dCache toolbox

8th international dCache users workshop

May 14-16, 2014, DESY, Hamburg

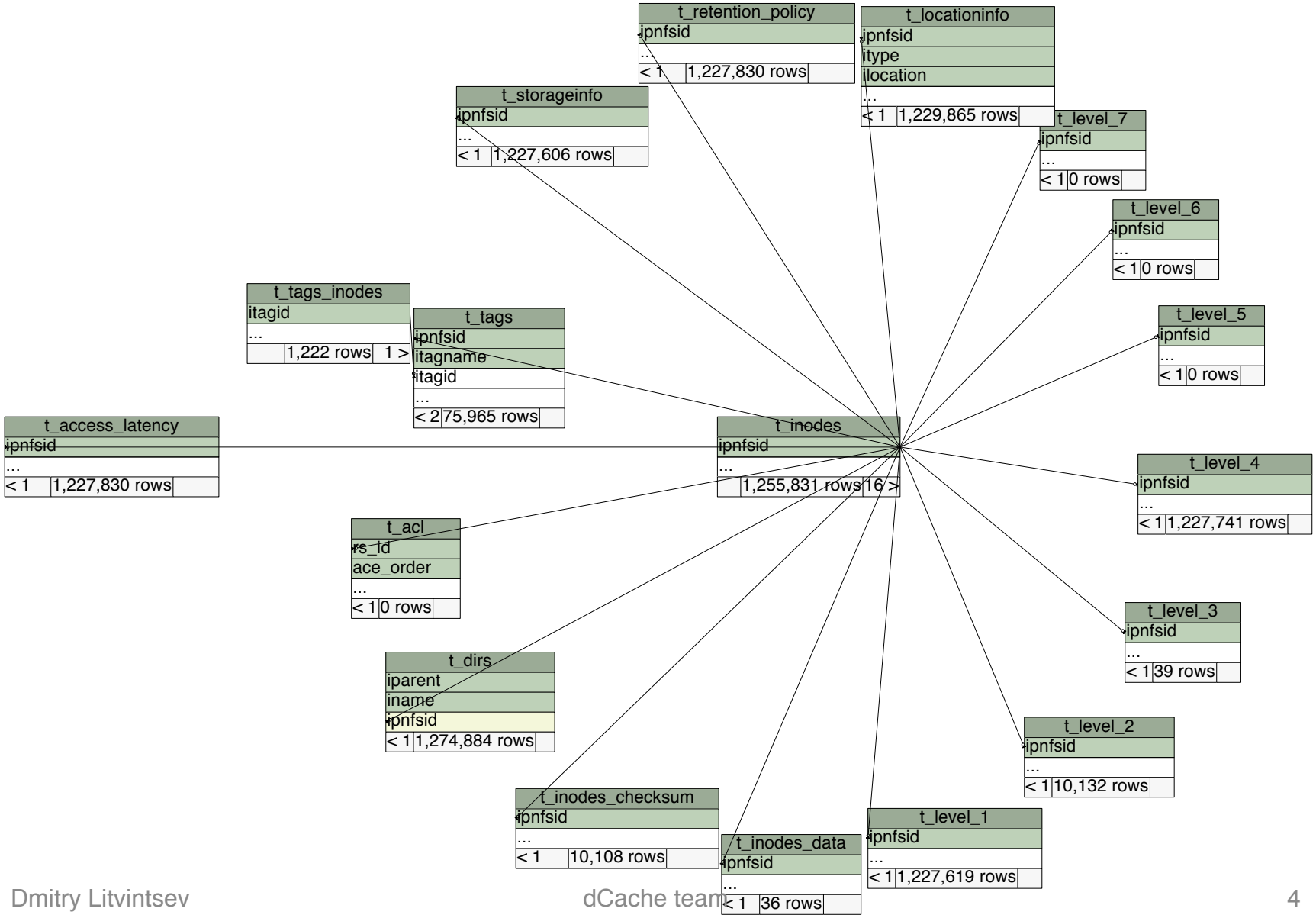
Outline

- The goal of this talk is to provide you with some concrete examples which you can use in your day-to-day dCache operations. Talk covers:
  - The joy of Chimera
  - Some pool operations
  - Consolidated logging
  - Spruce up your WebDAV

