Introduzione a GPFS: il file system

Alessandro Brunengo
Mirko Corosu
INFN-Genova

Contenuto

- Network Shared Disk
- Caratteristiche del file system
- Creazione del file system
- File system management
- Disk management

Network Shared Disk

Network Shared Disk

- Il file system GPFS e' costituito da una collezione di dischi (Network Shared Disk), su cui GPFS memorizza dati e metadati
  - su linux: ogni block device con una entry in /dev/* (HD, partizioni, LUN esportate da RAID controller, multipath device, ...)
- Ogni NSD deve essere inizializzato tramite il comando mmcrnsd
- mmcrnsd registra il disco come NDS nei file di configurazione, e scrive sul device un NSD descriptor
- l'NSD descriptor contiene le informazioni del cluster di appartenenza, l'NSD name e l'NSD id, tramite i quali potra' essere riconosciuto
  - indipendentemente dal device name

NSD server e failover

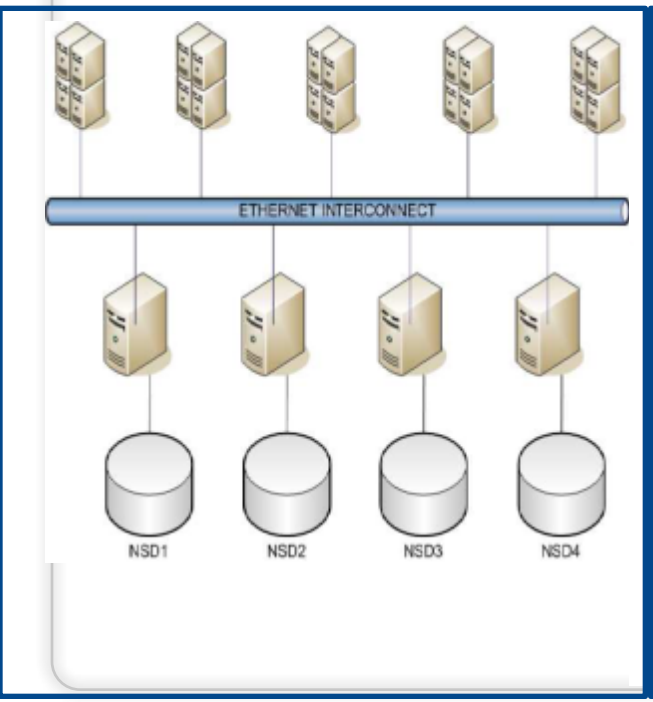
- Per ogni NSD possono essere definiti fino ad 8 NSD server
  - per ciascun NSD, solo un NSD server e' attivo in un dato istante
  - in occasione di server failure (server crash, server network failure, server SAN failure) l'NSD viene servito dall'NSD server successivo specificato nella lista
- Il failover in occasione di un NSD server failure ed il recovery sono automatiche ed integrate in GPFS
  - l'operazione di gestione della failure e di recovery con l'assegnazione di un nuovo server per gli NSD comporta un ritardo inferiore al minuto sugli I/O in corso
  - tipicamente le applicazioni non falliscono e proseguono le attivita' di I/O senza problemi
  - il failover interviene anche quando l'NSD server viene fermato utilizzando la procedura ordinaria di shutdown di GPFS (mmshutdown)

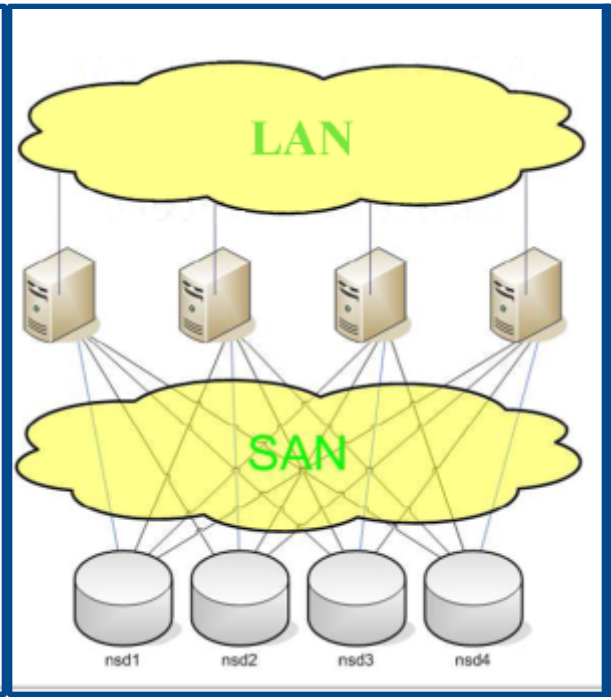
Differenti topologie

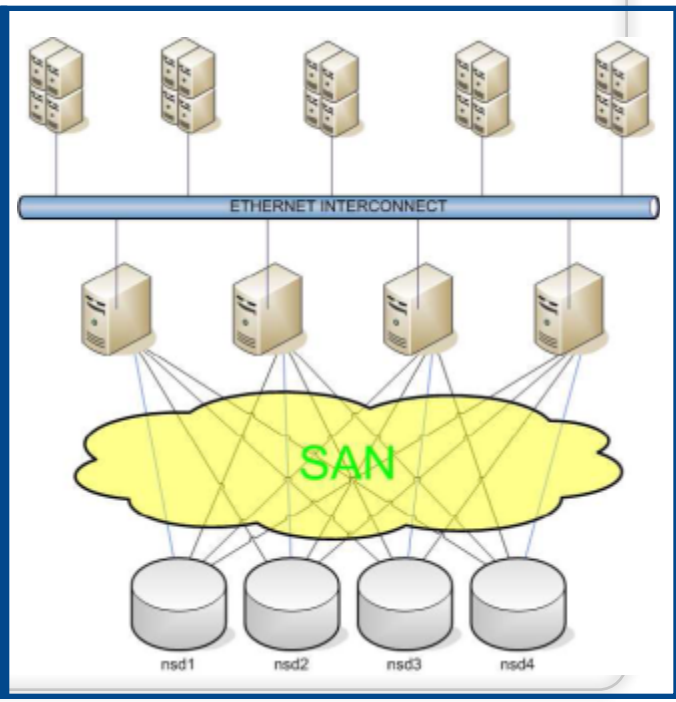
Direct attached disks.
No failover available.

