ATLAS Data Carousel

Martin Barisits[1], *Mikhail* Borodin[2], *Alessandro* Di Girolamo[1], *Johannes* Elmsheuser[3], *Dmitry* Golubkov[4], *Alexei* Klimentov[3], *Mario* Lassnig[1], *Tadashi* Maeno[3], *Rodney* Walker[5], *and Xin* Zhao[3,*]

[1]European Particle Physics Laboratory (CERN), Geneva, Switzerland
[2]University of Iowa, IA, USA
[3]Brookhaven National Laboratory, NY, USA
[4]Institute for High Energy Physics, Protvino, Russia
[5]Ludwig Maximilian University of Munich, Bavaria, Germany

Abstract. The ATLAS experiment at CERN's LHC stores detector and simulation data in raw and derived data formats across more than 150 Grid sites world-wide, currently in total about 200PB on disk and 250PB on tape. Data have different access characteristics due to various computational workflows, and can be accessed from different media, such as remote I/O, disk cache on hard disk drives or SSDs. Also, larger data centers provide the majority of offline storage capability via tape systems. For the High-Luminosity LHC (HL-LHC), the estimated data storage requirements are several factors bigger than the present forecast of available resources, based on a flat budget assumption. On the computing side, ATLAS Distributed Computing was very successful in the last years with high performance and high throughput computing integration and in using opportunistic computing resources for the Monte Carlo simulation. On the other hand, equivalent opportunistic storage does not exist. ATLAS started the Data Carousel project to increase the usage of less expensive storage, i.e. tapes or even commercial storage, so it is not limited to tape technologies exclusively. Data Carousel orchestrates data processing between workload management, data management, and storage services with the bulk data resident on offline storage. The processing is executed by staging and promptly processing a sliding window of inputs onto faster buffer storage, such that only a small percentage of input data are available at any one time. With this project, we aim to demonstrate that this is the natural way to dramatically reduce our storage cost. The first phase of the project was started in the fall of 2018 and was related to I/O tests of the sites archiving systems. Phase II now requires a tight integration of the workload and data management systems. Additionally, the Data Carousel studies the feasibility to run multiple computing workflows from tape. The project is progressing very well and the results presented in this document will be used before the LHC Run 3.

[*] email: xzhao@bnl.gov

1 Introduction

The High-Luminosity LHC [1] (HL-LHC) will begin operations in the year of 2027, with expected data volumes to increase by an order of magnitude as compared with present systems, assuming a flat budget model, as shown in Figure 1.

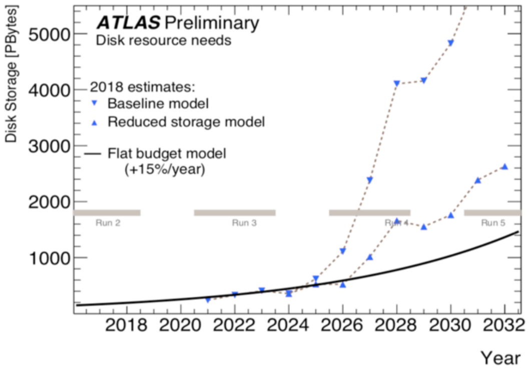


Fig. 1 Projection of the ATLAS data volume at the HL-LHC [2]

The basic considerations for the ATLAS experiment [3] to address this storage challenge at the HL-LHC are:

- "Opportunistic storage" does not exist for LHC experiments;
- Format size reduction and data compression are both long-term goals and these will require significant efforts from the software and distributed computing teams;
- The increased usage of cold, less expensive storage (currently tape) relative to disk is a natural way to dramatically reduce our storage costs. Actually, similar ideas have been explored and successfully implemented in some other scientific communities, like the Relativistic Heavy Ion Collider (RHIC) experiment [4].

ATLAS started the Data Carousel R&D project in June 2018, to study the feasibility to get inputs from tape directly, for various ATLAS workflows, such as derivation production and RAW data reprocessing.

2 Data Carousel and Objectives

2.1 Data Carousel

By Data Carousel, we mean an orchestration between workflow management (WFM), distributed data management (DDM/Rucio [5]) and tape services whereby a bulk production campaign with its inputs resident on tape is executed by staging and promptly processing a sliding window to disk buffer, such that only a small fraction of inputs are pinned on disk at any one time.

2.2 ATLAS staging process

Staging from tape is a complex process, which involves many major components in ATLAS Distributed Computing (ADC), including the production system (ProdSys2) [6], the data management system (Rucio), WLCG File Transfer Service (FTS) [7], and both storage elements (SEs) and tape systems at ATLAS Tier0 (CERN) and Tier1 sites.

In ATLAS production system, user requests are firstly injected into ProdSys2. For those requests with inputs on tape, ProdSys2 will decide how much data to stage from which tape sites, based on fair share and priority among user groups, and availability of disk and CPU resources. ProdSys2 then creates subscription rules in Rucio, which submits the requests to FTS.

