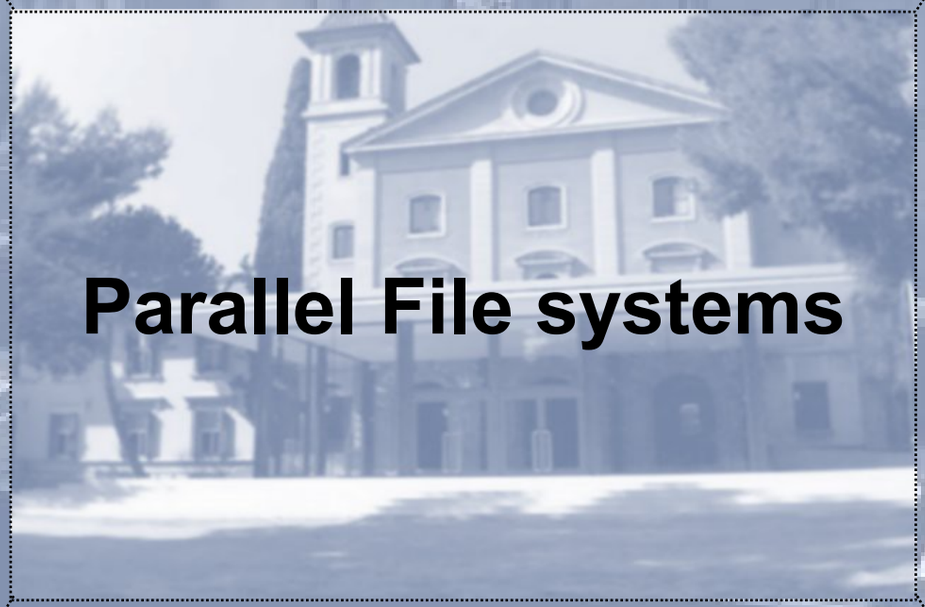
Barcelona Supercomputing Center
Centro Nacional de Supercomputación



Parallel File systems

RES User Seminar of Parallel Computing
Santander, Nov 05/11/2010

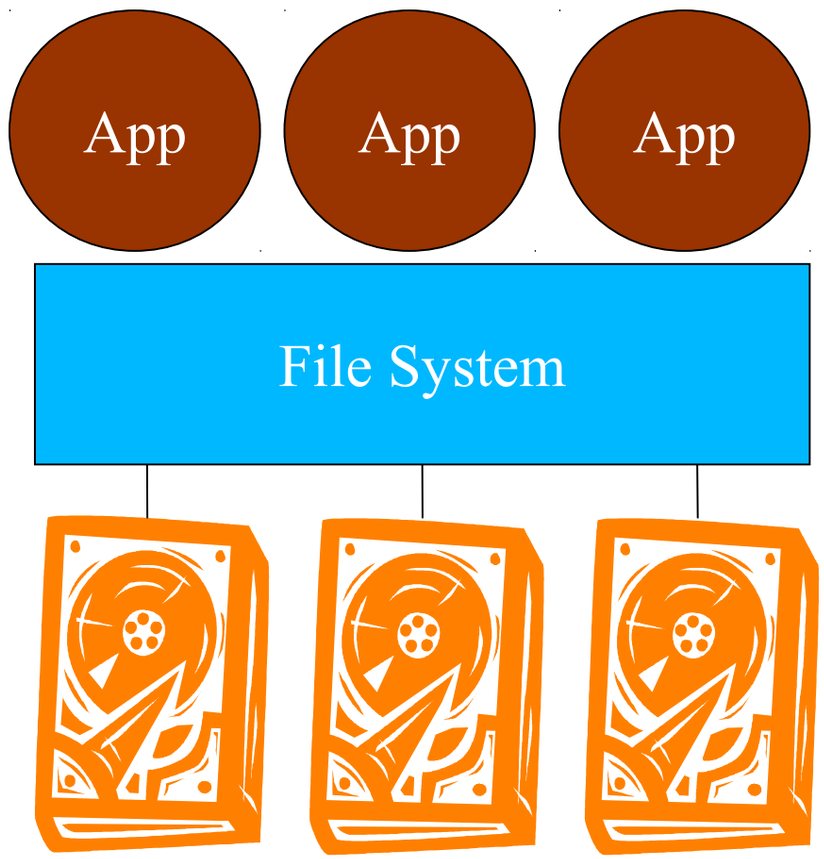
Sergi Moré
BSC - Sistemas

Índice

- Introducción.
- Lustre
- Panasas
- GPFS
  - Arquitectura
  - Roles
  - Bloques
  - Cache
  - Gestión de Locks
  - Multicluster
  - Storage Pools
  - FileSet
  - Recomendaciones
  - GPFS MareNostrum
  - GPFS MareNostrum Futuro

Conceptos - Sistema de Ficheros

- Interfaz de acceso a ficheros / directorios
- Coordinación del acceso a datos y metadatos
- Gestión de los medios físicos
- Responsable de mantener la integridad de los datos
- Tipos de sistemas de ficheros:
  - Sistemas de ficheros locales
  - Sistemas de ficheros distribuidos
  - Sistemas de ficheros paralelos

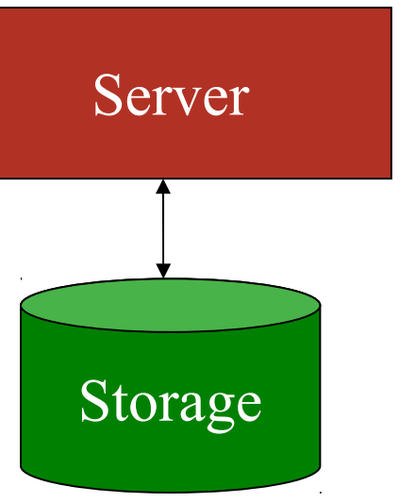


Conceptos – Requerimientos Storage HPC

- Escalable
    - En clientes
    - En espacio
- Alto rendimiento
- Queremos hacer input /output con multiples tasks:
    - Sin tener que buscar el disco/fichero de donde leer/escribir.
    - Sin preocuparnos de si alguien mas tiene el fichero abierto.
    - Sin preocuparnos de si alguien mas esta modificando ese fichero.
    - Sin preocuparnos por la consistencia de los datos.
    - Con un código portable (POSIX).

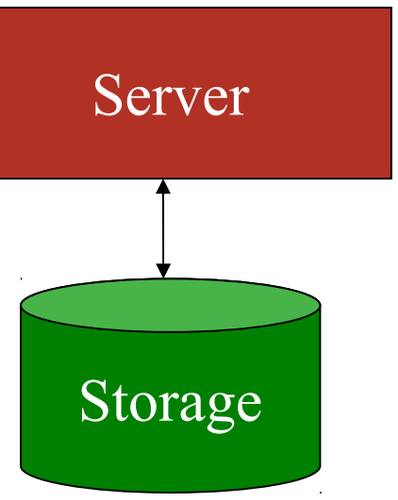
Conceptos - Sistema de ficheros local

- Generalmente integrados en el SO.
- POSIX.
- Limitadas formas de paralelismo:
  - Entre procesos locales.
  - A nivel de disco (Stripping).
- Ejemplos: ext4, NTFS, ReiserFS, XFS,….



