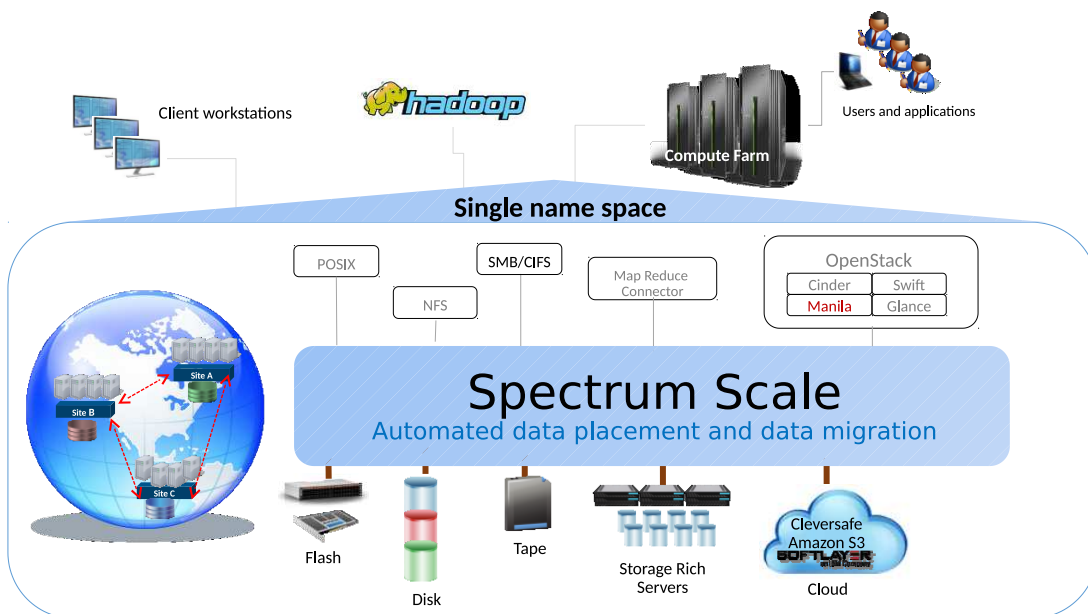


Figure 1. IBM Spectrum Scale at a glance

Spectrum Scale powers many of the world's largest scientific supercomputers and commercial applications that require high-speed access to large volumes of data such as digital media, engineering design, business intelligence, financial analytics, seismic data processing, geographic information systems, and scalable file serving.

XFS filesystem

XFS is a high-performance 64-bit open source filesystem merged into the Linux kernel.

XFS is supported by most Linux distributions, and is even used as the default filesystem for some Linux distributions.

XFS supports large files and large file systems. Therefore it can be used as Linux filesystem solution to store file contents for a FileNet repository (File Storage Area).

XFS is a journaling filesystem and maintains a change log holding the filesystem modifications before they are committed to the filesystem assuring the consistency of the filesystem.

Chapter 2. Summary

With the availability of Spectrum Scale for Linux on z Systems, the question arises as to what are possible workload candidates to share data and would require a parallel filesystem. Applications that run in a WebSphere Application Server (WAS) cluster on multiple nodes are good candidates to share data via a parallel filesystem.

The expectation was that in such a scenario requiring a parallel filesystem, Spectrum Scale would show its capabilities and strengths. An ECM system with IBM FileNet P8 (FNP8) and IBM Content Navigator (ICN) in a WAS cluster was considered to be an ideal candidate for this case study. This is because FNP8 supports Spectrum Scale as a cluster solution and can also store content in a file system location (Advanced File Storage Area) as well as a database repository. In the case of a database repository (Database Object Store), the database would control data access from multiple nodes. Such an ECM solution allows any impacts from the underlying filesystem to be shown, and is at the same time a customer-like workload.

This Part 1 single ECM node case study is followed by a Part 2 multiple ECM node cluster scale-out performance study, which shows the advantages of an ECM cluster solution. The objective of this first part of the study was the setup of a system under test (SUT) that establishes an ECM workload with a focus on filesystem disk I/O. The baseline for the performance test is defined with a single ECM node that uses XFS as Advanced File Storage Area. Next, Spectrum Scale is introduced as filesystem for the Advanced File Storage Area to expose any impact or overhead of a parallel filesystem compared to XFS, without using any cluster capabilities from Spectrum Scale for the single ECM node.

- *ECM application specific tuning fitting the workload*
 - Workload-dependent sizings for WAS WebContainer, ORB Thread Pool, JDBC Connection Pools, JVM heap sizes for ICN and FNP8.
 - Database backend tuning for FNP8 and an example of a performance-specific database index creation. The index was created for a long-running ICN transaction with a quite high response time. After index creation, the average response time for the transaction *was reduced from 7000 milliseconds to 200 milliseconds*. The db2top utility, for example, can be used to identify identify long-running transactions or CPU-expensive SQL statements.
 - With the above tunings applied, the transaction throughput *improved by more than 30%* and the average transaction response time *runs low from 1500 milliseconds seconds to 275 milliseconds*, compared to the “out of the box” behavior.
- *Spectrum Scale specific tuning*
 - Testing proved that the default blocksize of 256 KiB is a good default value for the workload.
 - For ECM workloads, the number of maximum inodes became an important parameter. Typically, ECM systems store massive amounts of data in a filesystem repository and the system can quickly run out of inodes. This causes “No space left on device” errors.
 - Other important parameters for cache usage were shown to be the pagepool size and the maxFilesToCache parameter. It is important to understand that both parameters relate to each other. For example, when using a large number

of small files then increasing the pagepool size requires that the value of `maxFilesToCache` should also be increased. The optimal value depended on specific patterns of the files used. For this case study, the Spectrum Scale `startIO` metrics was used to determine the pagepool cache efficiency. An important point was that the range for the `maxFilesToCache` value with good caching-hit ratio was relatively large for our workload. But for larger pagepool sizes, the parameter might need to be adapted. Setting a value for `maxFilesToCache` that was too high could have a negative impact on the pagepool hit ratio.

- *Finally, the number of virtual ICN users is scaled to vary the load level.* Due to the relative short think-times between transactions performed by the virtual users, the resulting workload is more powerful than with human users. The load levels vary from low to high. At the high end, 23 IFL processors were utilized which was considered to be the highest load level.
 - The SUT showed *very good linear scaling behavior* in relation to throughput rates.
 - In regard to response times, Spectrum Scale was *comparable to, and partially better than XFS* in the range 250 to 750 virtual ICN users. Spectrum Scale only had higher response times than XFS at the highest load level with 1000 virtual users.
 - In regard to CPU load, both filesystems *behaved comparably*.

It is clear that compared to a standard file system, a parallel filesystem needs more resources to manage multiple independent nodes that share a file system and keep the file system consistent. For a FileNet File Storage Area on a single ECM node, Spectrum Scale handles the additional effort *very efficiently*. Spectrum Scale *can compete with XFS for low to medium load levels*, and falls only slightly behind at the highest load level used in this case study.

This Part 1 of the case study showed *how well an ECM system with FNP8 and ICN scales* in a virtualized environment with Linux on z Systems.

Chapter 3. Description of the SUT used with Part 1

The system under test (SUT) was a single ECM node with the applications IBM FNP8 CPE and ICN for Linux on z Systems installed. Both applications ran via a WAS environment.

The SUT also included front-end and back-end server components, such as:

- IBM IHS as Web server
- IBM ITDS as Directory Service Provider
- IBM DB2® database server

The SUT followed a 3-tier architecture model, enabling client components to interact with data resources and legacy applications, with logical tiers distributed across independent systems:

- Client components on external machines (tier one)
- ECM server components on the z Systems server (tier two)
- Back-end components (for example database server) on the z Systems server (tier three)

Tier one components are typically responsible for the presentation and user interaction of second-tier processes. These components never access any third-tier services for example on database servers directly. Tier one would either be:

- A Web browser displaying the Content Navigators user interface.
- A custom application based on the Content Navigator APIs for the SUT.

For this case study, the virtual ICN users were simulated with a workload that was implemented via the IBM Rational® Performance Tester (RPT) and that called various functions from ICN user interface.

Tier two components, also known as the application logic layer, manage the business logic of the applications and provide access to third-tier services. Usually it is the layer where most of the data processing occurs. The application logic layer for the SUT is represented by:

- The Web server IHS
- The ECM components ICN
- FNP8 CPE

Tier three components are protected from direct access by any tier one components. Usually they reside in a secure network (for example inside the z Systems server). Interaction is only possible through the second-tier services. The DB2 ECM database server and Directory Service Provider are considered as third-tier services for the SUT.

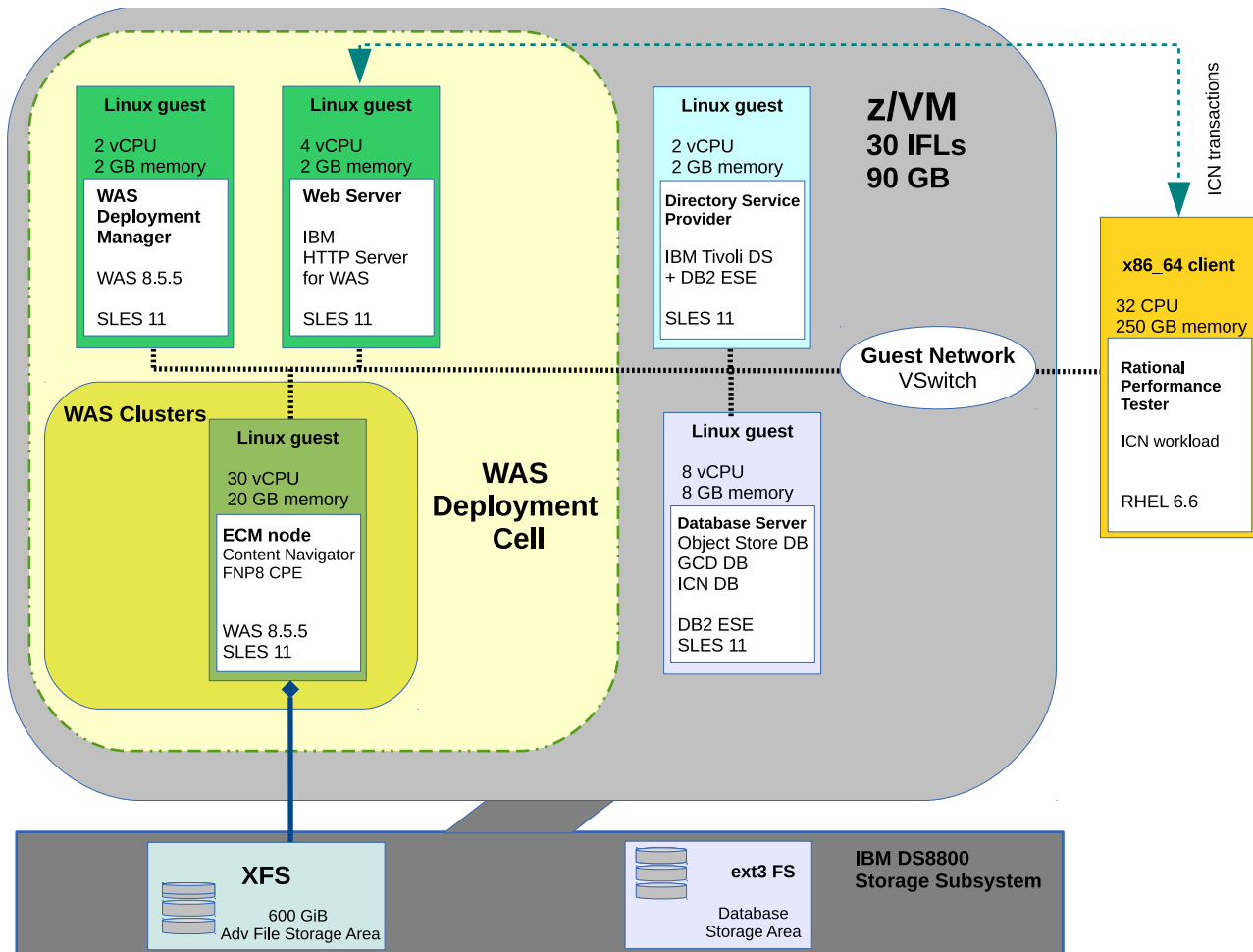


Figure 2. How the SUT was configured for the XFS FNP8 CPE File Storage Area

The SUT was implemented in an IBM z Systems server LPAR that runs a z/VM hypervisor for a virtualized environment.

- The ECM components were installed in single node WAS clusters.
- ICN and FNP8 CPE were each deployed in their own single node WAS cluster.
- The FNP8 CPE Advanced File Storage Area was 600 GiB large and resided either on an XFS highly scalable filesystem or on a Spectrum Scale parallel filesystem for the first part of the case study.

The performance of both filesystems as FileNet Advanced File Storage Area is considered in the results section.

Components of the ICN application were mapped to the Web server IHS. IHS was used to manage the HTTP traffic for ICN and to do the request balancing for the multiple ECM node setup (is described in the follow-on Part 2 white paper).

The WAS clusters, the WebSphere Deployment Manager (Dmgr) and IHS together form a “WebSphere Deployment Cell”.

The IBM DB2 ECM database server maintained three different databases:

- FNP8 CPE Object Store database
- FNP8 Global Configuration Data (GCD) database
- ICN database

The ITDS Directory Service Provider provided authentication methods for usernames and passwords for the ECM system.

For guest-to-guest communication, a z/VM Virtual Switch (VSWITCH) was used that also provided LAN access via a 10GbE OSA-Express® network card.

The load generator for ICN was a x86_64 machine running the IBM Rational Performance Tester (RPT) application with a custom workload. The x86_64 machine was connected over a 10GbE network to the IBM z Systems server.

Note: The WAS cluster was scaled-out to a multi ECM node cluster in the Part 2 follow-on case study. The performance test results for the multi node study are described in the Part 2 white paper.

The next figure basically shows the same layout of the SUT. The only difference is that the FNP8 Advanced File Storage Area is now on a 600 GiB Spectrum Scale filesystem.

