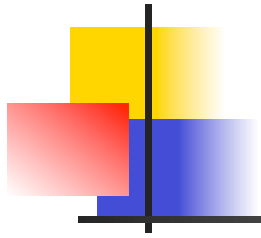




# Corso sul file system parallelo distribuito GPFS

---

Part 3: ILM and policies



# Outline

---

## Part 3:

- Intelligent Life-cycle Management (ILM)
- Storage pools and fileset
- Policies



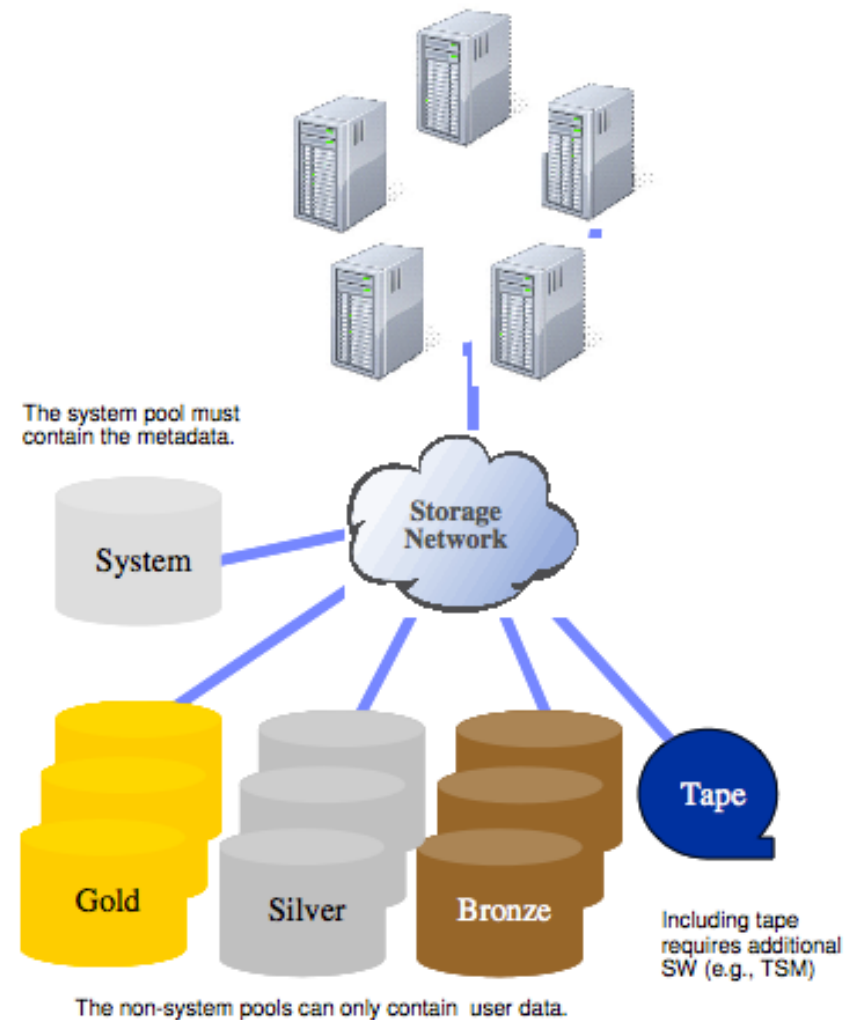
# Information Lifecycle Management (ILM)

---

- policy-driven automation and tiered storage management
- storage pools
  - **Storage pools** allow the creation of disk groups within a file system (**hardware partitioning**)
  - new feature in v. 3.2 - external storage pools (to interact with an external storage management, e.g. TSM)
- filesets
  - sub-tree of the file system namespace
  - administrative boundary to set quotas
  - control initial data placement or data migration
- user-defined policies

# Information Lifecycle Management in GPFS

- File placement policies
  - Define where the data will be created (appropriate storage pool)
  - Rules are determined by attributes like
    - File name
    - User name
    - Fileset
- File management policies
  - Possibility to move data from one pool to another without changing file location in the directory structure
  - Change replication status
  - Prune file system (deleting files as defined by policy)
  - Determined by attributes like
    - Access time
    - Path name
    - Size of the file





# Storage Pools

---

- ILM Manages sets of storage called "storage pools"
- What is a storage pool?
  - A named subset of disks and tapes
  - *Each file is assigned to a storage pool based upon **policy rules***
    - *placement policies* (where to place files upon creation)
    - *migration policies* (moving files from one pool to another)
    - *deletion policies* (removing files from the storage system)
  - *What are they are good for?*
    - *Tiered storage* (files aged to slower/cheaper disk)
    - *Dedicated storage* (e.g., per user or per project or per directory subtree)
    - *Failure containment*
      - *To limit the amount of data lost due to a failure*
      - *To bound the performance impact of RAID rebuild*
  - *Appropriate use of special-purpose storage*
    - *Different RAID levels Enterprise grade disk vs. consumer-grade disk*
    - *Multimedia friendly storage*