Chimera

- In the past we had PNFS – a Perfectly Normal File System where humans could not know where files came from or where they went
- Now we have DB schema to play, opening many possibilities
- Use a schema visualization tool to study Chimera.
- E.g.
  - schemaSpy + Graphviz
  - http://wiki.postgresql.org/wiki/SchemaSpy
  - http://schemaspy.sourceforge.net
  - http://srm.fnal.gov/schema/index.html

Chimera Schema



t_inodes

t_inodes		
ipnfsid		varchar[36]
itype		int4[10]
imode		int4[10]
inlink		int4[10]
iuid		int4[10]
igid		int4[10]
isize		int8[19]
iio		int4[10]
ictime		timestamptz[35,6]
iatime		timestamptz[35,6]
imtime		timestamptz[35,6]
< 0	1,255,831 rows	16 >

Generated by SchemaSpy

- itype = { 16384 : directory, 32768 : file, 40960 : link}
- iio = { 0 : data stored on pool, 1 : data stored in db (e.g. config files) or links }

t_locationinfo

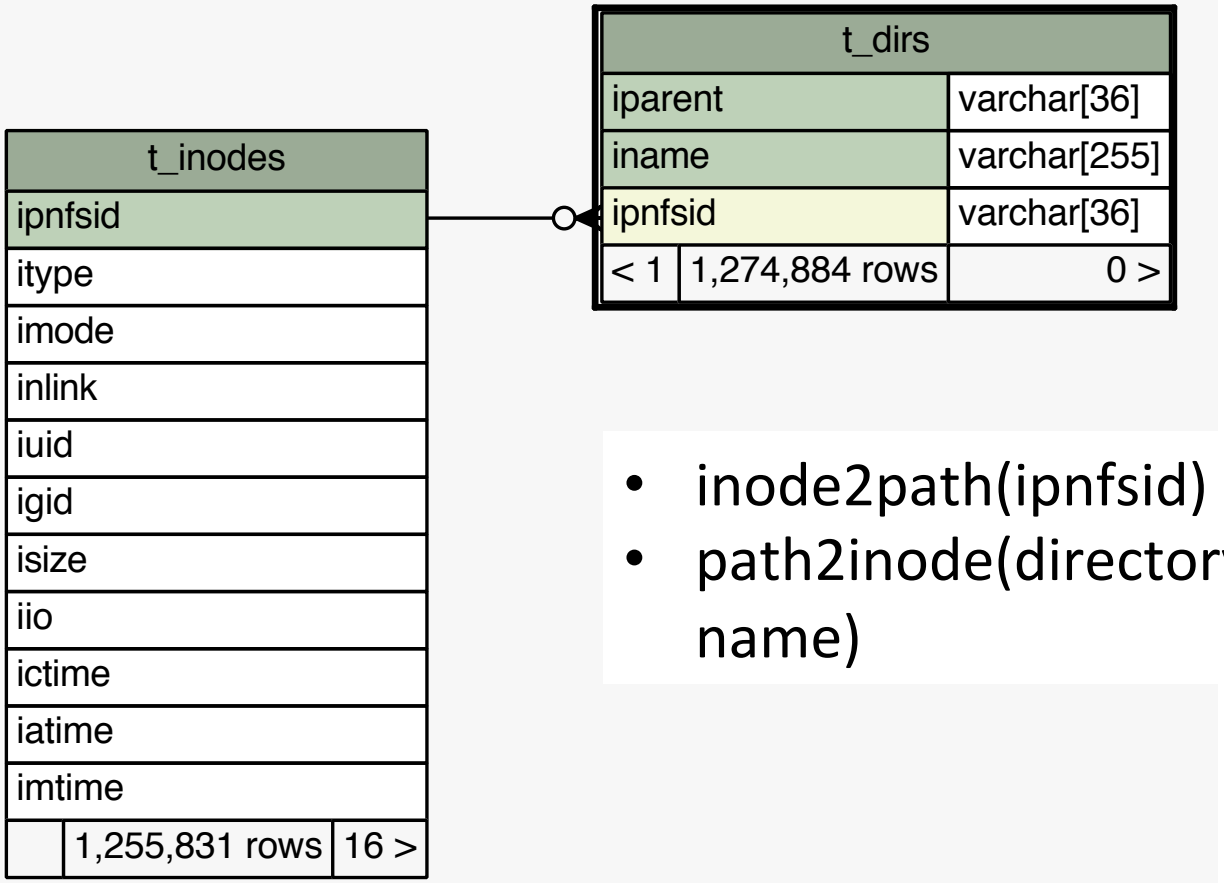
t_locationinfo		
ipnfsid		varchar[36]
itype		int4[10]
ilocation		varchar[1024]
ipriority		int4[10]
ictime		timestamptz[35,6]
iatime		timestamptz[35,6]
istate		int4[10]
< 1	1,229,865 rows	0 >

