Data Resilience in the dCache Storage System

A. L. Rossi (Fermilab) for the dCache collaboration

File Replication and dCache QoS

File durability and availability play a fundamental role in dCache Quality of Service. In the absence of a tertiary persistent back-end (i.e., in disk-only systems), dCache provides for them by using pool-to-pool replication to create multiple permanent copies of a file. The pre-set, configurable number of replicas of each file is maintained even when pools holding some of the replicas go offline.

The Original Replica Manager: History and Limitations

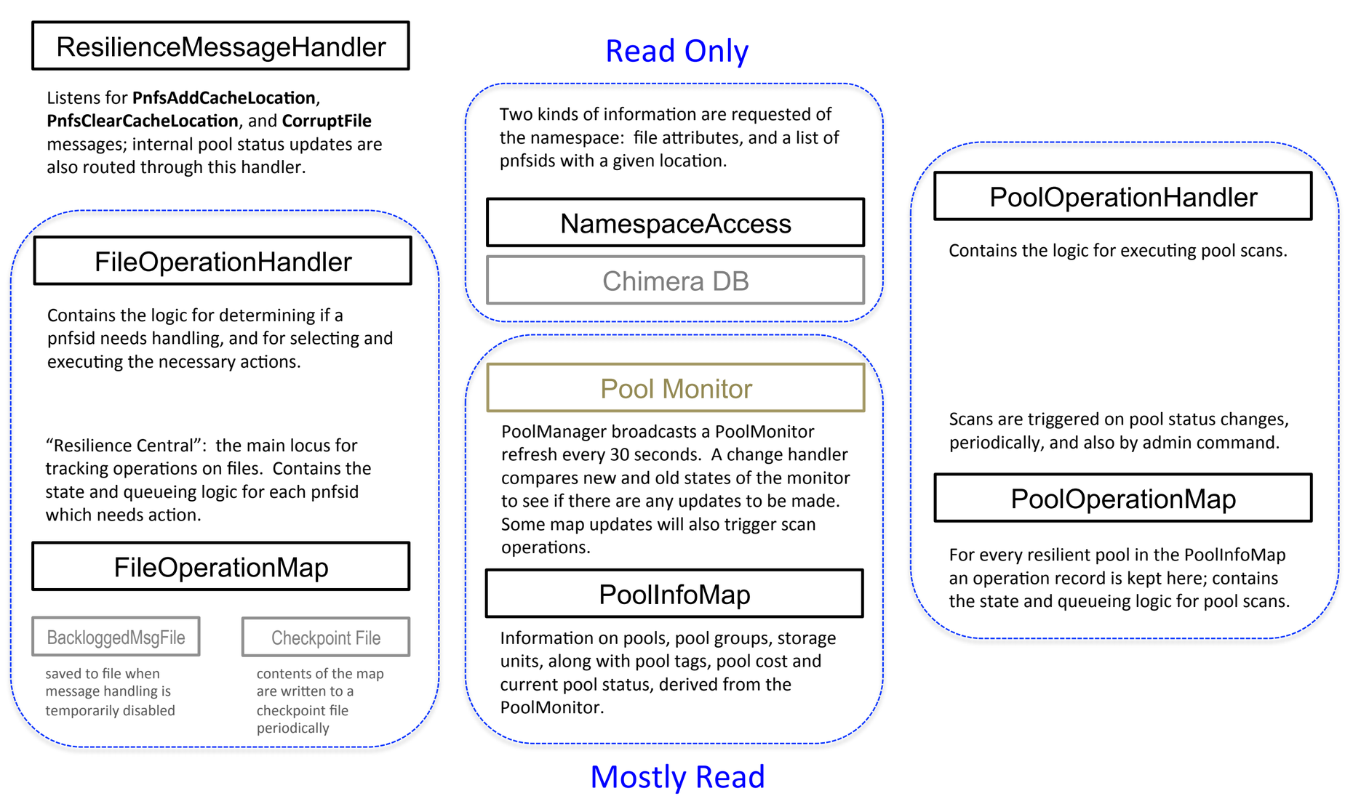
The original service used an RDBMS backend which remained unchanged since 2007. It is now obsolete because it has not kept up with major structural improvements to dCache. Other issues:

1. The database mirrored information now stored elsewhere (i.e., in the namespace).
2. Files were marked for replication based on pool attributes. Not only have these now been deprecated, but the approach is conceptually wrong, and interferes with the ability to write files intended to go to tertiary storage on the same pool.
3. The replication requirements it responded to were themselves too coarse-grained.

The New Resilience Service: Features

1. **No database.** In-flight or queued file operations are still recoverable if the service itself goes offline and is then restarted.
2. **No special pool configuration.** The same pool may host both 'resilient' files and files which go to tape.
3. **Replication** is defined on the basis of a file's class (**storage unit**) which specifies the number and distribution of copies.
4. **Broken/corrupt replicas** are handled by removal and recopying when possible.
5. **An alarm is raised** when there is a fatal replication error.
6. **A rich set** of diagnostic and statistics **commands** is available through the admin shell.

Basic Design



State is maintained in two map structures and operations executed concurrently by two dedicated handlers. Lost state is recovered from a checkpoint file and from periodic scans of all resilient file locations.

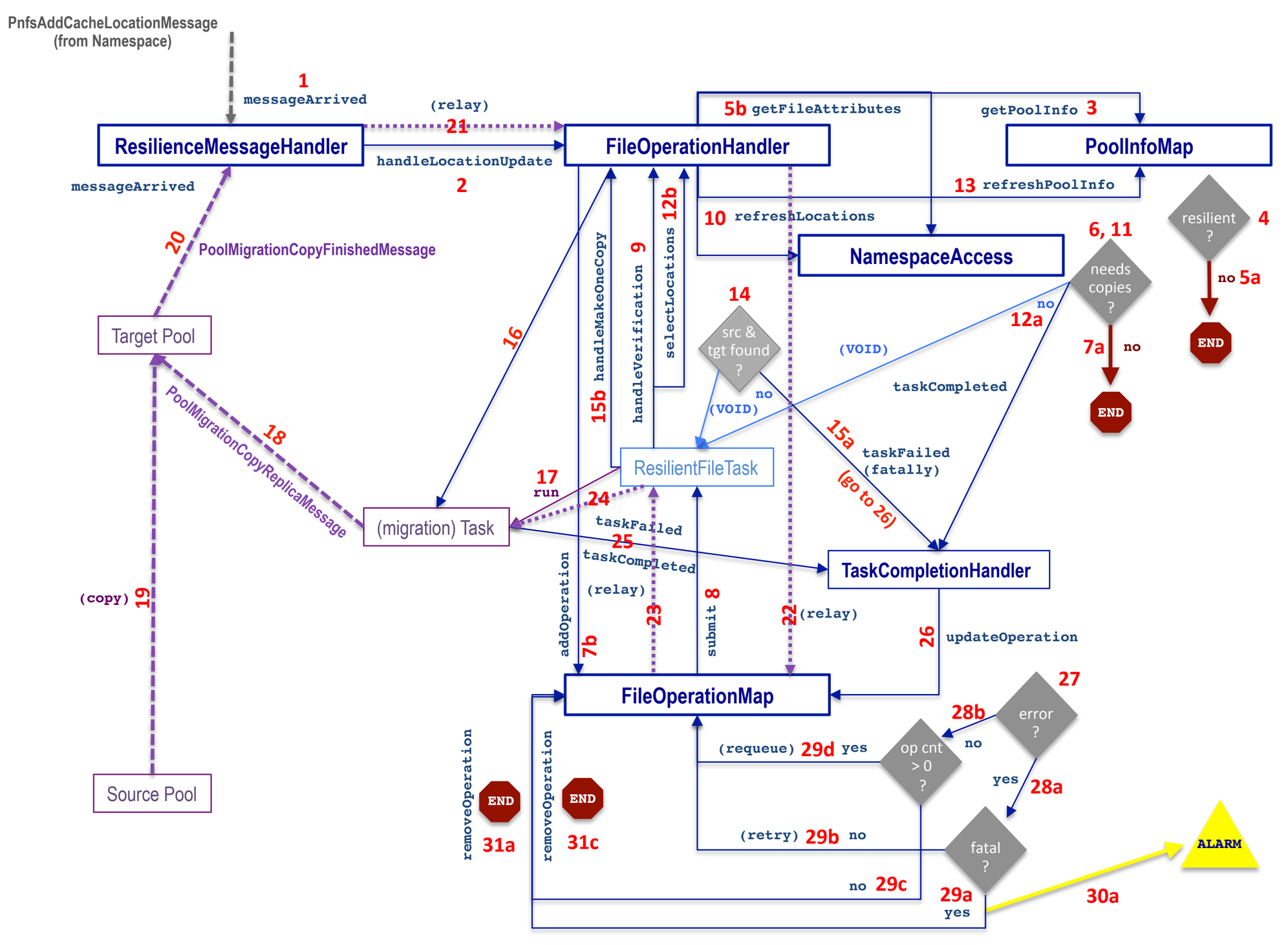
References

- www.dcache.org
- https://github.com/dCache/dcache/wiki/Resilience

Implementation

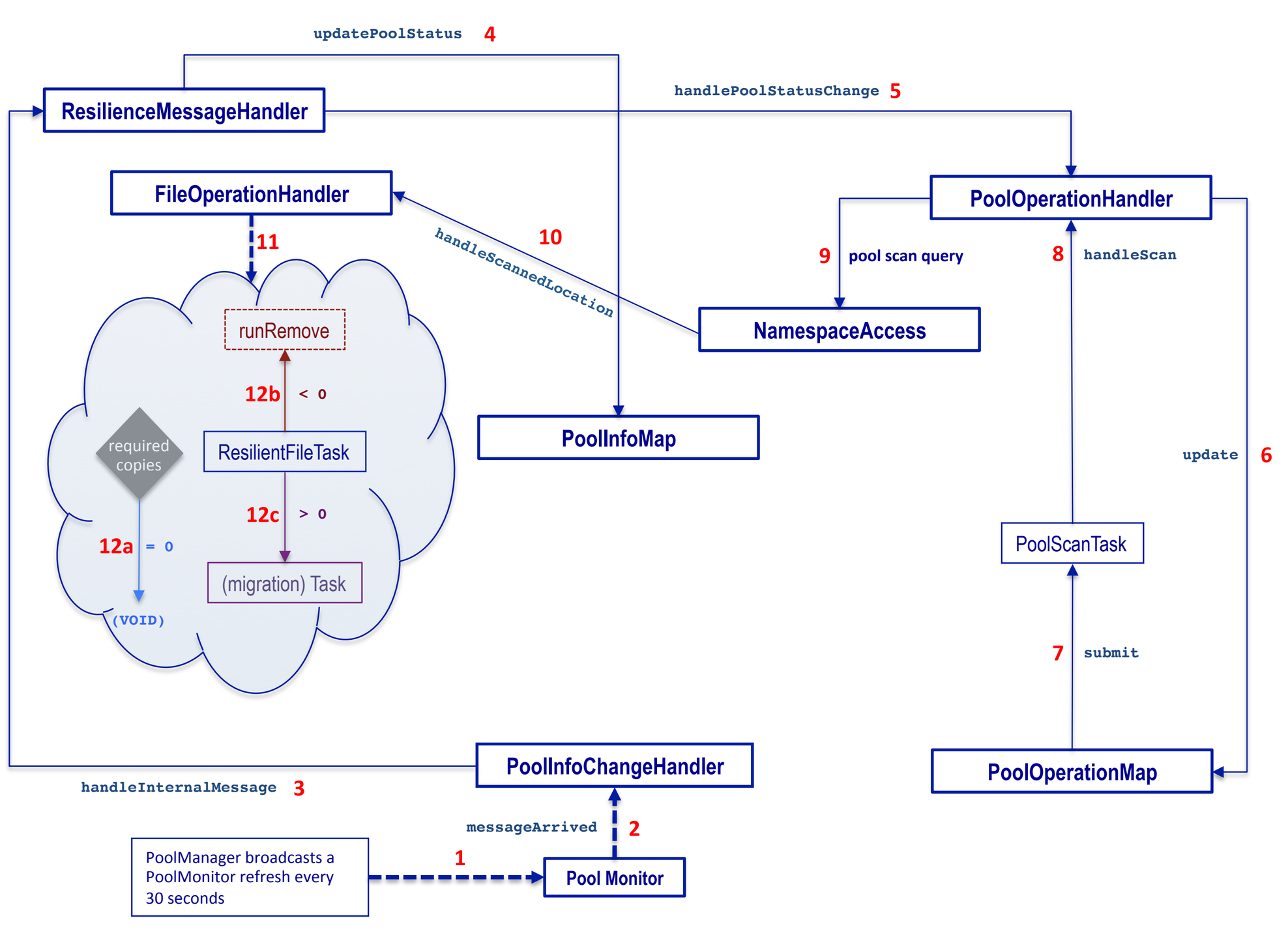
1. **Self-contained service** (dCache 'cell'); can run in its own domain or a shared domain; can be disabled and re-enabled without stopping and restarting the domain.
2. **More fully integrated** with the rest of dCache: receives PoolManager state updates periodically; accesses the namespace database directly; uses full-featured support of the pool's migration module to do pool-to-pool copying.
