Job optimization in ATLAS TAG-based distributed analysis

M Mambelli[3], J Cranshaw[1], R Gardner[3], T Maeno[2], D Malon[1] and M Novak[2]

1. Argonne National Laboratory, Argonne, IL 60439, USA
2. Brookhaven National Laboratory, Brookhaven, NY 10000, USA
3. University of Chicago, 5640 S Ellis, Chicago, IL 60637, USA

Email: marco@hep.uchicago.edu

Abstract. The ATLAS experiment is projected to collect over one billion events/year during the first few years of operation. The efficient selection of events for various physics analyses across all appropriate samples presents a significant technical challenge. ATLAS computing infrastructure leverages the Grid to tackle the analysis across large samples by organizing data into a hierarchical structure and exploiting distributed computing to churn through the computations. This includes events at different stages of processing: RAW, ESD (Event Summary Data), AOD (Analysis Object Data), DPD (Derived Physics Data). Event Level Metadata Tags (TAGs) contain information about each event stored using multiple technologies accessible by POOL and various web services. This allows users to apply selection cuts on quantities of interest across the entire sample to compile a subset of events that are appropriate for their analysis. This paper describes new methods for organizing jobs using the TAGs criteria to analyze ATLAS data. It further compares different access patterns to the event data and explores ways to partition the workload for event selection and analysis. Here analysis is defined as a broader set of event processing tasks including event selection and reduction operations ("skimming", "slimming" and "thinning") as well as DPD making. Specifically it compares analysis with direct access to the events (AOD and ESD data) to access mediated by different TAG-based event selections. We then compare different ways of splitting the processing to maximize performance.

1. Introduction

The Large Hadron Collider (LHC) [1] at the CERN laboratory, near Geneva, Switzerland, is a proton-proton collider with center of mass energy of 14 TeV. Over the next 10 years this facility will provide sufficient collisions to yield sensitive tests of the Standard Model and its various extensions such as SUSY. Two general purpose detectors, ATLAS [2] and CMS [2], as well as two special purpose detectors, ALICE [2] and LHCb [2], have been built to measure the properties of these collisions which can then be used to calculate physics results.

The amount of data produced by the detectors is large both because of the number of channels and because of the event rate. At ATLAS the raw data from the detector corresponds to approximately 2 MB per event with an event rate of 200-500 Hz. The raw data will then go through a standard series of reconstruction and particle definition steps, which will yield Event Summary Data (ESD), Analysis Object Data (AOD) and Derived Physics Data (DPD). Section 2 describes these data products, but exact definitions and uses are beyond the scope of this paper and are discussed in other documents such as the ATLAS Computing Technical Design Report [3]. The estimated total data production is

O(10 PB/yr) and thus requires a navigational infrastructure coupled with a distributed parallel processing system.

This paper is divided into three sections. Section 2 describes the standard ATLAS data products. In Section 3 is an analysis of ATLAS data access patterns and a study of several modifications to different components of ATLAS software in order to improve analysis. Finally, Section 4 introduces further optimizations of the system and measures with the resulting performance improvements.

2. ATLAS data management and TAG information

ATLAS facilities are structured in a hierarchical manner. The Tier 0 computing facility is adjacent to the experiment at CERN. Tier 1 facilities are distributed among various national institutes and laboratories, and Tier 2 facilities are smaller regional computing facilities located either at universities or other research institutes. Even smaller-scale computing resources, so-called Tier 3 facilities, will be used for final-stage analysis over highly summarized data products. The large amount of data produced by ATLAS will require different levels of processing across these facilities. Successive refinements of data needed for analysis will produce progressively smaller filtered samples tailored for certain physics requirements. Generally, smaller and refined samples will have more replicas far from the Tier 0. The software tools to move the data and the ones to process it have been tailored appropriate to the scale at each Tier. For example, data is moved across Tier 0 through Tier 2 using automatic replication and subscription services. Client tools may be used to copy or access datasets within a Tier or, for smaller sized datasets, between facilities. Datasets are comprised of one or more files in various formats. Centrally managed production files are processed with the Athena framework [4] and are stored using the LCG POOL [5] as persistency framework. For local analysis files ROOT Trees [6] are often used which are easy to handle if there are not too many elements.

