IBM Systems

Lab Experiences Running GPFS on Linux on System Z

Jay Brenneman

rjbrenn@us.ibm.com

Trademarks

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

AIX*	FlashSystem	Storwize*	Tivoli*
DB2*	IBM*	Spectrum Scale*	WebSphere*
DS8000*	IBM (logo)*	System p*	XIV*
ECKD	MQSeries*	System x*	z/VM*
		System z*	Z Systems*

* Registered trademarks of IBM Corporation

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.
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.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
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.
Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.
TEALEAF is a registered trademark of Tealeaf, an IBM Company.
Windows Server and the Windows logo are trademarks of the Microsoft group of countries.
Worklight is a trademark or registered trademark of Worklight, an IBM Company.
UNIX is a registered trademark of The Open Group in the United States and other countries.

* Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

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

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

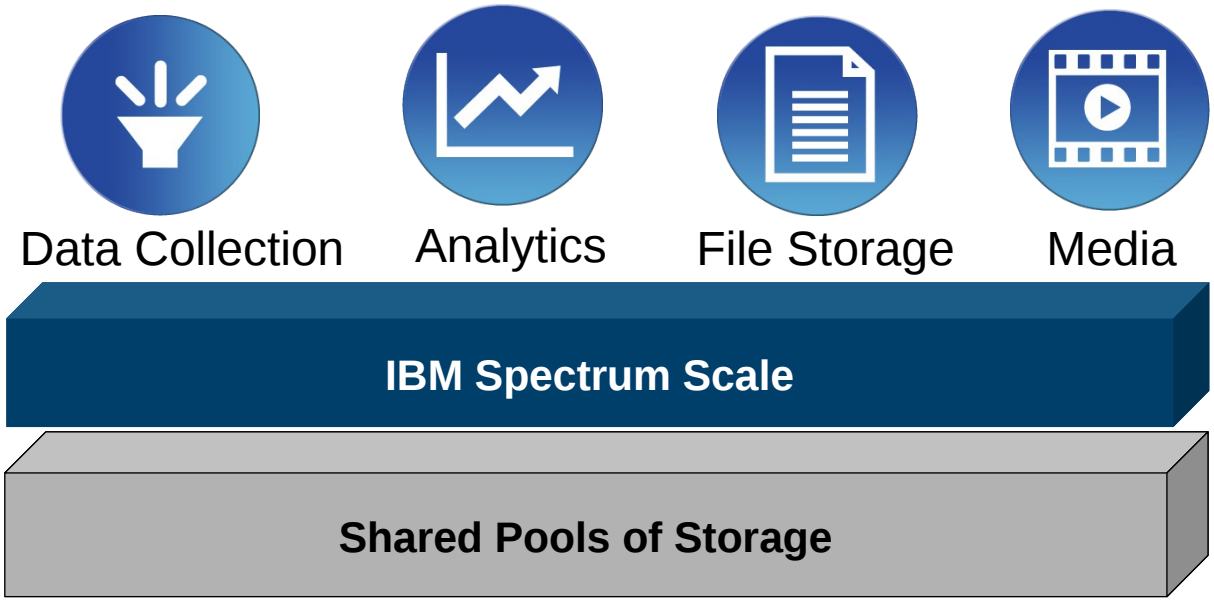
Agenda

- Overview of
  IBM Spectrum Scale / Elastic Storage / GPFS

- Installation and configuration without passwordless
  remote root

IBM Spectrum Scale*

Provides fast data access and simple, cost effective data management



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

* Formerly “Elastic Storage”

Clustered and Distributed File Systems

Clustered file systems

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

Available for Linux for z Systems:

SUSE Linux Enterprise Server

Oracle Cluster File system (OCFS2)

Red Hat Enterprise Linux

GFS2 (via Sine Nomine Associates)

