Evaluation of Multi-Source Downloads for FTS

Alin Grigorean[1,2], Mihai Carabas[1], Nicolae Tapus[1], Mihnea Dulea[2], Maria Arsuaga-Rios[3]
[1]University POLITEHNICA of Bucharest
[2]Department of Computational Physics and Information Technologies (DFCTI),
'Horia Hulubei' National Institute for R&D in Physics and Nuclear Engineering (IFIN-HH)
Bucharest, Romania
[3]Data & Storage Services Group, CERN, Switzerland

Abstract—**The data transfer in the Grid at CERN (the European Organization for Nuclear Research) has seen constant improvement, be it through optimizing the existing tools, GridFTP and GFAL2 or by adding new tools such as XRootd. Unfortunately, all these have reached the maximum limit in terms of throughput. They are limited not by the network infrastructure, but by the fact that they use a single source for transfers, despite the existence of multiple replicas. In this paper we take on the challenge of evaluating the effects of using multiple sources, over the throughput, by comparing the download speed of the tools mentioned above with Aria2 and an under development version of XRootd, both supporting the use of multiple sources.**

Keywords—grid; throughput; data transfer; downloads; multi-source; evaluation; GridFTP; Globus; Aria2; XRootd

I. Introduction

The File Transfer Service (FTS) is a core function of the CERN's Worldwide LHC Computing Grid (WLCG), whose efficiency is strongly impacted by its throughput optimization and the stability of connections. Therefore, choosing the right utility tool and protocol for transferring and downloading data is critical.

As of now, despite the existence of multiple replicas, all currently used data transfer tools are not taking advantage of them, limiting the transfers and downloads to a single source and, in consequence, limiting the throughput. To our best knowledge, no investigation has been published on the performance that could be gained from the use of multiple sources in the WLCG, which is why in this evaluation we present the improvement in speed that could be obtained through the use of multiple replicas as sources.

This paper comes in to evaluate the impact over throughput and stability given by the use of data transfers tools capable of using multiple sources. In order to do this we need a proper comparison between the utility tools currently used in the WLCG and a new utility tool with support for multiple sources, Aria2, a lightweight multi-protocol and multi-source download utility. Due to Aria2 being just a download utility and not being able to perform transfers, this comparison will only be done for downloads.

Therefore, in the second section of this paper we discuss the work that has been put into the optimization and improvement of the tools used in the Grid by shortly reviewing some of the results published in this field. Then, in section 3, we present the tools needed for this evaluation, both for realizing the comparison and analyzing the data. The architecture of the experiment, including the environment and the modules used for creating the comparison, is presented in section 4, while in section 5 we describe how those modules were implemented. Afterwards, in section 6, we will discuss the results of the comparison and analyze the data in order to take the proper conclusions, in section 7.

II. Related Work

There has been done extensive research in the field of Grid Computing with the purpose of optimizing and improving the data transfers, Grids becoming some of the most important resources used in scientific research. As stated in [1], the challenges of Grid environments "revolve around data - managing its access, distribution, processing and storage", which enforces the fact that optimizing data transfers and downloads has a major impact over the Grid, providing the motivation behind this paper. Moreover, [1] also points out that, since the beginning of the Grid technology, GridFTP was one of the most used data transfer protocols.

Storing all the data gathered from the experiments represents a difficult challenge. Not only has all the data to be kept for further use but also it must be made available when needed as fast as possible. For this the Grid at CERN uses two Large Scale Storage Systems: CASTOR and EOS. Out of the two we are going to work with EOS, a "a disk-only storage solution mainly focused on analysis and fast data processing with a very low access latency" [2].

More about data management challenges can be found in [3], where the authors talk about the data management system of the ATLAS experiment. Being one of the 4 largest experiments at CERN, ATLAS had produced more than 8 PB of data at the time when [3] had been written and, as described in it, that much data is extremely difficult to manage. In order to facilitate the management of the data, a series of services and feature have been added to Don Quijote 2, ATLAS Distributed Data Management system. Among those services, we found mentioned the Tracer, which is responsible with logging every access to data in a database for later use, when needed for deciding whether the data should be replicated or not. Due to this service together with the Database Replication Service, the use of multiple sources is possible.