Conceptos - Sistema de ficheros local

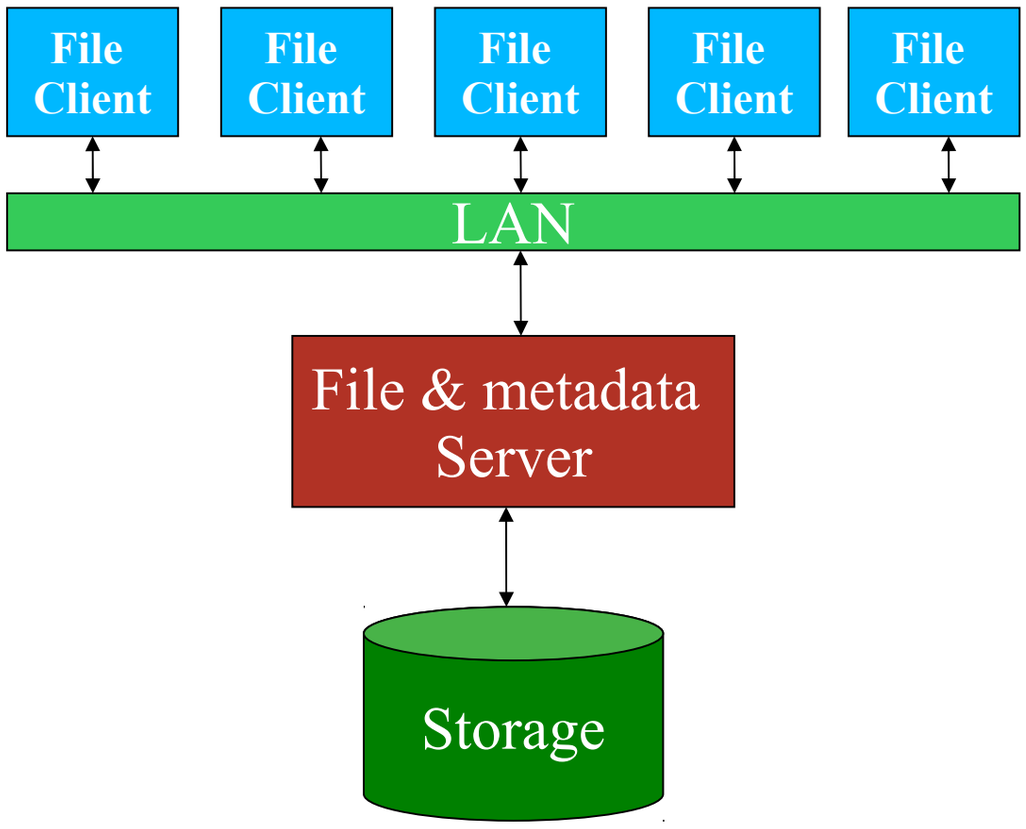
- Generalmente integrados en el SO.
- POSIX.
- Limitadas formas de paralelismo:
  - Entre procesos locales.
  - A nivel de disco (Stripping).
- Ejemplos: ext4, NTFS, ReiserFS, XFS,….



No es escalable en clientes!!

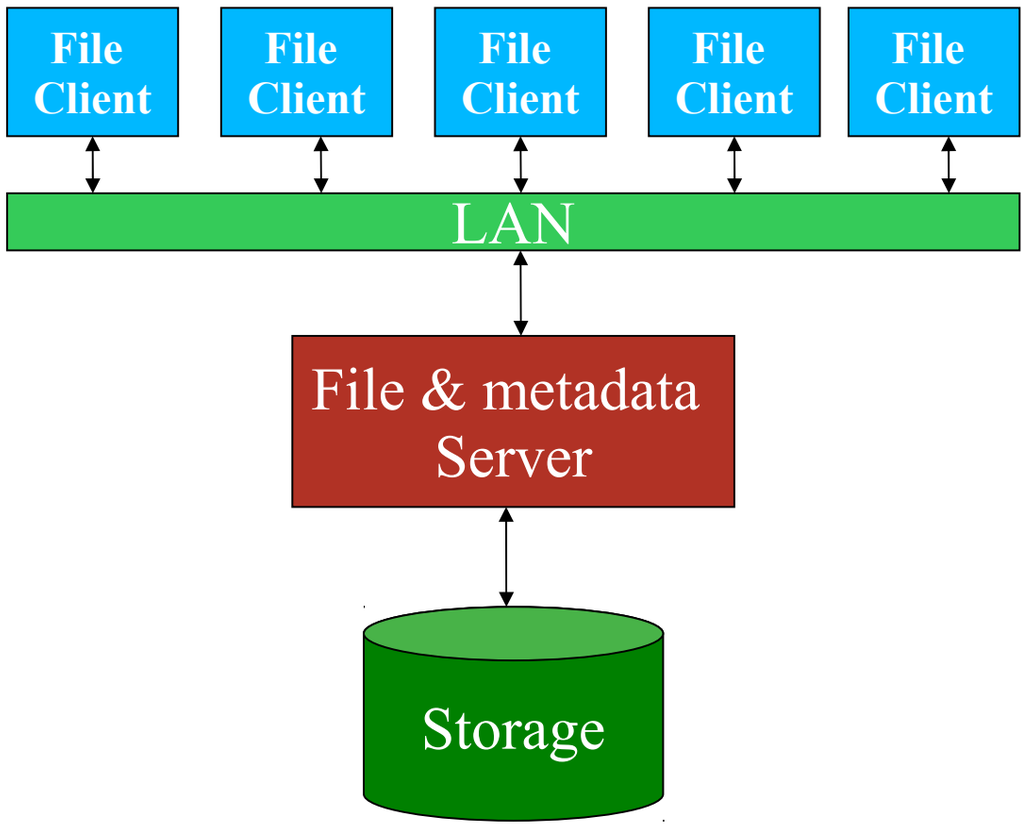
Conceptos - Sistema de ficheros distribuido

- Acceso a datos remotos.
- El acceso debe ser:
  - Transparente.
  - Eficiente.
  - Seguro.
- NFS: Inherente en sistemas Unix / Linux.



Conceptos - Sistema de ficheros distribuido

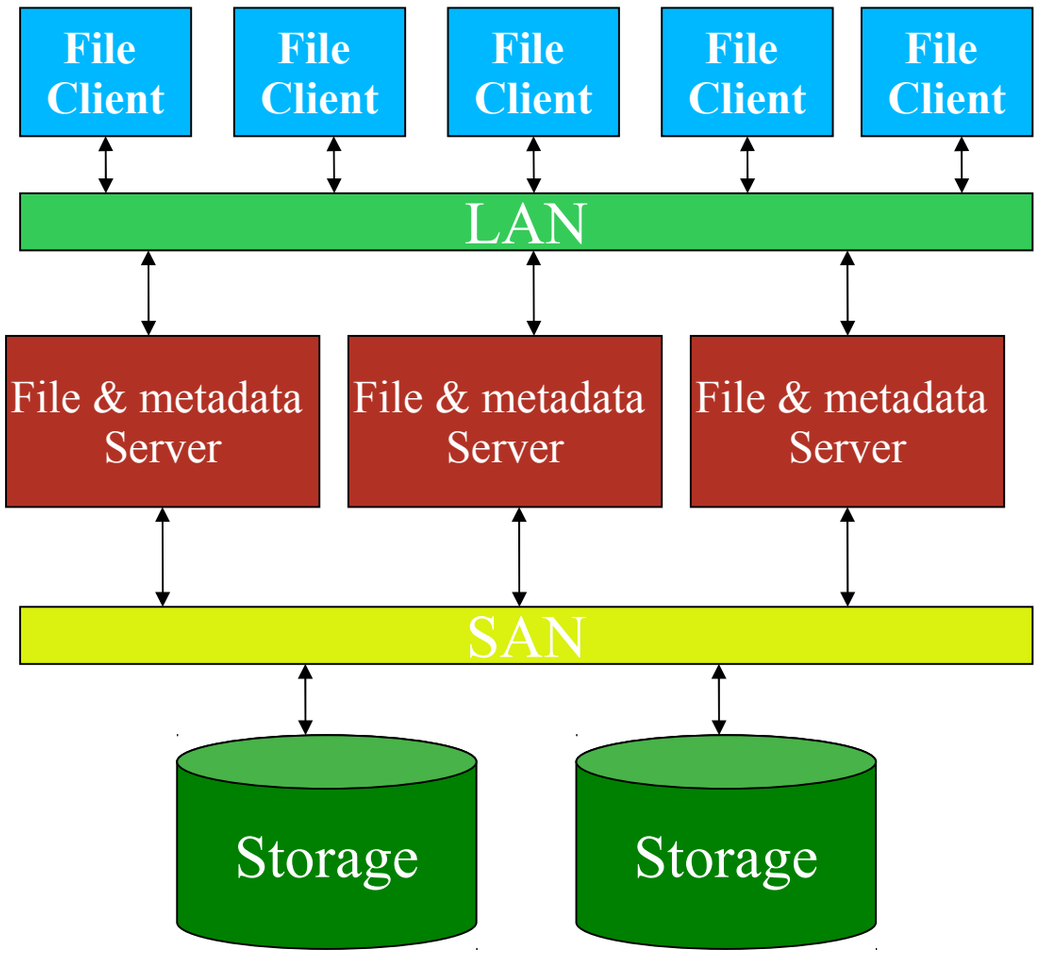
- Acceso a datos remotos.
- Debe ser:
  - Transparente.
  - Eficiente.
  - Seguro.
- NFS: Inherente en sistemas Unix / Linux..



Solo es escalable hasta el limite del servidor.
No puede garantizar la integridad de los datos!!

