GPFS Object Storage

Alessandro Brunengo
Mirko Corosu
INFN-Genova

Contenuto

- Clustered Export Services
- Object Service in GPFS
- Setup di CES e dell'object service

Clustered Export Services

Clustered Export Services

- A partire dalla release 4.1 GPFS offre un meccanismo integrato (CES) per consentire file e object access attraverso diversi protocolli:
  - file access via NFS
  - file access via Samba
  - object access via Swift
- CES permette di accedere a dati ospitati sul filesystem GPFS da client non-GPFS
- CES integra il supporto ai protocolli di export con la flessibilita', scalabilita' e alta affidabilita' offerti da GPFS
  - sfrutta la tecnologia di clustering per identificare failure ed attivare meccanismi di failover

CES overview

- Si configura un pool di nodi del cluster (CES nodes, o protocol nodes) per fornire una soluzione di export dei dati GPFS tramite NFS, Samba o Object in soluzione di alta affidabilita'
  - il pool di CES nodes si indica come CES cluster
- Si definisce un pool di IP address (CES address pool) che viene distribuito tra i CES nodes
  - quando un nodo entra o esce dal CES cluster gli indirizzi del CES address pool vengono redistribuiti, mantenendo continuita' operativa
- Funzioni di monitoring sia interne al CES node che a livello di cluster consentono il controllo della funzionalita' dei CES node e delle applicazioni di export, ed innescano i meccanismi di failover disabilitando GPFS sul nodo in failure e migrando gli IP
  - meccanismi quali gratuitous ARP e invito al lock reclame permettono di operare il failover senza interrompere le sessioni

Load balancing in CES

- Per ottenere load balancing in CES, si puo' definire un DNS name virtuale che risolva in round-robin i diversi indirizzi del CES pool
- Lato client, l'accesso deve essere configurato verso il DNS name virtuale

CES: limiti e requisiti

- Limiti:
  - ciascun CES node esporta tutti i protocolli abilitati
  - se si usa Samba, c'e' un limite di 16 CES nodes
- Requisiti:
  - CES e' supportato solo su RHEL7 o derivate
  - ogni CES node deve poter ospitare tutti gli indirizzi del CES address pool
    - ogni CES node deve avere una interfaccia fisica o logica su ciascuna rete dei CES IP address
    - l'indirizzo primario della (delle) interfaccia dedicata al CES address pool non deve far parte del pool
    - i CES IP address devono risolvere via DNS o hosts file