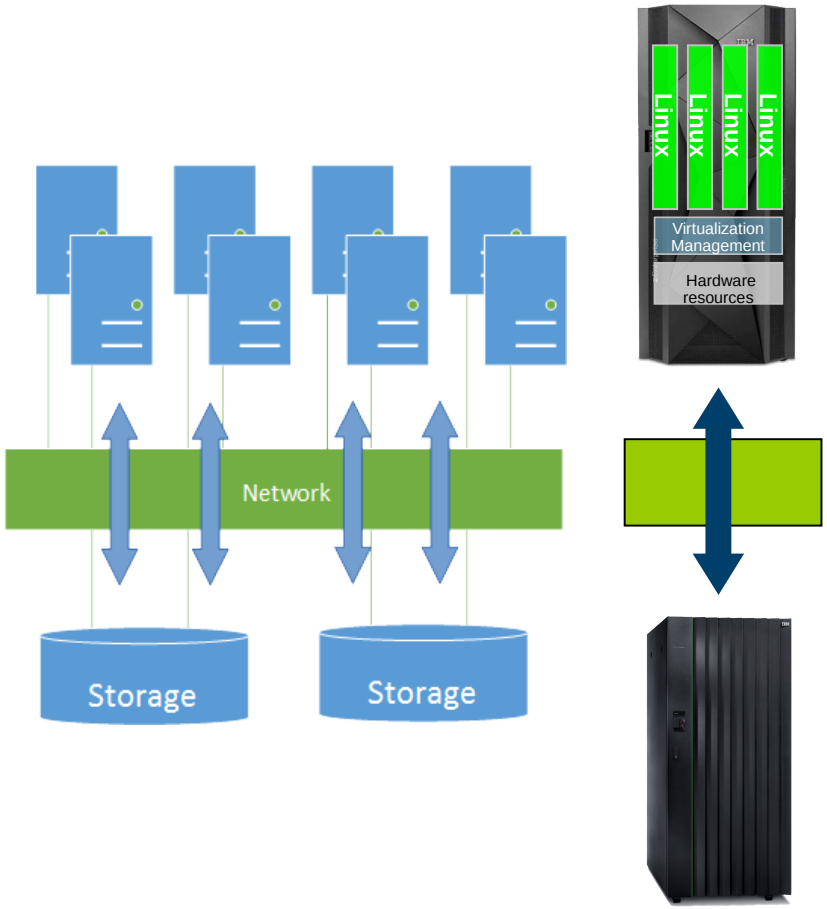
Distributed file systems

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

What is IBM Spectrum Scale?

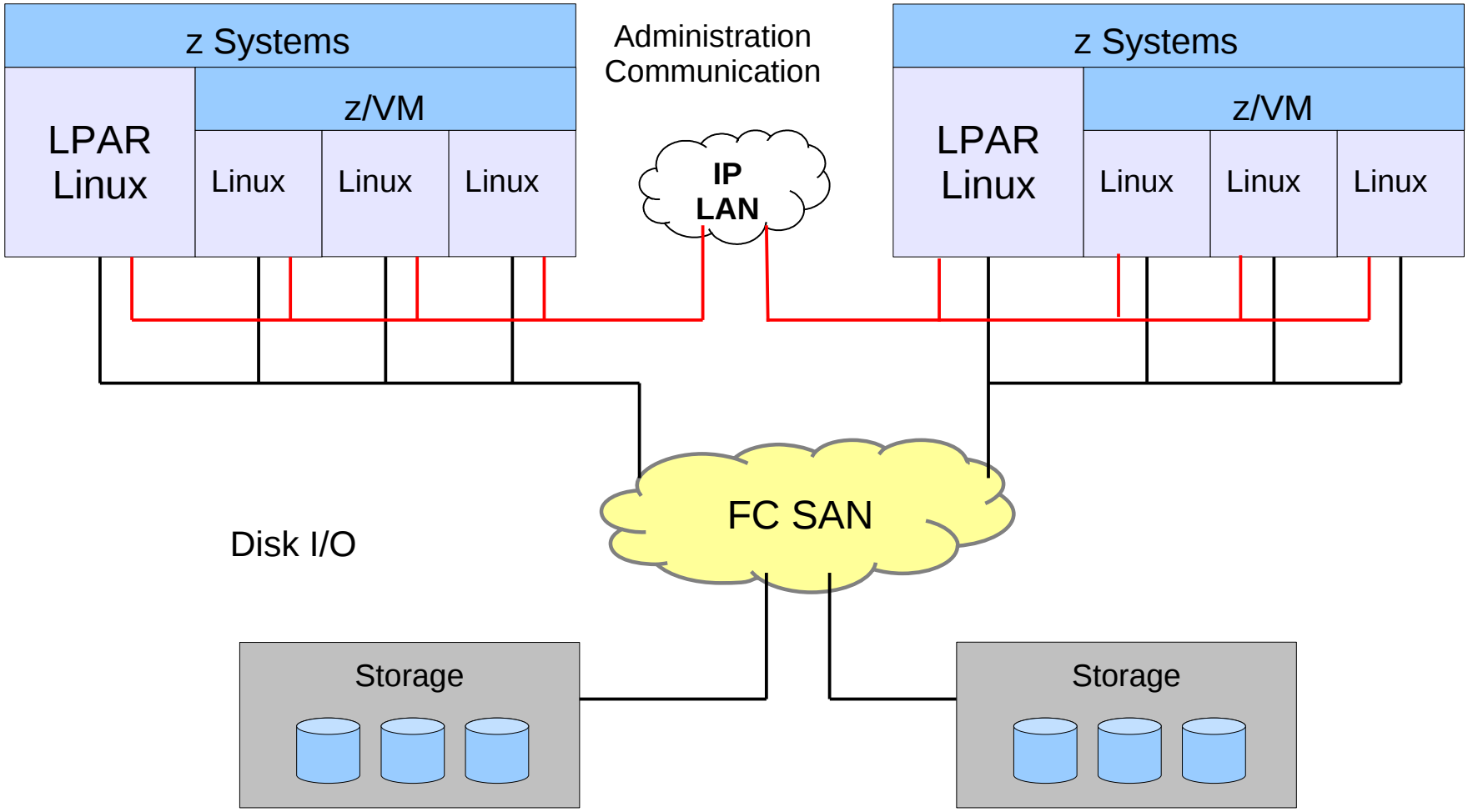
IBM's shared disk, parallel cluster file system

- 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.