All nodes connected to the SAN.
No NSD server needed.

Mixed topology.
NSD server with failover.







Caratteristiche dell'NSD

- **# mmcrnsd -F StanzaFile [-v {yes |no}]**

  StanzaFile deve contenere una serie di stanze che defniscono i psarametri di configurazione per ogni disco da inizializzare:

```
%nsd: device=DiskName
        nsd=NsdName
        servers=ServerList
        usage={dataOnly | metadataOnly | dataAndMetadata
                | descOnly}
        failureGroup=FailureGroup
        pool=StoragePool
```

  - **DiskName**: il block device name del disco da inizializzare (/dev/*)
    in assenza di ServerList, deve essere il device name con cui viene visto il volume dal nodo su cui si esegue il comando
  - **ServerList**: lista di NSD server (fino a 8)
    - opzionale: non necessario se tutti i nodi accedono alla SAN
    - se presente, il DiskName deve essere quello con cui il primo nodo in ServerList vede il device

Caratteristiche dell'NSD (cont.)

- **DiskUsage**: definisce cosa conterra' l'NSD (dataAndMetadata, dataOnly, metadataOnly, descOnly)
  - utilizzato all'atto della creazione del file system
- **FailureGroup**: intero (tra -1 e 4000) che indica il failure group dell'NSD. Informazione utilizzata all'atto della creazione del file system per decidere come replicare dati, metadati e filesystem descriptor (le repliche vengono collocate su NSD appartenenti a **failure group differenti**)
  - tutti i dischi con uno stesso point of failure dovrebbero essere configurati con lo stesso failure group (es: le LUN esportate da uno stesso controller o i dischi direttamente connessi ad un nodo)
- **DesiredName**: stringa che identifica l'NSD univocamente nel cluster
  - parametro utilizzato nei comandi GPFS per indicare l'NSD (o disk)
- **StoragePool**: nome dello storage pool a cui l'NSD deve appartenere
  - parametro utilizzato all'atto della creazione del file system
  - il nome deve essere univoco nell'ambito del file system

Note sui parametri dell'NSD

- **DesiredName**: non puo' essere cambiato
  - per cambiarlo si deve rimuovere l'NSD e ricrearlo
- **ServerList**: per modificare la lista degli NSD server si utilizza il comando **mmchnsd**:

 # mmchnsd {"DiskDesc[;DiskDesc...]" | -F StanzaFile}

dove **DiskDesk** e' una stringa "**DiskName:ServerList**"
  - se l'NSD fa parte di un file system, prima di eseguire mmchnsd il file system deve essere smontato su tutto il cluster
- **DiskUsage** e **FailureGroup**: e' possibile modificarli al volo senza interrompere l'I/O, tramite il comando **mmchdisk**
- **StoragePool**: per essere cambiato il disco deve essere rimosso dal file system (**mmdeldisk**) e poi riaggiunto (**mmadddisk**)

Visualizzazione degli NSD

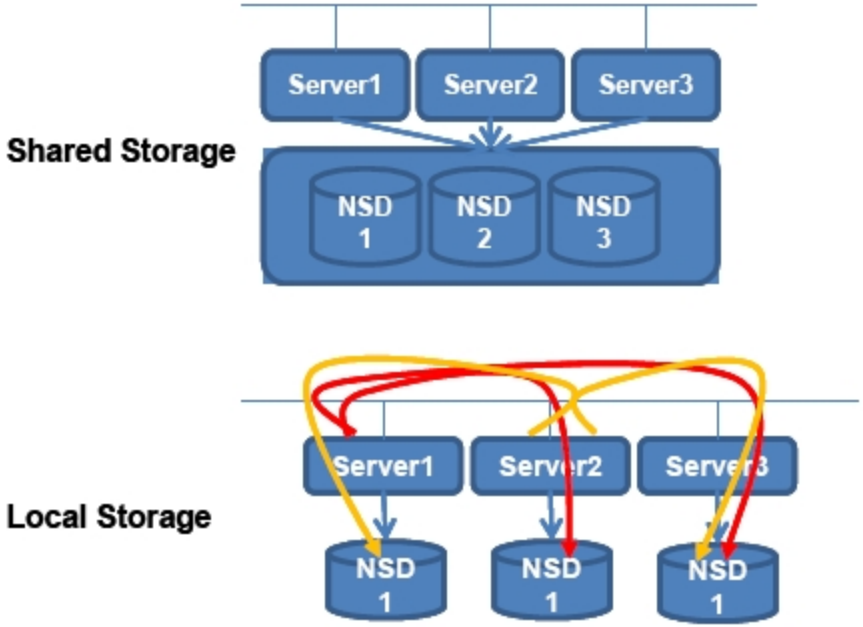
- Per visualizzare gli NSD definiti nel cluster si usa il comando **mmlsnsd**
  - visualizza gli NSD definiti, l'eventuale file system di appartenenza, l'elenco degli NSD server definiti
  - l'opzione **-m** permette di visualizzare il nome del device con cui l'NSD viene visto dagli NSD server

```
# mmlsnsd
 File system   Disk name    NSD servers
---------------------------------------------------------------------------
 backup_dev    f0_a4        bcksrv2.ge.infn.it
 test_dev      part1        (directly attached)
 test_dev      part2        (directly attached)
 (free disk)   part10       (directly attached)
 (free disk)   part3        (directly attached)
 (free disk)   part4        (directly attached)
 (free disk)   part5        (directly attached)
```