Object Service in GPFS

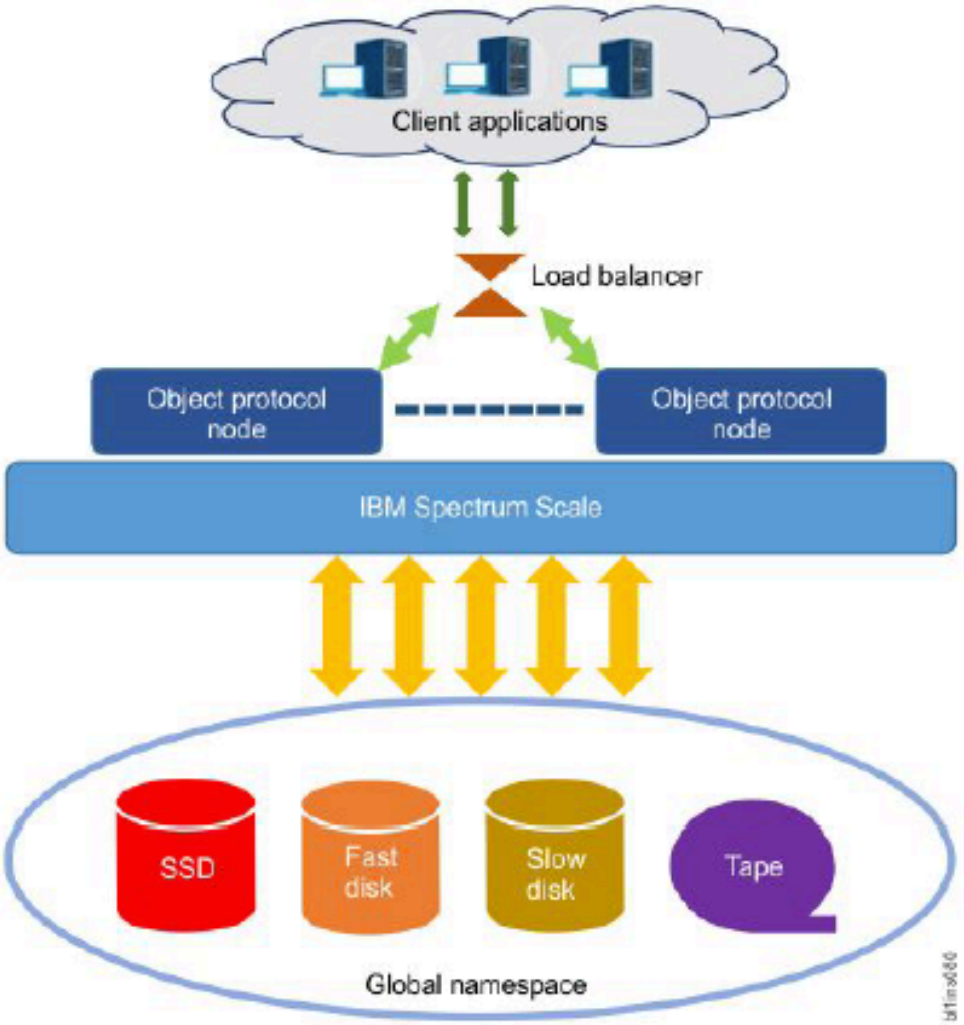
Object Service in GPFS

- L'object service su Spectrum Scale V4.2 e' implementato sulla base della versione di Swift in OpenStack Kilo
- A supporto, Spectrum Scale V4.2 implementa un servizio keystone
  - supportata la sola interfaccia API V3

Object Service features

- Spectrum Scale Object Storage Service supporta la gerarchia ordinaria di Swift
  - account (project, o tenant)
  - container (contenitore di oggetti definiti all'interno dell'account, su cui si possono applicare ACL)
  - oggetti (esistono all'interno di un container, ed ereditano i suoi ACL)
- Sono supportate le seguenti features
  - unified file and object access
  - accesso tramite interfaccia S3
  - multi-region object storage
  - compression (tramite policy)

GPFS arch. per object service



Implementazione

- Tutti i protocol node vedono il file system, quindi tutti gli oggetti
  - Non e' necessario configurare repliche in swift
  - Non e' necessario copiare dati tra i server per fare rebalancing
- L'alta affidabilita' ed il load balancing sono forniti
  - per il dato su disco, da fattori di replica del file system
  - per l'accessibilita' del dato, dai meccanismi di high availability di CES
- I ring sono costruiti tramite virtual device, che sono subdirectory nel fileset di Swift
- Nella costruzione degli object ring, i virtual device sono inseriti al localhost node, in modo che tutti i nodi accedano a tutti i virtual device
- Vengono utilizzati diversi virtual device per contenere le collisioni di accesso alla directory stessa nell'upload/rimozione degli oggetti
- Alcuni servizi di Swift possono quindi girare su un singolo nodo
  - object-replicator (cleanup di oggetti rimossi)
  - object-updater (update container listing)
  - …

Setup di CES e dell'object service

CES cluster and object service setup

- Il setup dell'object service richiede:
  1. installare il software per il protocol service sui nodi del CES cluster, ed inserire i nodi nel cluster
  2. configurare la cesSharedRoot
  3. abilitare i protocol nodes
  4. configurare i CES IP address
  5. configurare la policy di assegnazione degli indirizzi del CES pool
  6. configurare Swift (solo per OBJ protocol)
  7. abilitare i protocolli desiderati
  8. configurare la user authentication
     - per OBJ: gia' fatto con la configurazione di Swift

Installazione del software per il protcol service

Sui nodi che faranno parte del CES cluster:

- spacchettare il software Spectrum_Scale_Protocols_Standard-4.2.0.4-x86_64-Linux-install
- vengono creati repository in /usr/lpp/mmfs/4.2.0.4/:
  - gpfs_rpms (GPFS base)
  - ganesha_rpms (NFS)
  - smb_rpms (Samba)
  - object_rpms (Swift)
- Creare repository locali per utilizzare yum: creare un file /etc/yum.repos.d/gpfs.repo, del tipo

```
[gpfs-base]
name = gpfs-base
baseurl = file:///usr/lpp/mmfs/4.2.0.4/gpfs_rpms/
enabled = 1

[object-rpms]
...
```

Installazione del software per il protcol service (cont.)

- Installare i pacchetti
  - gpfs.base gpfs.docs gpfs.ext gpfs.gpl gpfs.gskit gpfs.msg gpfs.protocols-support
  - nfs-ganesha-gpfs nfs-ganesha-utils
  - gpfs_smb
  - spectrum-scale-object
- Inserire i nodi nel cluster e fare start di GPFS
  - mmaddnode –N <node-list> (su nodo in cluster)
  - mmstartup –N <node-list>

CES shared root

- Il cluster CES deve avere una directory (cesSharedRoot) in cui ospitare:
  - CES shared configuration data
  - informazioni per il protocol recovery
  - informazioni specifiche dei protocolli (es: keystone database)
- La cesSharedRoot deve risiedere su GPFS
  - possibilmente un filesystem dedicato
- E' quindi necessario creare la directory e configurare il parametro del cluster

```
# mmchconfig cesSharedRoot=<dir-name>
```

Abilitare i CES nodes

- Per inserire i nodi nel pool dei CES nodes:

  # mmchnode --ces-enable –N <node-list>

- Visualizzare il CES cluster:

  # mmlscluster --ces

Configurare il CES address pool

- Aggiungere gli indirizzi IP del CES address pool

  # mmces address add –ces-ip <ip>

- Visualizzare il CES address pool

  # mmces address list

- Rimuovere un indirizzo dal pool:

  # mmces address remove –ces-ip <ip>

IP distribution policy

- E' possibile configurare la policy per la distribuzione degli indirizzi sui CES node:

  # mmces address policy <policy>

  - even-coverage: distribuzione simmetrica per numero di indirizzi (default)
  - balance-load: distribuzione simmetrica per carico
  - node-affinity: assegnazione dell'indirizzo ad un nodo specifico (se possibile)
- E' possibile migrare manualmente indirizzi, o attivare manualmente un ribilanciamento

  # mmces address move …

Implementazione degli indirizzi del pool CES

- Gli indirizzi del pool sono assegnati tramite alias
  - l'interfaccia fisica dei CES node deve avere una interfaccia con indirizzo sulla stessa rete IP dei CES IP address, ma questo indirizzo non deve far parte del pool
  - l'indirizzo della interfaccia puo' essere l'indirizzi di management o di comunicazione del cluster GPFS
- Vantaggi della gestione tramite alias
  - il link layer e' gia' attivo, e controllato dal monitoring
  - IP failover molto piu' rapido
  - Separazione tra indirizzi virtuali (che migrano) ed indirizzi fisici (che restano sempre assegnati al nodo)

Configurazione di Swift

- Per il solo protocollo OBJ e' necessario effettuare la configurazione iniziale di Swift prima di abilitare il protocollo

  # mmobj swift base…

- Questo comando genera la configurazione iniziale per l'object service (Swift) e keystone (se utilizzato)

keystone locale

- L'object service utilizza un keystone service per l'autenticazione e l'autorizzazione (account/container/object e user)
- GPFS 4.2 integra un keyston service locale (solo API V3)
  - il keystone service viene esportato anch'esso dalla infrastruttura CES
  - sfrutta CES per fornire un servizio HA
  - il database e' postgress, e risiede nella cesSharedRoot
  - i file di configurazione di keystone, come quelli di CES, sono gestiti dal CCR (cluster configuration repository)

keystone remoto

- L'implementazione di Swift su GPFS puo' autenticare ed autorizzare su un keystone service remoto
  - puo' essere un keystone preesistente, ad esempio di una installazione OpenStack
  - e' supportata solo l'API V3
- In questo caso, la gestione di user e project viene fatta sul keystone remoto

mmobj swift base …

- Il comando di configurazione richiede i seguenti parametri
  - mount point del file system GPFS
  - nome del fileset su cui opera swift
    - swift utilizza un independent fileset, che verra' creato nella root del filesystem specificato
    - il fileset non deve esistere
  - nome DNS di accesso all'endpoint di swift (e keystone, se locale)
    - si deve utilizzare il DNS name che fa load balancing sui CES IP address
  - opzioni per accesso S3, file-access, multi-region (vedremo piu' avanti)

mmobj swift base… (cont.)

- Se si utilizza keystone locale:
  - password del database
  - user/password per admin
  - user/password per swift service
  - questi parametri vengono usati per il db di keystone
- Se si utilizza un keystone remoto:
  - URL dell'endpoint remoto
  - user/password per il servizio swift
  - opzione per configurare il keystone remoto
    - in questo caso serve specificare user/password di admin remoto
  - altrimenti il keystone remoto deve contenere
    - user swift e relativa password, come specificate dal comando
    - service swift, ed assegnare role admin allo user swift per il service
    - endpoint del service swift, che punti al CES IP address

Abilitazione e disabilitazione dei protocolli

- Per abilitare il protocol service:

  # mmces service enable [NFS|SMB|OBJ]

  - Nota: per il solo OBJ protocol, prima di eseguire questo comando si deve configurare Swift (vedi dopo)
- Per disabilitare un protocollo:

  # mmces service disable [NFS|SMB|OBJ]

  - Attenzione: disabilitare un protocollo comporta perdere tutte le configurazioni del protocollo stesso
    - un nuovo "enable" genera una configurazione di default
  - Nel caso di OBJ, significa perdere accesso ai dati gia' serviti dal protocollo (cleanup della confiugrazione di Swift)

Configurazione dello userauth

- Dopo l'ablitazone dei protocolli e' necessario configurare l'autenticazione e autorizzazione per i protocolli

  # mmuserauth service create …

- Per OBJ, il comando mmobj swift base configura una userauth iniziale, che puo' ora essere eventualmente cambiata

Autenticazioni supportate

- I protocol service supportano i seguenti backend di autenticazione
  - AD, LDAP (file, object)
  - NIS (NFS)
  - local keystone (object)
  - userdefined (file, object)
    - questo deve essere usato in caso di keystone remoto
- In generale, la configurazione di userauth mantiene al suo interno un mapping degli ID (tra ID del backend e ID locale, trasparente)
  - Il mapping puo' essere configurato

Object userauth

- Nel corso non vedremo utilizzo di backend di autenticazione AD o LDAP per l'object service
- La configurazione di swift genera una configurazione di userauth idonea
  - per le esercitazioni, non e' necessario configurarla nuovamente

Delete della configurazione di userauth

- Per rimuovere la configurazione di userauth:

```
# mmuserauth service remove \
    --data-access-method object|file
# mmuserauth service remove \
    --data-access-method object|file –idmapdelete
```

  - il primo comando rimuove la configurazione di server remoto, SSL, binding user, ..
  - il secondo comando rimuove l'ID user mapping
  - attenzione: non e' piu' possibile ripristinare l'accesso ai dati per gli user precedentemente definiti e mappati
- La rimozione di userauth e' necessaria se si vuole riconfigurarla con modalita' differenti

Esercitazione

- https://wiki.ge.infn.it/calcolo/index.php/Corso_Cloud_Storage_Es5