



Information Lifecycle Management e Policy

Alessandro Brunengo INFN-Genova

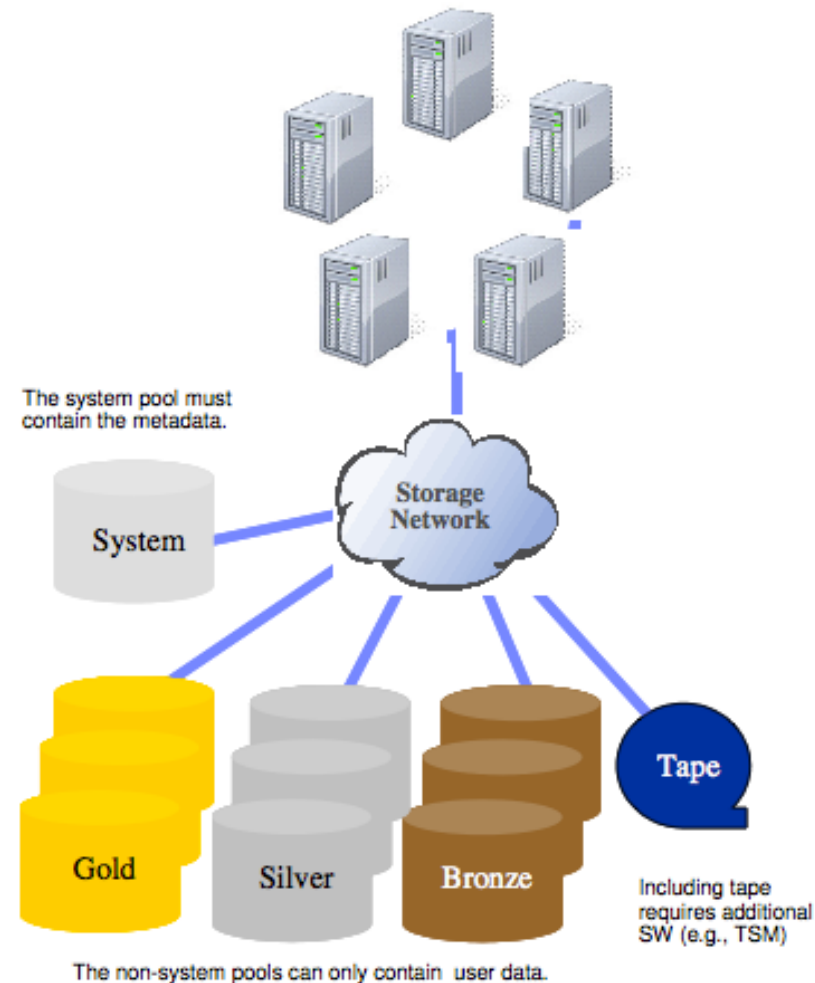


ILM

- ILM consiste nella gestione automatizzata di
 - collocazione e spostamento dei file su storage opportuni durante la loro vita (tiered storage)
 - cancellazione dei file
- GPFS supporta ILM attraverso
 - storage pool: partizionamento dell'hardware di storage in funzione di caratteristiche (affidabilità, velocità, costo...)
 - external storage pool: possibilità di utilizzo di storage esterno come tape library (HSM)
 - (user defined) policies: implementano collocazione e movimentazione automatizzata del contenuto di un file tra diversi storage pool in base a caratteristiche del file o della disponibilità di storage
 - fileset: permettono l'applicazione delle policy con una granularità più fine del file system

Information Lifecycle Man

- File placement policies
 - Definisce in quale pool mettere i dati alla creazione del file
 - Regole determinate da:
 - File name
 - User name
 - Fileset
- File management policies
 - Possibilita' di spostare on-line dati tra pool senza cambiare la collocazione nel file system
 - Modificare il replica status
 - Garbage collection (rimozione file)
 - Operazioni eseguite in base a attributi ordinari estesi del file





GPFS ILM/HSM Integration

- GPFS supporta l'integrazione di ILM con prodotti HSM basati su tape
- GPFS estende la funzionalità di ILM per integrare prodotti HSM grazie al supporto per external storage pool
 - il sistema HSM viene definito come (external) storage pool di GPFS
 - in questo modo può essere pool per l'applicazione delle policy di movimentazione dei dati
- I prodotti HSM supportati includono
 - High Performance Storage System (HPSS)
 - Tivoli Storage Manager (TSM)
- Prestazioni notevoli sullo scan dei file:
 - 1 million files in 13 seconds
 - 1 billion files in 75 minutes



Policy

- Regole che definiscono cosa muovere, da dove a dove, in quali condizioni
- Diversi tipi di policy
 - Placement
 - Migration
 - Deletation
 - Exclusion
 - File list
 - Group rule
 - File restore
 - External storage pool definition
 - External list definition

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/%')
```



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]
```



High Performance Extended Attributes

- Extended attributes in GPFS since 3.2
 - not commonly used, in part because of performance concerns.
 - In GPFS 3.4, a comprehensive redesign of the extended attributes support infrastructure was implemented, resulting in significant performance improvements.
 - In GPFS 3.5, extended attributes are accessible by the GPFS policy engine allowing you to write rules that utilize your custom file attributes.
- Now an application can use standard POSIX interfaces to manage extended attributes and the GPFS policy engine can utilize these attributes.

