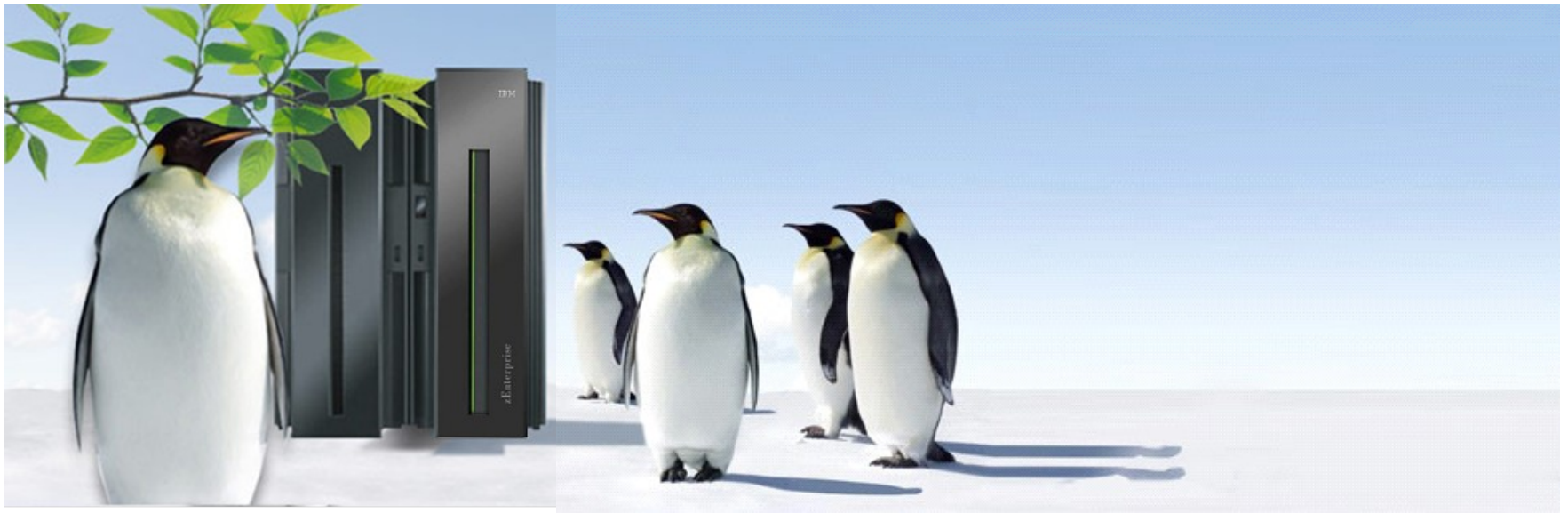
Robert J Brenneman – STG Systems Assurance <rjbrenn@us.ibm.com>

Elastic Storage 4.1 early experiences for **Linux on IBM System z**

A cluster file system with high-performance, high availability and parallel file access



Trademarks

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

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

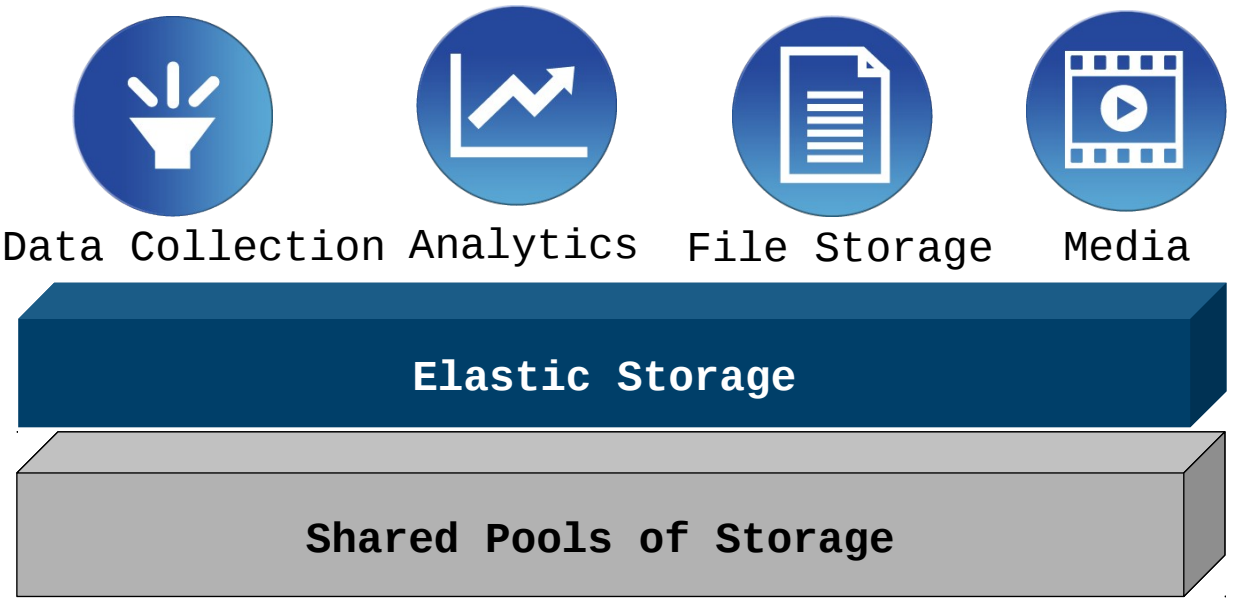
Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

