IBM Systems and Technology

GPFS – A Cluster File System

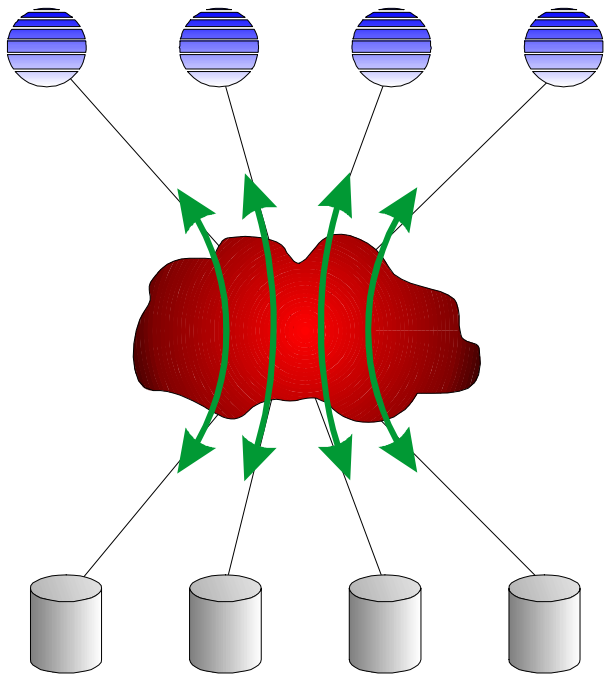
Klaus Gottschalk
gottschalk @de.ibm.com

NSC08
Linköping, 14.10.2008

GPFS: Parallel File Access

Parallel Cluster File System Based on Shared Disk (SAN) Model

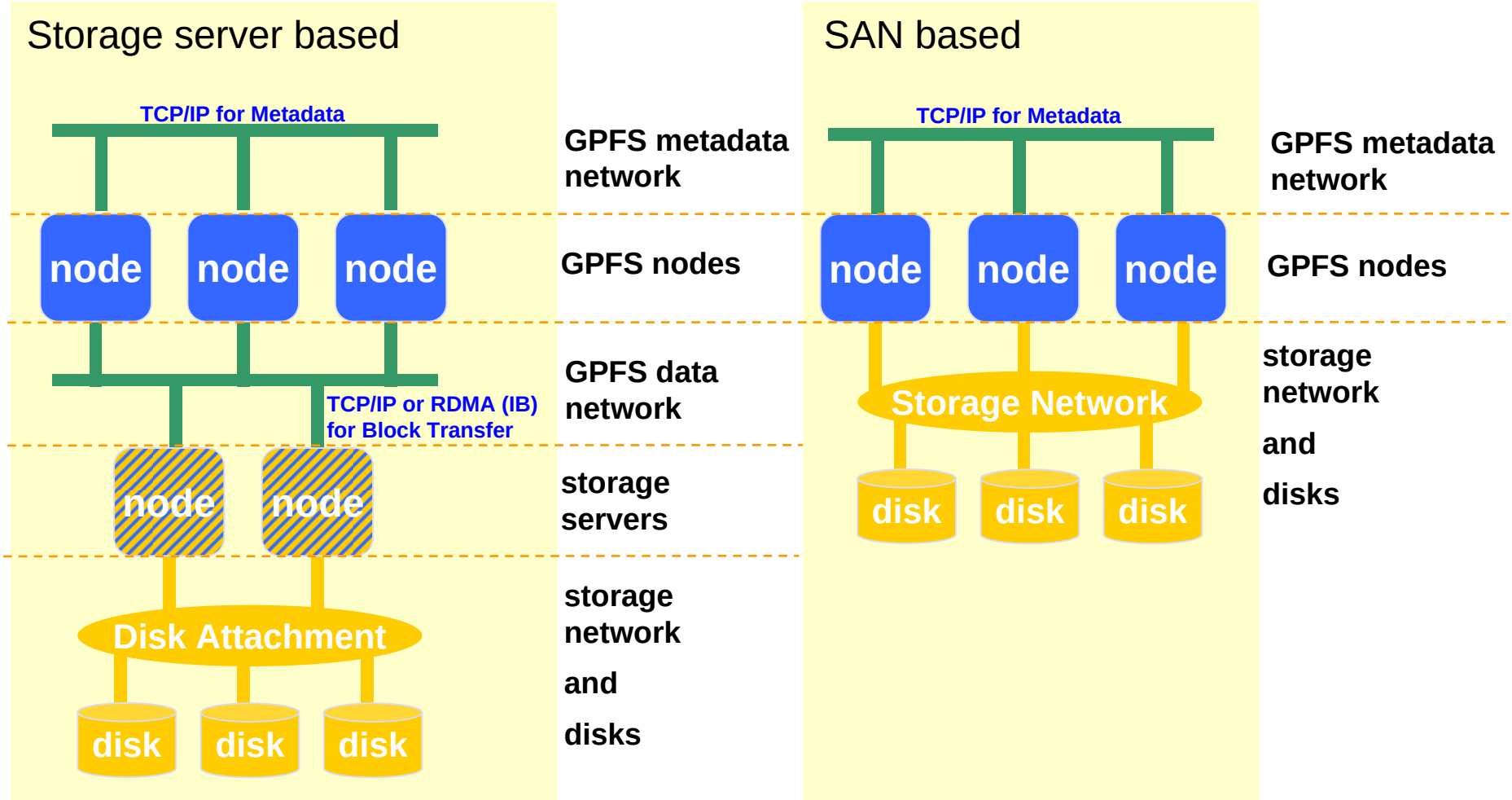
- *Cluster –* fabric-interconnected nodes (IP, SAN, …)
- *Shared disk -* all data and metadata on fabric-attached disk
- *Parallel -* data and metadata flows from all of the nodes to all of the disks in parallel under control of distributed lock manager.



GPFS Ideas

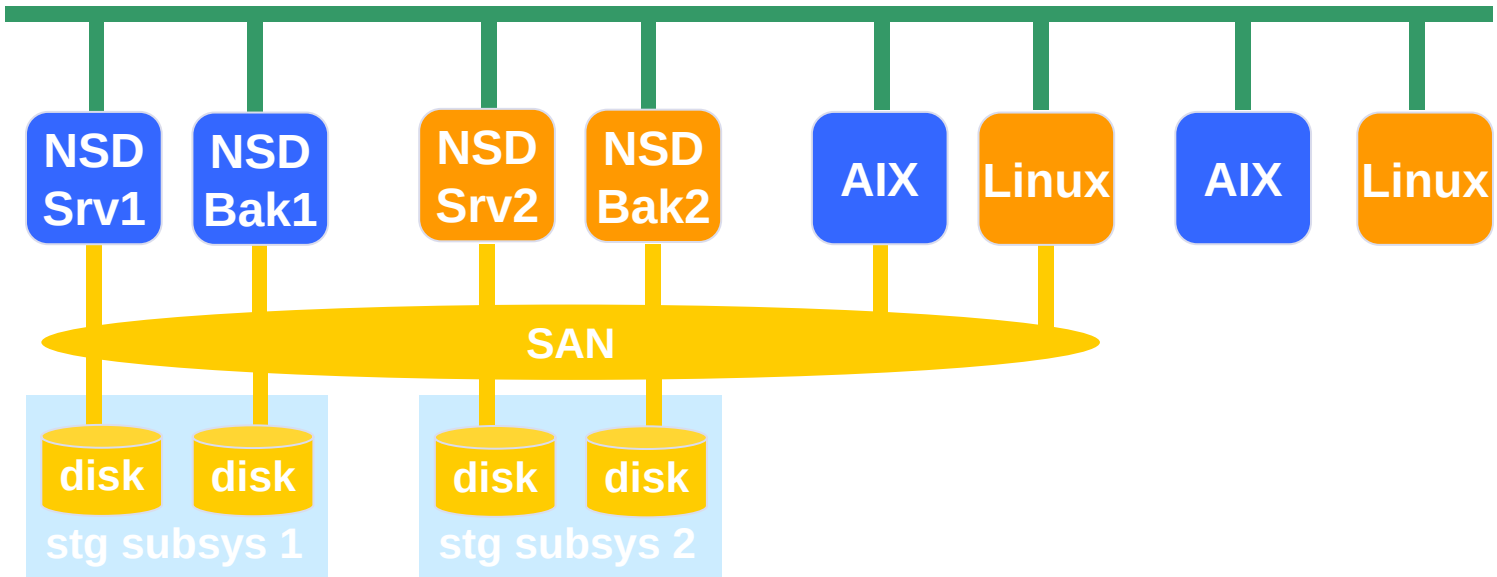
- A parallel Cluster File System for Multi Purposes with full POSIX compliance
  - HPC Cluster, Databases, Multimedia, Fileservers, …
  - Hardware independent, runs on the hardware you choose
- Parallel Scalability by “Scale Out” not “Scale Up”
  - Parallel Data- and Metadata Management, no central metadata server
  - Supports Clusters up to 2400 nodes
  - Aggregated IO Rates up to 100GB/s
- Heterogeneous Platform and Multi Cluster Configuration
  - Linux, AIX and Windows on POWER, x86 Hardware
  - Multi Cluster file sharing over WAN, i.e. DEISA, Terragrid
- Availability through Storage Hardware and use of HW Data Protection
  - No single point of failure
  - RAIDed LUNs as NSDs
- Online Reconfiguration of Cluster, Nodes, Disk, Data…
- A cluster is a administrative Unit with a trusted Sysadmin
- Information Lifecycle Management Functions as part of the File System

