# Resilient dCache

## Alex Kulyavtsev, FNAL
## For the dCache Team

# Resilience Manager Rationale

- ☐ T2 data centers often do not have tape backup for disk storage. Data generated on worker nodes later will be transferred to T1 data center.
- Improve **reliability** of disk data storage using **inexpensive** solution.
- ☐ When pool with single replica goes down, file can not be read anymore.
- Increase **availability** of the data.

# Resilience Manger

- a.k.a. Replica Manager (RM).
- "**File**" in distributed storage system: metadata for data "*somewhere*" in the storage.
- File **replica** – concrete copy of the file in some pool.
- Resilience Manager counts number of file replicas in pools in *online* and *offline* state and keeps

  Min <= N replicas <= Max

Warning: definition of "*offline*" state of the pool in RM differs from OFFLINE pool state on "Cells" status page.

- RM responds to messages
  - File added to/removed from pool
  - File removed from pnfs
  - Pool state change

# Resilient dCache

- **Resilient dCache** =

    dCache + Resilience Manager

All pools are considered as resilient.

- **Hybrid dCache**:

Only some pools are "resilient" as defined in the pool group ResilientPools in file PoolManager.conf *on the system start*.

Other pools are not resilient – volatile pools, HSM pools.

# Pool States

- **online** – "alive" pool in the system.
- **down** – pool crashed  or  going out of the system <u>permanently</u>,  files in pool are not safe.
- **offline** – pool goes out of the system <u>temporarily</u>, files in the pool stay intact.
- **drainoff** – preparation state before *down state*. Copy *unique* files out of the pool.
- **offline-prepare** – transient, copy *unique* files out of this pool.

# Pool States



Pool State,
Resilient dCache v1.0
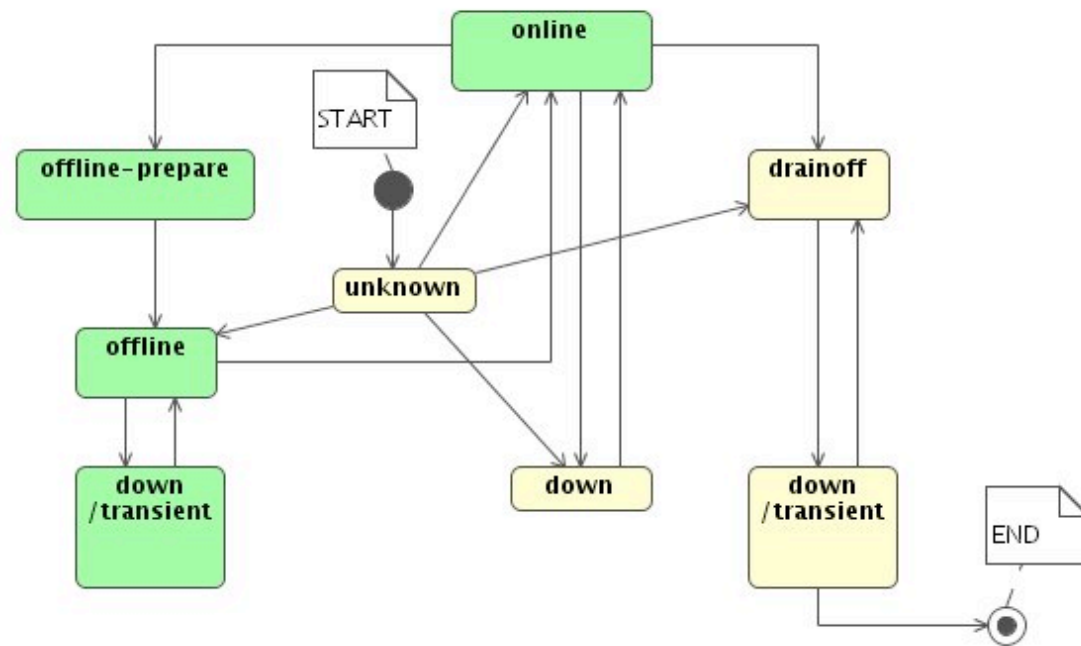
# 1: Initial state, 2<= N <= 3