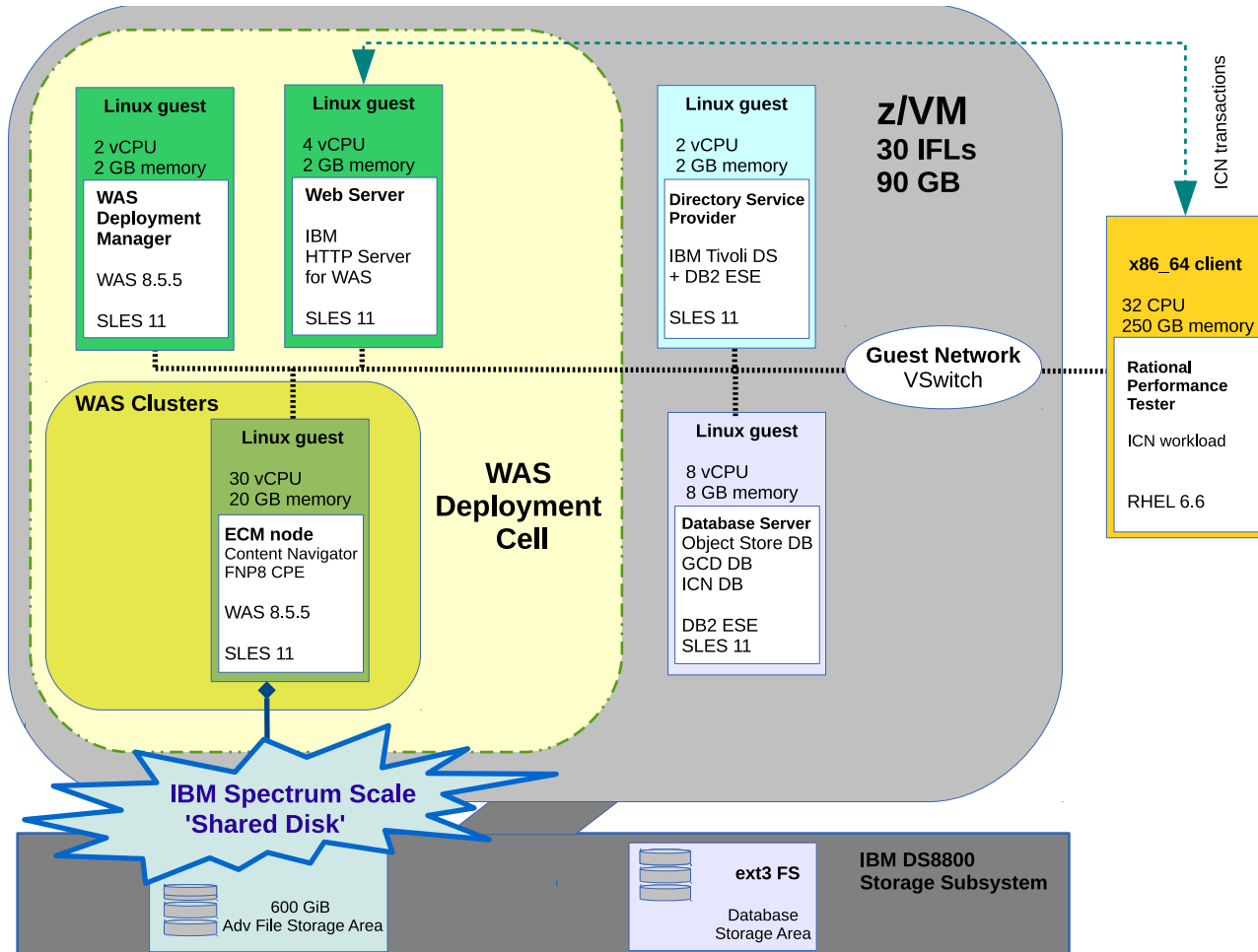


Figure 3. How the system under test (SUT) was configured for the Spectrum FileNet File Storage Area

The single ECM node used the Spectrum Scale cluster configuration “Shared Disk”, which means that the FCP/SCSI disks were directly attached to the node. A single

node Spectrum Scale cluster is the most basic cluster setup with no high availability. This simple Spectrum Scale cluster was used for the comparison to the XFS filesystem.

Chapter 4. Hardware and software configuration of the SUT

This topic describes the software and hardware that was used for the case study. The SUT formed a basic customer-like ECM system with FNP8 CPE and ICN to integrate the FileNet repositories.

The ECM system was implemented in a z Systems Server Logical Partition (LPAR) running a z/VM 6.3 hypervisor. The storage subsystem was an IBM DS8870 connected via FICON® Express.

IBM z Systems server

The case study involved the configuration of an LPAR in a zEnterprise® EC12 (zEC12) Model 2827.

The LPAR setup consisted of:

- 30 Integrated Facility for Linux (IFL) processors.
- 90 GiB Central Storage.
- 1 10GbE OSA-Express4S card.
- FICON Express8S LX (short wave) cards supporting ECKD™ for DASD device access or FCP devices.

Networking setup

The network used in this white paper is described in this topic.

- z/VM Virtual Switch (VSWITCH) for all virtual machines (guest-to-guest communication).
- 10 GbE OSA-Express4S card attached to the VSWITCH for external network connectivity.
- no network encryption.

IBM storage subsystem setup

The storage subsystem used in this case study was an IBM System Storage® DS8870 Model 2422.

These are the DASD disk devices that were used as z/VM Linux guest operating systems disks:

- different DASD models (model-27, model-54 and model-128) directly attached to the z/VM guests.
- DASDs selected from 2 ECKD storage pools (storage pool striping enabled) and picked from both internal storage subsystem servers.
- HyperPAV alias devices used for DASDs.

These are the FCP-attached SCSI devices that were used as ECM data disks (FileNet File Storage Area and ECM database):

- 100GiB and 250GiB SAN-attached SCSI devices.
- SCSI devices are picked from both internal storage subsystem servers.

Linux virtual machines

This topic describes the Linux virtual machines that were configured for the SUT. The SUT was implemented as a distributed setup with software components installed in different z/VM Linux virtual machines. These Linux virtual machines were grouped by their application tasks.

IBM ECM server

The FNP8 CPE and ICN applications represented the ECM system used in the case study.

FNP8 CPE addressed the enterprise content and process management requirements, while ICN is a Web client that provided a console for working with the contents of FileNet repositories.

Table 2. ECM virtual machine resources

ECM virtual machine
30 virtual CPUs
20GiB memory
10GbE LAN via VSWITCH
1 x DASD Model 27 for Linux - 4x HyperPAV aliases directly attached
6 x 100GiB FCP attached SCSI disks for FileNet CPE storage area
2 x 256GiB FCP attached SCSI disks for backup purposes

Table 3. ECM virtual machine software

Software	Service level
SLES11 for z Systems (64-bit)	Service Pack 3 (SP3)
IBM WebSphere Application Server Network Deployment (64-bit)	v8.5.5 Fix Pack 3 (8.5.5.3)
IBM WebSphere SDK for Java™ Technology Edition 7	v7.0.7.0
IBM FileNet CP Engine	v5.2.1
IBM Content Navigator	v2.0.3
IBM Spectrum Scale	v4.2.0.0

IBM HTTP Server

The Web communication for the ECM system was managed via an IBM HTTP Server (IHS).

The IHS received client requests and forwarded them to the ICN application. Components of ICN were mapped to the IHS to manage the HTTP traffic.

The IHS could also act as a load balancer, if there were more than one ECM nodes available.

Table 4. IHS virtual machine resources

IHS virtual machine
4 virtual CPUs
2 GiB memory

Table 4. IHS virtual machine resources (continued)

IHS virtual machine
10 GbE LAN via VSWITCH
1 x DASD Model 27 for Linux - 4 x HyperPAV aliases directly attached

Table 5. IHS software

Software	Service level
SLES11 for z Systems (64-bit)	Service Pack 3 (SP3)
IBM HTTP Server	v8.5.5 Fix Pack 3 (8.5.5.3)

IBM DB2 ECM database server

The database server hosted three databases for the ECM system.

- The *FNp8 CPE Object Store* database is a relational database that contains documents, workflows and other objects.
- The *FNp8 Global Configuration Data* (GCD) database basically stores FileNet configuration metadata.
- Finally, ICN stores configuration data and preferences in its *own* database.

Table 6. ECM database server virtual machine resources

ECM database server virtual machine
8 virtual CPUs
8 GiB memory
10 GbE LAN via VSWITCH
1 x DASD Model 27 for Linux – 4 x HyperPAV aliases directly attached
6 x DASD Model 54 for ICN and GCD database and database log files 24 x HyperPAV aliases directly attached
4 x 256 GiB FCP attached SCSI disks for FNp8 Object Store and database backups

Table 7. ECM database server software

Software	Service level
SLES11 for z Systems (64-bit)	Service Pack 3 (SP3)
IBM DB2 Enterprise Server Edition	v10.5 Fix pack 4 (10.5.0.4)

WAS Deployment Manager

The WAS Deployment Manager (Dmgr) is the main administration process that manages all other application servers in a WAS deployment cell, including node agents and application server processes.

Dmgr is required only for WAS configuration tasks. Therefore, this virtual machine typically had almost no CPU load when the ECM workload was running.

Table 8. Dmgr virtual machine resources

Dmgr virtual machine
2 virtual CPUs
2 GiB memory

Table 8. Dmgr virtual machine resources (continued)

Dmgr virtual machine
10 GbE LAN via VSWITCH
1 x DASD Model 27 for Linux – 4 x HyperPAV aliases directly attached

Table 9. Dmgr software

Software	Service level
SLES11 for z Systems (64-bit)	Service Pack 3 (SP3)
IBM WebSphere Application Server Network Deployment (64-bit)	v8.5.5 Fix Pack 3 (8.5.5.3)

IBM Tivoli Directory Server

The IBM Tivoli® Directory Server (ITDS) provided authentication methods for user names and passwords to the ECM system.

Table 10. ITDS virtual machine

ITDS virtual machine
2 virtual CPUs
2 GiB memory
10 GbE LAN via VSWITCH
1 x DASD Model 27 for Linux – 4 x HyperPAV aliases directly attached

Table 11. ITDS software

Software	Service level
SLES11 for z Systems (64-bit)	Service Pack 3 (SP3)
IBM Tivoli Directory Server	v6.3

Workload generator for ICN

The workload generator for ICN was an x86_64 machine that ran the IBM Rational Performance Tester (RPT) with a custom workload.

The workload mix was a typical ECM load with various calls to functions of the ICN user interface with a focus on tasks that access content in the FileNet File Storage Area.

Table 12. x86_64 hardware

Client hardware	Setup
IBM Flex System® x86_64	<ul style="list-style-type: none"> • 16 CPUs (2.9 GHz) • 250 GiB memory • 10 GbE network card

Table 13. Software running on x86_64 machine

Software	Service level
Red Hat Enterprise Linux Server Release	Release 6.6
IBM Rational Performance Tester	v8.5.1 Fix Pack 3

Chapter 5. Setting up and tuning the SUT

Setup and tuning modifications applied to the SUT are described in this topic. Changes were done for z/VM, Linux for z Systems, WAS and ECM software.

Setting up z/VM

This topic describes the setup that was specified for use with the z/VM hypervisor.

The resource definitions for the z/VM virtual machines are described under Chapter 4, “Hardware and software configuration of the SUT,” on page 13.

Setting up the virtual network

The z/VM virtual machines were connected to a Virtual Switch (VSWITCH) configured LAN for guest-to-guest communication.

A 10 GbE OSA-Express4 was also connected to the VSWITCH and the virtual machines connected to the VSWITCH share the same LAN as the OSA Express card. This allowed the configuration of virtual machine IP addresses from the same network as for the OSA Express card.

For a z/VM virtualized environment, the VSWITCH provides good performance and is the recommended method for internal and external z/VM network connectivity.

For further information about z/VM virtual networking and VSWITCH, refer to:
<http://www.vm.ibm.com/virtualnetwork/>

Setting up HyperPAV

The I/O throughput for ECKD DASD can be improved by using Parallel Access Volumes (PAV) or HyperPAV.

HyperPAV can be helpful when using DASDs as data disks. For example, database servers are candidates that typically do a lot of disk I/O. For this case study, the z/VM hypervisor was set up to assign HyperPAV aliases directly to the virtual machines.

- For details about using HyperPAV, refer to:
<http://www.vm.ibm.com/storman/pav/>
- See also “Setting up the Linux operating system” on page 18.

Setting up the Linux operating system

In addition to the z/VM resource allocations, various tuning activities were required for Linux networking, disk I/O, and some other parameters.

Details about setting up Linux virtual machines running under z/VM are provided in Chapter 4, “Hardware and software configuration of the SUT,” on page 13.

Setting up the network

The ECM software components communicate over a virtual guest-to-guest network provided by the z/VM hypervisor (VSWITCH).

When querying the devices with the Linux **lsqeth** command, the virtual network devices appear within Linux as type GuestLAN QDIO.

This is the command to list Linux qeth-based network devices:

```
# lsqeth eth1
Device name           : eth1
-----
card_type             : GuestLAN QDIO
cdev0                 : 0.0.1000
cdev1                 : 0.0.1001
cdev2                 : 0.0.1002
chpid                 : 02
online                : 1
portname              : osaport
portno                : 0
route4                : no
route6                : no
checksumming          : sw checksumming
state                 : UP (LAN ONLINE)
priority_queueing     : always queue 2
fake_broadcast        : 0
buffer_count          : 128
layer2                : 0
large_send            : no
isolation              : none
sniffer               : 0
```

Setting the buffer count

The buffer count for the qeth-based network devices was increased on all Linux servers.

The number of buffers for inbound traffic was set to 128 (the default is 64) in the file:

```
/etc/udev/rules.d/51-qeth-0.0.1000.rules
```

where the value 1000 represents the first cdev of the virtual network device.

In addition, to enlarge the buffer count this line was added to the udev rule:

```
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.1000", ATTR{buffer_count}="128"
```

Setting up the disk I/O

FCP-attached SCSI disk devices were used for the FNP8 CPE Object Store database and File Storage Area.

ECKD DASD disks were used for Linux root devices and database log files.

Using HyperPAV for ECKD DASD

The I/O throughput for ECKD DASDs disk can be improved by using *Parallel Access Volumes* (PAV) or *HyperPAV*.

z/VM HyperPAV is also described in “Setting up HyperPAV” on page 17.

The PAV hardware feature is especially important when DASDs are used as data devices with high disk I/O rates, but also as Linux root devices.