GPFS Storage Infrastructure

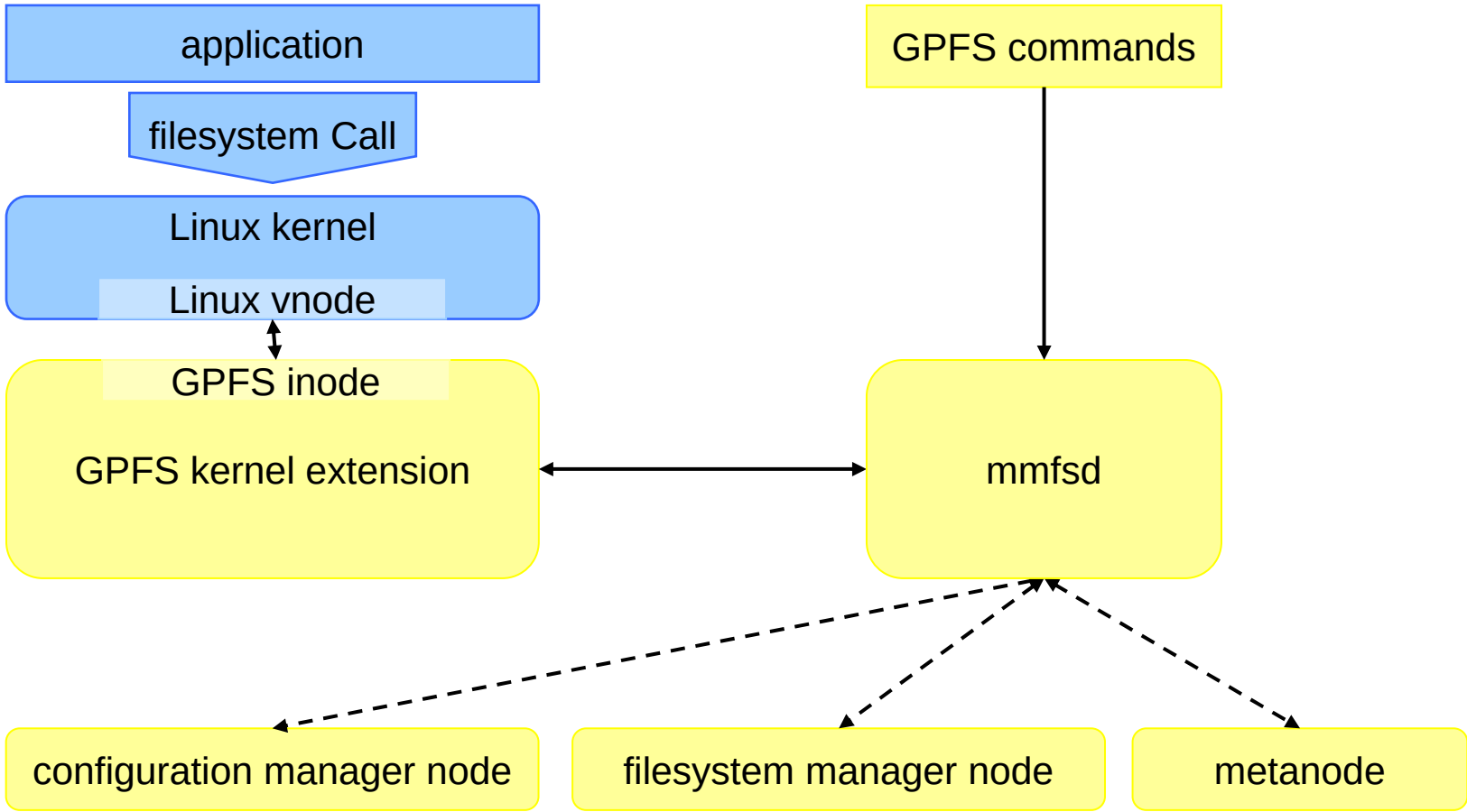


GPFS Storage Infrastructure

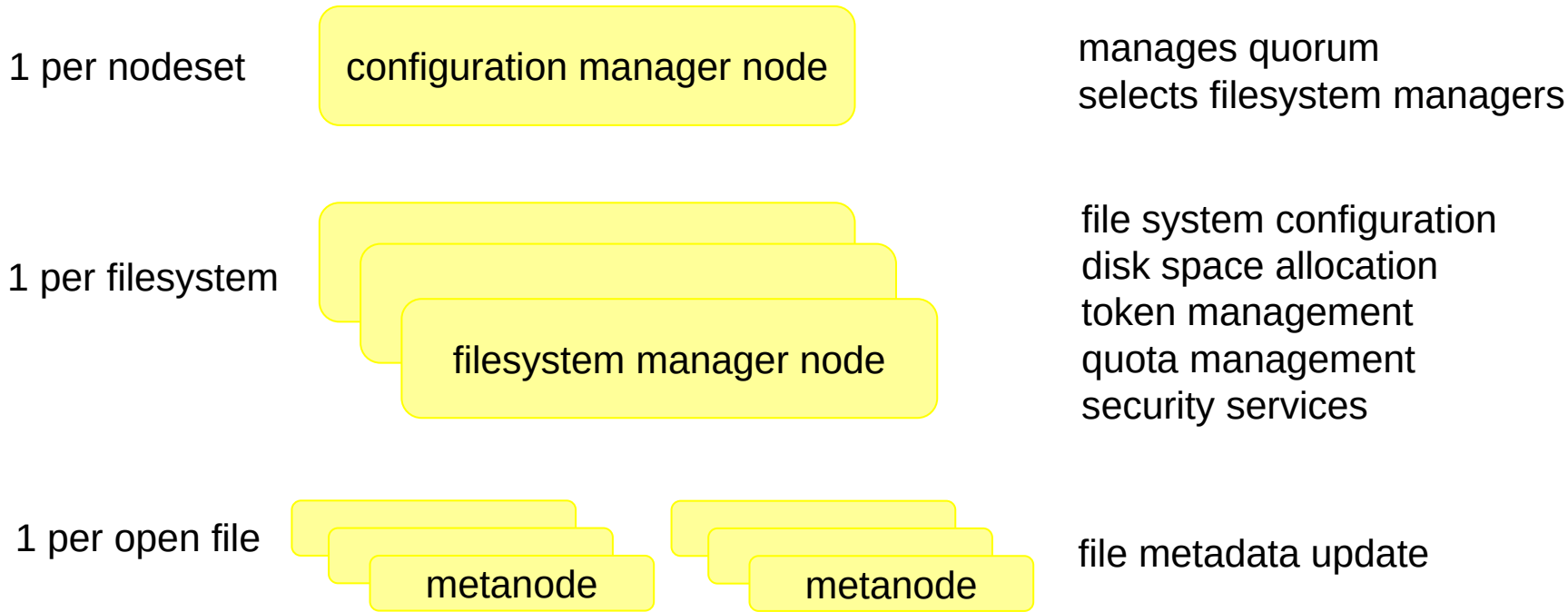
- GPFS requires TCP/IP communication between nodes
- GPFS allows a mixed storage infrastructure
- AIX and Linux nodes mixed in a single GPFS cluster
- Disk I/O accross SAN or Network Shared Disks (NSD)