Generated by SchemaSpy

- ipnfsid is FK to t_inodes.ipnfsid
- itype = { 0 : tape, 1 : pool}
- ilocation = { 0 : URI, 1 : pool name}

- istate = { 0 : off-line, 1 : on-line} (don't think it is used)

t_dirs



- inode2path(ipnfsid)
- path2inode(directory id ,relative path name)

Handle disk-only pool down

- Task
  - Generate list of files that were lost
  - Remove lost files from name space
  - Report list of lost file names to users
- Use t_locationinfo table

Handle disk-only pool down

- Generate list of files that were lost
  - small chimera instance:

```
psql –t –A –F ‘ ‘-U chimera chimera -c "select
inode2path(ipnfsid) from t_locationinfo t where
t.ilocation=‘pool1’ not exists  (select 1 from
t_locationinfo where ilocation<>’pool1' and
ipnfsid=t.ipnfsid)” –o lost.txt
```

- Remove lost files from namespace (simple removal of files using ‘’rm” takes care of spacemanager entries)

```
cat lost.txt | while read fp; do rm –f $fp; done
```

- Send the list to higher command

Handle disk-only pool down.

- On large chimera instance
  - Get list of ids and paths on down pool:

```
pqsl … -c “SELECT ipnfsid, inode2path(ipnfsid) FROM
t_locationinfo WHERE ilocation=‘pool1’” –o pool1.txt
```

  - See if some can be salvaged (replicas in other pools):

```
psql … -c “select ipnfsid  from t_locationinfo t where
ilocation = ’pool1' and exists  (select 1 from
t_locationinfo where ilocation<>’pool1' and
ipnfsid=t.ipnfsid)” –o not_lost.txt
```

  - Create list of lost files

```
cat not_lost.txt | while read id; do sed -i "/$id/d”
pool1.txt; done
cat pool1.txt | awk ‘{print $2}’ > lost.txt
```

How many duplicates?

