ATLAS Data Carousel

Xin Zhao, Alexei Klimentov (BNL)
On behalf of the ATLAS Data Carousel Team

GDB, November 20th, 2019

Team Effort ---

- *workflow management team(WFM)*
- *distributed data management team (DDM/Rucio)*
- *distributed production and analysis team (DPAs)*
- *operations team(Ops)*
- *monitoring team*
- *ATLAS distributed computing(ADC) coordinators and experts*
- *CERN T0 and all T1s storage and tape experts.*

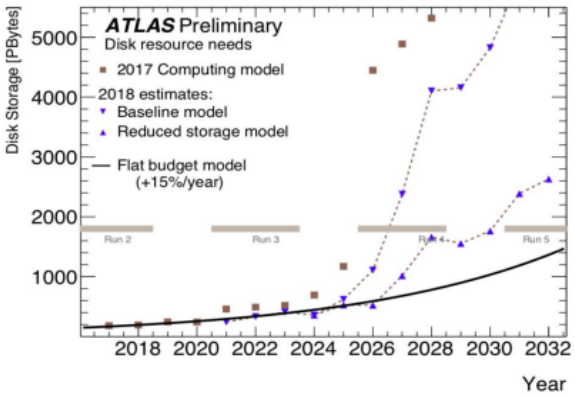
Outline

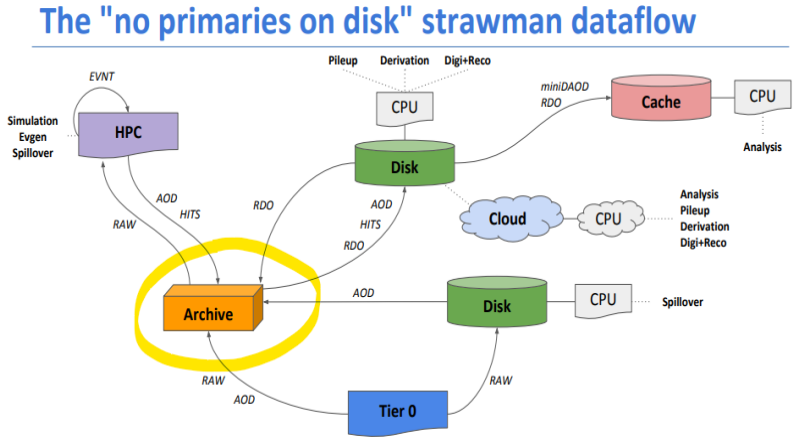
- Overview
- Phase I and its results
- Phase II ongoing activities
- Next steps

** Previous report at GDB, on Nov 2018 :* https://indico.cern.ch/event/651359/

Data Carousel: Why ?

- Facing the data storage challenge of HL-LHC, ATLAS is moving to a archive (tape) driven strawman dataflow model





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

Data Carousel : What ?

- By ‘data carousel’ ,we mean an orchestration between workflow management (WFMS), data management (DDM/Rucio) and tape services whereby a bulk production campaign with its inputs resident on tape, is executed by staging and promptly processing a sliding window of X% (5%?, 10%?) of inputs onto buffer disk, such that only ~ X% of inputs are pinned on disk at any one time.

  - “sliding window” :
    - Available disk space, which holds partial ***requested*** data by users. Not sliding along tapes
    - As data rotates between disk buffer and tapes, it presents different data to users as time goes on

Data Carousel : target tape throughput ?

- No pre-set target on tape throughput
  - A moving target … many factors in play here: luminosities, evolution of analysis model, evolution of computing model etc
- Instead, we focus on how to ***efficiently*** use the ***available*** tape capacities, ie. focus on the staging process itself
  - Introduce no or little performance penalty to tape throughput, after integrating tape into our workflow
    - Tape staging is a complex process, touches many layers in ATLAS Distributed Computing (ADC), ProdSys2, Rucio, FTS, SE, and tape system
  - Improve efficiency and throughput of tape systems itself, by orchestrating the various components in the whole system stack, starting from better organization of writing to tapes
  - Solutions should scale proportionally with future growth of tape capacities and tape technology

We are not asking for more tape drives, if our recall efficiency is only 20% !