GPFS Architecture



GPFS Architecture



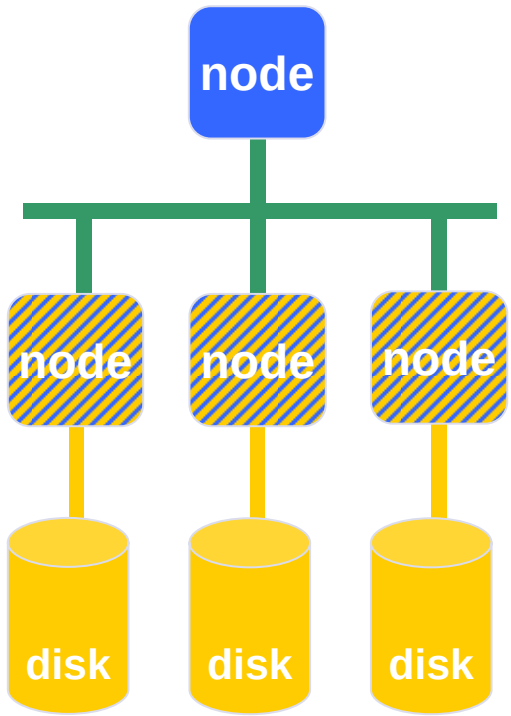
GPFS Concepts – File Striping

30 MB/s per job

GPFS storage client node

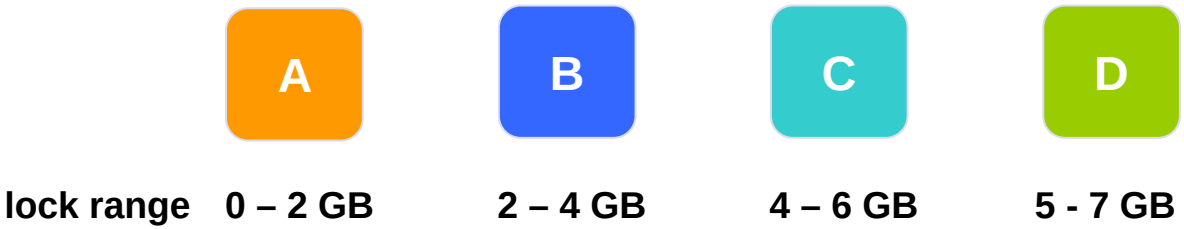
GPFS storage server node

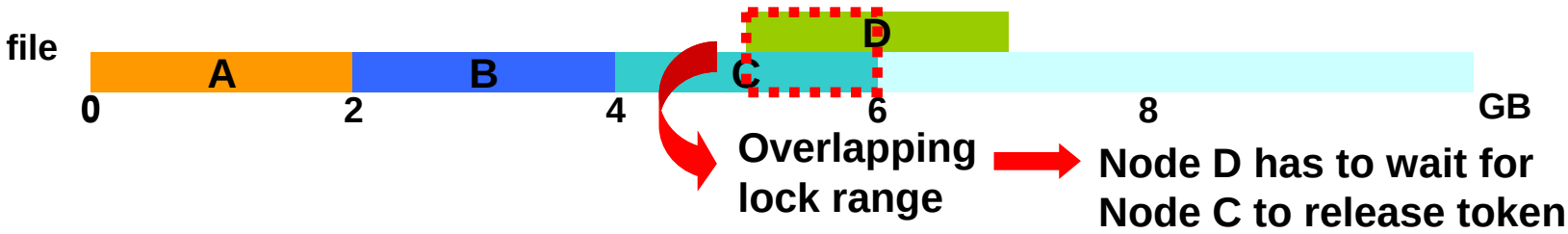
disk storage



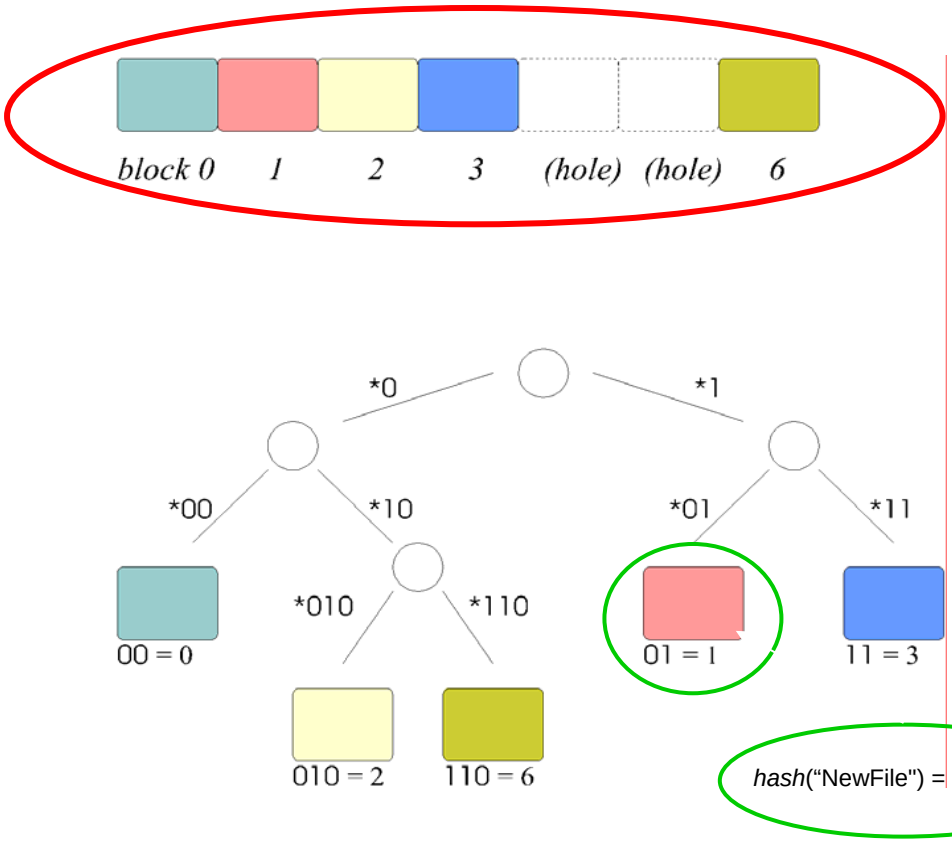
10 MB/s per disk

GPFS Concepts - Locking and Tokens





GPFS 3.2: Parallel file create



- File create previously locked the entire directory
  - Fine if files are in different directories
  - Not the best if they're in the same directory
  - Unfortunately, this is a natural way to organize files, and programmers are hard to re-educate

    Checkpoints, output for a time-step

- Optimization to improve write sharing
  - GPFS directories use extendible hashing to map file names to directory blocks

    Last n bits of hash value determine directory block number

    Example: *hash*("NewFile") = 10111101 means the directory entry goes in block 5
  - File create now only locks the hash value of the file name

    lock ensures against multiple nodes simultaneously creating the same file

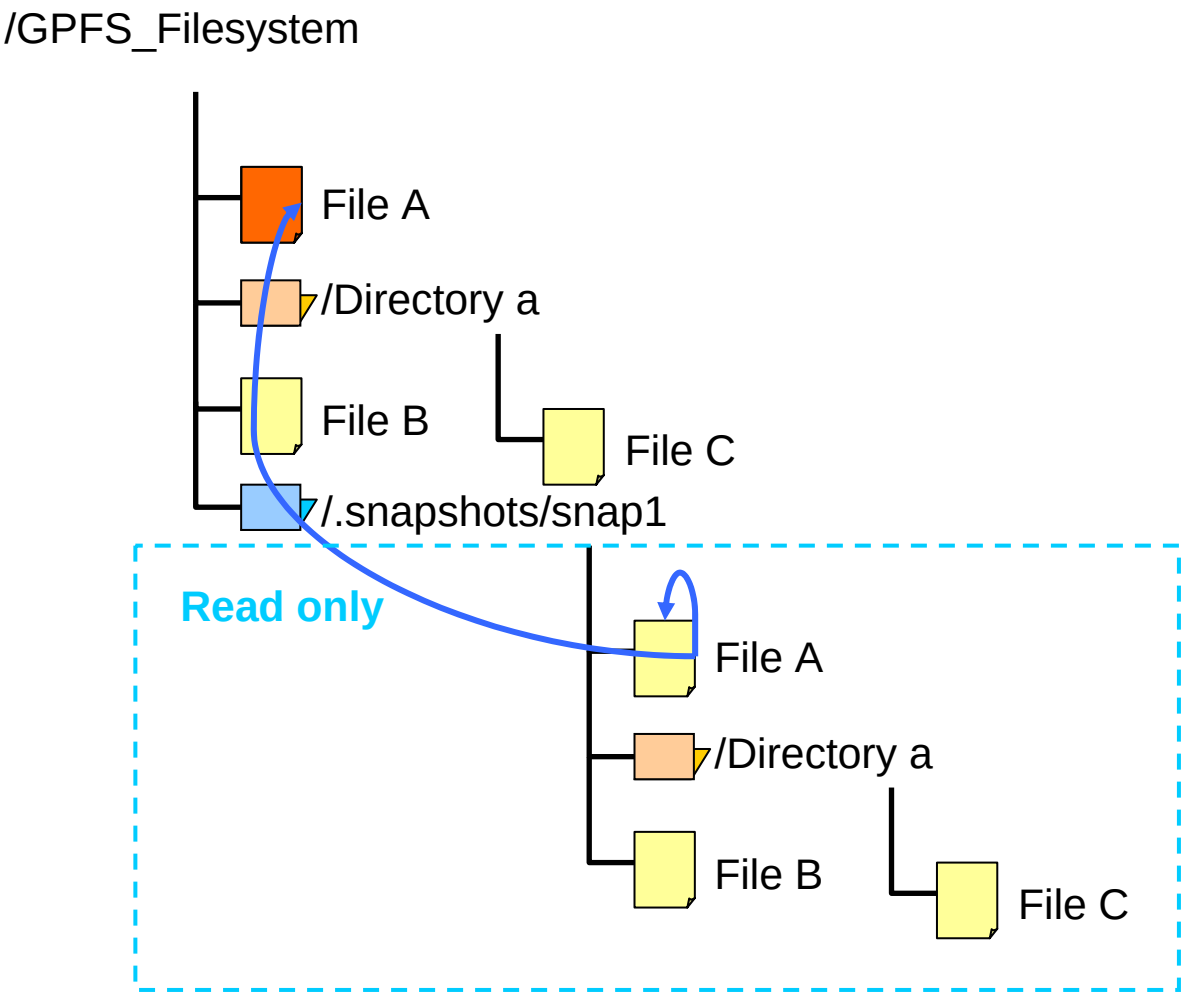
    actual update shipped to metanode

    If create requires directory block split, lock upgraded to cover both old and new block