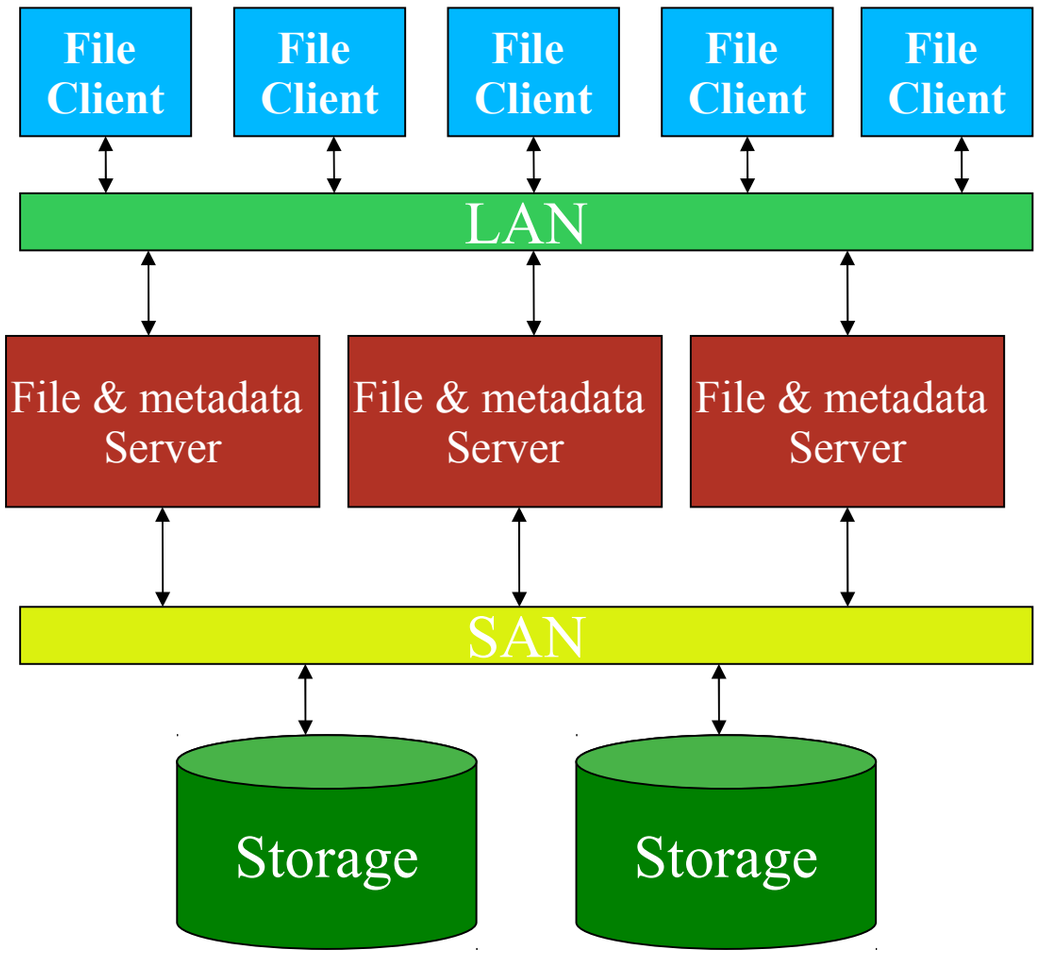
Conceptos - Sistema de ficheros Paralelo

- Cumple con la definición de sistema de ficheros distribuidos.
- Operaciones de I/O distribuidas y ejecutada en paralelo.
- Espacio de nombres global.
- Acceso concurrente a los ficheros.
- Ficheros repartidos en múltiples discos.
- Múltiples servidores.
- Generalmente configurados con clientes y servidores por separado.
- Ejemplos: Lustre, Panasas, GPFS,…



Conceptos - Sistema de ficheros Paralelo

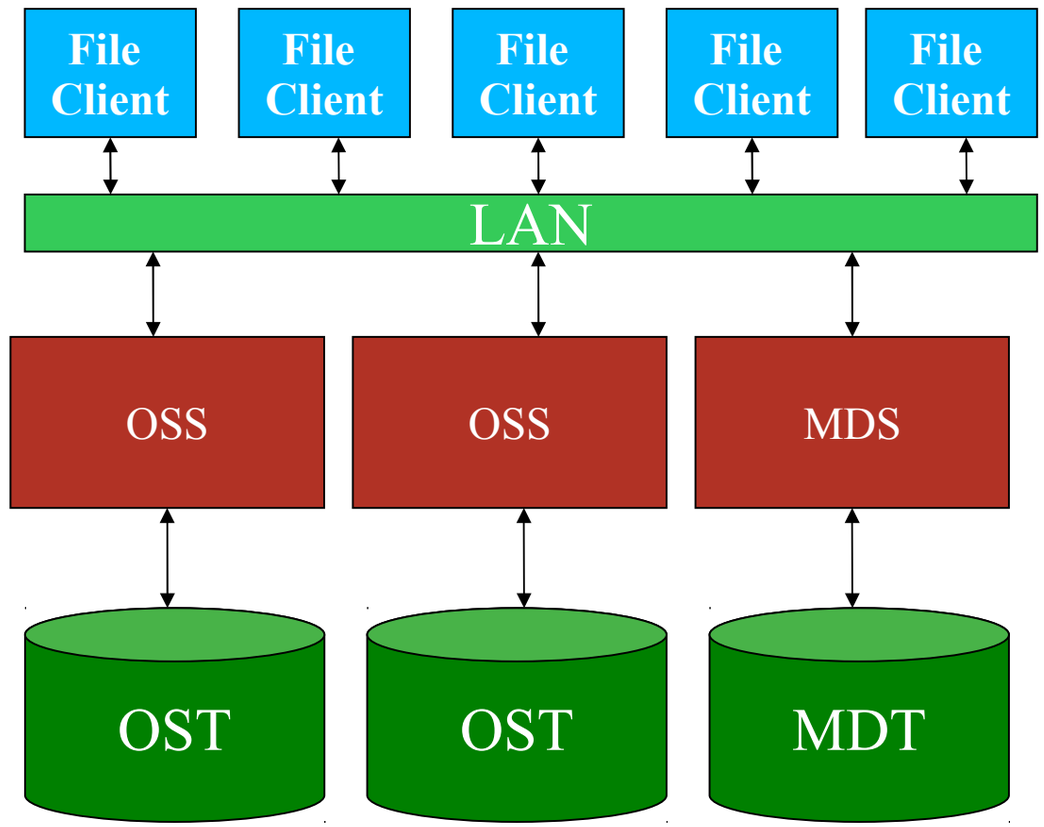
- Cumple con la definición de sistema de ficheros distribuidos.
- Operaciones de I/O distribuidas y ejecutada en paralelo.
- Espacio de nombres global.
- Acceso concurrente a los ficheros.
- Ficheros repartidos en múltiples discos.
- Múltiples servidores.
- Generalmente configurados con clientes y servidores por separado.
- Ejemplos: Lustre, Panasas, GPFS,…



Es escalable
Puede ofrecer un alto rendimiento
Cumple con los requerimientos de Input/Output

