

DCache INTRODUCTION COURSE

Christoph Anton Mitterer
christoph.anton.mitterer@lmu.de





VI. SPACE MANAGEMENT, SPACE RESERVATION AND SRM

This chapter is not held in class.





GENERAL CONCEPTS

The following describes the concepts of space management and space reservation as well as space tokens:

- Space management is the process or system of controlling the (total available) space (which can be free or used) for different purposes, for example concepts like space reservation or quotas.
- Space reservation is a concept or technique, where a specific amount of storage is reserved for a logical entity or purpose (for example a project, a group of users, an organisation or a certain type of usage). The reservation may have additional constraints like an expiration time or allowed usage as well as properties like a name or a description.

The space management system (and thus the storage management system) guarantees ("promises") the space reservation.

(The concept of quotas is similar, but usually, quotas specify only a maximum amount of storage, that a given entity is allowed to use, and do not guarantee it.)

- A space token is a concrete instance of reserved space.

The motivation behind this is usually to guarantee space to the accessing clients.



DCACHE'S SPACE RESERVATION AND ITS RELATIONS TO SRM

Space reservation and space tokens in dCache are closely modelled after the concepts defined and standardised by the SRM-protocol.

They are mainly used with SRM and quite closely connected with it.

It is however possible to use them independently of the SRM-protocol and in principle even without a SRM-door at all.

Amongst some others, the following services are additionally required for space management:

- **spacemanager**

The service which actually manages the space and reservations.

- **pinmanager**

This service manages the pinning of files, which means whether or not files might be completely moved off from disk (for example to tape).

The related data is stored in several tables within the **dcache** database.

This course covers space management mostly from the view of SRM.

RETENTION POLICY

The Retention Policy is a property defined by the SRM-standard, that relates to space reservations (“space tokens”) or files.

It specifies the quality of the storage that is used for the data and can have the following values:

- Replica

The lowest quality of storage, which usually means that there is only a single copy (“one replica”) per file which is located on a hard disk.

- Output

An intermediate quality of storage, which is better than Replica but worse than Custodial.

In the future, dCache might interpret this as having multiple copies (“multiple replicas”) per file which are all located on distinct hard disks.

- Custodial

The highest quality of storage, which usually means that there is at least one copy per file which is located on tape.

This does however not exclude additional copies on hard disk.



ACCESS LATENCY

The Access Latency is a property defined by the SRM-standard, that relates to space reservations (“space tokens”) or files.

It specifies how fast data is available (for reading) and can have the following values:

- Offline

Additional preparation steps which require human interaction might be necessary before the files can be read by clients.

- Nearline

Additional preparation steps (usually staging from tape) might be necessary before the files can be read by clients, which usually means that all copies are located on tape and none are retained on hard disk.

- Online

The files are readily available and can “immediately” be read by clients, which usually means that at least one copy is retained on hard disk even if copies are located on tape.



SPACE RESERVATION WITHIN dCACHE

dCache generally differentiates between the following two classes of files:

- Files that are not (or will not be) part of space reservation (often called “non-space-managed-files”).
- Files that are (or will be) part of space reservation (often called “space-managed-files”).

The class of a file is decided when it is being created (“written to the cluster by a client”) and can depend on several factors including the access-protocol, its configuration, its usage by the client as well as other configuration of dCache and the target directories’ PNFS-tags.

More on that during the following slides.

Currently, files cannot easily change between the above classes (or between different space reservations) during their lifetime.



SPACE RESERVATION WITHIN dCACHE

The following should be noted on space reservations:

- When a file is deleted its used space is returned back to the free space of the reservation.
- dCache “only” allows to reserve hard disk space which can be used for write-operations (“write space”). Tape space cannot be reserved.
- When a file “fully” went to tape (that means no copy is retained on hard disk) its used space is returned back to the free space of the reservation.
In other words, tape space is not accounted for reservations
- When a file is read or moved from tape back to disk (regardless of whether a copy remains on tape or not) its used space is not accounted for any reservation.

It follows from the above, that tape space is considered to be “infinite”.



