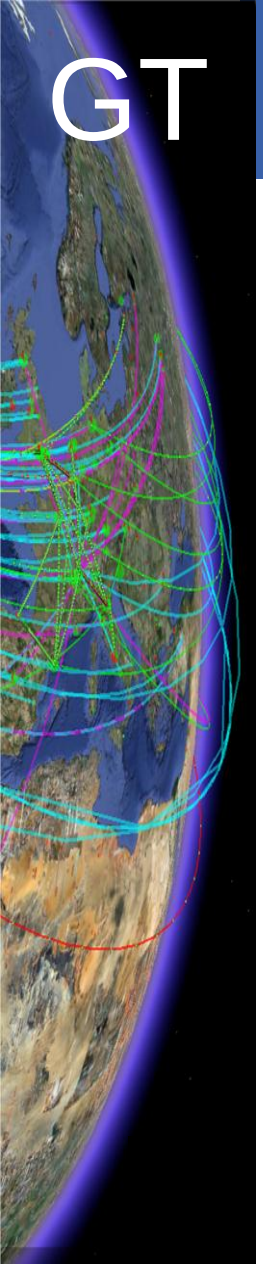


Overview of DMLite

Ricardo Rocha
(on behalf of the LCGDM team)

CERN IT Department
CH-1211 Geneva 23
Switzerland
www.cern.ch/it

EUROPEAN MIDDLEWARE INITIATIVE
EMI INFSO-RI-261611

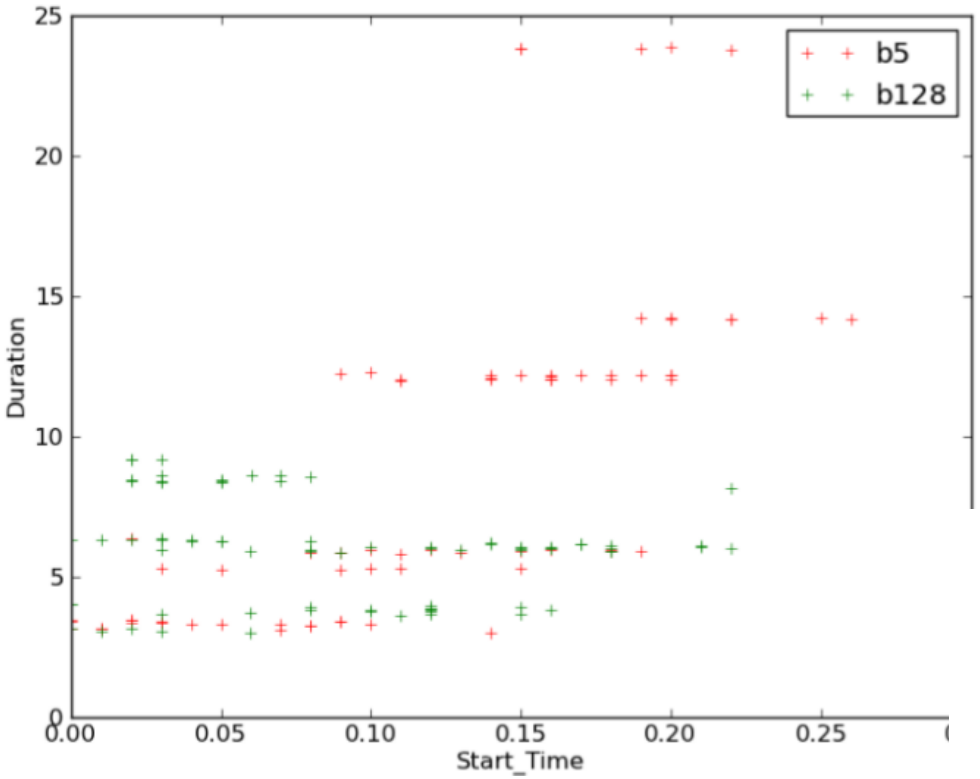
Reasoning for DMLite

- ~1.5 years ago we performed a full DPM evaluation
  - Using PerfSuite, out testing framework
  - https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Admin/Performance
  - (most results presented in the workshop come from this framework too)
- It showed the system had significant bottlenecks
  - Performance
  - Code maintenance (and complexity)
  - Extensibility