- How much space is “wasted” by multiple file replicas (non-resilient case)
- Use t_locationinfo table

```
SELECT tl.ipnfsid, count(*), sum(ti.isize) AS space
    INTO duplicates
    FROM t_locationinfo tl, t_inodes ti
    WHERE tl.ipnfsid=ti.ipnfsid and tl.itype=1
    GROUP BY tl.ipnfsid having count(*) > 1
```

- “wasted” space:

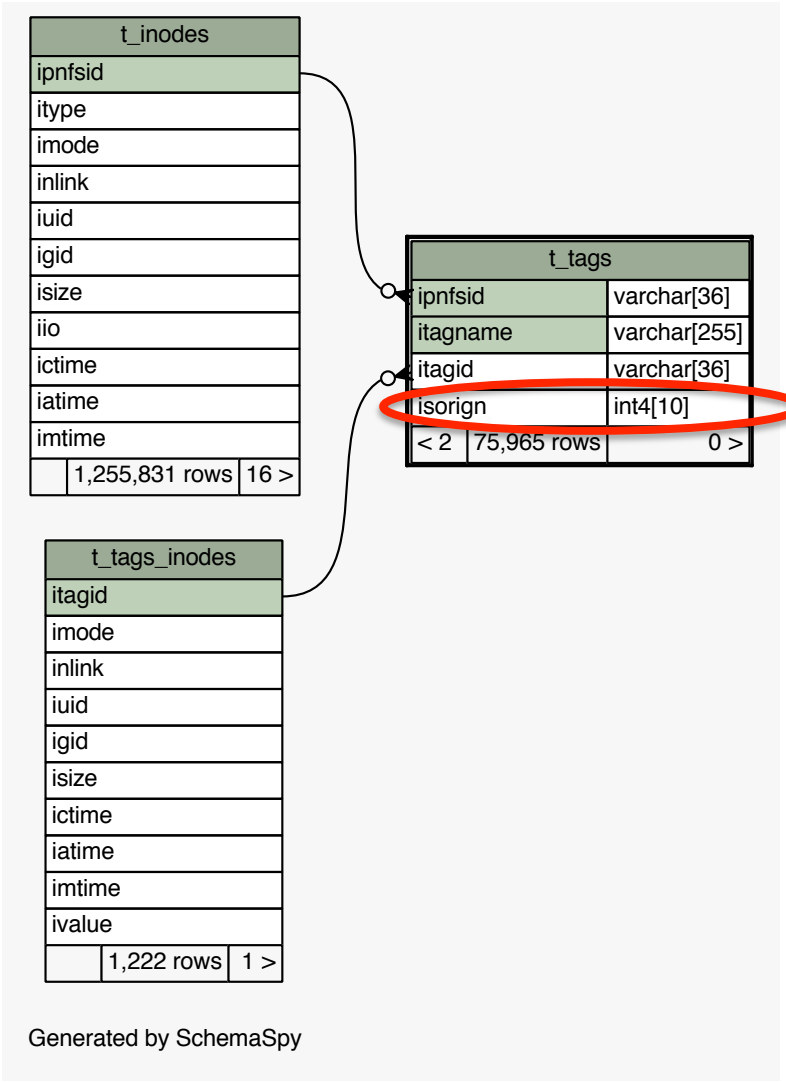
```
SELECT SUM(space) FROM duplicates;
```

