Remote data access in computational jobs on the ATLAS data grid

Volodimir Begy^{1,2,*}, Martin Barisits¹, Mario Lassnig¹, and Erich Schikuta²
on behalf of the ATLAS Collaboration

¹CERN, Geneva 23, Switzerland

²University of Vienna, Vienna, Austria

Abstract. This work describes the technique of remote data access from computational jobs on the ATLAS data grid. In comparison to traditional data movement and stage-in approaches it is well suited for data transfers which are asynchronous with respect to the job execution. Hence, it can be used for optimization of data access patterns based on various policies. In this study, remote data access is realized with the HTTP and WebDAV protocols, and is investigated in the context of intra- and inter-computing site data transfers. In both cases, the typical scenarios for application of remote data access are identified. The paper also presents an analysis of parameters influencing the data goodput between heterogeneous storage element - worker node pairs on the grid.

1 Introduction

The ATLAS Experiment at CERN [1] stores approximately 400 petabytes of data across more than 150 data centers within the Worldwide LHC Computing Grid (WLCG) [2, 3]. More than 5000 grid users submit analysis and production jobs. The data is accessed based on data placement, stage-in and remote access techniques. In practice, the data placement is typically employed for scheduled inter-storage element data transfers [4], while stage-in and remote access [5–7] are used by worker nodes locally within a single computing site. However, other data access profiles, such as direct reading from remote storage elements, or arbitrary combinations of the various access techniques within a single computational campaign are possible.

2 Remote data access

Consider a computational job scheduled to run at the computing site S_1 . The job requires some fraction of a file persisted at the remote computing site S_2 as input. According to the current data access policies of the ATLAS data grid, the complete file first needs to be placed from the remote storage element to the local one. Then, it can be accessed by the worker node in the stage-in manner. In this section, we refer to this procedure as *local access*. An alternative is to read the relevant fraction of the file directly from the remote storage element at S_2 . Here, we refer to this access profile as *remote access*.

*e-mail: volodimir.begy@cern.ch

2.1 Goodput break-even point

In this subsection we present hypothetical examples, demonstrating the suitability of local/remote data access profiles in various scenarios. The examples are based on a simplistic grid topology, depicted in figure 1. A grid comprises two sites S_1 and S_2 . Jobs are scheduled to run at the local site S_1 , which consists of the storage element SE_1 and 40 worker nodes $w_1, w_2, \dots, w_{40}$. The second site S_2 has a storage element SE_2 (and neglected workers). All the input data needed by computational jobs $j_1, j_2, \dots, j_n$ is stored in a single file F_1 with a size of 100GB. F_1 is stored in SE_2 . The sites S_1 and S_2 are connected over a 3Gbps WAN link. Each of the storage elements SE_1 and SE_2 is connected to the WAN link over a 3Gbps LAN link. Each of the workers $w_1, w_2, \dots, w_{40}$ is connected to the local storage element SE_1 and to the WAN link over a 1Gbps LAN link. Each of the storage elements SE_1 and SE_2 has a maximum data throughput of 5Gbps. Each worker $w_1, w_2, \dots, w_{40}$ has a maximum data throughput of 1Gbps. In the demonstrated examples the same effect is reached, in case the storage element SE_2 has a maximum throughput of 3Gbps, while the WAN link has a bandwidth of ≥ 3 Gbps. The one-way network latency accounts for the end-to-end delay. Between all local machines this latency is 20ms, between all remote machines it is 100ms. The grid employs a reliable data transfer protocol (thus, network latency has a direct impact on the overall data transfer time).

Figure 1. Example grid topology.

For the sake of simplicity, we assume that there is no slow-start phase in the data transfer protocol. We assume that packet switches are not slowing down the transfer process in any way. The packet size is set to 1MB (while the header bytes are neglected), the packet loss rate is 1% over WAN and 0.1% over LAN, while during the second attempt all the packets are guaranteed to be delivered. The protocol employs two asynchronous channels: a data channel and a control channel. The traffic needed for the transmission of control messages is neglected. Every time a lost packet is detected and retransmitted by the sender, the transfer time is increased by the network latency. We assume that the maximum storage capacities of all the machines are not reached, and that there are always available CPU cores for the processing. We assume that storage elements share the maximum data throughput and the WAN link shares the bandwidth equally between all parallel data flows.