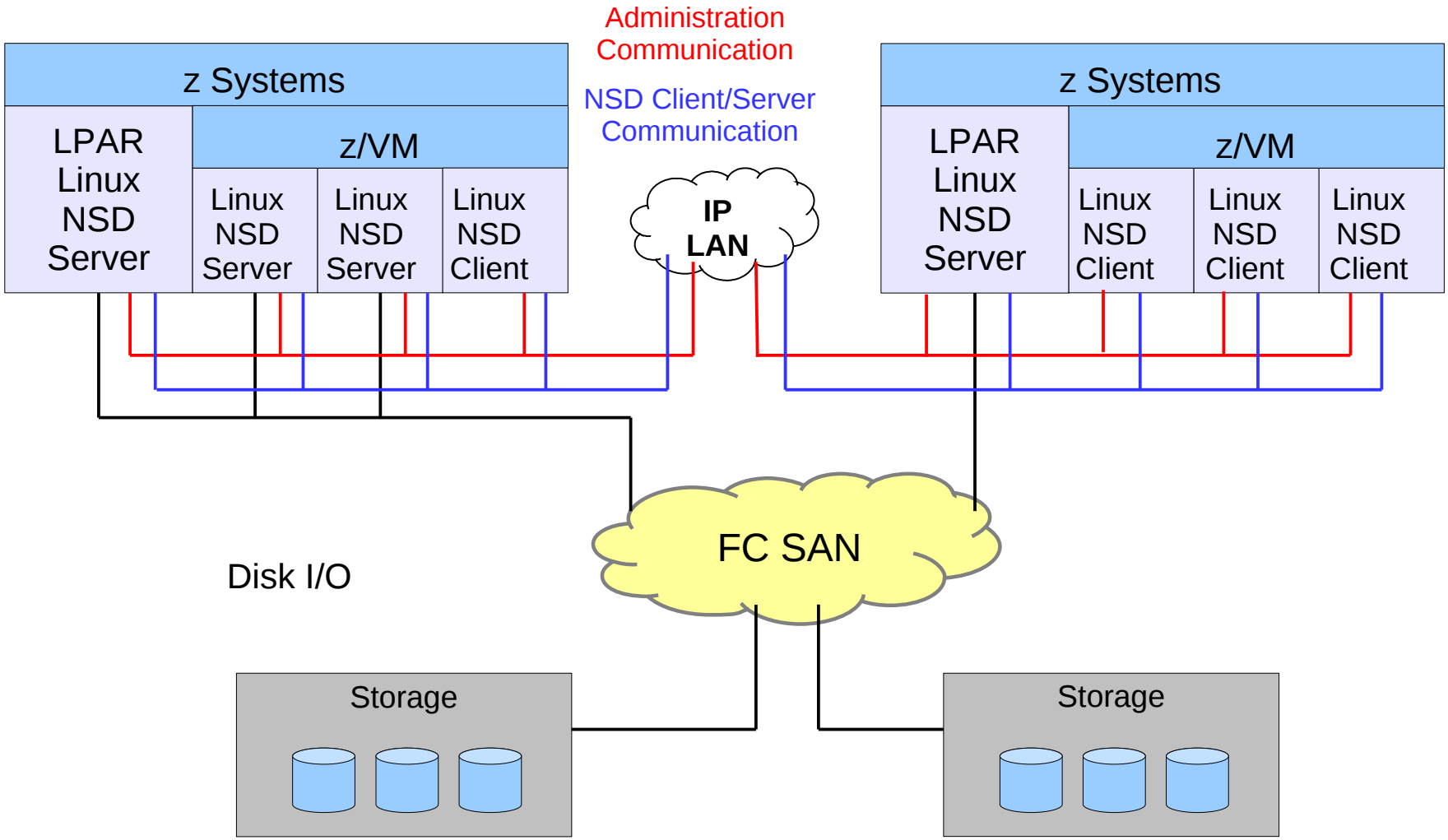


*largest cluster in production as of August 2014
Is LRZ SuperMUC 9400 Nodes of x86_64

