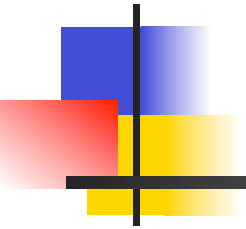
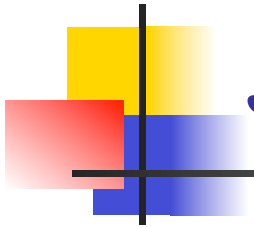


Advanced Features: subnet, snapshot, cloning

Alessandro Brunengo INFN-Genova



Subnet

GPFS network communication (I)

- Un nodo del cluster deve comunicare con
  - il cluster manager: allo startup ed in conseguenza di eventi relativi al cluster
    - modifica di configurazioni
    - spostamento di ruoli di servizio
    - join-disjoin di nodi
  - il file system manager: al mount o dismount, ed eventi relativi al file system (logs, allocazione di blocchi, recovery, check di quota, …)
  - gli NSD server (se necessario): per l'I/O di dati
  - lock manager e metanode server (potenzialmente tutti i nodi del cluster): per funzioni di lock di dati e metadati e funzioni interne
- Il path di comunicazione tra un nodo e gli altri viene stabilito allo startup e deve rimanere attivo per tutta la durata delle operazioni del nodo sul cluster

GPFS network communication (II)

- Comandi di amministrazione
  - i comandi di amministrazione possono essere processati su un singolo nodo o su piu' nodi, potenzialmente tutti
  - il nodo su cui viene dato il comando invia comandi e parametri ai nodi che devono eseguirlo via socket
- GPFS permette di specificare un indirizzo separato per il traffico di management dei nodi
  - mmchnode --admin-interface=<name> -N <node-name>

