Simulations of Distributed Systems in a Computing Centre

D. Dongiovanni, E. Ronchieri *Member, IEEE*, S. Dal Pra, L. dell'Agnello, T. Ferrari, S. Antonelli, A. Cavalli, D. Gregori, B. Martelli, A. Prosperini, P. P. Ricci

***Abstract*—Modern computing centres addressed to High Energy Physics user communities have to deal with rapidly hardware and software systems evolution. These centres normally face a variety of problems associated with the dimensioning and configuration of several services which must satisfy performance targets under different usage patterns. Therefore, the identification of key variables and the estimation of their impact on services performances is challenging. For example, given an hardware-software configuration for a considered service, how will service performance vary in relation to user dependent settings? Will it be able to support a certain number of requests per minute over the common parameter ranges? In addition, it is difficult to generalize the impact of same settings over different usage scenarios. Therefore, the design of a mathematical model able to relate services performance to key variables in the user computing patterns under common hardware-software settings can help to optimize the exploitation of computing resources.**

In the present work, starting from the analysis of a typical job of ATLAS as representative HEP user communities, we focus on how local data movement operations use hardware-software resources of INFN-CNAF computing centre and which variables affect data movement performances. As a result of this framework analysis we identify GridFTP protocol and GPFS data storage as core services whose performance to study in dependancy of typical user defined variables. We therefore decompose data movement commands in operations of increasing complexity i.e., `cp`, `globus-url-copy` with and without network, and after defining the *mean throughput* per file per unit size as target metric, we conduct a quantitative analysis of the contribution and relevance of considered variables across explored usage scenarios. Finally, we conduct a qualitative fit analysis of the behaviour of chosen throughput metric as a function of relevant user dependent variables. For each scenario and for each variables a best fit model function is selected according to *R-square* goodness of fit index.

I. Introduction

Computing centres committed to High Energy Physics (HEP) user communities normally have to deal with services dimensioning, optimization and configuration problems associated with data handling, job executions and user authentication-authorization. This is a challenging task given the need to match high performance targets across a variety of usage patterns and scenarios. Moreover the ongoing hardware and software setup upgrade make it difficult to predict the impact of settings on service performances and to have insight which variables affect more the user resource usage. For example, past experience cannot always answer the question whether a given setup will be able to support a certain number of data movement requests per minute. Therefore, the design of a mathematical model able to relate services performance to key variables in the user computing patterns under common hardware-software settings can help to optimize the exploitation of computing resources, representing a valid source of knowledge and verification for testing users needs in distributed computing paradigm.

HEP user communities work for the forthcoming Large Hadron Collider (LHC)[1] experiment design, accomplishement and data analysis. Computing and data storage of LHC are built and maintained by the Worldwide LHC Computing Grid (WLCG)[2] that is a global collaboration of more than 170 computing centres in 34 countries. WLCG computing centres are organized in tiers, which contribute to different aspects of WLCG. INFN-CNAF computing centre is one of LHC Tiers of level 1. In this study we selected ATLAS[3] as representative for typical computing resources usage. Typically, a computing job of ATLAS involves several entities of INFN-CNAF Tier1 schematized in data, farming, network, and infrastructure as shown in Fig. 1. Arrows in Fig. 1 represent interconnections between two entities. For example, an interconnection between data and farming entities is given by the need of a running job to get input data from, or to store the job output data into, storage resources. The job total execution time has many

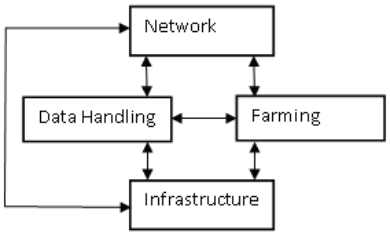

Fig. 1. Main entities of a computing centre.

contributions among which we highlight: i) time spent in resource queue and ii) time needed for data movement from and to storage are through network resources. The first is out

Corresponding Author: D. Dongiovanni is with the National Institute of Nuclear Physics, INFN-CNAF, Bologna, Italy (telephone: +390516092728, e-mail: danilo.dongiovanni@cnaf.infn.it).

Corresponding Author: E. Ronchieri is with the National Institute of Nuclear Physics, INFN-CNAF, Bologna, Italy (telephone: +390516092760, e-mail: elisabetta.ronchieri@cnaf.infn.it).

[1]The Large Hadron Collider, http://public.web.cern.ch/PUBLIC/en/LHC/LHC-en.html.

[2]The Worldwide LHC Computing Grid (WLCG), http://lcg.web.cern.ch/LCG/

[3]The ATLAS experiment, urlhttp://atlas.web.cern.ch/Atlas/index.html.