Shared Disk (SAN) Model

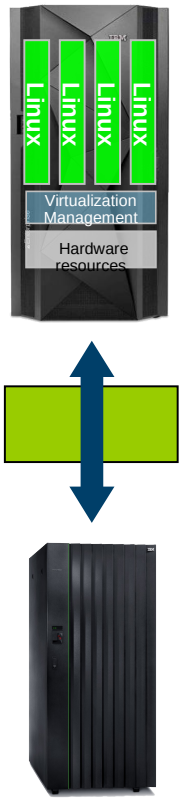


Network Shared Disk (NSD) Client/Server Model



IBM Spectrum Scale 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
  - ILM (storage pools), Snapshots
  - Backup HSM (DMAPI)
  - Remote replication, WAN caching
- High availability
  - Fault tolerance (node, disk failures)
  - On-line system management (add/remove nodes, disks, ...)



IBM Spectrum Scale for Linux on z Systems Version 4.1

- Linux instances in LPAR mode or on z/VM, on the same or different CECs
  - IBM Spectrum Scale has no dependency on a specific version of z/VM
- Up to 32 cluster nodes with same or mixed Linux distributions/releases
- Heterogeneous clusters with client nodes without local storage access running on AIX, Linux on Power and Linux on x86
- Support for ECKD-based and FCP-based storage
- Support for IBM System Storage DS8000 Series, IBM Storwize V7000 Disk Systems, IBM XIV Storage Systems and IBM FlashSystem Systems
  - EMC & Hitachi are supported through their normal support channels – there is no special sauce in GPFS other than a requirement for SCSI-3 Persistent Reserve for enhanced failure recovery paths
- Supported workloads are IBM WebSphere Application Server, IBM WebSphere MQ or similar workloads

The Express Edition does not include features, therefore IBM is planning to offer enhanced functionality in future versions of IBM Spectrum Scale for Linux on z Systems.

Agenda

- Overview of
  IBM Spectrum Scale / Elastic Storage / GPFS

- Installation and configuration without passwordless remote root

Linux Distribution and Storage Hardware Prerequisites

- 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	Red Hat Enterprise Linux 7.0	3.10.0-123.el7

- Supported Storage System
  - DS8000, XIV, V7000 and FlashSystem, or
  - Basically any SAN disk supported by Linux on Z if you're not going to try to exploit SCSI-3 PR – please coordinate with GPFS development
- IBM Spectrum Scale has no dependency on a specific version of z/VM

Software Prerequisites

- Additional Kernel Parameters
  - set the following kernel parameters in */etc/zipl.conf* when booting the kernel
  - `vmalloc = 4096G`
  - `user_mode = home` `# only required on RHEL 7.0`

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

- Ksh package
- Cluster system time coordination via NTP, STP or equivalent
- Required kernel development packages to be installed on at least one system to build the kernel modules
  - This system need not actually be a member of the cluster
- Passwordless communication between nodes of GPFS cluster

Install GPFS

- Extract the 84 MB tarball, accept the License, and install the RPMs contained therein on every cluster member