- Elastic Storage - General overview
- Elastic Storage for Linux on System z
  - Overview Version 1
  - Usage scenarios
    - WebSphere AppServer
    - WebSphere MQ
  - Outlook
- Quick Install Guide

Elastic Storage

Provides fast data access and simple, cost effective data management



- Streamline Data access
- Centralize Storage Management
- Improve Data Availability

Clustered and Distributed File Systems

Clustered file systems

- File system shared by being simultaneously mounted on multiple servers accessing the same storage
- Examples: IBM GPFS, Oracle Cluster File System (OCFS2), Global File System (GFS2)

Distributed file systems

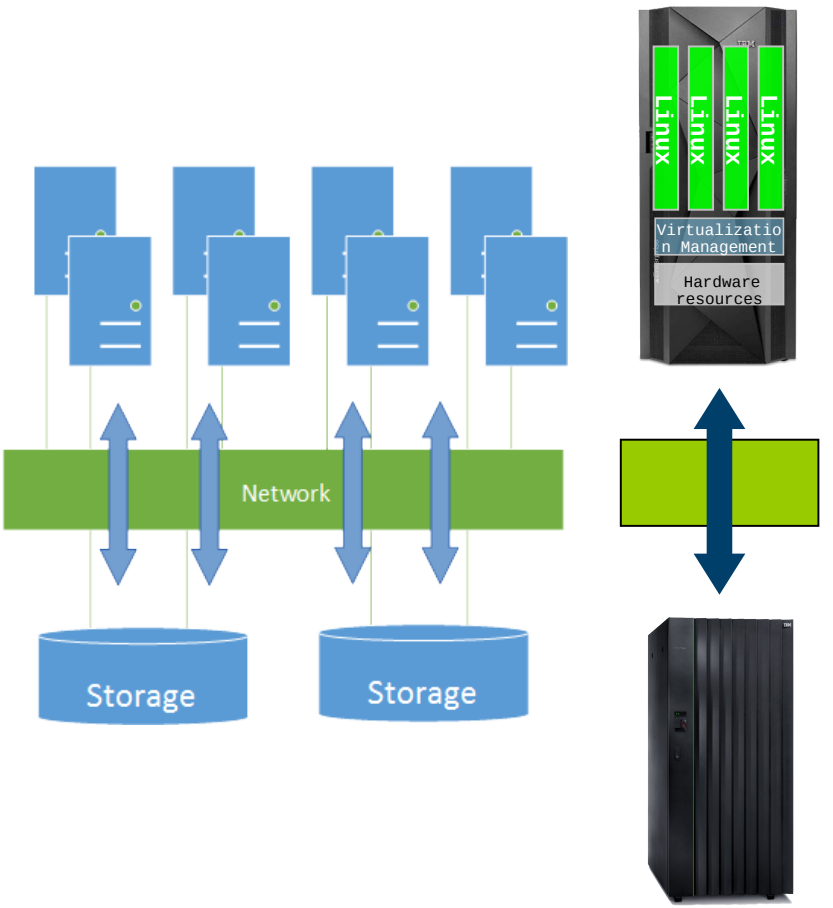
- File system is accessed through a network protocol and do not share block level access to the same storage
- Examples: NFS, OpenAFS, CIFS

Available for Linux for System z:

- SUSE Linux Enterprise Server
  - Oracle Cluster File system (OCFS2)
- Red Hat Enterprise Linux
  - GFS2 (via Sine Nomine Associates)

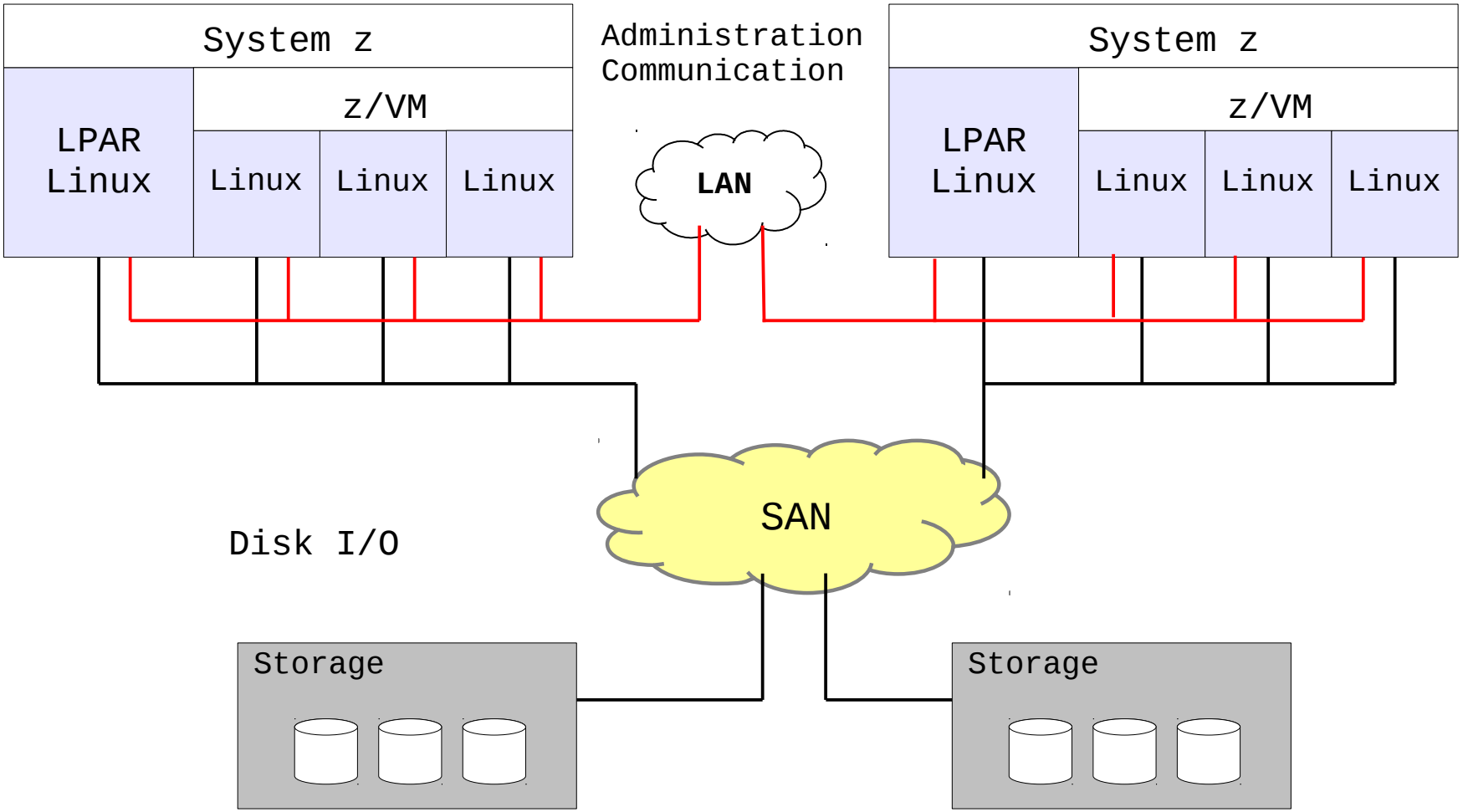
What is Elastic Storage?

- IBM's shared disk, parallel cluster file system
- Cluster: 1 to 16,384* nodes, fast reliable communication, common admin domain
- Shared disk: all data and metadata on storage devices accessible from any node through block I/O interface ("disk": any kind of block storage device)
- Parallel: data and metadata flow from all of the nodes to all of the disks in parallel.