# GPFS Filesets

---

- What they are:
  - - A named subtree of a GPFS file system
  - - Somewhat like a distinct file system, i.e. a *fileset* can be unlinked without
    - deleting it, and it can subsequently be linked using its name as a handle
  - What they are good for:
    - - Filesets can have *quotas* associated with them (global; not per-pool).
      - - Fileset quotas are independent of user and group quotas
    - - Filesets can be used to restrict the effect of policies to specific files
  - Side effects:
    - - Unlinked filesets can confuse programs that scan the file system (e.g. incremental backup programs)
    - - Moving and linking between filesets is not allowed, in keeping with their being like little file systems



# GPFS ILM/HSM Integration

---

- GPFS Integrates its ILM Policies with tape based HSM Products
  - GPFS extends its Information Lifecycle Management (ILM) functionality to integrate with HSM (Hierarchical Storage System) products.
    - A single set of policies is used to move data between GPFS storage pools and tape storage pools.
  - Supported HSM products include
    - High Performance Storage System (HPSS)
    - Tivoli Storage Manager (TSM)
  - Cool Feature: very fast file scans
    - 1 million files in 13 seconds
    - 1 billion files in 75 minutes



# Policies

---

- Initial placement policy:
  - Default rule:
    - `SET POOL 'system'`
  - Must be defined
    - For a file system with SP 'system' dedicated for metadataOnly
      - With default rule can't write a single byte
  - Can be changed at any time using  
`mmchpolicy 'fs name'`
  - To list policy use  
`mmlspolicy 'fs name' -L`



# Sample GPFS ILM Policy Statements

## ■ Initial placement:

```
RULE 'SlowDBase' SET POOL 'sata' FOR FILESET('dbase') WHERE NAME LIKE '%.data'  
RULE 'SlowScratch' SET POOL 'sata' FOR FILESET('scratch') WHERE NAME LIKE '%.mpg'  
RULE 'default' SET POOL 'system'
```

## ■ Movement by age:

```
RULE 'MigData' MIGRATE FROM POOL 'system' THRESHOLD(80,78)  
WEIGHT( TIME_SINCE_LAST_ACCESS ) TO POOL 'sata' FOR FILESET('data')  
RULE 'HsmData' MIGRATE FROM POOL 'sata' THRESHOLD(95,80)  
WEIGHT( TIME_SINCE_LAST_ACCESS ) TO POOL 'hsm' FOR FILESET('data')  
RULE 'Mig2System' MIGRATE FROM POOL 'sata' WEIGHT(Access_Time) TO POOL 'system' LIMIT(85)  
FOR FILESET('user','root') WHERE DAYS_SINCE_LAST_ACCESS_IS_LESS_THAN( 2 )
```

## ■ Lock in place:

```
RULE 'ExcDBase' EXCLUDE FOR FILESET('dbase')
```

## ■ Life expiration:

```
RULE 'DelScratch' DELETE FROM POOL 'sata' FOR FILESET('scratch') WHERE  
DAYS_SINCE_LAST_ACCESS_IS_MORE_THAN( 90 )
```

Rule to move  
Files to tape

```
RULE EXTERNAL POOL 'hsm'  
EXEC '/usr/local/bin/ProcMigrate' OPTS 'D1T0'
```

# Sample GPFS ILM Policy Statements

## Initial Placement

Rule name

Storage pool name  
(corresponds to class of storage)

Fileset name  
(corresponds to subdirectory)

Qualifiers

```
RULE 'SlowDBase' SET STGPOOL 'sata' FOR FILESET('dbase') WHERE NAME LIKE '%.data'  
RULE 'SlowScratch' SET STGPOOL 'sata' FOR FILESET('scratch') WHERE NAME LIKE '%.mpg'  
RULE 'default' SET STGPOOL 'system'
```

## Movement by Age

```
RULE 'MigData' MIGRATE FROM POOL 'system' THRESHOLD(80,78)  
WEIGHT( TIME_SINCE_LAST_ACCESS ) TO POOL 'sata' FOR FILESET('data')  
RULE 'HsmData' MIGRATE FROM POOL 'sata' THRESHOLD(95,80)  
WEIGHT( TIME_SINCE_LAST_ACCESS ) TO POOL 'hsm' FOR FILESET('data')  
RULE 'Mig2System' MIGRATE FROM POOL 'sata' WEIGHT(Access_Time) TO POOL 'system' LIMIT(85)  
FOR FILESET('user','root') WHERE DAYS_SINCE_LAST_ACCESS_IS_LESS_THAN( 2 )
```