FTS manages staging in two steps: the first step is to stage files from tapes to tape disk buffer (DATATAPE) on the sites (so called BringOnline operation), using the gfal2 API (Application Programming Interface) via SRM (Storage Resource Management [8]) or Xrootd [9] protocol. FTS polls for staging progress of each requested file and, once the file is staged from tape, the second step is to transfer the file to its final destination, which can be the data disk buffer (DATADISK) on the local site or a remote site.

Once the inputs become available on disk, production system will release jobs to run on the grid.

2.3 Objectives

It is expected that more tape drives will be needed to meet the data storage challenge of HL-LHC. However, it is too early to identify the target throughput out of tape, which depends on many factors, such as the luminosity of HL-LHC, the evolution of the ATLAS analysis model and computing model, changes in user requirements and user expectation. Instead of chasing a moving target, we currently focus on the tape staging process itself, and try to improve tape recall efficiency in our workflow, which is defined by the ratio of the throughput delivered to end users over the vendor-specified nominal throughput of the tape system.

After integrating the tape system into our workflow, firstly we want to make sure no or little performance penalties are introduced by the various layers to the throughput directly out of the tape system itself. To do that, we need to study the complex staging process, identify bottlenecks and improve performance in each layer. On the other hand, improving the recall throughput out of the tape system itself is another area we are working on, so called “smart writing”. By orchestrating the various components in the writing process, we try to achieve better file placement on tapes. For example, by collocating files belonging to the same dataset (therefore to be recalled together later) on tapes, we can get better recall efficiency.

In the end, whatever solutions we come up with should scale proportionally with future growth of tape capacities and tape technologies. For example, increasing the average file size to 10GB may be a perfect solution today, but it may not be as effective in the future when tape capacity grows 10 or more times bigger.

2.4 Three phases

The Data Carousel project consists of three phases:

- Phase I, evaluation of tape sites:
  Establish baseline measurements of current tape capacities, to understand tape system performance at Tier1 and Tier0 sites. More results will be discussed in Section 3.

- Phase II, integration of ProdSys2, Rucio and tape sites:
  In this phase, we will address issues found in Phase I. Also, the tape system will be integrated into the ATLAS workflow, which means a deeper integration among the workload management system (PanDA/ProdSys2), the data management system (Rucio), and Tier0 and Tier1 tape sites.
- Phase III: Run the Data Carousel with the production system, at scale, for selected ATLAS workflows.

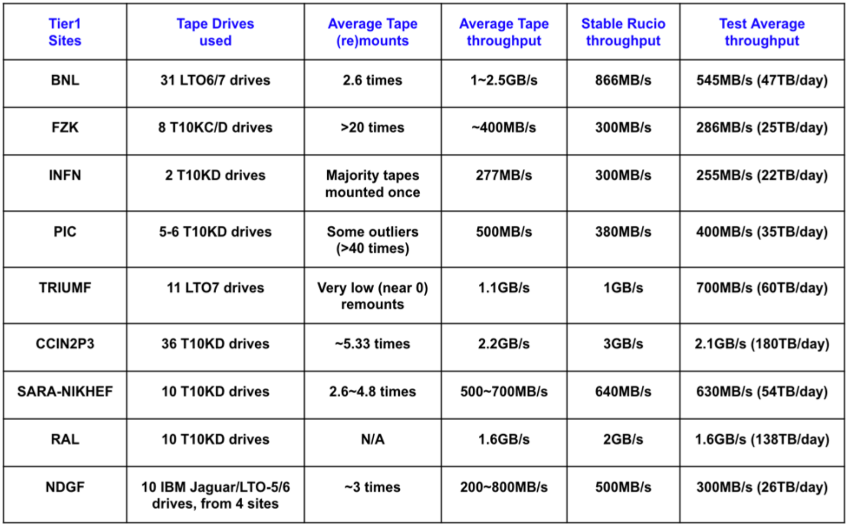
Throughout the whole process, iterative Data Carousel exercises will be conducted, sometimes combined with real production campaigns, to test improvements and reveal new bottlenecks. The current goal is to have Data Carousel in production before Run3.

3 Phase I: results and lessons learned

Over the second half of 2018, we conducted benchmark tape staging tests. CERN Tier0 and most Tier1 sites participated in this test. Results are shown in Table 1 below.

Three throughputs are used to measure performance. “Average Tape Throughput” is the throughput directly from the tape system, “Stable Rucio Throughput” is the throughput measured by Rucio (end user) over a stable run time, and “Test Average Throughput” is the ratio of the total data volume staged over the total walltime of the test. Overall throughput from Tier1s, as of November 2018, reached ~600TB/day. The CERN Tier0 conducted its own CTA (CERN Tape Archive) test, with throughput of 2GB/s.