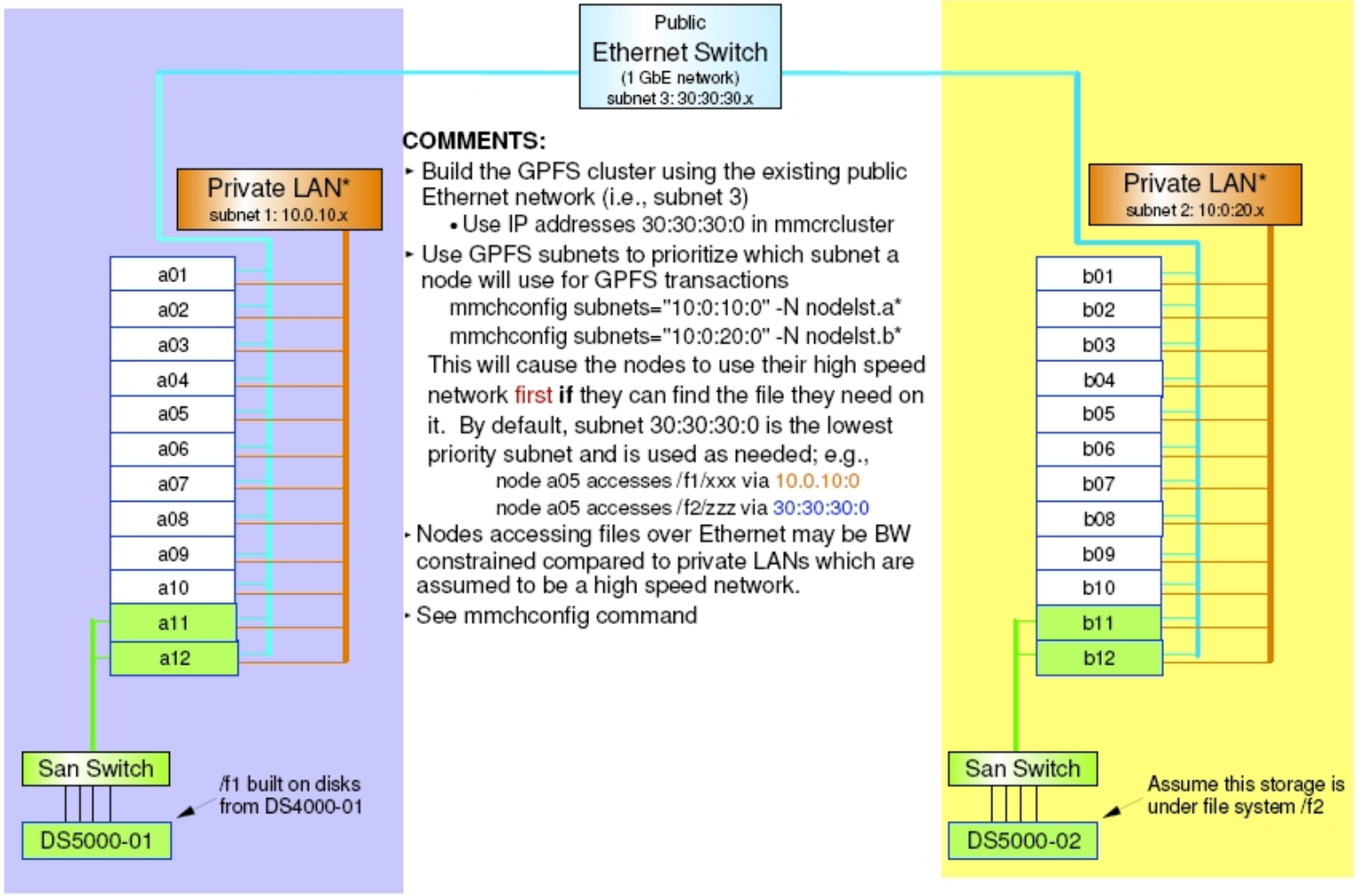
Subnet

- Il parametro di configurazione *subnets* permette di configurare reti alternative con cui raggiungere i nodi del cluster, in modo prioritario
- Se tutti o alcuni dei nodi del cluster sono connessi tramite piu' reti di differente prestazione, si puo' configurare GPFS per utilizzare, se possibile, la rete piu' performante

Configurazione di subnet

- mmchconfig subnets="<subnet-list>" [-N <node-list>]
  - subnet-list: <ip-net1> <ip-net2> …
- Questo comando fa si che i nodi provino a stabilire la conessione con altri nodi prima su <ip-net1>, poi su <ip-net2>
  - Se la rete della interfaccia con cui e' stato configurato il nodo nel cluster non e' presente nella lista di subnets, questa rete e' aggiunta per defaut alla fine dell'elenco
- La configurazione vale per tutto il cluster o per un sottoinsieme del cluster (specificando –N <node-list>)

GPFS Subnets



COMMENTS:

- Build the GPFS cluster using the existing public Ethernet network (i.e., subnet 3)
  - Use IP addresses 30:30:30:0 in mmcrcluster
- Use GPFS subnets to prioritize which subnet a node will use for GPFS transactions
  - mmchconfig subnets="10:0:10:0" -N nodelst.a*
  - mmchconfig subnets="10:0:20:0" -N nodelst.b*

  This will cause the nodes to use their high speed network first **if** they can find the file they need on it. By default, subnet 30:30:30:0 is the lowest priority subnet and is used as needed; e.g.,
  - node a05 accesses /f1/xxx via 10.0.10:0
  - node a05 accesses /f2/zzz via 30:30:30:0
- Nodes accessing files over Ethernet may be BW constrained compared to private LANs which are assumed to be a high speed network.
- See mmchconfig command

* The private LAN is generally a high speed switch; e.g., IB, 10Gb Ethernet, etc.
Assume nodelst.a contains the nodes a01-a12 and nodelst.b contains nodes b01-b12

Path di connessione (I)