Il file system

Il file system GPFS

- Il file system GPFS e' costituito dalla colleazione di uno o piu' dischi (Network Shared Disk) connessi a nodi appartenenti al cluster
- I dischi possono essere locali (direttamente accessibili da un solo nodo) o volumi visibili via SAN (direttamente accessibili da piu' nodi contemporaneamente)

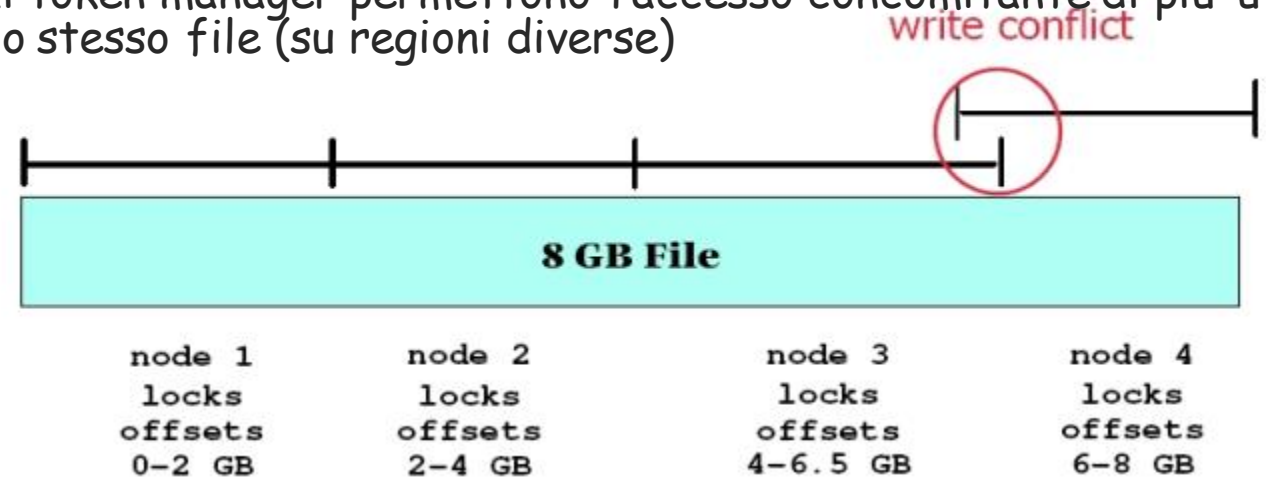


Posix e network file system

- Posix file system
  - supporto per le system call open(), read(), write(), close(), lseek(), unlink() etc..
  - supporto per tutti i comandi shell che operano sul file system (ls, rm, cd, …)
  - ownership e permission unix-like
  - supporto per ACL posix e NFSv4, extended attributes
- Network file system
  - il file system e' accessibile da tutti i nodi del cluster
    - su tutti i nodi il namespace e' omogeneo (stesso mount point)
  - supporto per l'accesso da parte di altri cluster GPFS in modalita' nativa (remote cluster export)
  - supporto per l'export via NFS o samba
    - supporta l'implementazione di NFS ad alta affidabilita' (CNFS)

Parallel file system

- Parallel file system
  - shared disk: tutti i dischi vengono utilizzati contemporaneamente da tutti i nodi (direttamente o tramite NSD server), per dati o metadati
  - stripe: il singolo file viene suddiviso in blocchi che vengono collocati su tutti i dischi del file system in operazioni di I/O parallele (concomitanti)
  - il meccanismo di byte range locking ed il controllo di accesso eseguito dai token manager permettono l'accesso concomitante di piu' utenti allo stesso file (su regioni diverse)



Solidita' ad affidabilita'

- GPFS e' un journaled file system
  - il file system dispone di *recovery logs* (uno per ogni nodo che accede al file system)
  - i recovery logs sono replicati su dischi appartenenti a diversi failure group
  - GPFS mantiene una rigida sequenza di operazioni e logging su data block e metadati
    - questo permette di recuperare la corretta struttura del file e del file system in occasione di crash di un nodo durante operazioni di I/O
- GPFS dispone di file system check
  - mmfsck

Alta disponibilita'

- Il cluster implementa meccanismi per la gestione della failure di **qualsiasi server**, operando il subentro di un altro nodo del cluster in failover, senza perdita di funzionalita'
  - mantiene la funzionalita' di I/O anche in occasione della perdita di un NSD server
  - mantiene la capacita' di recuperare lo stato del file system (compresi i lock) in caso di failure di file system manager o token manager
- Il file system supporta la **replica sincrona di dati e metadati** per sopportare la perdita di un disco
  - attraverso il concetto di failure group e' possibile automatizzare repliche su dischi senza point of failure in comune

