Tape Storage Optimization at BNL

2011 J. Phys.: Conf. Ser. 331 042045

(http://iopscience.iop.org/1742-6596/331/4/042045)

Journal of Physics: Conference Series **331** (2011) 042045 doi:10.1088/1742-6596/331/4/042045

Tape Storage Optimization at BNL

David YU, Jérôme LAURET
Brookhaven National Laboratory, Upton, NY 11973 – USA

E-mail: david.yu@bnl.gov, jlauret@bnl.gov

Abstract. The BNL's RHIC and Atlas Computing Facility (RACF), is supporting the RHIC experiment as its Tier0 center and the Atlas/LHC as a Tier1 center. The RACF had to address the issue of efficient access to data stored to disk and tape storage. Randomly restoring files out of tapes destroys access performance to tape by causing too frequently, high latency and time consuming tape mount and dismount. BNL's mass storage system currently holds more than 16 PB of data on tapes, managed by HPSS. To restore files from HPSS, we make use of a scheduler software, called ERADAT. This scheduler system was originally based on a code from OakRidge National Lab, and then it was renamed to BNL Batch at 2005 after some major modifications and enhancements. The new BNL Batch, ERADAT, provides dynamic HPSS resource management, schedule jobs efficiently, enhanced visibility of real-time staging activities and advanced error handling, to maximize the tape staging performance. ERADAT is the interface between HPSS and other applications such as the DataCarousel, our home developed production system and dCache. Scalla/Xrootd MSS can also be interfaced with HPSS via DataCarousel. ERADAT has demonstrated great performance in BNL and other institute.

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, mainly as a high capacity medium for backups and archiving.

The BNL [1] data center, hosting the RHIC [2] and Atlas [3] Computing Facility (RACF) holds more than 16 PB of data on tapes, serving science researchers from both RHIC and the LHC/US-Atlas collaborations. It is operated by a system called High Performance Storage System (HPSS) [4], a software stack able to manage Peta-Bytes of data on disk and robotic tape libraries. This facility serves as the Tier0 center for RHIC and as a Tier1 center for Atlas and is equipped, amongst other hardware, of six Sun/STK 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 single production account perspective, this implies, for

example, taking into account the time sequence and ordering of files on tape when reading them back and ordering job sent to a queue system accordingly. However, this simplistic approach becomes problematic if one has to produce or mine datasets from different time period; 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(s) may be staging (restoring) any number of files out of any random tape at any time, 24 x 7.

1.2 Technology issues overview

The immediate consequences of the use of tape and sequential access is that the media is good for data archiving (multiple copies from cache to tape may occur, flushing the data out at record speed) but not very well suited for reading as restoring from tape as data may end up being spread over multiple tapes. Reading back the data may then lead to erratic tape mount and dismount depending on where the data was initially written. Furthermore, the source of tape access latencies can be identified 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 on a tape (d) the rewind and dismount time (e) the number of tape marks (or separators between blocks on the media). 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

ERADAT (*Efficient Retrieval and Access to Data Archived on Tape*) and the DataCarousel are the 2 major software developed at BNL, both of which are designed to optimize the file retrieving from tapes storage.

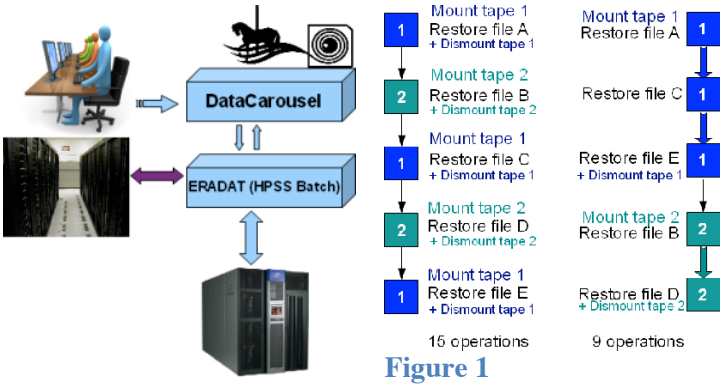


Figure 1

Figure 1 shows ERADAT and the DataCarousel relative interdependence. ERADAT sits at the lowest level, interfacing directly with the HPSS API and acts 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 and dismounts as seen and illustrated on the right hand-side panel.

