Resilient dCache (redux)

11th international dCache users workshop

May 29-30, 2017, #NeIC2017

Umeå, Sverige

Dmitry Litvintsev, Abert Rossi (Fermilab)

resilience

All Books Images News Videos More Settings Tools

About 59,700,000 results (0.70 seconds)

re·sil·ience

/rəˈzilyəns/

noun

noun: **resilience**; plural noun: **resiliences**; noun: **resiliency**; plural noun: **resiliencies**

1. the capacity to recover quickly from difficulties; toughness.
   "the often remarkable resilience of so many British institutions"
2. the ability of a substance or object to spring back into shape; elasticity.
   "nylon is excellent in wearability and resilience"

Translate resilience to Choose language

Use over time for: resilience

Show less

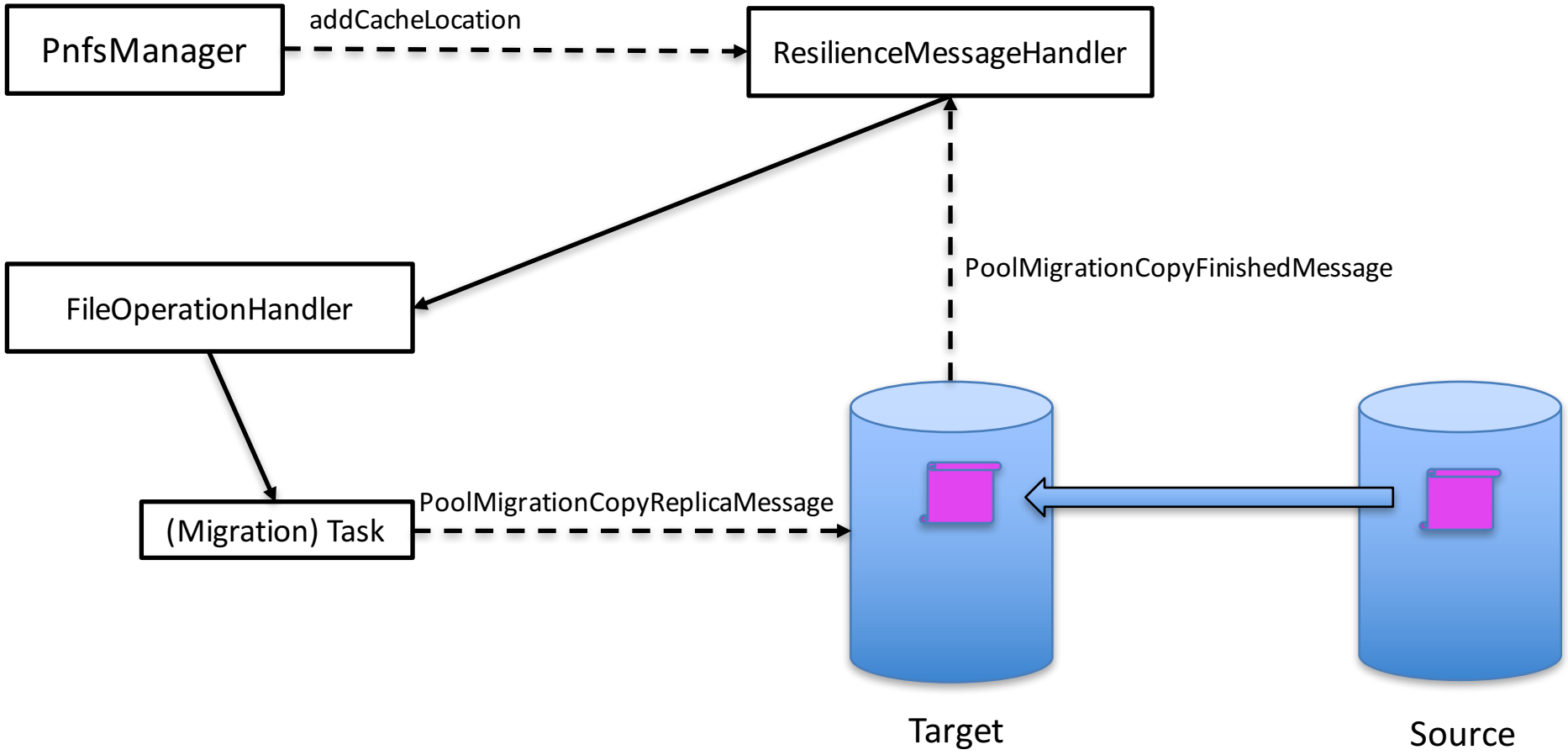
Resilience in dCache

- (File) Resilience in dCache is realized by utilizing pool-to-pool (p2p) transfers to create and maintain multiple file replicas across different pools within a pool group providing improved file availability and durability.
- This capability plays an important role in QoS system currently being developed by dCache collaboration within INDIGO project. Provides basis for an a la carte selection of file availability and durability policies.

