Karlsruhe Institute of Technology

Cost Calculation and Hot-Spot Replication

Christopher Jung, SCC, KIT

STEINBUCH CENTRE FOR COMPUTING - SCC

Overview

- Cost Calculation
  - motivation
  - overall cost
  - performance cost
  - space cost
  - sweeping and lru age
  - special scenarios
  - costs in admin interface and pcells
- Hot-Spot Replication
  - motivation
  - replication within the transfer process
  - fixed threshold value
  - percentile scheme
  - hot-spot replication and space tokens

Cost Calculation

Cost calculation

- Why?
  - dCache needs a mechanism to determine which pool to use for a file transfer

    → difficult and multi-dimensional problem
- How?
  - for each pool, two costs are calculated
    - performance cost (a.k.a. cpu cost)
    - space cost
  - dCache combines these costs linearly:

$$overallCost = sFactor \cdot spaceCost + pFactor \cdot performanceCost$$

    - cost factors are non-negative, real numbers
    - for reading files, only the performance cost is considered
  - dCache picks the least expensive pool for the file transfer

Performance Cost

- Calculate the ratio of total transfers to the maximum number of concurrent transfers allowed per transfer type (store, restore, pool-to-pool client, pool-to-pool server, client request); of course only for queues where maxAllowed≠0

$$typeCost = \frac{activeTransfers + queuedTransfers}{maxAllowed}$$

- Average over all allowed transfer types

$$performanceCost = \frac{\sum_{allowedtypes} typeCost}{\#allowed\ types}$$

Space cost: introduction

Space cost

- idea: space cost inversely proportional to free space unless pool nearly full (in this case: consider if files have been recently used)
- (there are an old and a new scheme for calculating space cost; only the newer is considered in the following)

Important parameters:

- gap parameter: if free space<gap parameter, pool is considered “(nearly) full”
- breakeven parameter: factor in calculation of space cost if pool is nearly full; also determines the scheme (old: breakeven>1.0, new: breakeven<1.0)

Space cost: free space > gap parameter

This is the simple case:

- formula:

$$spaceCost = 3 \cdot \frac{newFileSize}{freeSpace}$$

- if newFileSize<50 MB, it is considered to be 50 MB

  → space cost is negligible for new pools and small files

- the factor of 3 is importance for the choice of the gap parameter:
  - rule of thumb: choose

$$gapParameter \geq 3 \cdot maxFileSize$$

as this guarantees space cost ≤ 1 for

Space cost: free < gap parameter

- Important variable: lruAge
  - time since the least recently used file has been used the last time (in seconds)
- Formula:
  - lruAge<60 (does this ever happen?):

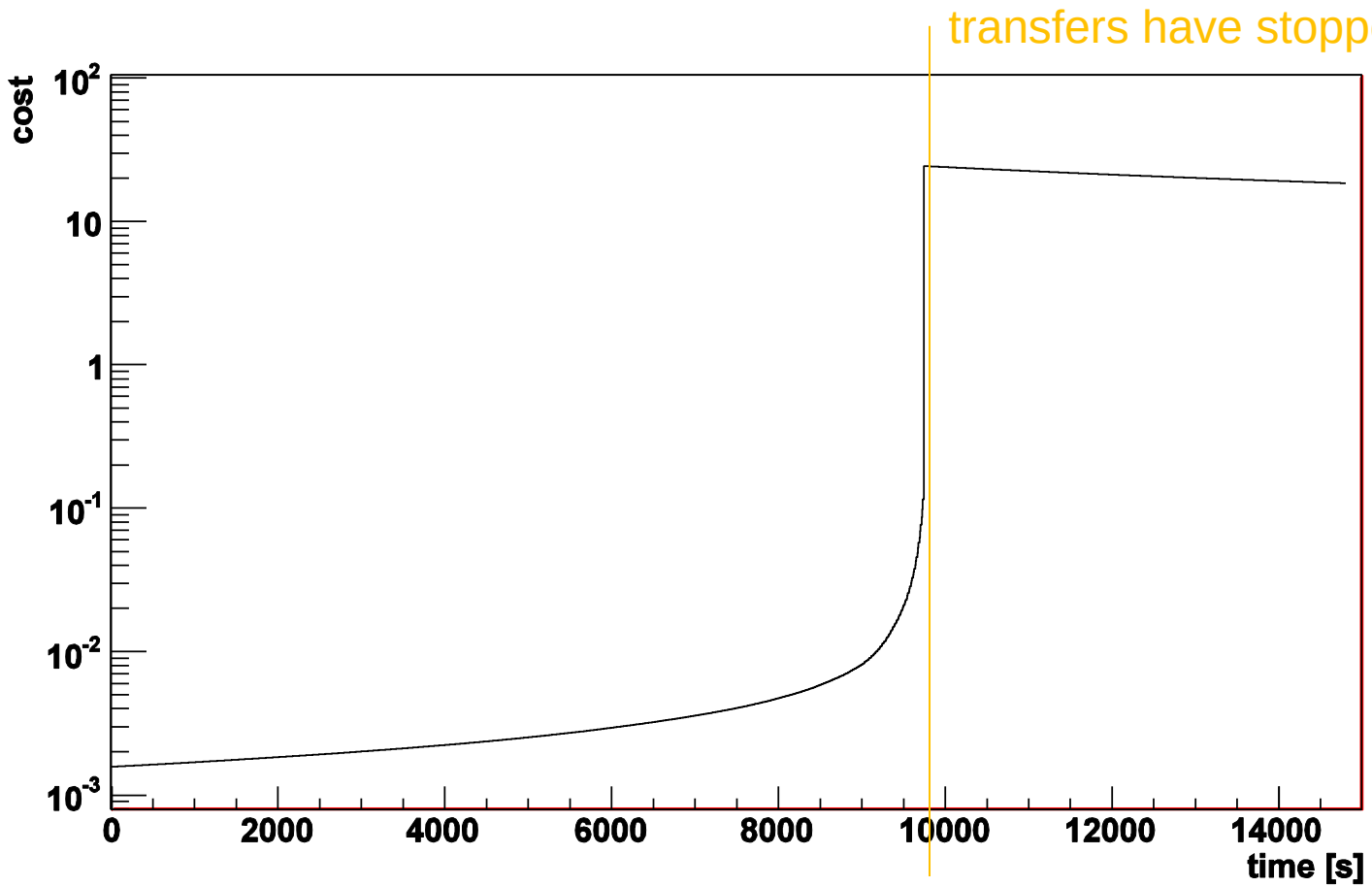
$$spaceCost = 1 + breakeven \cdot 7 \cdot 24 \cdot 60 = 1 + breakeven \cdot 10080$$