9781-4244-3962-1/09/$25.00 ©2009 IEEE

of scope of this study, while the second depends on variables either related to user community patterns or to hardware-software settings. Local data movement to run jobs exploits file protocol, `lcg-cp` or `lcg-cr` [1] commands commonly relying on GridFTP protocol [2]- [3] based on File Transfer Protocol [4]. At INFN-CNAF Tier1 IBM General Parallel File System (GPFS) [5] was the adopted solution for disk-based storage. Tests of read throughput versus time using different data access solutions like dCache [7] and xrootd [8] showed outstanding GPFS I/O performances and stability [6]. At INFN-CNAF Tier1 tests were performed for estimating a storage testbed set up which combines GridFTP servers with the IBM GPFS in order to evaluate interaction and performance issues [9].

In this paper, we focus on the analysis of HEP communities data movement activity at our centre, identifying the hardware and software resources that GridFTP and GPFS services exploit and the variables affecting GridFTP and GPFS performances. We decompose data movement activity in operations of increasing complexity able to give us insight about the real usage scenarios in order to define suitable metric and build data set allowing for a quantitative analysis of the contribution and relevance of considered variables across explored usage scenarios. In our approach, we use Matlab-code language for designing quantitative metrics, able to model complex systems with the use of small code size and capable to manage user written functions easy to maintain over time. The structure of the paper is as follows. Section II describes materials and methods adopted in our study to help understanding the work and the proper thought process. Section III presents results and analysis, while Section IV discusses results. Section V discusses the future works and Section VI concludes.

II. Materials And Methods

In this section we present the adopted approach to identify hardware and software components exploited in common usage patterns, key metrics to monitor and the way they depend on typical user-dependent variables. We also provide details about the testbed set up, test performed and adopted data analysis procedure to derive the target metric measurements under considered variables ranges.

Given the variety of user communities accessing a Tier1 computing centre, we first identified a HEP user community, ATLAS, representative for typical computing resources usage. Then, we analyzed the way a typical ATLAS computing job exploits storage, farming and network resources (Fig. 1). The job total execution time has many contributions among which we mention: time for authentication and authorization, time spent in resources queue, user application execution time and time needed for data movement from and to storage area through network resources. In the present study we focus on data movement from and to storage area.

ATLAS community jobs in a Tier1 exploit storage area either to move data from and to other remote computing centres or to run experiments analysis locally reading input data and writing outputs. Data movement between computing centres is based on File Transfer Service (FTS) [10] which relies on some Storage Resource Management (SRM) [11] systems such as CASTOR [12] or StoRM [13] which in turn rely on some file transfer protocols, among which GridFTP [14] is the most commonly used in distributed computing infrastructure (like Enabling Grid for E-sciencE[4], WLCG). Moreover, StoRM takes advantage from high performing cluster File System as GPFS from IBM (adopted at INFN-CNAF Tier1 as disk-base storage solution), but it supports also every standard POSIX File System [15]. Local data movement to run analysis jobs exploits `lcg-cp` or `lcg-cr` commands also relying on GridFTP protocol and file protocol. Therefore, GridFTP file transfer protocol is a core service for most data movement operations and GPFS optimizes data management. So, we further focused our study on: i) identifying the hardware and software resources that GridFTP and GPFS services exploit; ii) identifying the variables affecting GridFTP and GPFS performances; iii) defining a set of basic data movement operations of increasing complexity able to give us insight about real usage scenarios; iv) defining a suitable metric and build an experimental data set allowing for a quantitative analysis of the contribution and relevance of considered variables across explored usage scenarios.

A. Hardware set-up description

The hardware set-up underlying data movement operations in our study case is characterized by: GPFS File System, GridFTP servers, Fiber Channel links, LAN links and storage disk. As highlighted in Fig. 2, the hardware set-up can be roughly divided into three main layers detailed as follows:

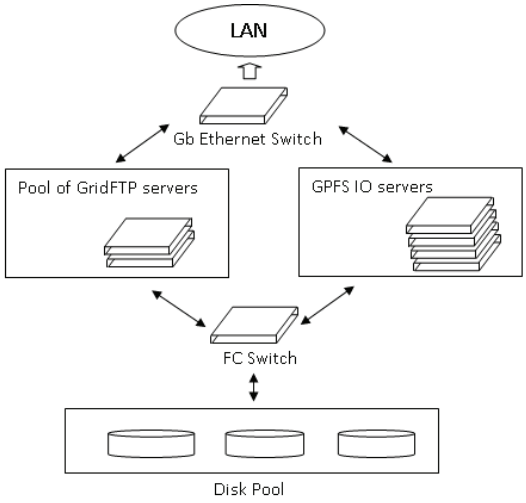

Fig. 2. Hardware set-up description.

1) Storage disks: This layer includes the disk pool, composed by four Logical Unit Numbers each of 8 (TB) that are served by an EMC CLARiiON CX3 model 80 subsystem. EMC CLARiiON CX3 model 80 is the largest, most powerful storage array in the CX3 series. It is based on the CLARiiON CX3 UltraScale architecture, providing high-performance high-capacity networked storage.