2.1. ATLAS data formats

The ATLAS detector produces data in units of events. Events that are selected by the triggers are written to output streams in RAW data format. Monte Carlo productions use the same event structure as detector data. The same event (2MB in size) may belong to more than one stream. ATLAS raw data is processed into reconstructed data objects that are then repackaged and distributed to facilities according to task. The ESD (Event Summary Data, 1 MB/event) is used primarily for fast reprocessing and detector studies. It contains reconstructed data with calorimeter, tracking, and trigger information. They are initially produced at the Tier 0 and distributed to the Tier 1 centers. The AOD (Analysis Object Data, 0.2 MB/event) is the first data product intended for general analysis. It contains reconstructed particles such as electrons, hadronic jets, etc. AOD data are produced at the Tier 0 and Tier 1 facilities and then distributed to the Tier 2 centers for subsequent analysis with Athena, ROOT or other tools. TAG (1 KB/event) contains event level metadata such as trigger information and is distributed to all Tier 2 centers, allowing users to quickly present select input events within a stream for their job inputs.

An additional class of formats has been specified by ATLAS [7]: Derived Physics Data (DPD). Primary DPD (D1PD) is a smaller version of the AOD. Secondary DPD (D2PD) is an augmented version of D1PD with calculated and derived quantities according to the requirements of physics or performance groups. Tertiary DPD (D3PD) datasets are typically based on ROOT Trees and are compact enough for local storage and quick iterations required for histogram making and statistical fitting. Further filtering and streaming will be applied to the production of primary DPD.

2.2. ATLAS data management

Both raw and processed event data are stored in files. In order to distribute this data ATLAS has deployed a distributed data management infrastructure (DDM) which uses an ATLAS-developed software component Don Quijote (DQ2) [8]. DQ2 supports grouping of files into datasets and containers for efficient transfers across the DDM infrastructure. A dataset is a set of related files that usually are produced or processed in the same location. A container is a set of datasets and usually corresponds to a physics dataset, data that shares the same meaning for ATLAS. To trigger data movement operators register subscriptions to the desired datasets to specific site-locations (i.e. Tier centers) in the infrastructure. The DDM then tracks where copies of the datasets exist and is then free to use internal optimizations to deliver the data, either already available or as soon as it becomes available, to the site that subscribed to it. Any service or client that needs to access the global ATLAS data store uses the DDM for location information about datasets. A central database in the DDM system knows the location of all replicas (complete or partial) of every dataset used by ATLAS, and additional (distributed) catalogs store site-level information about file availability at sites.

2.3. Event tagging and metadata

TAG metadata [9] consists of name-value pairs, and while stored separately from the event data, they contain navigational references back to the event data for the AOD, ESD, and RAW formats. The TAG uses the LCG POOL Collections package to provide a method to navigate to the events within the files in the data store. A set of utilities for accessing the collection data is also provided, although ATLAS has also extended these for ATLAS-specific applications. LCG POOL Collections also provide for multiple persistent storage mechanisms. In our case the two mechanisms of interest are relational databases and ROOT files. In POOL, references to events contain the file ID, object ID, and position within the file. For the TAG the object ID is always the Data Header for the event. These tools support multiple data transformations both with and without selection:

- AOD -> AOD-based on TAG (skim off a subset of events)
- TAG -> TAG-based on TAG (store a selection for later use)
- TAG -> new TAG with new attributes (store a selection with new attributes for later selection)
- AOD -> new TAG with new attributes

Combinations of these transformations are also possible.

The TAG data are written to files during AOD and DPD production and packaged and distributed to remote facilities like other datasets. The data are also loaded into a central relational database for browsing by users. This database may be replicated to other facilities. Additionally, TAG databases may be built using the distributed TAG datasets directly. The metadata in the TAG falls into these categories:

- Event ID (run number, event number, etc.)
- Quality (good calorimetric information, track quality, particle ID, etc.)
- Trigger (which triggers fired)
- Physics (number of electrons, missing Et, etc.)

A set of event-wise attributes was chosen constituting the TAG content in a review in early 2006 [10]. The content has since been further refined based on new use cases.