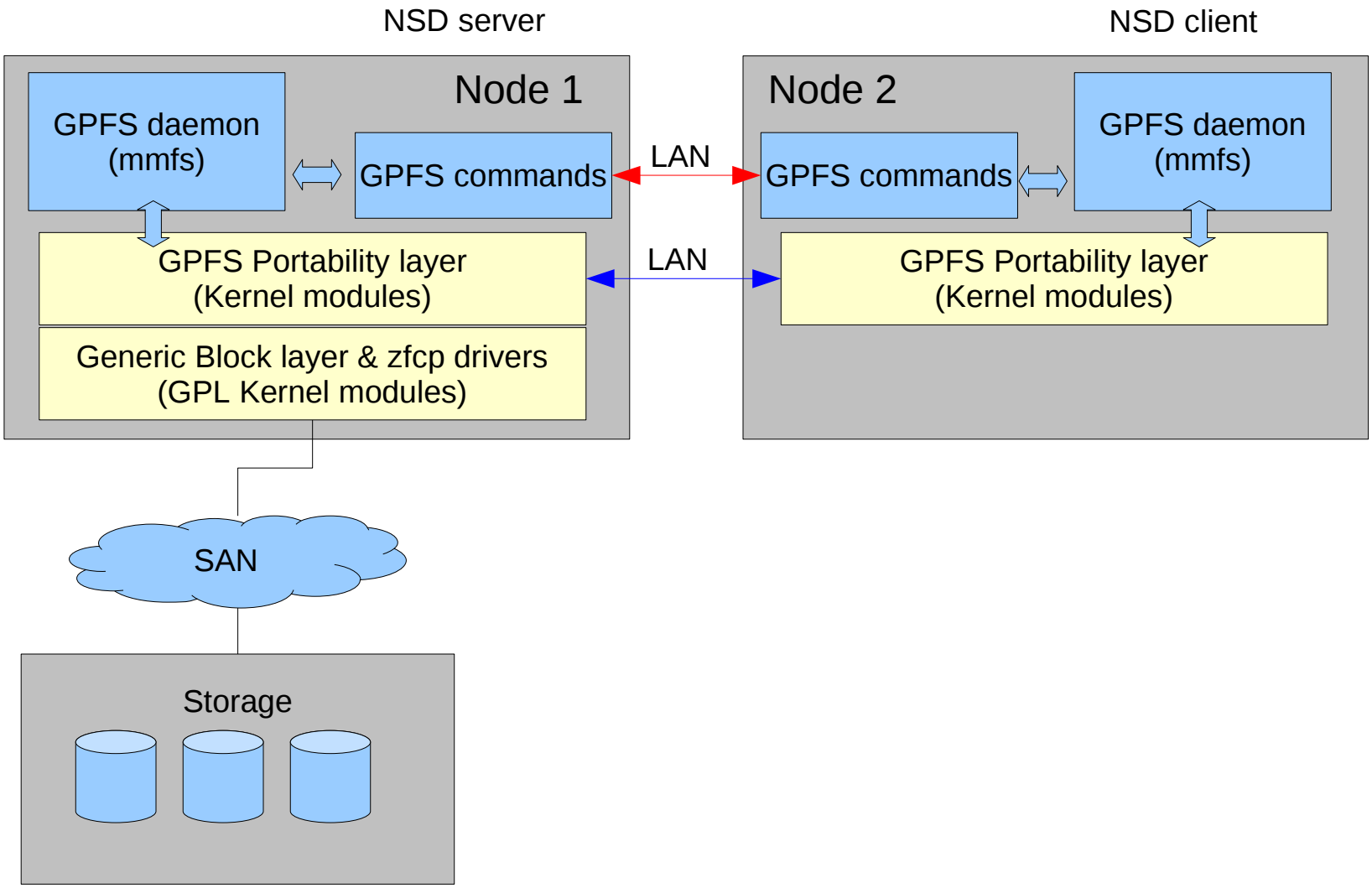
```
#> rpm -ivh gpfs*
Preparing...                          ################################# [100%]
Updating / installing...
   1:gpfs.msg.en_US-4.1.0-5           ################################# [ 25%]
   2:gpfs.gskit-8.0.50-32             ################################# [ 50%]
   3:gpfs.docs-4.1.0-5                ################################# [ 75%]
   4:gpfs.base-4.1.0-5                ################################# [100%]
```

- Build the kernel module by installing all the above plus the the gpl package on the build system and run mmbuildgpl to create an rpm for the real cluster members

```
#> rpm -ivh gpfs.gpl-4.1.0-5.noarch.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:gpfs.gpl-4.1.0-5                 ################################# [100%]
#> cd /usr/lpp/mmfs/src && ../bin/mmbuildgpl –buildrpm
------------------------------------------------------------
mmbuildgpl: Building GPL module begins at Mon Mar  2 19:46:03 EST 2015.
------------------------------------------------------------
<output snipped>
Wrote: /root/rpmbuild/RPMS/s390x/gpfs.gplbin-3.10.0-123.13.2.el7.s390x-4.1.0-5.s390x.rpm
------------------------------------------------------------
mmbuildgpl: Building GPL module completed successfully at Mon Mar  2 19:46:18 EST 2015.
------------------------------------------------------------
```

- Copy and Install the new GPFS kernel module RPM on every cluster member

Component Overview



Passwordless Communication between Nodes

- GPFS is a file system with both root daemon processes and kernel modules
- The commands must be able to copy config files among nodes and replicate commands to keep the cluster in sync
- Use either:
  - root to root passwordless ssh and scp
  - Sudo and ssh & scp wrappers to provide equivalent function while retaining auditability

Read https://ibm.biz/BdENLk
Then send a note to gpfs@us.ibm.com to request a example of sudo based wrappers

Passwordless non-root setup with sudo for GPFS

Make Keys and Users

- `#> ssh-keygen`
  `Generating public/private rsa key pair.`
  - Goes into /root/.ssh/id_rsa & id_rsa.pub
- `#> ssh root@<hostname> && ssh root@<hostname.fully.qualified.com>`
  - For each member of the cluster to fully populate .ssh/known_hosts
- `#> groupadd gpfsadmins`
  `#> useradd -m someguy -G gpfsadmins` (or use LDAP)
  `#> passwd someguy` (otherwise repeat on every node)
  `#> id someguy`
  `uid=1000(someguy) gid=1000(someguy) groups=1000(someguy),1001(gpfsadmins)`