In this highly abstracted calculation of the data transfer time [8] we only consider two dominant components: (1) the delay resulting from the first attempt to transfer all the packets;

(2) accumulated latencies resulting from the second attempt to transfer the lost packets. The general formula is:

$$t_{S \rightarrow D} = d_{transfer} + l * ((F/P) * R_{loss})$$

where $t_{S \rightarrow D}$ is the data transfer time from source to destination, $d_{transfer}$ is the aforementioned first component, l is the concerned latency, F is the size of the data being transferred, P is the packet size, R_{loss} is the loss rate.

$d_{transfer}$ itself is also calculated on a high abstraction level, by

$$d_{transfer} = F/T$$

where T is the end-to-end throughput of the transfer, which is fairly determined considering the amount of parallel data flows and the physical limits of hardware.

In the first example, each worker node $w_1, w_2, \dots, w_{40}$ is assigned a single computational job $j_1, j_2, \dots, j_{40}$ respectively. Each job $j_1, j_2, \dots, j_{40}$ requires a distinct fraction of the file F_1 , which is located at the remote storage element SE_2 as input. The size of the requested file fraction is varied from 1 to 50GB in 1GB steps (each neighboring pair of points in the plot is linearly interpolated). In the second example, a varied amount of worker nodes (1-40) is assigned a single computational job. Each job needs a distinct 10GB fraction of the file F_1 as input. Applying the above formula, we get the transfer times depicted in figure 2.

Figure 2. Transfer times.

We interpret the results as follows. Data caching by means of data-placement minimizes the limiting effects posed by large amounts of parallel data flows on throughput of network links or storage elements. Remote data access helps to avoid increased cumulative latency due to transfer of redundant file fractions.

Furthermore, remote data access is beneficial for the following reasons:

- The data is transferred asynchronously with respect to the job execution. Thus, the resource utilization is maximized.
- Such access methodology can help to bypass limited disk space and quotas imposed by virtual organizations by streaming data in chunks without persisting it.
- Remote data access has a smaller coordination overhead.

2.2 Observations in production

This section presents experiments conducted in the production grid in order to examine the behavior of remote data access. The data access is realized based on the reliable HTTP/WebDAV protocols. First, we examine the throughput bottleneck given a virtual link between a worker node pool and a storage element. For this purpose, we submit a computational job to the worker node pool of an anonymous computing site in Italy, which we denote by *WNP_ITALY*. Every 30 minutes, 4 threads commence to concurrently read files of various sizes (100MB, 300MB, 1GB and 2GB) from an anonymous storage element located in the UK, denoted by *SE_UK*. A single thread reads a single file respectively. Once per sampling day, the job is re-submitted to a randomly selected worker node. The sampling day is divided into 2 phases of equal length. During the first phase, 10 other threads within the same job concurrently read other files of various sizes. During the second phase, only the initially mentioned threads are active. The goal is to test whether the group means of transfer times for the 4 initial files differ among the 2 sampling phases. If this is the case, then the additionally imposed traffic during the first sampling phase causes a bottleneck effect on the given virtual link. The resulting time series for the transfer times of the original files are depicted in figure 3.

Figure 3. Throughput bottleneck.

Given a single initial file and a single sampling phase, the estimated [9] average concurrent traffic (AVG_CT) is reported in table 1.

To test whether the transfer times differ significantly among sampling phases [10], we perform the non-parametric two-sided paired Wilcoxon Signed-Rank Test. Specifically, for each original file, we split the sampled transfer times into two different groups in accordance to the sampling phase. The null hypothesis of the test states that the two samples are drawn from the same population. The test results are reported in table 2. According to the test results, the null hypothesis is rejected.

Further, we investigate whether the detected bottleneck is arising on the side of the worker node or the storage element. The second experiment is performed in the context of other computing sites. The original computational job is submitted to a worker node pool in the USA, *WNP_USA*. Just like before, once per day the job is resubmitted to a random worker

Table 1. Estimated average concurrent traffic within the first experiment.

Original file size (GB)	Phase	AVG_CT (GB)
0.1	1	1.99
0.1	2	0.4
0.3	1	5.18
0.3	2	0.67
1	1	10.2
1	2	1.44
2	1	11.3
2	2	1.4

Table 2. Test results.