Another key aspect of the Grid is represented by virtualization [1]. Grids are divided into VOs (Virtual

This work was partly funded by the Ministry of Research and Innovation under the contract no. 6/2016 (PNIII-5.2-CERN-RO).

Organizations) which have the purpose of separating and grouping the resources of the Grid in order for each VO to focus on a specific type of work. For this experiment we are going to use the VO DTEAM.

Due to the wide usage of GridFTP and the advancements of network technologies, a number of optimizations had to be done. One such example can be found in [4], where researchers have worked on creating a new transport driver for GridFTP capable of utilizing the full capacity of InfiniBand based networks.

Also, the GridFTP protocol has been investigated and compared with other protocols, such as FTP. In [5], the authors have conducted an evaluation similar to ours, but on a smaller scale, with just the two protocols. Their purpose was to decide on a protocol to be used for transferring files between Europe and Russia. For the tests they have chosen a file of 1GB and have performed a number of transfers with both GridFTP and FTP, concluding that the former can better saturate the bandwidth, resulting in much higher speed compared to FTP, while being considerably more stable.

Another way of optimizing data transfers has been the integration of new utility tools for data management.

One of those tools is GFAL2 (Grid File Access Library 2), a service of the gLite middleware, which handles data management and secure job execution [6]. Its main advantages are that it offers all the necessary commands for accessing and working with the Grid resources as with a simple file system. It also supports all major protocols used in the Grid: SRM, GSIFTP, HTTPS, DAVS and Xrootd [6], XRootd being added through the RGLite interface [7].

Another tool that is constantly developed and used in the Grid is Xrootd [8], "A highly scalable architecture for data access" [9]. It was created over rootd data server daemon with the purpose of managing the immense amount of data produced by the Stanford Linear Accelerator Center, being subsequently implemented in WLCG and becoming the only choice for the ALICE experiment storage solution [10]. It was also optimized by implementing a caching proxy in order to provide better "access to remote data, both in terms of latency and data reuse, as well as to facilitate more flexible data placement strategies among Tier 2 and Tier 3 centers" [11].

Unfortunately, as of now, both GFAL2 and XRootd lack the ability to use multiple sources for file transfers or downloads, thought, as expected in [7], the support for the use of multiple sources is currently being added to XRootd.

Although the Grid lacks an utility tool that can use multiple sources, it has the structure that would allow such a tool to be used, a lot of work being involved in replication [12,13]. By having multiple replicas we have multiple source available to be used in transfers or downloads at the same time.

III. Background on Grid MiddleWare

For this experiment we have used the protocols, utility tools and programming languages, described below.

Name	Description
Globus Toolkit	Open source utility used for building Grids and for data management in Grids
GFAL2	A C library that facilitates the use of the Grid storage as that of a file system
XRootd	A software created with the purpose of offering fast, low latency and scalable data access
Aria2	A lightweight multi-source download utility
HTTPS Protocol	An enhanced HTTP protocol, having an extra security layer, over TLS or SSL
GSIFTP Protocol	A subset of the GridFTP protocol, basically the FTP protocol with support for GSI security
XRootd Protocol	The protocol used by the utility tool with the same name, being developed with the same purpose as the utility
R	Programming language used for statistical analysis
Z Shell	Language used for shell scripting, similar to Bash

IV. Architecture of the Evaluation

Now we are going to talk about the architecture of the experiment. We are going to present how we have set up the environment for testing, and which have been the steps taken for the evaluation.

In order to make this experiment happen we first need the tools that are going to be used in the evaluation: Globus Toolkit, GFAL2, XRootd and Aria2. Additionally we need support for HTTPS and XRootd, due to Aria2 and XRootd working only with these protocols. Fortunately, the Grid at CERN has support for all needed protocols.

