Introduzione a GPFS: Features (alcune)

Alessandro Brunengo
Mirko Corosu
INFN-Genova

Contenuto

- Fileset
- Snapshot
- Cloni

Fileset

Fileset

- GPFS supporta il concetto di fileset, che e' sostanzialmente un partizionamento del namespace del file system che dal punto di vista amministrativo si comporta come un file system indipendente
  - e' possibile definire una quota per fileset
  - e' possibile definire user/group quota per fileset
  - si possono definire policy di collocazione e movimentazione di file per fileset
  - si possono creare snapshot di singoli fileset
- Il fileset e' identificato da una stringa di caratteri che deve essere univoca all'interno del file system

Fileset e i-node space

- Il fileset puo' essere
  - indipendente: lo spazio di i-node e' dedicato al fileset
    - ottimizza funzioni di scan dei file di un fileset, ad esempio nella applicazione di policy
    - rende possibile la creazione di snapshot dedicate
  - dipendente: lo spazio di i-node e' quello del file set *root* o di un altro fileset indipendente
- Alla creazione del file system viene automaticamente creato il fileset *root*
  - non puo' essere cancellato
  - la radice del fileset *root* coincide con la root del file system
  - contiene file di sistema, come i file di quota
    - nelle ultime versioni i file di quota stanno nel CCR

Junction del fileset

- Il fileset creato contiene una root directory vuota, e non e' visibile
- la accessibilita' del fileset viene realizzata creando un junction point all'interno del *root* fileset o di un fileset visibile
  - la junction ha l'aspetto di una normale directory (comprese le permission) ma non si possono eseguire le operazioni di rmdir e unlink su di essa

Creazione del fileset

- mmcrfileset <dev> <fs-name> [--inode-space <spec>] [-p <afm-attribute>...]
  - --inode-space new: fileset indipendente
    - si puo' specificare lo spazio di i-node per il fileset
  - --inode-space <existing-fileset>: crea un fileset dipendente che condivide l'i-node space con il fileset specificato

Link/unlink il fileset

- mmlinkfileset <dev> <fs-name> -J <junction path>
  - rende visibile il fileset posizionando la sua root sotto <junction path>, che viene creato col comando
- mmunlinkfileset <dev> {<fs-name>|-J <junction-path>} [-f]
  - rende il fileset invisibile
    - i file vengono conservati (ed i blocchi restano allocati!)
    - -f per forzare l'operazione, che fallisce se esistono file aperti dentro il fileset

Altre operazioni sul fileset

- mmlsfileset <dev> …
  - visualizza le caratteristiche di uno o di tutti i fileset
  - vedere la man page
- mmchfileset <dev> <fs-name> …
  - modifica i parametri del fileset (nuova junction point, i-node space per fileset indipendenti, attributi AFM)
- mmdelfileset <dev> <fs-name>
  - rimuove il fileset e tutti i dati contenuti

Snapshot

Snapshot

- GPFS supporta la creazione di **snapshot di un intero file system o di un indipendent fileset**
- La snapshot e' readonly
  - e' la **fotografia** del file system o fileset al momento della creazione della snapshot
  - puo' essere utilizzata da programmi di backup per ottenere **backup consistenti** durante le normali operazioni di I/O degli utenti
  - puo' essere usata come area di backup per **recupero rapido** di file perduti o per realizzare confronto di file con versioni vecchie

Copy on write

- Lo storage necessario alla snapshot viene preso **dai blocchi del file system** (tramite copy-on-write)
  - inizialmente la snapshot non occupa spazio su disco
  - ad ogni blocco modificato sul file system dopo la creazione della snapshot, il vecchio blocco viene mantenuto sul disco come parte della snapshot
- Il contenuto della snapshot mantiene **tutte le caratteristiche** dei file (**permission**, **ACL**, **attributes**)
- Le snapshot possono essere oggetto di applicazione di policy
  - in questo caso le operazioni definite vengono applicate ai file della snapshot
  - va ricordato che la snapshot e' readonly

Creare una snapshot

- Il comando per creare una snapshot e'