The Linux DASD device driver can use this IBM System Storage feature to perform multiple concurrent data transfer operations to or from the same DASD device.

To use HyperPAV, there must be base and alias devices available, which require z Systems Input/Output Configurations Data Set (IOCDS) definitions. For HyperPAV on an IBM System Storage subsystem, the alias devices are not exclusively referenced to a certain base device but are eligible for all base devices in the same logical control unit (LCU). Linux handles HyperPAV alias devices in the same way as a normal DASD base devices.

The **chccwdev** command can be used for management or defining the appropriate udev rules for alias devices. When listing the DASD devices with the Linux `lsdasd` command, HyperPAV aliases can be identified via the `alias` status tag. The usage of HyperPAV aliases is completely handled by the Linux kernel and transparent to the user.

Note: The sample command that follows indicates that the base and alias devices belong together according to their UIDs.

The following command can be used to display the DASD base and alias devices with their UIDs:

```
# lsdasd -u
Bus-ID      Name      UID
-----
0.0.c28a    dasda     IBM.75000000TH031.c280.0a.00000000000075620000000000000000
0.0.c2e0    alias     IBM.75000000TH031.c280.xx.00000000000000000000000000000000
0.0.c2e1    alias     IBM.75000000TH031.c280.xx.00000000000000000000000000000000
0.0.c2e2    alias     IBM.75000000TH031.c280.xx.00000000000000000000000000000000
0.0.c2e3    alias     IBM.75000000TH031.c280.xx.00000000000000000000000000000000
```

For details about IOCDS configurations, refer to *z Systems Input/Output Configuration Program User's Guide*, SB10-7037-11a. You can also obtain further information at the *IBM Resource Link*® Web page (a Sign-On is required):

<http://www.ibm.com/servers/resourceLink>

For more information about PAV and HyperPAV, refer to *How to Improve Performance with PAV*, SC33-8414. You can obtain this document at this IBM developerWorks® Web page:

http://www.ibm.com/developerworks/linux/linux390/development_documentation.html

The white paper *Oracle Database on Linux on System z - Disk I/O Connectivity Study* describes how HyperPAV can be used with databases. You can obtain this document at this IBM developerWorks Web page:

http://www.ibm.com/developerworks/linux/linux390/perf/tuning_database.html#con

FCP Hardware Data Router support

The latest Fibre Channel Protocol (FCP) devices have hardware data router support. This feature can improve the throughput for FCP/SCSI devices by reducing the path length when transferring the data from memory to an FCP device.

These are the minimum requirements for using the FCP hardware data router with Linux on z Systems:

- zEnterprise 196 GA2 and zEnterprise 114 with FICON Express8S.
- z/VM 6.3 or later, that introduced guest support for hardware data router.
- KVM for IBM z Systems.
- RHEL 6.4 or SLES11 SP3 (must be manually enabled).
- For RHEL 7 or SLES12 (are enabled per default).

The command for displaying the availability of an FCP hardware data router for a z/VM guest is:

```
CP Q V FCP
FCP 1722 ON FCP 1722 CHPID 55 SUBCHANNEL = 0000
    1722 DEVTYPE FCP VIRTUAL CHPID 55 FCP REAL CHPID 55
    1722 QDIO-ELIGIBLE QIOASSIST-ELIGIBLE
    1722 DATA ROUTER ELIGIBLE
```

To enable FCP data router support with a Linux kernel parameter (if not enabled by default):

- set the 'zfcp.datarouter=1' Linux kernel parameter

The ways in which the the status of the FCP hardware data router can be displayed in Linux are:

```
# systool -m zfcp -v
Module = "zfcp"

Attributes:
  initstate      = "live"
  refcnt         = "48"
  srcversion     = "52F4E65D11CC52E53FD2D0D"
  supported      = "Yes"

Parameters:
  ...
  datarouter    = "Y"
  ...
```

or

```
# cat /sys/module/zfcp/parameters/datarouter
Y
```

Note: The FCP hardware data router feature becomes active only for FCP devices that are based on adapter hardware with data routing support.

Linux Device Mapper-Multipathing (DM-MP)

The Linux *Device Mapper-Multipathing* (DM-MP) provides multipathing for Linux block devices and is the preferred solution for multipathing with FCP/SCSI devices for Linux on z Systems.

The `multipathd` user-space daemon can use multiple physical I/O paths between server nodes and FCP/SCSI devices in storage subsystems. The aggregated paths appear as a single device via the Linux device mapper. DM-MP also supports request re-routing (*failover*), and *request balancing* across available paths.

The following Linux commands display a **single** FCP/SCSI device that can be addressed via four paths by using four different FCP device and WWPN combinations. Note that the SCSI logical unit number (LUN) is the same for the listed SCSI device. Such a setup is a prerequisite for performing DM-MP.

In the following examples:

- **ls SCSI** lists the SCSI devices and the paths to these devices.
- **lszfc -D** lists the SCSI devices and zfc units.

```
# ls SCSI
[0:0:32:1086865428]disk  IBM      2107900      .307  /dev/sda
[1:0:26:1086865428]disk  IBM      2107900      .307  /dev/sdb
[2:0:29:1086865428]disk  IBM      2107900      .307  /dev/sdc
[3:0:16:1086865428]disk  IBM      2107900      .307  /dev/sdd

# lszfc -D
0.0.1731/0x500507630900c7c1/0x401440c800000000 0:0:32:1086865428
0.0.17b1/0x500507630903c7c1/0x401440c800000000 1:0:26:1086865428
0.0.1831/0x500507630908c7c1/0x401440c800000000 2:0:29:1086865428
0.0.5031/0x50050763090bc7c1/0x401440c800000000 3:0:16:1086865428
```

Enabling DM-MP

DM-MP was enabled with the following three steps:

Step 1: Ensure the `multipathd` daemon is running.

```
# rcmultipathd status
Checking for multipathd:      running
```

Step 2: Create a multipathing configuration in `/etc/multipath.conf`. The following sample is used throughout this case study.

```

defaults {
    user_friendly_names yes
}
blacklist {
    devnode "*"
}
blacklist_exceptions {
    devnode "^sd[a-z]+[0-9]*"
    device {
        vendor "IBM"
        product "S/390.*"
    }
}
devices {
    device {
        vendor "IBM"
        product "^2107900"
        path_grouping_policy "multibus"
        path_checker "tur"
        features "1 queue_if_no_path"
        rr_min_io_rq "100"
        # from device defaults
        hardware_handler "0"
        prio "const"
        rr_weight "uniform"
    }
}

multipaths {
    multipath {
        wwid "36005076309ffc7c100000000000014c8"
        alias "scsi14c8"
    }
    ...
}

```

The sample contains these variables and settings:

defaults section

Defines default values for attributes.

blacklist section

Defines which devices should be excluded for multipathing discovery.

blacklist_exceptions section

Defines which devices should be included for multipathing discovery.

devices section

Defines the device-specific settings:

path_grouping_policy "multibus"

Traffic is load-balanced across all active paths in the group.

path_checker "tur"

The method used to determine the paths state. "tur" is the preferred setting if the LUN supports it.

features "1 queue_if_no_path"

Set up I/O queuing for scenarios where all paths fail.

rr_min_io_rq "100"

Specifies the number of I/O requests to route to a path before switching to the next path in the current path group.

multipaths section

Defines the multipath topologies indexed by a World Wide Identifier (wwid)
The optional alias specifies a symbolic name for the multipath map.

Step 3: Check the status of DM-MP after the changes have been applied (the number of paths, and so on).

```
# multipathd -k'show topo'
scsil4c8 (36005076309ffc7c100000000000014c8) dm-0 IBM ,2107900
size=256G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
   | - 0:0:21:1086865428 sda 8:0 active ready running
   | - 1:0:21:1086865428 sdb 8:16 active ready running
   | - 2:0:18:1086865428 sdc 8:32 active ready running
   `-- 3:0:17:1086865428 sdd 8:48 active ready running

# multipathd -k'show maps status'
name      failback queueing paths dm-st write_prot
scsil4c8 -      on        4      active rw
```

Note that changes to `/etc/multipath.conf` are not effective when `multipathd` is running. To apply the changes:

1. Stop the `multipathd` daemon and clear the multipath bindings.
2. Trigger the creation of new multipath maps.
3. Restart `multipathd`.

For further information, proceed to the IBM Knowledge Center topic describing how to use FCP-attached SCSI devices with Linux on System z®:

https://www.ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_ts.html

Other Linux operating system adaptions

For the ECM applications and the database server, it was necessary to increase the Linux limits for number of open files and processes for servers on which a WebSphere Application Server runs.

The limits can be permanently set in the Linux configuration file:

`/etc/security/limits.conf`

This example shows the limits that were specified for the number of open files and processes:

```
# cat /etc/security/limits.conf
# /etc/security/limits.conf
#
#Each line describes a limit for a user in the form:
#
#<domain>      <type> <item>          <value>
...
*               -      nofile          65512
*               soft   nproc           32768
*               hard   nproc           32768
```

Tuning the ECM gateway

The Web communication for the ECM system was performed via an IBM HTTP Server (IHS) running in a single virtual machine.

For example, IHS received requests from either:

- custom applications
- users interacting directly with the IBM Content Navigator (ICN) user interface.

Components of ICN were mapped to IHS which then managed the HTTP traffic for the ICN application.

Maximum number of network clients for IBM HTTP Server

This `MaxClients` variable is used to increase the number of IBM HTTP Server (IHS) network clients to accommodate the amount of concurrent load to be expected on the system.

The highest ECM workload level that was used for the scale-out test involved 1000 virtual ICN users at a maximum.

In this case study:

1. File `/opt/IBM/HTTPServer/conf/httpd.conf` was edited so that the value of `MaxClients` was increased from 600 to 1000.
 2. The IBM HTTP Server was restarted.

Enterprise Content Management application tuning

This topic describes the tuning that was performed for the ECM (Enterprise Content Management) application tier.

The measurement series for the case study were performed with these tunings applied. Almost all of the parameters described here were used for tuning the performance of the WebSphere Application Server (WAS).

Common WebSphere Application Server tuning

The WAS tuning parameters described in this topic applied to both the *IBM Content Navigator* (ICN) and *FNP8 CPE* WAS servers.

Note: The highest ECM load level used for the scale-out test involved 1000 virtual ICN users at a maximum. The WAS tunings described here were increased to accommodate the maximum amount of concurrent virtual ICN users.

Thread Pool WebContainer

The WAS Thread Pool WebContainer maximum value was increased. Increasing the maximum vastly improved scalability for the FNP8 CPE servers.

In this case study:

1. The minimum/maximum values of WebContainer were changed from 50/50 to 50/1000.
2. The WAS was restarted.

WAS Administration console path: **Servers > Application servers > *server_name* > Thread Pools > WebContainer.**

Object Request Broker services

There are various settings that can be used to tune the internal Object Request Broker (ORB) processing.

ORB Thread Pool settings

A thread pool enables components of the server to reuse threads, which eliminates the need to create new threads at run time. Creating new threads expends time and resources.

In this case study:

1. The maximum size of the ORB Thread Pool was increased from 50 to 1000 threads.
2. The WAS was restarted.

WAS Administration console path: **Servers > Application servers > *server_name* > Container Services > ORB service > Thread Pool Settings.**

ORB fragmentation custom property

ORB fragmentation can be disabled by setting the CORBA fragmentsize set to zero. This helps to decrease FNP8 CPE CPU load and increase throughput.

However, if there are a lot of network packet retransmits in your environment it is not recommended to disable ORB fragmentation because the complete ORB request has to be resend.

In this case study:

1. The value of `com.ibm.CORBA.FragmentSize` was set to zero.
2. The WAS was restarted.

WAS Administration console path: **Servers > Application servers > *server_name* > Container Services > ORB service > Custom Properties.**

For more information about using the CORBA fragmentsize parameter for FileNet P8, refer to the article “Tune ORB in WebSphere to boost FileNet P8 performance” at this IBM developerWorks URL:

<http://www.ibm.com/developerworks/data/library/techarticle/dm-1004tuneorbperformance/index.html>

Java Database Connectivity connection pools

WAS *connection pooling* can improve the response time of any application that require connections.

Requests over the Web to a data resource are done through WAS *data sources*, for example the *FNPP8 Object Store database* or *ICN database*. These databases can be accessed through WAS data sources.

Typically ECM applications request large amounts of data and the total overhead can become very high. However, with appropriate connection pool sizes, ECM applications can improve performance and achieve better response times.

It is important to ensure that the database can have enough maximum connections to match the maximum number of JDBC connections.

The Java database connectivity (JDBC) connection pools for the FNP8 Object Store and the ICN databases are increased.

In this case study:

- The value of Maximum connections was increased from 10 to 1000 connections.

WAS Administration console path: **Resources > JDBC > Data sources > data_source_name > Connection pool properties.**

For further information about WAS connection pooling, proceed to this IBM Knowledge Center URL:

http://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5

Then select **Tuning performance**.

WebSphere Application Server Java virtual machine

Increasing the Java virtual machine (JVM) heap size will often improve performance for complex applications. JVM *heap size monitoring* can help to find suitable heap sizes for your Java applications.

IBM FileNet P8 Content Platform Engine (CPE)

The JVM parameters are described here that were applied during the case study for FNP8 CPE.

In this case study, these JVM parameters for FNP8 CPE were used:

1. The values of initial/maximum heap size were set to **8192MB/8192MB**.
2. These generic JVM arguments were defined:
-Duser.language=en -Duser.region=US -Duser.country=US -Xmn4096m -Xgcpolicy:gencon
3. The WAS was restarted.

WAS Administration console path: **Servers > Application servers > server_name > Java and process management > Process Definition > Java Virtual Machine.**

For information about tuning IBM FileNet P8, proceed to this URL at the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1

Then select: **Performance tuning** and proceed to the Web page that describes how to tune Content Platform Engine application servers.

IBM Content Navigator

JVM parameters applied for the IBM Content Navigator during the case study are described here.

In this case study:

1. The values of initial/maximum heap size were set to **4096MB/4096MB**.
2. These generic JVM arguments were defined:
 - Duser.language=en -Duser.region=US -Duser.country=US
 - Xmn2048m -Xgcpolicy:gencon -Djaxws.payload.highFidelity=true
3. The WAS was restarted.

WAS Administration console path: **Servers > Application servers > *server_name* > Java and process management > Process Definition > Java Virtual Machine**.