Flessibilita' di management

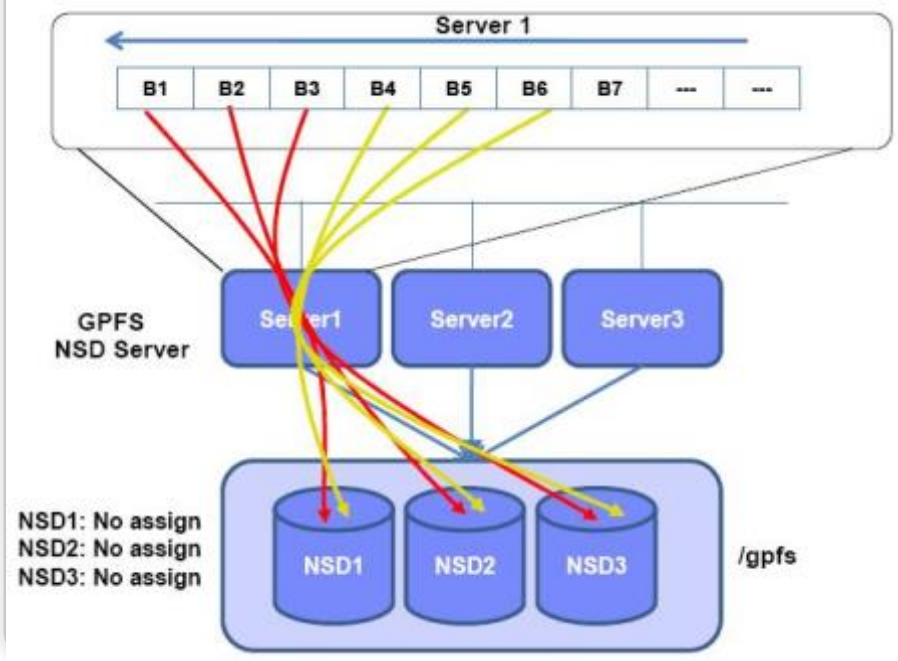
- Cluster file system: management omogeneo e semplificato anche in ambienti di grosse dimensioni
  - la configurazione e le caratteristiche del file system sono note e condivise da tutti i nodi
- Posix file system: posso utilizzare comandi standard Unix
- Supporto per aggiunta/rimozione/sostituzione di dischi dinamicamente, senza interruzione di servizio
  - supporto anche per la ridistribuzione dinamica di dati e metadati in occasione di inserimento di nuovo spazio disco
- Supporto di storage pool e policies di movimentazione dati automatiche
  - permette anche di definire gerarchie di storage (tiering)
- Supporto per separazione di dati e metadati
  - impatto sulle prestazioni

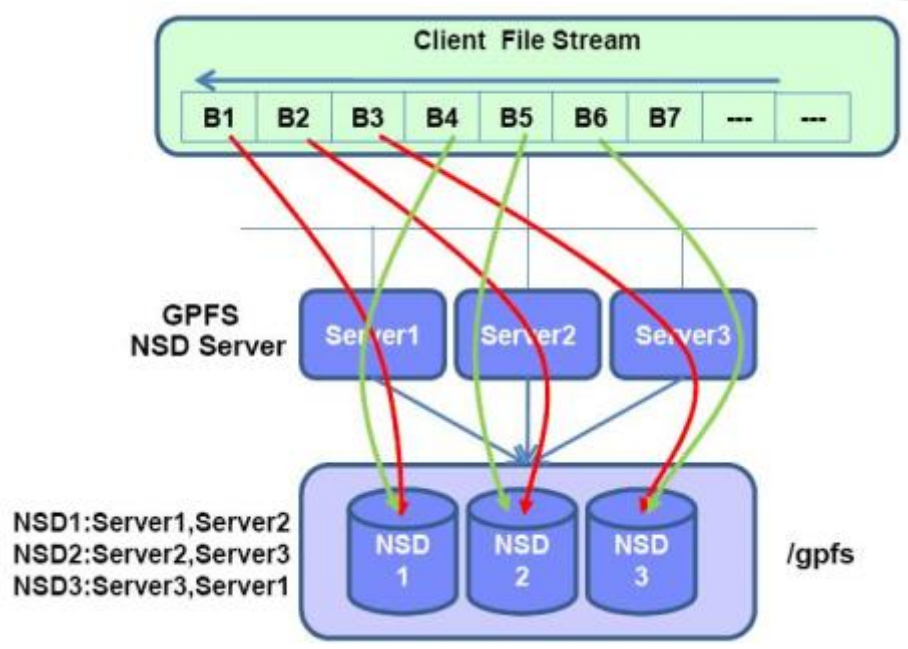
Performance e scalabilita'

- Accesso parallelo da parte di tutti i client su tutti i dischi utilizzati in modalita' stripe
  - permette di sfruttare la banda disponibile verso i dischi o verso gli NSD server
  - l'aggiunta di dischi o NSD server implica aumento di banda disponibile
- Riconoscimento di modalita' di accesso (sequenziale, strided) e sfruttamento della memoria per il read ahead e per write behind
- Scalabile dinamicamente grazie al supporto per l'espansione del file system
  - aggiunta/rimozione dinamica di dischi