Set Up sudo for the new group

- `#> visudo`
  `%s/Defaults requiretty/Defaults !requiretty/`
- `#> vi /etc/sudoers.d/00_gpfs`

```
## create a new file on RHEL 7 or just add in sudoers on others ##

Defaults env_keep = "LANG LC_ADDRESS LC_CTYPE LC_COLLATE LC_IDENTIFICATION
LC_MEASUREMENT LC_MESSAGES LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE
LC_TIME LC_ALL LANGUAGE LINGUAS XDG_SESSION_COOKIE MMMODE environmentType
GPFS_rshPath GPFS_rcpPath  mmScriptTrace GPFSCMDPORTRANGE GPFS_CIM_MSG_FORMAT"

%gpfsadmins ALL = (ALL) PASSWD: ALL, NOPASSWD: /usr/lpp/mmfs/bin/mmremote,
/usr/bin/scp, /bin/echo
```

Passwordless non-root setup with sudo for GPFS

- #> mkdir /home/someguy/.ssh
  #> cp /root/.ssh/id_rsa.pub /home/someguy/.ssh/authorized_keys
  #> cp /root/.ssh/known_hosts /home/someguy/.ssh
  #> chown -R someguy:someguy /home/someguy/.ssh
  #> ssh someguy@<hostname> && ssh someguy@<hostname.fully.qualified.com>
  - To check that it does not prompt for a password
- Copy the root and someguy .ssh directory contents to every cluster member.
  The root user must be able to ssh to any node as someguy without a password.
  Someguy doesn't need to be able to do this for GPFS, but you will want it for own purposes if you are not permitted to use root

Distribute ssh credentials

- Copy the ssh wrappers you got back from gpfs@us.ibm.com to /usr/lpp/mmfs/bin and make sure they are executable by root

```
#> ls -l /usr/lpp/mmfs/bin/*.pl
-rwx------ 1 root root 1688 Feb 26 20:45 /usr/lpp/mmfs/bin/scpwrap.pl
-rwx------ 1 root root  591 Feb 26 20:45 /usr/lpp/mmfs/bin/sshscpwrap.pl
-rwx------ 1 root root 3349 Feb 26 20:45 /usr/lpp/mmfs/bin/sshwrap.pl
```

- Make sure the Perl Env module is installed (required by wrappers)
  #> yum install perl-Env

Deploy Wrappers

Create a GPFS Cluster

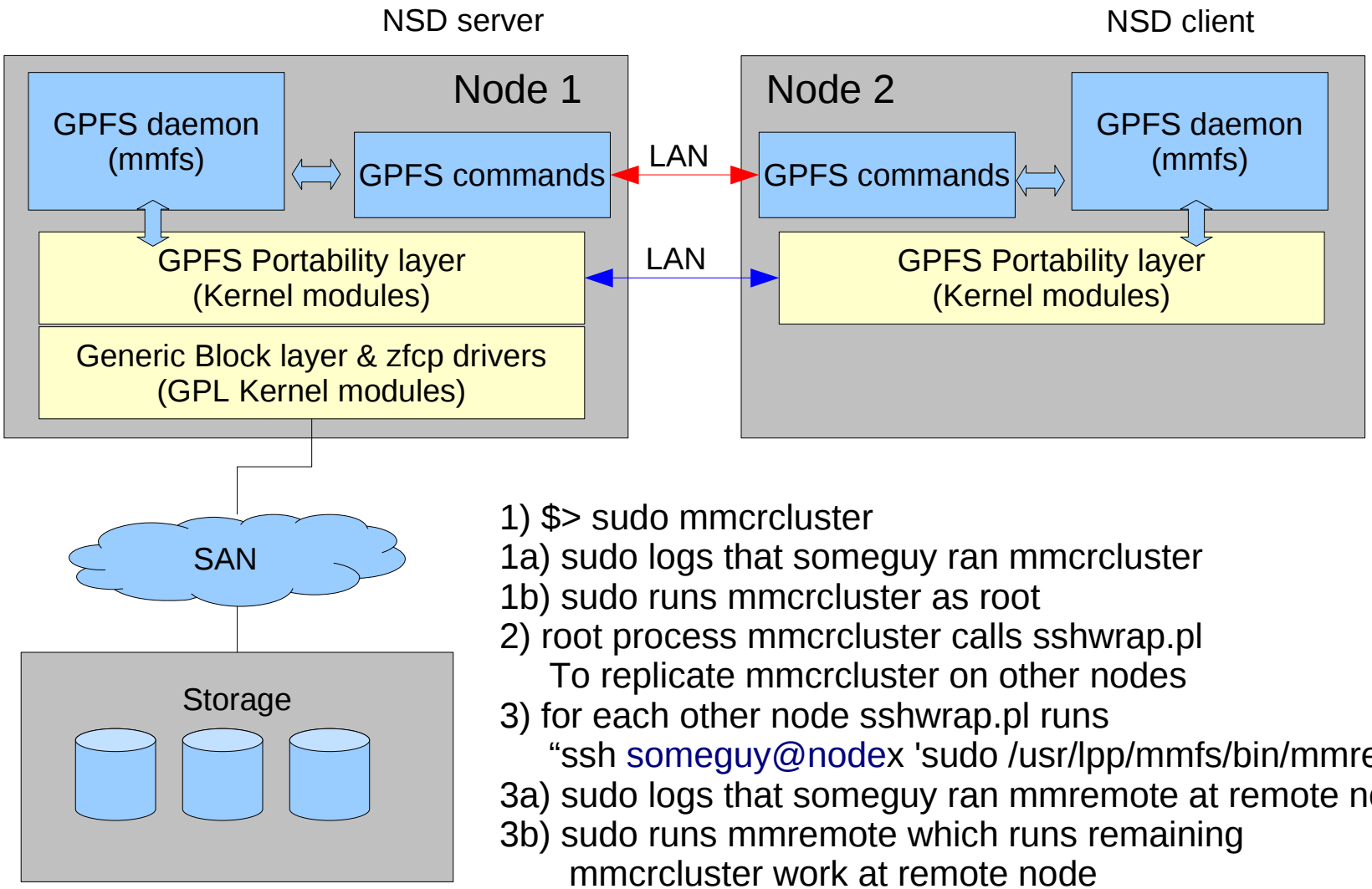
- Create a node file which lists each cluster member and its role

```
fpstoc1a:quorum-manager:
fpstoc1b::
fpstoc1c:quorum-manager:
fpstoc1d:quorum:
```