[4]Enabling Grid for E-sciencE, http://www.eu-egee.org/

2) Storage servers: This layer includes the servers hosting GridFTP and GPFS servers, consisting of six M600 Blades from Dell PowerEdge M1000e Modular Blade Enclosure. Each blade server has been equipped with:

- two quad core Intel(R) Xeon(R) CPU E5410 at 2.33 (GHz);
- 16 (GB) of RAM;
- 2 Serial Attached SCSI (SAS) Hard Drivers of 130 (GB) each configured in RAID 1 using on-board LSI Logic / Symbios Logic SAS1068E controller, and used as local disk.

These six blade servers have been divided in two groups: one composed by four blades that run the GPFS servers Network Shared Disk, the other composed by two blades used as GridFTP servers with direct access to GPFS by Fibre Channel (FC) connection on external storage disk layer. This configuration provides a complete separation between GridFTP and GPFS data flow as it is shown in Fig. 2.

3) Network: This layer includes two onboard Gigabit Ethernet ports configured in Channel Bonding, which allows for each server to use 2 (Gb/s) Ethernet interconnection with 48-port Extreme X450 switch with 10 (Gb/s) uplink. The interconnection with external storage disk layer provided by dual-channel QLogic Corporation QLA2432 4 (Gb/s) FC adapter and 4 (Gb/s) optical FC links, connecting each blade with two Brocade M4424 switches installed on the Dell M1000e Blade Enclosures.

Channel Bonding is a method in which data gets striped in each message across multiple network cards installed in each machine and combined for redundancy or increased throughput.

B. Variables Identification

Here we want to identify main variables possibly affecting the performance of data movement operations. These variables can be grouped into two main categories: i) variables related to Transmission Control Protocol (TCP) [16] and GPFS settings, which are fixed for users at run time, therefore here called static variables and summarized in Table I; ii) user-settable variables that we call dynamic variables, summarized in Table III.

1) Static variables: The following settings were adopted or observed:

- Round trip time: 0.1001 (ms)
- TCP buffer size: $\geq$ ($\frac{bandwidth_{max}}{8}$ x Round Trip Time)
- TCP window size: [0.016, 16] (Gb)
- GPFS pagepool: 8 (GB)

2) Dynamic user-dependent variables: Considering ATLAS's user community common settings in transferring data exploiting FTS, we identified the range of user-dependent variables as follows:

- size of file to be transferred in the range [0.15, 2] (GB)
- number of processed parallel files belonged to [10, 30]
- number of streams was 5
- type of protocol was gsiftp

TABLE I
STATIC VARIABLES.

Type	Description
Network latency	The amount of time for a packet to traverse the network.
TCP buffer size	Number of bytes read from or written to socket.
Round Trip Time	Time required for a signal pulse or packet to travel from a specific source to a specific destination and back again.
TCP window size	Number of bytes (beyond the sequence number in the acknowledgment field) that the receiver is currently willing to receive.
GPFS block size	File system block size.
GPFS pagepool	Mechanism used to cache user data and file system metadata. It allows GPFS to implement read as well as write requests asynchronously. Increasing the size of pagepool increases the amount of data or metadata that GPFS may cache without requiring synchronous I/O.

TABLE II
DYNAMIC VARIABLES.

Type	Description
Number of files	Number of files processed in parallel.
Size of file	The size of transferred file.
Number of streams	Number of parallel data connection used.

C. Considered scenarios

To study how considered variables affect the execution time of data movement operations we considered three types of scenarios of increasing complexity:

1) Scenario 1: This data movement scenario consists of a simple copy from source GPFS block device to destination GPFS block device operated with `cp` unix command on a SAN node. This is the simplest scenario, involving neither authentication-authorization nor network contributes to total execution time. So it provides us with an estimate of GPFS behaviour with respect to considered variables.

2) Scenario 2: This data movement scenario consists of multi-protocol data movement `globus-url-copy` from source GPFS block device to destination GPFS block device operated on a SAN with enabled GridFTP servers. In this scenario the final performances are possibly affected by overheads introduced by GridFTP and GSI security. Nevertheless, having configured network parameters to adopt loopback device, no network layer overhead is introduced.

3) Scenario 3: This data movement scenario consists of multi-protocol data movement `globus-url-copy` from source GPFS block device to destination GPFS block device operated between two SANs each with GridFTP servers enabled. In this last scenario, with respect to second scenario, the final performance is also possibly affected by the Local Area Network (LAN) layer overhead.