SPACE RESERVATION AND LINKGROUPS

Technically, space reservations (and thus space tokens) are made upon LinkGroups. LinkGroups are the “highest-level” objects within the pool management and pool selection system and have the following characteristics:

- A LinkGroup groups zero or more Links.
- A LinkGroup is “connected” with zero or more pools via its Links.
- The total available space of a LinkGroup is determined by the total available space of all its pools.
- The used space of a LinkGroup is determined by the space that is either actually used or reserved. Analogous, the free space of a LinkGroup is determined by the space that is neither actually used nor reserved.
- A LinkGroup has an unique ID.
- A LinkGroup has settings that specify which Retention Policies and which Access Latencies are supported and allowed (for reservation) by it.
dCache does not automatically check whether any of these are really supported.
- A LinkGroup has properties that specify who is allowed to make space reservations upon it.



SPACE RESERVATION AND LINKGROUPS

The following technical details on space reservation and LinkGroups should be noted:

- A space reservation is made upon the (total) free space of exactly one LinkGroup. It follows from this, that space reservations (and thus space tokens) cannot span over multiple LinkGroups.
- When a file is successfully created within a space reservation, it will be located on one of the pools connected with that reservation's LinkGroup.

Important:

In order for space calculation (this is the calculation of used and free space) being correct, the following has to apply:

- No pool must be connected to more than one LinkGroup (for example via different PoolGroups or Links).
- No pool must be connected "multiple times" to the same LinkGroup (for example via different PoolGroups or Links).

If any of the above would happen, the same space would be counted multiple times. Actually it is possible, to have such advanced setups in a safe and reasonable way.

SPACE TOKENS

As described above, a space token is a concrete instance of a space reservation.

Space tokens have the following properties:

- An unique name (which is a number in case of dCache) that is used by the storage management system and by clients to identify the token.
It is also common to refer to just this name when speaking about space tokens.
- A non-unique description, that is typically used by clients to discover all space tokens matching a given description.
- Its (total) space, used space and free space.
- Its Retention Policy and Access Latency.
- Its expiration time.
- Its creator's VO-group and VO-role.

Space tokens can be created via the following two ways:

- Using and SRM version 2.2 client that supports requesting space reservations.
- Using the `SrmSpaceManager`-cell within dCache's administration interface.



EXPLICIT AND IMPLICIT SPACE RESERVATION

dCache has two modes of space reservation when a file is created (“written to the cluster by a client”):

- **Explicit Space Reservation**

In this mode, an already existing space token is used for the reservation and the space used by the created file is accounted for it.

The token has to be explicitly specified somehow.

- **Implicit Space Reservation**

In this mode, a new (so called “virtual”) space token is generated with the size of the file and a lifetime of the time required to create the file (“the token expires after the file has been written”).

Of course, the (used/reserved) space remains removed from the LinkGroup’s free space after the token’s expiration.

Usually, explicit space reservation is generally preferred over implicit space reservation.

Implicit space reservation must have been enabled to be usable at all, but even then it is not yet guaranteed that implicit space reservation can be performed.

EXPLICIT AND IMPLICIT SPACE RESERVATION

The following describes which mode is used when and how:

1. Explicit space reservation is used...

1. ...if the client explicitly specified a space token via the access-protocol.
(currently only possible with SRM version 2.2)
 2. ...if the target-directory explicitly specified a space token via the `WriteTag-PNFS-tag`.
(works with any access-protocol, including SRM version 2.2 if no space token was specified)
- In all cases, the (explicitly) selected space token (and thus its LinkGroup) is used.

2. Implicit space reservation is used...

- ...with the SRM-access-protocol if `srmImplicitSpaceManagerEnabled` is set to yes and...
- ...with other access-protocols if `SpaceManagerReserveSpaceForNonSRM Transfers` is set to yes and...



EXPLICIT AND IMPLICIT SPACE RESERVATION

1. ...if the client specified only a Retention Policy and an Access Latency (but not a space token) via the access-protocol.
(currently only possible with SRM version 2.2)
 2. ...if the target-directory specified only a Retention Policy and an Access Latency (but not a space token) via the `RetentionPolicy-` and `AccessLatency-PNFS-tags`.
(works with any access-protocol, including SRM version 2.2 if no Retention Policy and Access Latency as well as no space token were specified)
 3. ...otherwise, with the cluster-wide default Retention Policy and default Access Latency.
(works with any access-protocol, including SRM version 2.2 if no Retention Policy and Access Latency as well as no space token were specified)
- In all cases, a LinkGroup that allows the selected Retention Policy and Access Latency and that has enough free space will be used to create the virtual space token.
If no such LinkGroup exists the access fails.