3. **Isolation**: does not affect the performance of PnfsManager or PoolManager (crucial core dCache components).
4. **Scalability (a)**: can manage millions of operations in memory without internal performance degradation.
5. **Scalability (b)**: can keep pace with PnfsManager's processing of new file locations (peak measured performance 1 KHZ).
6. **Fairness**: observes a policy of preferring the availability of at least two copies; operations on files with currently only one replica are promoted to the head of the waiting list.
7. **Anti-starvation**: ensures progress on both scanned file locations and new file locations when pool scans are taking place.

Handling New Files



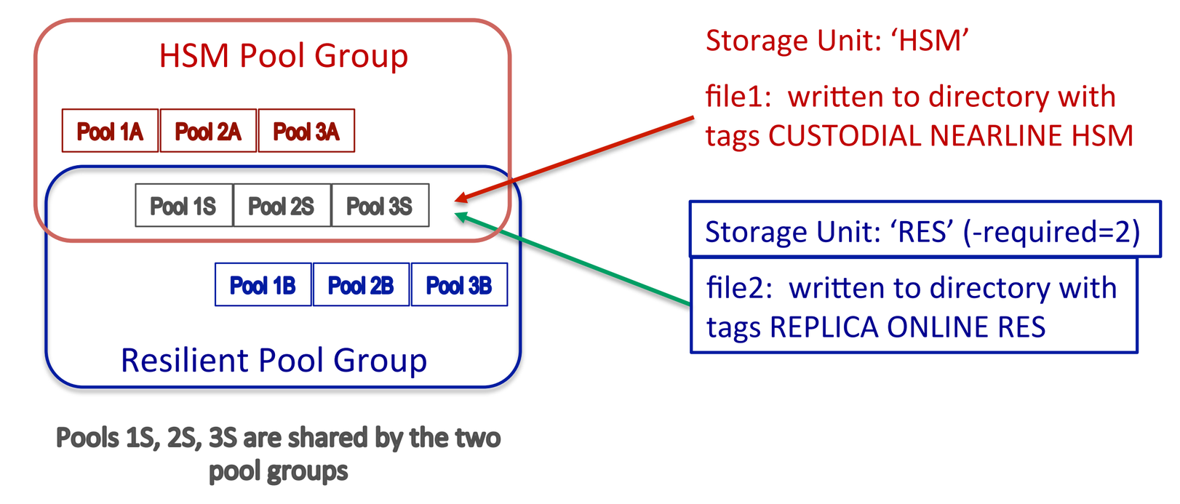
When a file is added to the namespace, the Resilience service is notified. It checks to see if the file should be replicated, and if so, schedules it for copying.