For information about Content Navigator tuning, proceed to this URL at the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSEUEX_2.0.3

Then proceed to the Web page that describes how to tune the IBM Content Navigator.

Enterprise Content Management database server

This topic describes the tuning for the ECM database server, in accordance with the guidelines for creating a FNP8 Object Store database.

DB2 workload type

In this topic the FileNet workload type for DB2 is specified.

Since DB2 version 9.7 (and DB2 version 9.5 fix pack 5) the workload type `FILENET_CM` is introduced as a value for `DB2_WORKLOAD`.

`DB2_WORKLOAD` is a DB2 registry parameter whose settings represent predefined groups of other DB2 registry parameters, which are precompiled to optimize performance of certain FNP8 components and processes.

To set the `DB2_WORKLOAD` workload type, this command was used:

```
# db2set DB2_WORKLOAD=FILENET_CM
```

After setting the `DB2_WORKLOAD` variable to `FILENET_CM`, a `DB2_WORKLOAD` in square brackets behind a registry variable then indicates that a variable was set by `DB2_WORKLOAD`:

```
# db2set
DB2_WORKLOAD=FILENET_CM
DB2_SKIPINSERTED=YES [DB2_WORKLOAD]
DB2_OPTPROFILE=YES [DB2_WORKLOAD]
DB2_USE_ALTERNATE_PAGE_CLEANING=YES [DB2_WORKLOAD]
DB2_MINIMIZE_LISTPREFETCH=YES [DB2_WORKLOAD]
DB2_EVALUNCOMMITTED=YES [DB2_WORKLOAD]
DB2_PINNED_BP=YES
DB2COMM=TCPIP
DB2AUTOSTART=YES
```

With these parameters set, unnecessary locking is avoided, which can improve performance. In some cases, this can also prevent deadlock or query timeout errors.

DB2 database configuration

In this topic, a check is made that database parameter CUR_COMMIT has been set.

The CUR_COMMIT parameter controls the behavior of cursor stability scans and works in conjunction with the FILENET_CM setting of the DB2_WORKLOAD registry parameter.

To check that the database parameter CUR_COMMIT is enabled:

1. A connection to the database was made.
2. The database configuration was obtained:


```
> db2 get db cfg
...
Currently Committed                (CUR_COMMIT) = ON
CHAR output with DECIMAL input      (DEC_TO_CHAR_FMT) = NEW
Enable XML Character operations      (ENABLE_XMLCHAR) = YES
WLM Collection Interval (minutes)    (WLM_COLLECT_INT) = 0
...
```
3. If required, the CUR_COMMIT parameter was enabled.

For information about tuning IBM FileNet P8 databases, proceed to this URL at the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1

Then select: **Performance tuning** and proceed to the Web page that describes how to improve the performance of IBM FileNet P8 components that use an IBM DB2 database.

Database indexing

The database indexing that was performed for the *IBM Content Navigator* and *IBM FileNet P8* is described here.

IBM Content Navigator

IBM Content Navigator (ICN) stores configuration and preferences data in a database. Enabling ICN features can create add-on database tables and columns that are not automatically indexed. After creating indexes for those, ICN queries for properties run faster and return results faster than queries on tables or columns that are not indexed.

For information about tuning Content Navigator database indexes, proceed to this URL at the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSEUEX_2.0.3

Then select: **Installing IBM Content Navigator** and proceed to the Web page that describes how to create database indexes to improve performance.

IBM FileNet P8

FNPP8 CPE stores configuration and preferences data in a global configuration database (GCD).

Documents, workflows and other objects data are stored in the Object Store database.

Indexing FNPP8 databases can improve the performance. Consider creating database indexes when the performance for transactions is not as desired. Database indexes can greatly improve the performance for transactions that run against FNPP8 databases.

For information about indexing FileNet P8 databases, proceed to this URL at the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1

Then select: **Performance Tuning** and proceed to the Web page that describes how to index IBM FileNet P8 databases.

Example of database indexing

This topic provides a database index suggestion for ICN performance tuning. It was taken from the ICN documentation and did help to improve the response time for a long-running transaction in the ICN performance workload case study.

The ICN product documentation (topic “Creating database indexes to improve performance”) makes this recommendation:

"For a large IBM FileNet P8 repository, that is slow or times out too quickly when opening the all searches view, create a database index on the table named Docversion. The index is for the database table Docversion on the *_Searchtype column."

Here is an example of how the above tuning recommendation was implemented:

1. This DB2 searchtype command was used:

```
> db2 "describe table docversion" | grep -i searchtype
UBE06_SEARCHTYPE          SYSIBM      INTEGER          4          0 Yes
```
2. This CREATE INDEX command was used:

```
-> CREATE INDEX DOC_SEARCH_TYPE_IDX ON DOCVERSION (UBE06_SEARCHTYPE)
```

After creating a database index related to the SQL query for the transaction in question (as shown in the above example), the response time decreased from 7000 to 200 milliseconds. This represented an improvement by a factor of 35. It also showed how database indexes can boost the performance for SQL queries that without an index, are expensive in terms of response time.

Long-running or CPU-expensive transactions can go back to a missing database index. It is worth taking time to analyze the SQL statements that belong to a transaction.

For example, you can use the db2top utility to identify long-running or CPU-expensive SQL statements.

In this case study, a full table scan was done when the problem transaction was executed. The solution was to create an additional database index on the FNP8 Object Store table Docversion.

Figure 4 shows the CPU loads on the ECM database server with and without the additional database index applied.

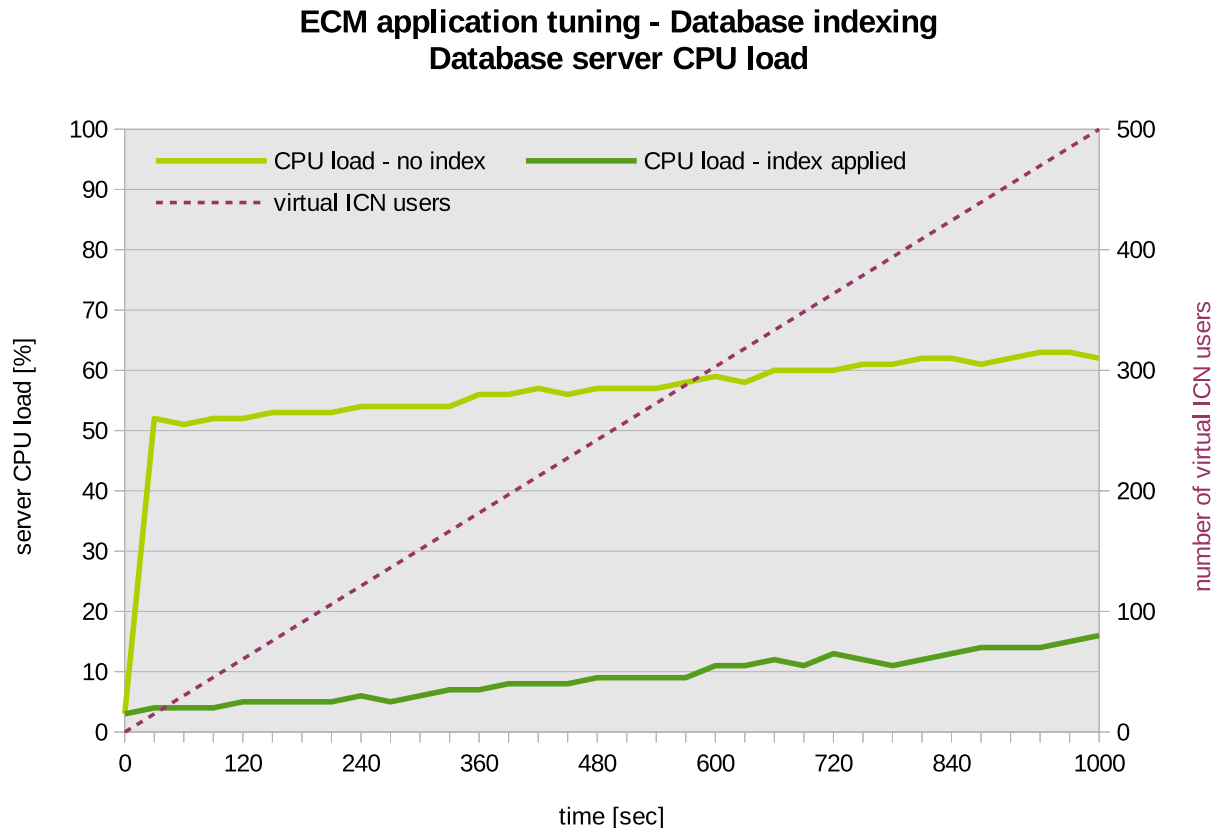


Figure 4. CPU loads on ECM database server with and without the additional database index applied

The ICN transaction with the high response time is executed during the virtual user rampup phase of the performance test. The virtual users are continuously login into ICN and start to execute a transaction mix. Every new virtual user executes the problem transaction onetime. Without the additional database index applied the CPU load on the database server is quickly at 50 to 60 % for 8 CPUs.

Repeating the same test run with the additional index applied the CPU load is only at 10-20%. The additional index does not only help to improve the transaction response time but also helps to save useful CPU capacity on the database server.

Results of Enterprise Content Management tuning

The results of the actions to tune the Enterprise Content Management (ECM) applications are now described.

Figure 5 shows the performance improvement after the ECM application tunings, described in the previous topics, are applied to the ECM Gateway and ECM application tier compared to the default settings.

The improvement for database indexing previously discussed does not count in this context, because it is not affecting the stable load phase from which the metric average response time is taken.

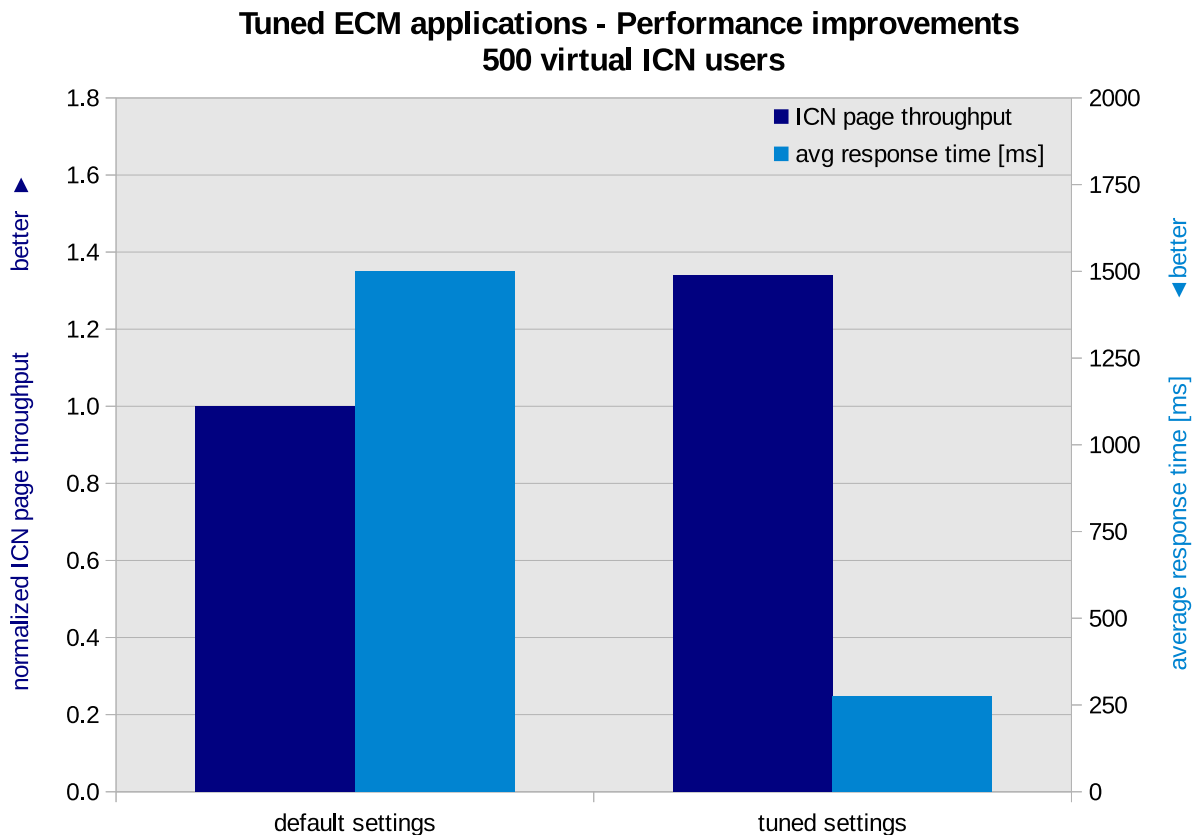


Figure 5. Summarized results of ECM application tuning

The transaction throughput rates were normalized against throughput number scored with no ECM application tuning at all. So the default settings normalized throughput rate was 1.0.

Figure 5 also shows the average response time in milliseconds for the transaction mix at a load level of 500 parallel virtual ICN users.

The throughput rate for the tuned environment *increased by more than 30%*, and the average response time *reduced from 1500 milliseconds seconds to 275 milliseconds* (a factor of 5.45).

The transaction throughput and response time improvements that resulted from this basic ECM application tuning *were impressive*. Most of the tuning settings were WAS parameters, which indicates that some relevant WAS parameters and their default values are only good enough for simple applications. More complex

applications require a tuning of these WAS parameters to allow for maximum application performance and maximum utilization of the available server resources.

Chapter 6. Enterprise Content Management workload

The workload for this case study was based on the *IBM Rational Performance Tester* (RPT) application. RPT is a product for automating the load and scalability testing of server-based software applications.

RPT can be used to prove that an application or an entire “system under test” (SUT) will scale and perform to meet defined load levels. The machine (x86_64) on which RPT ran, acted as an external workload generator for the SUT implemented in the z Systems server.

The users emulated by RPT are *virtual users*. Because of the short think times in the RPT scripts that were used, each virtual user generated the load-equivalent of many human end-users who would use ICN to accomplish real work tasks. Therefore, 1000 virtual ICN users probably represented a work mix equivalent to many thousands of real end users.

The workload was a typical mix of IBM Content Navigator (ICN) tasks triggered by virtual users on the ICN user interface, such as:

- add and remove documents
- view or download documents
- view document properties
- search for documents
- convert and download documents

Using RPT, it was possible to record user actions on graphical user interfaces (GUIs) and replay them at a later date. The workload consisted of user actions that followed a defined activity flow.