A. Environment

In order to be able to evaluate the performance gained from the use of multiple sources, we first have to select 3 endpoints out of 234 and, for that, we need to pass our endpoints through 2 modules, the HTTPS and Security Access Selection Module and the R/W Selection Module, which will eliminate all endpoints that we do not have access to, do not support HTTPS or we do not have both read and write rights for. Then, all we are left with are 39 endpoints, out of each we select 3. Afterwards, we initialize the selected endpoints, then run all the tests on them and pass all the results to the Shapiro-Wilk Test and Chart Module.

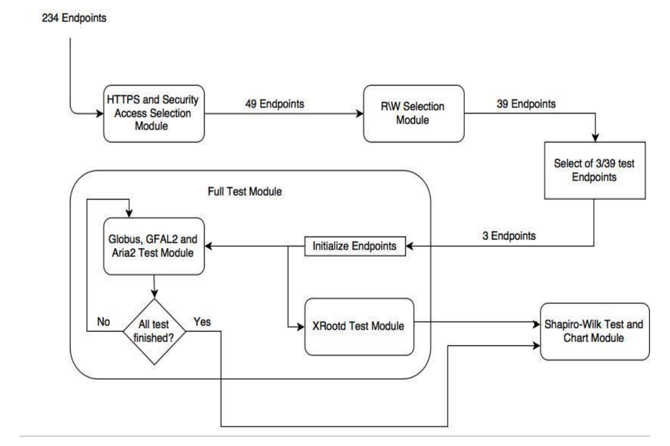


Fig. 1. Module Organisation

B. *Modules*

Now we are going to talk about the 4 modules that we need to pass through in order to be able to evaluate the performance of using multiple sources for downloads over the Grid.

1) HTTPS and Security Access Selection Module

Given that Aria2 is working only with HTTPS out of all the protocols that we test, we have to select only the endpoints that support this protocol, leaving us with a set of endpoints that are suitable for testing. Also in order to be able to work with an endpoint we need access to it.

Therefore this module is responsible with taking a number of endpoints as input and selecting from them only the ones that support HTTPS and do not have restrictions, outputting in a file the resulted endpoints.

2) R/W Selection Module

Once we have the results from the HTTPS and Security Access Module, we give them as input to this module, in order to select only the endpoints for which we have both read and write access.

After running the module, we end up with a number of endpoints that support HTTPS, can be accessed and we have read and write rights for them.

3) Full Test Module

Out of the remaining endpoints we need to select 3 and the criterion for selecting them is having support for GSIFTP and XRootd protocols too, having acceptable download speeds and being located in different geographical zones. As a result we have selected one in United Kingdom, one in Czech Republic and one in Italy.

Now, that we have selected 3 working endpoints that respect all that necessary requirements we have to initialize them with our test files. Firstly, we need files of sizes that are actually commonly used in the ATLAS experiment and that are manageable and keep the time needed to run the tests at an acceptable rate, thus we have chosen files of 1MB, 500MB, 1GB and 2GB.

Having everything prepared, we can now run the tests for generating the data needed for the analysis. The tests consist in taking each endpoint and then downloading every one of the 4 files 100 times. We are doing this for GFAL with GSIFTP as protocol, GFAL with HTTPS as protocol, GFAL with XRootd as protocol, Globus with GSIFTP as protocol, XRootd with XRootd as protocol and Aria2 with HTTPS as protocol. Also, in order for the tests to be as similar as possible they have been run during the weekend when the load on the endpoints should be at a minimum, so that the endpoints could give the best stability and performance.

4) Shapiro-Wilk Test and Chart Module

Once the tests finish and we have all the needed data, we have to check if the data is normally distributed in order to see if we can use the mean and standard deviation for analysis or we have to use the IRQ(Interquartile Range). Therefore, we run the Shapiro-Wilk test on the data. In order to do that, we need to import the data in the R script and run the test on it.