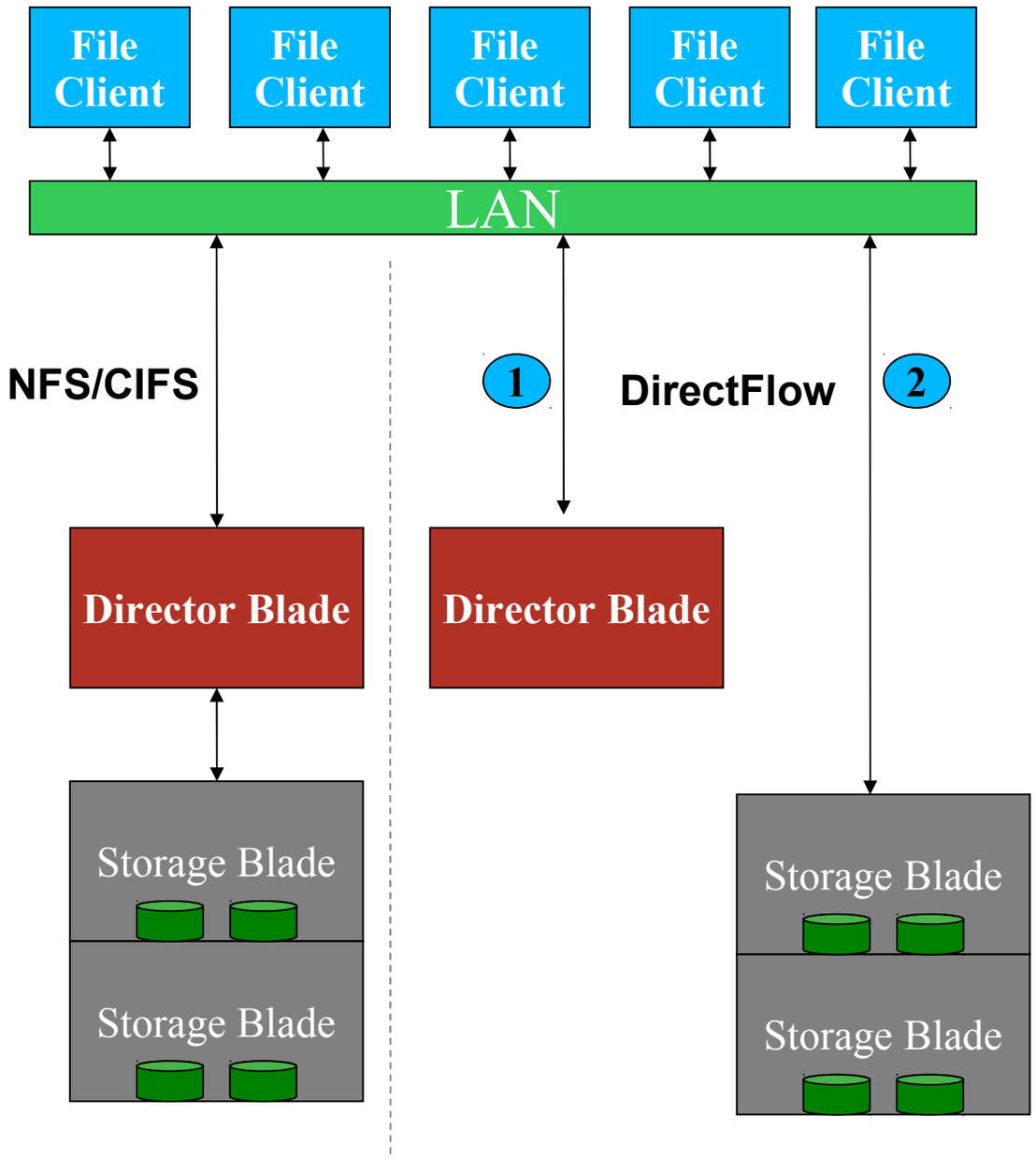
Lustre

- Sistema de Ficheros Orientado a Objetos.
- OpenSource (de momento..).
- Un Servidor de Metadatos (MDS) por file system, con su MDT (Metadata Target) asociado.
  - Gestion de Metadatos (nombres de ficheros, directorios, permisos,..).
  - Gestion de Locks.
  - Puede tener un servidor de backup (activo/pasivo).
- Multiples Object Storage Servers (OSS), con sus OST (Object Storage Target) asociados.
  - Guardan los datos en si.
  - La capacidad total del file system es la suma de la capacidad de todos sus OSTs.



Panasas

- Sistema de ficheros Orientado a Objetos.
- Propietario (Panasas, Inc)
- SO propio (ActiveScale)
- Implementa una versión propia de pNFS (DirectFlow), pèro también puede funcionar como servidor de NFS/Cifs estándar.
- Director Blade:
  - Gestor de metadatos.
  - Implementan volúmenes virtuales
  - Escalable.
  - Todos tiene accesos a todos los datos. Permite repartir la carga en caso de Nfs/Cifs.
- Storage Blade:
  - Almacena los datos.
  - Accesible por red directamente.

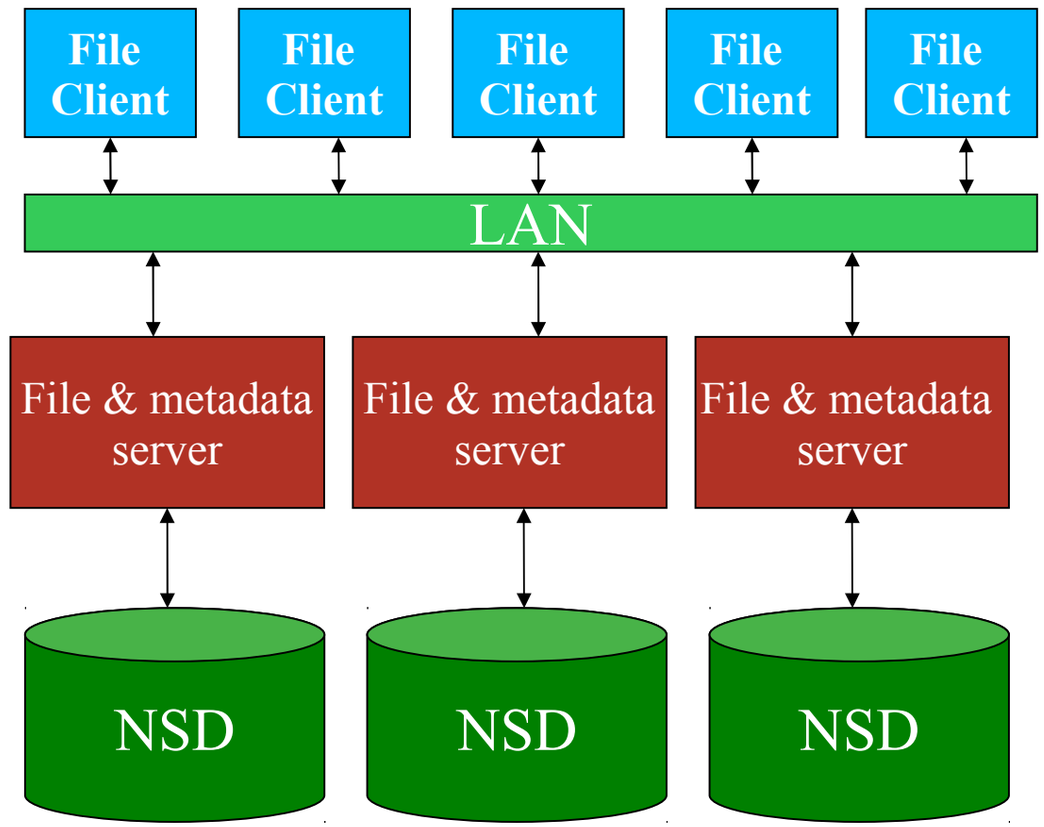


General Paralel File System (GPFS)

- Introducción.
- Lustre
- Panasas
- **GPFS**
  - **Arquitectura**
  - **Roles**
  - **Bloques**
  - **Cache**
  - **Gestión de Locks**
  - **Multicluster**
  - **Storage Pools**
  - **FileSet**
  - **Recomendaciones**
  - **GPFS MareNostrum**
  - **GPFS MareNostrum Futuro**

General Paralel File System (GPFS)

- Propietario (IBM).
- Desarrollado inicialmente para sistemas IBM SP con AIX (1991) y disponible clientes Linux desde 1998.
- Probablemente uno de los mas maduros y estables.
- El file system esta compuesto por nsd's (Network Storage Devices).
- NSD:
  - Disco a nivel de GPFS.
  - Se corresponde con una LUN en el server.



GPFS - Roles

- **Quorum Nodes**
  - Conjunto de nodos de los cuales tiene que haber la mitad mas uno para que el GPFS este online.
- **Configuration manager** (primario y secundario)
  - Comprueba que haya quorum suficiente para arrancar el filesystem
  - Selecciona el file system manager de cada file system, y reasigna en caso de fallada de este.
- **Filesystem manager** (1 por file system de GPFS)
  - Forma parte del grupo de nodos de management
  - Responsable de gestionar los cambios en la configuración, añadir /eliminar discos,…
  - Gestiona la alocatación en disco
  - Gestión de tokens.
  - Gestión de cuota.
- **Metanode** (1 por fichero abierto)
  - Mantiene la integridad del file system.
  - N clients pueden leer/escribir directamente, pero solo el metanode se encarga de los updates en los metadatos.