Parallelismo dell'I/O sugli NSD server

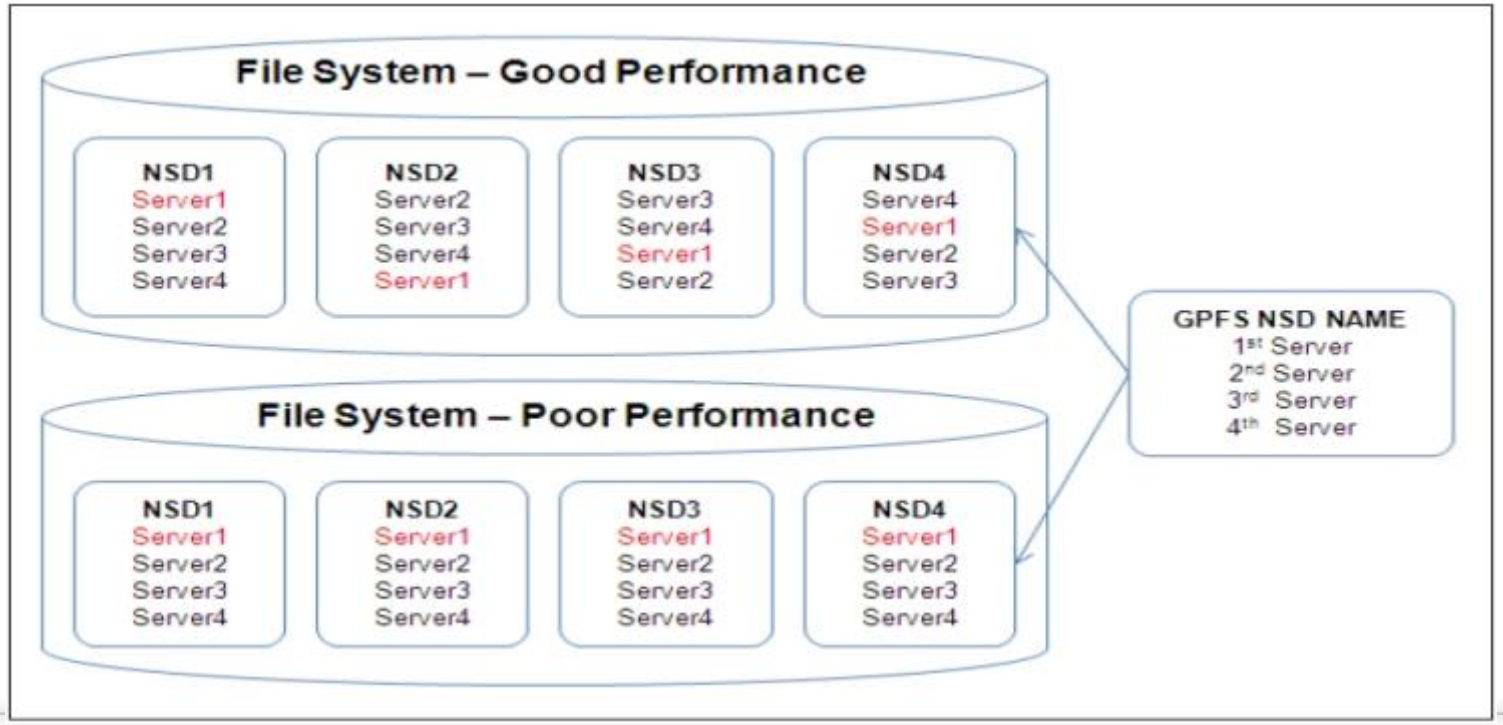
- Una configurazione bilanciata degli NSD server permette di sfruttare il parallelismo ed ottenere prestazioni migliori





NSD server e performance

- Per ottenere le migliori prestazioni si devono adottare configurazioni opportune sulla scelta degli NSD server: nel secondo caso i server 2, 3 e 4 restano sempre inattivi:

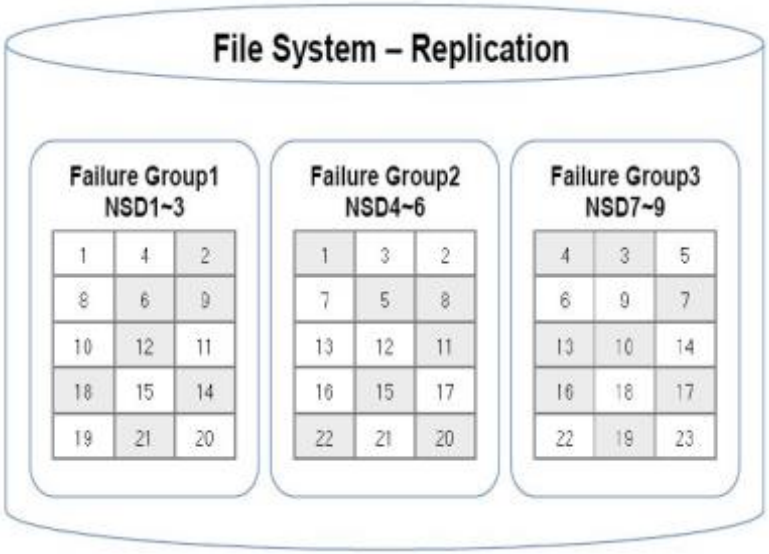


Altre caratteristiche

- Quota
- ACL (posix e NFS v4)
- Snapshot e cloning
- Fileset (partizionamento logico del file system in base al naming)
- Policy based data management
- Export nativo (remote cluster)
- Protocol export (NFS, Samba, Swift) con caratteristiche di HA
- File Placement Optimizer (in topologia shared nothing)
- Advanced File Management (repliche di tipo cache su cluster remoti)

Le repliche sincrone

- GPFS supporta la replica (max tre copie) di dati e/o metadati
  - ogni disco viene assegnato ad un failure group (definito dall'amministratore all'inserimento del disco nel file system)
  - GPFS realizza la copia dei dati e/o metadati tra dischi appartenenti a failure group differenti
  - e' possibile attivare la replica dei dati, dei metadati o di entrambi



- Il comando per definire il fattore di replica **di un file** e' mmchattr
- A livello di file system si definiscono i valori di default e max replica factor per dati e metadati
- Attenzione alla disponibilita' di spazio disco ed inodes:
  - la replica dei metadati raddoppia il numero di i-nodes utilizzati (e di spazio usato per i metadati)
  - la replica dei dati raddoppia lo spazio utilizzato

Struttura del file system GPFS

- file system descriptor
  - contiene, tra le altre cose, l'elenco ordinato dei dischi del file system ed il puntatore all'inode file
  - il file system descriptor viene creato in una o piu' repliche su dischi diversi alla creazione del file system o alla aggiunta di dischi al file system
- inode file: contiene alcuni inodes, tra cui quelli relativi alla root del file system, alla block allocation map, all'inode allocation file
  - la block allocation map contiene una mappa dei subblocks (1/32 della block size) allocati e liberi per ogni disco
    - la quantita' di subblocks indirizzabili per disco e' fissata alla creazione del file system (determina la massima dimensione di un disco per poter essere inserito nel disk pool del file system)
    - viene creata in porzioni distinte allocabili contemporaneamente: il numero determina quanti nodi possono contemporaneamente allocare/rilasciare data blocks sul file system
  - l'inode allocation file contiene l'inode allocation map, che contiene info sugli inode liberi ed occupati
    - puo' essere dinamicamente estesa fino al limite architetturale
- i file vengono messi su disco come in altri fs Unix: inodes, indirect blocks, data blocks