Dependency on NS/DPM daemons

- All calls to the system had to go via the daemons
  - Not only user / client calls
  - Also the case for our frontends (HTTP/DAV, NFS, XROOT, …)
  - Daemons were a bottleneck, and did not scale well
- Short term fix (available since 1.8.2)
  - Improve TCP listening queue settings to prevent timeouts
  - Increase number of threads in the daemon pools
    - Previously statically defined to a rather low value
- Medium term (available since 1.8.4, with DMLite)
  - Refactor the daemon code into a library

Dependency on NS/DPM daemons



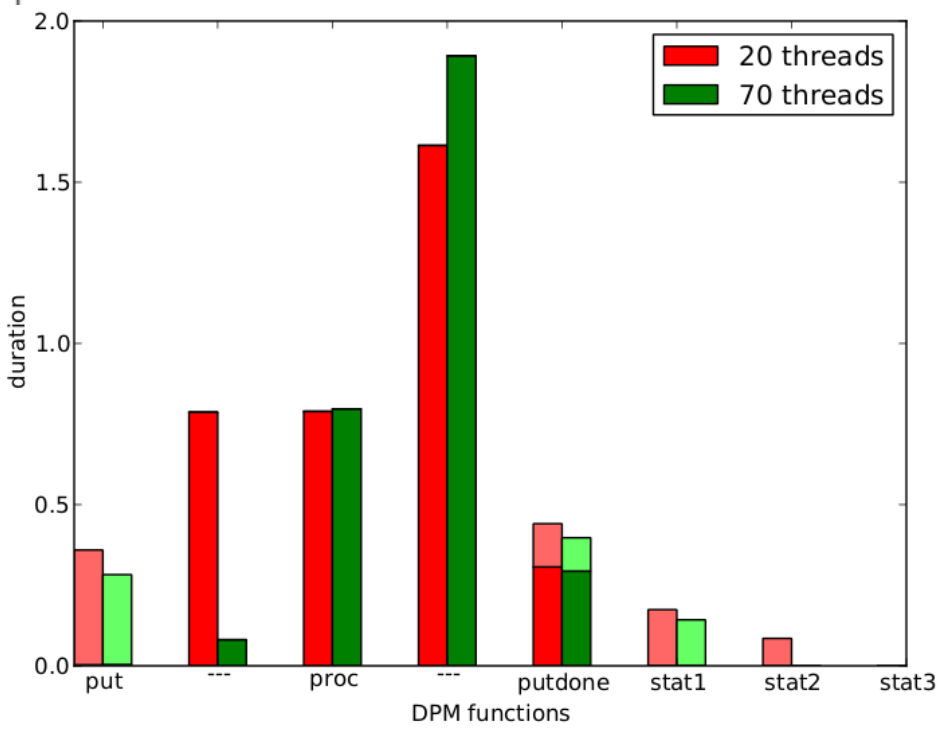
go via the daemons

ds (HTTP/DAV, NFS, XROOT, …)

