Journal of Physics: Conference Series **119** (2008) 052018 doi:10.1088/1742-6596/119/5/052018

Experiences with gStore, a scalable Mass Storage System with Tape Backend

H. Goeringer, M. Feyerabend, S. Sedykh

GSI, Gesellschaft für Schwerionenforschung Darmstadt, Germany

H.Goeringer@gsi.de

Abstract. GSI is a center for heavy ion research and host of an Alice Tier2 center. The GSI Mass Storage System gStore manages ~200 TB experiment data currently with different life times and access patterns. The data are available 24 hours per day and seven days per week for fast and highly parallel access. For Alice users all gStore data are worldwide accessible via Alice grid software, and for the end of 2007 it is planned to provide ~200TB via xrootd backed with gStore. Successfully in operation for more than 10 years gStore has been developed in parallel continuously by only two FTEs mastering a growth of nearly two orders of magnitude. In 2014 the future FAIR experiments at GSI, CBM and Panda, will have requirements for data capacity and I/O bandwidth reaching those of the current LHC experiments at CERN. This needs another growth of gStore of three orders of magnitude. This paper describes gStore and its potential to master also the challenges coming with the FAIR project.

1. Introduction

GSI [1] in Darmstadt (Germany) is a basic research center providing a linear accelerator, a synchroton, and a storage ring for heavy ions. The experiments at GSI mainly cover the area of nuclear physics, but also other areas such as atomic and plasma physics, biophysics and radiation medicine, and material research. GSI also serves an Alice Tier 2 center. For the ambitious goals of a new billion Euro project named FAIR [2], the GSI experiment facilities will be enhanced considerably till 2014.

The GSI Mass Storage System gStore [3] is a hierarchical storage system with a unique name space and tape backend. For user access gStore offers a command client and an application programmer's interface (API) client. The data are available 24 hours per day and seven days per week for fast and highly parallel access. gStore is designed for many parallel data streams thus avoiding performance bottlenecks.

All experiment data (220 TB in the mid of 2007) reside in background storage on tape, which has a maximum data capacity of 1.6 PB currently. Due to highly parallel access from the GSI compute farms the actually needed files must reside in online storage on disk. The overall capacity of the gStore read and write cache sums up to ~35 TB currently.

The current and future challenges for gStore are considerable. Till the end of 2007 it is planned for the Alice Tier2 center to provide ~200 TB via xrootd backed with gStore. At the end of 2008, the event builders of the Hades experiment will be able to write with overall data rates of 200MB/s into gStore write cache. When FAIR operation starts in 2014, the two main FAIR experiments CBM and Panda will have mass storage requirements similar to those of the current LHC experiments at CERN.

The main intention of this paper is to describe the design, architecture, and usage of gStore, taking into account also the dramatically increased mass storage requirements of the FAIR experiments.

2. The GSI Requirements

In 1996 the GSI tools available to archive experiment data to tape proved to be no longer sufficient, and the need for a new mass storage system became urgent. There were two main constraints the new system had to fulfill. Due to the low manpower available, the system should be easy to install, to operate, and to maintain, and also the further development necessary to grow with the foreseeable future requirements should be simple. Besides that, in those times data must be accessed from Unix and VMS platforms, independent of their origin.

2.1. Available Systems

Obviously a mass storage system for nuclear physics experiments supporting fast and highly parallel access to tape archive and online storage was commercially not available.

Already existing systems at other high energy physics (HEP) labs, however, did not meet the GSI demands. Castor [4], for example, developed at CERN was a complex mass storage system designed for much higher requirements. All aspects relevant for the CERN environment were handled by CERN software, from the level of tape drivers via online disk management up to the user interfaces. Correspondingly the manpower requirements for installation, operating, and maintenance of Castor were assumed to be too high. Besides that, the import of Castor at GSI would have caused a too strong dependency on CERN and the future Castor developments. Alternative systems at other HEP labs were either too special or supported larger commercial components, especially tape managers, not available at GSI.

2.2. The GSI Solution

Since many years the Tivoli Storage Manager (TSM) [5], a commercial software package, was successfully in operation at GSI for data backup and archive business. Therefore expert knowledge about TSM was available, and as TSM offers a comfortable API, it was decided to develop an own GSI mass storage system based on TSM. In this concept the biggest part of the job, namely the complete tape side, will be done by TSM, whereas the remainder will be done by GSI software. So it was easy to tailor the system according to the GSI environment.

The manpower effort necessary to develop a first operational version of gStore amounted to about one man year only, and gStore went successfully into operation at the beginning of 1997. This early start of gStore was considerably facilitated by the still rather moderate requirements of that time.