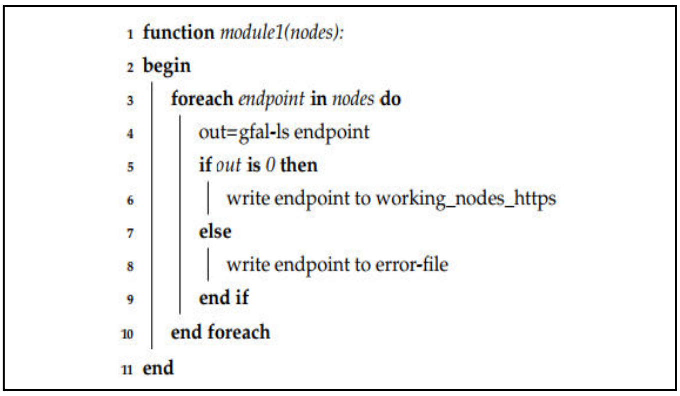
V. Evaluation

A. *Code Description*

The implementation of the code needed for this evaluation was done in 2 programming languages, Z Shell and R. The scripts responsible for preparing the environment and running the tests have been written in Z Shell and the scripts responsible for testing to see if the data follows a normal distribution and for generating the charts have been written in R. As can be learnt from the architecture, there are 4 modules that make this evaluation possible and now we are going to talk about them in more detail.

1) HTTPS and Security Access Selection Module

For filtering out the endpoints that do not support HTTPS we had to first initialize the endpoints with a file and the try to list that file using the HTTPS protocol. If an error is thrown we filter out that file, otherwise we select the node as good.



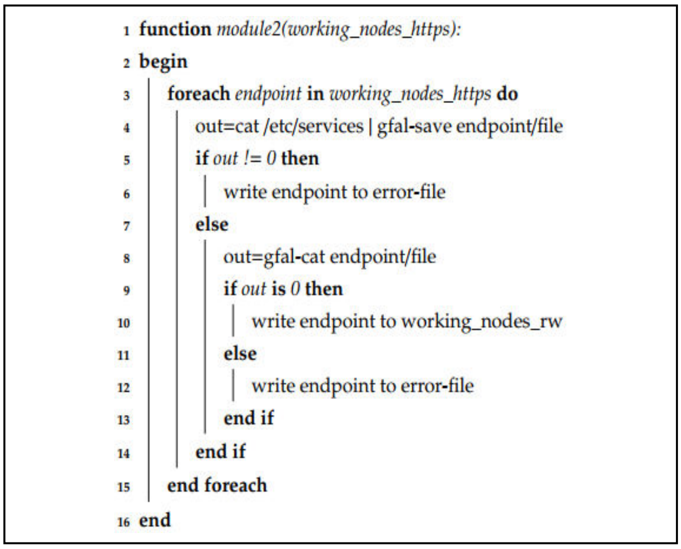
```
function module1(nodes):
begin
    foreach endpoint in nodes do
        out=gfal-ls endpoint
        if out is 0 then
            write endpoint to working_nodes_https
        else
            write endpoint to error-file
        end if
    end foreach
end
```

Fig. 2. HTTPS and SecurityAccess Selection Algorithm

2) R\W Selection Module

In order to obtain the endpoints that have both read and write access we have to write a file to each endpoint and check to see if the command generates an error. If an error is generated then the endpoint is filtered out, otherwise we test to see if we can read that file from the endpoint and if we can, no error being generated, the endpoint is selected. If not, the endpoint is filtered out.

This time an initialization of the endpoints is not necessary, because writing a file to the endpoints is exactly what the algorithm is doing in order to test the write access.



