Linux Lab:

GPFS
General Parallel FileSystem

Daniela Galetti
(System Management Group)

GPFS history

- Designed for SP cluster (power 3) with AIX
- Proprietary license

GPFS properties

- High filesystem performances
- Availability and recoverability
- Simple multinode administration
  (you can performe multinode command from any node in the cluster)

High filesystem performances

- Parallel accesses from multiple process of multimple nodes
  (trheaded daemon)
- Data striping across multiple disks and multiple nodes
- Client side data caching
- Ability to performe read-ahead and write-behind
- Optimized for high performances networks (myrinet)

Availability and Recoverability

- Distributed architecture: no single point of failure
- Automatical recover from nodes or disks failures
  - Multiple indipendent paths
  - Data and metadata replication NEW!
  - Monitoring of nodes status (heartbeat, peer domain)
  - Quorum definition (If the number of available nodes is less than Quorum number then the filesystem will be unmounted)

heartbeat and Quorum

- heartbeat tunable parameters:
  - frequency (period in seconds between two heartbeat)
  - sensitivity (number of missing heartbeat)
  - detection time = frequency*sensitivity*2
- default quorum definition:
  the minum number of nodes in the GPFS nodeset which must be running in order for GPFS daemon to start and for fs usage to continue

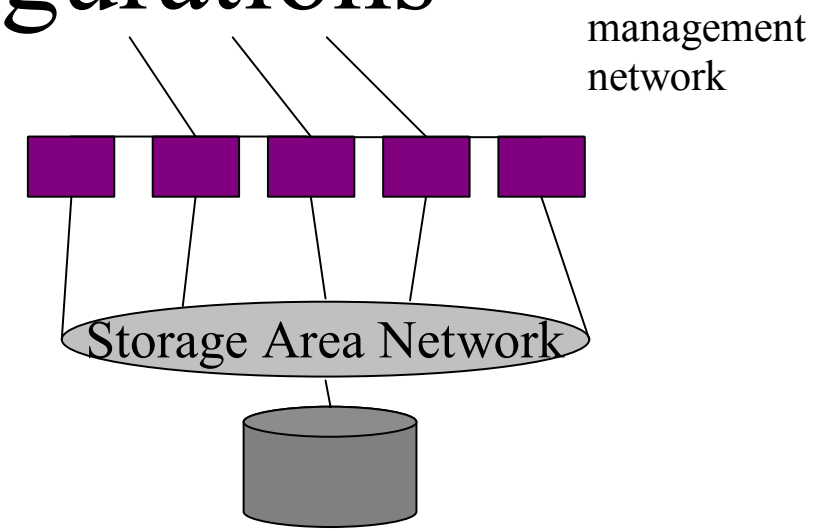
quorum = 50% + 1

NEW!

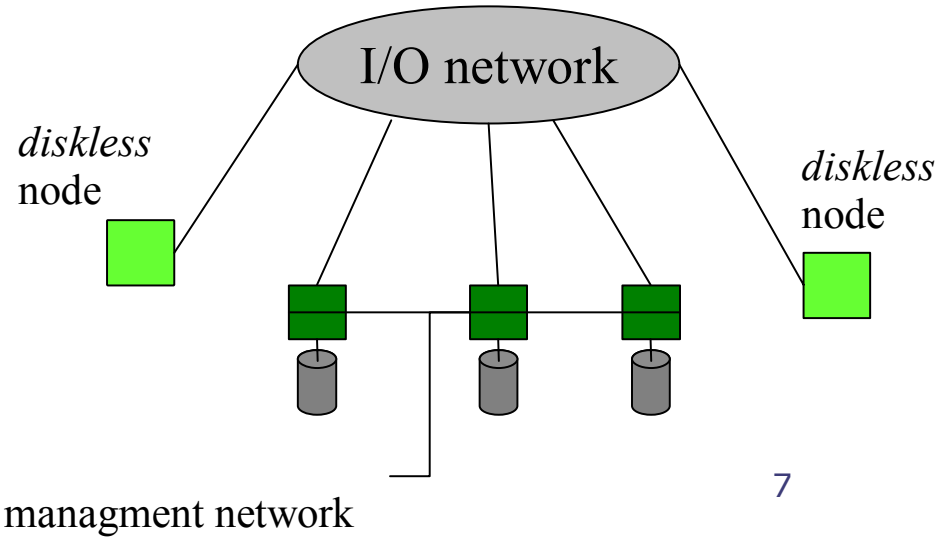
- customizable quorum:
  you may decide the pool of node from which quorum is derived