  - lruAge>60:

$$spaceCost = 1 + \frac{breakeven \cdot 7 \cdot 24 \cdot 3600}{lruAge}$$

    - lruAge=1 day → spaceCost=1+7·breakeven
    - lruAge=1 week → spaceCost=1+breakeven

Space cost for filling a new pool linearly



- Hyperbolic growth of space cost until gap parameter is reached
- When transfers have stopped, space cost declines hyperbolically.

(be aware: this is an idealized scenario)

How to find ‚old files' on pools

- List of removable files, sorted by last use
  - `sweeper ls [-l] [-a]:`

```
[](f01-065-105-e_3rT_cms) admin > sweeper ls -l
0 0000F8B2795C0B0B4C4C98CEC78FE969C8CE  CACHED   2986943416
15:38-01/13 16:06-10/01
1 0000CC3790F480584008B038CDD8729BFD53  CACHED   7297849666
15:38-01/13 16:06-10/01
2 00004D9DEDE9DD744DB19EB2C40BF5B65E8F  CACHED   2425666236
15:38-01/13 16:06-10/01
…
7166 00000778618381454 71E854C89E5BCE89474  CACHED   2037156281
15:28-03/25 15:28-03/25
7167 00001F40CD075D424DD19DFF8C3EBF12B151  CACHED   1061956366
15:38-01/13 15:35-03/25
7168 0000184783FF57C749728C357525BACB2B42  CACHED   1048119235
15:38-01/13 15:58-03/25
```

PNFSID state size ? dateOfMostRecentUse

How to remove ‚old files' from pools

- Getting the age of the least recently used file
  - `sweeper get lru`:

```
[](f01-065-105-e_3rT_cms) admin > sweeper get lru
15124511
[](f01-065-105-e_3rT_cms) admin > sweeper get lru -f
175 d 01:15:11
```

- Deleting all removable files: `sweeper purge`
- Freeing a given space size by removing the least recently used files
  - `sweeper free <bytesToFree>`

```
[](f01-065-105-e_3rT_cms) admin > sweeper free 1000000
Reclaiming 1000000 bytes
```

Special scenarios for cost factors

- Four categories of factors:
  - sFactor>0 & cFactor>0:
    - standard case
    - choice of pool difficult to predict in the medium and long term
  - sFactor=0 & cFactor>0:
    - transfers prefer pools with a low number of competing transfers
    - for a high number of transfers, the ratio of total transfers to maximum transfers will be the same on a pools used
  - sFactor>0 & cFactor=0:
    - transfers prefer empty pools
    - if all files have same size & free space>gap parameter & many files-> all pools will have same amount of free space
  - sFactor=0 & cFactor=0:
    - random selection of pools

Cost commands in the admin interface (I)

- What is the gap size (in bytes) on a pool?

```
[] (f01-065-105-e_3rT_cms) admin > info
…
--- pool (Main pool component) ---
Base directory    : /export/dc065105_3/pool
Revision          : [$Revision: 12882 $]
Version           : production-1.9.5-9(12882) (Sub=4)
Gap               : 4294967296
Report remove     : on
   …
```

- Setting gap parameter:
  - `set gap <always removable gap>/size[<unit>]`
    `# unit = k|m|g`
  - example:

```
[] (f01-065-105-e_3rT_cms) admin > set gap 5G
Gap set to 5368709120
```

Cost commands in the admin interface (II)

- Setting the breakeven parameter on a pool:
  - `set breakeven <breakEven> # free and recovable space`
  - example:

```
[] (f01-065-105-e_3rT_cms) admin > set breakeven 0.7
BreakEven = 0.7
```