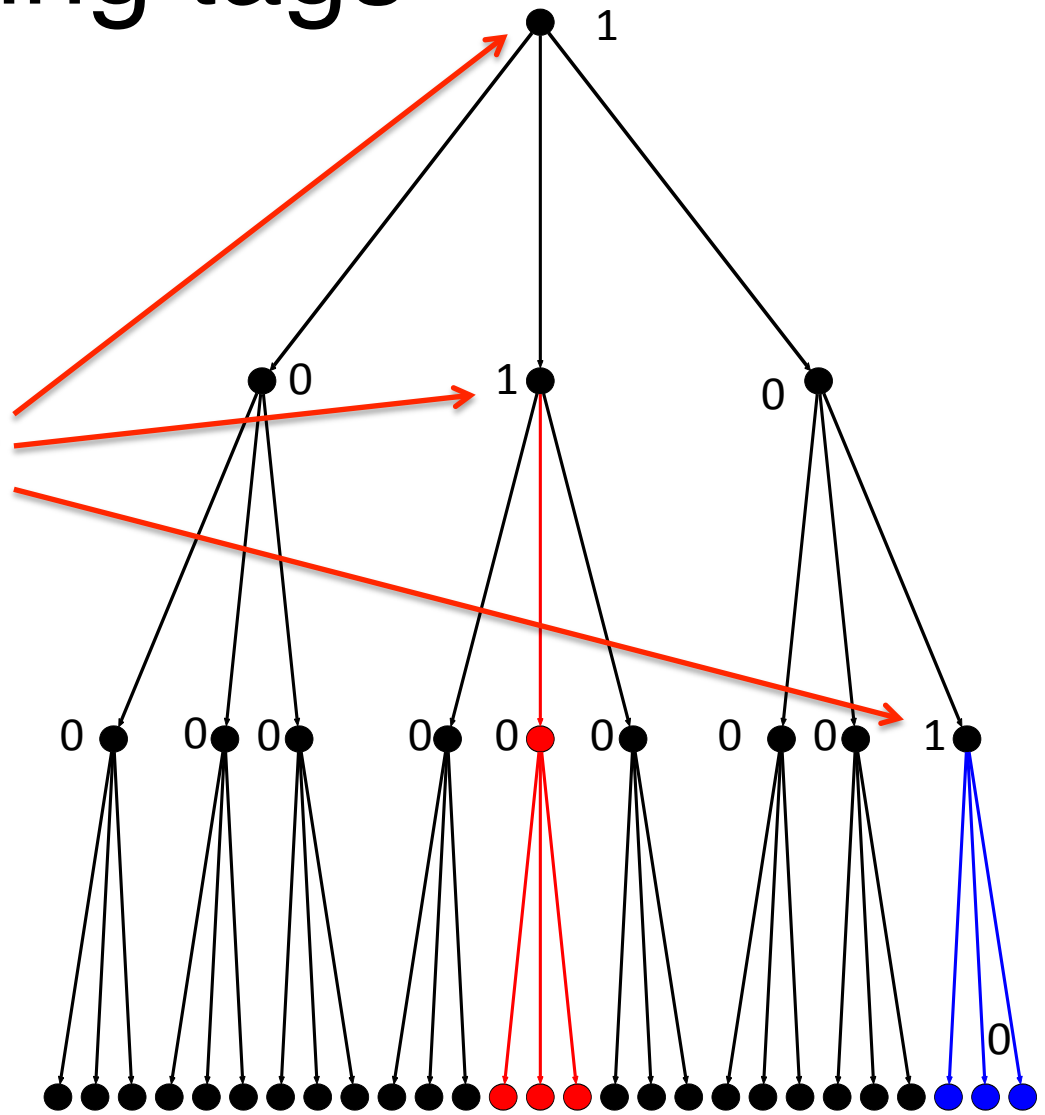
How many duplicates?

- Should you get rid of them?
  - I use jython API to connect to admin interface so that I can write python/Java scripts to deal with stuff like this

```
pooldata = sendCommand("PoolManager", "psu ls –l pgroup readWritePools")
pool_list = [a.strip().split()[0] for a in string.split(pooldata,"\n") \
             if a and a!="" ]
for pool in pool_list:
   repls = sendCommand(pool, "rep ls")
   for line in repls.split("\n"):
      pnfsid=line.split()[0]
      cacheLocations=sendCommand("PnfsManager", "cacheinfoof "+pnfsid)
      pools = [a for a in cacheLocations.strip().split() \
              if a != pool ]
      for p in pools:
         cmd=sendCommand(p,"rep rm "+pnfsid)
```

Fixing tags


t_inodes
ipnfsid
itype
imode
inlink
iuid
igid
isize
iio
ictime
iatime
imtime
1,255,831 rows
16 >
t_tags
ipnfsid
varchar[36]
itagname
varchar[255]
itagid
varchar[36]
isorign
int4[10]
< 2
75,965 rows
0 >
t_tags_inodes
itagid
imode
inlink
iuid
igid
isize
ictime
iatime
imtime
ivalue
1,222 rows
1 >
Generated by SchemaSpy


1
0
1
0
0
0
0
0
0
0
0
0
0
1
0
0

Fixing Tags

- In PNFS : if you modify tag in the middle of the tree, the tag changes in the whole sub-tree under the changed directory.
- In Chimera: if you modify tag in the middle of the tree, only this directory tag gets changed. Only new created subdirectories under this directory will have modified tag.