Two possible I/O configurations

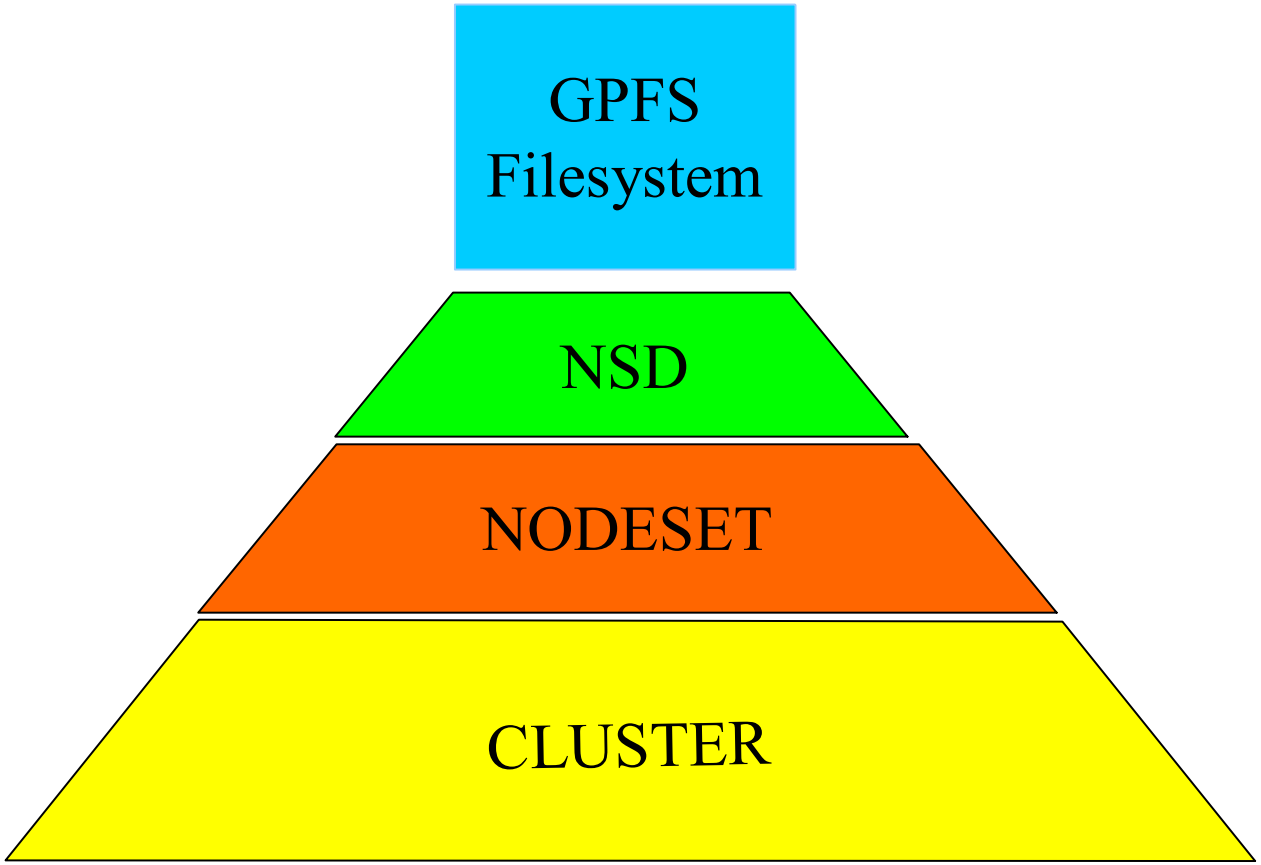
SAN model
each node that mount a GPFS fs must have a direct connection to the SAN