- Create a GPFS cluster with mmcrcluster

```
$> sudo /usr/lpp/mmfs/bin/mmcrcluster
          -N /scratch/fpstoc1.nodefile
          -C fpstoc1 --ccr-enable
          -r /usr/lpp/mmfs/bin/sshwrap.pl
          -R /usr/lpp/mmfs/bin/scpwrap.pl
mmcrcluster: Performing preliminary node verification ...
mmcrcluster: Processing quorum and other critical nodes ...
mmcrcluster: Processing the rest of the nodes ...
mmcrcluster: Finalizing the cluster data structures ...
mmcrcluster: Command successfully completed
mmcrcluster: Warning: Not all nodes have proper GPFS license designations.
    Use the mmchlicense command to designate licenses as needed.
mmcrcluster: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

Component Overview



1) $> sudo mmcrcluster
1a) sudo logs that someguy ran mmcrcluster
1b) sudo runs mmcrcluster as root
2) root process mmcrcluster calls sshwrap.pl
To replicate mmcrcluster on other nodes
3) for each other node sshwrap.pl runs
“ssh someguy@nodex 'sudo /usr/lpp/mmfs/bin/mmremote'”
3a) sudo logs that someguy ran mmremote at remote node
3b) sudo runs mmremote which runs remaining
mmcrcluster work at remote node

Create a GPFS Cluster

- License the cluster members

```
$> sudo /usr/lpp/mmfs/bin/mmchlicense server -N all

The following nodes will be designated as possessing GPFS server licenses:
          fpstoc1a.fpet.pokprv.stglabs.ibm.com
          fpstoc1d.fpet.pokprv.stglabs.ibm.com
          fpstoc1c.fpet.pokprv.stglabs.ibm.com
          fpstoc1b.fpet.pokprv.stglabs.ibm.com
Please confirm that you accept the terms of the GPFS server Licensing Agreement.
The full text can be found at www.ibm.com/software/sla
Enter "yes" or "no": yes
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

- Start the daemons on all nodes

```
$> sudo /usr/lpp/mmfs/bin/mmstartup -a
Thu Feb 26 21:10:44 EST 2015: mmstartup: Starting GPFS ...
```

- Set GPFS to automatically start the daemons on IPL

```
$> sudo /usr/lpp/mmfs/bin/mmchconfig autoload=yes
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

Create a GPFS Cluster

- Use mmlscluster and mmgetstate to check what you've created

```
$> sudo /usr/lpp/mmfs/bin/mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         fpstoc1.fpet.pokprv.stglabs.ibm.com
  GPFS cluster id:           13170850555610780057
  GPFS UID domain:           fpstoc1.fpet.pokprv.stglabs.ibm.com
  Remote shell command:      /usr/lpp/mmfs/bin/sshwrap.pl
  Remote file copy command:  /usr/lpp/mmfs/bin/scpwrap.pl
  Repository type:           CCR