Repair Tag Inheritance

```
create or replace function push_tag(varchar, varchar)
returns void as $$
declare
  root varchar := $1;
  tag varchar := $2;
  subdirs RECORD;
  tagid varchar;
begin
select into tagid itagid from t_tags
   where ipnfsid = root and itagname = tag;
for subdirs in
  select t_inodes.ipnfsid, t_dirs.iname from t_inodes, t_dirs
     where t_inodes.itype=16384 and t_dirs.iparent=root and
           t_inodes.ipnfsid=t_dirs.ipnfsid and
           t_dirs.iname not in ('.', '..') loop
     begin
        delete from t_tags
           where ipnfsid = subdirs.ipnfsid and itagname = tag;
        insert into t_tags values (subdirs.ipnfsid, tag, tagid, 0);
        perform push_tag(subdirs.ipnfsid, tag);
      end;
 end loop;
end;
$$
language 'plpgsql';
```

Tag Oops

- My finger slipped:
  `echo "12345" > ".(tag)(WriteFoken)"`
  How do I get rid of this tag?
- Install dCache 2.6 and use chimera CLI
  `rmtag path tag`

```
chimera:/# help rmtag path tag
NAME
        rmtag -- remove tag from directory

SYNOPSIS
        rmtag path tag


chimera:/#
```

Full File Listing

- Use this query to produce full file listing :

```
WITH RECURSIVE paths(pnfsid, path, type, fsize)
AS (VALUES('000000000000000000000000000000000000',''',16384,0::BIGINT)
   UNION SELECT d.ipnfsid, path||'/'||d.iname,i.itype,i.isize
   FROM t_dirs d,t_inodes i, paths p
   WHERE p.type=16384 AND d.iparent=p.pnfsid AND
         d.iname != '.' AND d.iname != '..' AND i.ipnfsid=d.ipnfsid)
          SELECT p.path, p.fsize::bigint, cs.isum
           FROM paths p, t_inodes_checksum cs
            WHERE p.type=32768 AND
                  cs.ipnfsid=p.pnfsid
...

/pnfs/fs/usr/snoplus/scratch/BiPo61.root 2819720 db4e5b30
/pnfs/fs/usr/snoplus/scratch/BiPo60.root 3104212 3e5651e3
/pnfs/fs/usr/snoplus/scratch/BiPo62.root 2973547 1f0da349
```

Space Usage by UID/GID

```
SELECT ti.iuid, ti.igid,
  sum(case when tl.itype=1 then ti.isize else 0 end) AS SPACE_ONLINE,
  sum(case when tl.itype=0 then ti.isize else 0 end) AS SPACE_ONTAPE,
  sum(case when tl.itype=1 then 1 else 0 end) AS FILES_ONLINE,
  sum(case when tl.itype=0 then 1 else 0 end),0) AS FILES_ONTAPE
FROM t_inodes ti, t_locationinfo tl
WHERE ti.itype=32768 AND tl.ipnfsid=ti.ipnfsid
GROUP BY ti.iuid, ti.igid order by SPACE_ONLINE desc;
```

iuid	igid	space_online	space_ontape	files_online	files_ontape
47683	9555	347202447548091	286771538790022	1179111	1069129
42417	9553	282293594518114	289597478116687	1350443	1308080
47823	9985	273868375870321	280852823940364	55368	56569
45172	9553	254982411783412	161145305051524	2346397	1196839
0	0	109253752746703	940810041910686	39162	209996
43506	9919	97882589611603	106504558327532	60543	57895
1121	1540	73411737935590	134730598744043	3924	10086
42411	5111	48683707523898	392934545663541	66905	1300172