  # mmcrsnapshot <dev> <snap-name> [-j <fileset>]

- Possono essere create fino a 256 snapshot contemporanee su un file system

```
# mmcrsnapshot /dev/home_dev testsnap

Writing dirty data to disk
Quiescing all file system operations
Writing dirty data to disk again
Resuming operations.
#
```

Accesso al contenuto della snapshot

- Il contenuto della snapshot e' accessibile in

  /<fs-mount-point>/.snapshots/<snap-name>/
  /<fset-junction>/.snapshot/<snap-name>/

- E' possibile (tramite comando **mmsnapdir**) creare un link **.snapshots** in ogni directory del file system, in modo da accedere alla snapshot di ogni directory tramite il link

  /<dir>/.snapshots/<snap-name>/

  - le directory .snapshots **non sono visibili** tramite ls, ma e' possibile listare il loro contenuto o attraversarle e discendere nel sottoalbero della snapshot (cd …)

Gestione delle snapshot

- Si possono visualizzare le snapshot definite tramite il comando **mmlssnapshot**
  - e' possibile visualizzarne anche l'occupazione di dati e metadati
- Si rimuove una snapshot tramite il comando

```
# mmdelsnapshot <device> <snap-name> [-j <fileset>]
```

  - viene liberato tutto lo storage occupato dalla snapshot, che non sara' piu' accessibile

Cloni

Cloni

- Un clone e' una snapshot scrivibile di un singolo file
- Creare un clone e' una operazione simile alla copia, ma piu' efficiente
  - il file clone viene creato immediatamente, ma non viene allocato spazio finche' la copia originale o il clone non vengono modificati (copy on write)
- Esempi di utilizzo:
  - provisioning di virtual machine tramite la creazione del virtual disk di base
  - cloning del disco di una VM come parte del processo di creazione di una snapshot individuale a scopo di backup

Creazione di un clone

- La creazione di un clone avviene in due step
  - Si crea una snapshot readonly del file da clonare (che viene definita "parent")
  - Si crea un clone dalla snapshot (che viene definito "child")

    # mmclone snap <orig> <orig.snap>
    # mmclone copy <orig.snap> <orig.clone>

  - Al termine della procedura avremo sul file system un file snapshot readonly (<orig.snap>) che occupa lo spazio disco con i blocchi originariamente appartenenti al file <orig>, e due file modificabili (<orig> e <orig.clone>) che inizialmente non occupano spazio sul file system
- Se si omette di specificare il nome dello snapshot file da creare, mmclone snap trasforma il file <orig> in una snapshot readonly, da cui sara' possibile creare cloni.

Rimozione di un clone

- Il parent non puo' essere rimosso finche' esistono suoi cloni ("figli")
- La rimozione dei cloni, e del clone-parent, si attua tramite il comando "rm"
- Visualizzazione dello stato di clone:

  # mmclone show <file>

  mostra le caratteristiche di <file> in relazione al cloning

Esempio 1

```
# ls -lis *
537122052 4096 -rw-r--r-- 1 root root 4194304 Sep 26 18:04 orig

# mmclone snap orig orig.snap
# ls -lis *
537122052    0 -rw-r--r-- 1 root root 4194304 Sep 26 18:04 orig
537122053 4096 -rw-r--r-- 2 root root 4194304 Sep 26 18:04 orig.snap

# mmclone show *
Parent  Depth  Parent inode   File name
------ ----- ------------- ---------
   no     1     537122053  orig
  yes     0                orig.snap
```

Esempio 1 (cont.)

```
# mmclone copy orig.snap orig.clone
# ls -lis *
537122052    0 -rw-r--r-- 1 root root 4194304 Sep 26 18:04 orig
537122054    0 -rw-r--r-- 1 root root 4194304 Sep 26 18:05 orig.clone
537122053 4096 -rw-r--r-- 3 root root 4194304 Sep 26 18:04 orig.snap

# mmclone show *
Parent  Depth  Parent inode   File name
------ ----- ------------- ---------
   no     1     537122053  orig
   no     1     537122053  orig.clone
  yes     0                orig.snap
```