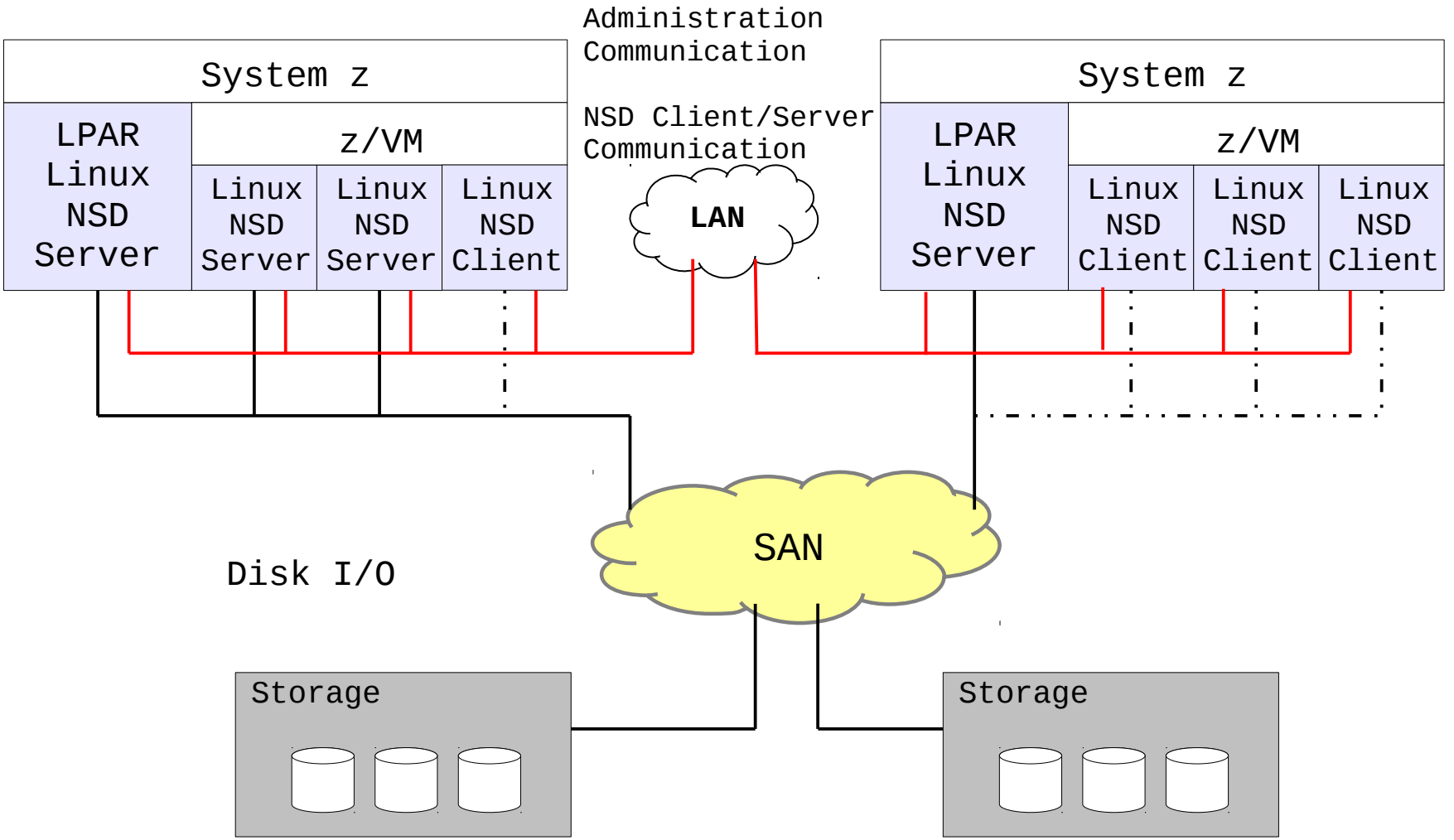


*largest cluster in production as of August 2014
Is LRZ SuperMUC 9400 Nodes of x86_64

Shared Disk (SAN) Model

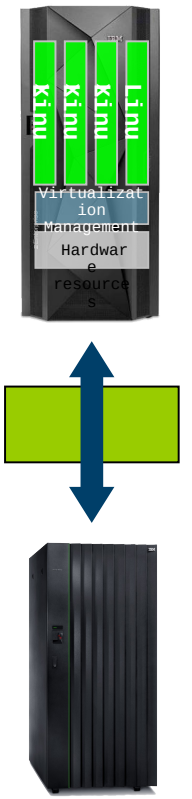


Network Shared Disk (NSD) Model



Elastic Storage 4.1 Features & Applications

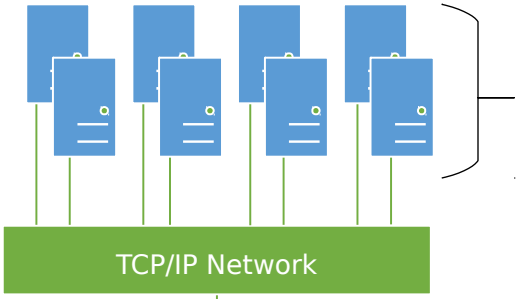
- Standard file system interface with POSIX semantics
  - Metadata on shared storage
  - Distributed locking for read/write semantics
- Highly scalable
  - High capacity (up to 2^{99} bytes file system size, up to 2^{63} files per file system)
  - High throughput (TB/s)
  - Wide striping
  - Large block size (up to 16MB)
  - Multiple nodes write in parallel
- Advanced data management
  - Snapshots, storage pools, ILM (filesets, policy)
  - Backup HSM (DMAPI)
  - Remote replication, WAN caching
- High availability
  - Fault tolerance (node, disk failures)
  - On-line system management (add/remove nodes, disks, ...)



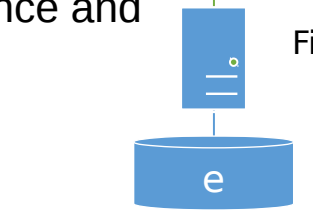
What Elastic Storage is NOT

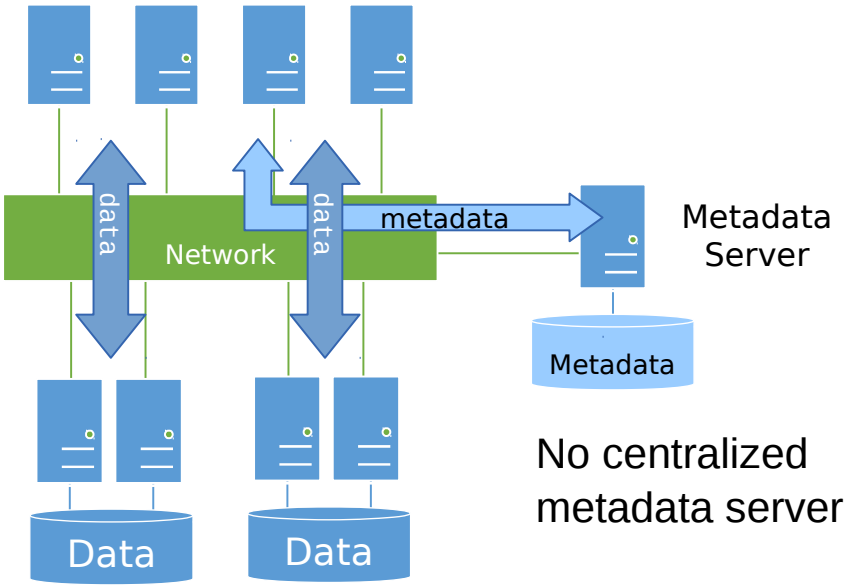
Not a client-server file system

like NFS, CIFS or AFS



No single-server performance and bottleneck scaling limits





No centralized metadata server

Elastic Storage – The Benefits

- Achieve greater IT agility
  - Quickly react, provision and redeploy resources
- Limitless elastic data scaling
  - Scale out with standard hardware, while maintaining world-class storage management
- Increase resource and operational efficiency
  - Pooling of redundant isolated resources and optimizing utilization
- Intelligent resource utilization and automated management
  - Automated, policy-driven management of storage reduces storage costs 90% and drives operational efficiencies
- Empower geographically distributed workflows
  - Placing of critical data close to everyone and everything that needs it

Elastic Storage for Linux on System z

Positioning

Elastic Storage 4.1 for Linux on System z will enable enterprise clients to use a highly available clustered file system with Linux in LPAR or as Linux on z/VM®.