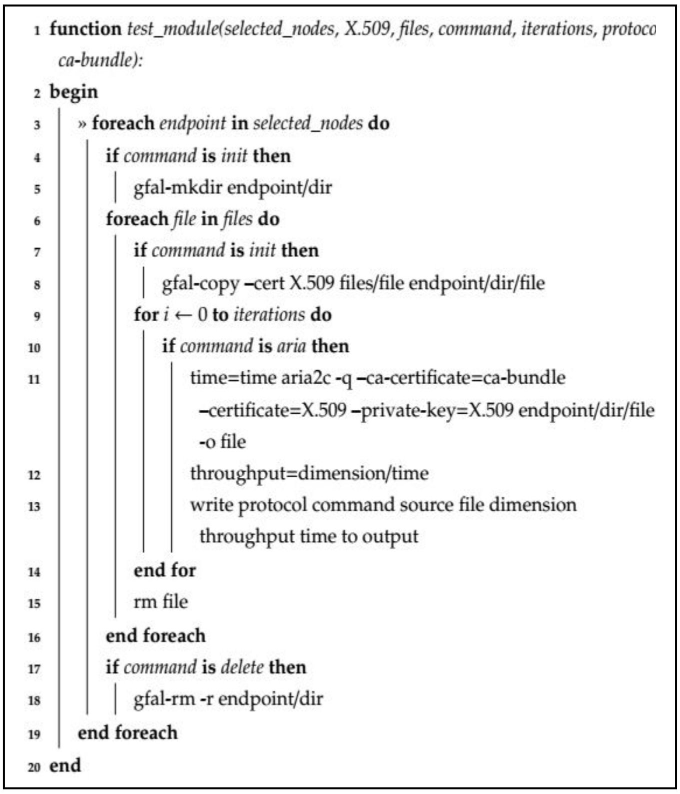
```
function module2(working_nodes_https):
begin
    foreach endpoint in working_nodes_https do
        out=cat /etc/services | gfal-save endpoint/file
        if out != 0 then
            write endpoint to error-file
        else
            out=gfal-cat endpoint/file
            if out is 0 then
                write endpoint to working_nodes_rw
            else
                write endpoint to error-file
            end if
        end if
    end foreach
end
```

Fig. 3. R/W Selection Algorithm

3) Full Test Module

There are 3 scripts responsible for running the tests, one for running the XRootd tests, one for running either of the other utility tools tests and one for running everything together and adding the data to a single file ready to be given to the R script for analysis.

Each of the first two scripts get as input: the certificates and proxies necessary for accessing the endpoints, a file with the 3 endpoints on which the tests will be run on, a path to the folder that contains the test files, the name of the utility tool which the tests should be run with, the protocol that should be used for the endpoints and the number of downloads the script should be doing. In addition to the parameters specified above, the script for XRootd also gets a path to the meta-links folder and the script for the remaining tools also gets a CA-Bundle for Aria2.



```
function test_module(selected_nodes, X.509, files, command, iterations, protoco
  ca-bundle):
begin
    » foreach endpoint in selected_nodes do
        if command is init then
            gfal-mkdir endpoint/dir
        foreach file in files do
            if command is init then
                gfal-copy –cert X.509 files/file endpoint/dir/file
            for i ← 0 to iterations do
                if command is aria then
                    time=time aria2c -q –ca-certificate=ca-bundle
                      –certificate=X.509 –private-key=X.509 endpoint/dir/file
                      -o file
                    throughput=dimension/time
                    write protocol command source file dimension
                      throughput time to output
            end for
            rm file
        end foreach
        if command is delete then
            gfal-rm -r endpoint/dir
    end foreach
end
```

Fig. 4. Test Algorithm for Aria2 Single Source Only

The algorithm represents the single source test part for Aria2, where it downloads each file 100 time for every endpoint given. The multi-source test is similar with a few more parameters given to the aria2c command.

4) Shapiro-Wilk Test and Chart Module

Now, after having the file with the data out of the tests, the file is given as input to the R script which cleans the data, making it easier to present in a chart, runs the Shapiro-Wilk test on it and then plots the data for the comparison.

B. System Description

The WLCG is an infrastructure which enables the storage and distribution of data all over the world, more than 35.4 PB per month. In order to make this data available as fast as possible, the Grid is divided in 4 "Tiers", numbered from 0 to 3. Tier 0 is represented by the CERN Data Center, being the place where all data from the experiments pass through. Tier 1 consist of 13 computer centers dispersed all around the world and it is the place where Tier 0 distributes all the data it has. Tier 1 passes data, next, to Tier 2 which consist in universities and institutes that have enough processing power and storage to handle a sufficient amount of data and Tier 3 consist of any PC or local cluster that accesses the Grid.