nd did not scale well

- Previously statically defined

- Medium term (available sin
  - Refactor the daemon code ir

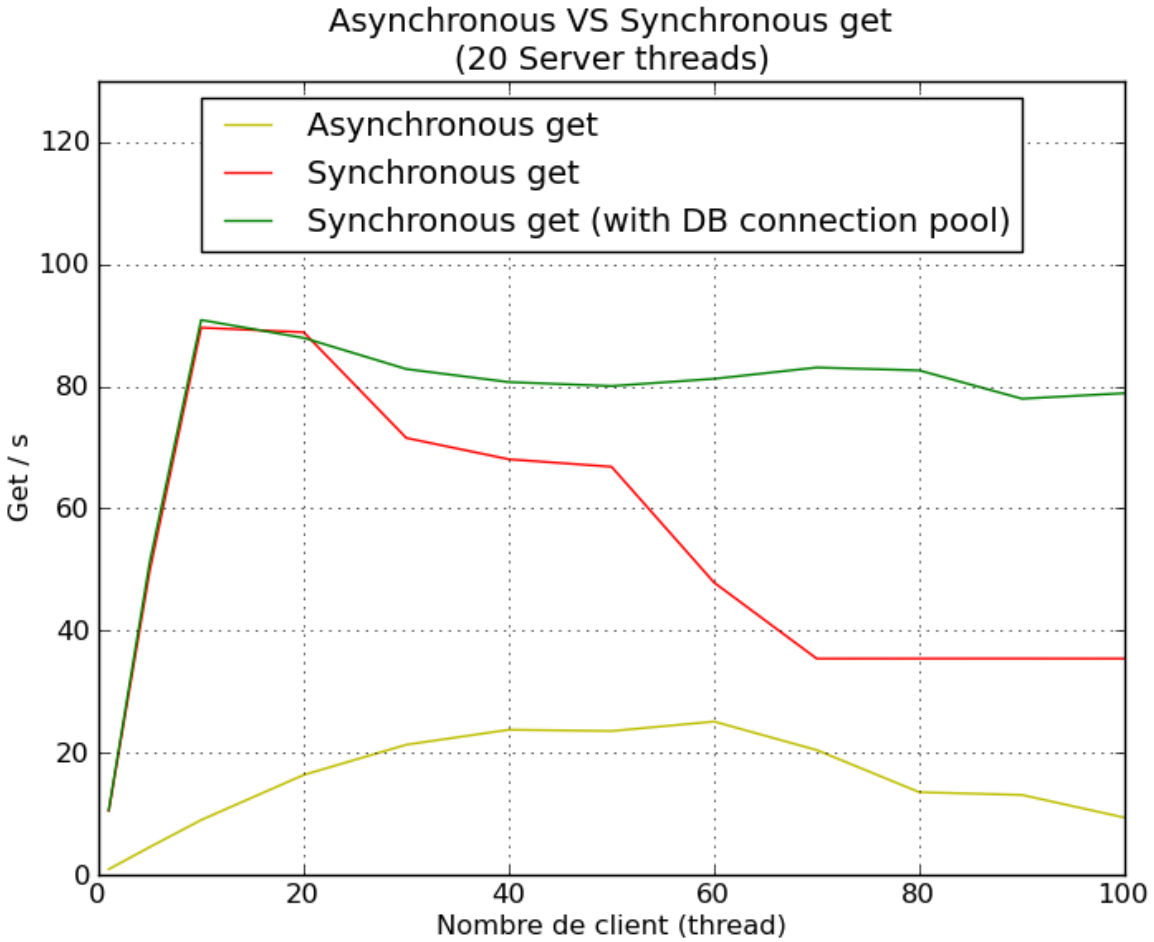


GET asynchronous performance

- DPM used to mandate asynchronous GET calls
  - Introduces significant client latency
  - Useful when some preparation of the replica is needed
  - But this wasn't really our case (disk only)
- Fix (available with 1.8.3)
  - Allow synchronous GET requests
- DMLite has the same sync behavior (but faster ☺)

GET asynchronous performance

- DPM used to mandate asynchronous GET calls
  - Intr
  - Use
  - But
- Fix
  - Allc