IBM and ISV solutions will provide higher value for Linux on System z clients by exploiting Elastic Storage functionality:

- A highly available cluster architecture
  - Improved data availability through data access even when the cluster experiences storage or node malfunctions
- Capabilities for high-performance parallel workloads
  - Concurrent high-speed, reliable file access from multiple nodes in the cluster environment
- Smooth, non disruptive capacity expansion and reduction
- Services to effectively manage large and growing quantities of data

Elastic Storage for Linux on System z – Version 4.1

- Express Edition of Elastic Storage 4.1 + Service Updates is the base for the Linux on z support
  - **Express Edition: Contains the base GPFS functions**
  - Standard Edition: Includes the base function plus Information Lifecycle Management (ILM), Active File Management (AFM) and Clustered NFS
  - Advanced Edition: Includes encryption and the features of Standard Edition

- Content comprises:
  - Express Edition with base GPFS functions
  - Linux instances in LPAR mode or on z/VM (on the same or different CECs)
  - Up to 16 cluster nodes (same or mixed Linux distributions/releases)
  - Support for ECKD-based storage and FCP-based storage
    - DS8000 is supported only

Not Final Yet!

Elastic Storage for Linux on System z – Version 4.1

Minimum supported Linux distributions:

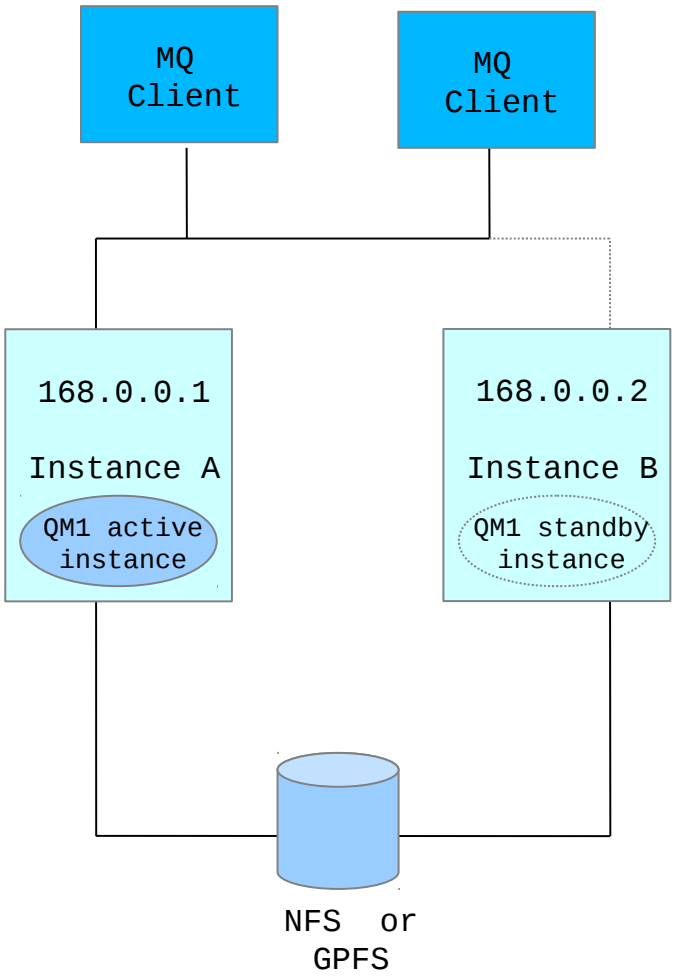
- SUSE Linux Enterprise Server (SLES) 11 SP3 + Maintweb-Update
- Red Hat Enterprise Linux (RHEL) 6.5 + Errata Update

While Elastic Storage V4.1 for Linux on System z does not support all functionality available for other platforms, this gap will be closed with the next updates.

Elastic Storage for Linux on System z is part of the mainstream development, all future enhancements of Elastic Storage will become available for Linux on System z.