NSD model
a subset of the total node population is attached to disk drives. They are defined Network Shared Disks storage nodes

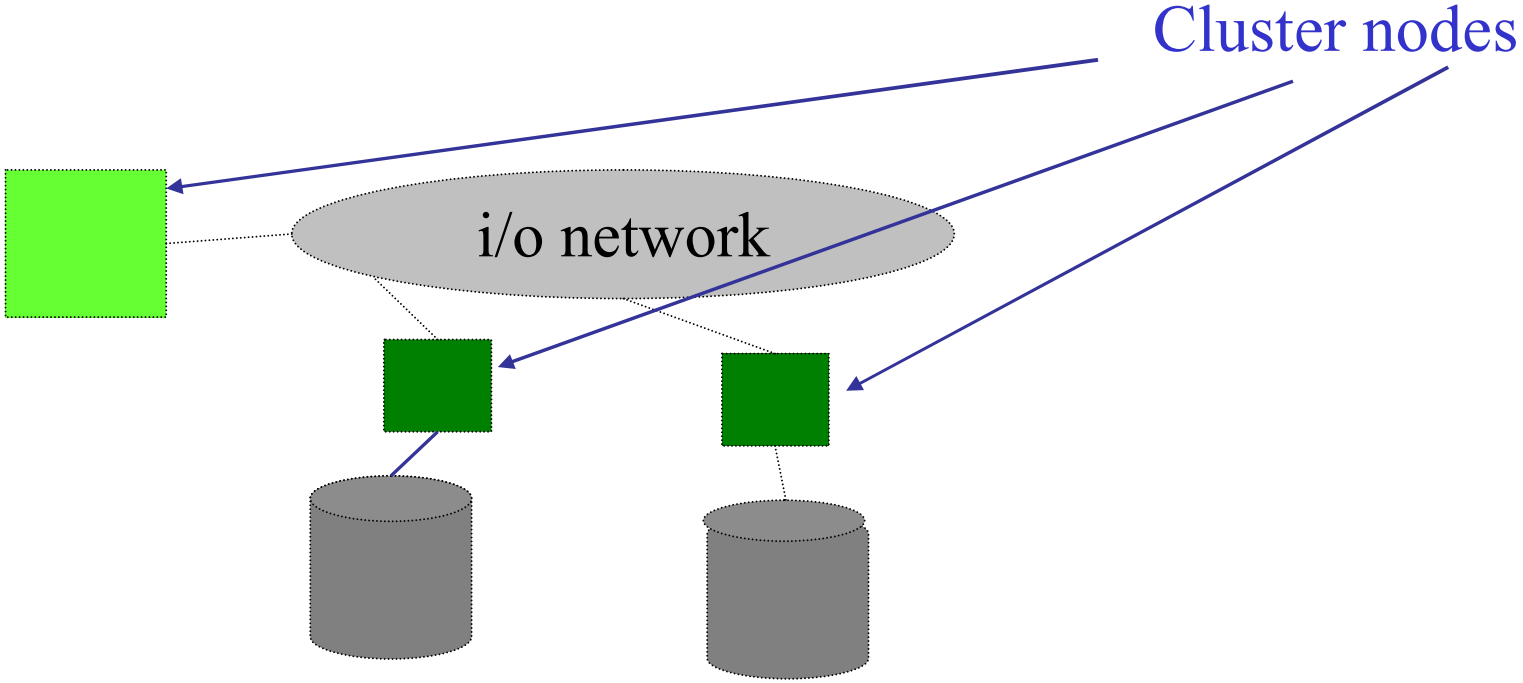


GPFS structure