Table 1. Phase I tape test results (Tier1s)



Tier1 Sites	Tape Drives used	Average Tape (re)mounts	Average Tape throughput	Stable Rucio throughput	Test Average throughput
BNL	31 LTO6/7 drives	2.6 times	1~2.5GB/s	866MB/s	545MB/s (47TB/day)
FZK	8 T10KC/D drives	>20 times	~400MB/s	300MB/s	286MB/s (25TB/day)
INFN	2 T10KD drives	Majority tapes mounted once	277MB/s	300MB/s	255MB/s (22TB/day)
PIC	5-6 T10KD drives	Some outliers (>40 times)	500MB/s	380MB/s	400MB/s (35TB/day)
TRIUMF	11 LTO7 drives	Very low (near 0) remounts	1.1GB/s	1GB/s	700MB/s (60TB/day)
CCIN2P3	36 T10KD drives	~5.33 times	2.2GB/s	3GB/s	2.1GB/s (180TB/day)
SARA-NIKHEF	10 T10KD drives	2.6~4.8 times	500~700MB/s	640MB/s	630MB/s (54TB/day)
RAL	10 T10KD drives	N/A	1.6GB/s	2GB/s	1.6GB/s (138TB/day)
NDGF	10 IBM Jaguar/LTO-5/6 drives, from 4 sites	~3 times	200~800MB/s	500MB/s	300MB/s (26TB/day)

One lesson learned from the Phase I tests is that the tape frontend (SE) is a bottleneck. Tape systems like bulk requests, which allows them to optimize tape mounts for efficient file retrieval. But on many sites, tape frontend SE services are overloaded when there are too many concurrent staging requests to fulfill. As a result, this limits the size of the bulk requests it can pass to the backend tape system, the number of staged files to retrieve from tape disk buffer, and this also affects the number of files to transfer to their final destination, which could be on a remote storage element.

On the other hand, the test shows that writing is important, better file placement on tapes leads to better staging performance. ATLAS files are grouped by datasets, and usually files belonging to the same datasets are recalled together. So sites that group files by datasets on tapes delivered better throughputs. Also, between two sites that have similar hardware resources and software configurations, file placement on tapes is usually the reason for the difference in performance.

4 Phase II

In Phase II, which is currently ongoing, we are addressing the issues found in Phase I, and also working on a deeper integration of the workflow management, workload management and data management systems. By the end of 2019, several rounds of Data Carousel exercises have been done, and a lot of experience was gained. Below we will go over the major ongoing activities in Phase II.

4.1 Integrate tape into ATLAS workflow

With the Data Carousel, there will be no more manual pre-staging campaigns in ATLAS. Instead, tape staging has become part of the ATLAS workflow, as shown in Figure 2.

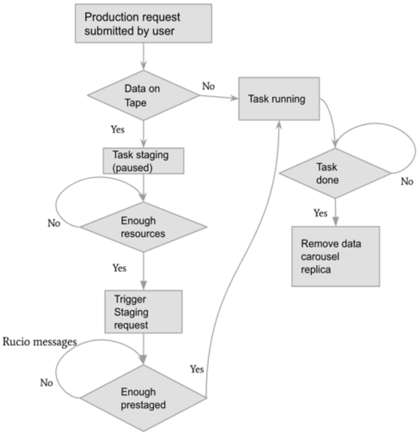


Fig. 2 : Workflow of the Data Carousel

When ProdSys2 receives new production requests, if the input files are only on tape, and there is enough space on disk, then staging requests are sent to Rucio, ProdSys2 periodically checks for staging progress, when there are enough files staged, the task is released. After the task is finished, the data is removed from disk.

So far, the basic workflow has been implemented in ProdSys2, and communication protocols between Rucio and ProdSys2 have been defined and tested.

In the future, intelligent prestaging algorithms will be developed, which will respect priorities and fair share among different user groups, and also take into account availability of computing and storage resources.

4.2 Site staging profile

One of the key features to ensure efficient usage of the tape system is to submit staging requests in bulk mode. But tests showed that a too big bulk request overloaded storage and transfer services, such as dCache and FTS, and resulted in lower overall throughput.

To manage the submission of staging requests, we define a site staging profile. In the profile, each site defines an upper limit and lower limit on the number of concurrent staging requests that site allows. New submissions will not be triggered to a site if the available staging requests is below the lower limit. And we will not send more than the upper limit number of active requests to a site at any one time. Some possible extensions to this staging profile are under discussion. For example, a time delay may be added in between submissions, to help reduce tape rewinds for some sites.

4.3 Smart writing for efficient reading

As our Phase I test shows, we should pay attention to writing to tapes, as it sets the tone for reading.