Out of the 4 Tiers, for out tests, we are going to use, as destination, endpoints from the second Tier and, for the source, a virtual machine of medium size having 2 virtual CPUs and 4GB of RAM, located in CERN Geneva A availability zone, running CentOS 7 as operating system.

VI. Evaluation of Multi-Source Downloads

We are going to evaluated the performance gained by the use of multiple sources in comparison with single sources. Each of the charts below are going to be created out of the data gathered from downloading each of the files for 100 times, for the respective utility tool and protocol combination.

As can be seen, the data used in the charts is consistent, all utility tools having similar behavior from test to test, except for a few exceptions. We are dealing with test endpoints here and there are going to be a few inconsistencies due to this factor, but they are isolated and do not affect the end results.

Also, after running the Shapiro-Wilk test on the data we have observed that the data is not normally distributed, therefore we are going to use IRQ for plotting the results, meaning that each column from the chart is split into five parts, 2 whiskers denoting the maximum and minimum values, and 3 quartiles, for upper median, median and lower median. Simply put, the bigger the quartiles portion is, the less stable the downloads are.

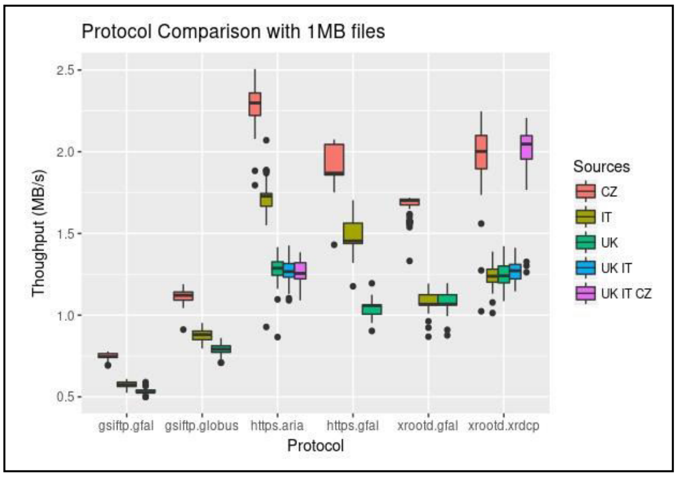


Fig. 5. Comparison between GFAL2, Globus, Aria2 and XRootd for 1 MB file

For the 1MB file test we can see that Aria2 is getting the best performance in single source tests, followed by XRootd and then GFAL2 using HTTPS which gets better results for the Italy endpoint where XRootd was probably affected by the endpoint instability.

Apart from that, it can be observed that for this dimension multiple sources have no impact for Aria2, which is normal, 1MB being too small to take advantage of them. That is not the case with XRootd thought. The way XRootd manages multiple sources results in better performance than Aria2 here.

When it comes to stability all utility tools are quite stable, given the small file size, and the impact of using multiple sources cannot be seen. The difference is going to be noticed in the next tests, once the download time get longer.