During the rampup phase, virtual users logged in to ICN one-by-one, with a small delay. Instantly after login, the virtual users started to execute a few one-time transactions.

Once the one-time transactions were completed, the users entered a loop and randomly executed transactions from a pool of transactions with a well-defined distribution of transaction types.

The stable workload phase was reached when all virtual users were logged in and were executing their transaction loop. Between the transaction execution was a thinktime, where the virtual user was idling for a few seconds before starting the next transaction. The metrics transaction throughput and average response time were taken from this stable workload phase.

After a certain time period, the transaction loop ended and the virtual users started to log out from ICN. This was the rampdown phase of the workload, lasting until all virtual users were logged out.

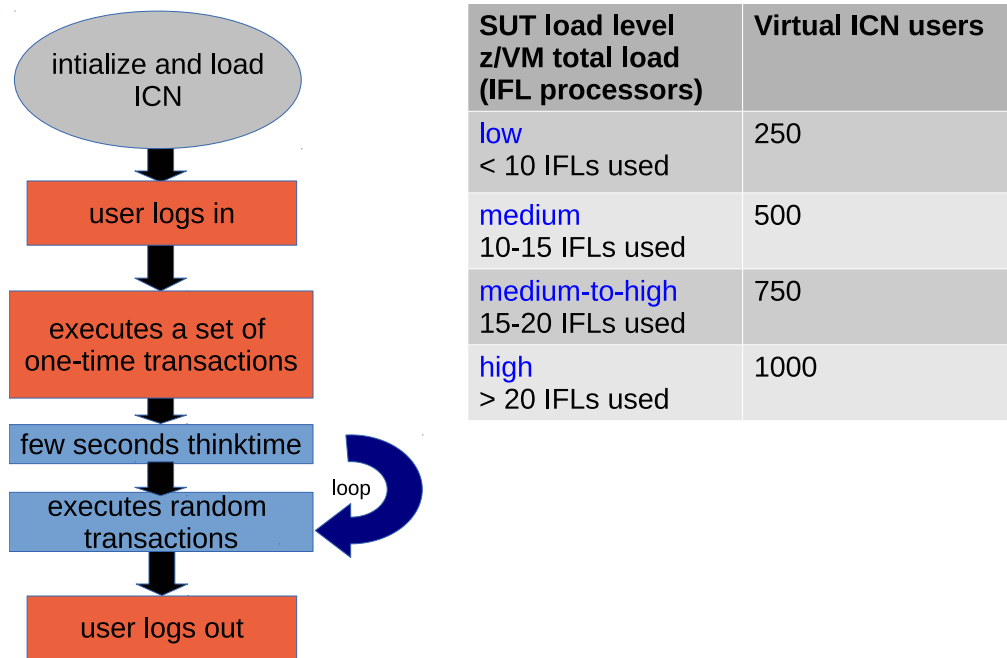


Figure 6. Virtual user activity flow and workload levels

The more virtual users were logged in parallel during the stable workload phase. The smaller the thinktime, the higher was the workload level for the SUT.

The z/VM hypervisor had 30 IFL processors in total and for the case study four load levels were defined. Starting from low to medium, medium-to-high and high (shown in Figure 6). These load levels were used throughout the scale-out study.

FileNet Advanced File Storage Area

FileNet repositories can either be a *database Object Store* or a *filesystem File Storage Area*. For this case study, an Advanced File Storage Area was used.

If the FileNet repository is a File Storage Area, FileNet stores content in a filesystem location. The File Storage Area filesystem type is XFS at first and IBM Spectrum Scale next.

In this case study, the focus for FileNet content data is the File Storage Area. Therefore, most of the content population is done to this FileNet repository. The File Storage Area is shared among the clustered ECM nodes in the scale-out part.

Table 14 shows the populated data for the File Storage Area and the database Object Store. The Storage Area is 600 GiB in size and around 300 GiB are populated with content.

Table 14. Initial content population for the FileNet repositories

Object Content	Count	Total Size	Location
1 KiB text documents	3,000,000	2.9 GiB	File Storage Area
10 KiB text documents	1,500,000	14.6 GiB	File Storage Area
100 KiB XLS documents	450,000	43.9 GiB	File Storage Area
500 KiB PPT documents	150,000	73.2 GiB	File Storage Area

Table 14. Initial content population for the FileNet repositories (continued)

Object Content	Count	Total Size	Location
2 MiB DOC documents	15,000	29.3 GiB	File Storage Area
5 MiB PDF documents 1	2,000	9.8 GiB	File Storage Area
10 MiB JPG documents	1,500	14.7 GiB	File Storage Area
9 MiB DOC documents	4,000	35.1 GiB	File Storage Area
5 MiB PDF documents 2	4,000	19.5 GiB	File Storage Area
6 MiB PPT documents	4,000	23.4 GiB	File Storage Area
7 MiB XLS documents	4,000	27.3 GiB	File Storage Area
Workflows	180,000		Database Object Store
Total documents		~ 300GiB	

The content population as shown in the table is the initial starting point. When the ECM performance workload is running users continuously add several GigaByte of additional content data to the repositories depending on the workload runtime and number of users.

Advanced File Storage Area on an XFS filesystem

During the first part of the single ECM node study, the high-performance filesystem XFS is used for the Advanced FileNet File Storage Area. XFS is a good choice for large files or large file systems.

In addition, the Linux Logical Volume Manager (LLVM) is used to manage a Logical Volume (LV).

The XFS filesystem is created on a striped LV consisting of 6 x 100 GiB FCP/SCSI disks.

The following example shows how the XFS File Storage Area is populated with the data that is shown in Table 14 on page 34.

```
# df -hT
Filesystem                Type      Size  Used Avail Use% Mounted on
/dev/dasda1                ext3      20G   14G  4.6G   76% /
...
/dev/mapper/FSDData_vg-FSDData_lv xfs      600G   310G  291G   52% /data/FSDData
```

Advanced File Storage Area on Spectrum Scale

IBM Spectrum Scale is a scalable, high-performance data and file management solution that is based upon *General Parallel File System* (GPFS).

For the single ECM node study, XFS was replaced by the IBM Spectrum Scale filesystem at a later stage. This enabled a comparison to be made between the capabilities of IBM Spectrum Scale and a non-clustered high-performance filesystem.

The following example shows how the Spectrum Scale File Storage Area is populated with the data that is shown in Table 14 on page 34.

```
# df -hT
Filesystem                                Type  Size  Used Avail Use% Mounted on
/dev/dasda1                              ext3   20G   15G   4.2G   78% /
...
/dev/FSData                              gpfs   600G  337G  264G   57% /data/FSData
```

Note: The ECM node runs FNP8 CPE and the ICN in the same Linux node. Therefore, this node is the only node in the Spectrum Scale cluster for this part of the study. The follow-on Part 2 white paper addresses the *multi-node* Spectrum Scale cluster scale-out, either as ‘Shared Disk’ or ‘Network Shared Disk’ model.

This example shows a Spectrum Scale cluster with single ECM node:

```
# mm1scluster

GPFS cluster information
=====
GPFS cluster name:      ECM_1_node
GPFS cluster id:       714383681xxxxxxxxx
GPFS UID domain:       ECM_1_node
Remote shell command:  /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:       CCR

Node  Daemon node name  IP address      Admin node name  Designation
-----
1     ECMnod1             xx.xxx.xx.xxx   ECMnod1          quorum-manager
```

This example shows six 100 GiB FCP/SCSI disks for the File Storage Area that are directly attached to the ECM node:

```
# mm1snsd

File system    Disk name      NSD servers
-----
FSData        nsd14cb        (directly attached)
FSData        nsd15cb        (directly attached)
FSData        nsd14cc        (directly attached)
FSData        nsd15cc        (directly attached)
FSData        nsd14cd        (directly attached)
FSData        nsd15cd        (directly attached)
```

Chapter 7. IBM Spectrum Scale tuning parameters

This topic introduces various IBM Spectrum Scale parameters that are commonly used for the performance tuning of ECM workloads.

Linux system clock synchronization

The Linux system clocks of all nodes in a Spectrum Scale cluster must be synchronized. If this is not the case, Spectrum Scale file system operations might be disrupted.

For the single-node Spectrum Scale cluster, this is not relevant. However, it becomes more important when extending to a multi-node cluster.

For all Linux virtual machines in the SUT, the Network Time Protocol (NTP) is enabled for clock synchronization.

IBM Spectrum Scale cache usage

Spectrum Scale maintains a number of cache segments on each node in the cluster.

The behavior of cache usage is controlled by the following three Spectrum Scale 4.2 parameters:

- `maxFilesToCache` (default 4000)
- `pagepool` size (default 1 GiB)
- `maxStatCache` (default 1000)

These parameter defaults are set at cluster creation time, but can be changed with the Spectrum Scale `mmchconfig` command.

Note: Spectrum Scale does not use the Linux page cache!

Setting the `maxFilesToCache` parameter

The `maxFilesToCache` parameter is the maximum total number of different files that can be cached at one time in the pagepool.

Every entry in the pagepool requires some pageable memory to hold the content of the file's inode plus control data structures. For systems with applications that handle a large number of files like ECM systems, increasing the value for `maxFilesToCache` (`mFTC`) may prove beneficial. This is especially true when the system accesses a large number of small files.

The value should be large enough to handle the number of concurrently open files plus allow caching of recently used files.

Figure 7 on page 38 shows the Spectrum Scale `startIO` metric for different `mFTC` values. The `startIO` metric was taken from the statistics for the Spectrum Scale *virtual file system* (VFS). It summarizes the Spectrum Scale I/O requests that trigger an I/O operation for the Linux operating system in a specific time interval.

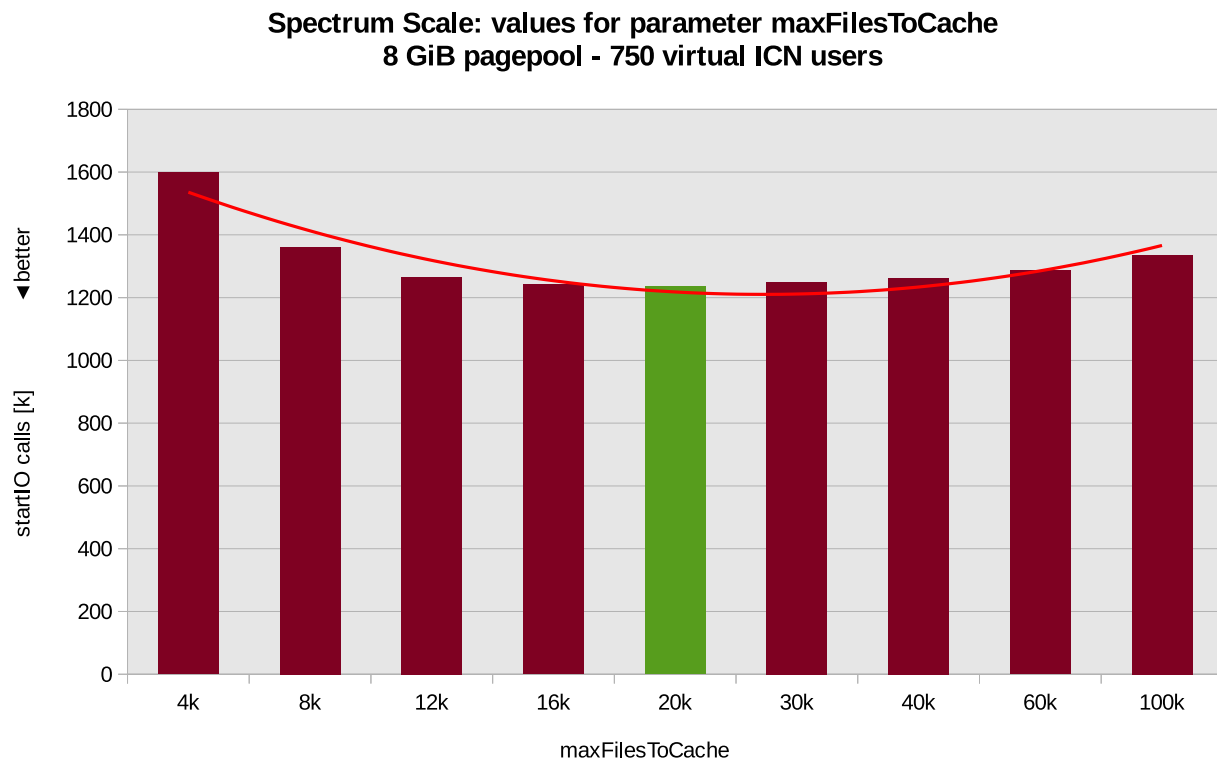


Figure 7. Spectrum Scale maxFilesToCache

The measurement series was done for an 8 GiB pagepool and a medium-to-high workload level (750 virtual ICN users).

The default mFTC value of 4k (4000) showed the highest startIO number. This indicates that the pagepool was not optimally utilized for the workload. For the measurement series, mFTC was increased step-by-step up to a value of 100k (100.000) files.

The number of 20k (20.000) for mFTC has the lowest startIO number and seems to utilize the pagepool best. Note that the range of mFTC values is broad with a similar caching behaviour, but once the value is set to high, the startIO calls indicate a degrading cache-hit ratio.

Note: mFTC is set to 20k for *all* subsequent measurement series of this case study.

This example shows how mFTC is set to 20.000:

```
# mmlsconfig maxFilesToCache
maxFilesToCache 4000
# mmchconfig maxFilesToCache=20000
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
              affected nodes. This is an asynchronous process.

# mmlsconfig maxFilesToCache
maxFilesToCache 20000

# mmdiag --config | grep maxFilesToCache
! maxFilesToCache 4000
```

Note:

- `mmfsconfig` only shows the planned value. Use `mmdiag` to display the currently-active value. It can be different from the changed value, because a GPFS daemon restart is required in order to become the planned value active. `mFTC` cannot be changed immediately (via the `-i` or `-I` options).
- Setting `mFTC` to a high value on purpose is not recommended. If the value is too high, the pagepool can become ineffective.

Setting the pagepool size

The Spectrum Scale pagepool is used to cache user data and file system metadata.

In general, Spectrum Scale read and write requests going through the pagepool are executed asynchronously. Hence the size of the pagepool limits the amount of data or metadata that can be cached without requiring synchronous I/O.