File	test statistic	p-value
100MB	2628	1.694e-13
300MB	2628	1.694e-13
1GB	2628	1.694e-13
2GB	2628	1.694e-13

node. Once per 30 minutes 4 threads within the job concurrently read files of various sizes (100MB, 300MB, 1GB and 2GB) from a storage element in Australia, *SE_AUSTRALIA*. A single thread reads a single respective file. However, this time there are no additional concurrent threads in the original job. Instead, at every second sampling step two computing jobs running in Australian (*WNP_AUSTRALIA*) and Italian (*WNP_ITALY*) worker node pools respectively commence to stream files of various sizes. Now the sampling phases alternate at every second time step. The resulting time series are presented in figure 4.

Figure 4. No bottleneck induced by the storage element.

Compared to the time series generated by the previous experiment, the transfer times stay relatively constant. However, an interesting observation is that the worker nodes with the MAC address prefix 18:66:DA exhibit a rather unstable behavior.

Given a single original file and a single sampling phase, the estimated average concurrent traffic is reported in table 3. In this case we differentiate between within job concurrent traffic (AVG_CT1) and cumulative concurrent traffic from the 2 other virtual links (AVG_CT2).

Table 3. Estimated average concurrent traffic within the second experiment.

Original file size (GB)	Phase	AVG_CT1 (GB)	AVG_CT2 (GB)
0.1	1	0.58	0
0.1	2	0.57	1.02
0.3	1	1.06	0
0.3	2	1.07	2.41
1	1	1.62	0
1	2	1.62	5.68
2	1	1.38	0
2	2	1.39	8.57

We have performed an analogous non-parametric two-sided paired Wilcoxon Signed-Rank Test, with the null hypothesis stating, that the transfer times of individual files do not differ among sampling phases. The results are depicted in table 4.

Table 4. Test results.

File	test statistic	p-value
100MB	5225	0.9928
300MB	5159	0.904
1GB	5943	0.1496
2GB	5166	0.915

Based on these test results the null hypothesis is clearly confirmed. Thus, we conclude, that the bottleneck detected in the first experiment originates in the network I/O of the worker nodes.

3 Conclusions and Future Work

In this paper we have described the current state of the remote data access methodology in the ATLAS data grid. Furthermore, we have motivated its use for inter-computing site transfers, which is currently not deployed. The presented empirical experiments show that the I/O bottlenecks for remote data access tend to be exclusively on the side of worker nodes (given practical workloads). Finally, we have detected the unstable throughput of certain network interface cards.

Currently we are working on the following aspects as part of the future work:

- Mathematical formalization and forecasting model for network throughput of all the data access profiles (remote data access, data placement and stage-in).
- Incorporation of the throughput formalization in a simulation model.
- Evolutionary optimization of hybrid data access profiles using memetic algorithms.

References

[1] ATLAS Collaboration, 2008 JINST 3 S08003.

- [2] I. Bird, Annual Review of Nuclear and Particle Science **61**, 99 (2011), <https://doi.org/10.1146/annurev-nucl-102010-130059>
- [3] I. Foster, I. Foster, C. Kesselman, E.I. Inc, *The Grid 2: Blueprint for a New Computing Infrastructure*, The Elsevier Series in Grid Computing (Elsevier Science, 2004), ISBN 9781558609334, <https://books.google.ch/books?id=8-0BofIhoU0C>
- [4] V. Garonne et al on behalf of the ATLAS Collaboration, Journal of Physics: Conference Series **513**, 042021 (2014)
- [5] G. Bernabeu, F. Martinez, E. Acción, A. Bria, M. Caubet, M. Delfino, X. Espinal, Journal of Physics: Conference Series **331**, 072003 (2011)
- [6] R. Gardner et al on behalf of the ATLAS Collaboration, Journal of Physics: Conference Series **513**, 042049 (2014)
- [7] J. Elmsheuser, R. Walker, C. Serfon, V. Garonne, S. Blunier, V. Lavorini, P. Nilsson, Journal of Physics: Conference Series **664**, 042014 (2015)
- [8] J. Kurose, K. Ross, *Computer Networking: A Top-Down Approach, Global Edition* (Pearson Education Limited, 2017), ISBN 9781292153605, <https://books.google.ch/books?id=IUh1DQAAQBAJ>
- [9] V. Begy et al on behalf of the ATLAS Collaboration, Tech. Rep. ATL-SOFT-PROC-2018-001, CERN, Geneva (2018), <https://cds.cern.ch/record/2621616>
- [10] D. Freedman, R. Pisani, R. Purves, *Statistics: Fourth International Student Edition* (W.W. Norton & Company, 2007), ISBN 9780393930436, <https://books.google.ch/books?id=mviJQgAACAAJ>