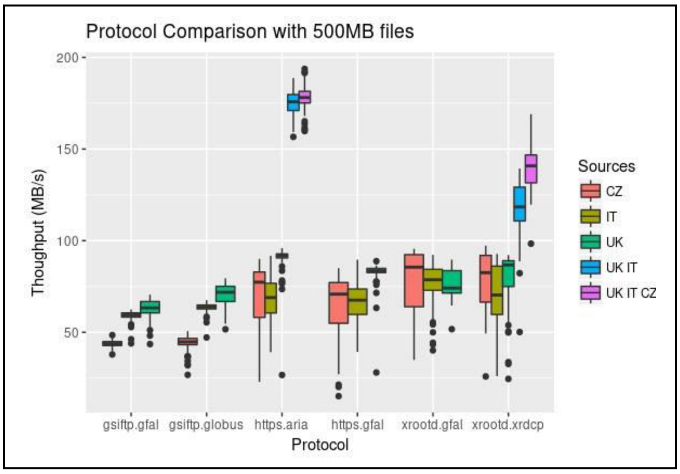


Fig. 6. Comparison between GFAL2, Globus, Aria2 and XRootd for 500MB file

Starting from the 500MB tests we can observe the real improvement brought by using multiple sources. Both Aria2 and XRootd are giving throughputs of over 100MB/s, Aria2 over 150MB/s even, while every other single source result is hardly reaching 90MB/s for the faster endpoints. Additionally, the throughput fluctuations are small for the multi-source results of Aria2.

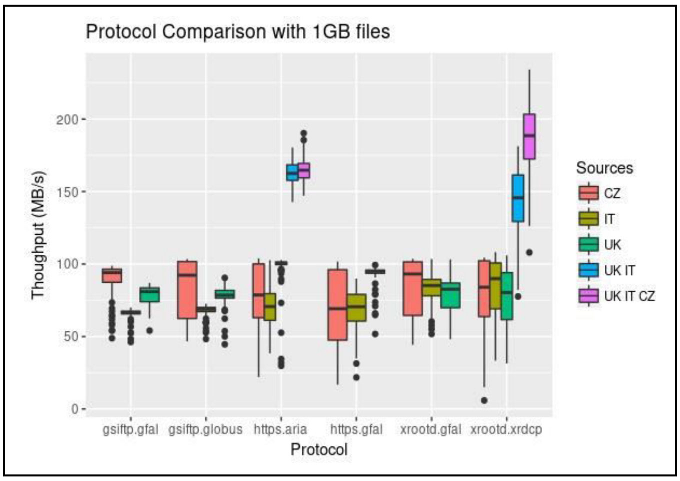


Fig. 7. Comparison between GFAL2, Globus, Aria2 and XRootd for 1GB file

At the 1GB file test we can see the same results as above. Using multiple sources clearly gives better performance than using a single source. Regarding XRootd, here it seems to give better performance than Aria2 for the 3 sources result.

In terms of stability, Aria2 is consistent and very stable while XRootd is performing similar with its single source results.

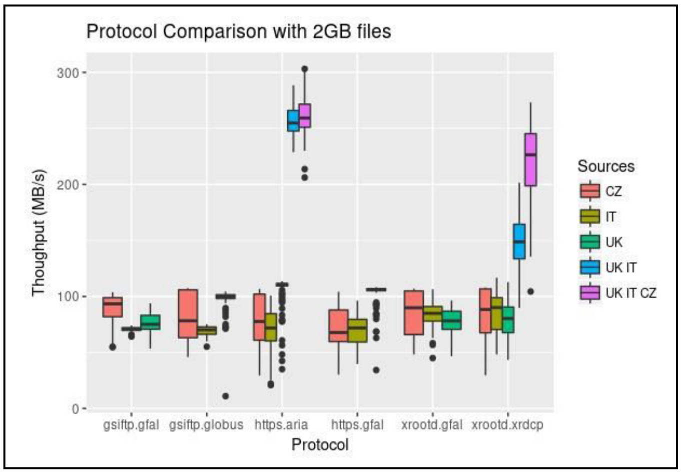


Fig. 8. Comparison between GFAL2, Globus, Aria2 and XRootd for 2GB file

In Fig. 8 we find the results for the 2 GB file test, that best demonstrates the impact of using multi-source downloads. For the single source downloads we are getting throughputs in the neighborhood of 100MB/s, while for the multiple sources, in the 3 sources case, Aria2 is reaching towards 300MB/s and XRootd towards 250MB/s.

VII. Conclusion

In this paper we have evaluated the impact over throughput and stability of using multiple sources for downloads in the Grid for the most frequent scenarios of file transfers and downloads of the ATLAS experiment and we have presented the advantages of using Aria2 and XRootd over the commonly used utility tools in the Grid - Globus Toolkit and GFAL2.