From the beginning the user acceptance was very high. As expected, in the past ten years the GSI mass storage system had to be enhanced and developed further continuously to keep up with the always and rapidly growing user requirements. However, our concept proved to be successfully, as during this time period on average two FTEs only managed a growth of nearly two orders of magnitude in data capacity and I/O bandwidth. This was possible because gStore is a fully scalable system, and this fundamental property is also needed to master the future challenges.

3. Design Principles

3.1. Hardware independency with TSM

TSM is a commercial software package, widely used in industry for data backup and archiving. In gStore, TSM handles the complete tape side, namely the automatic tape libraries (ATLs), the tape drives, and the storage media. As leading Storage Manager in industry TSM supports practically all relevant hardware on all relevant platforms. Therefore gStore is nearly independent of the hardware for background storage, and introduction of new hardware or a new operating system means only moderate effort if supported by TSM.

3.2. Make yourself what cannot be bought

GSI software provides the functionalities that had to be tailored for the large scientific user community at GSI and elsewhere in the world accessing the GSI mass storage. It utilizes the TSM functionality via the TSM API, provides the client interfaces, and manages the read and write disk caches. The GSI software consists in the meantime of about 100,000 lines of C-code and uses TCP sockets for data transfer over network. The relatively simple code runs with only minor adaptations on practically all operating systems. Another big advantage of home made software is the potentially easy modification of gStore for cooperation with external software packages and middleware.

3.3. Separation of Control and Data Flow

There is a strict separation of control and data flow to avoid performance bottlenecks. Servers handling only control functions are the TSM servers managing a data base and the tape hardware and operations, the entry server, which accepts the client requests and performs queries, and the cache managers.

The servers transferring the mass data are called **data movers** in gStore. The tape I/O operations are done by TSM Storage Agents, lean TSM servers without own data base, running on the data movers. With several data movers many parallel data streams can be maintained. Obviously the number of data movers performing I/O operations with tapes and clients should exceed the number of tape drives to maintain the balance of the whole system.

3.4. Scalability

To fulfil the always growing user requirements gStore must be fully scalable in storage capacity and I/O bandwidth. This has been achieved by connecting data movers and tape drives via a storage area network (SAN). Additionally this solution provides high flexibility, as each data mover can in principle be connected with any tape drive in any tape library. In practice, however, there are restrictions if more than one TSM server is involved, because a data mover can be assigned to only one TSM server.

3.5. Node Independency

Selecting data movers for a given request gStore does some load balancing. Therefore files may be written to tape from one data mover and read again from another data mover, possibly even running with another operating system. This required the introduction of a proxy node name for TSM, a common node name for all TSM API clients running on the different data movers.

4. gStore Architecture

4.1. Storage View

Figure 1 provides a schematical overview of the different storage levels. These are the gStore background storage, provided by the tape libraries, the gStore online storage, built from large read and write disk caches, and the local client storage. Clients have only access to read or write cache. There is no direct connection between background storage and client storage.

Requested files not yet available in online storage must be 'staged' first from tape to read cache. Two read cache storage pools with different attributes are available. Files in a so called StagePool have a guaranteed life time, whereas files in the so called RetrievePool may be erased at any time if space for new files is needed.

Files archived to gStore are written to write cache first. If a certain fill level is reached, the files are moved by gStore servers, which work as TSM API clients, asynchronously to tape. All write cache files are also available for online access. However, as clients have no read access to write cache, requested files are copied internally to read cache first.

4.2. gStore Clients

There are two principal client types available, a command client and an RFIO client. The command client enables to copy sets of files, specified either by file names with wildcard characters (*, ?) or by file lists. Entries in file lists may also contain wildcard characters. Besides file copy the command client provides also query of file attributes and delete functionality.

Additionally RFIO clients for gStore have been implemented at GSI. RFIO (Remote File I/O) [6] is an API developed at CERN for the HEP community. RFIO clients need no local disk storage but handle the data in local memory. They provide selective read access, which may save a lot of data transfer if only small portions of large files are needed. This occurs frequently in HEP environments, for example, if selected parts of trees in ROOT files [7] are analyzed.