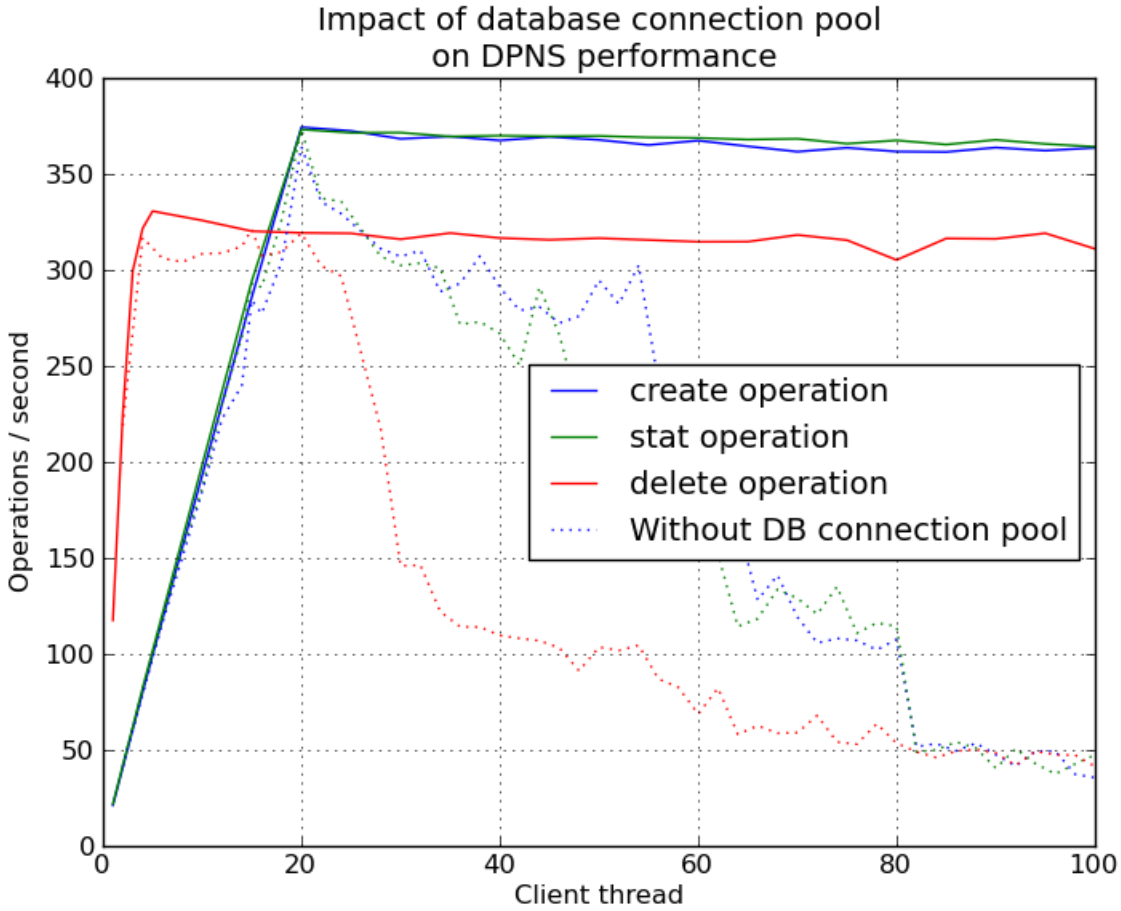


Database Access

- No DB connection pooling, no bind variables ☹
  - DB connections were linked to daemon pool threads
  - DB connections would be kept for the whole life of the client
- Quicker fix (available with 1.8.6)
  - Add DB connection pooling to the old daemons
  - Good numbers, but needed extensive testing… took some time
- Medium term fix (available since 1.8.4 for HTTP/DAV)
  - DMLite, which includes connection pooling
  - Among many other things…

Database Access

- No DB connection pooling, no bind variables ☹
  - DB c
  - DB c e client
- Quicke
  - Add
  - Goo
- Mediu DAV)
  - DML
  - Amc



Dependency on the SRM

- SRM imposes significant latency for data access
  - It has its use cases, but is a killer for regular file access
  - For data access, only required for protocols not supporting redirection (file name to replica translation)

- Fix (all available from 1.8.4)
  - Keep SRM for space management only (usage, reports, …)
  - Add support for protocols natively supporting redirection
    - HTTP/DAV, NFS 4.1/pNFS, XROOT
    - And promote them widely…
    - Investigating GridFTP redirection support (seems possible!)

Future Proof with DMLite

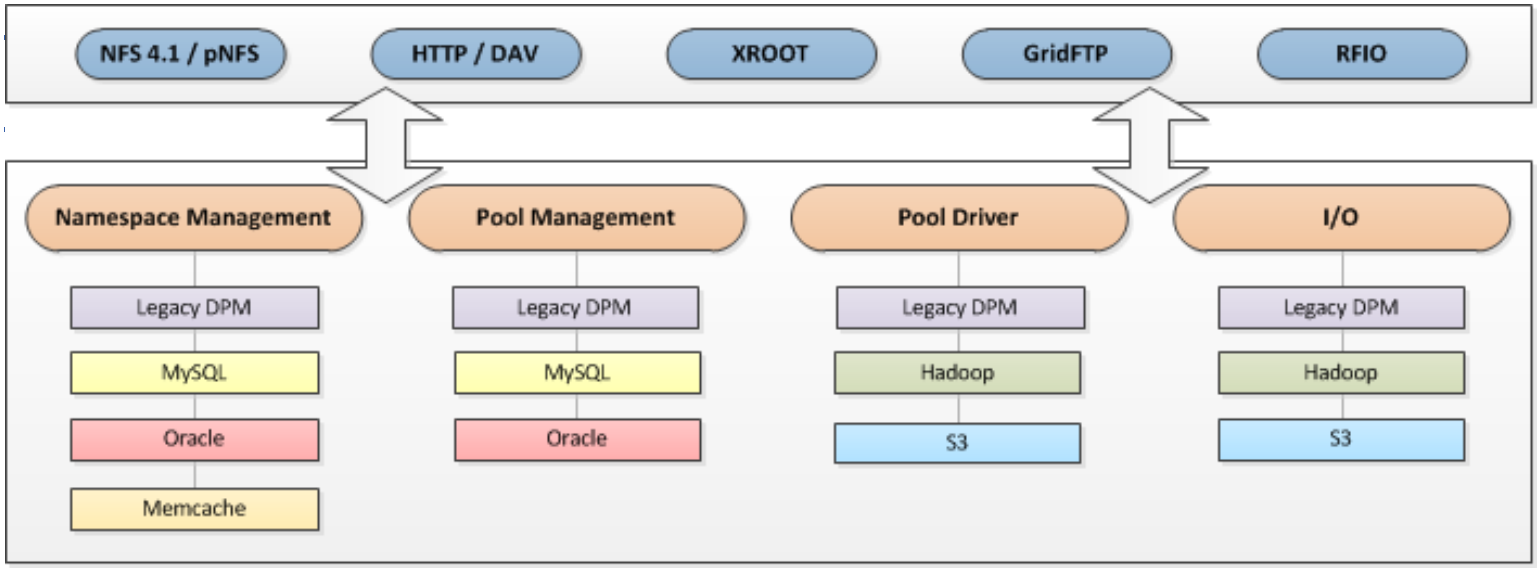
Future Proof with DMLite

- DMLite is our new **plugin based** library
- Meets goals resulting from the system evaluation
  - Refactoring of the existing code
  - Single **library** used by all frontends
  - Extensible, open to external contributions
  - Easy integration of standard building blocks
    - Apache2, HDFS, S3, …

https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Dev/Dmlite

Future Proof with DMLite

- DMLite is our new **plugin based** library
- Meets goals resulting from the system evaluation
  - Refactoring of the existing code
  - Single **library** used by all frontends



https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Dev/Dmlite

Summary and Status

- DMLite is a single library used by all DPM components
- In production today
  - Already used by HTTP/DAV, soon by all frontends
- We've opened DPM to other systems
  - Many widely used in the industry (HDFS, S3, …)
  - And the work has just started
- Clean, well defined interfaces
  - And APIs in different languages, much easier to contribute
- Performance improved drastically!

- Plugin details come next…