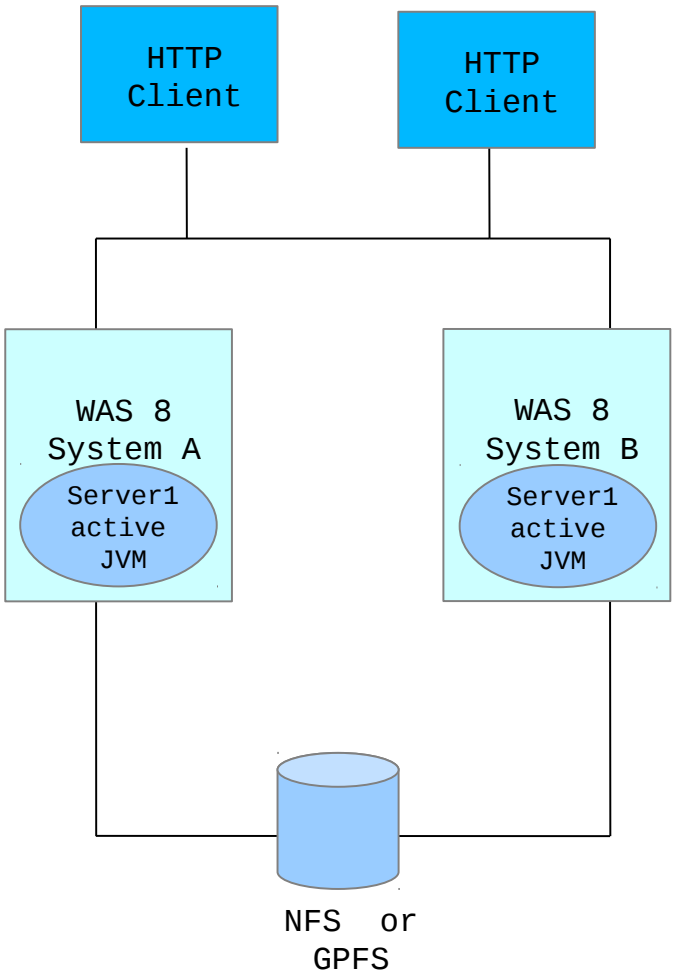
Use Case for WebSphere MQ Multi-Instance Queue Manager (MIQM)

- High availability configuration of WebSphere MQ with two instances of the queue manager running on different servers, and either instance can be active.
  - A shared file system is required on networked storage, such as a NFS, or a cluster file system such as GPFS



Use Case for WebSphere AppServer HA Cluster

- High availability configuration of WebSphere AppServer with two instances of the application running on different servers, and both instances are active.
  - A shared file system is required for transaction logs on networked storage, such as a NFS, or a cluster file system such as GPFS



Use Case for WebSphere MQ or WebSphere AppServer

- Advantages of GPFS versus NFS
  - No single-server bottleneck
  - No protocol overhead for data (network) transfer
  - Interacts with applications like a local file system, while
  - delivering high performance, scalability and fault tolerance by allowing data access from multiple systems directly and in parallel
  - Maintaining file-data integrity while allowing multiple applications / users to share access to a single file simultaneously

Outlook

- Multi-Cluster support
- Stretch-Cluster support (20, 40, 100, 200km for active/active DR configurations)
- Active File Management (AFM) / Information Lifecycle Management (ILM)
- AFM for active/backup configurations for clients not basing on hardware-based cross-site data replication (HA and DR)
- Tivoli Storage Manager (both backup and Hierarchical Storage Management (HSM))
- Support for heterogeneous clusters (mix of AIX, Linux on System x,p,z)
- Encryption
- Support for Storwize, XIV, FlashSystem, SVC storage servers

Quick Install Guide
Linux on System z

Prerequisites Linux Distribution and Storage Hardware

- Supported Linux Distribution

Distribution	Minimum level	Kernel
SLES 11	SUSE Linux Enterprise Server 11 SP3 + Maintweb Update or later maintenance update or Service Pack	3.0.101-0.15-default
RHEL 6	Red Hat Enterprise Linux 6.5 + Errata Update RHSA-2014-0328 or later miner update	2.6.32-431.11.2.el6
RHEL 7		3.10.0-123.el7

- Supported Storage System
  - DS8000

Software Prerequisites

- Additional Kernel Parameter
  - set the following kernel parameters in */etc/zipl.conf* when booting the kernel
  - `vmalloc = 4096G`
  - `user_mode = home`

```
# cat /etc/zipl.conf
Parameters = "... vmalloc=4096G user_mode=home ..."
```

- Passwordless communication between nodes of GPFS cluster
- Cluster system time coordination via NTP or equivalent
- Required kernel development packages to be installed on at least 1 system to build the kernel modules

Exchange ssh keys between all GPFS nodes

- Passwordless access between all GPFS nodes is a prerequisite
- Exchange ssh key from one node to all other nodes
  - Create ssh-keys at node1:

```
# cd .ssh
# ./ssh-keygen  #hit return by all questions
```

  - Copy ssh keys to authorized_keys at node1:

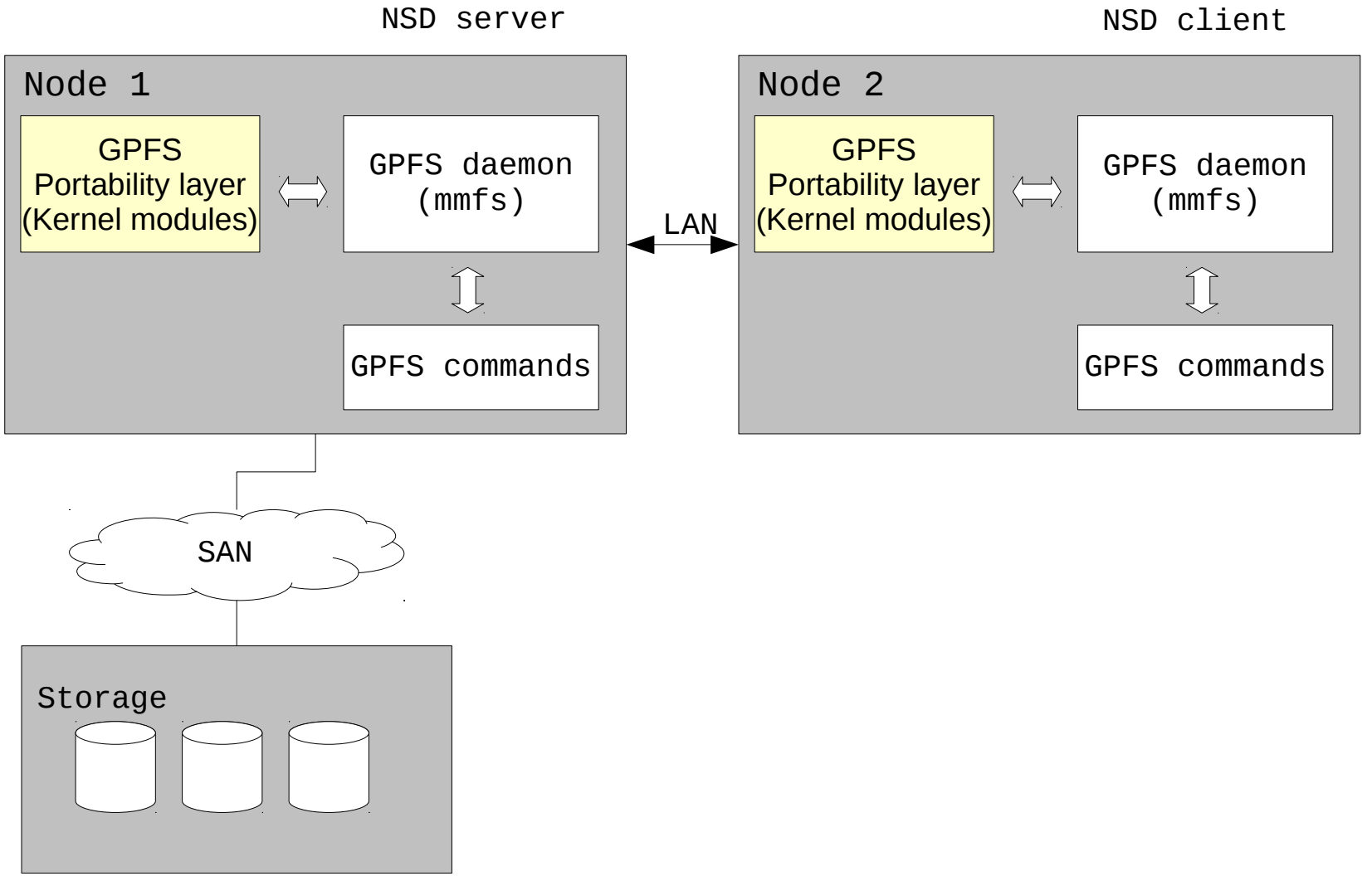
```
# cat id_rsa.pub >> authorized_keys
# ssh localhost
# ssh node1
# ssh node1.domain.com
```

  - Copy id_rsa.pub to other nodes

```
# ssh-copy-id -i /root/.ssh/id_rsa.pub root@node2
```

  - Do ssh connects from each node to each other node and localhost (with and without the domain name)

Overview



Install GPFS product

- Install GPFS product RPM packages on all nodes of the cluster
  - Packages name: gpfs.*.rpm
- GPFS product files can be found after installation at
  - /usr/lpp/mmfs
- Build the GPFS kernel modules (portability layer) e.g. development system

```
# cd /usr/lpp/mmfs/src/
# make Autoconfig
# make World
# make InstallImages
```

- Build an rpm (`make rpm`) and install this rpm on all related nodes
- Reboot all nodes

Plan for GPFS Cluster

- Create a NodeFile to define the role of the nodes (FS Manager): e.g. nodes.file

```
node1:quorum-manager:
node2:quorum-manager:
node3:quorum:
node4::
```

- Create a stanza file to define Network Shared Disks (NSD) to be used by GPFS file systems : e.g. nsd.file

```
%nsd: device=/dev/dm-4
      nsd=NSD_1
      servers=node1,node2
      usage=dataAndMetadata
%nsd: device=/dev/dm-5
      nsd=NSD_2
      servers=node1
      usage=dataAndMetadata
%nsd: device=/dev/dm-6
      nsd=NSD_3
      servers=node1
      usage=dataAndMetadata
```

Quick Install Guide

- Create a GPFS cluster
  - -A options: Start GPFS daemons automatically when nodes come up

```
node1# mmcrcluster -N nodes.file -C cluster1 -r /usr/bin/ssh
-R /usr/bin/scp -A
```

- Change the type of GPFS license associated with the nodes

```
node1# mmchlicense server --accept -N node1,node2,node3
node1# mmchlicense client --accept -N node4
```

- Start the GPFS cluster on all nodes

```
node1# mmstartup -a
```

Quick Install Guide (cont'd)

- Get information about the previously activated GPFS cluster

```
node1# mmlscluster
GPFS cluster information
========================
  GPFS cluster name:         cluster1
  GPFS cluster id:           18000255686092070264
  GPFS UID domain:           cluster1.domain.com
  Remote shell command:      /usr/bin/ssh
  Remote file copy command:  /usr/bin/scp
  Repository type:           CCR

GPFS cluster configuration servers:
-----------------------------------
  Primary server:    node1.domain.com (not in use)
  Secondary server:  (none)

Node  Daemon node name    IP address   Admin node name   Designation
-------------------------------------------------------------------
  1   node1.domain.com   10.20.80.86   node1.domain.com  quorum-manager
  2   node2.domain.com   10.20.80.87   node1.domain.com  quorum-manager
  3   node3.domain.com   10.20.80.88   node1.domain.com  quorum
  4   node4.domain.com   10.20.80.89   node1.domain.com
```