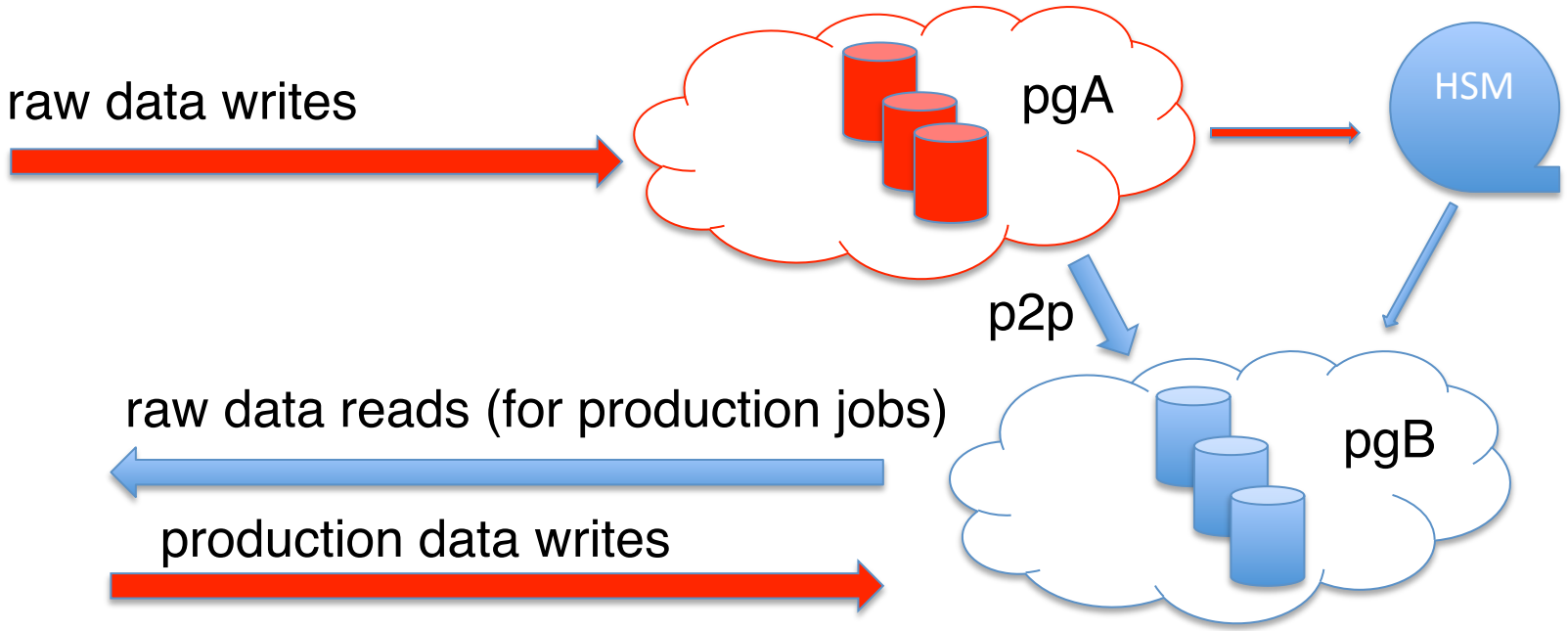
‘du’ on a directory

```
create or replace function dir_size(varchar)
returns BIGINT as $$
declare
  dir_id   VARCHAR := $1;
  ssum     BIGINT  := 0;
  children RECORD;
begin
for children in select t_inodes.ipnfsid, t_inodes.isize,t_inodes.itype
    from t_inodes, t_dirs
    where t_inodes.itype<>40960 and
          t_dirs.iparent=dir_id and
          t_inodes.iio=0 and
          t_inodes.ipnfsid=t_dirs.ipnfsid and
          t_dirs.iname not in ('.', '..') loop
             IF children.itype = 32768 THEN
               ssum := ssum + children.isize;
             ELSE
               ssum := ssum + dir_size(children.ipnfsid);
             END IF;
end loop;
return ssum;
end;
$$
language 'plpgsql';
```