Data Carousel: Three Phases

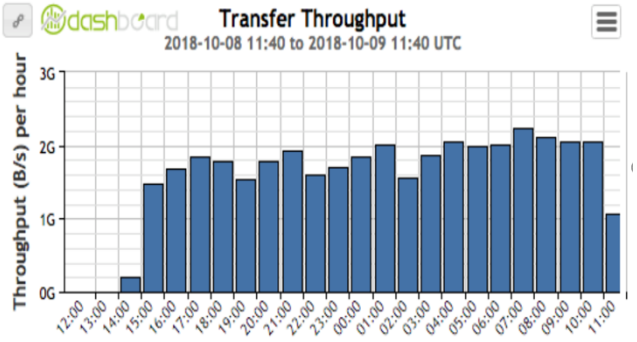
- Phase I : Tape Sites Evaluation
  - Conduct benchmark tape staging tests, understand tape system performance at all tape sites
- Phase II : ProdSys2/Rucio/Facilities integration
  - Address issues found in phase 1
  - Deeper integration between workload and data management systems (PanDA/PS2/Rucio), plus facilities
- Phase III
  - Integrate with production system and run production, at scale, for selected workflows
  - Address it in cold/hot storage context

Throughout the whole process, iterative data carousel exercises will be conducted, sometimes combined with real production campaigns, to test our improvements and reveal new bottlenecks.

Goal : to have data carousel in production before Run3

Phase I : done

- Established baseline measurement of current tape capacities
- Most T1s, plus CERN, participated
- Overall throughput from T1s (as of Nov, 2018) reached ~600TB/day
- CERN conducted its own CTA (CERN Tape Archive) test, reached ~2GB/s throughput



CERN CTA (CERN Tape Archive) test

Tape Test : Throughput

Site	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)

* Average Tape Throughput: throughput directly from local site tape monitoring
* Stable Rucio Throughput: from rucio dashboard, over a “stable” run time
* Test Average Throughput: total volume staged / total walltime of the test

Phase I : lessons learned

- Tape frontend --- limiting factor for fully utilizing tape capacities
  - Limiting number of incoming staging requests
  - Limiting number of staging requests to pass to backend tape
  - Limiting number of files to retrieve from tape disk buffer
  - Limiting number of files to transfer to the final destination

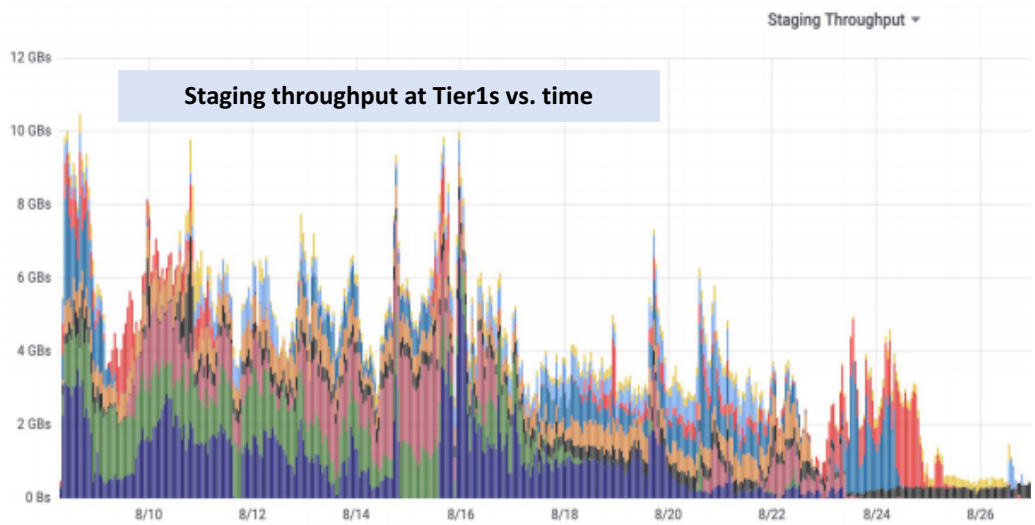
- Writing is important --- write in the way to read back later
  - Good throughput seen from sites who organize writing to tape (especially in case of grouping files on tape by datasets)
  - Usually the reason for performance difference between two sites that have similar hardware and software setup

Phase II : ongoing

- Deeper integration of workflow/workload management (ProdSys2/JEDI/PanDA) and distributed data management (DDM/Rucio) systems, plus facilities
- Two rounds of data carousel exercises done so far in phase II, the second one was combined with real production campaign (2018 RAW RPVLL reprocessing)
  - A lot of experience gained.