□ All pools are online

| | Pool 1 | Pool 2 | Pool 3 | Pool 4 | Pool 5 | Count |
|---|---|---|---|---|---|---|
| File A | A | A | | | | 2 |
| File B | B | | B | | | 2 |
| File C | | C | C | | | 2 |
| File D | | | D | D | | 2 |
| | online | online | online | online | online | |

# 2: Pools 1 and 2 went down

☐ Can't access File A; replicate B and C

| | Pool 1 | Pool 2 | Pool 3 | Pool 4 | Pool 5 | Count |
|---|---|---|---|---|---|---|
| File A | A | A | | | | 0 |
| File B | B | | B → *B* | | | 1 |
| File C | | C | C → | | *C* | 1 |
| File D | | | D | D | | 2 |
| | down | down | online | online | online | |

# 2: Set pools 1 and 2 to drainoff

☐ File A extracted from pool 1

|        | Pool 1 | Pool 2 | Pool 3 | Pool 4 | Pool 5 | Count |
|--------|--------|--------|--------|--------|--------|-------|
| File A | A | A | | | *A* | 0 |
| File B | B | | B | *B* | | 1 |
| File C | | C | C | | *C* | 1 |
| File D | | | D | D | | 2 |
|        | drainoff | drainoff | online | online | online | |

# 2: set pool pool_1 offline

☐ Temporarily take pool out, no replication

|  | Pool 1 | Pool 2 | Pool 3 | Pool 4 | Pool 5 | Count |
|--------|---------|--------|--------|--------|--------|-------|
| File A | A | A |  |  |  | 2 |
| File B | B |  | B |  |  | 2 |
| File C |  | C | C |  |  | 2 |
| File D |  |  | D | D |  | 2 |
|  | offline | online | online | online | online |  |

# Replica count

- ☐ RM keeps data about replicas and pools in postgres Database
- ☐ When pool reconnects or comes to 'online' state, RM requests the pool to send a list of file replicas currently in the pool and updates DB.
- ☐ RM evaluates several expressions on DB updates:
  - Counts *deficient* and *redundant* replicas in *online* and *offline* pools only (*down*, etc. are ignored).
  - Counts *unique* replicas in drainoff and *offline-prepare* pools

# Excluded files

- ☐ Sometimes things go wrong during *replication* or *reduction*
- ☐ RM can retry replication
- ☐ RM can *exclude* file from further consideration (from *adjustments*)
- ☐ User can *release* file to continue
- ☐ User can *clear* file from DB - e.g. when file was removed from pnfs.

# Admin interface

ls pnfsid

*Lists pools where file is found*

set pool pool_1 offline

show pool_1

set pool pool_1 drainoff

ls unique pool_1

*prints count on screen and dumps list in the log file.*

*see more commands in RM cell "help"*

# Monitoring

- Tomcat servlet displays
  - Status of replica manager threads
  - Pools status
  - List of 'exluded' files

- More details in documentation

# Startup

- □ Cold Start – All system starts. Final list of 'online' pools is unknown. Pools start slowly, some will not start at all. Wait for "quorum" for period of time long enough for all pools to start, then start replication.

- □ Hot restart – continue operation.

  RM stores status of pools in DB. When all pools last seen "online" are reconnected, RM does not wait for other pools and starts replications and removals.

# RM requires special treatment

- ☐ Replica Manager is service on the top of dCache, it proactively copies and removes files. Does not remove single replica.

- ☐ If we do not start half of the pools or shutdown some pools when RM is active, Replica Manager may start massive file replications to compensate for "missing" files.

- ☐ Inform RM about pools status change (set them *offline)* or shutdown RM first.

- ☐ Do not overflow pools, remove files from pnfs promptly

# Links

- Currently recent documentation is linked from USCMS T1 dCache web page,

  http://cmsdca.fnal.gov/

- See "Resilient dCaches Documents ":

  Monitor     - RM monitoring

  Fixing      - RM troubleshooting

  Install      - Installation instructions. RM is in dcache distribution now.

Manual – linked as "Resilient Manager" from pages above

http://cmsdcam.fnal.gov/dcache/resilient/Resilient_dCache_v1_0.html