2 ERADAT

The *E*fficient *R*etrieval and *A*ccess to *D*ata *A*rchived on *T*ape or ERADAT, is a file retrieval scheduler for IBM HPSS. ERADAT evolved from the Oak Ridge batch code[1] as a prototype, and then modified to fit BNL's requirements. ERADAT has evolved to include many additional features such as dynamic drive usage allocation, support for multiple projects and groups, support for multiple drive technologies (9940 and LTO-3 and LTO-4 drive series), request lifetime expiration, and multiple staging algorithm including 'by-demand' and FIFO. ERADAT also keeps

[1] To our knowledge, this project and initial code was not documented and was not published.

all transaction history for performance reporting purposes but the historical usage also helps guiding and fine-tuning the system. ERADAT collects additional data from other sources such as library controller, for cross-reference checking. ERADAT allows modifying the drive allocation on the fly via a Web-based monitoring system and control interface.

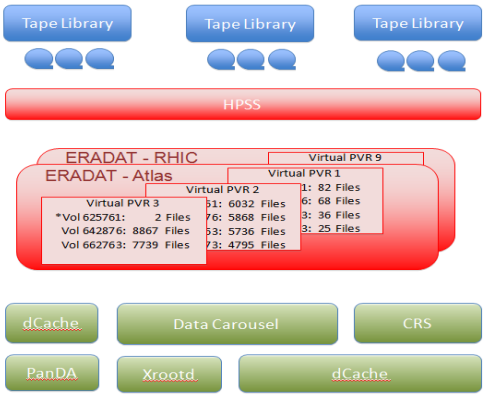


Figure 2

ERADAT (as shown in Figure 2) sits in front of HPSS, interfacing directly with the HPSS API and acts as a request queuing system for users like Xrootd, dCache, DataCarousel and the RHIC data production system (AKA "CRS").

ERADAT solves the problem from many different areas: tape latency control optimization, resource usage optimization, and error handling optimization. In addition, it provides live performance monitoring tools, as well as tools for historical performance analysis.

2.1 Tape Latency Control Optimization

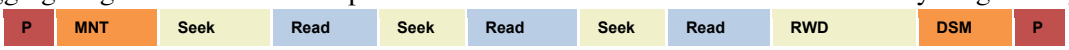
All requests are sorted by tape cartridge ID and then by file position on the tape, so that all the requests for the same tape can be read sequentially in order to reduce redundant tape mounts with minimum rewind / forward.

Tape has quite a long latency for random accessing since the drive must rewind an average of one-third of the tape length to move from one arbitrary data block to another.

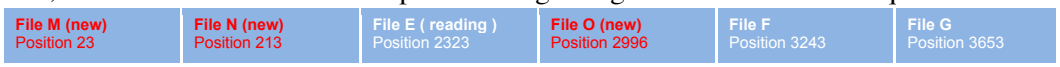
According to the manufacturer's parameters, we have:

- A tape delivery time of 5 sec (P)
- Mounting / Loading: 19 sec (MNT)
- Positioning on file location (assumed to be in the middle of the tape in average): 53 / 2 = 26 sec (Seek)
- Actual data transfer: 80 MB/sec (Read)
- Rewinding the tape:: 98/2 = 49 sec (RWD)
- Dismount / Unload: 19 sec (DSM)
- Place the tape back into the library: 5 sec (P)

Aggregating files on the same tape and then read once can reduce the unnecessary long latency.



New requests may be inserted into the queue, while the queue is already in staging mode. In order to minimize the number of rewinds, the stager will continue to read the next file in the queue until the end, and then it will rewind the tape to the beginning of the first file in the queue.



Files M, N, O are the new inserted requests, while File E is being read. The stager will continue to read O,F, G and then rewind to position 23, and read files M and N.

2.2 Resource Usage Optimization

ERADAT has several optimization levels and rests on a few key principles addressing the tape technology limitations listed in section 1.2 as well as practical organizational realities:

- Priority Staging: Requests marked as high priority will be processed by separate queue, and allows such requests to take the next available drive, like the FastPass in some Theme Park.

- Dedicated Resource: Equipments are purchased by funding from each individual experiment (STAR, Phenix and US-Atlas), ERADAT is designed to support multi-domain partitioning of equipment in order to guarantee the resource availability at all times

- Resource Sharing: 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. This feature is mainly used to separate data production, massive restore (Scalla/Xrootd requests for example) and individual user requests.

- Resource Locking: Each experiment may have multiple types of drives, such as 9940B, LTO-3 and LTO4. ERADAT has a virtual Physical Volume Repository (or 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, as shown in Figure 4.

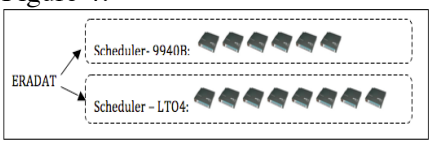


Figure 3

Figure 3:ERADAT supports multiple drive and tape technologies.

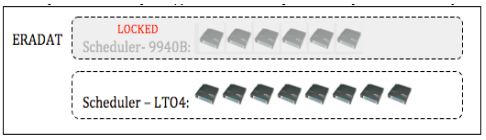


Figure 4

Also, any scheduler can be locked at any time, and other schedulers can still continue to serve files. New requests for locked PVR will be queued in memory

- Requests Sorted: ERADAT sorts requests by tape cartridge and file position, so that all the requests on the same tape can be read at once sequentially in order to minimize tape mounts.

- Resource Consume: Use just enough resource in HPSS. Files are being retrieved sequentially, so there is no need to queue too many requests in HPSS.

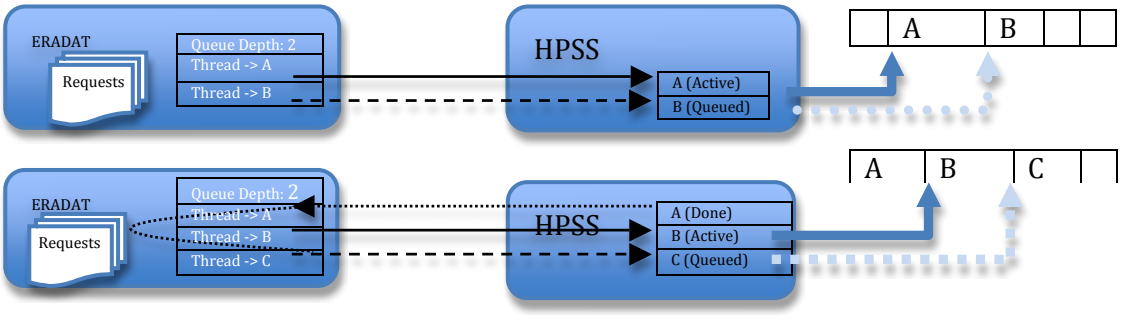


Figure 5 When file A is completed, file B will be activated immediately since B is already queued in HPSS. While B is being read from tape, file C will be queued in HPSS.

2.2.1 Flexible Staging Algorithm

ERADAT implements FIFO and "by demand" policies and they may be enabled on the administrator request. We will discuss the relative merits of handling this at ERADAT or DataCarousel level in the next section.

2.3 Error Handling Optimization

Every request consumes resources, and if it leads to an error, further request may consume

resources for no purpose. ERADAT provides the following Error Handling Policy:

- Configurable Mapping for Error Code and Retry Attempts – for each user.
- LSM (Handbot) Failure – Will not attempt to access the tapes on the failed LSM.
- Black List (No-Access-List) – Block file access from file level.
- Automatic Resource Throttling – Adjust the tape drive usage upon disk availability.

2.3 Monitoring Tools

ERADAT keeps all request status in a database, and do cross-reference with data collected from ACSLS and other sources.

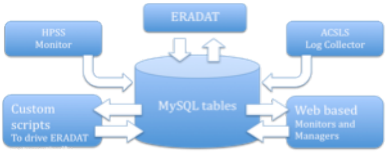


Figure 6

Figure 6 illustrates the brief architecture of ERADAT Monitoring system: HPSS Monitor collects HPSS related info and dump to database, ACSLS Log Collector collects ACSLS's important messages and insert into database.

2.4 Reporting Tools - There are many other tools that generate different kind of reports, such as performance graph. All historical transaction records are kept in separated database for reporting use. These reports are being used to demonstrate the capability of retrieving files out of tapes.

2.5 Remote Monitoring and Management - ERADAT's Web-based Monitoring Tools and Control Panels provide enough application level's operation, and they can be accessed over Internet. All Monitoring pages are in simply HTML, browser friendly and smart phone friendly.

Figure 7: Remote monitoring from smart phones.
Supported OS: Windows Mobile 6, Blackberry OS, Android phone, Apple IOS (iPhone 4)

3 The DataCarousel

The DataCarousel is an extendable and fault tolerant policy driven framework and API allowing, in a multi-user environment, for collaboration to make requests for files archived in a Mass Storage System (HPSS) and have all requests managed and coordinated the same way a full-fledge "batch system" would. The DataCarousel is composed of two separate components, a server and a client, and a few other tools such as a Web interface for monitoring supplement the system with minor additional functionalities. The DataCarousel is entirely written in perl and a SQL based database storage in the back end.

The client is a thin script which sole purpose is to add requests to a central database. The server is the heart of the system and sorts the records, creates a job of N files to retrieve and submits that job to ERADAT as a bundle of requests according to policies. ERADAT call back feature would then call a DataCarousel handler after the file appears on HPSS cache. The handler purpose is to pull the file out of cache on behalf of the user and update the primary request status accordingly (depending on success or failures). The implemented default policies include:

- NONE: no policy is applied; unless other options are applied, FIFO will be used.
- EQUAL: all users are provided equal shares of the resources (in essence, each user has its own FIFO)
- GROUP: users are arranged into groups and all groups are provided equal share – groups are loosely defined and users may be sorted in usage scope such as "data management", "production" or "user" (the default). Groups have also been defined based on physics topics (in this later case, the usage is based on an honored contract as switching between groups is based on the client interface command line parameter).

- GRPW: same as the previous policy but each group is assigned a weight
- The DataCarousel allows extending the SHARE policies using a simplistic yet very flexible mechanism. Any new method BLA could be implemented as a stand-alone perl module.

Throttling is also automated if network communication problems (congestion, downtime, etc) are detected. Emails notices and optimal system parameters "hints" are sent to the DataCarousel manager. With all its features, it is nearly self-operating and self-recovering since 2007 and its easy interface has allowed the STAR experiment to interface Scalla/Xrootd [5] file retrieval to this system for achieving coordination of file requests from MSS to distributed storage.

3.1 DataCarousel file restore for distributed storage at STAR and discussion

Our last use case is a massive file restores using ERADAT in FIFO mode and allowing the DataCarousel to use the tape ID file sorting. The statistics was based on an exercised performed on 2010/02/04 involving 15 LTO-3 drives (with an overlap shared by data production) where 7,187 files were restored over 106 tapes for a total of 4.4 TB. In average, the tapes were mounted 1.21 times instead of an average of 2 in the ERADAT alone was used, leading to an even greater performance. However, this raises the question: why not the asymptotic value of once only?

For one thing, the DataCarousel uses tape ID as they are made available: each request trigger a lookup for its associated tape ID and this is done in a separate process and asynchronously to request processing. This MetaData lookup is slow and the initial few first job submission to ERADAT may then not be sorted by tape ID (as the information may not yet be fully available). This is a minor impact and only goes for the first 30 minutes or so for this amount of records. The remainder of the discrepancy is attributed to HPSS API and behavior: tape may be forced to be dismounted as write operations take precedence over read.

4. Conclusions

ERADAT is a file retrieval scheduler for general use, DataCarousel is designed for further optimization for STAR's environment. ERADAT and DataCarousel system combination have also reached a level where the tool is fully self-adapting and self-recovering from errors and external conditions. With the combinations of those tools, BNL has achieved and demonstrated one to two order of magnitude improvements in data restore speed from tape. ERADAT was especially conceived and used for the RHIC data processing and now adopted by the LHC/US-Atlas community. The DataCarousel has been used to allow large amount of users, user groups or physics activities to share and access files and resources from archival storage in a completely transparent and fairshare manner.

ERADAT has generated interest in other communities – in 2009 and out of discussion held at HEPiX 2009[6], it became apparent that previous discussion with colleagues at the IN2P3 had lead to the adaptation of ERADAT to the use at the CCIN2P3 center and re-branded under a new branding name of TReqS. From then a few month of experience, similar better resource usage were observed and sharing resources between multiple experiments minimizing conflicts achieved. We view the propagation of our approach as a proof of its value.

References

[1] Brookhaven National Laboratory (BNL) – http://www.bnl.gov/
[2] 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
[3] BNL is one of the Tier1 facilities for the LHC/Atlas project.
[4] High Performance Storage System (HPSS)
[5] Fair-share scheduling algorithm for a tertiary storage system – Pavel Jakl, Jerome Lauret and Michael Sumbera, 2010 J. Physics:Conf.Ser.219052005
[6] HEPix 2009, https://www.hepix.org/mtg/Fall%202009%20Meeting