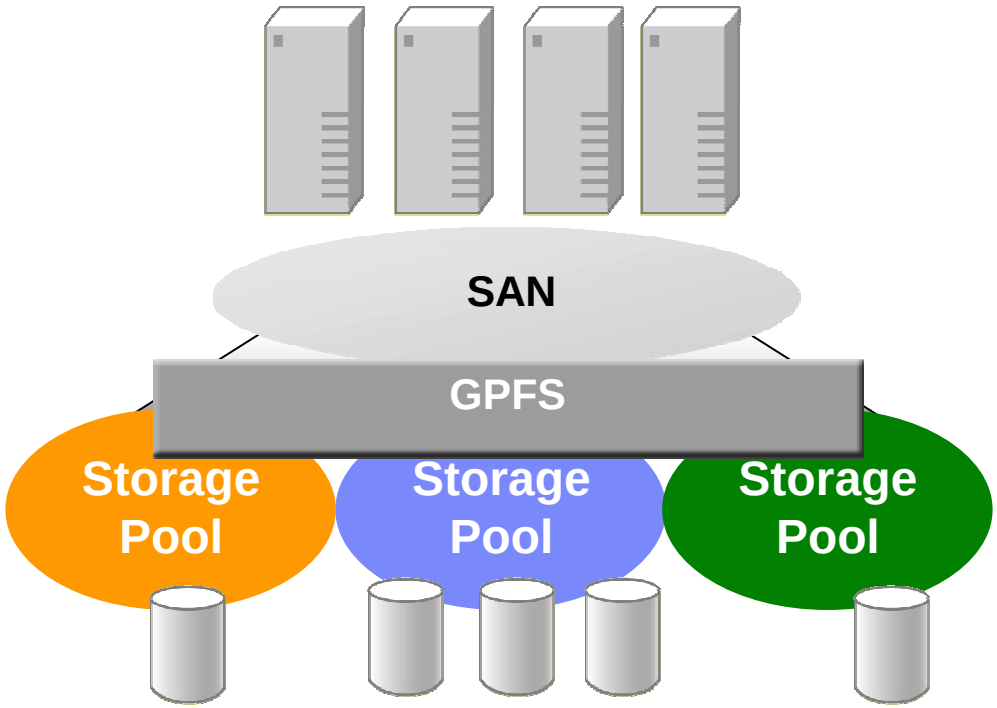
    Parallel file create performance no longer depends upon whether or not the files are in the same directory

    (except when splitting a block)

GPFS Concepts – File System Snapshot



GPFS 3.1 with NSD Pools



- One name space across all pools.

- Files in the same directory can be in different pools.

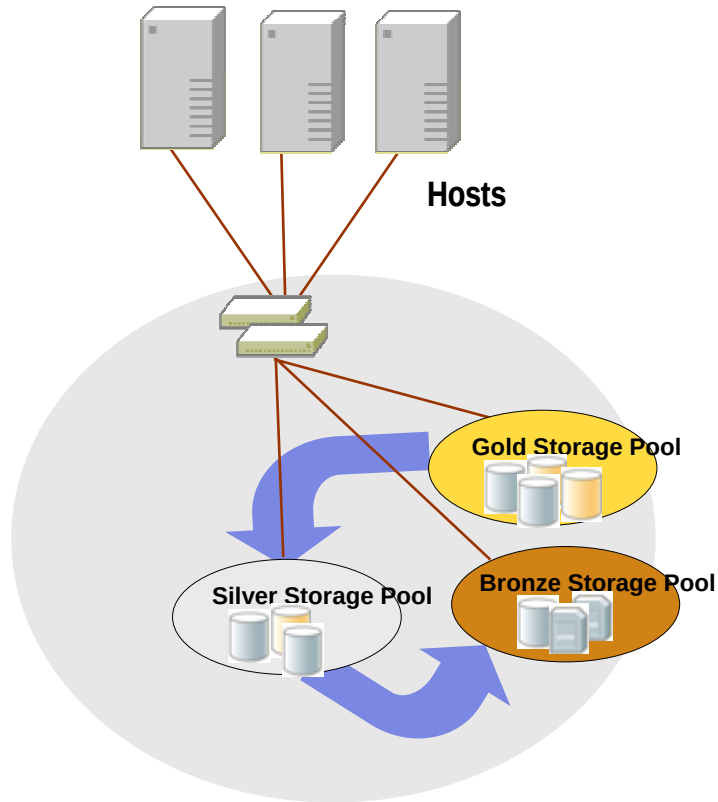
- Files placed in storage pools at create time using policies

- Files can be moved between pools for policy reasons

- Initial release allows 8 pools per file system

GPFS Tiered Storage Pools

- Storage pools are not inherently tiered. Tiered storage can be implemented by rules that specify criteria:
  - Move files not accessed in last 30 days from gold to silver
- Can specify migration through policies that moves data from one pool to another without interruption of service and without name change
  - Can be controlled by size, age, owner …



GPFS Filesets

- A fileset is a subset of the name tree which can be used for management purposes. It is possible to specify quotas on a fileset. It is also possible to use the fileset as a parameter to policies.
- Filesets can be unlinked and made invisible if desired. This may confused programs which scan file systems such as backup programs or HSM programs

GPFS Policy Implementation

- Allows creation of rules to specify:
  - Initial placement of files within storage pools
  - Migration of files to another GPFS storage pool
  - Automatic deletion of files
- Placement rules run at file creation time
- Migration rules run by command with the suggestion that they run as a cron job

Policy Set Execution Semantics

- Policies are defined in a policy set
  - Policy set is a series of ordered policy statements
  - Generally, each statement defines one specific action on a group of files
- Semantically, a policy set is implemented as follows:
  - Placement policies: they are evaluated, in order, until a match, for each file creation and the matched policy determines file placement (there is always a default placement policy)
  - Non-placement policies: For each file in the system, the action corresponding to the first matching policy is carried out (there may not always be a match)

Placement Policy

RULE rule-name SET POOL object-name

- [REPLICATE (data-repl-factor)]
- [FOR FILESET object-name,…]
- [WHERE {
- NAME LIKE string-pattern |
- UID = number | GID = number
- [AND [NOT] | OR [NOT] … }]
-

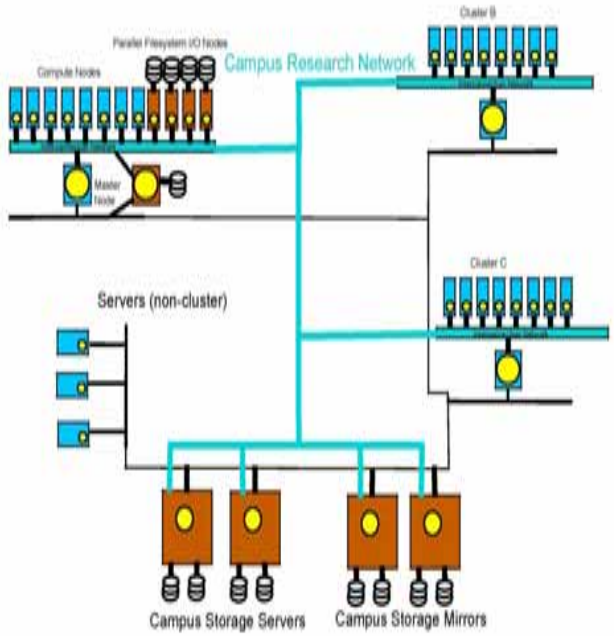
RULE mpg_policy SET POOL “scsi” WHERE NAME LIKE “bob”

GPFS Multi-clustering

Each FS belongs to one “owning” cluster, responsible for administration, lock management and recovery

“Remote” nodes can mount FS and request locks. They access data & metadata directly over the SAN/NSD connection

- Security capabilities exist where nodes authenticate themselves to the owning cluster.
- User ID’s in one cluster can be mapped to user IDs in the owning cluster.
- Performance largely a function of the network.



GPFS as Enterprise Wide File System

To enable GPFS as an Enterprise Wide File System, IBM has several initiatives to encourage GPFS support on all hardware platforms…

GPFS on Non IBM x86 and Power

- IBM GPFS is functionally capable on standard x86 and power hardware
- IBM provides support options on non IBM hardware and may require problem recreation on IBM hardware
- Vendors may include GPFS as part of their value-added solution

GPFS on unique hardware

- Limited Availability for GPFS Buildable Source License
- For unique hardware, IBM's GPFS IP is available under appropriate Licensing Agreement

GPFS 3.2 Platforms

- Linux (SuSE und Red Hat Enterprise Distibutions)
  - x86, x86_64
  - POWER5, POWER6
  - Blue Gene/L, Blue Gene/P
  - Itanium (on Request)
  - SGI (on Request)
- AIX
  - POWER5, POWER6, PPC970
- Future (SOD)
  - Microsoft Windows
  - Solaris