D. Target metric definition and data set construction

To compare performances over considered scenarios and identify the relevant variables in each case we adopted the

mean throughput per file per unit size as target metric. The mean throughput was derived as a function of mean execution time in the three considered scenarios, according to fraction $\frac{\text{size\_of\_file}}{\mu_{\text{time}}}$. For each considered scenario, tests consisted of executing operation like `cp` or `globus-url-copy` as described above. The execution time was measured for each test. Each test was iterated 200 times to build a representative sample of the execution time population. The mean execution time μ_{time} and its associated error σ_{time} were calculated for each sample and the target metric $\mu_{\text{throughput}}$ and its associated error $\sigma_{\text{throughput}}$ derived consequently. For all tests, the static variables described above were kept fixed to defaults values while the dynamic user-dependent variables (number of files, size of files, number of streams) were varied across the following ranges covering typical user communities range:

number of files = [1, 2, 5, 10, 15, 20, 25, 30, 35, 40, 45]
size of files = [0.01, 0.1, 0.2, 0.5, 1, 2] (GB)
number of streams = [1, 5, 10, 15, 20]

We observe that $\sigma_{\text{throughput}}$ was calculated according to (1):

$$\sigma_{\text{througput}} = \frac{\text{size\_of\_file}}{\mu^2_{\text{time}}}\sigma_{\text{time}} \quad (1)$$

Notice that for tests with the number of files processed in parallel greater than 1, the throughput mean and its associated error were derived for each sample associated to each file, and the final mean throughput μ_f was extracted using weighted throughput mean and its associated error σ_f, expressed by (2) and (3)

$$\mu_f = \frac{\sum_{i=1}^{\text{number\_of\_files}} \frac{\mu_{\text{throughput}_i}}{\sigma^2_{\text{throughput}_i}}}{\sum_{i=1}^{\text{number\_of\_files}} \frac{1}{\sigma^2_{\text{throughput}_i}}} \quad (2)$$

$$\sigma_f = \frac{\sqrt{k\sum_{i=1}^{\text{number\_of\_files}} \frac{\mu^2_{\text{throughput}_i}}{\sigma^2_{\text{throughput}_i}}}}{\sqrt{\sum_{i=1}^{\text{number\_of\_files}} \frac{1}{\sigma^2_{\text{throughput}_i}}}} \quad (3)$$

where k is $\frac{1}{200-1}$.

III. Results

In this section we present experimental results describing the behaviour target metric identified in Section II as mean throughput of data movement operations as a function of three main user-dependent variables identified as: i) size of file to be transferred; ii) number of files transferred in parallel; iii) number of streams per transferred file. To evaluate the contribution of the transfer protocol used in the transfer test, we explored performances under the three scenarios of increasing complexity: i) a simple `cp gpfs_source gpfs_destination` command; ii) `globus-url-copy gpfs_source gpfs_destination` performed on single SAN nodes with enabled GridFTP server; iii) `globus-url-copy gpfs_source gpfs_destination` performed between two SAN nodes with enabled GridFTP server. The GPFS set up and testbed described in Section II were kept fixed over all tests performed. The throughput was derived as the ratio between transferred file size (GB) and the mean time (s) elapsed for each single transfer operation considered. Results presented in the following describe the behaviour of throughput per file and per size. Moreover performed tests involve operations exploiting GPFS both in read and write modality in a symmetric way so that reported throughput must be considered as the observed mean value over the two access modalities.

A. Scenario 1

In the first considered file transfer scenario, `cp gpfs_source gpfs_destination`, we studied the behaviour of mean throughput as a function of transferred file size and number of parallel files transferred in a common user range [0.01, 2] (GB) (covering the typical range of ATLAS HEP user community), with the number of parallel files transferred ranging from 1 to 45. The variable number of streams is not defined in this scenario, therefore it has not been considered. Across the considered range, we observed a similar behaviour of throughput as a function of size and number of parallel files as shown in Fig.4 and Fig. 3 respectively: an initial fast decrease is followed by slow decreasing range. To model this behaviour we performed

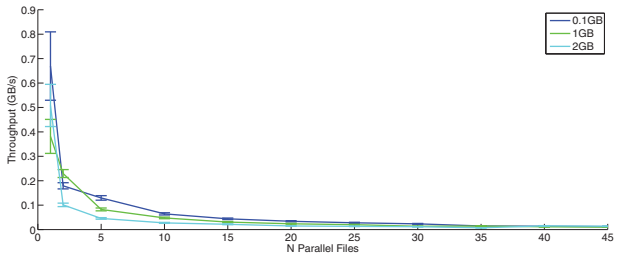

Fig. 3. CP test: Throughput vs Number of Parallel transferred Files

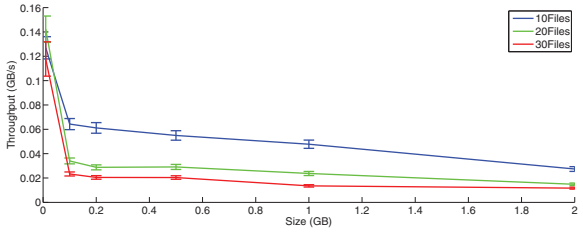

Fig. 4. CP test: Throughput vs Size of transferred Files

a fit analysis on the experimental data using a set of trial target functions and then ranking fit results according to *R-square* goodness of fit index [18]-[17] defined as a function of the sum of squares of the regression and the total sum of squares, as reported in equations (4) [22]. The fit analysis

was conducted considering the two variables as independent.