Creazione del file system

Creazione del file system

- Per creare un file system:

  mmcrfs <device name> -F StanzaFile

- Il comando crea un file system utilizzando gli NSD specificati nello StanzaFile, e crea un device file in /dev, associato al file system.
- Il device viene creato su tutti i nodi del cluster

File system block size

- Si possono specificare numerosi parametri per definire le caratteristiche del file system. Attenzione: alcuni non sono piu' modificabili.
- La block size del file system e' un parametro non modificabile
- La block size (BS) e' la quantita' di dati scritti per singola operazione di I/O su ciascun disco del file system
  - se il file system e' costituito da N dischi, l'I/O verra' realizzata operando N operazioni di I/O ciascuna di BS bytes, in parallelo
    - un file di dimensione inferiore alla block size sara' scritto su parte di un unico blocco di un singolo NSD
  - la BS e' quindi la massima quantita' di dati che GPFS gestisce in una singola operazione di I/O
- GPFS supporta block size da 64 KB a 16 MB, con default 256 KB
  - per usare BS maggiori di 1M, si deve aumentare il parametro di configurazione del cluster **maxblocksize** in modo opportuno, con il comando **mmchconfig** (**richiede cluster shutdown**)

Subblocks

- GPFS divide un block in 32 subblocks
  - il subblock e' la minima quantita' di spazio allocabile
  - ogni file occupera' un certo numero di blocchi interi, piu' un certo numero di subblocks.
    - le porzioni di blocco utilizzate parzialmente sono dette frammenti
    - le operazioni di creazione e rimozione continua di file genera un aumento della frammentazione
  - l'occupazione minima di spazio dati un file e' pari ad un subblock (cioe' 1/32 della BS)

Scelta della block size

- La scelta opportuna dipende da diversi fattori
  - **ottimizzazione dell'occupazione**: scelta da operare in funzione della dimensione media dei file: se ci sono molti file piu' piccoli della dimensione del subblock si spreca molto spazio
  - **ottimizzazione delle prestazioni** di I/O sul volume RAID: e' sempre opportuno utilizzare una block size che sia multiplo della stripe size dei volumi RAID sottostanti
  - ottimizzazione in funzione del **pattern di accesso** dell'applicativo
    - applicativi che fanno I/O **sequenziale** di **grandi file** possono avere ottime **prestazioni** usando BS grandi (>= 1MB)
    - per accesso **randomico** di **piccole quantita' di dati** (general file service), si ottimizza **l'occupazione** con una BS piccola (64-512 KB)

Block Allocation Map type

- Il parametro –j di mmcrfs controlla il **modo** in cui, all'interno di un disco, vengono scelti i blocchi da allocare
  - **cluster**: l'allocazione viene fatta cercando di mantenere adiacenti i blocchi dei dati di uno stesso file
    - adatto a cluster piccoli (e' default per cluster con meno di 8 nodi)
    - prestazioni leggermente migliori inizialmente
    - causa un degrado delle prestazioni con l'aumentare dell'utilizzo di spazio e col numero dei nodi attivi sul file system
  - **scatter**: l'allocazione dei blocchi viene fatta con scelta random
    - adatto a cluster di dimensioni non piccole
    - le prestazioni sono inizialmente inferiori rispetto al cluster, ma costanti nel tempo (media sulla posizione del settore)
    - la frammentazione e' gestita meglio
- Non puo' essere modificato dopo la creazione del file system
  - Il parametro puo' essere definito differentemente su diversi storage pool entro lo stesso file system

Numero di nodi con accesso concorrente al file system

- Su ciascun disco del file system viene creata una Block Allocation Map per indicare lo stato di utilizzo dei subblocks del disco (liberi/occupati)
- La Block Allocation Map viene creata in parti allocabili separatamente, cioe' contemporaneamente
  - quando un nodo chiede o rilascia blocchi, la porzione della Block Allocation Map coinvolta viene marcata locked, modificata, quindi rilasciata
  - il numero di parti in cui e' divisa definisce il grado di parallelismo nella allocazione/deallocazione di blocchi sul disco
- Questo parametro viene definito all'atto della creazione del file system tramite l'opzione **-n** di **mmcrfs**
  - questo valore puo' essere in seguito modificato con il comando **mmchfs**, ma la nuova modifica riguardera' **solo i dischi di storage pool creati successivamente**
  - e' meglio sovrastimare questo valore che sottostimarlo

Max e default replica factor

- Il numero **massimo** di repliche supportate per i dati (**MaxDataReplica**) e per i metadati (**MaxMetadataReplica**) sono definite tramite le opzioni **-R** e **-M** di **mmcrfs**
  - i possibili valori per entrambe sono 1 (non si possono avere repliche), 2 o 3 (si possono' avere fino a due repliche)
  - questi parametri non sono modificabili dopo la creazione del file system
- Il numero di repliche per dati e metadati creati **per default** (alla creazione di un nuovo file o di nuovi metadati) e' definita tramite le opzioni **-r** e **-m** di **mmcrfs**
  - entrambe possono valere al minimo 1, al massimo il valore MaxDataReplica o MaxMetadataReplica rispettivamente
  - entrambe possono essere modificate tramite il comando *mmchfs*
    - solo i dati e metadati creati dopo la modifica adotteranno la nuova configurazione
    - per replicare tutti i dati e metadati secondo il corrente fattore di replica, si deve utilizzare il comando *mmrestripefs* con le opportune opzioni

Altre opzioni

- **Deny-write open lock**: -D [posix | NFS4]
  - Definisce se un deny-write open lock debba bloccare write via NFS (NFS4 lo richiede, NFS3 no)
  - posix: per file system esportati via NFS v3 o non esportati
  - NFS4: per file system esportati via NFS v4, samba, o montati su nodi Windows
- **Mount allo startup**: -A [yes | no | automount]
- **Suppress atime update**: -S [no | yes]
  - la soppressione dell'update dell'access time riduce l'I/O sui metadati, ma puo' comportare problemi per policies basate sull'attributo ACCESS_TIME