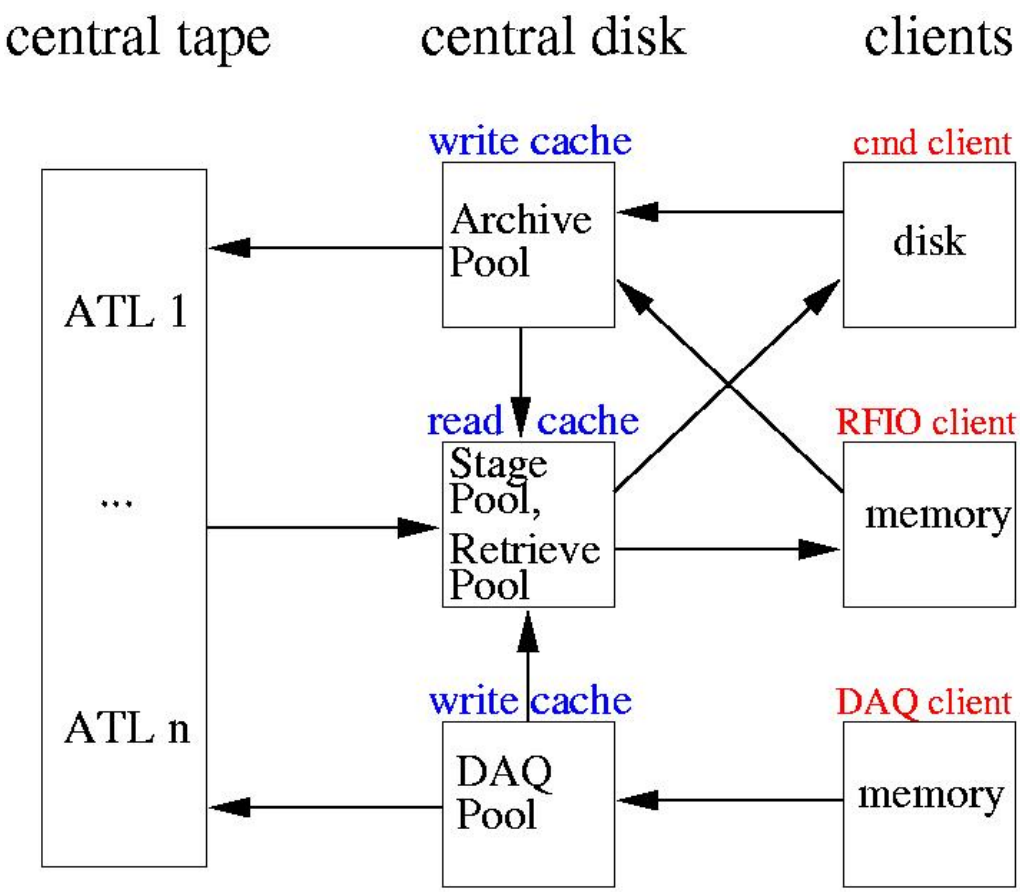

Figure 1: gStore storage levels

4.3. DAQ Connection

Special RFIO clients provide direct connections from the data acquisition of a running experiment to dedicated gStore write caches. In this way the relatively constant data streams originating from the event builders are separated from the heavy and strongly varying load caused by the highly parallel requests of batch farm analysis. For this purpose the RFIO interface had to be enhanced by some GSI extensions.

To prevent from data loss in case of tape damage raw data from an experiment are stored twice on different tape media. The data in question must be written to gStore file systems following a particular naming convention. These file systems are attached to properly defined TSM storage pools causing TSM to maintain the double storage automatically and in background.

4.4. Control Flow

In the simplified architecture overview in figure 2 the flow of control is schematically indicated by thin arrows. If a client submits a request, it connects at first to the entry server. Here a dedicated server process is forked for each client. It queries all information needed from the TSM servers and the read and write cache managers. The information is sent to the client, and the server process terminates afterwards.

Data movers for staging from tape are selected by the read cache manager, taking into account the current load of all read cache data movers. Similarly data movers for archiving from write cache to tape are choosen by the write cache manager.

4.5. Data Flow

The data flow in gStore is indicated by thick arrows in figure 2. It takes place either between a tape drive and the local disk cache of a data mover (via SAN), or between disk cache and client storage (via LAN).

More detailed, in case of staging from tape to read cache, tape data are at first read via SAN by the TSM Storage Agent running on the selected data mover. The Storage Agent sends the data buffers via socket connection to a gStore server, which works as TSM API client and runs on the same data mover. Finally the gStore server writes the data to a RAID in the local read cache. If files are retrieved from read cache, TSM is not involved. The data are read by a gStore server and sent via socket connection to the client.

In our concept it is not necessary to connect the read and write cache disks to SAN.