Figure 3 is a simulation result, which shows how the tape read efficiency changes with different file sizes and how much data is read per tape mount. The more files are read per tape mount and/or the bigger the files, the better the efficiency. And a contiguous block of files, if read together, is equivalent to a "big fat file", as far as read efficiency is concerned. (The simulation is done with 15 TB tape size, 10s seek time, 60s mount time and 350MB/s read speed.)

As we study smart writing, one basic assumption here is that ATLAS recalls files in unit of datasets. A dataset is much smaller than a file family, which is the default grouping mechanism in many tape systems. Also, there can be tens of thousands of datasets, so we need a finer grained grouping mechanism than file family.

Creating bigger files is another option. As shown in Figure 3, 10GB files for a 15TB tape can easily lead to a read efficiency above 75%. The concern here is that, as the tape technology evolves, tape capacity will grow bigger, hence a proper file size today will not be as efficient in the future. Therefore, we now focus on co-locating files from the same dataset on tapes, a.k.a. creating contiguous file blocks. We have started discussion with storage experts, like the dCache team, on mechanisms of grouping files by dataset when writing to tapes. Some new features from tape system vendors also look promising.

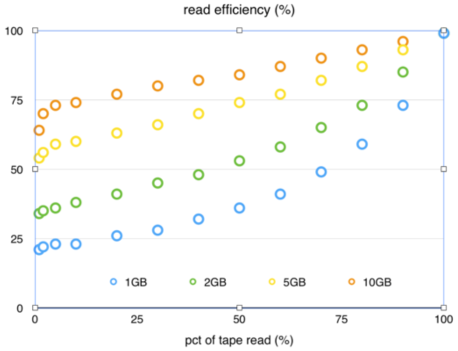


Fig. 3: Simulation of tape read efficiency (courtesy of Luc Goossens from CERN)

4.4 Release of tasks and jobs

One observation from our exercise is the slow ramp-up of running jobs when inputs are from tape. ATLAS jobs are released at task level, each task contains up to thousands of jobs. In order to promptly process staged files, we release tasks when their datasets are 70% staged. But we ended up getting slower ramp-up time in filling all the available CPU slots. The reason is that, when a task is released, those jobs whose inputs are still to be staged from tape wait in “assigned” state, and they quickly hit the cap of the number of jobs allowed in a certain state, which prevents new jobs from being released, even though their inputs are already on disk.

To address this issue, we will leverage orchestration with iDDS (intelligent Data Delivery Service). iDDS [10] is a R&D project in ATLAS towards the HL-LHC, its inter-service messaging will allow production system to monitor staging progress at file level instead of at task level, making it possible to quickly release jobs whose inputs are already staged on disk.

5 Future plans

We will continue to move forward on the various areas in Phase II, as described in Section 4. In particular, we will work closely with various service providers, such as the dCache and FTS teams, to improve on the scalability of the services, and also explore ways to increase tape recall efficiency, starting with smart writing mechanisms.

Several Data Carousel exercises have been planned as well, including much bigger scale reprocessing campaigns, derivation campaigns with inputs from tape, and technical exercises with other R&D projects like iDDS. In Run3, we expect major campaigns requesting data from tape will run under Data Carousel mode, while we continue to improve tape recall efficiency and grow tape capacity towards the needs of the HL-LHC.

Acknowledgements

We would like to thank contributions from our ATLAS Distributed Computing colleagues, ATLAS sites and CERN Tier0 operations and experts. The work is funded in part by the U.S. Department of Energy, Office of Science, High Energy Physics and Advanced Scientific Computing contracts.

References

1. L. Evans et al, *LHC Machine*, JINST 3 (2008) S08001
2. ATLAS Experiment Computing and Software – Public Results. https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults
3. ATLAS collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST 3 (2008) S08003
4. D. Yu and J. Lauret, *Tape Storage Optimization at BNL,* J. Phys.: Conf. Ser. 331, 042045 (2011)
5. M. Barisitis et al, *Rucio: Scientific Data Management*, Comput. Softw. Big Sci. (2019) 3: 11
6. F. H. Barreiro et al, *The ATLAS Production System Evolution: New Data Processing and Analysis Paradigm for the LHC Run2 and High-Luminosity*, J. Phys.: Conf. Ser. 898, 052016 (2017)
7. A A Ayllon et al, *FTS3: New Data Movement Service for WLCG,* J. Phys.: Conf. Ser. 513, 032081 (2014)

8. A. Shoshani, A. Sim and J. Gu, *Storage Resource Managers: Middleware Components for Grid Storage.* Proceedings of the 9th IEEE Symposium on Mass Storage Systems (MSS'02), (2002)
9. XRootD project, https://xrootd.slac.stanford.edu
10. B. Bockelman et al, *Event Streaming Service for ATALS Event Processing.* Proceedings of the 24th International Conference on Computing in High Energy & Nuclear Physics (2019)