Policy example

Macros
manipulate data

```
define(east_adjustment,  
  CASE  
    WHEN XATTR_FLOAT('user.e',1,-1,'DECIMAL') < 0  
      THEN 180+(180+XATTR_FLOAT('user.e',1,-1,'DECIMAL'))  
    ELSE XATTR_FLOAT('user.e',1,-1,'DECIMAL')  
  END )  
define(west_adjustment,  
  CASE  
    WHEN XATTR_FLOAT('user.w',1,-1,'DECIMAL') < 0  
      THEN 180+(180+XATTR_FLOAT('user.w',1,-1,'DECIMAL'))  
    ELSE XATTR_FLOAT('user.w',1,-1,'DECIMAL')  
  END )  
define(north_adjustment, 90+XATTR_FLOAT('user.n',1,-1,'DECIMAL'))  
define(south_adjustment, 90+XATTR_FLOAT('user.s',1,-1,'DECIMAL'))
```

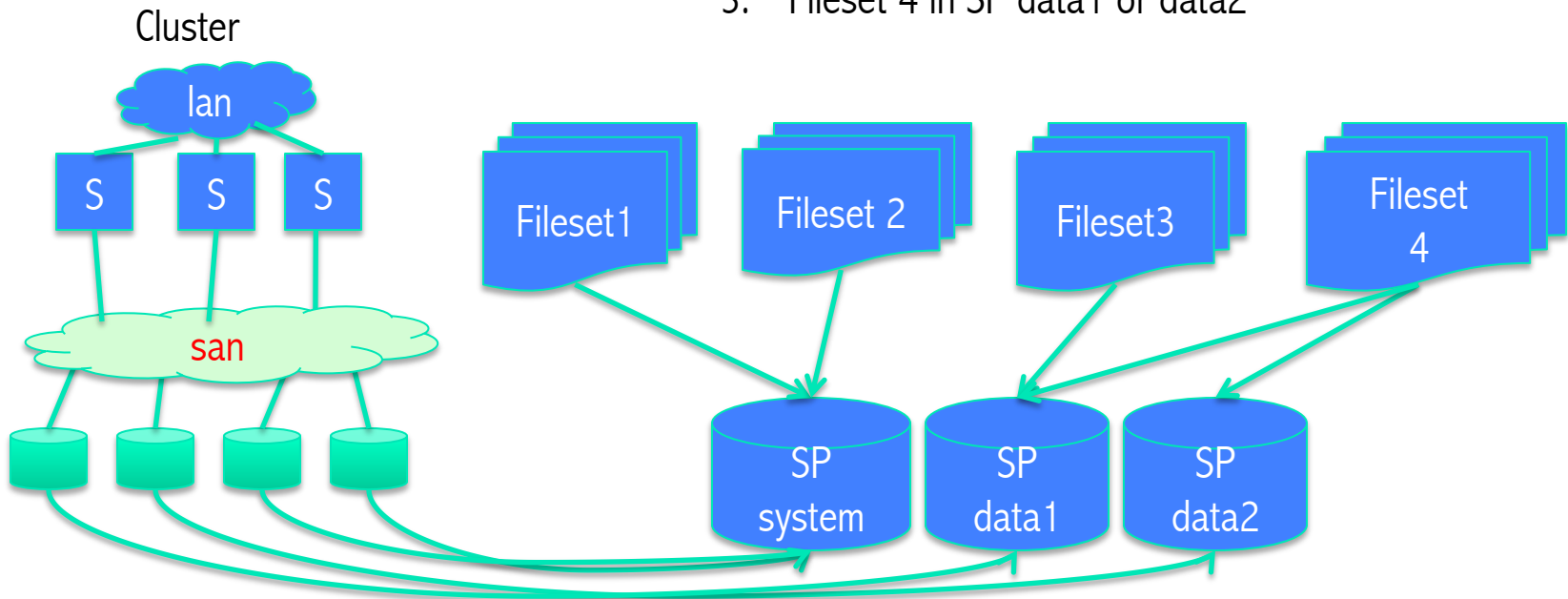
Query custom file
extended attributes

Policy calls
macros

```
RULE 'listall' list 'geo_files'  
  SHOW( varchar(kb_allocated)|| ' ' || fileset_name )  
WHERE KB_ALLOCATED > 0  
  AND FILESET_NAME='master_t1'  
  AND south_adjustment <= 130.993664  
  AND north_adjustment >= 126.994021  
  
  AND east_adjustment >= 250.964755  
  AND west_adjustment <= 257.946178  
  
  AND DAYS(XATTR('user.t')) >= (DAYS(CURRENT_TIMESTAMP)-90)
```

Exercise 1:

1. Create file system with 3 Storage pools
2. Define replication factor 2 for metadata
3. Create 4 filesets
4. Create placement policy:
 1. fileset 1 and 2 in SP "system"
 2. Fileset 3 in SP data1
 3. Fileset 4 in SP data1 or data2





Exercise 2:

- Create policy and migrate data from Storage pool “system” to SP “data1”
- Create new file in fileset “1” and check its location (mmlsattr).
- Change replication factor for that file to 2
- Change placement policy to set replication =2 for data in storage pool “system”