$$R\text{-}square = 1 - \frac{SSE}{SST}$$
$$SSE = \sum_{i=1}^{n} w_i \cdot (\hat{y}_i - \bar{y})^2$$
$$SST = \sum_{i=1}^{n} w_i \cdot (y_i - \bar{y})^2 \quad (4)$$

In the set of trial target functions we considered the following: polynomial, power, exponential and gaussian. The fit analysis showed that for both user dependent variables in `cp` *scenario* the function providing the best fit *R-square* index for the observed throughput behavior is given by a sum of two exponential functions (5), where each exponential term accounts for the behaviour in the initial and final range respectively.

$$f(x) = a \cdot e^{-b \cdot x} + c \cdot e^{-d \cdot x} \quad (5)$$

with $a, b, c, d \in \Re^+$. The mean *R-square* was respectively equal to 0.93 for throughput as a function of size at different number of parallel files curves and 0.97 for throughput as a function of files at different size curves.

B. Scenario 2

In the second considered scenario, `globus-url-copy gpfs_source gpfs_destination`, the resulting mean throughput takes into account the GridFTP transfer protocol overload with respect to the first scenario, which mainly consists of the authentication operation performed via the Generic Security Services (GSS) API [19] and the authorization operation performed via Community Authorization Service (CAS) API [20]. We expect no network overhead in this case cause of the loopback configuration described in Section II, so the number of streams variable will not be considered. Therefore, in the following we analyse the behaviour of mean throughput for size of transferred file and number of parallel files transferred varying in the range [0.01, 2] (GB) and [1, 30] respectively. In this second scenario, the

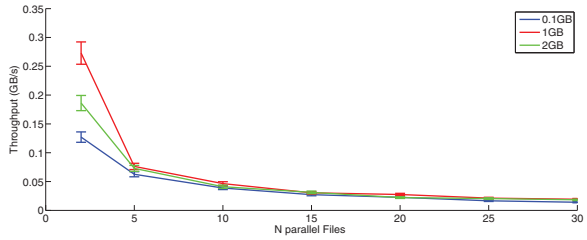

Fig. 5. Globus-Url-Copy test on single SAN node: Throughput vs Number of Parallel transferred Files

observed throughput behaviour differs with respect to the two considered variables. Mean throughput as a function of the number of parallel files transferred shows a similar behaviour to the one observed in the first scenario: an initial fast decrease is followed by a slowly decreasing range as shown in Fig. 5. Again fit analysis shows that the function providing the best fit indexes for the observed throughput behavior is given by a

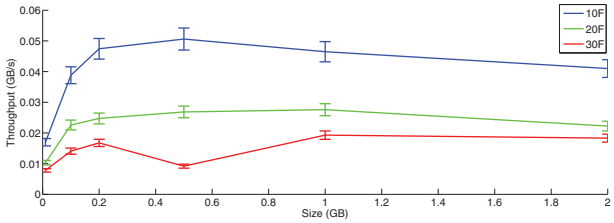

Fig. 6. Globus-Url-Copy test on single SAN node: Throughput vs Size of transferred Files

sum of two exponential functions (5), with a mean *R-square* index of 0.99 over curves at different sizes. On the other hand, mean throughput as a function of the size of transferred files shows an initial fast increase followed by a slowly increasing area asymptotically reaching a plateau as shown in Fig. 6. Fit analysis in this case shows that the observed behaviour can be modelled by an exponential function of the form (6), providing a mean *R-square* index of 0.89 over curves at different number of parallel files.

$$f(x) = a \cdot e^{-b \cdot x} - c \cdot e^{-d \cdot x} \quad (6)$$

with $a, b, c, d \in \Re^+$. A possible explanation for lower throughput values for smaller size files can be found in a higher relative contribute of `globus-url-copy` authentication/authorization time for small size files. To confirm this hypothesis we estimated the authorization mean time by repeating the `globus-url-copy` test with zero size files, varying the number of parallel files transferred again in the range [1, 30]. Then we calculated the mean throughput as a function of the size of transferred files subtracting the estimated authorization time at corresponding values of parallel files. Resulting throughput behavior in Fig. 7 shows an initial fast decrease followed by a plateau, similarly to behaviour observed in `cp` scenario above.