2018 RAW RPVLL reprocessing

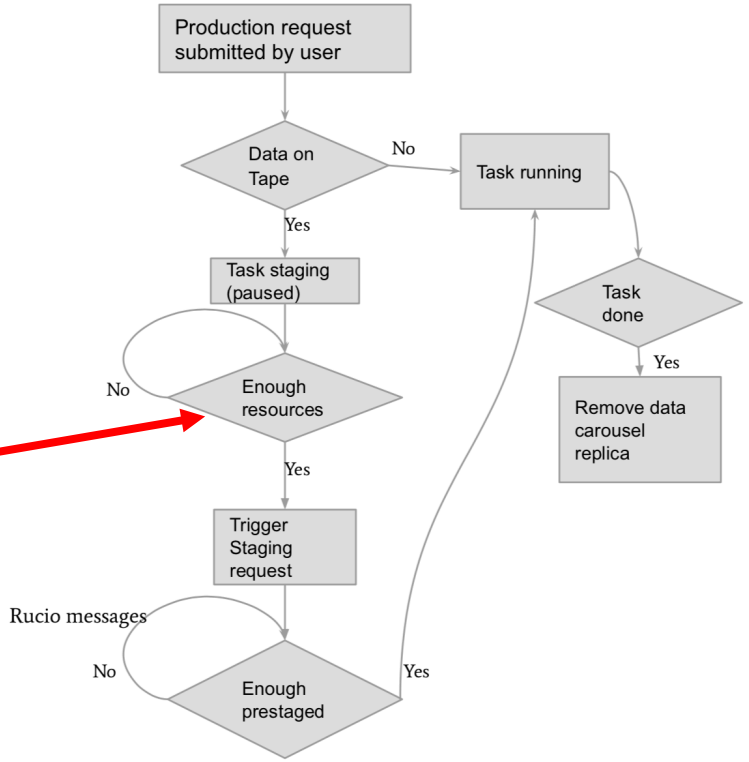
- Data carousel model used, eight T1s (INFN was in downtime) and T0 tape systems participated
- 238 datasets staged from tape. 6.9PB, 3.1M files, 6.4B events
- Average file size ~2GB/s



- Currently, AOD(Analysis Object Data) access in ATLAS from disk, is 100PB/month

Phase II : data carousel model

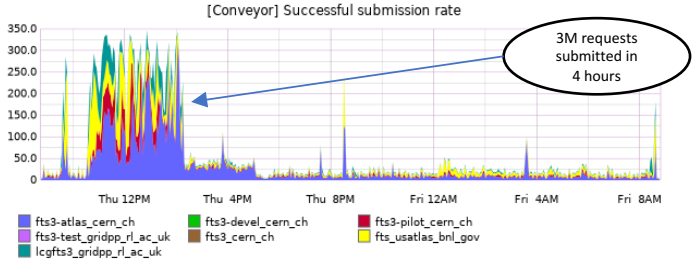
- Integrate tape into ATLAS workflow
  - No more manual pre-staging campaign
  - Define and establish communication protocols among Rucio, ProdSys2 and JEDI
  - Algorithm development for intelligent prestaging
    - Respect priorities, shares, availability of computing and storage resources…
    - Decide the "sliding window" (see page 5 on what is data carousel)

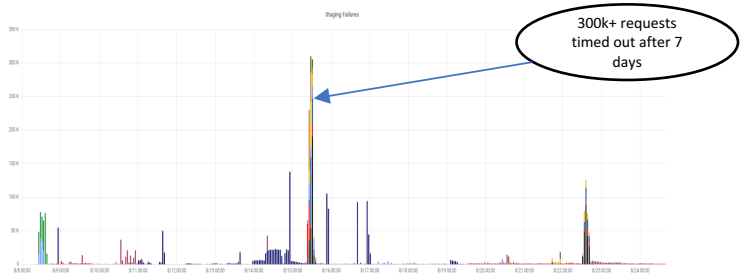


Phase II : submission of staging requests

Q: When given 500k staging requests, how T1s/T0 want them to be submitted to their sites?