Esempio 2

```
# ls -lis *
537122052 4096 -rw-r--r-- 1 root root 4194304 Sep 26 18:04 orig

# mmclone snap orig
# ls -lis *
537122056 4096 -rw-r--r-- 1 root root 4194304 Sep 27 11:46 orig

# mmclone show *
Parent  Depth  Parent inode   File name
------ ----- ------------- ---------
  yes     0              orig

# echo ok >> orig
-bash: orig: Read-only file system
```

Esempio 2 (cont.)

```
# mmclone copy orig orig.clone
# ls -lis *
537122056 4096 -rw-r--r-- 2 root root 4194304 Sep 27 11:46 orig
537122057    0 -rw-r--r-- 1 root root 4194304 Sep 27 11:48 orig.clone

# mmclone show *
Parent  Depth  Parent inode   File name
------ ----- ------------- ---------
  yes     0                 orig
   no     1      537122056  orig.clone
```

Attributi del clone

- Il file clone e' un file distinto dal file clonato
  - ha un proprio i-node
  - ha proprie ownership e permission, indipendenti dal parent
    - creare un clone richiede solo read access sul parent
  - gli extended attributes non sono migrati dal parent al clone
- Puo' avere diversa policy di replica o pool placement
  - le differenti policy si applicano solo ai blocchi modificati dopo la sua creazione

Clone di cloni: clone depth

- E' possibile creare cloni di file che sono cloni di altri file
- Si genera una dipendenza gerarchica:
  - ogni file ha blocchi conservati in uno dei parent della catena, fino al parent di livello zero
  - nessun parent puo' essere rimosso se esistono suoi child
- La clone depth indica il numero di livelli di questa catena

Esempio (clone depth)

```
# ls -lis *
537122058 4096 -rw-r--r-- 1 root root  4194304 Sep 27 12:01 orig

# mmclone snap orig
# mmclone copy orig level1
# dd if=/dev/zero bs=1M count=1 >> level1
# mmclone snap level1
# mmclone copy level1 level2
# dd if=/dev/zero bs=1M count=2 >> level2
# ls -lis *
537122051 1024 -rw-r--r-- 2 root root 5242880 Sep 27 12:02 level1
537122059 2048 -rw-r--r-- 1 root root  7340032 Sep 27 12:03 level2
537122058 4096 -rw-r--r-- 2 root root  4194304 Sep 27 12:01 orig

# mmclone show *
Parent  Depth   Parent inode   File name
------ -----  -------------  ---------
   yes     1      537122058  level1
    no     2      537122051  level2
   yes     0                 orig
```

Riassumendo

- Il file clone-parent e' immutable e non puo' essere modificato
- Per rimuovere un clone-parent si devono prima rimuovere o disassociare tutti i file clone che hanno lui come parent (diretto o indiretto)
- Si possono fare cloni di cloni (vedere l'output di *mmclone show* per la depth)

Separazione di un clone dal parent

- Due modi:
  - mmclone redirect: separa il clone dal clone-parent diretto
    - il clone parent diretto puo' essere rimosso
    - il file rimane un clone, il cui parent e' il clone-parent del vecchio parent
  - mmclone split: separa il clone da tutta la catena di clone-parent
    - il clone diviene un file ordinario

Cloni di snapshot

- Si puo' creare il clone di un file appartenente ad una snapshot
  - in questo caso non e' necessario il comando mmclone snap: il file originale e' gia' idoneo ad essere clone-parent, in quanto parte di una snapshot
  - la snapshot (tutta) non puo' essere rimossa finche' esistono child di file della snapshot

Cloni e fileset

- La capacita' di gestire il file cloning e' limitata al boundary di un independent fileset
  - il reindirizzamento dei blocchi di clone e parent deve necessariamente riguardare blocchi e i-nodes all'interno dello stesso independent fileset
- Spostare un child su fileset differente equivale ad eseguire prima uno split, e poi spostarlo
  - il file rinominato non sara' piu' un child, ed occupera' lo spazio disco fisico con tutti i suoi blocchi

Esercitazione

- https://wiki.ge.infn.it/calcolo/index.php/Corso_Cloud_Storage_Es3