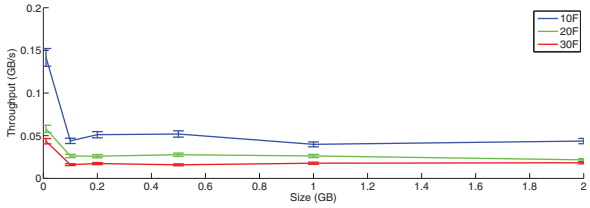

Fig. 7. Globus-Url-Copy test on single SAN node: Throughput vs Size of transferred Files with estimated authorization time subtracted

C. Scenario 3

In the third considered scenario, `globus-url-copy gpfs_source gpfs_destination`, performed between two different SAN nodes with GridFTP servers enabled, a new user dependent variable is introduced as a consequence of network exploitation: the number of streams per transferred file. In our set up, involving a `globus-url-copy` within a LAN, with small round trip time (see Section II), we did not observe a significant mean throughput variation when increasing the number of streams used per file transfer in the

variables range considered as shown in Fig. (8) (also observed in [21]). Therefore, in the following we analyse the behaviour

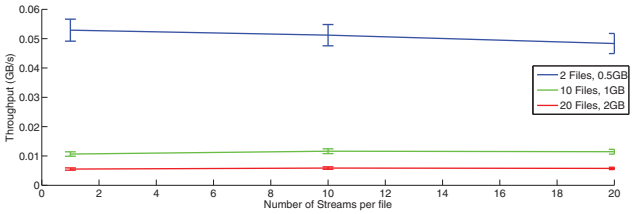

Fig. 8. Globus-Url-Copy test between two SAN node: Throughput does not significantly vary when varying the number of streams per transferred File

of mean throughput fixing the number of streams variable to 1, while size of transferred file and number of parallel files transferred, vary in the range [0.01, 2] (GB) and [1, 30] respectively.

From Fig. 9 and Fig. 10 we notice how throughput as a function of the number of parallel files transferred shows a behaviour which is consistent with the one observed in scenario 2 while the behaviour as a function of size of files transferred is qualitatively consistent with the one in scenario 2 only when the number of files transferred is lower than 10, suggesting that when bandwidth approaches saturation size of files becomes irrelevant. Fit analysis report results which are consistent with what observed in the second scenario with function (5) providing the best fit indexes, mean *R-square* index of 0.98 over curves at different sizes for throughput as a function of the number of parallel files. Throughput as a function of size can be modelled by an exponential function of the form (6), mean *R-square* index of 0.90 over curves at different number of parallel files up to 10 files.

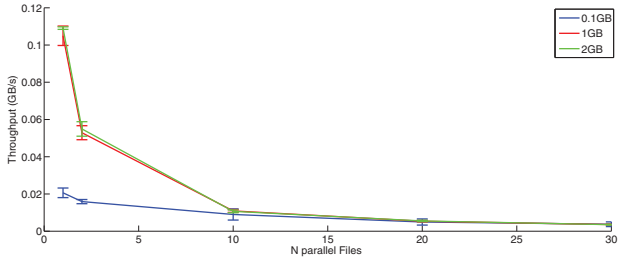

Fig. 9. Globus-Url-Copy test between two SAN nodes: Throughput vs Number of Parallel transferred Files

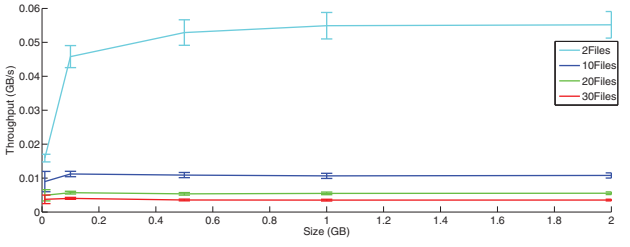

Fig. 10. Globus-Url-Copy test between two SAN nodes: Throughput vs Size of transferred Files

IV. Discussions

For a comparative overview of results for all three scenarios, we report a summary of fit results in Table IV.

Comparing results over scenarios and variables reported in the table, we notice that the exploited protocol (`cp` in scenario 1 vs `globus-url-copy` in scenarios 2 and 3), seems to explain the observed difference in throughput behaviour as a function of transferred file size, with authorization-authentication contribution accounting for lower throughput observed at small size as suggested before in this section. No qualitative throughput behaviour difference has been observed across all scenarios when varying the number of parallel files processed, suggesting that increasing the number of files has the main effect of reducing the amount of available bandwidth.

It is also interesting to consider the total throughput derived as a mean value across all considered sizes and number of files processed in parallel. The total mean was obtained multiplying the throughput per file considered in the analysis above by the number of files. Results about the mentioned mean total throughput are reported in Table IV. Comparing results across

TABLE IV
Total Throughput

Scenario 1	Scenario 2	Scenario 3
0.76 (GB/s)	0.38 (GB/s)	0.10 (GB/s)

scenarios we notice that both exploited protocol and network contributions affect the total throughput observed which constantly decrease when changing protocol from scenario 1 to 2 and when adding network from scenario 2 to 3.