GPFS - Bloques

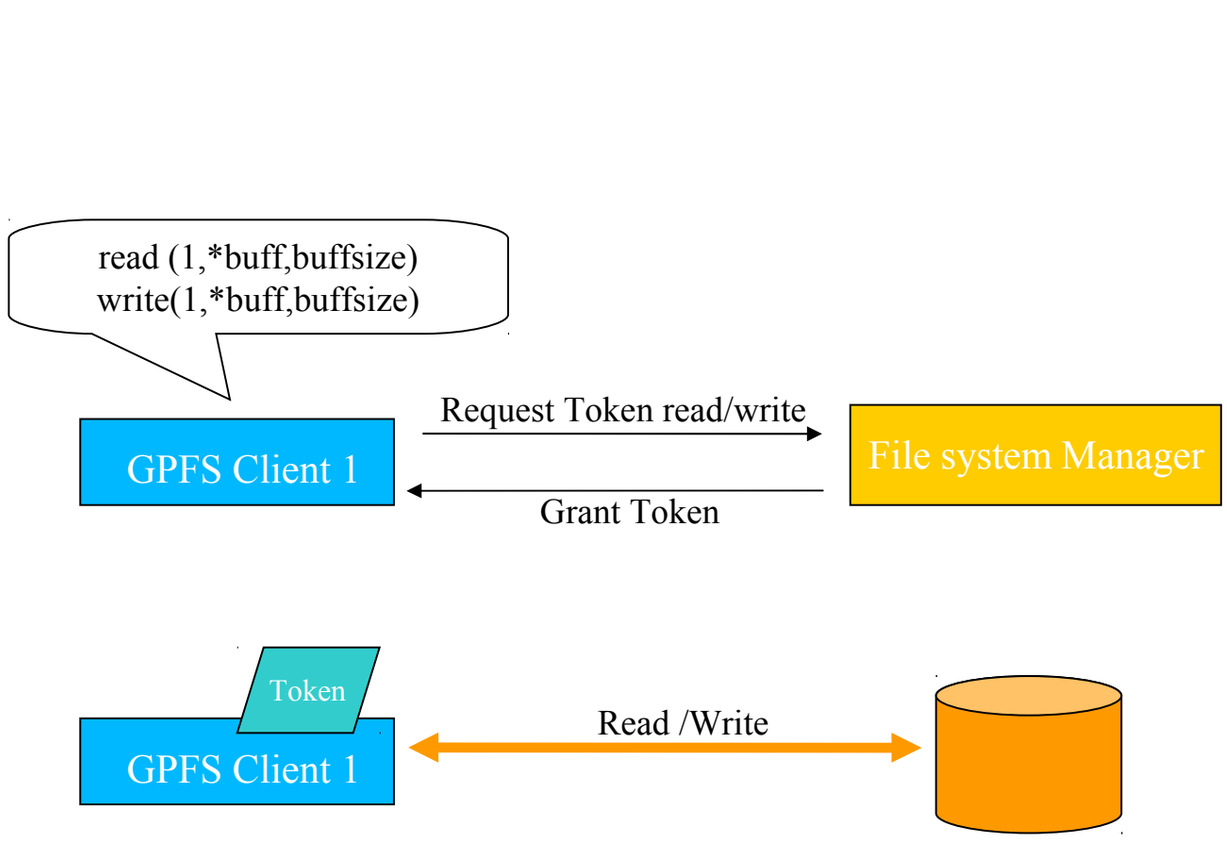
- Bloque: Mayor unidad de transferencia en GPFS.
  - Ejemplo: if (Bloque =1MB) and (record= 4MB) => 4 operaciones de I/O.
- Sub-Bloque: Minima unidad de transferencia (Bloque/32).
  - Un fichero va ocupar como mínimo un sub-bloque.
- GPFS Permite definir bloques de hasta 4MB.
- Stripping: Los bloques consecutivos de cada fichero se alocatan en diferentes discos siguiendo un algoritmo de round-robin.
  - Incrementa el ancho de banda disponible
  - Balancea la carga sobre todos los discos del file system.

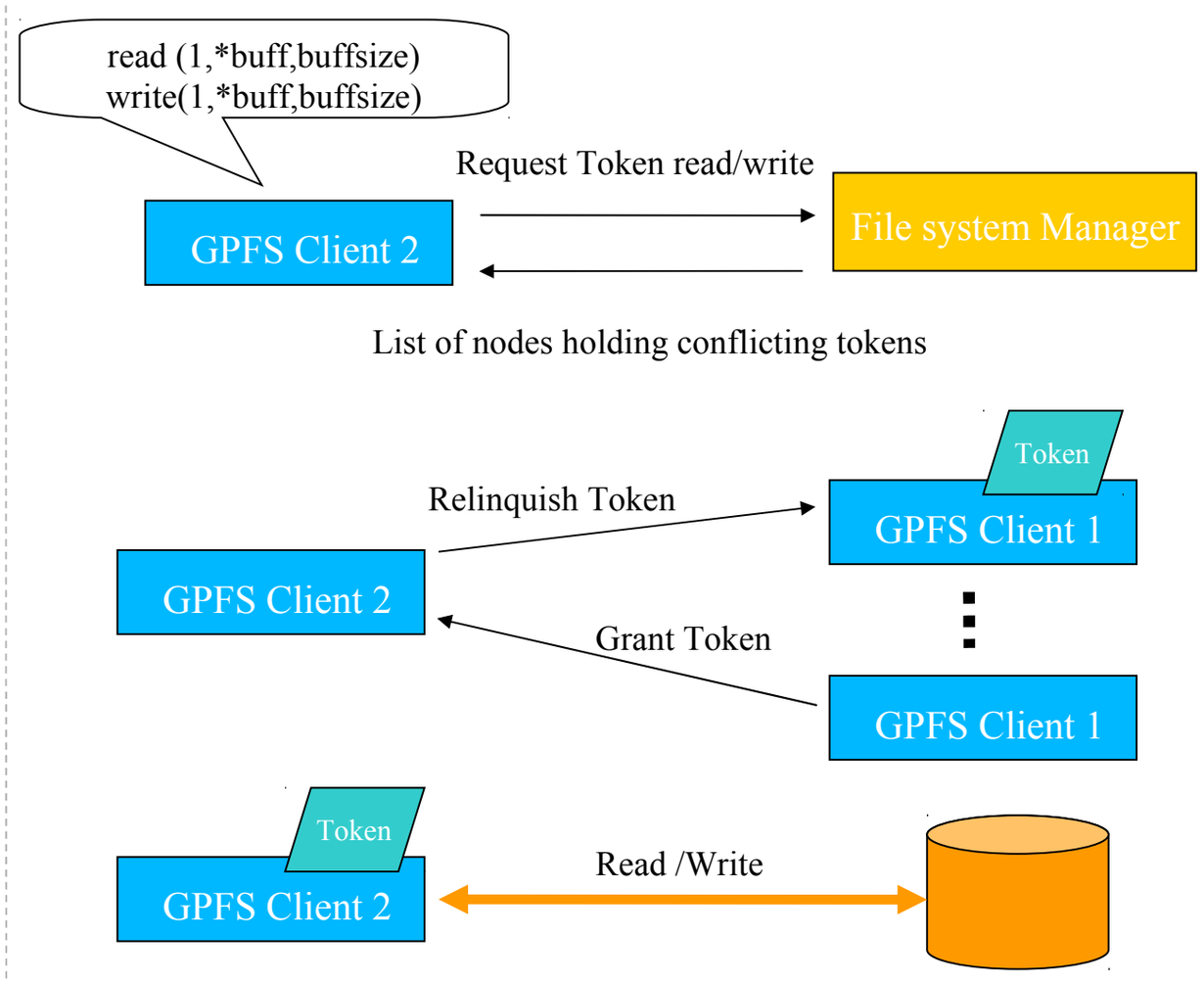
GPFS - Cache

- Utiliza una cache local en cada nodo (pagepool).
- Útil cuando accedemos siguiendo patrones que GPFS es capaz de reconocer:
  - Acceso sequencial (Forward / Backward).
  - Acceso por partes (Forward/Backward).
  - Ficheros pequeños
- GPFS realiza prefetch a cache de forma a asyncrona y con operaciones paralelas.

GPFS - Gestión de Locks

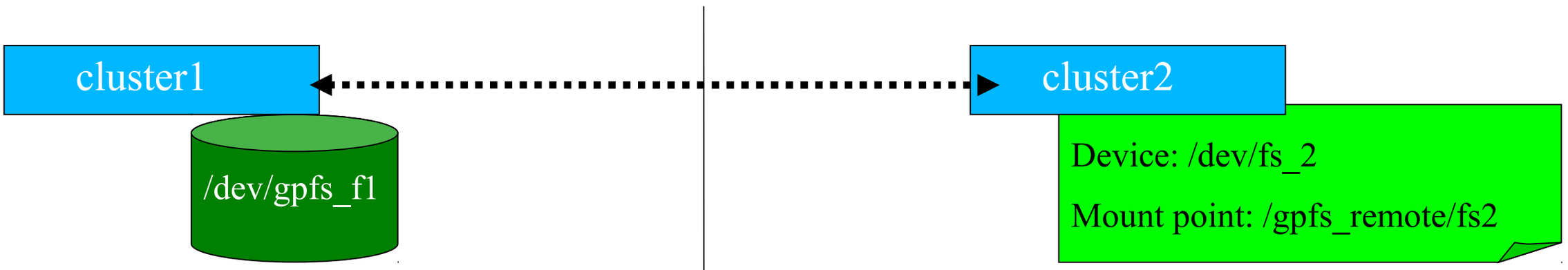
- Para mantener la consistencia de los ficheros GPFS tiene un sistema distribuido de locks a base de tokens.
- Los tokens pueden ser a nivel de bloque, o a nivel de fichero, dependiendo de la operación que se este realizando.
- El File system manager es el encargado de la gestión de tokens. Coordina el acceso a los ficheros dando privilegios de lectura/escritura a datos/metadatos