IBM Systems and Technology

Tutorial
GPFS Tuning and Best Practice

Klaus Gottschalk
gottschalk@de.ibm.com

NSC08
Linköping, 14.10.2008

GPFS Best Practices

- Make sure that TCP/IP Setup and DNS is clean and consistent on all Nodes
  - All nodes must be able to connect all other nodes with the GPFS node name
  - Don’t use /etc/host files. Do use DNS!
- Configure rsh/ssh for password-less command access for root between all GPFS cluster nodes
  - Required for cluster configuration and automatic recovery
- Use Storage Subsystems LUNs as NSD
  - Redundant Controllers and Disk Paths
  - Redundancy by hardware scales better
  - RAID1, RAID5 or RAID6 LUNs
- Configure redundant Paths to NSD and share NSD servers
  - Primary and Backup NSD Servers
  - Dual Active NSD Servers in multiple Networks supported with GPFS V3.2
- Match Disk-Array and GPFS File System Blocksize
  - RAID5 Array 128K stripe size, 4+P Drives => GPFS blocksize 512K
  - See example on next foils
- Switch Cache Mirroring and Write Cache off
  - Cache Mirroring between Controls result in Performance Penalty

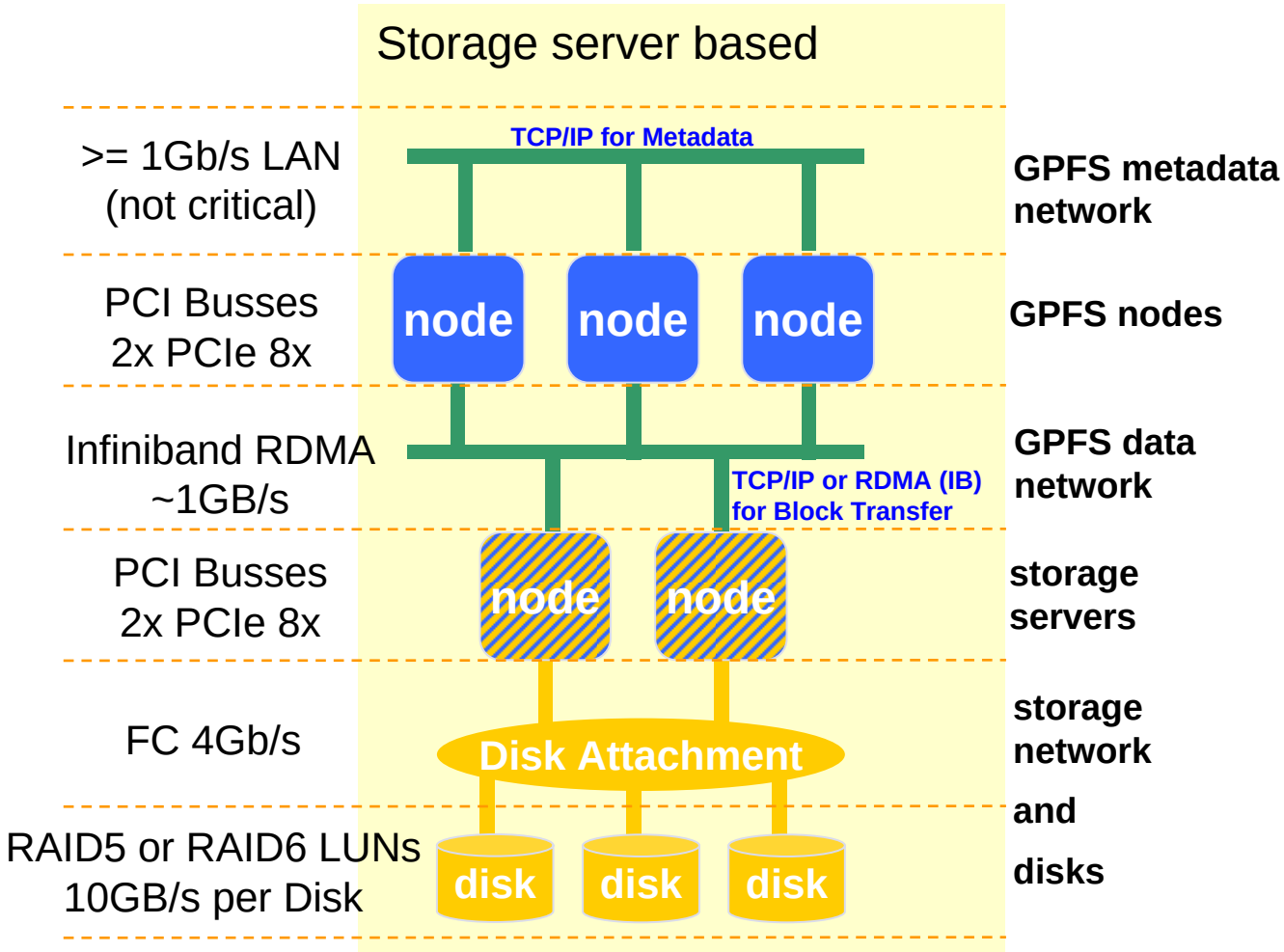
GPFS Best Practices

- Consider to Separate Data and Metadata
  - Use different Disk Types for dedicate usage
  - Metadata: fast Disks, RAID1
  - Data: streaming Disks, RAID5, RAID6 Tune GPFS Node Memory Usage
  - Inode Cache
  - Page Pool
  - Socket Buffers
- Parameter maxMBpS limits Throughput per Client
  - See mmchconfig, Default 150MB/s
- Consider GPFS Quorum und Plan Quorum Nodes
- Unmount of GPFS most likely caused by
  - Quorum lost due to missed heart beats
  - Memory exhaustion by applications
  - See /var/mmfs/gen/mmfslog on each node
- GPFS Related Redbooks
  - http://www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=gpfs
- Use Snapshots and mmbackup for Backup of large Cluster file systems

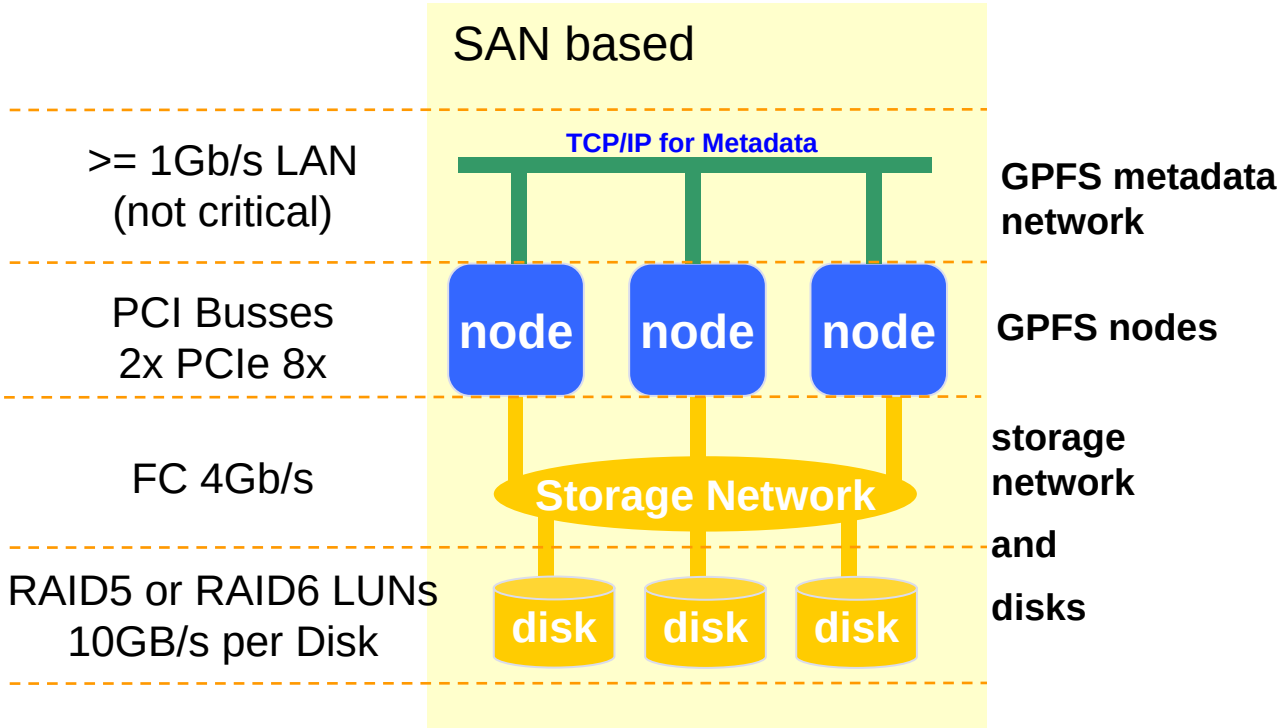
Tuning GPFS

- Measure your LUN performance prior to configure GPFS
  - Single LUN
  - Scalability Tests
  - Storage Subsystem
- Estimate the GPFS performance prior to configure the cluster
  - Single Client Throughput
  - Aggregated Throughput
- You can expect from GPFS …
  - Nearly full channel bandwidth
  - Nearly ideal scaling
- Try to identify potential performance bottlenecks of your setup

IO Throughput Factors



IO Throughput Factors



Example script to estimate LUN performance:

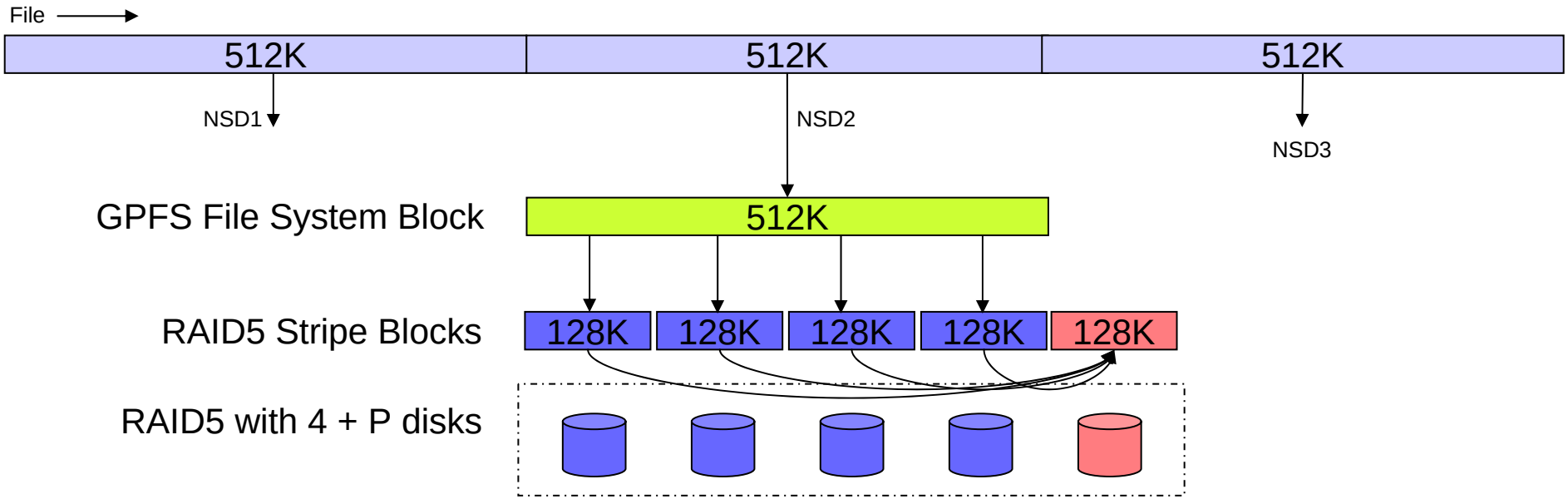
Caution: this will destoy all data on disk!!

```
#!/bin/bash
BS="1m"
CT="1000"
DISKS ="sda sdb sdc sdd"

for i in $DISKS; do
    dd if=/dev/zero of=/dev/$i bs=$BS count=$CT &
done
wait
```

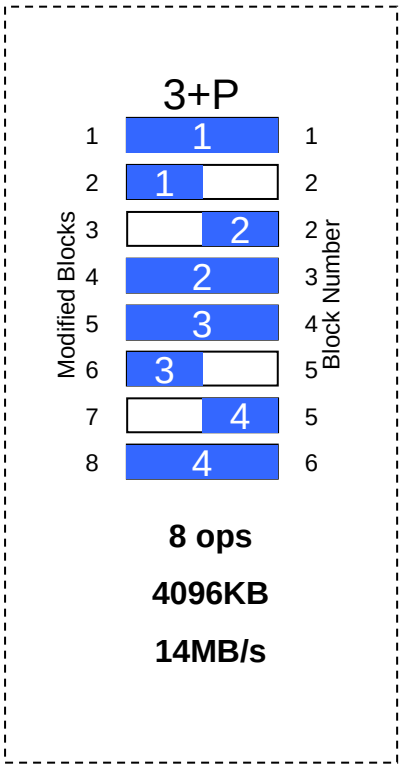
GPFS Blocksize and RAID Stripesize

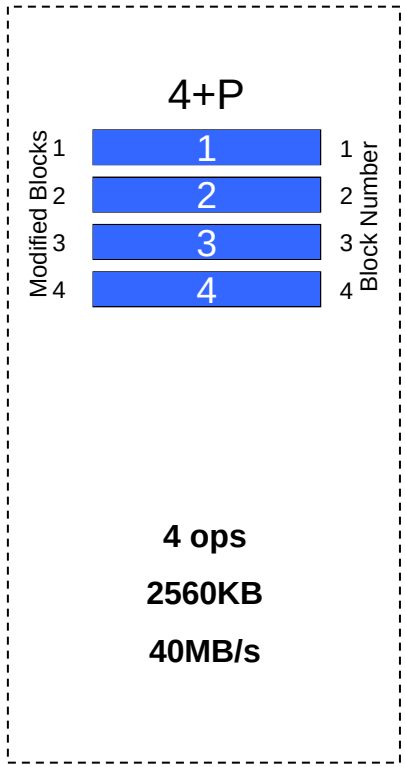
- GPFS Filesystem Blocksize should be a full stride on each NSD
- GPFS RAID LUNs should always consist of 2^n data drives + Parity drives
- 4+P, 8+P are common choices
- Example 2MB file, 512KB GPFS Block size, RAID5

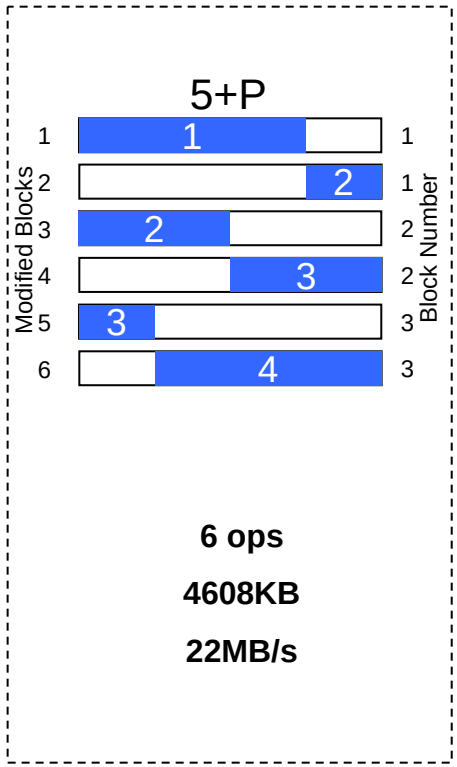


GPFS Performance Setup

- Performance Impact of different LUN sizes
- GPFS 512K Block Size, 2MB File (4 GPFS Blocks)

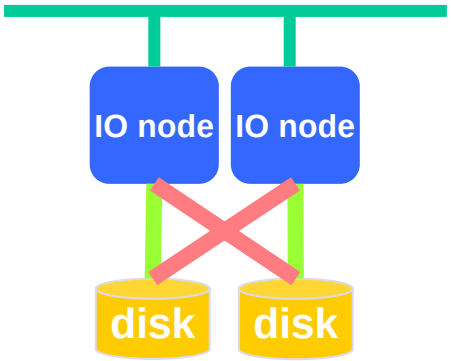




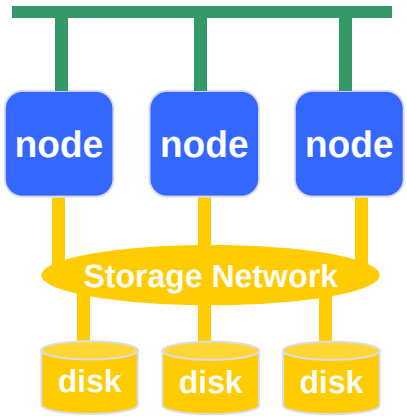


Storage Subsystem IO	3+P	4+P	5+P
IO Operations	**8 ops**	**4 ops**	**6 ops**
Parity & IO	**4096KB**	**2560KB**	**4608KB**
Perf. Estimate	**14MB/s**	**40MB/s**	**22MB/s**

Plan for redundant NSD Servers



- Use RAIDed LUN for Hardware Redundancy
  - Use dedicated LUNs in Production File Systems
  - RAID5 or RAID6 for Data or Mixed NSDs
  - RAID1 for Metadata LUNs
- Attach NSDs to two servers
  - Primary and Backup NSD Server
  - Automatic Takeover will ensure availability



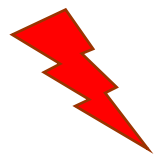
- For high Metadata IO rates
  - Separate data and metadata
  - Give nodes direct NSD access

GPFS Concepts - Node Quorum

- GPFS quorum = ½ * (number of nodes) +1

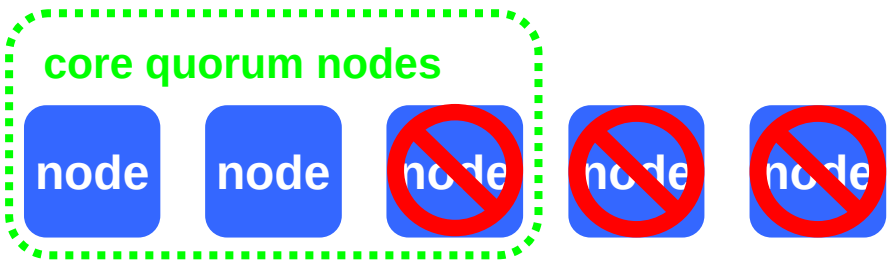
quorum is met ⇨
filesystems available

quorum not met ⇨
filesystems NOT available!!!

