# Introduzione a GPFS: il cluster

Alessandro Brunengo
Mirko Corosu
INFN-Genova

# Contenuto

- Introduzione
- Il cluster
- Note sulla installazione

# Credits

Registered trademarks of International Business Machines Corporation

- IBM®
- IBM Spectrum Scale™
- GPFS™

# Introduzione

- GPFS: General Parallel File System
  - prodotto proprietario di IBM
- Dalla release 4.1 cambia nome:
  IBM Spectrum Scale
  - Evidenzia il fatto che non e' solo un file system, ma un sistema integrato ed evoluto di data management

# Spectrum Scale features

- Scalabile (decine di PB, migliaia di nodi, miliardi di file)
- Multipiattaforma (Linux, AIX, Windows, Power)
- Parallele Network Shared file system
- Unico namespace, accesso Posix
- Journaled file system con file system check
- Separazione dati/metadati
- Replica indipendente di dati e metadati
- ILM, tramite raggruppamento di dischi (storage pool), partizionamento del namespace (fileset), e policy engine per placement e movimentazione dati, anche integrato con HSM (disk-to-tape)
- Efficiente scan di metadati
- Snapshot (copy on write) a livello di fileset
- Clone di file (copy on write)
- Soluzioni di performance per particolari workload (local readonly cache e high available write cache su SSD)
- Multicluster: separazione amministrativa tramite remote cluster access
- Active File Management: accesso a dati remoti (GPFS o NFS) via cache locale, stesso namespace
- File Placement Optimizer: controllo sulla collocazione di repliche in configurazione shared nothing
- Elastic Storage Solution: declustered raid sowftare
- Policy driven compression
- QoS per gestire system I/O vs user I/O
- Asynchronous Disaster Recovery solution
- Encryption
- Supporta accesso da client non GPFS attraverso diversi protocolli: NFS, Samba, Object (OpenStack Swift o Amazon S3 interface)
- Integrazione dell'accesso tramite file e object protocol
- Automatic failover di tutte le componenti: no single point of failure

# Il cluster

# Architettura clustered

- L'architettura di GPFS e' basata sul concetto di cluster
- I nodi del custer cooperano e condividono la conoscenza esatta della configurazione del file system
  - tutti i nodi sono potenzialmente in grado di svolgere qualsiasi funzione
  - non e' una architettura client-server
- Il cluster permette di implementare
  - mantenimento globale della coerenza di dati e metadati tramite la gestione distribuita delle funzioni di controllo (token, lock)
  - possibilita' di implementare meccanismi di failover in caso di failure di un qualsiasi componente
  - possibilita' di eseguire operazioni di management da qualsiasi nodo del cluster
- La consistenza del cluster e' garantita da un meccanismo di quorum

# Qualifica dei nodi: server vs client

- Il software installato sui nodi del cluster e' lo stesso indipendentemente dal ruolo ricoperto dal nodo
  - fa eccezione l'object service, che vedremo dopo
  - la configurazione distingue i nodi in termini di licenza e di funzonalita'
- Il licensing distingue
  - server: nodi con funzioni di management (cluster manager, quorum nodes, file system e token manager, NSD server) o che esportano i dati del file system attraverso applicativi (es: web server)
  - client: nodi che si limitano ad accedere (direttamente o indirettamente) ai dati tramite protocollo nativo GPFS

  Per visualizzare e cambiare il licensing: mmlslicense, mmchlicense
- La funzionalita' dei nodi si distingue in:
  - manager vs client: tra i manager vengono eletti i nodi con ruoli di management (devono avere licenza 'server')
  - quorum vs nonquorum

  Per visualizzare i ruoli della configurazione: mmlscluster, per modificarlil: mmchnode

# Ruoli dei nodi

- Quorum nodes: nodi preposti a definire il quorum per il cluster
  - GPFS su un nodo opera solo se sono visibili 'trunc(N/2)+1' quorum nodes
  - Opportuno che i quorum node siano in numero dispari
  - Esiste una configurazione che include uno o piu' tiebreaker-disk (non la vediamo)

  per visualizzare i quorum nodes: mmlscluster | grep quorum

  per aggiungere un quorum node: mmchnode –quorum –N <node-name>
- Cluster manager: uno per cluster, eletto tra i quorum nodes
  - gestisce le failure di nodi del cluster e il recovery
  - monitora i disk lease
  - elegge il file system manager
  - distribuisce informazioni di configurazione tra cluster remoti

  Per visualizzare il cluster manager: mmlsmgr –c

  Per modificarlo: mmchmrg –c <node-name>

# Ruoli dei nodi (cont.)

- File system manager: uno per file system, scelto dal cluster manager al momento del primo mount del file system
  - non e' indispensabile che il nodo sia configurato come manager
  - gestisce aggiunta/rimozione dei dischi, ed il loro stato
  - gestisce il repair del file system
  - gestisce l'allocazione
  - gestisce e controlla la quota
- Token manager: uno o piu' per file system, eletto tra i nodi manager
  - si occupa di coordinare l'accesso ai dati ed ai metadati dei dischi, in modo da garantire la consistenza del file system, tramite token