Altre opzioni

- **Report exact mtime**: -E [no | yes]
  - yes: stat() e fstat() riportano il valore corretto
  - no: stat() e fstat() riportano il valore all'ultimo sync del client che modifica il file; in questo caso, possono esserci effetti non voluti per operazioni di backup o polocies che utilizzano l'attributo MODIFICATION_TIME
- **ACL type**: -k [posix | nfs4 | all]
  - posix per l'utilizzo di ACL tradizionali, nfs4 per il supporto di ACL NFS v4 o Windows
- **Strict replication**: -K [no | whenpossible | always]
  - no: se non puo' creare la replica, l'operazione di I/O non ritorna errore
  - whenpossibile: forza la replica se la configurazione dei dischi lo permette (abbastanza failure group)
  - always: forza sempre
    - la creazione del file o della directory fallisce se non e' possibile creare la replica quando deve essere forzata

Altre opzioni (cont.)

- **Mountpoint directory**: -T <mountpoint>
  - defnisce il mount point presso il quale il file system verra' montato
    - la directory viene creata allo startup di GPFS se necessario
  - e' lo stesso su tutti i nodi del cluster
- **Attivazione quota**: -Q [yes | no]
  - attiva il supporto per la gestione della quota (user, group, fileset)
- **Visualizzazione quota per fileset**: --filesetdf | --nofilesetdf
  - se e' abilitata la quota, il comando df mostra valori corrispondenti alla quota del fileset e non al file system complessivo
- **Inode**: --inode-limit MaxNumInodes[:NumInodesToPreallocate]
  - definisce il numero massimo di inode per il file system, ed opzionalmente quanti sono preallocati
  - puo' essere modificato con il comando *mmchfs*

Altre opzioni (cont.)

- **Block size per i metadati**: --metadata-block-size <size>
  - definisce la block size per il pool system
- **Scopo di user/group quota**: --perfileset-quota | --noperfileset-quota
  - definisce i limiti di quota per user e group a livello di singolo fileset o di tutto il file system
- **Mount priority**: --mount-priority Priority
  - permette di specificare l'ordine in cui montare i file system allo startup: file system con maggiore priority vengono montati dopo; zero indica nessuna priorita, ed i file system verranno montati per ultimi

File system management

Mount e dismount

- **mmmount**: mount del file system
  - supporto anche per il comando Unix **mount**, ma senza features GPFS
  - supporto per binding mount su sistemi chrooted
- **mmumount**: dismount del file system
  - supporto per il comando Unix **umount**, ma senza features GPFS
  - non si puo' smontare se il kernel ha riferimenti attivi a file nel file system (NFS export, file aperti, CWD di processi)
- **mmlsmount**: mostra chi monta il file system (anche nodi remoti, o internal mount)
  - gli internal mount possono essere dovuti a
    - **binding del file system (sistemi chrooted)**
    - **esecuzione di operazioni di movimentazione dei dati (ad esempio restripe)**

Rimozione del file system

- **# mmdelfs <filesystem-device>**
  - Il file system deve essere smontato
  - i dati non possono essere piu' recuperati
  - gli NSD del file system vengono nuovamente marcati come "available" per essere inseriti in un altro file system
- Se ci sono dischi non piu' accessibili, il comando fallisce
  - si deve utilizzare l'opzione -p per indicare di procedere ugualmente

File system attribute

- **# mmlsfs <device> ...**
  - visualizza tutti i parametri di configurazione del file system
  - puo' richiedere il singolo parametro (usando il relativo switch)
- **# mmchfs <device> ...**
  - modifica uno o piu' parametri del file system
  - la modifica di alcuni parametri richiede il dismount del file system su tutto il cluster
    - vedere la man page

mmfsck

- **mmfsck** permette di analizzare il file system e di operare le necessarie modifiche per correggere inconsistenze
- opera in modalita'
  - **online** (sconsigliata): si limita a recuperare blocchi allocati ma non usati
    - puo' capitare per allocazioni fallite per problemi di spazio disco, concomitanti con node failure
    - blocchi rimangono marcati allocati ma non sono realmente usati
    - altre inconsistenze sono riportate ma non corrette
  - **offline** (a file system smontato): fa cose analoghe a fsck
    - trova file orfani (blocchi allocati ma directory entry missing): lost+found
    - dir entry che punta a i-node free: rimuove la dir entry
    - incorrect link count: corregge
    - incorrectly formed dir entry (non corrisponde il generation number dell'i-node): rimuove la dir entry
    - ...

Occupazione del file system

- E' supportato il comando Unix "df"
- Esiste un comando GPFS: **mmdf**
  - visualizza l'occupazione **per disco**, **per storage pool**, e fornisce informazioni sui dischi (**failure group**, **store metadata**, ...)
  - visualizza anche l'occupazione di **i-nodes**
  - E' un comando che effettua I/O sui metadati del file system (eseguire con criterio): ci puo' mettere un po'
- **mmlspool** per un summary degli storage pool (senza i-node counting)

Disk management

Visualizzare i dischi del cluster

- **mmlsnsd**
  - Visualizza tutti i **dischi inizializzati** (NSD) del cluster
  - Permette di visualizzare le caratteristiche di configurazione degli NSD (**NSD server**, **file system** di appartenenza)
- **mmlsdisk**
  - Visualizza lo **stato corrente** dei dischi appartenenti ad un file system (**status**, **availability**, NSD server **attualmente** usato per l'accesso, **tipo di utilizzo**, **storage pool**)