Quick Install Guide (cont'd)

- Get information about the status of the GPFS cluster

```
node1# mmgetstate -a
 Node number  Node name        GPFS state
------------------------------------------
       1      node1            active
       2      node2            active
       3      node3            active
       4      node4            active
```

- Create Network Shared Disks used by GPFS

```
node1# mmcrnsd -F nsd.file
```

- Create an GPFS file system
  - -A option: File system will be mounted when GPFS daemon starts

```
node1# mmcrfs esfs1 -F nsd.file -T /elastic_storage -A yes
node1# mmcrfs esfs2 “NSD_4;NSD_5” -T /elastic_storage2 -A yes
```

Quick Install Guide

- Retrieve information about the Network Shared Disks

```
node1# mmlsnsd

 File system   Disk name    NSD servers
---------------------------------------------------------------
 esfs1         NSD_1        node1.domain.com,node2.domain.com
 esfs1         NSD_2        node1.domain.com
 esfs1         NSD_3        node1.domain.com
```

- Mount all GPFS file systems on all nodes in the cluster

```
node1# mmmount all -a
```

Manage GPFS Cluster: useful commands

- Manage GPFS Cluster / Node
  - mmcrcluster, mmchcluster, mmlscluster
  - mmstartup, mmshutdown
  - mmchlicense
  - mmaddnode, mmchnode, mmdelnode, mmlsnode
- Manage Network Shared Disks (NSD)
  - mmcrnsd, mmchnsd, mmdelnsd, mmlsnsd
- Manage GPFS Filesystem
  - mmcrfs, mmchfs, mmdelfs, mmlsfs
  - mmcrsnapshot, mmdelsnapshot, mmlssnapshot
  - mmadddisk, mmchdisk, mmdeldisk, mmlsdisk

Resources

- ibm.com:

ibm.com/systems/platformcomputing/products/gpfs/

- Public Wiki:

ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General Parallel File System (GPFS)

- IBM Knowledge Center:

ibm.com/support/knowledgecenter/SSFKCN/gpfs_welcome.html?lang=en

- Data sheet: IBM General Parallel File System (GPFS) Version 4.1

ibm.com/common/ssi/cgi-bin/ssialias?subtype=SP&infotype=PM&appname=STGE_DC_ZQ_USEN&htmlfid=DCD12374USEN&attachment=DCD12374USEN.PDF

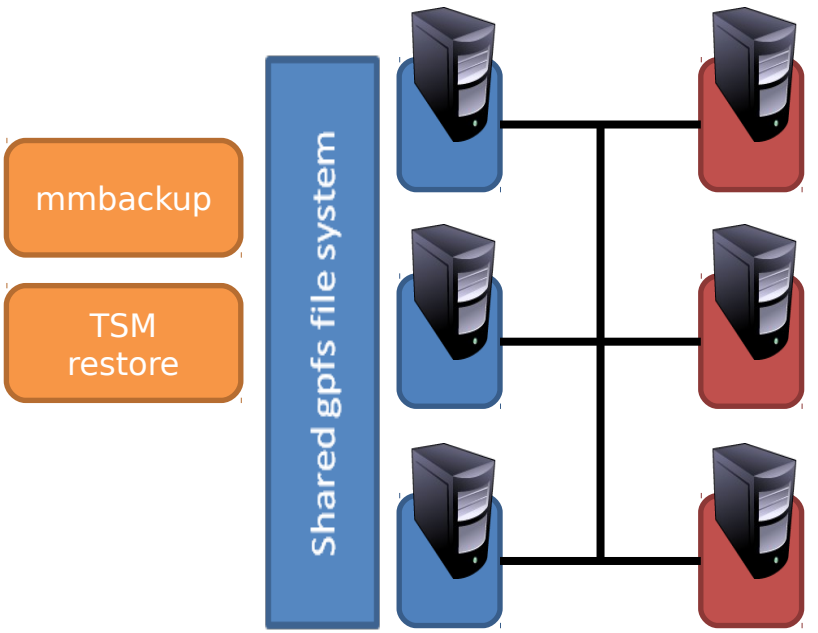
Questions?

Appendix

Use Case for Large File System Backup Leveraging Tivoli Storage Manager (TSM)

Support planned

- Much faster file system scanning times allows TSM backups to scale to many more objects compared to TSM BA client
- mmbackup can utilize multiple GPFS nodes to scan the file system and take backups
- Multiple TSM servers can be used to protect a single GPFS file system
- TSM GUI or CLI can be used to traverse the protected data for individual file restore



Use Case for Large File System Backup (cont'd) Leveraging Tivoli Storage Manager (TSM)

Support planned

- Advantages
  - High backup scalability. Only file system metadata (inode & path) has to be backed up.
  - High restore performance. File data resides on the TSM Server and recall happens on demand.
  - Lifts the ACL/EA limitation of the TSM Server. Complete inode information is part of the image file.
  - Recreates the whole directory tree in one step with all permissions and all files in stub format.