- GPFS startup:
  - il nodo fa una lista degli IP delle proprie interfacce di rete che fanno il match con una delle subnets definite per il cluster
  - se necessario aggiunge in fondo alla lista l'indirizzo utilizzato quando il nodo e' stato aggiunto al cluster
- Cluster join:
  - il nodo comunica la lista dei propri indirizzi validi al cluster manager, che la inoltra a tutti i nodi del cluster

Path di connessione (II)

- Prima connessione con un altro nodo:
  - il nodo che attiva la connessione cerca nella lista degli indirizzi della destinazione quelli che fanno parte di una rete IP a cui appartiene uno dei propri indirizzi.
  - se trova piu' indirizzi utilizzabili, li ordina in priorita' secondo la lista
  - alla fine della lista, se mancante, aggiunge l'ultimo indirizzo della lista del destinatario (l'indirizzo pubblico, usato nella configurazione del nodo stesso)
- Scelta del path di connessione:
  - il nodo tenta di stabilire una connessione utilizzando la lista costruita come detto. Al primo successo viene definito l'indirizzo come path di comunicazione verso quel nodo

Path di connessione (III)

- una volta scelto l'indirizzo con cui contattare un nodo (il path), questo viene mantenuto fino allo shutdown di GPFS
  - non c'e' processo di failover: se si perde la connessione verso l'indirizzo del path, i due nodi non possono comunicare anche in presenza di altri path operativi (ed uno dei due nodi viene estromesso dal custer)
- Visualizzazione dei path di connessione
  - per visualizzare quali sono i path di connessione con i nodi del cluster:

    mmdiag --network

Subnets in multi-cluster

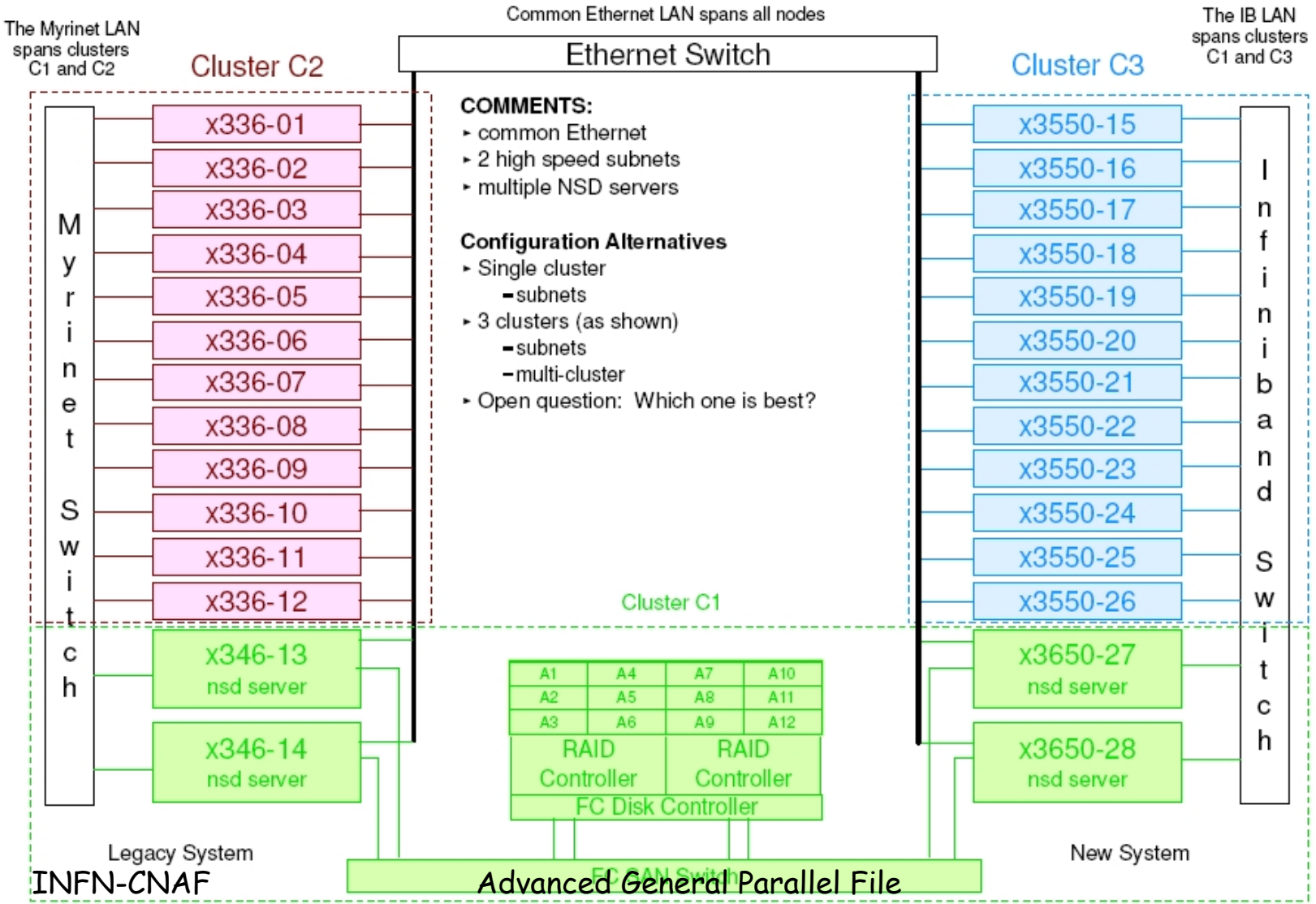
- Il parametro di configurazione subnets configura una sequenza di reti IP preferenziali per accedere ai nodi del cluster
  - quando due nodi si contattano utilizzano gli IP corrispondenti al nome con cui si conoscono (tipicamente IP pubblici)
  - dopo si scambiano tutti gli IP che hanno, per vedere se esiste una subnet configurata in comune: se c'e', la usano
  - la scelta e' definitiva: in caso di failure di una connessione di rete non si cerca di tornare indietro su un'altro indirizzo
- E' possibile configurare tale parametro anche per accedere a nodi di cluster remoti:

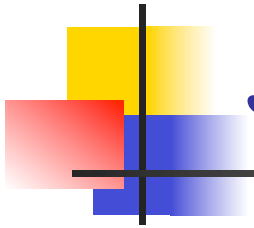
  # mmchconfig subnets="10.10.10.0/cl1;cl2"

  indica di utlilzzare quella rete per connettersi con i nodi dei cluster cl1 e cl2
  - se si configurano reti IP private per remote cluster, GPFS assume che siano reti disgiunte
  - per operare tra cluster remoti su una rete privata connessa, la subnet va configurata specificandi i due cluster name assieme, come nell'esempio

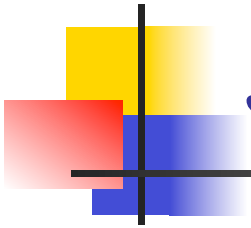
Subnet *vs.* Multi-Cluster

Combining Subnets with Multi-Clusters to Support Multiple Fabrics





Snapshot



Snapshot

- GPFS supporta la creazione di **snapshot di un intero file system o di un fileset** (a partire dalla 3.5)
- 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, o come area di backup per **recupero rapido** di file perduti o per realizzare confronto di file con versioni vecchie

Snapshot

- Lo storage necessario alla snapshot viene preso **dai blocchi del file system** (copy-on-write)
- 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 (**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 listarne il contenuto o attraversarle con cd

Gestione delle snapshot

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

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

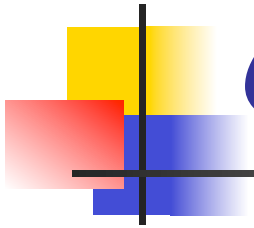
  - viene liberato tutto lo storage occupato dalla snapshot, che non sara' piu' accessibile

Recovery del file system da snapshot

- E' possibile eseguire un restore del file system a partire da una snapshot globale:

  # mmrestorefs <device> <snap-name>

  - il file system deve essere smontato su tutti i nodi
  - le snapshot non sono coinvolte nel processo di restore, quindi rimangono visibili anche dopo il restore di una snapshot vecchia



Cloning

Clone file

- Un clone file e' una snapshot scrivibile di un singolo file
- Creare un clone e' una operazione simile alla copia, ma piu' efficiente
  - il 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

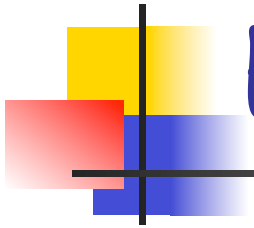
Creazione di un clone

- La creazione di un clone avviene in due step
- Creazione di una snapshot readonly di un file, che diviene il clone parent del clone che si sta' creando

  # mmclone snap <source-file> [<clone-parent-file>]

  <clone-parent-file> e' una copia readonly di <source-file>
  Se non si specifica la destinazione, <source-file> diviene il clone parent (diventa un file readonly)
- Creazione del clone a partire dal clone parent

  # mmclone copy <clone-parent-file> <clone-file>



Esempio

- **# dd if=/dev/zero of=orig_file bs=1M count=100**
- **# ls -lis**
- total 101376
- 513034 101376 -rw-r--r-- 1 root root 104857600 Dec 12 01:04 orig_file

- **# mmclone snap orig_file orig_file.clone_parent**
- **# ls -lis**
- total 102400
- 513034 0 -rw-r--r-- 1 root root 104857600 Dec 12 01:04 orig_file
- 513035 102400 -rw-r--r-- 2 root root 104857600 Dec 12 01:05 orig_file.clone_parent

- **# mmclone copy orig_file.clone_parent orig_file.clone**
- **# ls -lis**
- total 102400
- 513034 0 -rw-r--r-- 1 root root 104857600 Dec 12 01:04 orig_file
- 513029 0 -rw-r--r-- 1 root root 104857600 Dec 12 01:05 orig_file.clone
- 513035 102400 -rw-r--r-- 3 root root 104857600 Dec 12 01:05 orig_file.clone_parent

Rimozione di un clone

- Il file clone-parent e' immutable e non puo' essere modificato
- Il clone-parent non puo' essere rimosso se esistono suoi cloni
  - error: read only file system (misleading)
- Per rimuovere un clone-parent si devono prima rimuovere o disassociare tutti i file clone che lo hanno come parent
- La rimozione dei cloni, e del clone-parent, si attua tramite il comando "rm"

Visualizzazione

- Si possono fare cloni di cloni (vedere l'output di *mmclone show* per la depth)
  - si genera una gerarchia di cloni
- Listato dei un clone

  # mmclone show <file>

  mostra le caratteristiche di <file> relative al cloning (depth, parent id, se e' clone-parent)
- mmclone show su un clone mostra l'i-node number del clone parent; per identificare il file name utilizzare tsfindinode:

  # tsfindinode –i <i-node> <starting-point-path>

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 file clone diviene un file ordinario

Cloni di snapshot

- Si puo' creare il clone di un file di una snapshot
  - in questo caso non e' necessario il comando mmclone snap: il file originale e' gia' immutabile, e diviene un clone-parent
- Prima di eliminare una snapshot si devono separare o rimuovere tutti i cloni che hanno un clone-parent nella snapshot
  - si perdono i blocchi originali e non modificati del clone
  - la cancellazione viene eseguita, ed il file clone rimane corrotto

Allocazione

- Clone e clone-parent sono file indipendenti (diversi i-node) e quindi possono avere differenti
  - ownership
  - policy di allocazione
  - replica factor
  - attributi,acl, …
- La quota occupata dal clone viene conteggiata per i soli blocchi scritti sul clone (quelli modificati dall'originale)
- Il file clone potrebbe non rispettare policy e replica factor per i blocchi non modificati (che seguono le proprieta' del clone-parent)