PROCEEDINGS OF SCIENCE

ERADAT and DataCarousel systems at BNL: A tool and UI for efficient access to data on tape with fair-share policies capabilities

Jérôme LAURET[1]
Brookhaven National Laboratory
Upton, NY 11973 – USA
E-mail: jlauret@bnl.gov

Yu David
Brookhaven National Laboratory
Upton, NY 11973 - USA
E-mail: david.yu@bnl.gov

[1] Speaker

PROCEEDINGS OF SCIENCE

The BNL facility, supporting the RHIC experiments as its Tier0 center and thereafter the Atlas/LHC as a Tier1 center had to address early the issue of efficient access to data stored to Mass Storage. Random use destroys access performance to tape by causing too frequent, high latency and time consuming tape mount and dismount. Coupled with a high job throughput from multiple RHIC experiments, in the early 2000, the experimental and facility teams were lead to consider ingenuous approaches. A tape access "batch" system integrated to the production system was first developed, based on the initial OakRidge National Lab (ORNL) Batch code. In parallel, a highly customizable layer and UI known as the DataCarousel was developed in-house to provide multi-user fairshare with group and user level policies controlling the sharing of resources. The simple UI, based on a perl module, allowed to create user helper script to restore datasets on disks as well as had all the features necessary to interface with higher level storage aggregation solutions. Hence, beyond the simple access at data production level, the system was also successfully used in support of numerous data access tools such as interfacing with the Scalla/Xrootd MSS plug-in back end, similarly the dCache back end access to MSS. Today, all RHIC and Atlas experiments use a combination of the Batch system and the DataCarousel following a 10 years search for efficient use of resources. In 2005, BNL's HPSS team decided to enhance the new features such as improve the HPSS resource management, enhance the visibility of real-time staging activities, statistics of historical data for performance analysis. BNL Batch provides dynamic HPSS resource management and scheduled read job efficiently while the staging performance can still be further optimized in user level using the DataCarousel to maximize the tape staging performance (sorting by tape while preserving fair-shareness policies). In this presentation, we will present an overview of our system and development and share the findings of our efforts.

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research - ACAT 2010
Jaipur, India
February 22–27 2010

1. Introduction

When storing large amounts of data, tape can be substantially cost effective (in terms of the media cost, power consumption, and air conditioning cost), compared to modern storage technologies such as hard-disk or other data storage devices. Therefore, tape storage is still commonly used in large computer centers, primarily being used as a high capacity medium for backups and archives.

The BNL[2] data center, hosting the RHIC[3] and Atlas[4] Computing Facility (RACF), holds near 15 PB of data in tapes, serving science researchers from both RHIC and LHC/US-Atlas, operated by a system called HPSS [1], a software stack able to manage Peta bytes of data on disk and robotic tape libraries. The facility erves as the Tier0 center for RHIC and as a Tier1 center for Atlas and is equipped, amongst other hardware, of six Sun/TSK SL8500 each able to support up to 5 PB of data.

1.1 Problematic

Tape technologies and tape access are inherently sequential. As such, collaborations have put a great deal of thoughts into how their data is saved onto tapes and how to optimize data mining and data production workflows, from a production account perspective, taking into account the time sequence and ordering of files on tape. However, this simplistic approach becomes problematic if one has to produce or mine datasets from different period in time, the stochastic nature of the workflow causing an access pattern forcing tape mount/dismount to satisfy all requests. The problem is exacerbated if there is a real need for users (one to two order of magnitude more access pattern complexity) to access data on tape. Since tape storage are so much cheaper and disk buffer still limited, the tape storage system in fact is being used as a near real-time random access device. This means user may be staging (restoring) any number of files out of any random tape at any time, 24 x 7.

1.2 Technology issues overview

As we noted, tape access is sequential in nature: one may fast forward or backward but optimal access would be if one could access all files from one tape sequentially without having to dismount it ever. Tape systems are good for archiving, but not for reading because of the long latency for random accesses: whenever there are lots of tapes queued up for staging, the bottleneck is the limited number of tape drives. Furthermore, the source of tape access latencies can be indentified as (a) the time it takes to transport the tape inside the library (b) the mount time (c) the position to find/seek a file (d) the rewind and dismount time (e) the number of tape

[2] Brookhaven National Laboratory (BNL) – http://www.bnl.gov/

[3] The Relativistic Heavy Ion Collider (RHIC) is the first machine in the world capable of colliding ions as heavy as gold and the only machine in the world capable of colliding polarized protons beams.

[4] BNL is one of the Tier1 facilities for the LHC/Atlas project.

marks. An efficient system would need to consider all of those aspects to resolve the problem at hand.

1.3 Tools developed at BNL and timelines

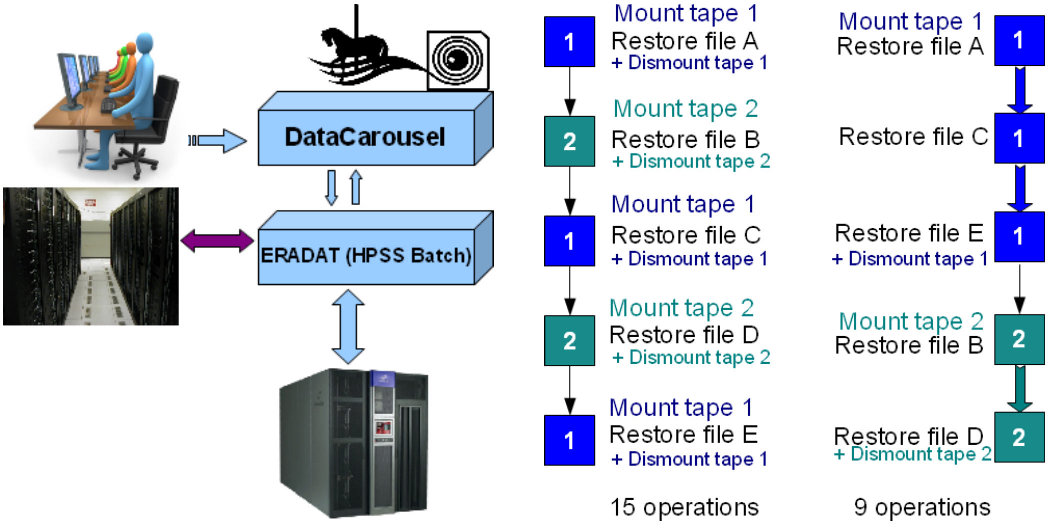

Figure 1: ERADAT and the DataCarousel relative interdependence. ERADAT sits at the lowest level, interfacing directly with the HPSS API and act as a queuing system. The facility production jobs may directly interact with it. Users or high level services typically interact with the DataCarousel, implementing advanced features such as fair-share and resource handling policies. The system essentially allows minimization of tapes mounts/dismounts as seen and illustrated on the right hand-side panel.

In the early 2000, the experimental and facility teams were pushed to consider ingenuous approaches to retrieve files from mass storage during the data production workflows. A tape access "batch" system integrated to the production system was first developed, based on the initial OakRidge National Lab (ORNL) Batch code. The system could be described as a DAG, whereas the job would wait for its requested file before moving to the processing stage but not before a load condition would be reached. Since data production were usually as a sequence of jobs ordered in time (hence more or less accessing the same tapes at any given time), this level of optimization was sufficient. In parallel, a highly customizable layer and UI known as the DataCarousel was developed in-house to provide multi-user fair-share with group and user level policies controlling the sharing of resources. The tool was delivered in 2001 and has showed great success in allowing users and working group to restore in a coordinated manner datasets from mass storage on live disks depending on their physics needs.

As the increasing demand of staging files from tapes as well as the new tape drive technologies were added to the system, the initial version of Batch could no longer handle the diversity of hardware as well as new requirements suggested from users not easily handled by the DataCarousel alone. Such features included treating the biggest request queue first (load the tape with most files requests first). In 2005, the DataCarousel was also used as a back-end to Scalla/Xrootd by the STAR experiment [2] and the system drove even more requirements such as request expiration and fine grain control at class of service level. By the end of 2005, the

BNL RACF HPSS team worked on enhancing the "HPSS Batch" system, in order to provide better performance and resource management. In 2010, reaching full and demonstrated maturity and stability over years of production mode operation, the new Batch was renamed to ERADAT, standing for *E*fficient *R*etrieval and *A*ccess to *D*ata *A*rchived on *T*ape. The overall relation between the DataCarousel and ERADAT is illustrated in Figure 1. The right side panel illustrated a simple on how, by ordering requests, one can reduce fifteen operations to nine operations and saves overheads. In the next sections, we will describe the systems in more details.

2. ERADAT and the DataCarousel

2.1 ERADAT – Restore from tape (staging) optimization

There may thousands of staging requests from all different users, in order to optimize the tape reading performance, we need to allocate the resources effectively and provide resource reservation control.

2.1.1 Resource reservation control

Equipments are purchased by the funding from each individual experiment - Star, Phenix and LHC/Atlas. Each experiment must have dedicated number of drives available for their users all the time. ERADAT is designed to support multi-domain, in order to guarantee the resource availability all time.

2.1.2 Resource allocation

Each experiment may have multiple groups of users, and sometimes they want to have resource reservation for the end users, ERADAT can create virtual partitions between groups – to throttle the tape-drive usage by group. The partition can be dynamically re-adjusted without interrupting any running process. It is very important that we constantly have to watch the read and write traffic, and fine tune the resource allocation so the drives can be fully utilized.

[***]

In above example: There are a total of 12 drives available for this Domain. The drives are logically divided into 4 partitions; 3 drives are reserved for writing. Group A can only use up to 2 drives. Group B can use up to 4 drives. The allocation is based on the agreement made between the groups. If Group A needs to borrow drives from Group B, we can always adjust the partition without any service interruption.

2.1.3 Efficiency differentiated by tape technology

There may have multiple types of tape-drives in the system, such as 9940B, LTO-3 and LTO4. ERADAT has a virtual PVR that is very similar to HPSS's PVR; tapes are scheduled by PVR, so the tape-drives usage can be efficiently used. Each PVR has a dedicated manager thread for scheduling task.