- Bulk mode !!
  - Key to ensure efficient use of tape capacity, not to introduce significant performance penalty to tape systems
- We did just that in our last exercise. BTW, maybe too much….
  - Bumpy staging → lower overall throughput
    - FTS overloaded, tape frontend (dCache pools) crashed, files pin lifetime vs purge policy…





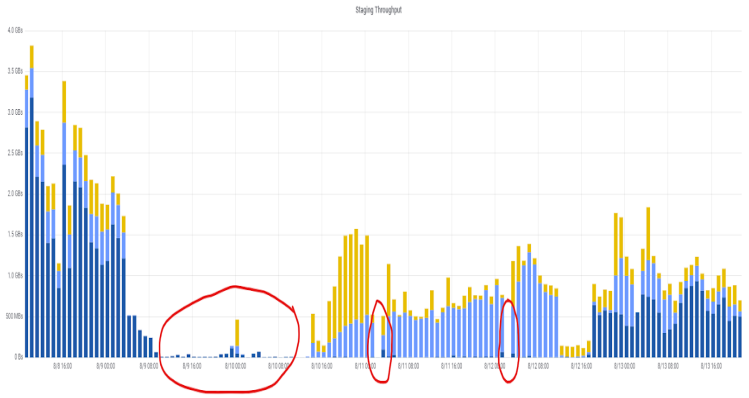
Staging Failures vs. time

- Site staging profile
  - Currently, we follow the upper/lower limits on the number of concurrent staging requests, defined by each sites (2500~300k)
    - We won't send staging requests to a site if the accumulated requests are below the lower limit (e.g. 5k). And we won't submit more than the upper limit number of active staging requests to a site at any one time
  - Possible extension under discussion
    - Add time delay in between each submission bunches, wait till the active requests drop below a threshold (50%?)

Phase II : staging rate > transfer rate ??

- Staged files are purged from disk buffer (DATATAPE), before they can be transferred to the final destination
  - Staging rate by site: 300MB/s ~ 2GB/s, way below any limits of disk-disk transfer

- FTS issues:
  - Bulk submission of staging requests (1.5M+ in 4 hours) to single FTS instance, caused FTS scheduler degradation. Overloaded FTS DB slows down submission of transfer commands
  - Purged files increased transfer failure, which in turn triggered FTS optimizer to throttle down the number of parallel transfer limits on the FTS links to minimum (2)
  - FTS team has plan to tackle all the above issues

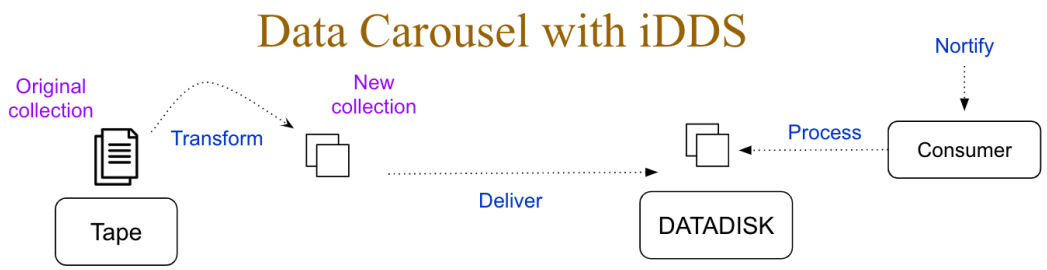


Staging throughput from three sites (colored) over time

- Tape frontend (dCache) issues
  - Can't handle the bulk size, pools crashed, slow I/O nodes caused higher failure rate, which triggered FTS optimizer to reduce link limit …
  - not new, seen in Phase I, will continue to work on them

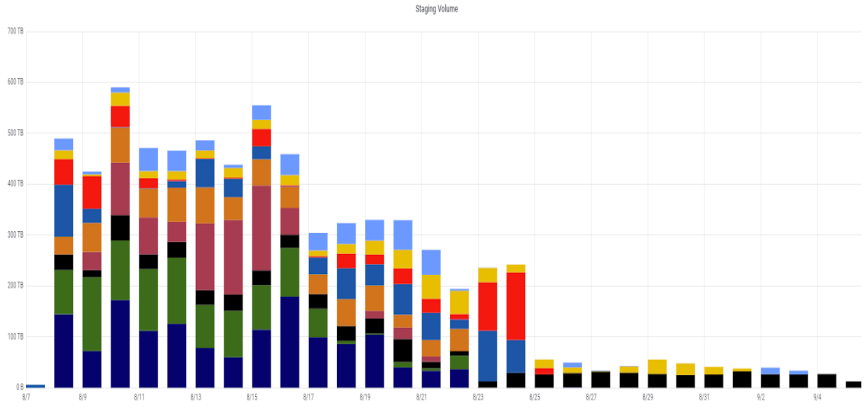
Phase II : tasks/jobs release

- Slow ramp-up of running jobs
  - ATLAS jobs are released at task level. In order to promptly process staged files, we started the campaign by releasing tasks when its datasets are 70% staged. But ended up getting slower ramp-up in filling available CPU slots.
  - Those jobs in the task, whose inputs are still on tapes, will have to wait in "assigned" state, quickly hit the cap of number of jobs in certain allowed states. This prevents new jobs to be released, even though their inputs are already staged.
- One possible solution will be orchestration by iDDS (intelligent Data Delivery Service) with inter-service messaging