3. Data access in ATLAS analysis

A recent ATLAS report [11] describes analysis use cases and summarizes four major activities:

- Data distribution chain in the steady state: a steady flow starting with the RAW data produced at the detector and ending with AODs and D1PD at the Tier 1 centers
- Dataset creation: production of D2PD or D3PD datasets for general ATLAS consumption, executed at Tier 2 centers
- Monte Carlo production: simulation of the detector is driven event generation at the Tier 1 centers with detector simulation executed mainly at Tier 2s
- Chaotic data analysis: customized analyses performed by users on DPD and possibly the (larger) AOD datasets

The ability to efficiently analyze samples for customized analysis frequently requires reducing the datasets using specialized criteria. Forms of data reduction fall into the following categories:

- Skimming: The extraction of events of interest from the data store
- Slimming: Storing a subset of the data classes rather than a full copy of the event

- Thinning: Limiting the objects based on usefulness for their physics

Here physicists will inspect the data for irregularities, compare it to Monte-Carlo data, isolate signal from background and verify various tests. All of this will be likely repeated multiple times applying various weighting factors and/or changing the selection.

Analysis jobs usually include iteration on ROOT Trees or running multivariate techniques over local datasets, but a majority of the processing effort and resources will involve the production of D3PD or ROOT Trees and histograms starting from ESD, AOD or DPD datasets. These are the use cases targeted by the optimizations discussed in this paper and described in more detail in this section. Their execution will follow a sequence of similar steps starting with the selection of the events and ending with the retrieval of the analysis results.

3.1. Event selection

TAG information can be accessed using POOL command line utilities, the Athena framework, or the interactive Web frontend ELSSI [12]. The selection process uses the information stored in the event attributes to select events of interest and can be done interactively, from a script, or from within an analysis job. Depending on the tool, a user or algorithm can list available attributes, count the events matching a selection, plot attribute distributions, and finally define a query that will select the desired events. The selection output is typically a ROOT file containing event metadata and navigational references (*Extraction file*).

3.2. File location

The navigational references in the *Extraction file* point to the event in each of the RAW, ESD and AOD formats. These uniquely identify the file and the position of the event (its Data Header) within the file. A POOL utility (*CollListFileGUID*) was developed to list the file unique identifiers (GUIDs) for each of the format types. This list can be used to organize the input data for analysis jobs that by policy cannot trigger wide area data transfers from within the job itself and must be accessible by Athena-supported (local) protocols. In practice this means analysis jobs must run on sites where the input files are already available, or a pre-staging job must occur in advance of the analysis job.

A JSON-based web service [13] has been developed to translate the list of GUIDs into file and dataset names, and to locate them on the Grid. Current work involves integrating this service into the ELSSI Web frontend described above. The Grid locations can be used as analysis execution sites for the selected events or sources for transfers to other sites from pre-staging jobs. Pathena, the analysis client used by the Panda production system [14], has been modified to use the TAG *Extraction file* to resolve references locally at the analysis site and to put them into the format expected by Athena (*PoolFileCatalog.xml*).

3.3. Analysis execution

There are two basic modes for TAG-based analysis in ATLAS. The first involves event selection in advance of job submission; the second involves event selection during job execution. Each introduces important execution optimization choices in terms of job splitting and file access.

In the first mode, knowledge of the referenced events and files is known at job submission time and therefore allows one to organize individual jobs so as to minimize the overall execution time. Files in the input dataset containing no selected events can be skipped. Additionally, improvements to Pathena included using the information stored in the *Extraction file* (i.e. the event selection) to optimize job splitting by optionally varying the number of events or files per job. This allows experimenting to get the best throughput through the Grid scheduling systems.