- Getting the breakeven parameter (and the gap size, too):
  - example:

```
[] (PoolManager) admin > cm ls f01-065-105-e_3rT_cms
f01-065-105-e_3rT_cms={R={a=0;m=100;q=0};
S={a=0;m=20;q=0};M={a=7;m=212;q=0};PS={a=0;m=10;q=0};
PC={a=0;m=10;q=0};SP={t=15891378995200;f=227686115296
6; p=0;r=13607968679386;lru=15635953;
{g=4294967296;b=0.7}};XM={defaultq={a=0;m=5;q=0};
gsidcapq={a=0;m=2;q=0};gridftpq={a=1;m=5;q=0};
dcapq={a=6;m=200;q=0};};}
```

Sections (I)

- Depending on the functionality of the pool (e.g. write buffer, read pool, stager) and on other factors (local FS, network connection), one might need different cost factors (and other parameters). These can be defined in sections (a.k.a. partitions).
- List all cost sections:
  - `pm ls [<section>] [-l]`
  - Example:

```
[] (PoolManager) admin > pm ls -l
default
   -cpucostfactor=1.0
   -spacecostfactor=1.0
   -p2p=2.0
…
nocostSection
   -cpucostfactor=4.0
   -spacecostfactor=0.1
…
```

Sections (II)

- Each link can be assigned to exactly one cost section:
  - `psu set link <link> -section=<section>|NONE`
  - If assigned section does not exist or is not set, the `default` section is used.
- Which cost section does a link use?

```
[] (PoolManager) admin >  psu ls link  -l cms-stage-
link
cms-stage-link
 readPref  : 0
 cachePref : 70
 writePref : 0
 p2pPref   : 0
 section   : StageSection
 linkGroup : None
 UGroups :
   cms-tape-store  (links=5;units=21)
   world-net  (links=9;units=2)
```

Sections (III)

- Setting values for a section:

```
[] (PoolManager) admin > help pm set
pm set [<partitionName>|default]  OPTIONS
    OPTIONS
       -spacecostfactor=<scf>|off
       -cpucostfactor=<ccf>|off
       -p2p=<value>|off
       …
```

  - If a section called `<partition name>` does not exist, it is automatically created.
  - If a parameter is set to off, it is no longer overwritten but inherited from `default` section.

- Deleting a section:

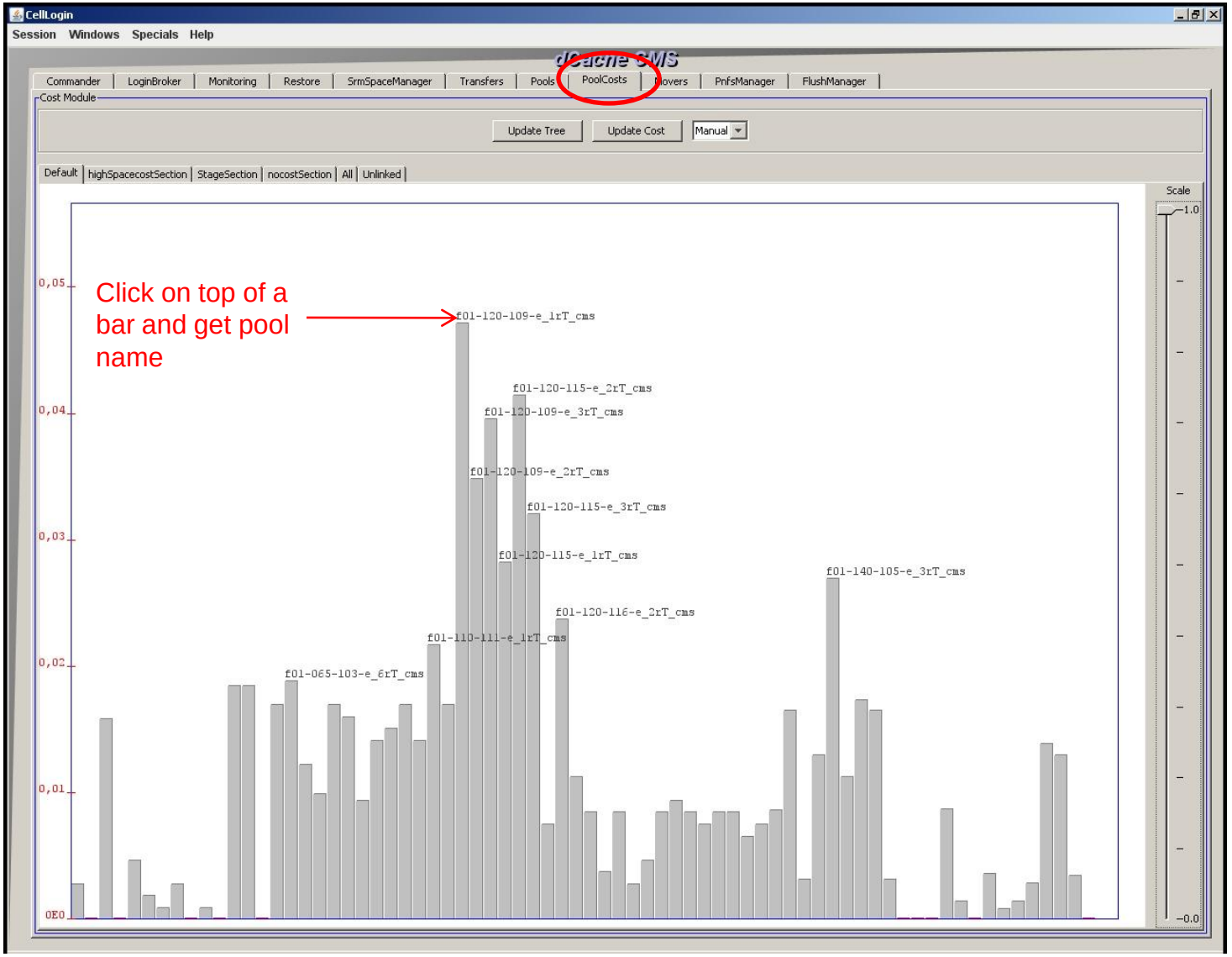
```
pm destroy <partitionName> # destroys parameter partition
```