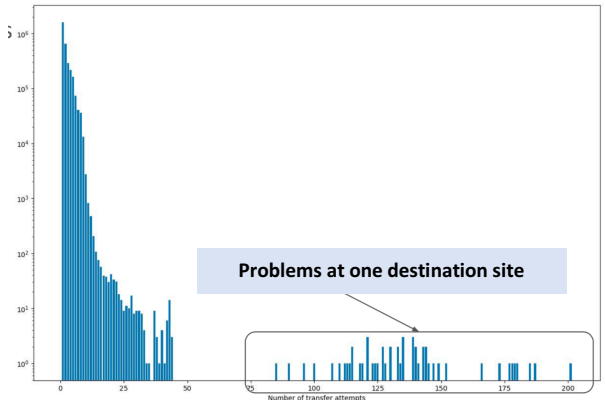


Phase II : tail effect

- Sites that took significant longer time to stage all files
- Long delay between 90% and 100% completion, which happened to many sites
  - FTS issue as mentioned above
  - Problem at the destination (took up to 200 attempts to transfer a file)
  - Rucio and ProdSys2 tuning



Overall staging throughput (TB/day) over time

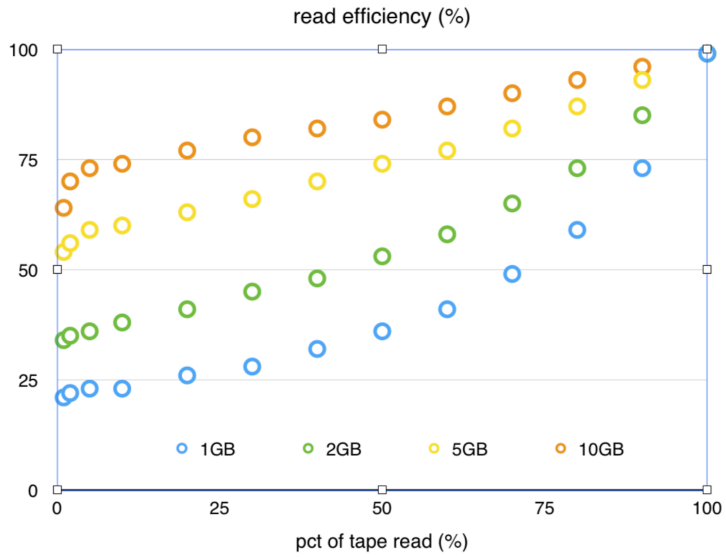


Number of transfer attempts

Phase II : smart writing for efficient reading

Writing is important, setting the tone for reading

- Simulation of tape read efficiency (see plot)
  - Prefer bigger file or bigger contiguous file block
  - Prefer more files to read per tape mount
- ATLAS recalls files by datasets
- Our options
  - Tape families --- too high of a layer than datasets, won't help much
  - Bigger files
    - Zip small output files before writing to tape.
    - Target 10GB
  - Co-locating files from the same dataset on tape
    - Since they will be recalled together, equivalent to "bigger fat file"
    - We have site that put all files of a dataset on one tape (or 1+ for bigger dataset). Reach almost stream reading speed of a tape drive per tape mount
  - Rucio will pass meta-info about files grouping to FTS, so sites will have hints about how files should be grouped when writing to tape.



(Plot is courtesy of Luc Goossens (CERN))

Next Steps

- Continue to work on the various areas as planned
  - Plan a face-to-face meeting with all relevant parties, including dCache, FTS and site tape experts, during the December ATLAS Software & Computing week at CERN, to tackle tape writing and tape frontend issues

- Iterative data carousel exercises
  - Technical exercises with one or two sites
  - Derivation with AOD from tape
  - New real production reprocessing campaigns
  - Collaborative exercise with other R&D projects (e.g. iDDS)