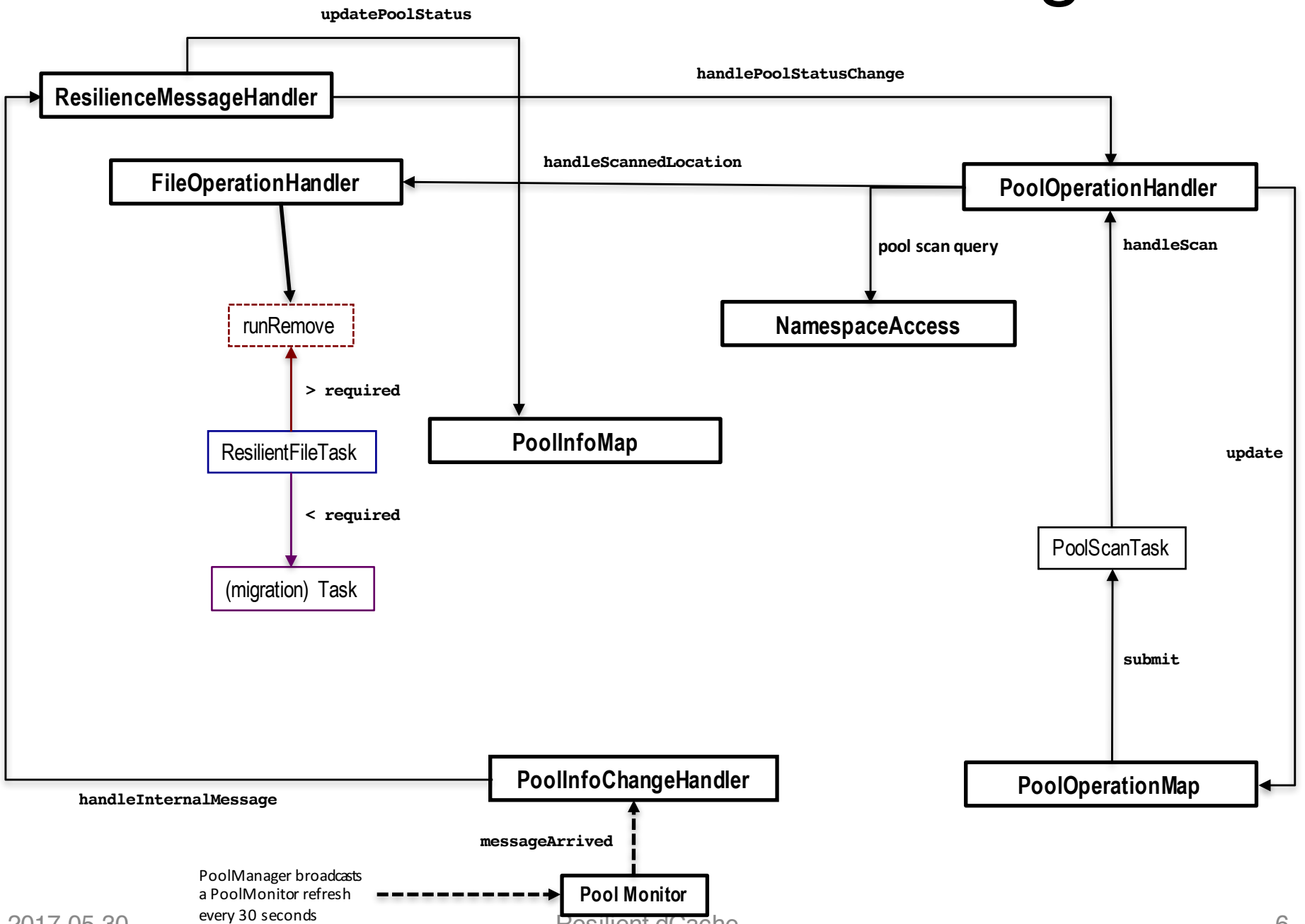
Resilient dCache

- New Resilience Service was introduced in dCache version 2.16
- Completely independent from existing (legacy) Replica Manager which hasn't changed much since 2007 and is effectively de-supported.
- Resilience Service overcomes major limitations of the Replica Manager and provides functionality and rich set of features that meet customer requirements as well as aligns well with modernized dCache code base and future/ongoing QoS system development.

Create new replica



Handle Pool status change



Old vs new comparison

Feature/Implementation detail	Replica Manager	Resilient Service
Dedicated DB to keep replica counts and pool status	YES	NO
Flexible Resilience constraints	NO (defined per Replica Manager Instance)	YES (defined per storage group)
File replication criterion	AccesLatency = NEARLINE(ONLINE) RetentionPolicy = CUSTODIAL	AccessLatency=ONLINE
Partition replicas across pool	YES (based on pool hostname tag)	YES (based on arbitrary or combination of pool tags)
Requires special pool setup	`lfs=precious`	NO
Change in resilience constraints requires service restart	YES	NO
Adding/removing pools from pool group requires service restart	YES	NO
Memory intensive pool repository scan	YES	NO (replaced with chimera location info scan)

NOT GOOD

GOOD

NEUTRAL

Old vs new comparison (Cont.)

Feature/Impl. detail	Replica Manager	Resilient Service
Non-fatal failures are retried automatically	YES	YES
Broken and corrupt replicas are handled by removal and recopying	NO	YES
Alarm is raised if there is a fatal replication error	NO	YES
Scalability	NO (Huge memory overhead for large pools when performing pool scans)	YES (verified by extensive testing)
Admin interface	Very limited	Full featured

Adding Resilient Service

- Add

  ```
  [domainName]
  [domainName/resilience]
  ```

  to your layout file.
- Since Resilience talks to Chimera DB directly, the Chimera connection parameters need to be defined.
- It is best to run Resilience in its own domain and allocate at least 8 GB of JVM heap memory space to it.

Configuring Resilience

- A file becomes resilient if it:
  - Has storage group that has *resilient storage unit* defined.
  - Has AccessLatency = ONLINE.
- Resilient storage unit is defined by setting *required* and optionally *onlyOneCopyPer* properties.
- A pool group is defined as resilient by adding *–resilient* flag to the pool group definition.
- *Resilient storage unit* must be linked to resilient pool group when defining link.
- Hot pool replication must be disabled on the link containing resilient pool group.
- A pool may belong to only one resilient pool group.

Storage unit definition details

- Two attributes are added to storage unit definition:
  - `required=<number of copies>` a mandatory (for Resilience) requirement of how many file replicas to maintain.
  - `onlyOneCopyPer=<tag1,tag2,tag3>` an optional parameter that allows to specify comma separated list of pool tags. Resilience will take this into account and place replicas such that no two replicas belong to pool with the same set of tag values. For instance to place replicas on different hosts this attribute needs to be set to *hostname* and a hostname tag needs to be added to pool layout configuration.

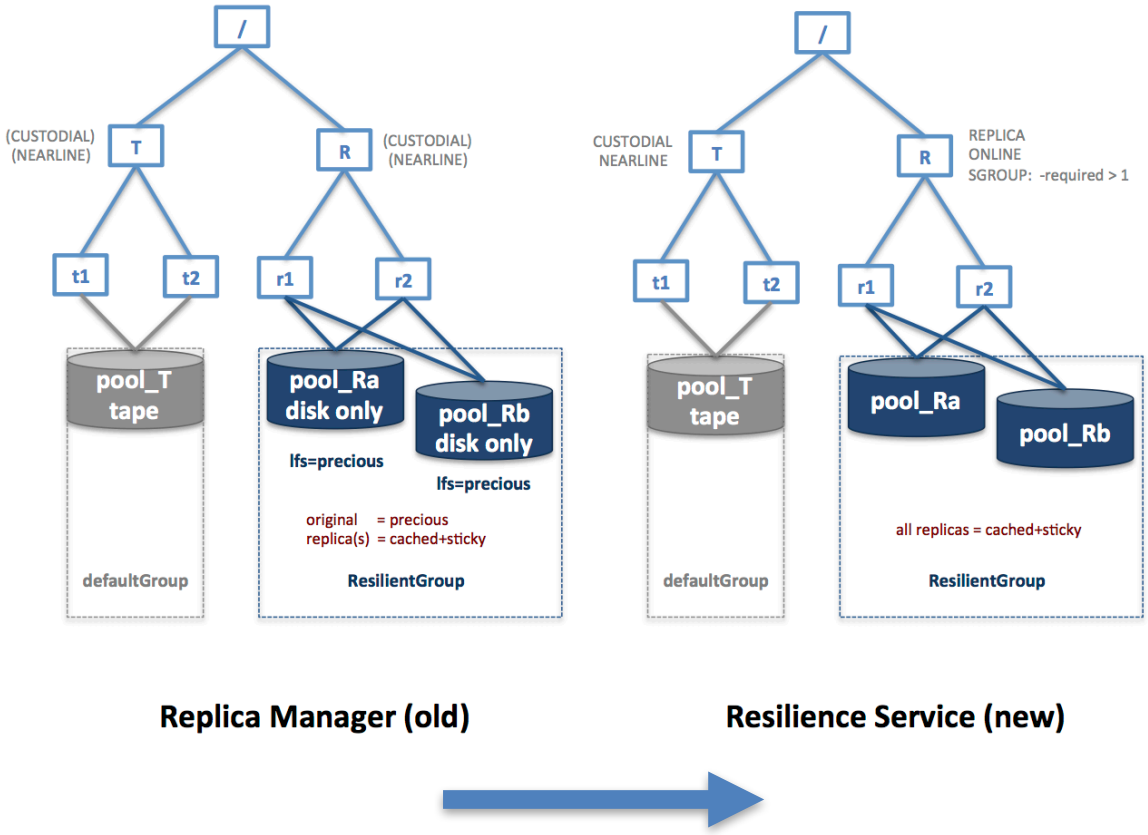
Example

```
psu create unit -store foo:bar@osm
psu set storage unit foo:bar@osm -required=2 -onlyOneCopyPer=hostname

psu create ugroup resilient-ugroup
psu addto ugroup resilient-ugroup foo:bar@osm

psu create pool pool1

psu create pgroup resilient-pools -resilient
psu addto pgroup resilient-pools pool1

psu create link resilient-link resilient-ugroup
...
psu add link resilient-link resilient-pools
```