‘du’ on a directory

```
chimera=# select
inode2path('00007EA6B4B0343A4A89835D9080A03FA45A'),dir_siz
e('00007EA6B4B0343A4A89835D9080A03FA45A');
     inode2path     |    dir_size
--------------------+-----------------
 /pnfs/fs/usr/minos | 659479573862539
(1 row)
```

How to “replicate” files between pgroups(links)



- Setup a permanent migration job. For each pool in source pgroup pgA

```
(pool_1) migration copy –tmode=cached –target=pgroup –permanent pgB
```

Decommission a Pool

- Set the pool read-only

```
(local) admin >  cd <pool>
(<poolname>) > disable –rdonly
```

- Move data off of the pool

```
(<poolname>) > migration move -target=pool|pgroup|link <name>
```

- Make sure no files left on the pool by running rep ls.
  - Handle left overs as necessary
- Stop the pool

How to find OFFLINE cells

- Setup a periodic job that does one of the following:
  - Scrapes http://<host>:2288/cellInfo (bad idea)
  - Uses shell/python/jython/Java to talk to collector@httpdDomain

```
res=sendCommand("collector@httpdDomain","dump info")
for line in res.split("\n"):
  data=line.strip().split()
  if len(data) != 8 :
     listOfOfflineCells.append(data[1])
if len(listOfOfflineCells)>0 :
   ... send mail ...
```

  - Uses info + XSLT style-sheets

```
output=$(xsltproc --stringparam cells "PoolManager PnfsManager ..." \
 /usr/share/dcache/xml/xslt/wait-for-cells.xsl \
 http://<host>:2288/info/domains)
if [ "${output:-NONE != "NONE" ]; then echo "$output" | mail ... ; fi
```

How to get rid of OFFLINE cells

- Inversely - suppose I stopped the pools on purpose and now I am inundated by all these found OFFLINE cells emails (I have setup previously). How to stop it?

```
$ ssh –c blowfish -i ... –p <port> admin@<host> << EOF
cd collector@htpdDomain
unwatch cell@domain
EOF
```

Log Consolidation: use rsyslog

- edit `/etc/dcache/logback.xml`
  - Add new appender, let's call it "syslog"

```
<appender name="syslog" class="ch.qos.logback.classic.net.SyslogAppender">
<syslogHost>host.example.org</syslogHost>
<facility>USER</facility>
<suffixPattern>\\(%X{cells.cell}\\) [%X{org.dcache.ndc}] %m%n</suffixPattern>
</appender>
```

  - Add "syslog" appender to root of log hierarchy and to dummy logger

```
<root>
   <appender-ref ref="syslog"/>
   …
</root>
```

```
<dummy>
   <appender-ref ref="syslog"/>
   …
</dummy>
```

See Johan Guldmyr's post to dCache user forum

Log Consolidation: use rsyslog

- Configure rsyslog host to receive log messages
  - edit file `/etc/rsyslog.conf`, uncomment

```
$ModLoad imudp
$UDPServerRun 514
```

  - Add remote rule before any local rule. E.g.:

```
if $fromhost-ip startswith '131.169.6' then /var/log/dcache.log
&~
```

  - Restart rsyslog

My WebDAV looks dull

- My WebDAV interface looks dull and is missing upload functionality
- Drop this IT Hit add-on on it:
  http://www.dcache.org/chimera/dcache-webdav-2.3.0SNAPSHOT-1.noarch.rpm