The WLCG is a system for which transfers and downloads represent a core function, a function whose optimization improves greatly the Grid itself. Therefore this evaluation comes in with the purpose of presenting how using multiple sources increases the throughput of downloads and in consequence the performance of the Grid for data movement.

After running the test we have observed a performance increase in terms of throughput and stability, for downloads in the Grid, when using multiple sources. We have seen how using Aria2 with 2 sources gives an increase of 69MB/s for the worst case and 155MB/s for the best case and with 3 sources shows a throughput increase of 71MB/s in the worst case and 160MB/s in the best case for the median values of the results. Similarly with XRootd, we have seen a slightly smaller improvement but still significant, for 2 sources worst case we got an increase of 36MB/s and for the best case 52MB/s, while for 3 sources worst case we have got 57MB/s and 121MB/s for the best case. Also the stability of the downloads are equal or better when using multiple sources.

In order to obtain these results we have run 100 downloads for multiple combinations of utility tools and protocols, for each chosen source. Then we have analyzed the distribution of the resulted data, concluded it is not normally distributed and plotted the charts, accordingly, using IRQ. That has given us a proper representation of the data, which could then be easily analyzed.

Acknowledgment

A. G. thanks the Data and Storage Services Group at the CERN IT Department for support.

References

[1] R. Buyya and S. Venugopal, “A gentle introduction to grid computing and technologies”, CSI Communications, July 2005 (19pp.).

[2] X. Espinal, et al., “Disk storage at CERN: Handling LHC data and beyond”, J. Phys.: Conf. Ser. 513 042017 (2014).

[3] V. Garonne, et al., “The ATLAS distributed data management project: past and future”, J. Phys. Conf. Ser., vol. 396, 032045 (2012).

[4] H. Subramoni, P. Lai, R. Kettimuthu, and D. K. Panda, “High performance data transfer in grid environment using GridFTP over InfiniBand”, 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, DOI: 10.1109/CCGRID.2010.115 (2010).

[5] R. Esposito, P. Mastroserio, G. Tortone, F. M. Taurino, “Standard FTP and GridFTP protocols for international data transfer in Pamela Satellite Space Experiment”, CHEP 2003, La Jolla, California, March 24-28 2003 (3 pp.).

[6] F. Donno, G. Vaglini, and A. Domenici, “Storage Management and Access in WLHC Computing Grid”, Pisa : Pisa Univ., 2007. - 174 p.

[7] P. Malzacher, A. Manafov, K. Schwarz, “RGLite, an interface between ROOT and gLite – PROOF on the Grid”, J. Phys. Conf. Ser., vol. 119, 072022 (2008).

[8] L. Bauerdick, et al., “Using Xrootd to Federate Regional Storage”, J. Phys. Conf. Ser., vol. 396, 042009 (2012).

[9] A. Dorigo, P. Elmer, F. Furano, A. Hanushevsky, “XROOTD - A highly scalable architecture for data access”, Proceeding TELE-INFO'05 Proceedings of the 4th WSEAS International Conference on Telecommunications and Informatics Article No. 46.

[10] D. Adamova and J Horky, “– An optimization of the ALICE XRootD storage cluster at the Tier-2 site in Czech Republic”, J. Phys. Conf. Ser., vol. 396, 042001 (2012).

[11] L. Bauerdick, et al., “XRootd, disk-based, caching proxy for optimization of data access, data placement and data replication”, J. Phys. Conf. Ser., vol. 513, 042044 (2014).

[12] H. Lamehamedi, B. Szymanski, Z. Shentu and E. Deelman, “– Data replication strategies in grid environments”, Proceeding ICA3PP '02 Proceedings of the Fifth International Conference on Algorithms and Architectures for Parallel Processing, p. 378.

[13] C. Dabrowski, “Reliability in grid computing systems”, Journal of Concurrency and Computation: Practice & Experience - A Special Issue from the Open Grid Forum Volume 21 Issue 8, June 2009, pp. 927-959.