Disk availability

- Lo stato di un disco e' la combinazione del **disk status** e **disk availability**
- La disk availability segnala se GPFS e' in grado di scrivere o leggere dal disco
  - **up**: condizione di normalita': il disco e' usato per operazioni r/w
  - **down**: il disco non e' utilizzato per operazioni di I/O
    - condizione automatica in conseguenza di errori di I/O ripetuti
    - e' una condizione permanente: si deve usare mmchdisk per modificarla
  - **recovering**: condizione transitoria tra lo stato down e lo stato up: GPFS sta' verificando il contenuto del disco prima di renderlo disponibile; e' permessa la scrittura, non la lettura
  - **unrecovered**: GPFS non ha potuto completare l'operazione di recovering

Disk status

- Il disk status controlla il data placement e la migrazione dei dati
  - **ready**: il disco e' in condizioni normali, ed usato per dati e/o metadati secondo la sua configurazione
  - **suspended**: il dati sul disco possono essere letti o aggiornati, ma non viene allocato nuovo spazio
  - **being emptied**: condizione transitoria per un disco in attesa di completare la sua rimozione
  - **replacing**: condizione transitoria per un disco mentre e' in corso la sua sostituzione
  - **replacement**: condizione transitoria per un disco che sostituisce un altro disco
- GPFS alloca spazio solo su dischi ready o replacement

Modificare l'availability di un disco

- **# mmchdisk <device> {stop|start} -d "<disk-descr>"**
  - **stop**: il disco viene messo "**down**" e non deve piu' essere utilizzato da GPFS
    - avviene anche automaticamente
    - il restart di GPFS non modifica lo stato "**down**"
    - non puo' essere utilizzato nemmeno per **mmfsck**
  - **start**: fa ripartire il disco "**down**"
    - il disco diviene "**recovering**", quindi "**up**" o "**unrecovered**"
    - il disco "**unrecovered**" puo' essere utilizzato per operare un **mmfsck**

Modificare lo status del disco

- **# mmchdisk <device> {suspend|resume} -d "<disk-descr>"**
  - **suspend**: istruisce GPFS di non allocare piu' spazio su questo disco
    - generalmente prima di rimuovere un disco: in questa condizione si puo' migrare via tutti i dati dal disco e rimuoverlo
    - lo status "suspended" **non viene modificato** nemmeno da GPFS restart: **solo manualmente**
  - **resume**: istruisce GPFS di rimettere in stato "ready" un disco precedentemente messo in "suspended"
    - il disco torna ad essere pienamente disponibile (se l'availability lo permette)

Modificare disk usage e failure group

- **mmchdisk** puo' essere utilizzato per modificare **disk usage** e **failure group** di appartenenza del disco:

  # mmchdisk <device> change -d "<disk-descr>"

  <disk-descr> e' una stringa del tipo:

  DiskName:::DiskUsage:FailureGroup:::

  che specifica i parametri desiderati
- Questo comando **non sposta dati**: se si modifica il failure group si deve eseguire un **mmrestripefs** per rimettere a posto le cose

mmrestripefs

- **mmrestripefs** viene utilizzato per migrare dati in funzione dello stato dei dischi e della replica di dati e metadati
  - **b: ribilancia tutti i dati e metadati su dischi non suspended (da eseguire dopo aggiunta/rimozione di dischi dal file system)**
  - **m: migra tutti i dati e metadati che non esistono altrove da dischi suspended e fa restripe**
    - quelli replicati altrove non li sposta
  - **r: sposta tutti i dati e metadati da dischi suspended e fa restripe**
    - ricrea le repliche, ed alla fine il disco e' vuoto
    - come -m se non c'e' replica di dati e metadati
  - **p: rimette a posto la collocazione dei dati di file ill placed (cioe' con blocchi nello storage pool errato)**
  - **R: modifica il replica setting di tutti i file al default del file system e crea o rimuove le repliche secondo la necessita'**
  - **b implica r (o m) e p**
- E' una operazione che genera molto I/O

Aggiunta di un disco al file system

- Il nuovo disco deve essere inizializzato come NSD (mmcrnsd), definendo failure group, disk usage e storage pool di appartenenza
  - i dischi inizializzati ma non ancora assegnati ad un file system possono essere visti tramite il comando **mmlsnsd -F**
- Il nuovo disco puo' essere aggiunto al file system con il comando

  # mmadddisk Device {"DiskDesc"| -F StanzaFile} [-a] [-r]

  dove DiskDesc e' una stringa del tipo:

  DiskName:::DiskUsage:FailureGroup::StoragePool

  - l'opzione -r richiede che il file system **ribilanci il suo contenuto** su tutti i dischi, compreso il nuovo (operazione I/O intensive)
  - l'opzione -a indica che il comando non deve aspettare la fine del ribilanciamento per ritornare

Considerazioni sulla aggiunta di un disco al file system

- E' opportuno mantenere **omogenea** la dimensione e la velocita' dei dischi all'interno di uno storage pool
- E' opportuno che la stripe size del volume RAID sia comunque un **sottomultiplo** della block size del file system
  - GPFS esegue singole I/O delle dimensioni della block size
  - se la block size non e' multiplo della stripe size avremo una inefficienza RAID level

Rimozione di un disco dal file system

- E' possibile rimuovere a caldo un disco dal file system:

  # mmdeldisk <device> "<disk-name>"

- I dati contenuti sul disco vengono **automaticamente migrati** su altri dischi dello stesso pool
  - usa mmdf per verificare la disponibilita' di spazio
- Si possono usare le opzioni (-b, -m, -r) di **mmrestripefs**
- In caso di fallimento, il disco rimane in stato suspended
  - si puo' rieseguire la rimozione dopo aver sistemato la causa dell'errore
- Se c'e' un disco disponibile, si puo' usare **mmrpldisk** (sostituisce il disco copiando il suo contenuto sull'altro)

Esercitazione

- https://wiki.ge.infn.it/calcolo/index.php/Corso_Cloud_Storage_Es2