 Node  Daemon node name                          IP address    Admin node name                          Designation
-------------------------------------------------------------------------------------------------------------------
   1   fpstoc1a.fpet.pokprv.stglabs.ibm.com  10.20.80.246  fpstoc1a.fpet.pokprv.stglabs.ibm.com  quorum-manager
   2   fpstoc1c.fpet.pokprv.stglabs.ibm.com  10.20.80.248  fpstoc1c.fpet.pokprv.stglabs.ibm.com  quorum-manager
   3   fpstoc1d.fpet.pokprv.stglabs.ibm.com  10.20.80.249  fpstoc1d.fpet.pokprv.stglabs.ibm.com  quorum
   4   fpstoc1b.fpet.pokprv.stglabs.ibm.com  10.20.80.247  fpstoc1b.fpet.pokprv.stglabs.ibm.com

$> sudo /usr/lpp/mmfs/bin/mmgetstate -a

 Node number  Node name        GPFS state
------------------------------------------
       1      fpstoc1a         active
       2      fpstoc1c         active
       3      fpstoc1d         active
       4      fpstoc1b         active
```

- Look in /var/adm/ras/mmfs.log.latest to see what the deal is if all is not right

Create NSDs for the GPFS Cluster

- Create a NSD file

```
%nsd:     device=/dev/disk/by-id/dm-uuid-mpath-36005076303ffd4120000000000000f000
          nsd=NSD_DS8_F000
          servers=fpstoc1a,fpstoc1b,fpstoc1c,fpstoc1d
          usage=dataAndMetadata
%nsd:     device=/dev/disk/by-id/dm-uuid-mpath-36005076303ffd4120000000000000f100
          nsd=NSD_DS8_F100
          servers=fpstoc1a,fpstoc1b,fpstoc1c,fpstoc1d
          usage=dataAndMetadata
```

- Create the NSD volumes

```
$> sudo /usr/lpp/mmfs/bin/mmcrnsd -F /scratch/fpstoc1.nsdfile
mmcrnsd: Processing disk disk/by-id/dm-uuid-mpath-36005076303ffd4120000000000000f000
mmcrnsd: Processing disk disk/by-id/dm-uuid-mpath-36005076303ffd4120000000000000f100
mmcrnsd: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

Create a file system and mount it on the cluster

- Create a file system with mmcrfs using the NSDs that you just created

```
$> sudo /usr/lpp/mmfs/bin/mmcrfs gpfs0 "NSD_DS8_F000;NSD_DS8_F100" -T /storage0 -A yes

The following disks of gpfs0 will be formatted on node fpstoc1a:
    NSD_DS8_F000: size 524288 MB
    NSD_DS8_F100: size 524288 MB
Formatting file system ...
Disks up to size 4.4 TB can be added to storage pool system.
Creating Inode File
Creating Allocation Maps
Creating Log Files
Clearing Inode Allocation Map
Clearing Block Allocation Map
Formatting Allocation Map for storage pool system
Completed creation of file system /dev/gpfs0.
mmcrfs: Propagating the cluster configuration data to all
  affected nodes.  This is an asynchronous process.
```

- Then Mount it using mmmount

```
$> sudo /usr/lpp/mmfs/bin/mmmount all -a
Fri Feb 27 17:42:42 EST 2015: mmmount: Mounting file systems ...
```

And check your work

- Did it work?

```
$> df -m /storage0
Filesystem                    1M-blocks   Used Available Use% Mounted on
/dev/gpfs0                      1048576   2343   1046233   1% /storage0

$> sudo /usr/lpp/mmfs/bin/mmdf gpfs0
disk                disk size  failure holds    holds              free KB             free KB
name                    in KB    group metadata data        in full blocks        in fragments
--------------- ------------- -------- -------- ----- -------------------- -------------------
Disks in storage pool: system (Maximum disk size allowed is 4.0 TB)
NSD_DS8_F000        536870912       -1 Yes      Yes       535671040 (100%)           536 ( 0%)
NSD_DS8_F100        536870912       -1 Yes      Yes       535671296 (100%)           520 ( 0%)
                -------------                         -------------------- -------------------
(pool total)       1073741824                            1071342336 (100%)          1056 ( 0%)

                =============                         ==================== ===================
(total)            1073741824                            1071342336 (100%)          1056 ( 0%)

Inode Information
-----------------
Number of used inodes:            4038
Number of free inodes:          496058
Number of allocated inodes:     500096
Maximum number of inodes:      1048640
```

Manageability

- File system replication parameters can be changed at runtime
  - Use mmchfs then mmrestripefs to control how data is replicated
- The cluster and disks can also be scaled while I/O runs
  - Use mmadddisk to add additional NSD volumes to a file system
  - Use mmdeldisk && mmrestripefs to remove a volume and then juggle the metadata and data around to keep the proper number of replicas
  - No sub volume increments:
    - 1 SAN or DASD Volume or 1 Minidisk == 1 NSD for adding or removing from a file system
  - Use mmaddnode and mmdelnode to add and remove nodes from the cluster

Questions ?