Handling Pool Status Changes



When pools go offline or come back online, the change is communicated to the Resilience service, which schedules a scan of the pool's files to see if more copies are needed or if copies can be removed.

Pool Sharing



Pools can belong to a resilient pool group and to other non-resilient pool group(s). The shared pools can contain both disk-only replicas and cached copies of files stored on tape, because the Resilience service operates on the basis of storage units and directory tags. This allows for more economical use of disk space.

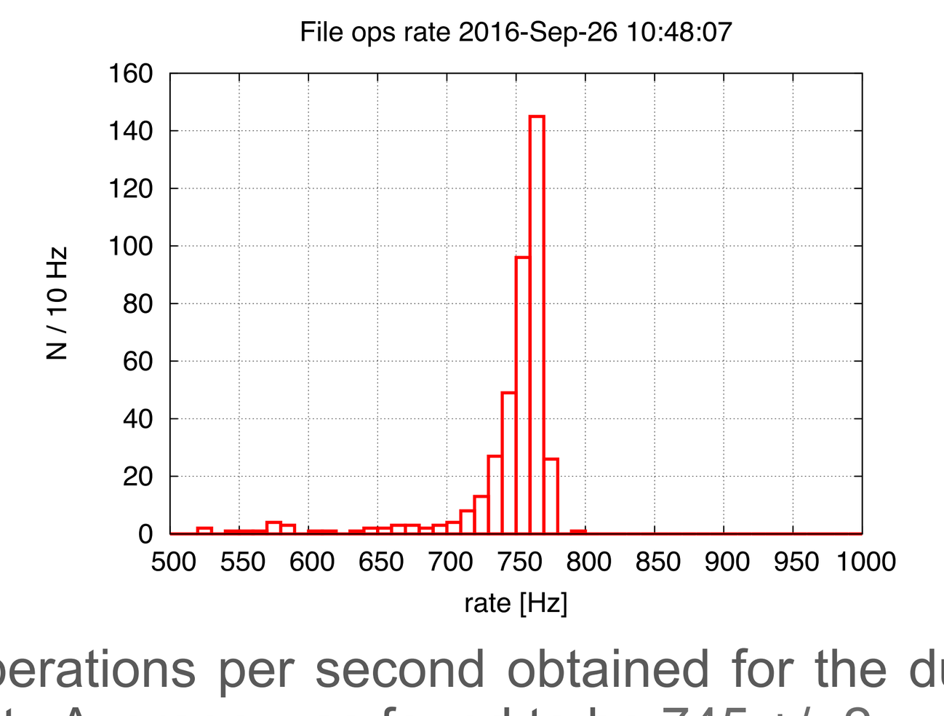
Diagnostics

```
[fndcatemp2] (Resilience@resilienceDomain) admin > diag .*
Running since: Fri Jun 03 14:47:40 CDT 2016
Uptime 3 days, 19 hours, 56 minutes, 36 seconds

Last pnfs sweep at Tue Jun 07 10:44:17 CDT 2016
Last pnfs sweep took 0 seconds
Last checkpoint at Tue Jun 07 10:43:24 CDT 2016
Last checkpoint took 0 seconds
Last checkpoint saved 86 records

MESSAGE                          received  msgs/sec
    CLEAR_CACHE_LOCATION               19         0
    CORRUPT_FILE                        5         0
    ADD_CACHE_LOCATION           67898905       515
    POOL_STATUS_DOWN                    0         0
    POOL_STATUS_UP                    169         0

OPERATION                       completed   ops/sec      failed
    FILE                         67901908       517          23
    POOL_SCAN_DOWN                      0         0           0
    POOL_SCAN_ACTIVE                  228         0           0

TRANSFERS BY POOL                    from          to      failed       size        removed      failed     size
dmsdca15-1                         959166      962097           0  939.55 KB              0           0        0
dmsdca15-1.1                       972972      973463           0  950.65 KB              0           0        0
dmsdca15-2                         967710      970284           1  947.54 KB              0           0        0
dmsdca15-2.1                       968995      972802           0  950.00 KB              0           0        0
dmsdca15-3                         956909      963618           1  941.03 KB              0           0        0
dmsdca15-3.1                       970977      973876           0  951.05 KB              0           0        0
dmsdca15-4                         959851      959739           0  937.25 KB              0           0        0
dmsdca15-4.1                       926010      926611           0  904.89 KB              0           0        0
...
TOTALS                           67901908    67901908          22   64.76 MB              0           0        0
```