GPFS - Multicluster

- Existe un mecanismo de uid mapping para poder realizar mapeo de usuarios entre clusters
- Se establece una conexión SSL usando claves públicas y privadas para la comunicación entre los cluster
- Desde que se realiza el montaje el cluster remoto es como si fuera local a nivel de locks.
- Todos los nodos de los dos clusters deberían ser capaces de ser accesibles mediante TCP/IP




```
cluster1# mmauth genkey
cluster1# mmchconfig cipherList=AUTHONLY
```

```
cluster2# mmauth genkey
cluster2# mmchconfig cipherList=AUTHONLY
```

```
cluster1# mmauth add cluster2 -k cl2_id_rsa.pub
```

```
cluster2# mmremotecluster add cluster1 -n node1,
node2 -k cl1_id_rsa.pub
```

```
cluster1# mmauth grant cluster2 -f /dev/gpfs_fs1
```

```
cluster2# mmremotefs add /dev/gpfs_fs2 -f
/dev/gpfs_fs1 -T /gpfs_remote/fs2
```

GPFS - Storage pools

- Colección de discos o RAIDs con las mismas características (performance, localidad, …)
  - Internal storage pools: Storage pools definidos y gestionados por GPFS
  - External storage pools: Storage controlado por otros aplicativos: Tivoli Storage Manager
- System storage pool
  - Contiene estructuras del filesystem, ficheros reservados, directorios, symlinks, dispositivos y los metadatos. También puede contener datos
  - Sólo hay un system storage pool por filesystem y es automáticamente creado con el filesystem.
- User storage pool
  - Se especifica en la definición de los NSD
- GPFS asigna datos a los internal storage pool de la siguiente manera:
  - Cuando un fichero se crea se determina el storage pool por la: file placement policy
  - Cuando los atributos del fichero (tamaño, acces time, …) hace matching con una regla de una política definida que indique migrar esos datos a otro storage pool.

GPFS – Storage Pools: Políticas

- File placement rule
  RULE ['*RuleName*'] **SET POOL** '*PoolName*'
  [**LIMIT** (*OccupancyPercentage*)]
  [**REPLICATE** (*DataReplication*)]
  [**FOR FILESET** (*FilesetName*...)]
  [**WHERE** *SqlExpression*]
- File migration rule
  RULE ['*RuleName*'] [**WHEN** *TimeBoolean*]
  MIGRATE
  [**FROM POOL** '*FromPoolName*'
  [**THRESHOLD** (*HighPercentage*[,*LowPercentage*
  [,*PremigratePercentage*]])]]
  [**WEIGHT** (*WeightExpression*)]
  TO POOL '*ToPoolName*'
  [**LIMIT** (OccupancyPercentage)]
  [**REPLICATE** (*DataReplication*)]
  [**FOR FILESET** (*FilesetName*...)]
  [**SHOW** (['*String*'] *SqlExpression*)]
  [**SIZE** (*numeric-sql-expression*)]
  [**WHERE** *SqlExpression*]
- File deletion rule
  RULE ['*RuleName*'] [**WHEN** *TimeBoolean*]
  DELETE
  [**FROM POOL** '*FromPoolName*'
  [**THRESHOLD**(*HighPercentage*[,*LowPercentage*])]]
  [**WEIGHT** (WeightExpression)]
  [**FOR FILESET** (*FilesetName*,...)]
  [**SHOW** (['*String*'] *SqlExpression*)]
  [**SIZE** (*numeric-sql-expression*)]
  [**WHERE** *SqlExpression*]
- File exclusion rule
  RULE ['*RuleName*'] [**WHEN** *TimeBoolean*]
  EXCLUDE
  [**FROM POOL** '*FromPoolName*']
  [**FOR FILESET** (*FilesetName*[,*FilesetName*]...)]
  [**WHERE** *SqlExpression*]

- Some SQL Expressions
- File attributes
  - ACCESS_TIME
  - BLOCKSIZE
  - CHANGE_TIME
  - CREATION_TIME
  - DEVICE_ID
  - FILE_SIZE
  - FILESET_NAME
  - GROUP_ID
  - INODE
  - KB_ALLOCATED
  - MODE
  - MODIFICATION_TIME
  - NAME
  - NLINK
  - PATH_NAME
  - USER_ID
- Built-in functions
  - CHAR(expr,length)
  - CONCAT(x,y)
  - HEX(x)
  - LENGTH(x)
  - LOWER(x)
  - SUBSTR(x,y,z)
  - UPPER(x)
  - CURRENT_DATE
  - CURRENT_TIMESTAMP
  - DAYOFWEEK(x)
  - DAY(x)
  - DAYS(x)
  - DAYSINMONTH(x)
  - DAYSINYEAR(x)
  - HOUR(x)
  - MINUTE(x)
  - MONTH(x)
  - SECOND(x)
  - TIMESTAMP(x)
- and much more …

GPFS - Filesets

- Es como un file system virtual dentro de un file system de GPFS:
  - Es como un subárbol de un file system de GPFS pero con funcionamiento independiente, permite operaciones administrativas a granularidad más fina
- El fileset al cual un fichero pertenece es completamente transparente desde el punto de vista de uso.
- Se pueden definir cuotas a nivel de filesets
  - Ejemplo: 1 GPFS filesystem exportado por CIFS donde tenemos espacios para los homes de los usuarios internos del BSC y espacio para los departamentos del BSC.
- Se pueden establecer políticas de placement y migration diferentes por fileset
- Se pueden establecer hasta 1000 filesets por GPFS filesystem
- Cuando un filesystem se crea sólo existe un fileset llamado root (contiene root directory del filesystem).

- `# mmcrfileset gpfs_projects usuarios_chungos -t “Fileset para usuarios chungos”`
- `# mmlinkfileset gpfs_projects usuarios_chungos -J /gpfs/projects/chungos`

GPFS - Recomendaciones

- **Evitar operaciones que deban ejecutarse secuencialmente.**

  - Redirigir el output de todas las tasks al final de un único fichero → **MAL**
  - Crear un fichero por task → **BIEN**

  - Que cada task cree su propio fichero de output → **LENTO**
  - Que el master cree todos los ficheros de output → **MAS EFICAZ**
  - Que el master cree un directorio por task → **AUN MEJOR**