And which pool costs do I have? (I)

- Just go to PoolManager

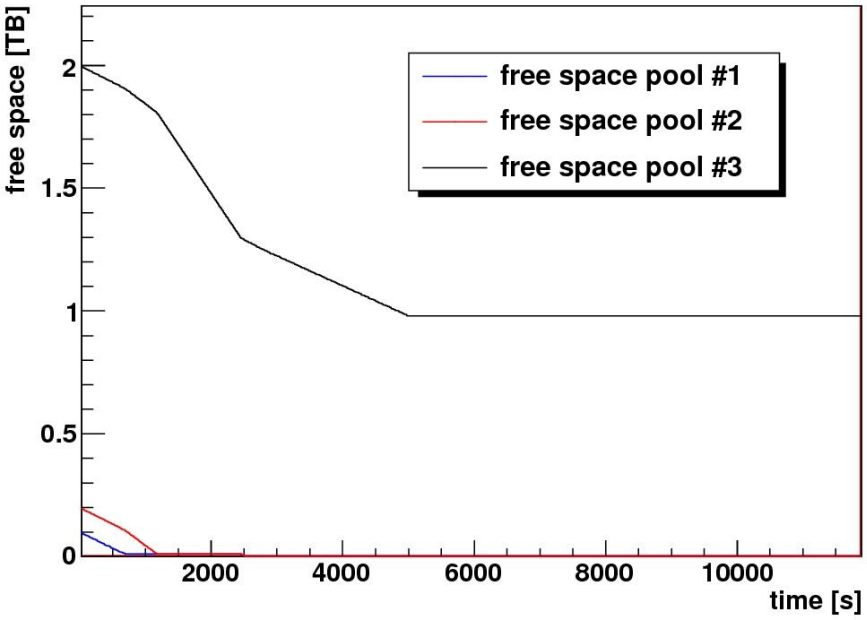
```
[] (PoolManager) admin > cm ls -d f01-065-105-e_3rT_cms
f01-065-105-e_3rT_cms=
{R={a=0;m=100;q=0};S={a=0;m=20;q=0};M={a=11;m=212;q=0};
PS={a=0;m=10;q=0};PC={a=0;m=10;q=0};SP={t=15891378995200;
f=2070135272140;p=0;r=1381255341483;lru=15698938;
{g=4294967296;b=0.7}};XM={defaultq={a=0;m=5;q=0};
gsidcapq={a=0;m=2;q=0};gridftpq={a=2;m=5;q=0};
dcapq={a=9;m=200;q=0};};};}
f01-065-105-e_3rT_cms={Tag={{hostname=f01-065-105-
e}};size=0;SC=7.245903300074572E-5;CC=0.055625;}
```

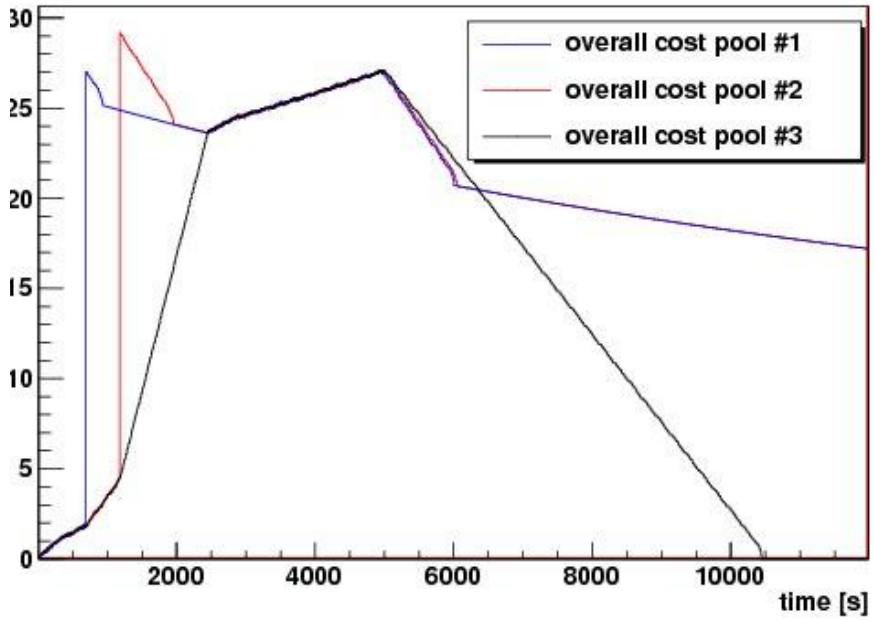
And which pool costs do I have? (II)



Using pcells gives a better overview!