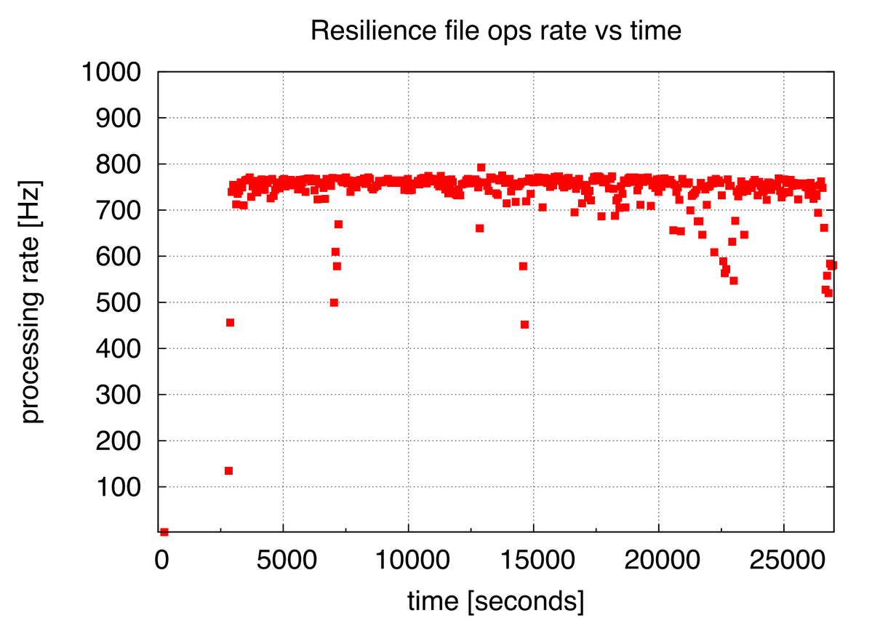
This is one example of the diagnostic output from commands which allow the administrator to check system state. Such commands also allow for the inspection of file and pool operations currently queued or running, and for the gathering of timing statistics, among other things.

Performance

Tests used a constant load of 105 concurrent clients continually writing 1 byte files, with 75% of the files given 2 copies and 25% given 3. Performance and stability are shown in terms of the rate of file operations sustained over time, and the ratio of copy operations to incoming new file locations over time.



File operations per second obtained for the duration of the test. Average was found to be 745 +/- 2.



The plot above shows the stability of file operations per second over time.

Ratio of file operations to new locations over time. The deviation from unity indicates a small amount of queuing.

Acknowledgements

We would like to thank **FermiCloud Support** for providing client nodes for performance testing.

Fermilab is Operated by the Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy