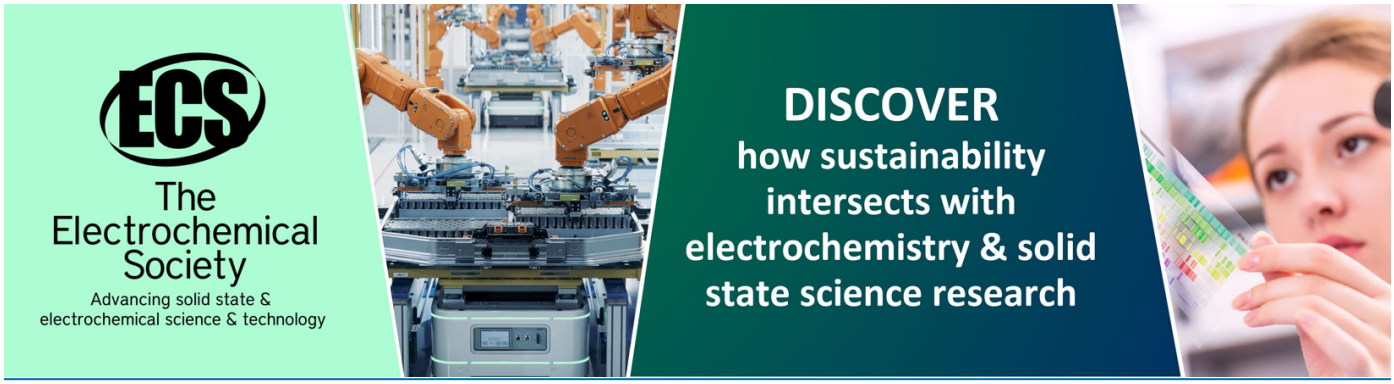
Direct data access protocols benchmarking on DPM



Journal of Physics: Conference Series **664** (2015) 042018 doi:10.1088/1742-6596/664/4/042018

Direct data access protocols benchmarking on DPM

Fabrizio Furano, Adrien Devresse, Oliver Keeble, Valentina Mancinelli

CERN, European Organization for Nuclear Research

E-mail: fabrizio.furano@cern.ch, adrien.devresse@cern.ch, oliver.keeble@cern.ch, valentina.mancinelli@cern.ch

Abstract.

The Disk Pool Manager is an example of a multi-protocol, multi-VO system for data access on the Grid that went though a considerable technical evolution in the last years. Among other features, its architecture offers the opportunity of testing its different data access frontends under exactly the same conditions, including hardware and backend software. This characteristic inspired the idea of collecting monitoring information from various testbeds in order to benchmark the behaviour of the HTTP and Xrootd protocols for the use case of data analysis, batch or interactive. A source of information is the set of continuous tests that are run towards the worldwide endpoints belonging to the DPM Collaboration, which accumulated relevant statistics in its first year of activity. On top of that, the DPM releases are based on multiple levels of automated testing that include performance benchmarks of various kinds, executed regularly every day. At the same time, the recent releases of DPM can report monitoring information about any data access protocol to the same monitoring infrastructure that is used to monitor the Xrootd deployments. Our goal is to evaluate under which circumstances the HTTP-based protocols can be good enough for batch or interactive data access. In this contribution we show and discuss the results that our test systems have collected under the circumstances that include ROOT analyses using TTreeCache and stress tests on the metadata performance.

1. Introduction

Comparing practices and data access protocols is generally a very difficult task that typically involves setting up separated, equivalent server deployments and configuring them with the same content used by a set of benchmarks. Invariably, different implementations of frameworks will have different characteristics (e.g. the latency to open a file in a large system) that make the evaluation of pure data access more challenging.

In the case of the Disk Pool Manager, one can deploy a single system that can serve multiple protocols, being sure that the implementation details of the backends (e.g. involving database queries) will be equivalent. This possibility reduces the margin of uncertainty for performance measurements of different frontend protocol implementations.

This work tries to better understand under which circumstances the HTTP-based protocols can be good enough for batch or interactive data access. The reference for this tasks is the Xrootd system, which DPM can use as one of its frontends, together with the Apache httpd.

From the functional point of view, HTTP (or WebDAV [12]) and Xrootd [2] share several aspects, first of all the fact that they both can be used as posix-like data access protocols whose preferred usage option is through a client component that is embedded in the application. This

classic architecture is not limited to High Energy Physics applications, and is the architecture used also in Web browsers and in several mainstream mediaplayer softwares [8] [5] [1].

The aspects peculiar to High Energy Physics are mainly related to performance and scalability and the techniques that are used to achieve this. We can summarize the main use cases as:

- High rates of metadata requests, including remote file opens
- Full file data transfers, with broad variations in file sizes (from kBs to a few GBs)
- Massive usage of *vectored read requests* (implemented in HTTP through multi-range requests and multi-part responses) for the data analysis use cases

1.1. DPM and DMLite

One of our most important contributions to the usage of HTTP in High Energy Physics has been the evolution of the Disk Pool Manager (DPM) architecture towards our scalable and flexible framework for designing data management systems, called *dmlite* [4].

The Disk Pool Manager (DPM) is a lightweight solution for grid enabled disk storage management. Operated at around 200 sites, it has the widest distribution of all grid storage solutions in the WLCG infrastructure. It provides an easy way to manage and configure disk pools, and exposes multiple interfaces for data access (xrootd, GridFTP and HTTP/WebDAV) and control (SRM).

Some of the most important supported features are:

- Provide HTTP multi-stream transfers for high performance WAN access, matching it with the strict requirement of DPM about coordination of accesses.
- Support for third party copies.
- Support for X.509 authentication with proxy certificates and VOMS extensions
- Support for user credential delegations.
- Support for modern configuration and monitoring solutions based on the industry standards Puppet and Nagios.

2. Contributing DAVIX: a full-featured client

Davix is a toolkit that we created for High Performance Remote I/O with HTTP-based protocols in a High Performance Computing environment. It is currently one of our contributions to the EPEL [9] software distribution.

In the context of the tests under discussion in this work, the importance of DAVIX is linked to it being a coherent, high quality implementation of generic data/metadata access primitives on top of HTTP/WebDAV. This makes it straightforward to make a sophisticated test program work for different protocols, making also comparisons simpler. The need for such a component was triggered by the very uneven support for advanced features in the mainstream HTTP clients. To give an example, X.509 authentication, which is fundamental for Grid usage, is supported only partially and by one or two clients, which do not support other fundamental features, like the vectored access.

Davix provides a set of simple command line tools and a high level POSIX-like API for fast optimized data access, abstracting the details of the HTTP protocol and of the interactions with the servers. Davix has been built as a layer on top of the official WebDAV library, called libneon [11].

Davix supports WebDAV and the S3 protocol for meta-data operations and file manipulation. It also supports supports advanced features like:

- Vectored read operations, that raise the I/O performance for the applications that can use them (e.g. ROOT I/O).

- X509 and VOMS authentication, compatible with the Grid standards.
- Authentication session pools, to reuse previously established authentication sessions.
- Redirection on all the operations and redirection caching.

3. Some benchmarks

The DPM development and validation infrastructure regularly executes several kinds of tests, which can be used to understand the usability of HTTP for accessing data in a local area network and towards data hosted at a distant site.

3.1. Nightly local stress tests

The nightly local stress tests are run twice per day from a virtual machine used as a client, towards the so-called DPM trunk testbed, which runs on a 4-core server an always up to date setup of the trunk version of the DPM components.
This test focuses on the peak metadata performance by calling stat() on a fixed set of 100K files. The main parameter of this test is the size of the pool of threads that cooperate in the client to stat() all those files. In order to reduce errors, the test is repeated 5 times and the results averaged. The error bars shown in the graphs correspond to a 95% confidence interval.
In the case of HTTP, the stat() function is implemented through a WebDAV PROPFIND request, while for Xrootd its kXR_stat request is used.
This simple test is one of the most used tools that supported the recent big DPM performance improvements. The DPM development servers keep all the history of these results of the previous months, to help developers to avoid performance regressions.

The graph shown in Figure 1 shows how many stat() requests per second can be fulfilled by the test DPM through the Apache httpd frontend. The graph shown in Figure 2 shows how many stat() requests per second can be fulfilled by the same test DPM through the Xrootd frontend.

Our interpretation of the results is that they are comparable. The Xrootd case gives results that are almost constant with respect to the number of client threads. It seems faster for a smaller number of client threads, while the WebDAV case seems faster roughly starting with 50 clients, with a peak around 100 clients and a slight degradation when this number exceeds 250. These small differences are very difficult to explain, and could also be related to the efficiency and compatibility of the client side implementation with this kind of test, which remains a stress test and is only representative of the peak performance for a particular request.

3.2. Regular Wan HC tests

Another kind of test that is being continuously run in the context of the DPM testing consists in a simple yet realistic data analysis performed through Wide Area Network between different locations.

The little analysis being run is a ROOT [10] analysis on an ATLAS data file, whose pattern we know and is easily reproduceable. This analysis correctly triggers the sophisticated informed prefetching mechanisms of the ROOT framework, thus being able to cancel the high network latencies involved [5] [1].

The applied methodology is:

- Test jobs are continuously submitted at a slow pace towards 5 sites: CERN, RHUL (London), SHEF (UK), Australia (Melbourne), ASGC (Taipei), BNL (USA)

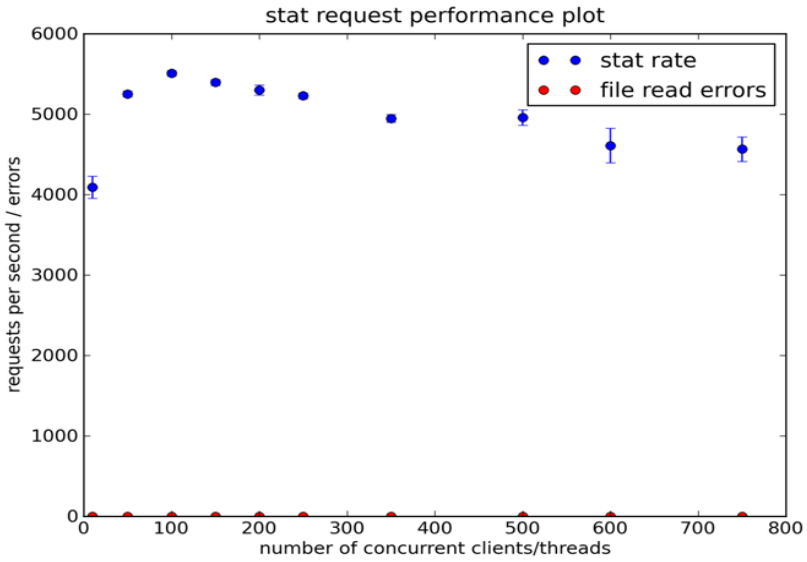


Figure 1. Performance obtained invoking Stat() over 100k files with an increasing number of threads towards DPM using WebDAV

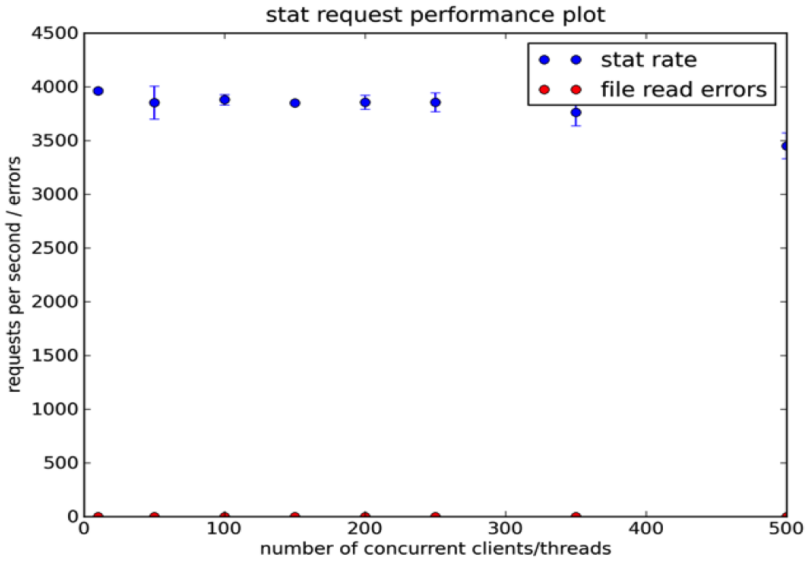


Figure 2. Performance obtained invoking Stat() over 100k files with an increasing number of threads towards DPM using Xrootd

- To make the tests more relevant, up to 50 of these jobs have been concurrently submitted in each site as a manual operation
- Each job runs the same analysis many times per day towards various remote storage services
- Each jobs monitors the time that it dedicated to input/output of the data to be analyzed in the various cases
- At the end of each analysis run, a measure of the accumulated IO time is sent to a central service for storing into a database and into an instance of Elastic Search [13] at CERN

- The content of these databases are then used to produce various types of graphs showing averages and errors. The error bars shown in the graphs correspond to a 95% confidence interval.

We would like to emphasize that the graphs shown in Figure 3 and 4 show IO times that account just for a percentage of the total duration of the data analysis they refer to. We did the choice of measuring only the IO network time because the various jobs are run in different infrastructures hosting many different kinds of worker nodes. The goal was to measure the time spent reading data, and not the performance differences of various kinds of computers.

Figure 3 shows the IOtime for a job run in various sites reading data from a DPM storage at CERN, using HTTP and Xrootd. We can see that the differences are small, with all the error bars intersecting. Our conclusion is that the performance is comparable even in the worst cases.

Figure 4 shows the IOtime for a job running at CERN reading data from various sites running DPM. Surprisingly enough, the small differences among the performance using HTTP or Xrootd in this case have opposite sign with respect to the previous Figure 3. In this case it's the xrootd data access that is very slightly faster. At the best our knowledge and understanding we don't have a good explanation for the slight differences that we see, sometimes in favor of Xrootd, sometimes of HTTP. In the case these differences represent a more relevant percentage of the total execution time, we believe that further investigations are necessary to understand them.

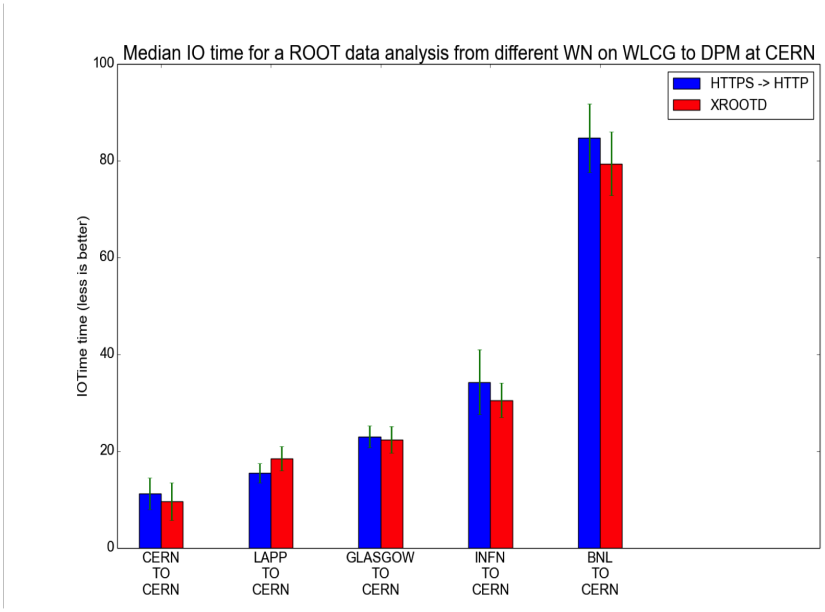


Figure 3. Pure IO time for a simple analysis job run at various locations towards a DPM server at CERN

4. Conclusions

At the best that we can do with our test cases, we do not see dramatic differences in the IO time or in the frontend access speed. At the same time, although quite intense, the test did not reach a server side 100% CPU utilization. It would be useful to see at which level of performance the CPU utilization in the server will be the limiting factor.