Simulation of adding an empty pool





Scenario:

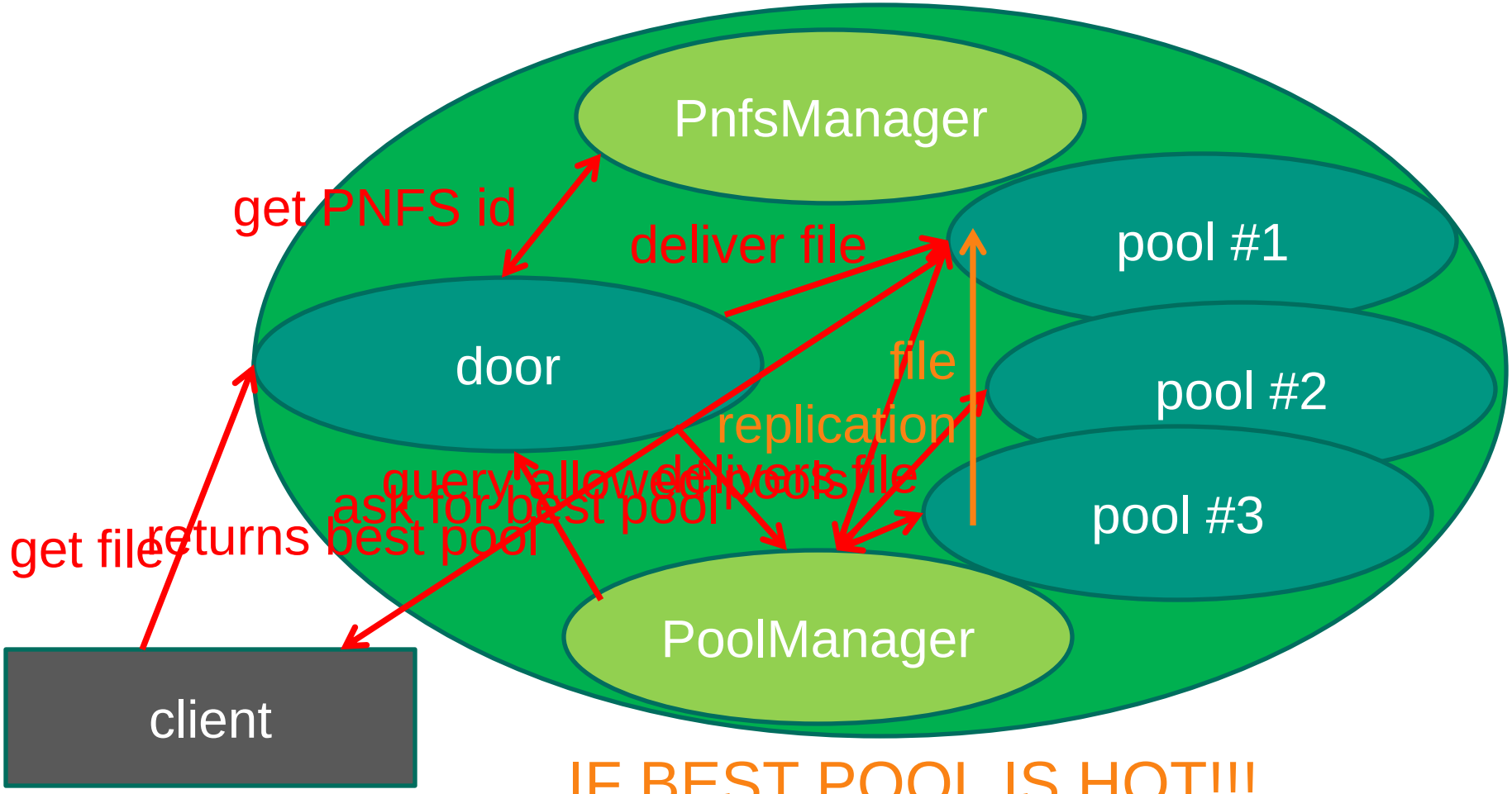
- one empty pool (#3) is added to two nearly full pools (cached files only)
- huge load of incoming files