V. FutureWorks

In this work we presented a quantitative analysis of user-dependent variables across explored usage scenarios and model observed behaviours using *R-square* index.

Planned future enhancements to the analysis include the following: tests changing TCP settings like window size; tests changing GPFS settings; tests estimating the effect of WAN or network latency; tests validating quantitative models using other HEP communities.

VI. Conclusions

Given the hardware and software set-up (GPFS, GridFTP, FC links, LAN links) available in our computing centre, we considered the variation of throughput performances in dependency of three main user-defined variables: size of transferred files, number of files transferred in parallel, number of streams per transferred file. The mean throughput per size and per file was experimentally derived as a function of the execution time of three data movement core operations: `cp` from and to GPFS files system, `globus-url-copy` from and to GPFS file system within same GridFTP server and between two different GridFTP servers across LAN. Experimental data were derived for the three scenarios over typical HEP community ranges for considered user variables, i.e., size of file in the range [0.1,2] (GB), number of transferred files in the range [1,30] and number of streams per file in the range [1,20].

TABLE III
OVERVIEW OF MODELING RESULTS

Operation Performed	Source	Dest	Var Considered	Var Range	Best Fit Function	Mean $R-square$ Index
`cp`	gpfs-source	gpfs-dest	N parallel files	[1,45]	$f(x) = a \cdot e^{-b \cdot x} + c \cdot e^{-d \cdot x}$	0.93
`cp`	gpfs-source	gpfs-dest	Size of files	[0.01, 2] (GB)	$f(x) = a \cdot e^{-b \cdot x} + c \cdot e^{-d \cdot x}$	0.97
`globus-url-copy` 1 SAN node	gpfs-source	gpfs-dest	N parallel files	[1, 30]	$f(x) = a \cdot e^{-b \cdot x} + c \cdot e^{-d \cdot x}$	0.99
`globus-url-copy` 1 SAN node	gpfs-source	gpfs-dest	Size of files	[0.01, 2] (GB)	$f(x) = a \cdot e^{-b \cdot x} - c \cdot e^{-d \cdot x}$	0.89
`globus-url-copy` 2 SAN nodes	gpfs-source	gpfs-dest	N parallel files	[1, 30]	$f(x) = a \cdot e^{-b \cdot x} + c \cdot e^{-d \cdot x}$	0.98
`globus-url-copy` 2 SAN nodes	gpfs-source	gpfs-dest	Size of files	[0.01, 2] (GB), $<$ 10 parallel files	$f(x) = a \cdot e^{-b \cdot x} - c \cdot e^{-d \cdot x}$	0.90

Considering the throughput per second versus number of files transferred in parallel, we observed that throughput decreases exponentially when increasing the number of files, with a faster decrease affecting the range 1 to 10 files. This behaviour is consistent across all three considered scenarios and can be effectively modelled by a sum of negative exponential functions. The throughput per second versus size of transferred files showed a different behaviour when passing from `cp` to `globus-url-copy` data movement protocol. When considering `cp` scenario we observed the same throughput behaviour as the one observed for number of files variable, again effectively modelled by a sum of negative exponential functions. When considering `globus-url-copy` scenarios we observed a fast increase in throughput for files of size < 0.2 (GB), followed by a range of no significant variation for higher size files. This behaviour is effectively modelled by a difference of negative exponential functions. This dependency on the movement protocol used can be explained by the additional cost of authentication, authorization operations performed in `globus-url-copy` protocol, whose overhead is more relevant for small size files. We did not observe a significant variation of throughput per second over considered range of number of streams per file which can be explained with the small round trip time 0.1 (ms) measured in our testbed. Finally we noticed that despite the qualitative throughput behaviour homogeneity across scenarios for number of parallel files variables and within same transfer protocol for file size variable respectively, both exploited protocol and network contributions quantitatively affect the total throughput observed which constantly decrease when changing protocol passing from scenario 1 to 2 and when adding network contribution passing from scenario 2 to 3.

REFERENCES

[1] A. Delgado Peris, P. Mèndez Lorenzo, F. Donno, A. Sciabà, S. Campana, and R. Santinelli, *LCG-2 USER GUIDE MANUALS SERIES*, August 4, 2005, https://edms.cern.ch/file/454439//LCG-2-UserGuide.pdf.

[2] W. Allcock, *GridFTP: Protocol Extensions to FTP for the Grid*, Global Grid Forum GFD-R-P.020, 2003.

[3] R. Kettimuthu, L. Wantao, J. Link, and J. Bresnahan, *A GridFTP Transport Driver for Globus XIO*, in Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2008), Las Vegas, Nevada, USA, July 14–17, 2008, Vol. 2, pp. 843–849.