Also, any scheduler can be locked at any time. For example, if 9940B PVR needs to be down for maintenance, other schedulers can still continue to stage files from LTO4 tapes. All new requests/jobs for 9940B tapes will be queued in memory.

2.1.4 Tape mount control optimization

Tape has long latency for random accesses since the deck must wind an average of one-third the tape length to move from one arbitrary data block to another[i]. For example, HP LTO-4 media can do 120 MB/s native data transfer, but the other latencies are making performance drop. Typically, it takes about at least 5 seconds of delivery time, 19 seconds for mount-time, access time may take up to 62 seconds from the beginning of the tape, and it takes 124 seconds to rewind from the end of the tape[ii]. That means each tape mount requires at least 117 second + actual read time. Clearly, the tape mount is the real performance bottleneck.

To solve the performance problem, requests are sorted by tape cartridge and file position, so that all the requests on same tape can be read at once sequentially in order to reduce redundant tape mounts. The next question is which tape should be processed first? The decision is optional to the user – FIFO, or By Demand.

2.1.5 Flexible staging algorithm

Staging algorithm is optional for each group. The most popular algorithm is “By Demand”, which means ERADAT sorts the tapes by demand (AKA by popularity) – by the number of requests on each tape; the high-demanded tape has higher priority. By-demand optimization usually provides best performance in general. However the con side of this option is that it puts the less popular tape on low priority and these files may have to wait for a long time.

FIFO is the alternate staging option, which allows the user-group to use their own algorithm. If the user-group wants to prioritize their own requests, then ERADAT can process them in FIFO. A typical example is DataCarousal, it aggregates all requests from all clients (many users could be considered as separate clients) and re-order them according policies, and possibly aggregating multiple requests for the same source into one request to the mass storage.

2.1.6 Priority staging

Every request consumes resource, and it would be more efficient if we can eliminate all known errors.

ERADAT has a “cartridge” table that is being updated in near real time. ERADAT fails the request immediately under the following 3 conditions:

1. Request is invalid (bad filename or file is not in HPSS)

2. Tape is locked in HPSS
3. LSM is down

ERADAT also checks the file on disk, if the file is already on HPSS's disk cache, the file will be marked as "staged successful" immediately.

Callback feature, ERADAT calls a "callback script" that is provided by user when the file is staged. The callback script should know how to deliver the file upon available. Each file maybe handled differently and that is totally up to user's own preference since user provides the script.

2.2 Error handling

2.2.1 Retry policy

Some errors worth for retry, some are not. The retry logic is defined in ERADAT's configuration table, per user group. For example:

[****]

2.2.2 Dead jobs

Sometimes a job maybe hanging there due to some hardware problems and that is wasting resource and affecting the overall performance. ERADAT watches all running jobs and highlight any job that is taking abnormally longer time. If a job has been taking way longer time, an email will be sent to the administrator.

3. Results

There are 5 major groups of users in ERADAT, we used 2 real examples:

1. Data Mining at RHIC (STAR CRS Job Processing)
2. ESD re-processing at US-Atlas

Assumption: All calculation is based on hardware specification provided by manufactory. Actual performance maybe varies, due to other ambient factors such as tape's media condition, tape's delivery time (location), and other factors.

3.1 Data Mining at RHIC

Using default optimization option – By Demand

[***]

RHIC/STAR CRS Job Processing

3.2 ESD processing at the LHC/US-Atlas

LHC/US-Atlas is using "By Demand" optimization option.

[***]

If we take a sample of Tape #500425, 2277 requests were on this tape and they were received in 530 different time. With ERADAT's high-demand optimization, Tape #500425 were only mounted 3 times, staged 2277 files, 77 GB of data. Avg file size: 34 MB. Since the 2279 files were arrived in 530 different timestamp (bundles). If processing them in FIFO – no optimization, the tape maybe mounted 530 times, that's about average 4.3 files per mount, and how long would it take?

4. Conclusion

ERADAT was formally named "BNL Batch". BNL Batch was developed in the RHIC data processing era, it has demonstrated a great reading performance for RHIC experiment. It has now been adopted by LHC/US-Atlas helping with data processing.

References

[1] High Performance Storage System (HPSS)

[2] P. Jakl, J. lauret et al., *Grid data access on widely distributed worker nodes using Scalla and SRM*, Journal of Physics: Conference Series **119** (2008) 072019

[3] F. Baggins, *Quantum effects of the One Ring*, *JHEP* **01** (3021) 006 [`hep-th/2001033`]

[4] B. Baggins, *There and back again*, Imladris Editions, Rivendell 3018.

[5] W.A. Mozart, *Don Giovanni*, in proceedings of *Mock conference* `PoS(MC2000)002`.

[i] Wikipedia: http://en.wikipedia.org/wiki/Tape_storage

[ii] HP Support document, HP StorageWork LTO-4 Ultrium 1840 Tape Drive – Overview. http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?objectID=c01121300&lang=en&cc=us&taskId=101&prodSeriesId=3454484&prodTypeId=12169