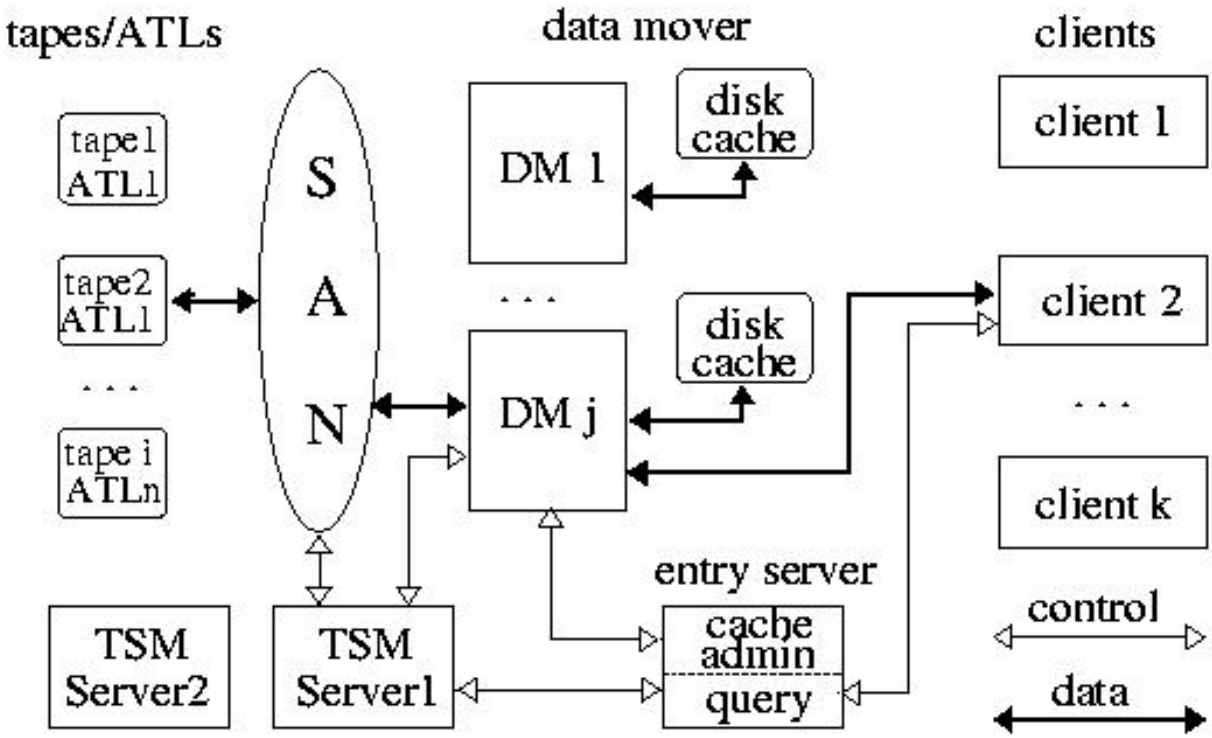

Figure 2: Simplified Architecture Overview

4.6. The Disk Cache Managers

For performance reasons, there are two independent servers managing the read caches for the data files online available and the write caches for new data. Their main tasks are administration of the cache file meta data, the locking and unlocking of files currently in use, and the assignment of data movers for data transfers balancing the load of all data movers in the system.

5. Current System

5.1. History

As already explained in section 2.2, we started at the beginning of 1997 with a very small system according to the comparably low user requirements of that time. It consisted of only one ATL (IBM 3494), also used for daily data backup and recovery business, and of one AIX workstation as data mover, which hosted also the TSM server. There was still no SAN and even no disk cache initially.

Due to high acceptance and heavy usage from the beginning the user requirements increased rapidly since then. Especially the urgent need for disk cache became obvious very quickly, as within less than a year the first tape media with more than 5000 mounts appeared. The first read cache with 80 GB data capacity went into operation in the mid of 1998. At the end of 2002 the first SAN installation connecting 4 LTO1 tapes with 10 data movers was deployed.

5.2. Summary Current Hardware

The entry server is critical for the availability of mass storage and therefore built from a fail-safe pair of Linux server nodes. Currently there are two TSM servers in operation. Each of them manages an ATL and has about ten data movers assigned. More details can be found in tables 1 and 2. In 2006 outdated IBM 3590 tape drives in the IBM 3494 ATL were replaced by actual 3592 tape drives. Since May 2007, all new data are written only to the 3494 ATL, which has sufficient data capacity for the next few years.