The performance of both modes depends critically on the access method for input files. Note that job input files include both the TAG files (either the *Extraction file* or a file from the TAG dataset) and the event input files (i.e. ESD or AOD formats). Data access methods vary significantly in a Grid execution environment. Files may located on disk local to the compute node, or in a local storage element, usually based on a storage system technology such as dCache [15], XROOTD [16], Castor [17], or a global file system such as GPFS, Lustre or Hadoop [18]. The critical decision is whether to copy input data to the local disk, or to read directly from the storage element via an appropriate protocol (i.e. supported by Athena). Different combinations of direct access or local-copy were tested and compared. In both cases we found it more efficient when Panda copies the TAG files to local disk. This is not surprising given the relatively small file size of TAG files. For the event data input files, it was more efficient to read the files directly rather than copy to local disk, provided the storage system was suitably configured (see Section 4). This is also not surprising since the TAG reference allows for navigation within the file, accessing and transmitting over the network only the selected events of interest.

3.4. Result retrieval

It is ATLAS policy to leave the output files on the site where each analysis job was executed. A DDM command, *dq2-get*, allows retrieval of all the output produced by the whole analysis even if it resides on different sites.

4. Analysis performance

The processing times of different analysis jobs have been measured to compare different analysis workflows and to choose the default procedure for a TAG-based analysis.

Most of the tests were performed on the ATLAS Midwest Tier 2 Center [19]. The cluster has two hundred multicore CPU servers in a PBS job queue. A gigabit Ethernet switch interconnects the nodes and a 200 TB capacity dCache storage system. Jobs were submitted locally or via the Grid using Pathena. Jobs focused on comparing different workflows and on tuning dCache performance.

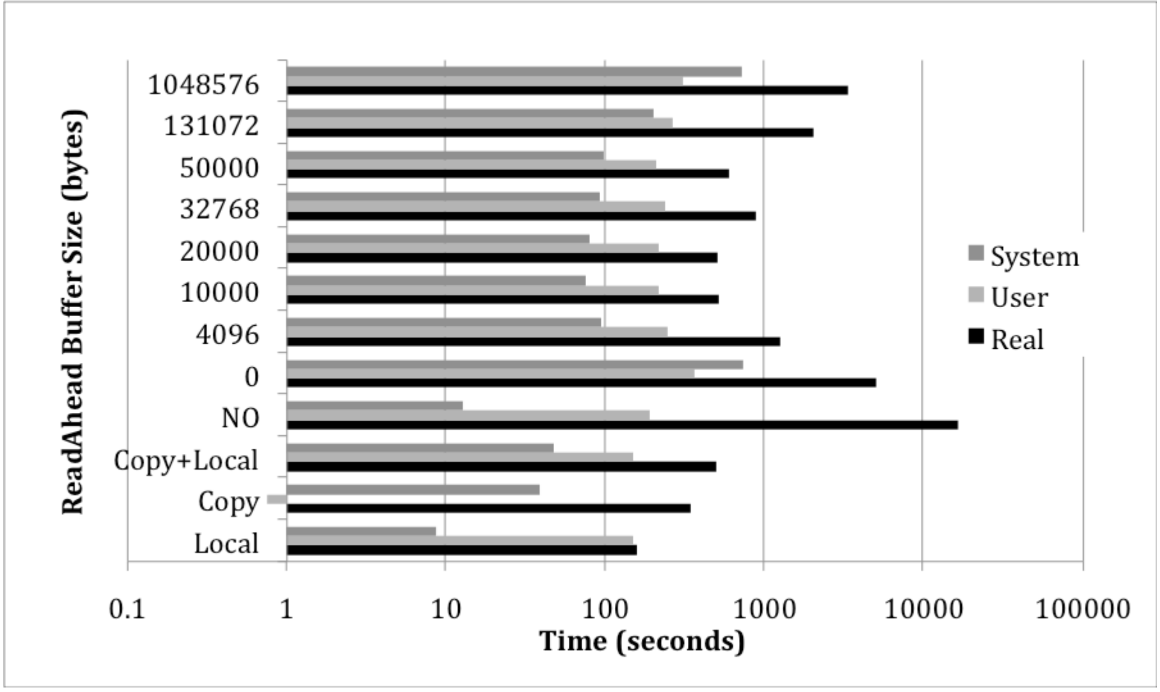

Figure 1. dCache timing for TAG-based analysis with different ReadAhead buffer configurations

Figure 1 shows the effect of dCache ReadAhead buffer in a skim job where almost all events are selected. Enabling the buffer increases file access speed (lower *Real Time*) even if it increases also the system load (*System* and *User Time*) because most of the data buffered is discarded (the efficiency was always below 3%). The length of the buffer is not really important as long as it is enabled. When accessing all the events the performance is comparable with local copy + local execution.