# Ruolo dei nodi (cont.)

- Metanode: uno per ciascun file aperto
  - qualunque nodo puo' assumere questo ruolo (di solito il nodo che ha aperto il file da piu' tempo)
  - ha il compito di garantire l'integrita' dei metadati del singolo file
- Cluster configuration server
  - Uno o due nodi sono configurati come responsabili del repository di riferimento della configurazione
    - la configurazione risiede su tutti i nodi, ma quella dei configuration server fa fede
  - Ruolo superato nelle nuove release (dalla 4.1): la configurazione viene mantenuta dai nodi quorum nel CCR (Cluster Configuration Repository)
    - la configurazione riguarda nodi, nsd, file system, object export configurazione dei parametri, …

# Note sulla installazione

# Comunicazione ssh tra i nodi del cluster

- GPFS opera in modo che tutti i nodi possano eseguire operazioni di management del cluster
  - si puo' configurare diversamente
- Questo richiede che tutti i nodi possano eseguire comandi ssh come utente root su tutti gli altri nodi del cluster
  - in aggiunta, il comando ssh non deve avere output o richiedere dati in input, come l'accettazione di una nuova chiave
- Questo requisito e' spesso considerato un buco di sicurezza
  - e' possibile utilizzare configurazioni per mitigare questa cosa
  - ma quando usi un cluster, con un software che ha accesso in kernel mode ai device, puoi mitigare solo apparentemente
- L'installazione deve quindi essere preceduta da una preparazione dei nodi
  - scambio di chiavi ssh e authorized_keys file
  - build e distribuzione del known_hosts file, o come alternativa, configurare su tutti i nodi sshd con il parametro "StrictHostKeyChecking = no"
- Va poi verificato che tutti i nodi da inserire in cluster possano eseguire comandi ssh privilegiati

```
# for i in $nodes; do for j in $nodes; do ssh $i "ssh $j hostname"; done; done
```

# User database

- Ogni nodo del cluster accede al file system con uid/gid del processo che effettua l'accesso
  - standard Posix
- Di norma si presume che i nodi di un cluster GPFS condividano lo user database
  - non e' obbligatorio, ma con UID/GID differenti si creano problemi sul controllo degli accessi ai file tra nodi diversi del cluster
- Per l'accesso remoto, GPFS supporta meccanismi di uid-remapping

# Software

- La release 4.2.0 e' disponibile in tre versioni
  - Express
  - Standard (quella licenziata per l'INFN, include CES)
  - Advanced (include l'encryption del file system)
- Il software viene distribuito in due pacchetti
  - Funzionalita' base del cluster (multipiattaforma):

    Spectrum_Scale_Standard-4.2.0.4-x86_64-Linux-install
  - Supporto dei Clustered Export Services: export del file system attraverso i protocolli NFS, Samba, Swift (solo per RHEL 7 e derivate):

    Spectrum_Scale_Protocols_Standard-4.2.0.4-x86_64-Linux-install
- La release 4.2 supporta (oltre a AIX e Windows) le distribuzioni linux: RHEL e derivate, SUSE, Debian, Ubuntu (limiti sulle versioni minime compatibili)

# Installazione manuale di base

- Preparare tutti i nodi del cluster
  - Creare e distribuire sui nodi
    - chiavi ssh per root
    - authorized_keys e knnown_hosts
    - /etc/profile.d/gpfs.sh (/usr/lpp/mmfs/bin nel PATH)
- Spacchettare la distribuzione base
  - Mette il necessario in /usr/lpp/mmfs/4.2.0.4
- Installare i pacchetti necessari
- Fare il build del Portability Layer
  - Usare il comando mmbuildgpl
  - Su derivate della RHEL e' necessario prima defnire:

```
# export LINUX_DISTRIBUTION=REDHAT_AS_LINUX
```

# Installazione automatizzata

- SpectrumScale 4.2 supporta una utility di deploy di GPFS su piattaforma RHEL
  - /usr/lpp/mmfs/4.2.0.4/installer/spectrumscale
- Questa installa il software, esegue il build del portability layer, crea il cluster e esegue una configurazione iniziale
  - supporta anche alcune modifiche di configurazione successive
- Noi non useremo questa utility
  - l'utility non e' ancora sufficientemente flessibile e richiede comunque configurazioni manuali
  - faremo l'installazione completamente manuale

# Esercitazione

- https://wiki.ge.infn.it/calcolo/index.php/Corso_Cloud_Storage_Es1