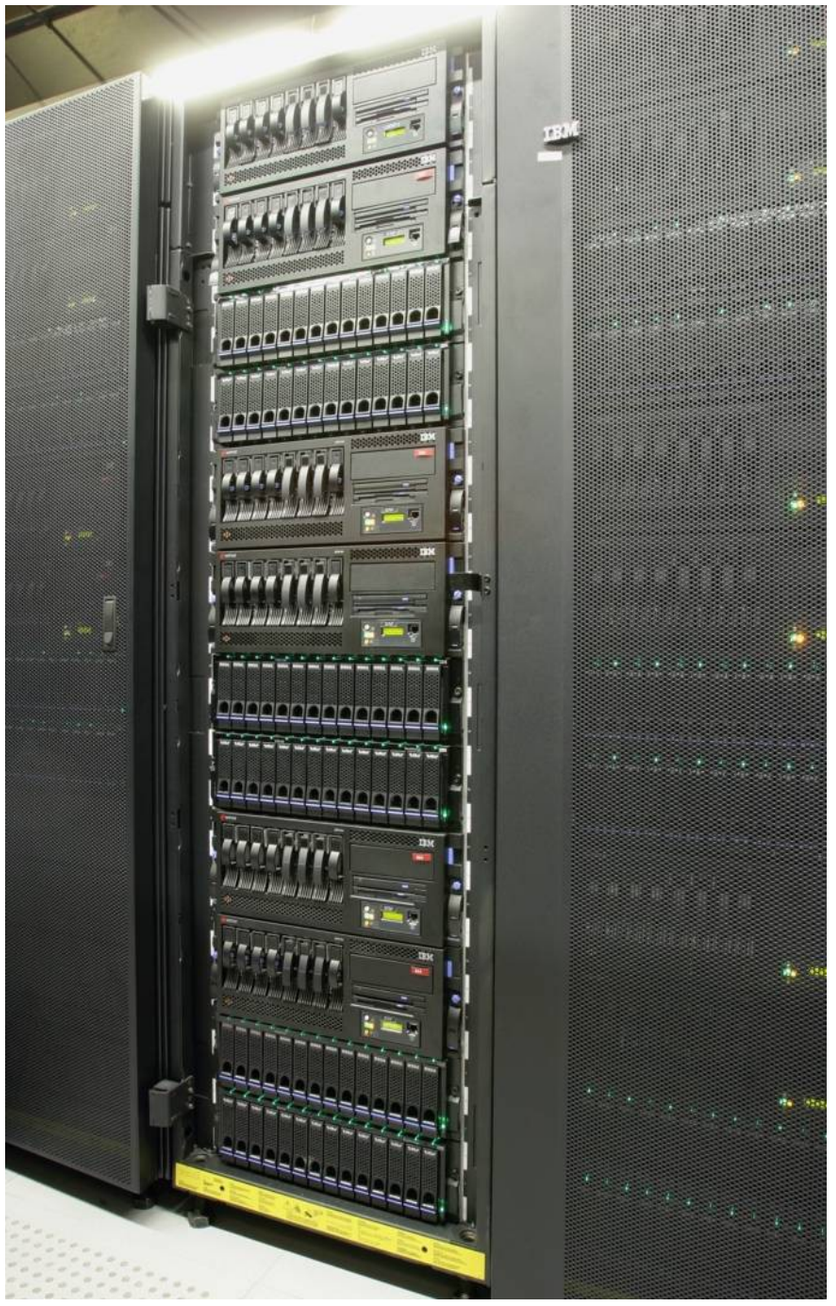
GPFS - MareNostrum

Total de 230TB de datos, 165TB netos disponibles para el usuario.

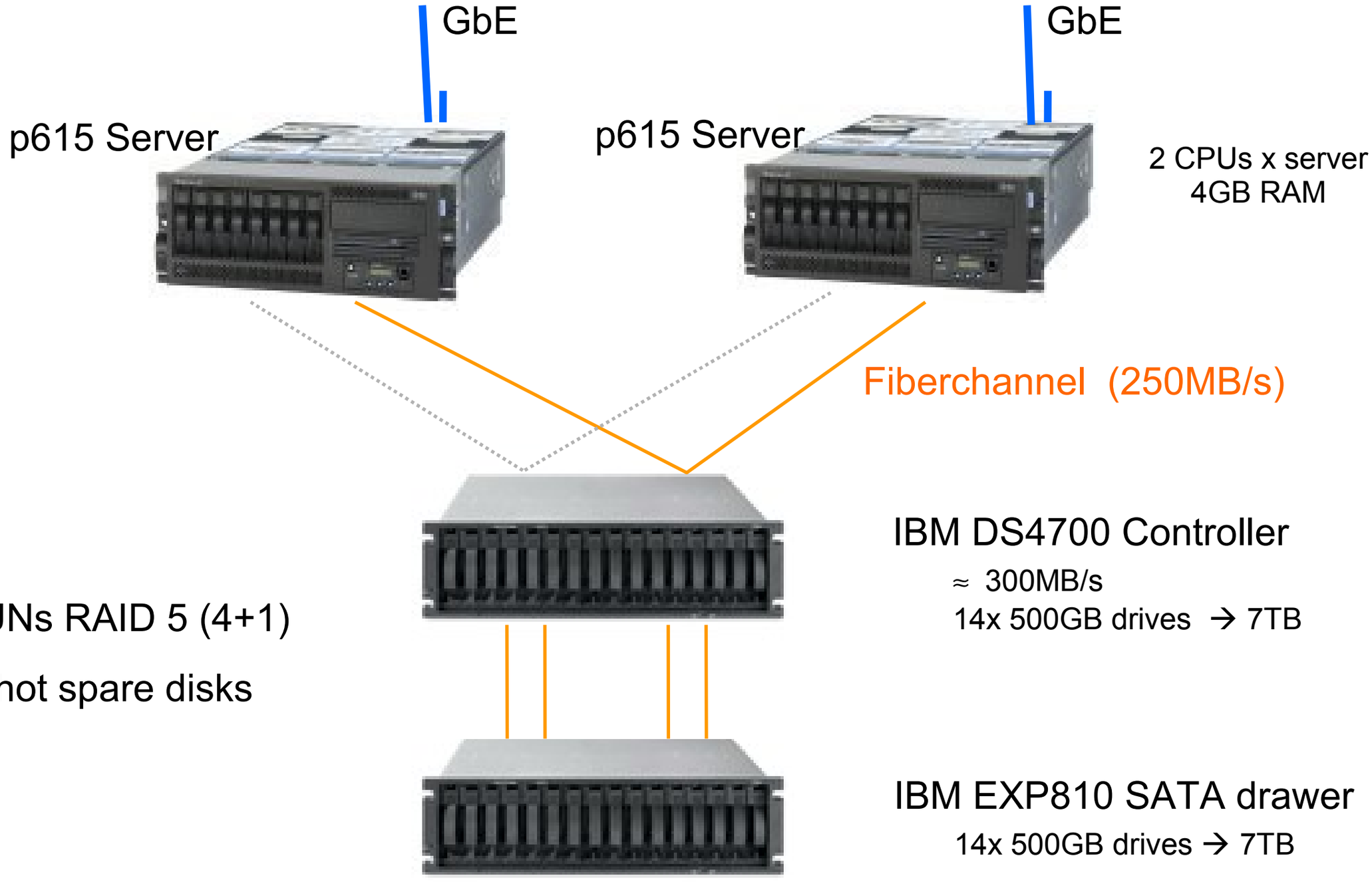
- **/gpfs/projects :**
  - Espació dedicado a los datos que deben ser compartidos por los usuarios de un mismo grupo (inputs, restarts,…)
  - Tamaño 73 TB; Block size 1 MB
- **/gpfs/scratch**
  - Destinado a datos temporales necesarios durante la ejecución de un job. Una vez finalizado los datos deben ser eliminados o movidos hacia un file system permanente.
  - Tamaño: 64 TB; Block size 1MB
- **/gpfs/home**
  - El home directory esta destinado a sources/binarios de aplicaciones propias y datos personales.
  - Tamaño: 19 TB; Block size 256 KB
- **/gpfs/apps**
  - Aplicaciones compiladas y optimizadas por el equipo de suport.
  - Tamaño: 9.1 TB; Block size 512 KB

GPFS - MareNostrum - Storage Hardware

- Total de 20 storage nodes, 20 x 14 TBytes
- Cada storage node consta de:
  - 2 x IBM p615
  - DS4700
  - EXP810
  - 28 x 500GB disks



GPFS - MareNostrum - Storage node



5 LUNs RAID 5 (4+1)

3 hot spare disks

IBM DS4700 Controller

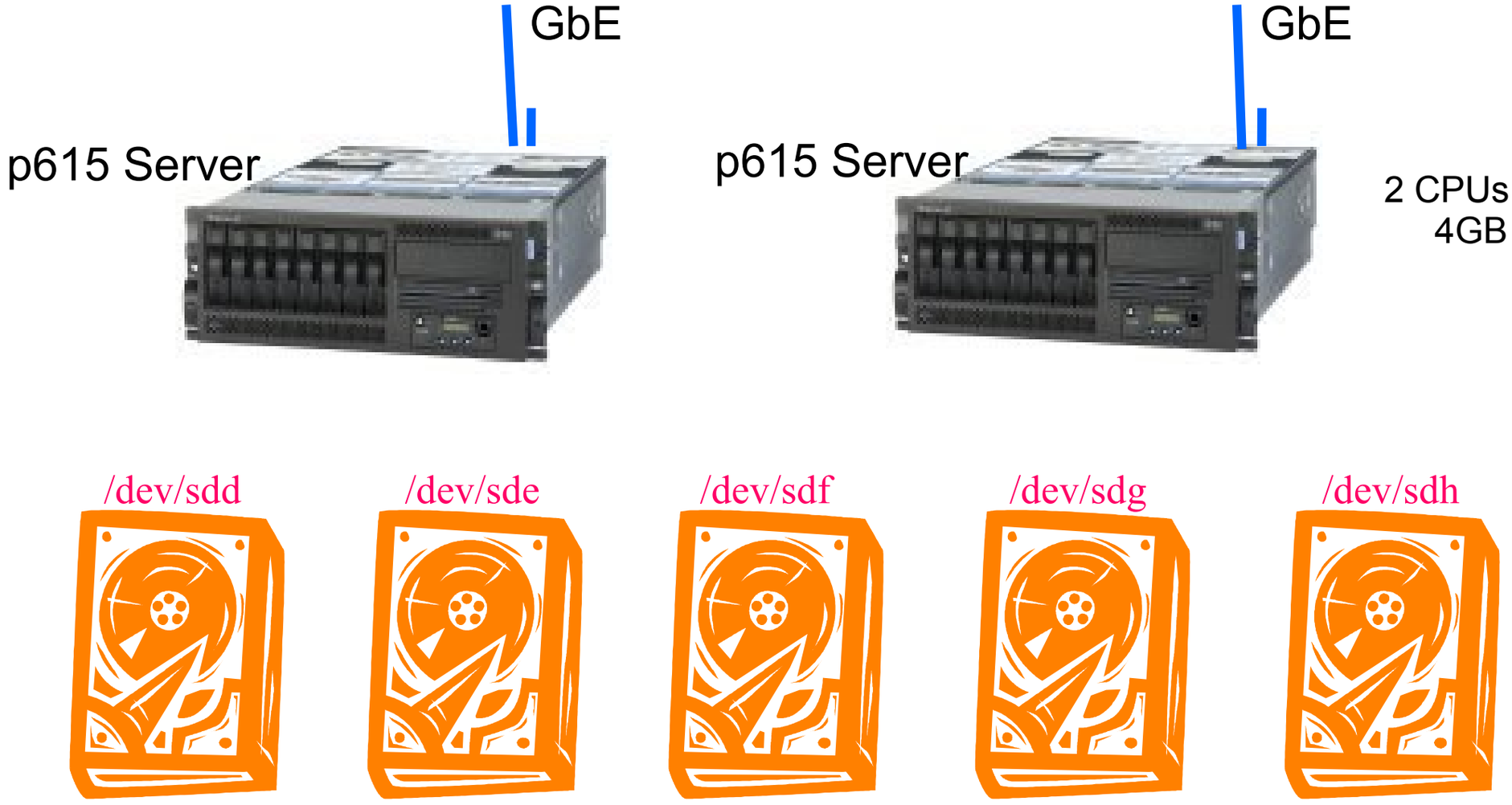
≈ 300MB/s

14x 500GB drives → 7TB

IBM EXP810 SATA drawer

14x 500GB drives → 7TB

GPFS - Marenostrum - Storage Node: Logical view



- Cada LUN es un RAID5 con un tamaño de 2 TB
- Cada LUN se corresponde con un NSD (Network Shared Disk) de GPFS

MareNostrum – Futuro

- Total: 2400TB de datos (1920 TB netos).
- Se mantendran los 4 mismos filesystems actuales:
  - /gpfs/home → 80 TB.aprox. BlockSize 256K
  - /gpfs/apps → 16 TB. aprox. BlockSize 512K
  - /gpfs/scratch -> 1024 TB. aprox BlockSize 4MB.
  - /gpfs/projects -> 800 TB. aprox BlockSize 4MB.

MareNostrum – Futuro - Datos

- 3 Data building block en total
- 8 Data Servers

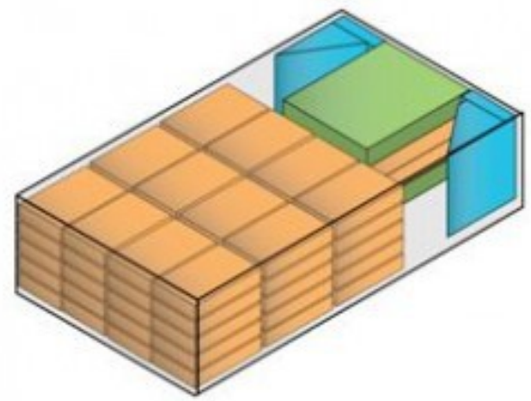
  X3550M2 48GB RAM SLES

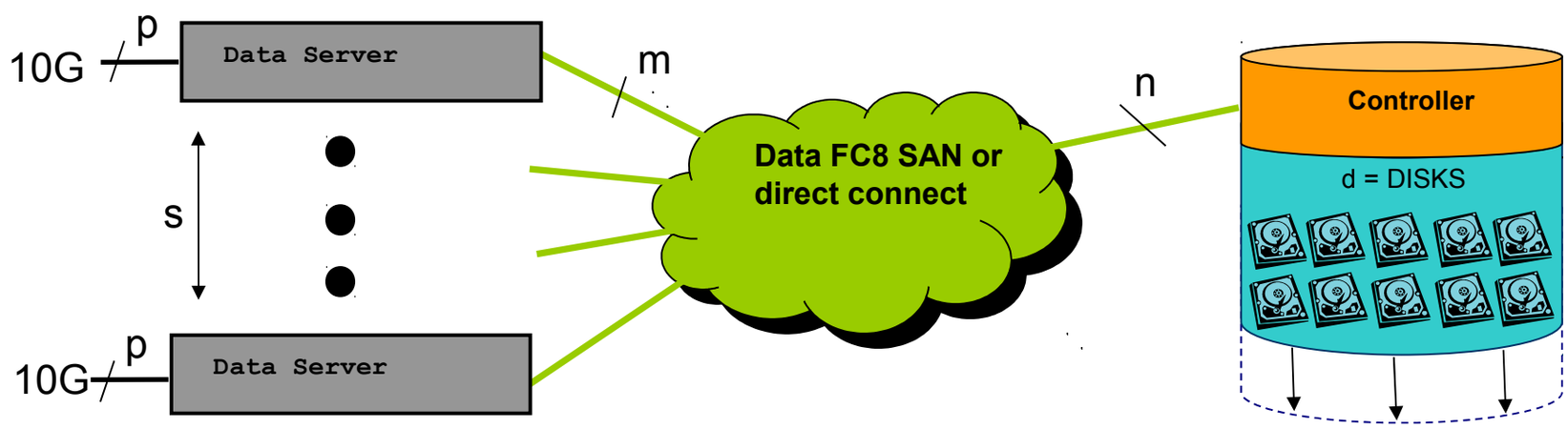
  1 interfaz 10GE y 2 FC8

- Controladora DS5300 y 8 bandejas EXP5060

  400 discos SATA 2TB @ 7200rpm

  40 RAID6 (8+2P)





MareNostrum – Futuro - Metadatos

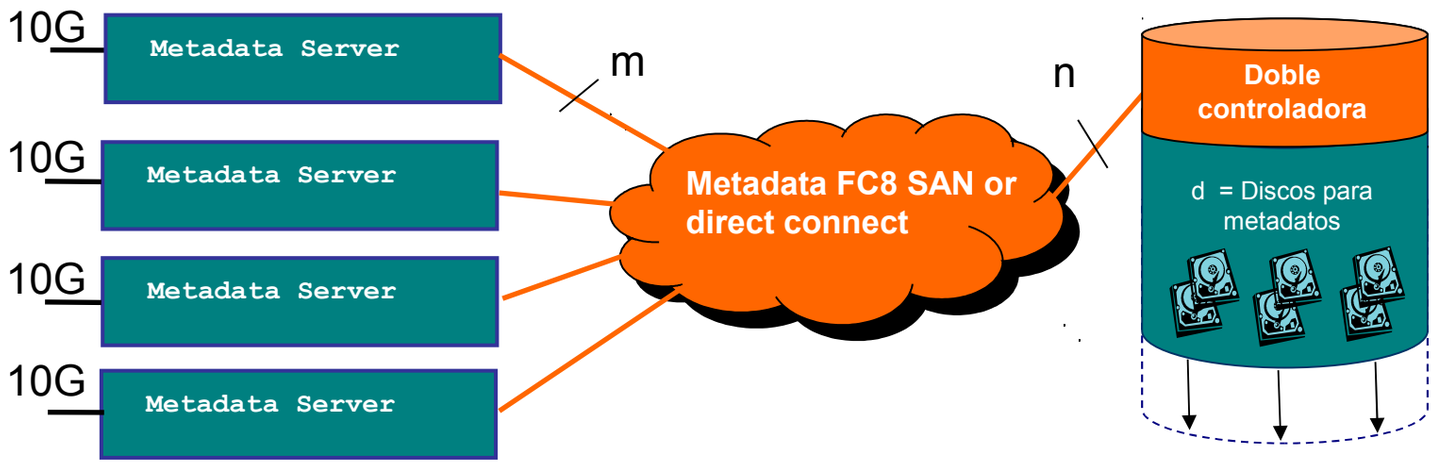
- 6 Metadata Servers

  X3650M2 64GB RAM SLES

  1 interfaz 10GE y 2 FC8

- Controladora DS5300 y 7 EXP500

  112 discos FC 600GB 15Krpm



Dudas

Preguntas?

Gracias !