The amount of memory available for a pagepool on a particular node may be restricted by the operating system, hypervisor and other software running on the node. The pagepool sizes on different cluster nodes must not be equal. Every node in a cluster can have its own pagepool size according to the intended role or task of the node.

A pagepool size of 8 GiB was chosen for the single-node cluster to allow a similar comparison to the XFS filesystem which had around 8 GiB Linux page cache available.

This example shows how the pagepool size was increased from 1GiB (default) to 8 GiB:

```
# mmfsconfig pagepool
pagepool 1G
# mmchconfig pagepool=8G
mmchconfig: Command successfully completed
# mmfsconfig pagepool
pagepool 8G
```

Note:

- To change the pagepool size immediately (without restarting the GPFS daemons), use the `mmchconfig -i` or `-I` option.
- If no node is specified for the `mmchconfig` command, the new pagepool size applies to all nodes in the cluster.

Figure 8 on page 40 shows the Spectrum Scale `startIO` calls for 750 virtual ICN users (medium-to-high workload). The larger the pagepool size, the less `startIO` calls are required. For a 1 GiB pagepool, around 2000k (2.000.000) `startIO` calls were done during the workload execution phase. For the 8 GiB pagepool, this number decreased to 1200k (1.200.000) `startIO` calls. As a result the overhead for the operating system disk I/O was lower.

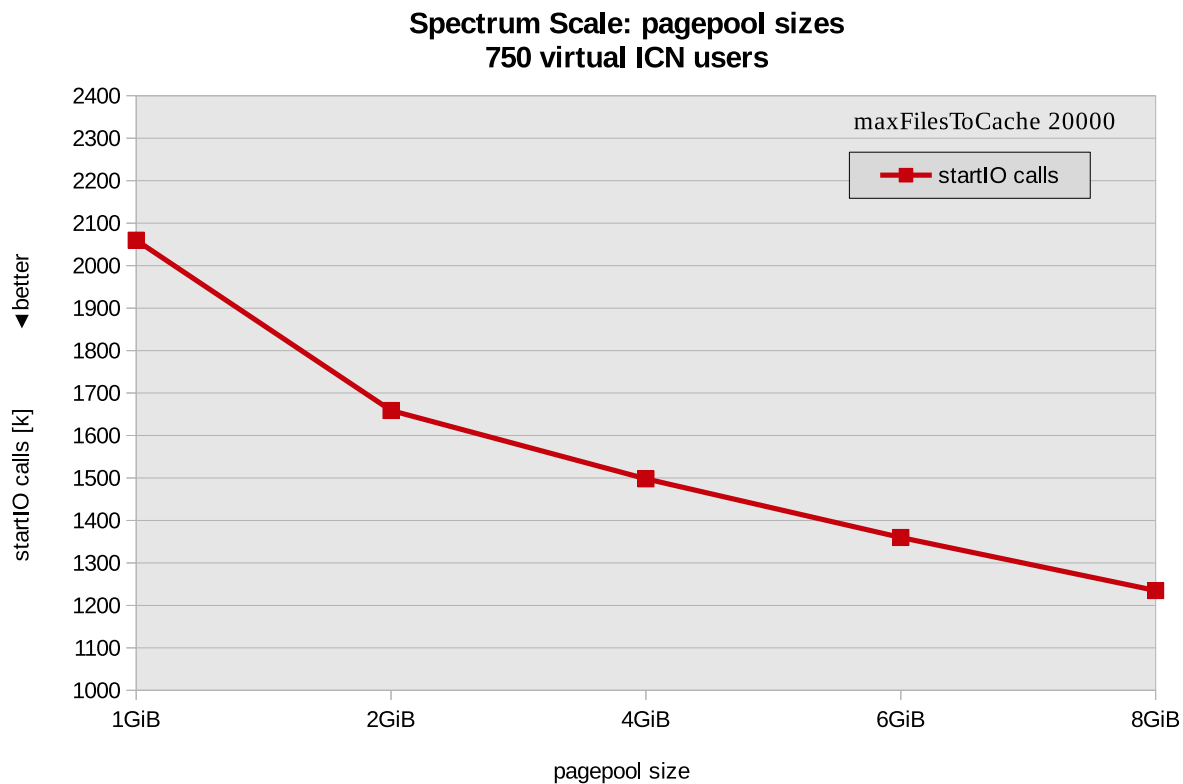


Figure 8. Spectrum Scale startIO calls for different pagepool sizes

Relation between pagepool size and level of maxFilesToCache

Both the pagepool size and maxFilesToCache influence the cache usage. This indicates that a proper value for maxFilesToCache (mFTC) that fits to the pagepool size is important.

In addition, consider the number of concurrently open files for a workload when sizing mFTC. Otherwise, the pagepool might not be optimally utilized.

Figure 9 on page 41 shows some unfavourable effects when scaling the values for mFTC.

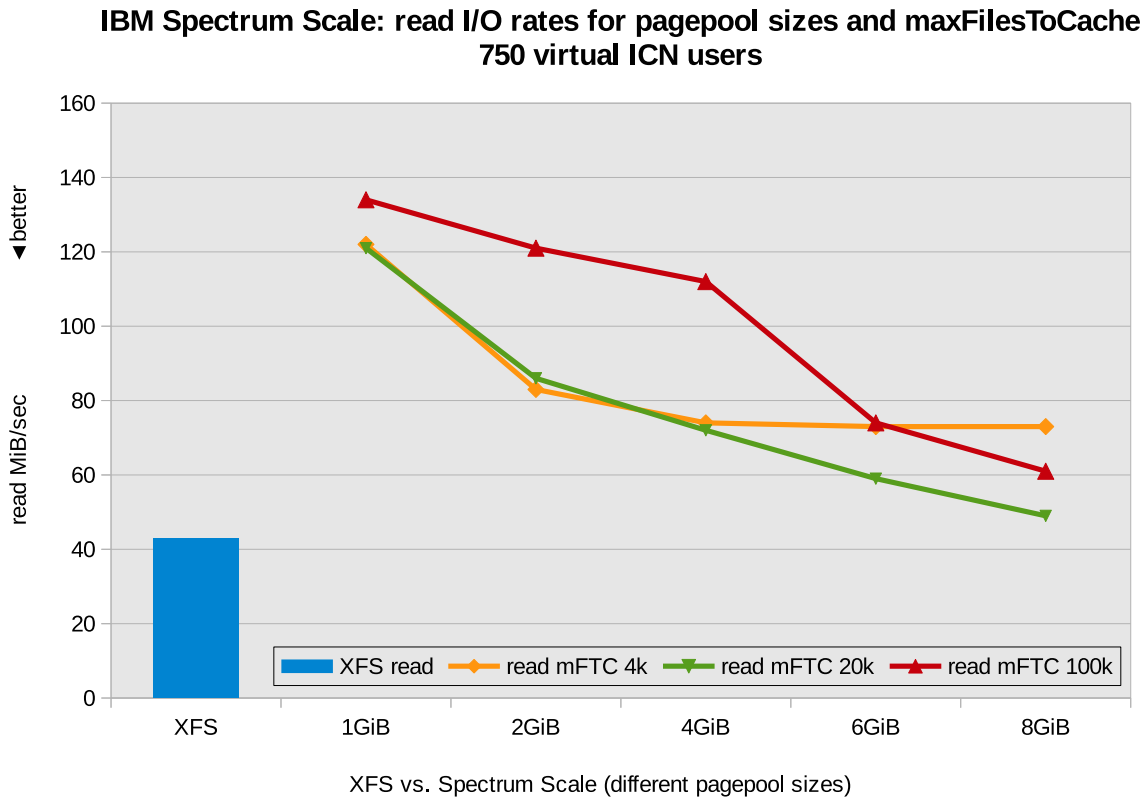


Figure 9. XFS versus Spectrum Scale (different pagepool sizes and mFTC values)

Again, the medium-to-high workload with 750 virtual ICN users is used.

- The blue bar shows the read I/O rate for XFS (XFS has around 8 GiB Linux page cache available).
- For Spectrum Scale there are three measurement series done scaling the pagepool sizes and mFTC values. The pagepool size starts at 1 GiB (default) and is increased up to 8 GiB in 2GiB steps. For each pagepool size three runs with different mFTC values (4k, 20k and 100k) are done represented by a colored line for each mFTC value.

Two Unfavourable Scenarios That Might Occur

Scenario 1: maxFilesToCache is set too high for a pagepool size.

- For example, this is the case for pagepool sizes ≤ 4 GiB and mFTC 100k (red line). The pagepool holds too many metadata and cached files for its size to work efficiently. It becomes ineffective, which results in high read I/O rates (> 100 MiB/sec for this example).
- However, the larger the pagepool becomes then the better fits a large mFTC value. For the 8 GiB pagepool in this measurement series mFTC 100k fits better and better.

Scenario 2: maxFilesToCache is set too small for a pagepool size.

- For example, this happens for pagepool sizes > 4 GiB and mFTC 4k (yellow line). Larger pagepool sizes have no effect any more because mFTC limits the number of files that can be cached in the pagepool. As a result, the read I/O rate remains at a certain level even when enlarging the pagepool.

The value of 20k for mFTC fits well for the workload and pagepool size of 8 GiB (green line). Compared to XFS the read I/O rate is at a similar level for this combination (41 vs. 49 MiB/sec).

Setting the value of maxStatCache

Basically the maxStatCache parameter is not effective on the Linux platform. Therefore one can set the maxStatCache to a smaller value, such as 512.

This example shows how maxStatCache can be defined:

```
# mm1sconfig maxStatCache
maxStatCache 1000
# mmchconfig maxStatCache=512
mmchconfig: Command successfully completed
# mm1sconfig maxStatCache
maxStatCache 512
```

Memory consideration for IBM Spectrum Scale cache usage

This topic describes how the Spectrum Scale metadata for open files will increase in relation to the maxFilesToCache value.

In Figure 10, a value of 100k for maxFilesToCache already needs around 230 MiB memory for metadata compared to 45MiB for 20k.

Spectrum Scale: metadata used for open files related to maxFilesToCache

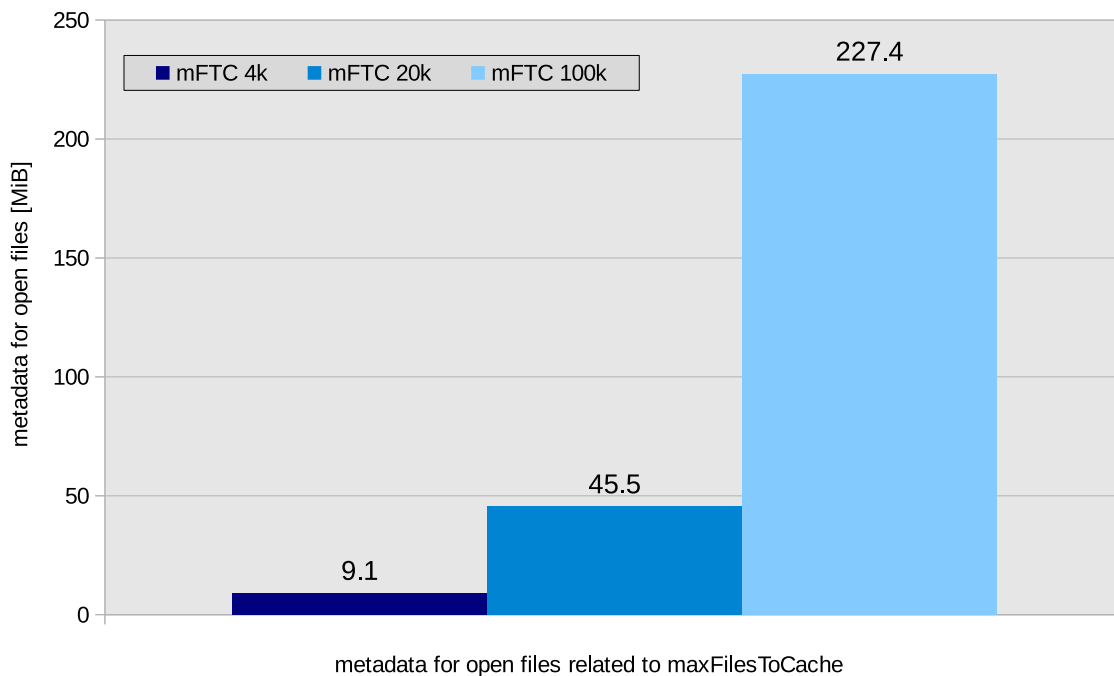


Figure 10. Spectrum Scale maxFilesToCache metadata for open files

The open file metadata size can be checked using the **mmdiag -stats** command.

This example shows how to query the open file count metadata for mFTC, which has been set to 100k:

```
# mmdiag --stats
=== mmdiag: stats ===
Global resources:
OpenFile counts: total created 104795 (in use 100000, free 4795)
  using 232812K memory
  cached 100000, currently open 2+3, cache limit 100000 (min 10, max 100000),
  eff limit 100000
stats: ins 20869218 rem 20769218 creat 1548983 destr 1444188 free 10496248
reuse 10491453 steals 10390822 (clean 9391515, dirty 999307)
```

A rule of thumb: The total amount of memory Spectrum Scale uses to cache file data and metadata is estimated by adding together:

1. The pagepool size.
2. The amount of memory required to hold inodes and control data structures ($\text{maxFilesToCache} \times 3 \text{ KiB}$).
3. The memory for the stat cache ($\text{maxStatCache} \times 400^{\text{®}}$ bytes).

IBM Spectrum Scale filesystem parameters

The IBM Spectrum Scale filesystem parameters can be tuned using values for block size and the number of inodes.

Setting the value of blocksize

The default file system blocksize (256KiB) should be sufficient for most applications, although a larger blocksize can be considered if large files are being stored on the file system.

For the ECM scale-out study described in this white paper series, smaller and larger block sizes than the default blocksize have been tested without any influence on the reported transaction response times.

The default file system blocksize was used throughout the case study.

Setting a value for the number of inodes

ECM systems often store a very large number of files in a File Storage Area.

The system under test had more than 5 million files in the File Storage Area and required a large number of filesystem inodes to manage the files. Spectrum Scale has a default inode limit that can become a threshold for the filesystem and lead to filesystem full conditions. There are a couple of indications that point to an “inode full” condition (for example, **df** command, **mmdf** command, GPFS and Linux log files).