Hot-Spot Replication

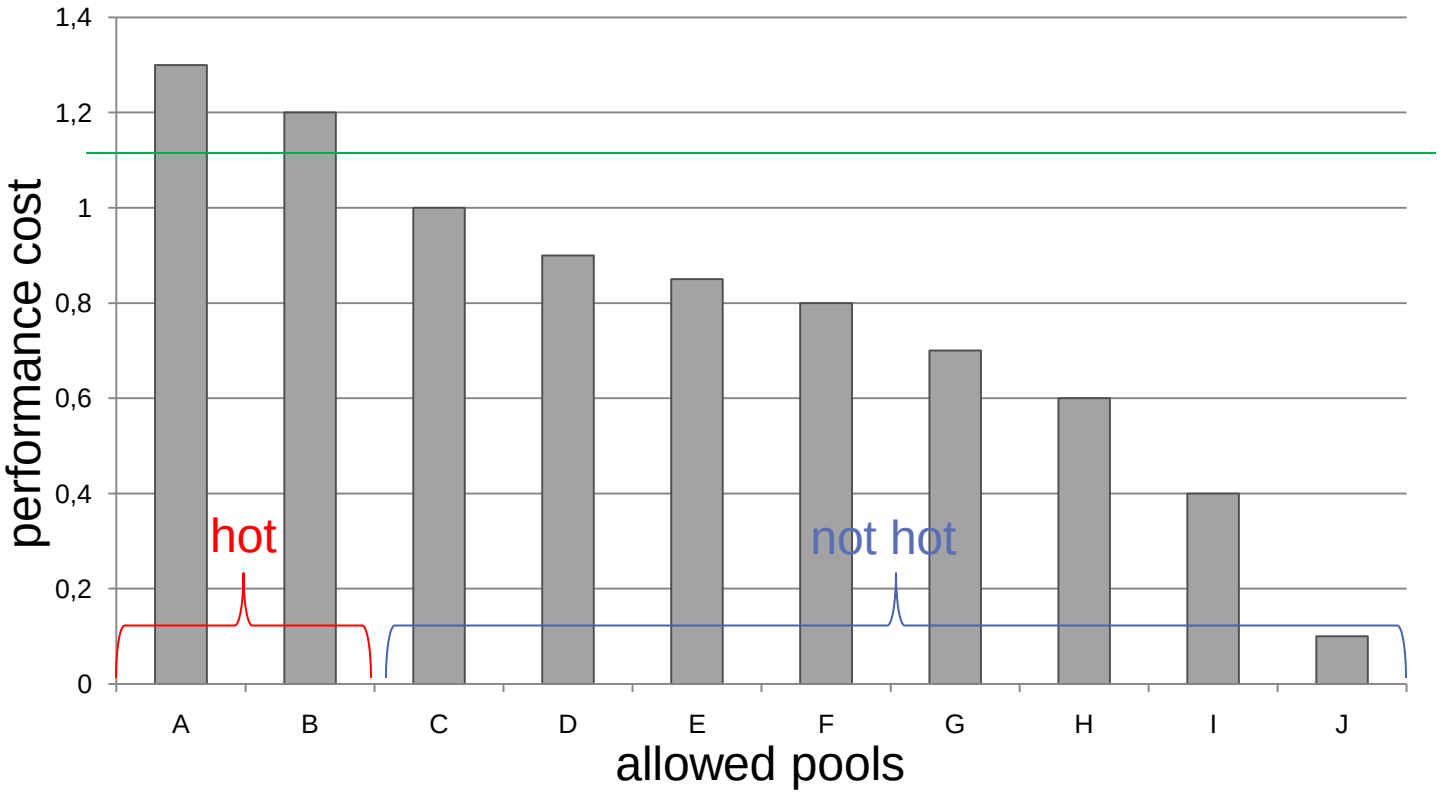
Hot-spot replication

- Idea:
  - Avoid heavy loads by replicating files from heavily used pools to less-loaded pools
- Remark:
  - Hot-spot replication and hot file replication are similar, but not the same
- How is a hot-spot detected?
  - By looking at the performance costs!
  - These are either evaluated
    - by their absolute value (fixed threshold scheme) or
    - by comparing them to each other (percentile scheme).