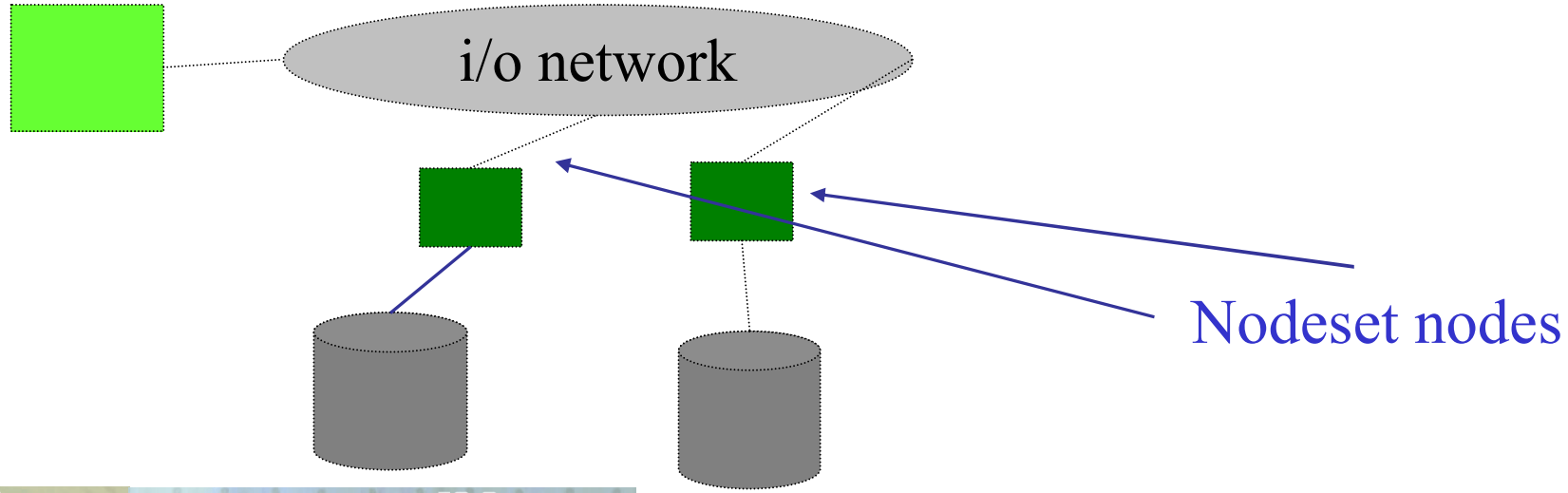
CLUSTER

the set of nodes over which GPFS is defined
these nodes can be directly attached to the disks or simply perform I/O access

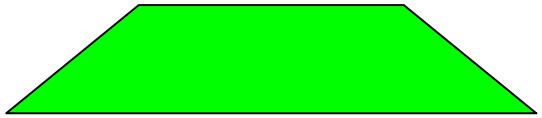


NODESET

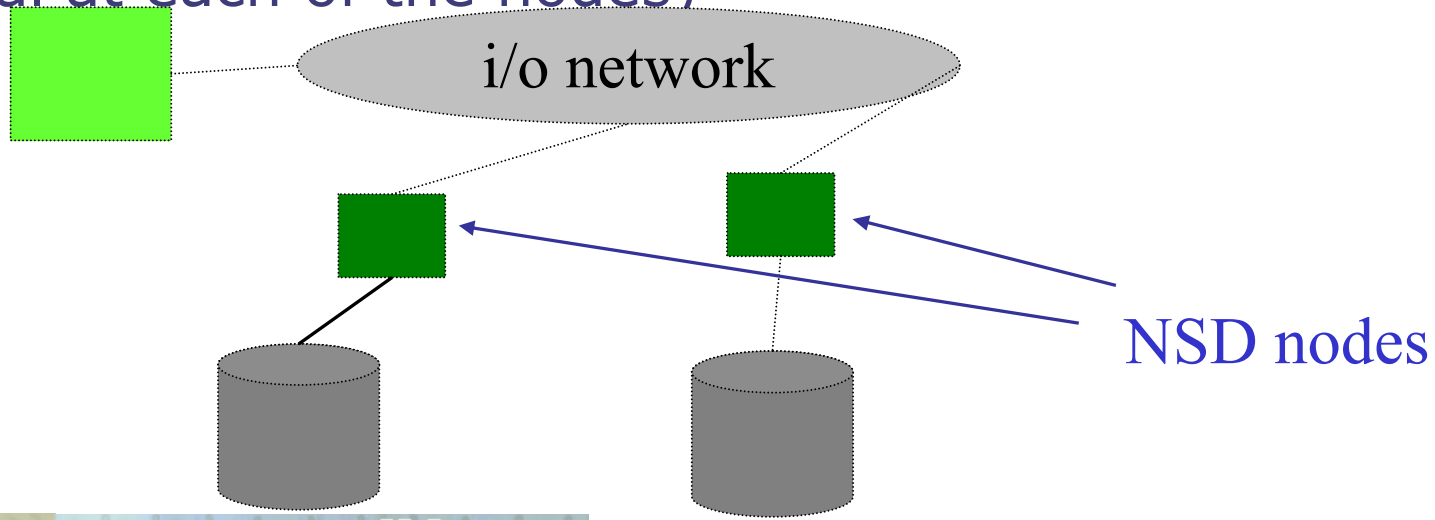
a group of nodes that give their disks to build the same filesystem
(It could be more than one nodeset in the samed GPFS cluster)



NSD



NSD- Network Shared Disks
the devices on which the filesystem is build, given by the nodes in the nodeset
(The GPFS function allows application programs executing at different nodes of a GPFS cluster to access a raw logical volume as if it were local at each of the nodes)



GPFS managers and servers functions

- ***GPFS cluster (primary and secondary) server***
  defined with mmcrcluster; server node for the GPFS configuration data (used to store the GPFS cluster data)

- ***Filesystem (configuration) manager (or client)***
  defined with mmconfig; it provides the following services for all of the nodes using the fs:
  - ***adding disks***
  - ***changing disks availability***
  - ***repairing fs***
  - ***controlling which region of disks are allocated to each node, allowing effective parallel allocation of space***
  - ***token and quota management***

GPFS managers and servers functions (cont.)

- ***GPFS (configuration) manager***
  one per nodeset. The oldest continuously operating node in the nodeset (defined by the Group Services). It choses the fs manager node.

- ***Failure group***
  a set of disks that shares a common point of failure that could cause them all to became unavailable (e.g. all disks that are either attached to the same adapter or NSD server)
  GPFS assure that two replicas of the same data or metadata will not be place in the same failure group.

Software architecture

- GPFS kernel extension:

translates standard filesystem calls from the operating system to gpfs filesystem calls

- GPFS daemon:

manages with the kernel extension the files lock

- open source portability layer:

interface between the GPFS and the linux kernel. It allows to not modify the GPFS installation changing kernel release

- heartbeat implementation:

if a node stops to send heartbeat signal to the server it will be fenced

GPFS for Linux at CINECA

Last successfully tested release: 2.2

scalable up to 512 nodes

requires SuSE SLES 8 SP3
(kernel 2.4.21)

GPFS release 2.2

rpm packages:

```
src-1.2.0.4-0.i386.rpm
rsct.core.utils-2.3.2.1-0.i386.rpm
rsct.core-2.3.2.1-0.i386.rpm
rsct.basic-2.3.2.1-0.i386.rpm
gpfs.base-2.2.0-1.i386.rpm
gpfs.gpl-2.2.0-1.noarch.rpm
gpfs.msg.en_US-2.2.0-1.noarch.rpm
gpfs.docs-2.2.0-1.noarch.rpm
```

Packages contents

`rsct`: Reliable Scalable Cluster Technology
heartbeat and reliability function

`gpfs`: installation and management
commands

`docs`: man pages and documentation

Hardware requirements

GPFS release	Server models	Max scalability
1.1.0	xSeries x330 x340	32 nodes
1.1.1	xSeries x330 x340 x342	32 nodes
1.2	xSeries x330 x340 x342 CLuster 1300	128 nodes
1.3	xSeries x330 x335 x340 x342 x345 Cluster 1300 Cluster 1350	256 nodes (512 on demand)
2.2	xSeries x330 x335 x340 x342 x345 x360 x440 Cluster 1300 Cluster 1350 Blade Center	512 nodes

Software requirements

GPFS release	Linux distribution	kernel version
1.1.0	RedHat 7.1	2.4.2-2
1.1	RedHat 7.1	2.4.2-2 2.4.3-12
1.2	RedHat 7.1	2.4.2-2 2.4.3-12 2.4.9-12
1.3	RedHat 7.2 RedHat 7.3 SUSE 7	2.4.9-34 2.4.18-5 2.4.18
2.2	Red Hat EL 3.0 Red Hat Pro 9 SuSE SLES 8.0	2.4.21-4* 2.4.20-24.9 2.4.21 (service pack 3)

*hugemem kernel that ships with RHEL 3.0 is incompatible with GPFS.

Peer Domain command 1/3

preprpnode: establish the initial trust between each node that will be in your peer domain (at least the quorum nodes).

```
node251:~ # preprpnode node251 node252 node253 node254
node252:~ # preprpnode node251 node252 node253 node254
node253:~ # preprpnode node251 node252 node253 node254
node254:~ # preprpnode node251 node252 node253 node254
```

Peer Domain command 2/3

mkrpdomain: establish the peer domain

```
node251:~ # mkrpdomain TestDomain node251 node252 node253 node254
```

lsrpdomain: displays peer domain information for the node

```
node251:~ # lsrpdomain
Name       OpState RSCTActiveVersion MixedVersions TSPort GSPort
TestDomain Offline  2.3.2.1           No            12347  12348
```

Peer Domain command 2/3

startrpdomain: brings the peer domain online

```
node251:~ # lsrpdomain
Name       OpState RSCTActiveVersion MixedVersions TSPort GSPort
TestDomain Online  2.3.2.1           No             12347  12348
```

lsrpnode
addrpnode
startrpnode
rmrpnode
rmrpdomain

GPFS Commands (1/4)

mmcrcluster: builds the cluster.
It defines the cluster server

```
mmcrcluster ... -p Primary server -s Secondary server
mmchcluster
mmdelcluster
mmlscluster
```

GPFS Commands (2/4)

mmconfig: defines the nodeset and the protocol type used on the I/O network

```
mmconfig ... -n NodeFile
mmchconfig
mmdelconfig
mmlsconfig

[root@node01 /root]# cat nodefile_config
node01.cineca.it:manager-quorum
node02.cineca.it:manager-quorum
node03.cineca.it:client-nonquorum
```

GPFS Commands (3/4)

mmcrnsd: formats the devices where the GPFS filesystem will reside

```
mmcrnsd -F nsdfile_out
mmchnsd
mmdelnsd
mmlsnsd
```

```
[root@node01 /root]# cat nsdfile_in
/dev/sda11:node01.cineca.it:node11.cineca.it::1
/dev/sda13:node02.cineca.it:::2
/dev/sda11:node03.cineca.it:::3
```

GPFS Commands (4/4)

mmcrfs: creates the filesystem.
It is possibile to define a quota limit

```
mmcrfs Mountpoint Device .... -B Blocksize
mmchfs
mmdelfs
```

PROs and CONs

- wide range of configuration
- high availability implementation
- easy installation
- a lot of good documentation

- not very cheap (academic license!)
- designed for high level servers

References

GPFS:

http://www-1.ibm.com/servers/eserver/clusters/software/gpfs.html

Open source portability layer:

http://oss.software.ibm.com/developerworks/projects/gpfs