This example shows an error indication on a **tar** restore operation:

```
...
tar: content/79: Cannot mkdir: No space left on device
```

This example shows how the **df**, **mmfsfs**, and **mmdf** commands can be used:

```
# df -h
Filesystem                Size  Used Avail Use% Mounted on
/dev/FSData                600G   45G   556G   8% /data/FSData

# df -i
Filesystem                Inodes   IUsed   IFree  IUse% Mounted on
/dev/FSData                614464  614464      0   100% /data/FSData

# mmfsfs FSData --inode-limit
flag              value              description
-----
--inode-limit    614464              Maximum number of inodes

# mmdf FSData -F
Inode Information
-----
Number of used inodes:      614464
Number of free inodes:      0
Number of allocated inodes: 614464
Maximum number of inodes:   614464
```

This example shows a GPFS log entry:

```
Tue Dec  8 10:14:57.454 2015: [W] Inode space 0 in file system FSData is
approaching the limit for the maximum number of inodes.
```

This example shows a Linux log entry:

```
Dec  8 10:14:57 1e2ep223 mmfs: Error=MMFS_SYSTEM_WARNING, ID=0x4DC797C6,
Tag=6725521: File system warning. Volume FSData. Reason: [W] Inode space 0 in
file system FSData is approaching the limit for the maximum number of inodes.
```

Although the file system seems to have plenty of space left (**df** command reports only 8% usage), the inode information reveals that the file system runs out of inodes.

The solution for the problem is straightforward - simply increase the inode limit for the Spectrum Scale filesystem. There is no need to recreate the file system.

This example shows how to increase the inode limit:

```
# mmchfs FSData --inode-limit 15M
Set maxInodes for inode space 0 to 15728640
Fileset root changed.

# mmdf FSData -F
Inode Information
-----
Number of used inodes:      4039
Number of free inodes:      610425
Number of allocated inodes: 614464
Maximum number of inodes:   15728640
```

For information about how to calculate a meaningful inode limit in the Spectrum Scale **mmchfs** command, proceed to this URL at the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_4.2.0

Then search for the **mmchfs** command.

Chapter 8. Single Enterprise Content Management (ECM) node results

For the measurement series discussed in the results section, all setup and tuning modifications described were applied to the “system under test” (SUT).

The workload as described in Chapter 6, “Enterprise Content Management workload,” on page 33 with the different SUT load levels was used for this scale-out case study. For the measurement series, the virtual ICN users were scaled from 250 to 1000 users, either:

- With the FileNet File Storage Area on an XFS file system.
- On a Spectrum Scale file system.

One of the major objectives for this study was to analyze IBM Spectrum Scale as a file system for the FileNet File Storage Area for Linux on z Systems. This first part of the white paper established the SUT, verified various tuning settings, and compared it with a high performance file system such as XFS.

The second white paper in this series will discuss the usage of the IBM Spectrum Scale parallel file system for a multiple node ECM cluster.

The SUT load levels were defined as follows for the scale-out study. Usually, the ICN transaction throughput is at a fixed rate which depends upon the number of virtual ICN users and the thinktime between successive transactions.

Note: The average response time for all transactions was considered as an indication of performance.

The ratio of CPU overcommitment (30 IFL processors vs. 46 virtual CPUs) was 1:1.5. Note that the SUT z/VM hypervisor had 30 IFL processors in total.

Table 15. Scale-out case study SUT load level definitions

SUT load level	Number of virtual ICN users	z/VM total CPU load % (3000% = 30 IFLs)
low	250	0-1000 (< 10 IFLs)
medium	500	1000-1500 (10 - 15 IFLs)
medium-to-high	750	1500-2000% (15 - 20 IFLs)
high	1000	2000-3000% (> 20 IFLs) - see Note below

Note: The highest z/VM total CPU load level was 2300% (23 IFL processors used) for the single ECM node case study with 1000 virtual ICN users.

The two most utilized Linux guests were:

- The single ECM node (ICN and FNP8 CPE) with 1960% (19.6 IFL processors).
- The ECM database server with 230% (2.3 IFL processors) at the high SUT load level.

Enterprise Content Management user scaling for XFS

For this measurement series, the FileNet File Storage Area was on an XFS filesystem.

The XFS filesystem was created on a Linux Logical Volume (LV) extending across 6x 100 GiB FCP/SCSI disks.

The ECM node ran the ICN and FNP8 CPE applications, where each application was in its own single node WAS Cluster.

For a graphical display of this aspect of the SUT, see Figure 2 on page 10.

Transaction throughput and z/VM total CPU load

This topic provides information about the normalized ICN page throughput rates at different SUT load levels.

Figure 11 shows the normalized ICN page throughput rates at different SUT load levels. The throughput rate increased as the number of virtual ICN users increased.

The z/VM total CPU load is an indicator for the utilization of the SUT, where the top CPU consumers are the single ECM node and the ECM database node.

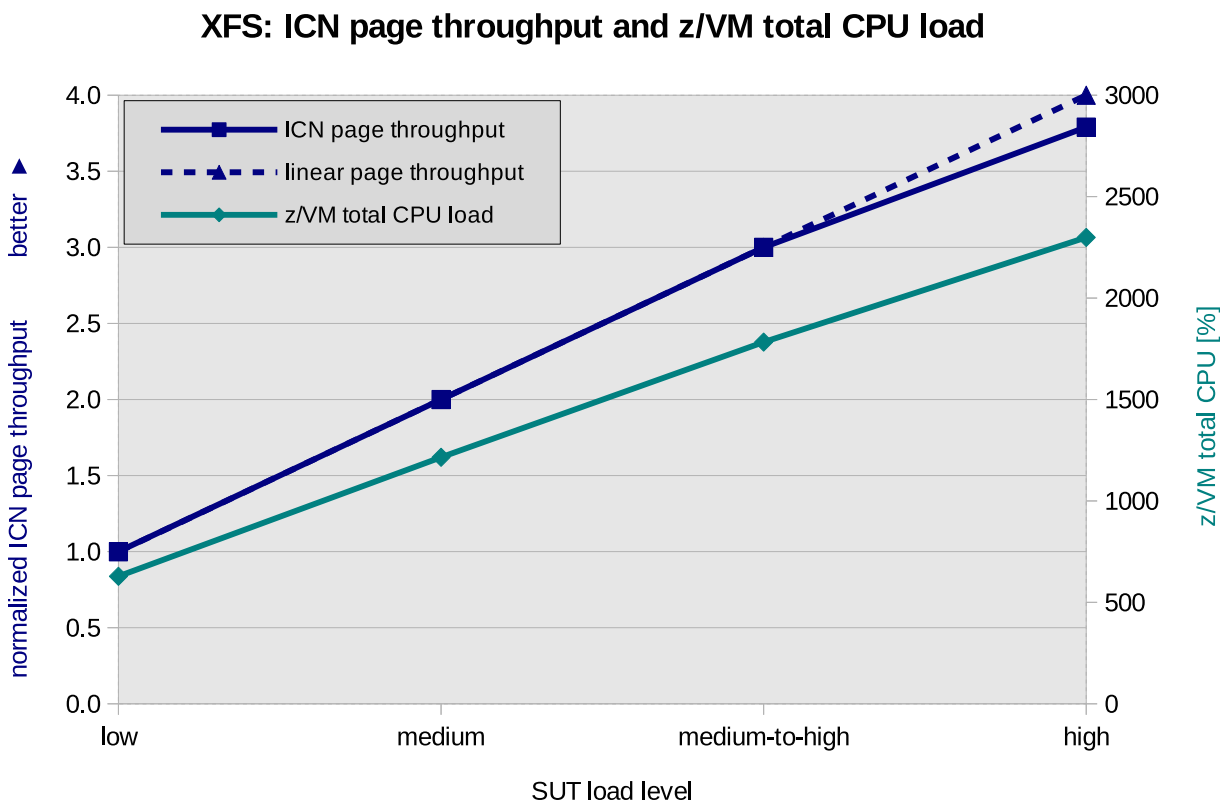


Figure 11. Normalized ICN page throughput and z/VM total CPU load

Observations

The throughput rate shows almost linear characteristics correspondent to the increasing number of virtual ICN users beginning from a low load level (250 users) to a high load level (1000 users).

For the low load level the z/VM total CPU load shows around 600% (6 IFL processors used) and for the high load level around 2300% (23 IFL processors used).

Conclusion

Over the entire load level bandwidth, the ECM system showed stable page throughput rates almost following an ideal linear trend-line (throughput line vs. dashed linear throughput line). The *transaction throughput quadrupled* at the highest load level compared to the lowest workload level (4x times the number of virtual ICN users).

The z/VM total CPU load also showed linear trend equivalent to the adjusted workload level.

In summary, the SUT showed *an excellent scaling behavior for the entire ECM system towards high CPU loads and high user numbers!*

Average ICN transaction response times

This topic describes the average ICN transaction response times for different load levels.

Figure 13 on page 49 shows the average ICN transaction response times for different load levels.

- The average response time was calculated over the complete set of executed ICN transactions during the stable workload phase (rampup and rampdown phase excluded).
- There were short running transactions with response times from 100 - 600 ms, as well as a few longer running transaction with response times from 1000-2000ms for the 1000 virtual ICN user load.
- In the following description, only the *average response time* is considered.

XFS: ECM user scaling - average ICN transaction response times

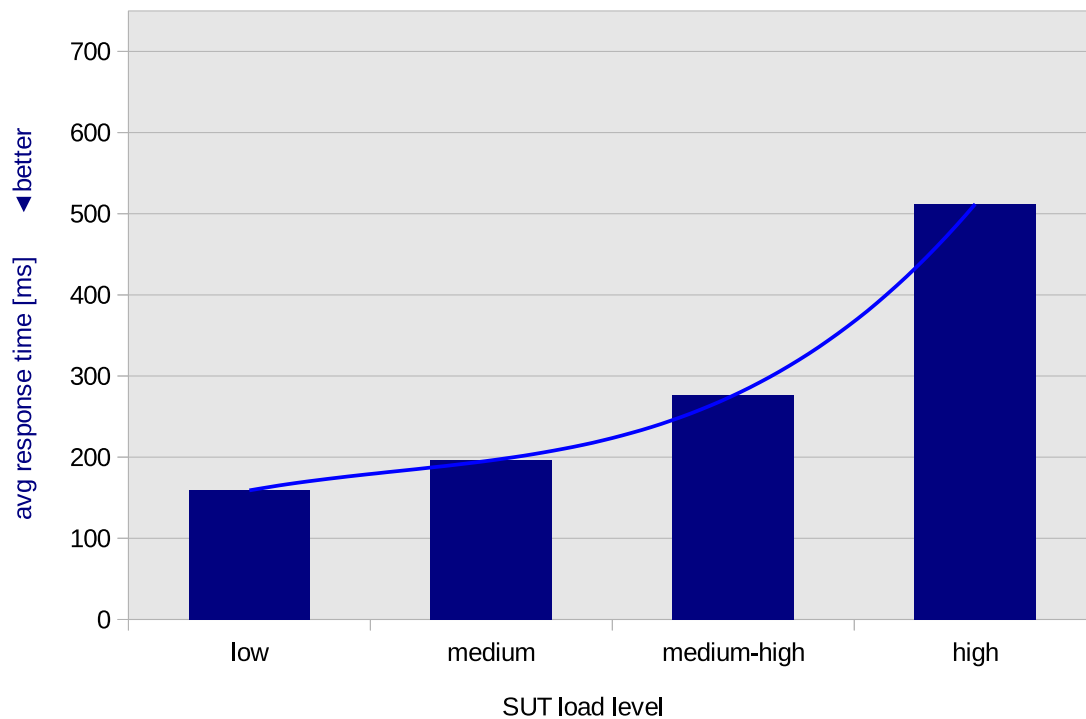


Figure 12. Average ICN transaction response times

Observations

The average ICN transaction response time started at 150 ms for a low load level and increased up to 500 ms for a high load level in a non-linear manner.

Conclusion

For this particular ECM workload, an average ICN response time below 1 second was expected and good. *The SUT showed overall good response times, even with a heavily increase of load at the highest load level!*

ECM user scaling for IBM Spectrum Scale single node cluster

For this measurement series, the FileNet File Storage Area was on a Spectrum Scale filesystem. Spectrum Scale also had 6x 100 GiB FCP/SCSI disks directly attached to the ECM node in Shared Disk (SD) cluster configuration.

The described Spectrum Scale tuning was applied to the single node cluster. Apart from that, the SUT looks the same as for the XFS measurement series. The ECM node ran the ICN and FNP8 CPE applications each application in their own single node WAS Clusters.

For a graphical display of this aspect of the SUT, see Figure 3 on page 11.

Average ICN transaction response times

This topic describes the average ICN transaction response times for XFS and Spectrum Scale at the different load levels.

Figure 13 shows the average ICN transaction response times side-by-side for XFS and Spectrum Scale at the different load levels.

- The average ICN response time was calculated over the complete set of executed ICN transactions during the stable workload phase (rampup and rampdown phase excluded).
- The throughput rate increased as the number of virtual ICN users increased.