Replication within the transfer process

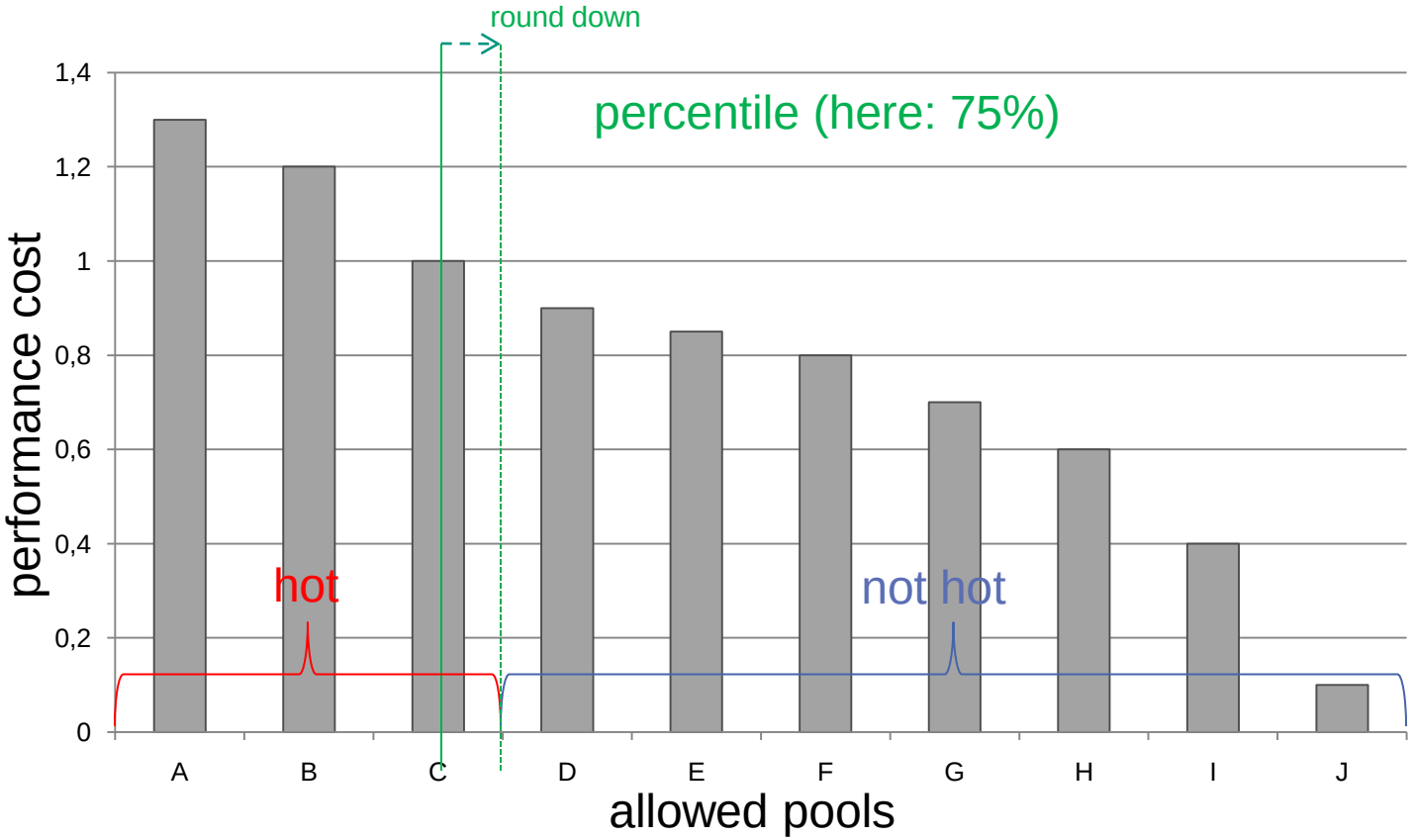


Fixed threshold scheme



- All pools with a cost higher than the threshold are hot.
- Choosing threshold value is difficult.

Percentile scheme



- Sort dCache pools in ascending order of costs.
- The percentile cost is the cost of the pool that is 75% along the list.

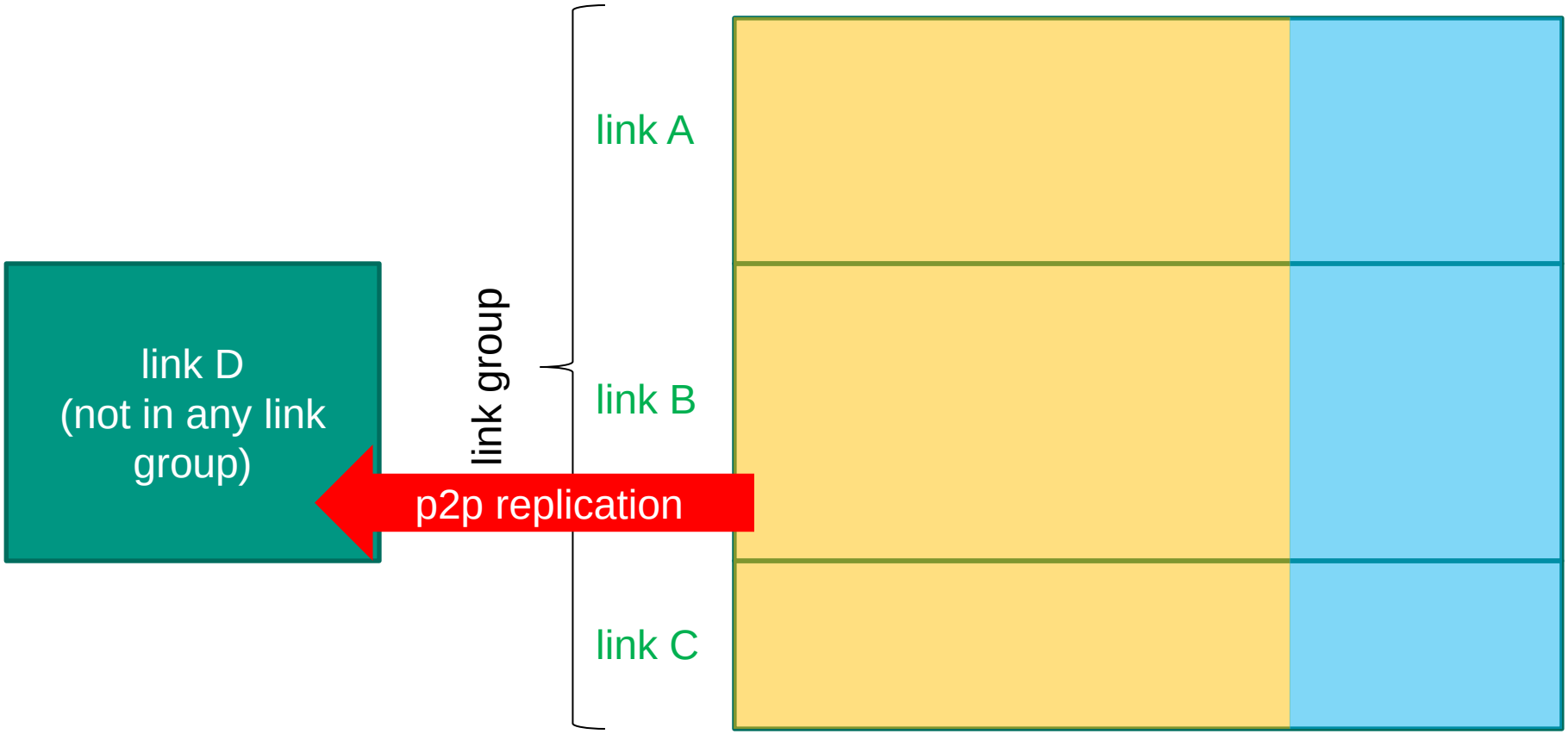
Setting up hot-replication

- This is done by using sections:
  - examples:

```
[] (PoolManager) admin >  pm set default -p2p=0.75
[] (PoolManager) admin >  pm set default -p2p=90%
```