Rule to  
move files  
to HPSS

## Lock in place

```
RULE 'ExcDBase' EXCLUDE FOR FILESET('dbase')
```

## Life expiration

```
RULE 'DelScratch' DELETE FROM POOL 'sata' FOR FILESET('scratch') WHERE  
DAYS_SINCE_LAST_ACCESS_IS_MORE_THAN( 90 )
```



# More complex example

---

Example of balanced placement policy for a file system with 4 storage pools:

```
RULE 'DATA3' SET POOL 'data1' LIMIT(99) WHERE INTEGER(RAND()*40)<10
RULE 'DATA4' SET POOL 'data2' LIMIT(99) WHERE INTEGER(RAND()*30)<10
RULE 'DATA5' SET POOL 'data3' LIMIT(99) WHERE INTEGER(RAND()*20)<10
RULE 'DATA6' SET POOL 'data4' LIMIT(99)
```

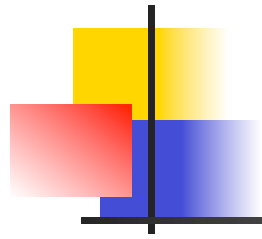
```
RULE 'DATA32' SET POOL 'data1' LIMIT(99) WHERE INTEGER(RAND()*30)<10
RULE 'DATA42' SET POOL 'data2' LIMIT(99) WHERE INTEGER(RAND()*20)<10
RULE 'DATA52' SET POOL 'data3' LIMIT(99)
```

```
RULE 'DATA33' SET POOL 'data1' LIMIT(99) WHERE INTEGER(RAND()*20)<10
RULE 'DATA43' SET POOL 'data2' LIMIT(99)
```

```
RULE 'DATA34' SET POOL 'data1'
```

See documentation at

**<http://publib.boulder.ibm.com/infocenter/clresctr/vxxr/index.jsp>**



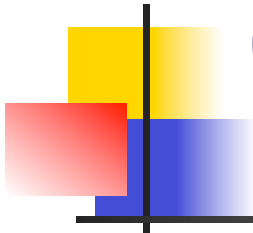
# Migrate and delete policies

- If the storage pool named **"system"** has an occupancy percentage above 90% now, bring the occupancy percentage of **"system"** down to 70% by migrating the largest files to storage pool **"data"**:

```
RULE 'mig1' MIGRATE FROM POOL 'system' THRESHOLD(90,70)
  WEIGHT(KB_ALLOCATED) TO POOL 'data'
```

- Delete files from the storage pool named **"system"** that have not been accessed in the last 30 days, and are named like **temporary** files or appear in any directory that is named **tmp**:

```
RULE 'del1' DELETE FROM POOL 'system' WHERE
  (DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME) > 30) AND
  (lower(NAME) LIKE '%.tmp' OR PATH_NAME LIKE '%/tmp/%')
```



# Rules to control the placement, migration and retention of files

---

- Placement policies, evaluated at file creation, example rule 'home' set pool 'gold' for fileset 'home\_sven'
- Migration policies, evaluated periodically  
rule 'hsm' migrate from pool 'sata' threshold(90,85) weight(current\_timestamp – access\_time) to pool 'hsm' where file\_size > 1024kb  
rule 'cleansilver' when day\_of\_week()=Monday migrate from pool 'silver' to pool 'bronze' where access\_age > 30 days
- Deletion policies, evaluated periodically  
rule 'purgebronze' when day\_of\_month()=1 delete from pool 'bronze' where access\_age>365 days
- List Rule generate reports on file system contents  
rule 'listall' list 'all-files' SHOW(varchar(file\_size) || ' ' || varchar(user\_id) || ' ' || fileset\_name )



# Execution of migrate policy

---

- Manually via mmapplypolicy:  
mmapplypolicy 'fs name' -P 'policy file' -I yes
- Via 'callback' (user exit)  
mmaddcallback
- cannot be run from a Windows node

```
mmapplypolicy {Device | Directory} [-A IscanBuckets] [-a IscanThreads] [-B
MaxFiles] [-D yyyy-mm-dd [@hh:mm[:ss]]] [-e]
[-f FileListPrefix] [-g GlobalWorkDirectory] [-I {yes | defer |
test | prepare}] [-i InputFileList] [-L n] [-M
name=value...] [-m ThreadLevel] [-N {all | mount |
Node[,Node...] | NodeFile | NodeClass}] [-n DirThreadLevel] [-P
PolicyFile] [-q] [-r FileListPathname...] [-S SnapshotName] [-s
LocalWorkDirectory] [--single-instance] [--sort-buffer-
size Size]
```

# Lab setup