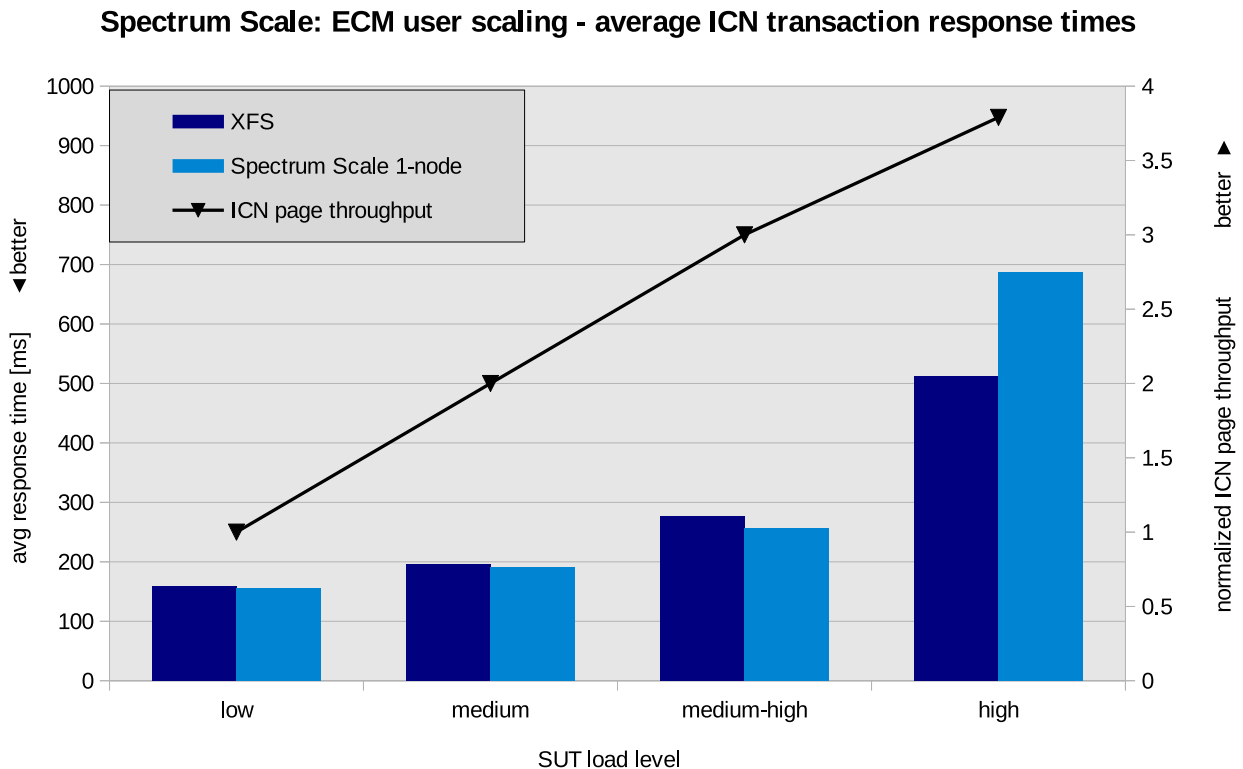


Figure 13. Average ICN transaction response times for XFS and Spectrum Scale

Observations

The average ICN transaction response time started at 150 ms for the low load level and increased up to 700 ms for the highest load level with Spectrum Scale. Up to the medium-high load level showed Spectrum Scale a slightly better response times than XFS. For the highest load level, Spectrum Scale showed a higher average response time than XFS.

Conclusion

For this particular ECM workload, an average response time below 1 second was considered as good. In general, the parallel Spectrum Scale filesystem *can compete* with a high performance filesystem like XFS. There was *no significant difference*

between whether the ECM File Storage Area was on Spectrum Scale or XFS for the single ECM node. Only at the highest load level did Spectrum Scale show a higher average response times than XFS.

It is generally understood that a parallel file system such as Spectrum Scale adds additional management overhead to the filesystem, which appeared in this single ECM node setup just as overhead compared to a high performance non-parallel file system. However, this behavior emerged *only at the highest load level*, which can be considered as an extremely high load. At lower load levels, Spectrum Scale (with the applied SUT setup and this particular ECM workload) could compete well with XFS (which usually shows good performance).

Part 2 white paper

The Part 2 white paper discusses the ECM system scale-out to a four node WAS cluster. The multi ECM node setup benefits from:

- The capability of workload balancing in a WAS cluster.
- The strength of Spectrum Scale as a parallel filesystem.

For the multiple ECM node setup, the FileNet File Storage Area on a Spectrum Scale filesystem is shared among the nodes in the clustered setup.

In the Part 2 white paper, it is interesting to see how the multiple ECM node setup behaves compared to the single ECM node.

FileNet file storage area disk I/O

In addition to the CPU resources, the disk I/O for the FileNet Database Object Store and the File Storage Area were of interest. Especially the File Storage Area was considered in this case study.

Figure 14 on page 51 shows the disk I/O rates (read and write) for the File Storage Area at the different load levels for Spectrum Scale and XFS.

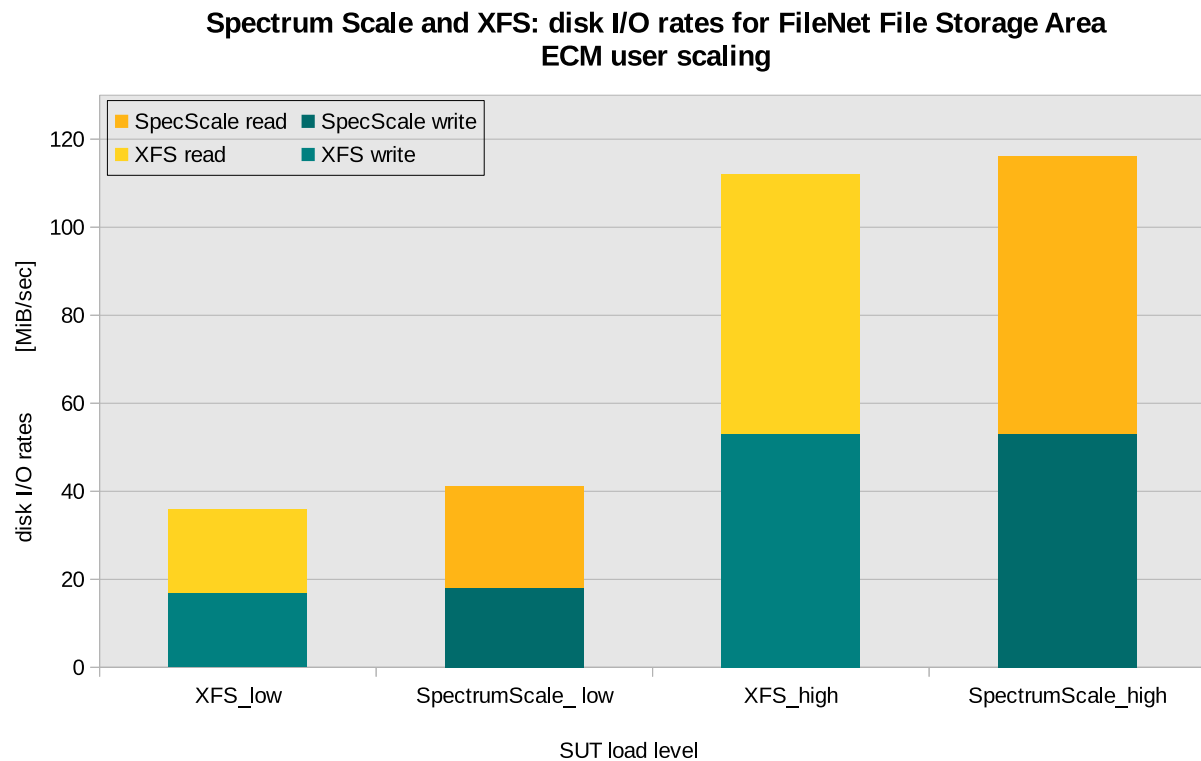


Figure 14. Spectrum Scale and XFS - FileNet File Storage Area disk IO rates

Note: XFS used the Linux page cache and Spectrum Scale a dedicated pagepool to cache filesystem data. For the single ECM node, a Spectrum Scale pagepool size of 8 GiB with `maxFilesToCache 20000` was chosen to allow for a fair comparison with XFS. XFS has around 8 GiB memory as Linux page cache available that is not used by any other application or service.

Observations

The disk I/O rates (read+write) for the FileNet File Storage Area were around 40 MiB/sec for the low load level, and increased up to 115 MiB/sec for the high load level. The percentage of 55% read versus 45% write requests was based on the chosen ECM performance workload for this case study.

The disk I/O rates for Spectrum Scale and XFS *were similar* across all load levels. The write I/O rate was almost exactly at the same rate and the read I/O slightly differs for both file systems.

Conclusion

The Spectrum Scale pagepool efficiency was close to the Linux page cache for XFS. In particular, the memory for pagepool and the Linux page cache were equal during the stable load phase of the workload.

However, the Linux page cache size was likely to change over the run-time period whereas the Spectrum Scale pagepool *had a fixed size*.

For Spectrum Scale, the `maxFilesToCache` parameter value was important. A proper value did not interfere with a good pagepool hit ratio (for details, see “IBM Spectrum Scale cache usage” on page 37).

z/VM total CPU load for the ECM system

This topic describes the z/VM total CPU loads with all running Linux guests.

Figure 15 shows the z/VM total CPU loads (hypervisor view) with all Linux guests running.

- The z/VM total CPU load is an indicator for the utilization of the SUT, where the top CPU consumers are the single ECM node and the ECM database node.
- Both z/VM total CPU loads for XFS and Spectrum Scale are shown side-by-side, together with the ICN page throughput for the different SUT load levels.

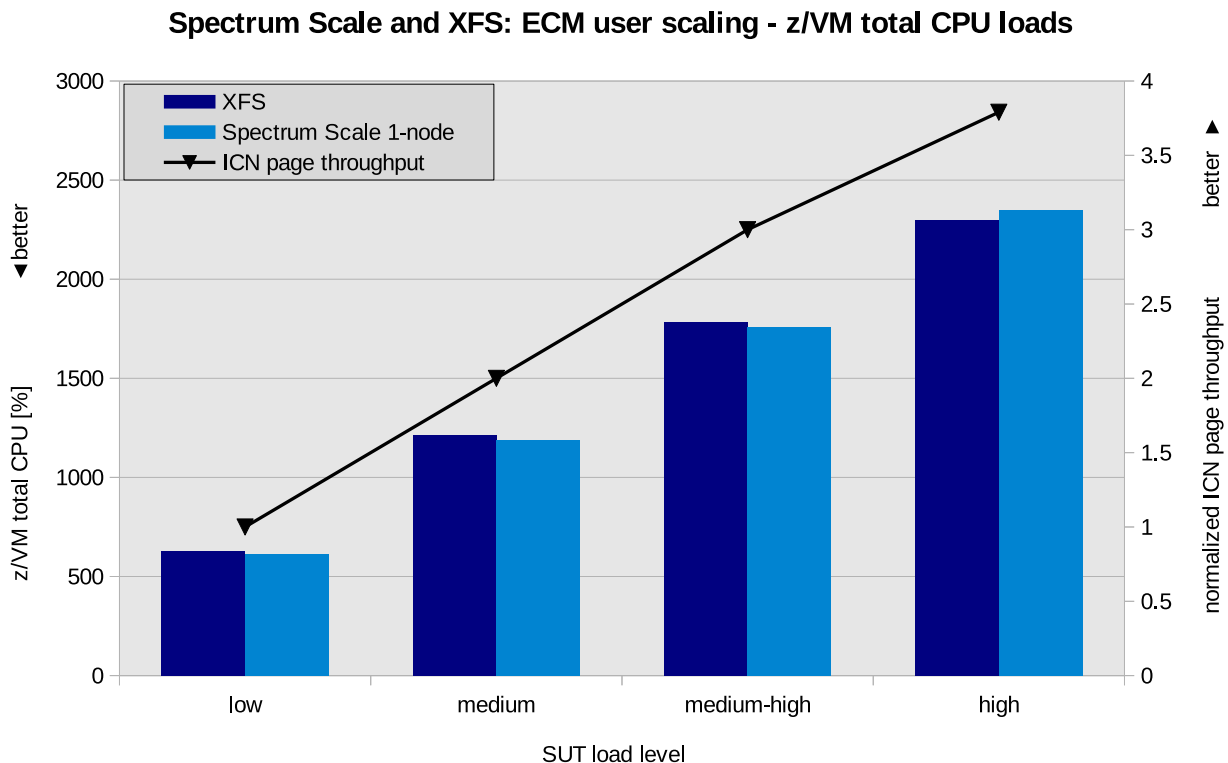


Figure 15. Spectrum Scale and XFS - z/VM total CPU loads and ICN page throughput

Observations

For the low load level, the z/VM total CPU was at 600% (6 IFL processors), and for the high load level around 2300% (23 IFL processors). The z/VM total CPU load *was similar* for both setups. For Spectrum Scale, the CPU load was up to 750 users which was *slightly lower* than for XFS. Only at the high load level was it slightly higher.

Conclusion

The CPU costs for Spectrum Scale (single node cluster) and XFS as FileNet File Storage Area filesystem are comparable. The expected additional overhead for a parallel filesystem did not occur in terms of significant additional CPU cost for the ECM node.

Chapter 9. Multiple ECM node setup with Spectrum Scale 4.2 (Part 2 white paper)

The Part 2 white paper in this series scales from a single ECM node up to a multiple ECM node setup with an IBM Spectrum Scale file system used as a shared FileNet File Storage Area.

The title of the white paper is:

*IBM Enterprise Content Management for Linux on z Systems, Scale-Out Case Study
(Part 2):*

*Multiple ECM Nodes with IBM WebSphere Application Server Cluster and Spectrum
Scale 4.2*

References

View a list of documents referenced in this white paper.

IBM FileNet P8 Platform V5.2.1 documentation (IBM Knowledge Center):

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1

IBM Content Navigator V2.0.3 documentation (IBM Knowledge Center):

http://www.ibm.com/support/knowledgecenter/SSEUEX_2.0.3

IBM WebSphere Application Server Network Deployment V8.5.5 documentation (IBM Knowledge Center):

http://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5

IBM Spectrum Scale V4.2.0 documentation (IBM Knowledge Center):

http://www.ibm.com/support/knowledgecenter/STXKQY_4.2.0

IBM Spectrum Scale and ECM FileNet Content Manager Are a Winning Combination: Deployment Variations and Value-added Features (IBM Redpaper™):

<http://www.redbooks.ibm.com/abstracts/redp5239.html?open>

IBM DB2 10.5 for Linux, Unix and Windows documentation (IBM Knowledge Center):

http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0

Linux on z Systems - Device Drivers, Features, and Commands, SC33-8411-nn:

http://www.ibm.com/developerworks/linux/linux390/development_documentation.html

Details about Linux on z Systems network tuning (IBM developerWorks):

http://www.ibm.com/developerworks/linux/linux390/perf/tuning_networking.html

How to Improve Performance with PAV, SC33-8414

z/VM V6R3 Library (IBM Knowledge Center):

http://www.ibm.com/support/knowledgecenter/SSB27U_6.3.0

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of 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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating

platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Adobe is either registered trademarks or trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of the manufacturer.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of the manufacturer.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any data, software or other intellectual property contained therein.

The manufacturer reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by the manufacturer, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

THE MANUFACTURER MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THESE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



Printed in USA