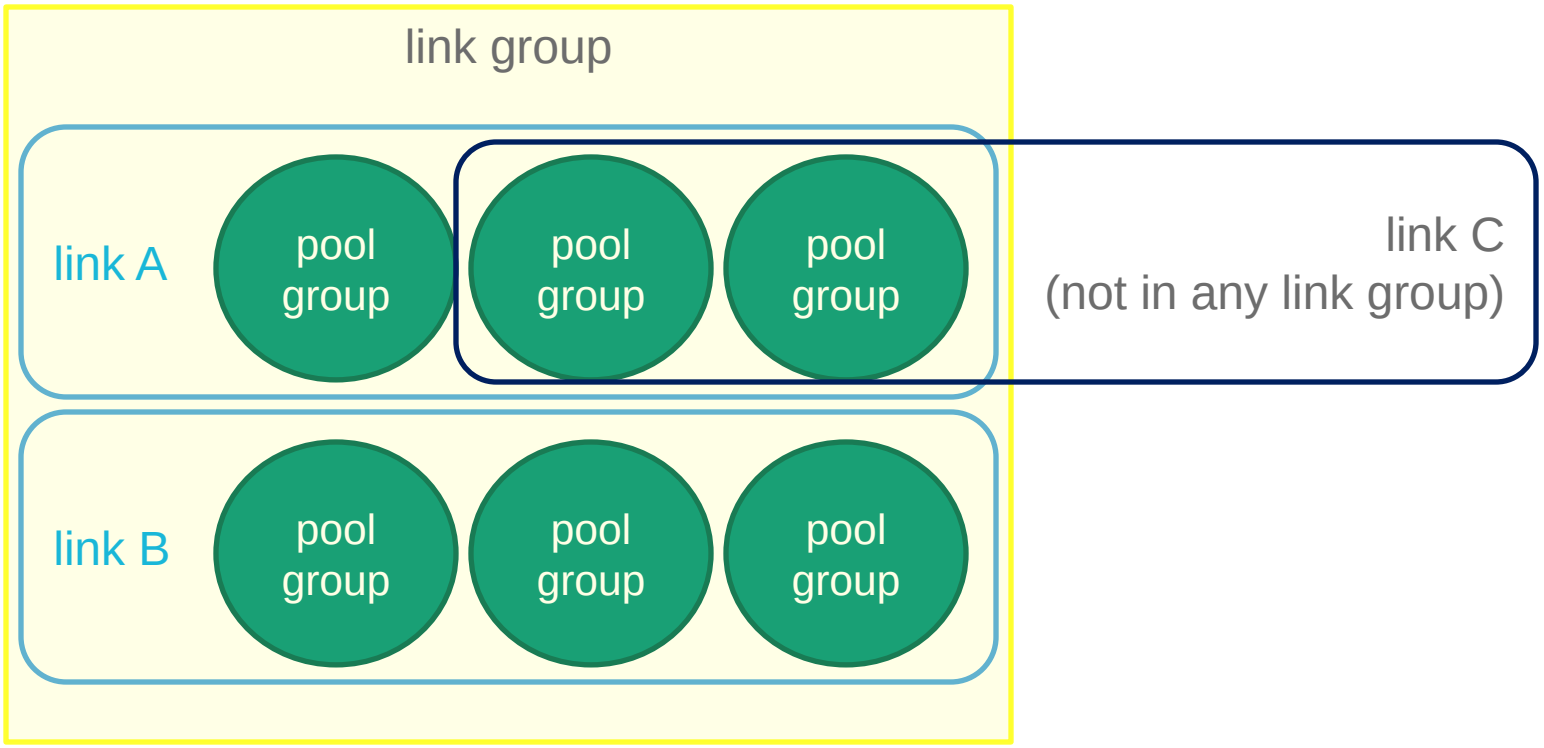
  - If p2p is set to zero, then pool to pool transfers are switched off.
  - If p2p is set to a percentile, the percentile must fulfill 0%<percentile<100%.
- Enabling/disabling pool to pool in a link associated to a section:
  - This is done via `pm set` in the PoolManager; the options are:
    - `p2p-allowed=on`: transfers will be initiated if a file resides on a pool which is not allowed to deliver this file to the client.
    - `p2p-oncost=on`: additionally to the scenario mentioned above, the cost scheme will be used.

Hot replication and space tokens (I)



- p2p replication to pools outside of space token

Hot replication and space tokens (II)



- a pool group can be in several links
- this scenario is not really supported, because it confuses the space calculation!

Summary

- Cost calculation considers:
  - number of current transfers and queue lengths,
  - free space.
- You have four parameters for tuning
  - 2 cost factors,
  - breakeven,
  - gap parameter.
- Right choice of parameters is not easy and depends on how system is used.
- Hot-spot replication:
  - is not a hot file replication,
  - offers fixed threshold scheme and percentile scheme,
  - can confuse space calculation (depends on set up).