[4] J. Postel, *RFC765 - File Transfer Protocol specification*, June 1980, http://www.faqs.org/ftp/rfc/pdf/rfc765.txt.pdf.

[5] S. Fadden, *An Introduction to GPFS Version 3.2.1*, ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/clw03005usen/CLW03005USEN.PDF.

[6] A. Fella, F. Furano, L. Li Gioi, F. Noferini, M. Steinke, D. Andreotti, A. Cavalli, A. Chierici, L. dell'Agnello, D. Gregori, A. Italiano, E. Luppi, B. Martelli, P. P. Ricci, E. Ronchieri, D. Salomoni, V. Sapunenko, and D. Vitlacil, *A Comparison of Data-Access Platforms for BaBar and ALICE analysis Computing Model at the Italian Tier1*, submitted for pubblication to J. Physics: Conference Series, 2009.

[7] P. Fuhrmann, and V. Glzow, *dCache, Storage System for the Future*, Springer-Verlag Berlin Heidelberg 2006, W. E. Nagel et al. (Eds.): Euro-Par 2006, LNCS 4128, pp. 1106–1113.

[8] A. Hanushevsky, A. Dorigo, and F. Donno, *THE NEXT GENERATION ROOT FILE SERVER*, in Proceedings of the International Conference on Computing in High Energy Physics and Nuclear Physics 2004 (CHEP'04), Interlaken, Switzerland, 27 Sep–1 Oct 2004, pp. 680–684.

[9] A. Cavalli, C. Ciocca, L. dell'Agnello, T. Ferrari, D. Gregori, B. Martelli, A. Prosperini, P. P. Ricci, E. Ronchieri, V. Sapunenko, A. Sartirana, D. Vitlacil, and S. Zani, *On Enhancing GridFTP and GPFS performances*, submitted for pubblication to J. Physics: Conference Series, 2009.

[10] P. Baldino, P. Z. Kunszt, and G. McCance, *The FTS paper for CHEP06 : The gLite File Transfer Service*, in Proceedings of the 5th International Conference on Computing In High Energy and Nuclear Physics, Mumbai, India, 13–17 Feb 2006, pp. 685–688.

[11] T. Perelmutov, J. Bakken, and D. Petravick, *STORAGE RESOURCE MANAGER*, in Proceedings of the International Conference on Computing in High Energy and Nuclear Physics 2004 (CHEP'04), Interlaken, Switzerland, 27 Sep–1 Oct 2004.

[12] G. Lo Presti, O. Barring, A. Earl, R. M. Garcia Rioja, S. Ponce, G. Taurelli, D. Waldron, and M. Cohelo Dos Santos, *CASTOR: A Distributed Storage Resource Facility for High Performance Data Processing at CERN*, in Proceedings of MSST 2007, pp. 275–280.

[13] A. Carbone, L. dell'Agnello, A. Forti, A. Ghiselli, E. Lanciotti, L. Magnoni, M. Mazzucato, R. Santinelli, V. Sapunenko, V. Vagnoni, R. Zappi, *Performance Studies of the StoRM Storage Resource Manager*, in Proceedings of eScience 2007, pp. 423–430.

[14] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke, *Data management and transfer in high–performance computational grid environments*, Parallel Computing, vol. 28, issue 5, May 2002, pp. 749–771.

[15] IEEE Std 1003.1-2001, Open Group Technical Standard, Issue 6, *Standard for Information Technology–Portable Operating System Interface (POSIX)*, 2001, ISBN 0-7381-3010-9, http://www.ieee.org/.

[16] DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION, *RFC793 - Transmission Control Protocol*, September 1981, http://www.faqs.org/ftp/rfc/pdf/rfc793.txt.pdf

[17] M. J. R. Healy, *The Use of R2 as a Measure of Goodness of Fit*, J. Royal Statistical Society, vol. 147, part. 4, 1984, pp. 608–609.

[18] A. Colin Cameron, and F. .A. .G. Windmeijer, *An R-squared measure of goodness of fit for some common nonlinear regression models*, J. Econometrics, vol. 77, 1997, pp. 329–342.

[19] I. Foster, C. Kesselman, G. Tsudik, and A. Tuecke, *A Security Architecture for Computational Grids*, in Proceedings of the 5th ACM Conference on Computer and Communications Security, 1998, pp. 83-91.

[20] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, *A Community Authorization Service for Group Collaboration*, in Proceedings of

the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02), 2002, pp. 50–59, Monterey, CA, USA, 2002.
[21] W. Allcock, J. Bester, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, and I. Foster, *The Globus Striped GridFTP Framework and Server*, in Proceedings of the ACM/IEEE Supercomputing (SC) 2005, 12–18 Nov. 2005, pp. 54–65.
[22] MATLAB, *Curve Fitting Toolbox^{TM} 2, User's Guide*, The MathWorks, chap. 5, pp. 32.