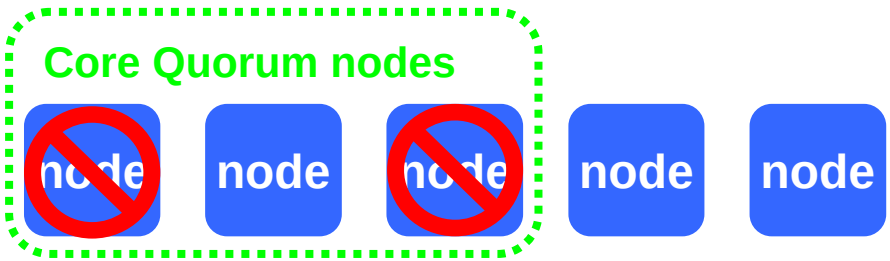


GPFS Concepts - Core Node Quorum

- GPFS 2.2 and later
- ONLY core quorum nodes have to be member of RSCT domain
- GPFS quorum = ½ * (number of core quorum nodes) +1



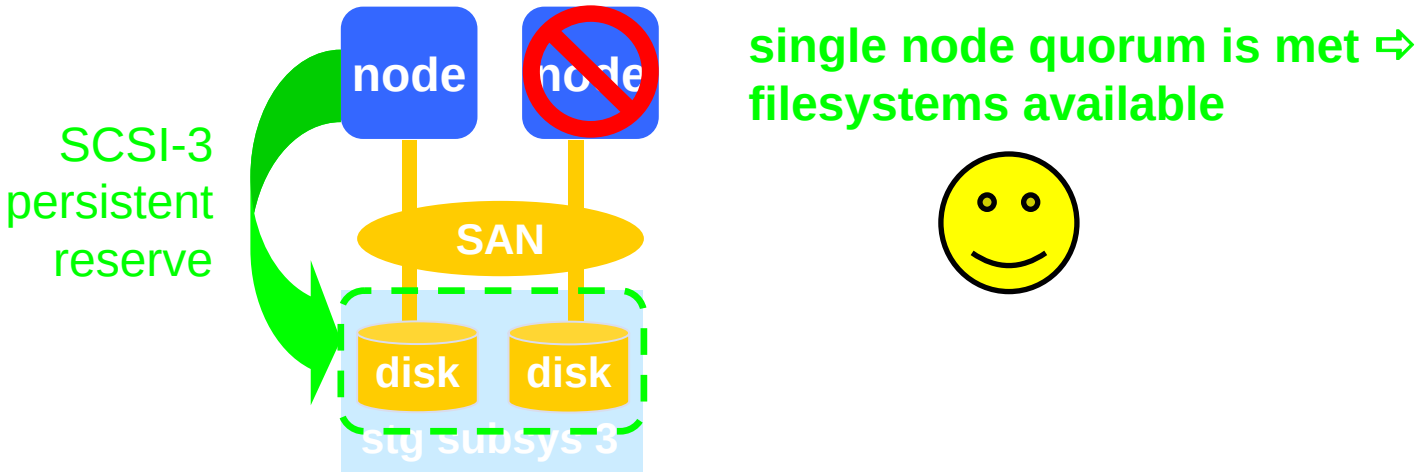
quorum is met ⇨
filesystems available



quorum not met ⇨
filesystems NOT available!!!

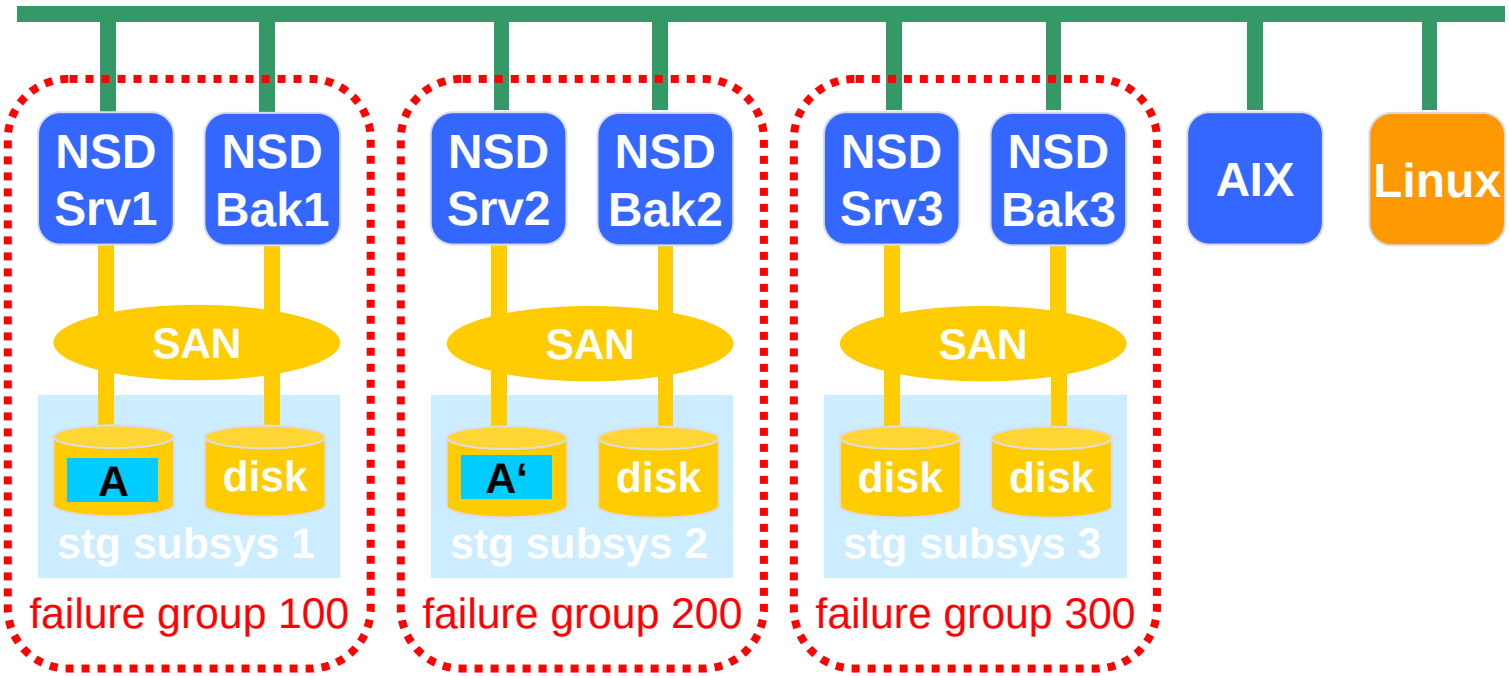
GPFS Concepts – Single Node Quorum

- 2-node clusters!
- Disk fencing based on SCSI-3 persistent reserve
- Restricted to SSA and ESS with MPIO/SDDPCM



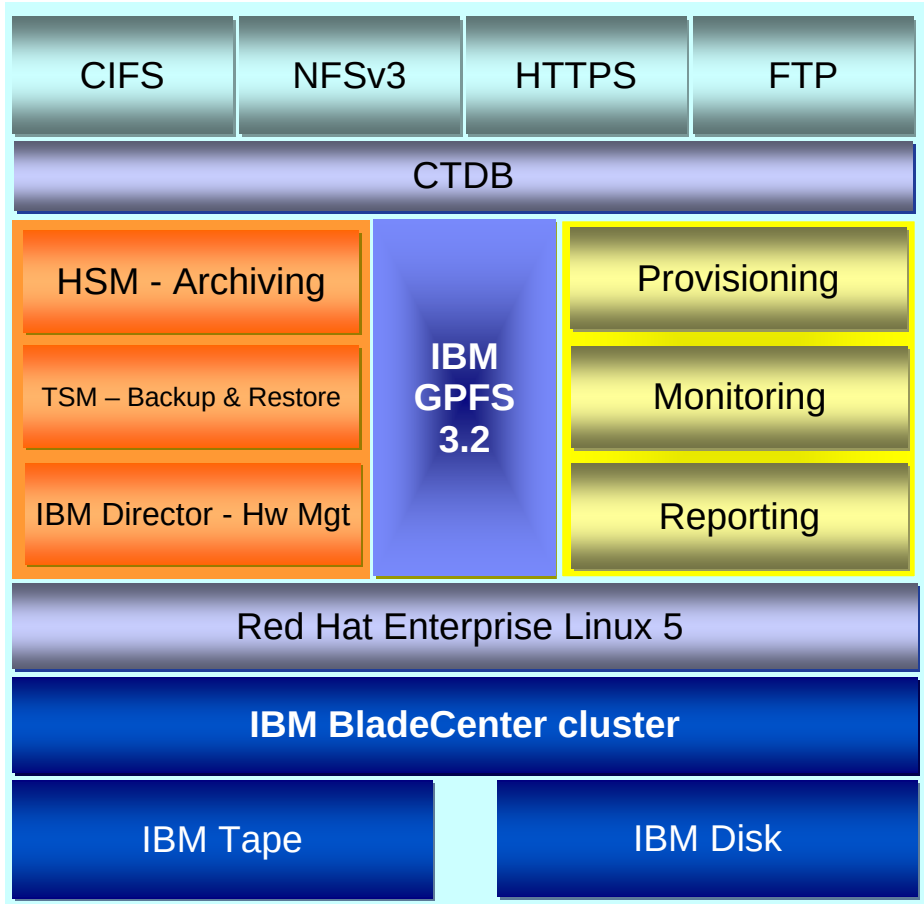
GPFS Concepts – File Replication (mirroring)