EXPLICIT AND IMPLICIT SPACE RESERVATION

- If neither explicit- nor implicit space reservation can be performed (for the later, because it is not enabled for the particular access-protocol), space reservation will not be performed at all.

In this case, the file may only went to pools which are not connected in any way with any LinkGroup.

The actual pool is then selected via dCache's usual pool selection mechanism, as described in chapter IV.

ACCESS CONTROL FOR SPACE RESERVATIONS

Space is a resource and access to it (and thus to space reservations) must be controlled.

When a client requests the creation of a space reservation (which is currently only possible with SRM version 2.2), dCache extracts its certificate's FQAN and evaluates it against the rules in the "SpaceManager-LinkGroup-authorization-file".

The client is granted "reservation-permissions" for those LinkGroups, where a rule matches.

It should be noted, that even for the creation of a space reservation via the `SrmSpaceManager`-cell, matching rules must exist.

It should also be noted, that extracting FQANs requires gPlazma to be used with the SRM-door.

The pathname of the "SpaceManager-LinkGroup-authorization-file" is specified via the `SpaceManagerLinkGroupAuthorizationFileName`-configuration-parameter.

“SPACEMANAGER-LINKGROUP-AUTHORIZATION-FILE” SYNTAX AND SEMANTICS

A rules-block, specifying the reservation-permissions for a given LinkGroup is introduced via a line with the following syntax:

LinkGroup *link-group-name*

link-group-name is the name of the LinkGroup for which the following rules should apply.

Each line following that introduction is a rule with the following syntax:

fqn

fqn is an FQAN for which reservation-permissions should be granted.

“*” may be used as “wildcard character” for group names and attribute values.

Blocks must be separated with an empty line.

Lines starting with “#” and empty lines are ignored by dCache.



"SPACEMANAGER-LINKGROUP-AUTHORIZATION-FILE" SYNTAX AND SEMANTICS

Example:

```
LinkGroup atlas_DataDisk-linkGroup  
/atlas/Role=production
```

```
LinkGroup atlas_GroupDisk-linkGroup  
/*Role=production
```

```
LinkGroup atlas_LocalGroupDisk-linkGroup  
/atlas  
/atlas/Role=production
```

```
LinkGroup atlas_ScratchDisk-linkGroup  
/atlas/Role=*
```

CONFIGURATION OF SPACE MANAGEMENT

The following configuration-parameters are related to space management:

- `srmSpaceManagerEnabled`

Set to **yes** if space management should be enabled, or to **no** otherwise.

This must be set in the configuration-file for any door within the cluster, that wants to use space management (and not just for the SRM-door).

If space management is disabled, the following parameters are ignored:

- `srmImplicitSpaceManagerEnabled`

Set to **yes** if implicit space reservation should be enabled for the SRM-access-protocol, or to **no** otherwise.

This must be set (at least) in the configuration-file of the SRM-door.

- `SpaceManagerReserveSpaceForNonSRMTransfers`

Set to **true** if implicit space reservation should be enabled for a non-SRM-access-protocol, or to **false** otherwise.

This must be set in the configuration-files of the non-SRM-doors that should have implicit space reservation enabled.



CONFIGURATION OF SPACE MANAGEMENT

- `DefaultRetentionPolicy`

Specifies the cluster-wide default Retention Policy and can be set to `REPLICA`, `OUTPUT` or `CUSTODIAL`.

- `DefaultAccessLatency` (^{def} `DefaultAccessLatencyForSpaceReservation`)

Specifies the cluster-wide default Access Latency and can be set to `NEARLINE` or `ONLINE`.

`DefaultRetentionPolicy` and `DefaultAccessLatency` are actually “native” settings to dCache and independent of space reservation at all.



Finis coronat opus.