In a selective skim, a skim with a low percentage of selected events, direct access is much better. In fact in a selective skim (about 20% of acceptance rate) the completion time was less than half than the unselective skim charted in figure 1, where almost 100% of the events are selected.

We also found significant improvement in the overall execution time after options for job splitting at submission time were added to Pathena. Without the TAG files as input, Pathena would submit a separate job per input file over an entire dataset, resulting in some jobs which produced no output. Currently files with no selected events are skipped, reducing the number of jobs or the number of accessed files. Similarly job splitting based on the number of input events per job is more uniform when Pathena is TAG aware, and the more even numbers of events to process in each job result in a shorter completion time of the analysis.

5. Conclusions

This paper presented modifications implemented in the ATLAS software, especially Pathena and the LCG POOL utilities, to improve TAG-based data analysis. It further explained the advantages of TAG-based analysis and compared different possible access patterns to the data giving useful insight on how ATLAS jobs should interact with data storage systems. Further tests will measure the performance gain in selective skims. Developments discussed here are being integrated in ELSSI (cf. J. Cranshaw et al. [12]) to provide a streamlined solution for TAG-based data analysis in ATLAS.

6. Acknowledgements

The research was supported in part by DOE –Brookhaven National Laboratory Grant 103181.

References

[1] The Large Headron Collider Project at CERN, http://lhc.web.cern.ch/lhc/
[2] CERN LHC Experiments:
A Toroidal LHC ApparatuS (ATLAS), http://atlas.web.cern.ch/Atlas/
Compact Muon Solenoid (CMS), http://cms.cern.ch/
The Large Hadron Collider beauty experiment (LHCb), http://lhcb.web.cern.ch/lhcb/
A Large Ion Collider Experiment (ALICE), http://aliceinfo.cern.ch/
[3] ATLAS Computing TDR, CERN-LHCC-2005-022 (2005)
[4] The Athena framework, https://twiki.cern.ch/twiki/bin/view/Atlas/AthenaFramework
[5] POOL - Persistency Framework, http://lcgapp.cern.ch/project/persist
[6] R. Brun and F. Rademachers, “ROOT: an object oriented data analysis framework”, Nucl. Instrum. Meth. A 389, pp 81-86, http://root.cern.ch
[7] D. Constanzo, I. Hinchliffe, and S. Menke. Analysis model report, 2008. draft 1.4
[8] Don Quijote2 (DQ2), the ATLAS Distributed Data Management (DDM), https://twiki.cern.ch/twiki/bin/view/Atlas/DistributedDataManagement
[9] D. Costanzo, et. al., “Metadata for ATLAS”, CERN-ATL-COM-GEN-2007-001 (2007).
[10] K. Assamagan, et. al., “Report of the Event Tag Review and Recommendation Group”, ATL-COM-SOFT-2006-003 (2006).
[11] R. Brock, et. al., “U.S. ATLAS Tier 3 Task Force” report, April 2009
[12] J. Cranshaw et al. “Event Selection Services in ATLAS”, CHEP2009 (2009), these proceedings
[13] Javascript Object Notation (JSON) http://www.json.org/ and http://json-rpc.org/
[14] T. Maeno for the ATLAS collaboration, “PanDA: Distributed production and distributed analysis system for ATLAS”, J. Phys. Conf. Ser. 119 062036, 2008
[15] P. Fuhrmann et al, “dCache, a distributed data storage caching system”, CHEP2001 (2001)
[16] A. Dorigo, P. Elmer, F. Furano, A. Hanushewsky, “XROOTD - A highly scalable architecture for data access”, WSEAS – Prague (2005), http://xrootd.slac.stanford.edu/
[17] CASTOR, the CERN Advanced STORage manager: http://castor.web.cern.ch/castor/
[18] Hadoop file system and fremework: http://wiki.apache.org/hadoop/ProjectDescription
[19] ATLAS Midwest Tier2, http://www.mwt2.org/