TSM server I	POWER5 p5x0, AIX Version 5, TSM Version 5.4
IBM 3494 ATL	2263 media slots, ~1.6 PB max data capacity
6 IBM 3592 tape drives	>100 MB/s, 700 GB/tape
8 data movers	Suse Linux, 31 TB disk cache

Table 1: gStore Hardware I

TSM server II	Windows 2000 cluster, TSM Version 5.4
Sun StorageTek L700 ATL	700 media slots, ~140 TB max data capacity
9 IBM LTO2 tape drives	35 MB/s, 200 GB/tape
9 data movers	Windows 2000, 4 TB disk cache
1 data mover	Suse Linux, 0.4 TB disk cache

Table 2: gStore Hardware II

5.3. Performance

The gStore entry server processes up to 50 single file queries per second scanning the read and write cache data bases and the data bases of the two TSM servers. For higher loads in the future, additional entry servers can easily be added.

Staging files from IBM 3592 tape via SAN to a data mover RAID, file transfer rates of up to 122 MB/s have been measured. This is achieved under optimal circumstances, if the data file is already located on the tape already mounted, and if there is no concurrent I/O on the data mover RAID. However, when reading a list of files from tape, such rates are often reached, because TSM provides means to sort the files for optimal read order, and because files are often read in the same order as written to tape. Though in a stage process two servers on the data mover are involved (see section 3.5), obviously the nominal tape I/O rate of 100 MB/s is clearly exceeded. Probably this results from the effect that most data files at GSI are slightly compressible with the algorithms used by TSM. Similar data rates >100 MB/s are achieved when writing from write cache to IBM 3592 tape.

With all 6 tape drives used in parallel, the current maximum I/O bandwidth between disk cache and IBM 3494 ATL amounts to about 700 MB/s. Additionally available are up to 300 MB/s I/O bandwidth between disk cache and Sun StorageTek L700 ATL.

5.4. Current usage

Table 4 shows the number of Terabytes (TB) transferred with gStore in 2006 and 2007 through the GSI network. The LAN data traffic flows between clients and online storage, whereas the SAN data traffic takes place between online storage and tape libraries. In the last two years, on average

significantly more than 1 TB of data was moved each day, and about half a PB each year. Currently 270 TB of experiment data are stored on tape, including 50 TB of doubly stored 'valuable' raw data.

5.5. Alice Tier 2 Support

For Alice users all gStore data are worldwide accessible via Alice grid software. In a test environment for the Alice tier 2 center at GSI gStore as backend for xrootd has been implemented successfully. Till the end of 2007, it is planned for the Alice Tier2 center to provide ~200 TB via xrootd backed with gStore.

time range	LAN	SAN	sum
year 2006	333	120	453
Jan-Sep 2007	273	159	432
average day 2006	0.91	0.33	1.24
average day 2007	1.01	0.59	1.60
top day (Dec 31, 2006)	9.62	0	9.62

Table 4: gStore Data transfers [TB]

6. Conclusions

With 0.27 PB stored data and an integrated yearly data transfer of about 0.5 PB currently gStore is established as a medium size mass storage system at GSI. Over ten years of continuous development a growth of nearly two orders of magnitude has been managed by on average only two FTEs. Due to increased usage and importance since 2006 a third FTE is partially working for gStore. Hardware from different vendors and servers on various platforms have been deployed successfully.

This was possible for two main reasons. gStore is completely scalable in data capacity and I/O bandwidth, and gStore is based on TSM, which, as leading Storage Manager in industry, handles practically all relevant hardware on all relevant platforms.

This full scalability and flexibility of gStore is necessary to keep up with the challenges of the future and the basis to master the required growth of three more orders of magnitude necessary for the planned FAIR experiments, CBM and Panda. To attain this goal the number of parallel data streams to be handled must be increased comprehensively, but it was one of the design goals of gStore to enable this. The accomplishment of this task will be facilitated by the expected technical developments of the next decade as well as the fact that gStore can easily be adapted for cooperation with external software packages and middleware, because central parts of the gStore software have been developed at GSI.

References

[1] Gesellschaft für Schwerionenforschung GmbH Darmstadt
http://www.gsi.de
[2] Facility for Antiproton and Ion Research
http://www.gsi.de/fair/index.html
[3] The GSI Mass Storage
GSI Report 2007-1, 210
http://www.gsi.de/library/GSI-Report-2007-1
[4] Castor - CERN Advanced STORage manager
http://castor.web.cern.ch/castor/
[5] Tivoli Storage Manager
http://publib.boulder.ibm.com/infocenter/tivihelp/v1r1/index.jsp
[6] Remote File I/O
http://castor.web.cern.ch/castor/docs/guides/man/CASTOR2/#rfio
[7] ROOT
http://root.cern.ch/