Our understanding is that the conditions that we created for the test in the Grid production system are likely not enough to reach these extreme conditions, and there is the need for more specialised tests to look for practical differences in a production system when the load approaches

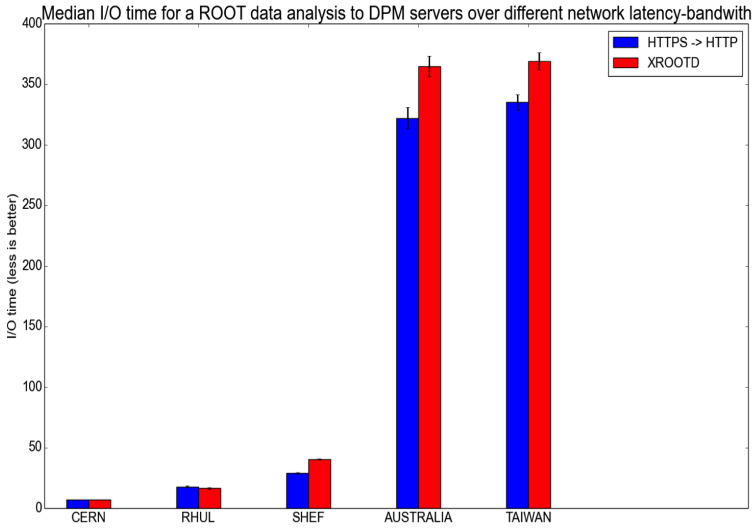


Figure 4. Pure IO time for a simple analysis job run at CERN towards various DPM servers

the saturation of the available storage system performance.

This is difficult to match with what comes out from an analysis of how the two data protocols work; one could argue that there should be some difference in favor of a highly polished binary protocol like Xrootd versus a text-based one with variable-sized headers like HTTP.

Our result is then that under the conditions that we have been able to create in a production system, the performance is comparable. This also leaves space for future work trying to measure differences in the CPU consumption of the daemons that implement the protocols, which we have not taken into account in this paper.

References

[1] Scalla/xrootd WAN globalization tools: Where we are Fabrizio Furano and Andrew Hanushevsky 2010 J. Phys.: Conf. Ser. 219 072005 http://iopscience.iop.org/1742-6596/219/7/072005/
[2] The xrootd.org homepage http://www.xrootd.org
[3] DPM: Future Proof Storage Alejandro Alvarez, Alexandre Beche, Fabrizio Furano, Martin Hellmich, Oliver Keeble, Ricardo Rocha CHEP2012
[4] Web enabled data management with DPM & LFC Alejandro Alvarez Ayllon, Alexandre Beche, Fabrizio Furano, Martin Hellmich, Oliver Keeble and Ricardo Brito Da Rocha CHEP2012
[5] Furano F. Data Management in HEP: an approach. The European Physical Journal Plus Volume 126, Number 1 (2011), 12, DOI: 10.1140/epjp/i2011-11012-2
[6] DPM components https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Dev/Components
[7] Cristian Traian Cirstea Grid Data Access: Proxy Caches and User Views Eindhoven University of Technology Stan Ackermans Institute / Software Technology ISBN 978-90-444-1067-9
[8] VideoLAN VLC media player http://www.videolan.org/vlc/
[9] EPEL - Fedoraproject http://fedoraproject.org/wiki/EPEL
[10] ROOT: An Object-Oriented Data Analysis Framework http://root.cern.ch/
[11] neon: an HTTP and WebDAV client library, with a C interface http://www.webdav.org/neon/
[12] HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV), section 8.8 http://tools.ietf.org/html/rfc2518#section-8.8
[13] Elasticsearch: RESTful, Distributed Search and Analytics https://www.elastic.co/products/elasticsearch