- Data and/or metadata may be replicated (mirrored)
- Replication can be activated on a file level
- Replication mirror copies are placed in different failure groups



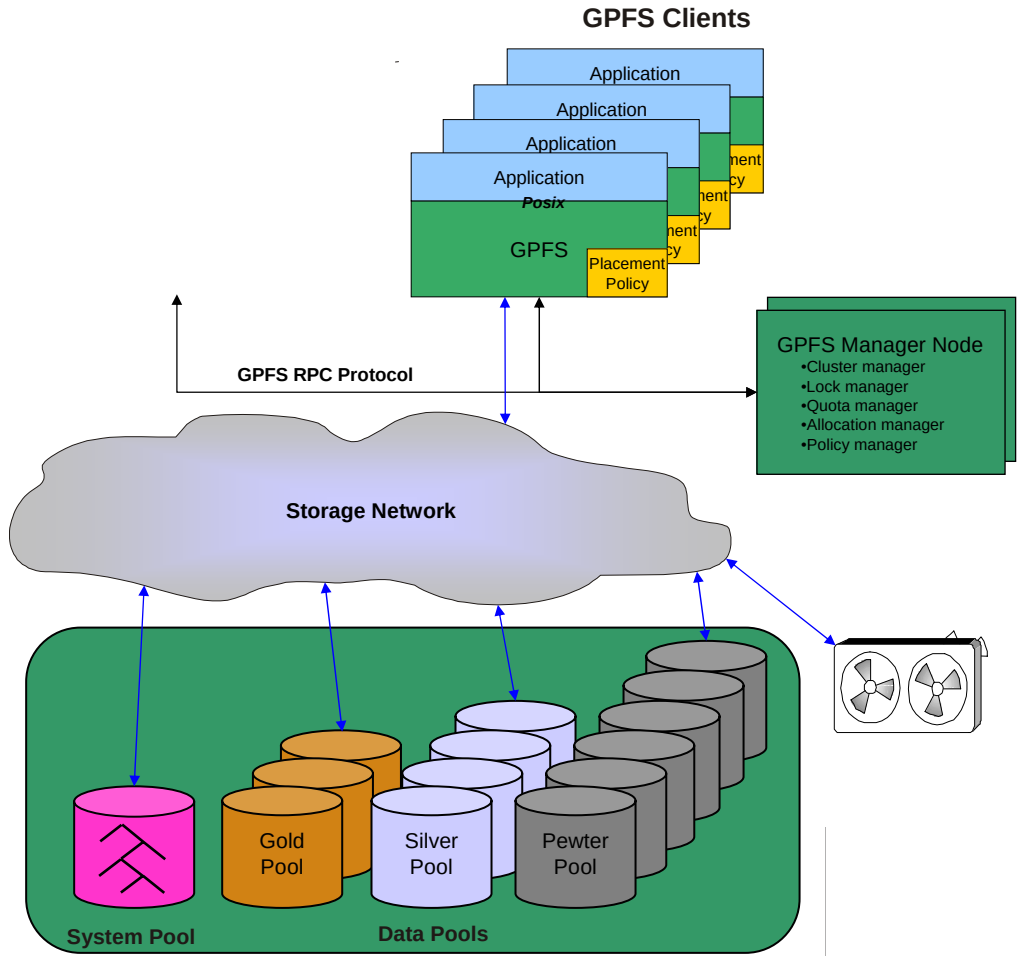
SoFS extends GPFS

- GPFS 3.2 – IBM's clustered file system for HPC environment
- CTDB – Controls the cluster and the file service daemons
- Samba – Provides CIFS export
- RHEL5 – Base OS, provides NFS, FTP and HTTP daemons
- SoFS Package – Provides Mangement GUI, Apache file server module, tools, etc.
- IBM Hardware

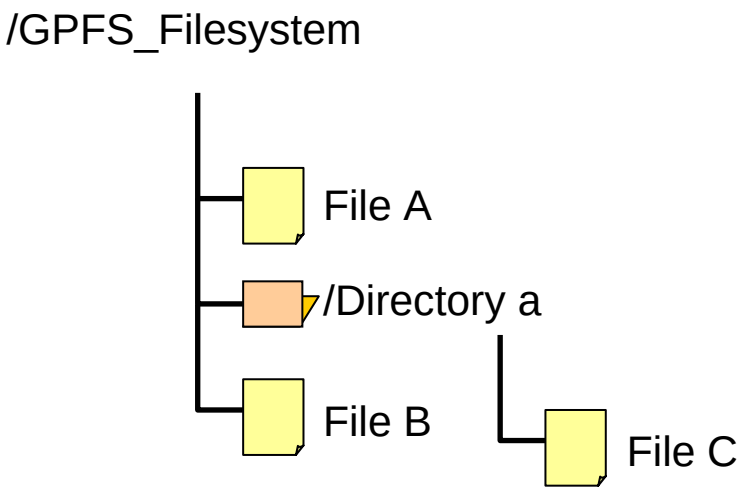


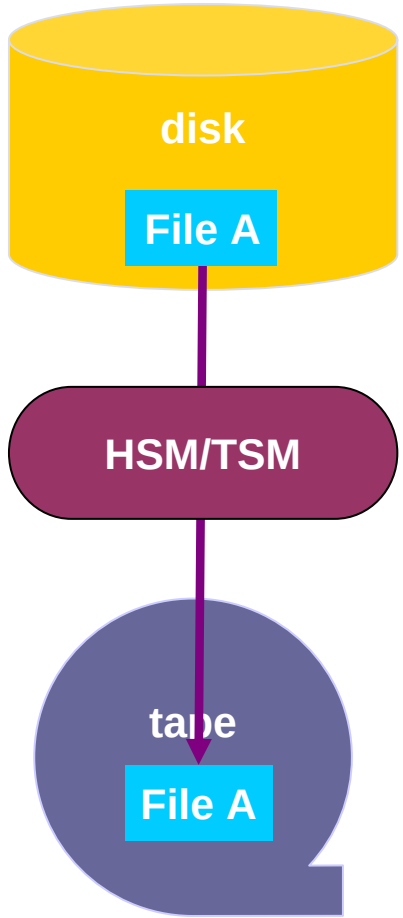
GPFS Information Lifecycle Management

- GPFS supports Information Lifecycle Management (ILM) via three new abstractions: storage pools, filesets, and policies
  - Storage pool – group of LUNs
  - Fileset: subtree of a file system
  - Policy – rule for assigning files to storage pools
- Types of policy rules
  - **Placement**, e.g. place database files on RAID1 storage, place other data files on RAID6
  - **Migration**, e.g. move project files to SATA storage after 30 days, to tape after 60 days
  - **Deletion**, e..g delete scratch files after 7 days



GPFS Concepts – „Indefinite“ File System Size





GPFS - ILM Integration with HSM

- The idea: Integrate GPFS Policies with HSM
- The advantages:
  - Integration of disk-to-disk and disk-to-tape into fully-tiered storage
  - Finer control of data movement across storage tiers
  - More efficient scans and data movement using internal GPFS functions
  - Possibility of coalescing small disk files into large tape files
- GPFS/HPSS Integrated ILM Technology Demo at SC06
  - New list-oriented interface to HSM system (move list of files to/from disk)
  - HPSS prototype drives parallel data movement to tape over this interface

Compile GPFS Compatibility Layer

- Required for Linux hardware/OS pairs
- Available in source code under GPL

- Prerequisites
  - A node with a full C/C++ development environment
  - Kernel source code installed

- Please read /usr/lpp/mmfs/src/README

- Edit /usr/lpp/mmfs/src/config/site.mcr
- # export SHARKCLONEROOT=/usr/lpp/mmfs/src
- # make World
- # make InstallImages

- Test with mmstartup && dmesg
- Install the four built images on all nodes

GPFS Configuration Steps

- Please see GPFS Installation and Planning Guide
- # mmcrcluster – Setup of the GPFS Cluster
  - mmaddnode / mmdelnode / mmchconfig
- # mmcrnsd – Formats a NSD and
  creates a cluster wide name for a disk
- # mmcrfs – Creates a GPFS file system
  - mmchfs / mmadddisk / mmdeldisk / mmrestripefs
- # mmstartup – Start GPFS on a node / on a cluster
  - mmshutdown

GPFS settings not changeable later

- Global GPFS Cluster
  - GPFS Clustername
- GPFS Filesystem
  - Filesystem Blocksize
  - Filesystem maximum Number of Data- and Metadata Replikas
  - Expected Number of Nodes Mounting the file system
- For other settins see mmchcluster, mmchfs commands and „GPFS Advanced Administration Guide“ for change Procedures

GPFS Literature

- IBM Documentation for GPFS
  - Concepts, Planning, and Installation Guide, GA76-0413
  - Administration and Programming Reference, SA23-2221
  - Advanced Administration Guide, SC23-5182
  - Problem Determination Guide, GA76-0415
  - Data Management API Guide, GA76-0414
- GPFS Frequently Asked Questions
  - Limits, Tested Hardware, Performance Estimates, Hints & Tips
- IBM Redbooks on GPFS
  - http://www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=gpfs
- All available as PDF or HTML from the IBM Website
  (just google for „GPFS FAQ“)