Directory containing files to be replicated has to have these tags

```
echo “ONLINE” > “.(tag)(AccessLatency)”
echo “bar” > “.(tag)(sGroup)"
```

Migration from old Replica Manager



General Steps

- Change *AccessLatency* and *RetentionPolicy* of current files and directories belonging to the resilient (disk-only) pools.
- Change the metadata of existing replicas on the resilient pools.
- Configure the `poolmanager.conf` with the necessary resilience attributes.

These steps can be performed on a live system provided old Replica Manager is stopped and Resilient Manager is not (yet) running.

AccessLatency/RetentionPolicy

- On chimera host run script

```
/usr/share/dcache/migration/migrate_from_repman_to_resilience.py
```

```
python migrate_from_repman_to_resilience.py -help
Problem parsing options
Usage: -help  -i <inputfile> -o <outputfile> {-H <host>[localhost] -D
<dbname>[chimera] -U <user>[dcache] -P <port>[5432]}
```

inputfile - '\n' separated list of resilient pools

outputfile - list of directories containing the files (to be used if creating new storage group tags is necessary e.g. in cases where system had multiple replica managers).

- The script does the following:
  - Queries chimera for all files in specified pools.
  - Changes their AccessLatency/RetentionPolicy to ONLINE/REPLICA.
  - Walks up the directory tree containing these files and changes directory tags (if present) to ONLINE/REPLICA.

On the resilient pools

- Make sure that all replicas are *sticky.*
  - Old replica manager kept original file replicas as *precious* and copies as *cached+sticky*. Cached replicas were ignored.
  - Resilience manager assumes that *cached* replicas belong to NEARLINE/CUSODIAL files.
  - There should be no *cached* only replicas of ONLINE/REPLICA files after migration. Therefore execute on all resilient pools:

```
rep set sticky -all
```

On resilient pools

- *Precious* replicas:
  - If old pool will continue to be used as is, the replicas may remain *precious*.
  - If pool is intended to hold replicas of CUSTODIAL and REPLICA files then precious replicas have to be made cached+sticky. Run this migration task (that changes only replica metadata):

```
migration copy -state=precious -smode=cached+system \
-tmode=cached+system -target=pgroup <pgroupname>
```

  - Remove lfs=precious flag from pool layout configuration.

PoolManager configuration

- Simplest case: entirely disk-only with only one resilience requirement:

```
psu set storage unit *@* -required=<number of copies>

# define resilient group(s)

psu create pgroup <name1> -resilient
psu create pgroup <name2> -resilient
…
```

- set

```
pnfsmanager.default-access-latency = ONLINE
pnfsmanager.default-retention-policy = REPLICA
```

PoolManager configuration

- Disk-only part has only one resilience requirement, but the system also has files which are written to tape.
  - Setup resilient part:

```
psu create pgroup resilient -resilient
psu create unit -store foo:production_resilient@osm
psu set storage unit foo:production_resilient@osm -required=2
```

```
echo “production_resilient” > “.(tag)(sGroup)”
echo “ONLINE” > “.(tag)(AccessLatency)”
echo “REPLICA” > “.(tag)(RetentionPolicy)”
```

- Above setup adds resilient disk-only pool group to a system with default pool group that stores CUSTODIAL/NEARLINE files.

Concluding Remarks

KEEP
CALM
AND
HAVE
RESILIENCE