High Performance Experiment Data Archiving with gStore

Chep 2012, New York
May 21, 2012

Horst Goeringer, Matthias Feyerabend, Sergei Sedykh
H.Goeringer@gsi.de
Overview

1. Introduction GSI and FAIR
2. How gStore works
3. gStore SW and HW components
4. some features
5. gStore usage
   - gStore and lustre
   - online data storage from running experiments
6. Outlook
GSI Helmholtzzentrum für Schwerionenforschung – Center for Heavy Ion Research

Budget: 106 Mio. € (90% Bund, 10% Hessen)

Employees: 1100

External Scientific Users: 1200

Large Scale Facilities: Accelerators and Experiments
Research Areas at GSI

Nuclear Physics (50%)
- Nuclear reactions up to highest energies
- Superheavy elements
- Hot dense nuclear matter

Atomic Physics (15%)
- Atomic Reactions
- Precision spectroscopy of highly charged ions

Biophysics and radiation medicine (15%)
- Radiobiological effect of ions
- Cancer therapy with ion beams

Plasma Physics (5%)
- Hot dense plasma
- Ion-plasma-interaction

Materials Research (5%)
- Ion-Solid-Interactions
- Structuring of materials with ion beams

Accelerator Technology (10%)
- Linear accelerator
- Synchrotrons and storage rings
FAIR – Facility for Antiproton and Ion Research

GSI-today
- all kinds of ions
- max. 90% speed of light

GSI-tomorrow / FAIR
- new isotopes
- Anti-Protons
- 10,000 times more sensitive
- higher speed
FAIR – Facility for Antiproton and Ion Research
gStore: storage view

- Tape
- Disk cache
- SAN
- Write cache
- LAN
- Read cache
- StagePool, Retrieve Pool
- Write cache
- DaqPool
- Clients
  - Cmd client
  - Disk
  - API client
  - Memory
  - DAQ client
  - Memory
gStore: software view

two main parts:

1. **TSM: Tivoli Storage manager (IBM)**
   - handles automatic **tape libraries (ATL)** and **tape drives**
   - all devices supported by TSM also supported by gStore
   - utilized by GSI software via TSM API
2. **GSI Software** (>100,000 lines of C-code):
   - interfaces to users (command, API)
   - interface to TSM (API)
   - entry servers
   - data mover servers
   - read/write cache managers:
     - meta data management
     - cache file locking
     - space management
     - data mover selection *(load balancing)*
IBM 3584-L23 tape library (ATL)

- 8 IBM 3592-E07 tape drives (SAN)
  - 250 MB/s read/write per drive
  - 4 TB/medium uncompressed
- 8.8 PB overall data capacity
  - ~1 PB used
- 40 €/TB media costs:
  - add 60% for library/drives =>
  - for reliable long term archiving: no similar inexpensive alternative for tape
- really green IT
IBM 3584-L23 tape library (ATL)

- copies of raw experiment data
- 4 IBM 3592-E06 tape drives (SAN)
  - 160 MB/s read/write per drive
  - 1 TB/medium uncompressed
- 1.2 PB overall data capacity
  - 200 TB used
- in different building: enables disaster recovery
gStore: Hardware Status

currently **17 data movers:**

- **Suse** Linux
- 3 – 20 TB disk cache
- 4 Gb **SAN** connection to ATL
- **Ethernet** connection to clients:
  - 10 Gb (9x, limit 40 Gb switch)
  - 1 Gb (8x)
gStore: Hardware Status

data movers overall:

• **200 TByte disk cache** (read/write/DAQ)

• **max. I/O bandwidth:**
  – disk cache <-> tape: 2 GByte/s
  – disk cache <-> clients: 5 GByte/s
gStore: how it works

tapes/ATLs

TSM Server1

TSM Server2

data mover

DM 1

DM j

data mover

disk cache

disk cache

entry server

cache admin query

clients

client 1

client 2

client k

data

control
gStore: design principles

gStore:
• reliable long-term archive storage
• high-performance access
• fully scalable in data capacity
• fully scalable in I/O bandwidth
gStore: some features

- 64 bit servers
- 32/64 bit clients
  - command clients
  - API clients
- recursive file operations
  - wildcards in file names
  - file lists
gStore: some features

• large file transfer with **single command**
  – **performance increase** by parallelization

• staging big file sets:
  – files on **different tapes**: *copy in parallel* to different data movers
    • decreases staging time
    • enables highly parallel access
  – files on **same tape**: *distribute* to several data movers (**sequentially**)
    • enables highly parallel access
    • important as **media size increases**
  – impossible for user (no tape info)
# gStore Usage

<table>
<thead>
<tr>
<th></th>
<th>TB transferred</th>
<th>average MB/s</th>
<th>no. of files transferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1 – May 13, 2012</td>
<td>710</td>
<td>61</td>
<td>880,000</td>
</tr>
<tr>
<td>average day</td>
<td>5.3</td>
<td>61</td>
<td>6,567</td>
</tr>
<tr>
<td>top day (Aug 12, 2011)</td>
<td>46.7</td>
<td>540</td>
<td>21,600</td>
</tr>
</tbody>
</table>
lustre: GSI online mass storage

- ~ 3 PB size
- small experiments: gStore cache, no lustre data transfers

**gStore <-> lustre:**

- gStore data movers <-> lustre OSTs
  - up to 500 MB/s (single file)
  - max bandwidth 5 GB/s

- or tape <-> lustre
  - up to 250 MB/s (single file)
  - max bandwidth 2 GB/s
gStore: online data storage

On-line data storage: constant, continuous data streams from data acquisition over long time ranges

- many data streams in parallel
  - e.g. Hades experiment: 16 data streams
- distribution to DMs: load balancing
- undisturbed by offline business
- fast data availability in lustre
  - for online monitoring & analysis
gStore: online data storage

storage order:

1. gStore write cache
2. optionally copy to lustre:
3. finally migration to tape
   - if preset cache fill level reached

overall bandwidth:

- 500 MB/s: with full copy to lustre
- 1 GB/s: no copy to lustre
two online copy modes to lustre:

1. parallel copy
   • data buffer level
   • problem: lustre latencies
     -> delay of DAQ read-out

2. sequential copy
   • file level
   • storage to write cache independent of lustre
Hades experiment march/april 2012:
• 5 weeks beam time
• 16 data streams in parallel
• acquisition rate ~100 MB/s
  – storage to write cache
  – copy to lustre (all files)
  – migration to tape
• overall ~200 TB of data
• in parallel up to 3 add. experiments (~MB/s)
• handled by gStore without problems
gStore 2012

- **lustre**
  - 3 PB
- **Tape Robot**
  - Capacity 8.8 PB
  - Expandable: 50 PB

**gStore Data Movers**
- 200 TB Buffer Storage
- 5 GB/s

**offline Clients**

**online DAQ Clients**

- 0.5 GB/s

- 2 GB/s
current/future projects:

1. optimal utilization of available bandwidth: automatic parallelization of large data transfers (single command)
   - for staging already done
     • all data transfers on server side
   - senseless for some client storage, e.g.
     • desktops
     • overloaded file/group servers
     • servers with small network bandwidth
gStore: Outlook

next: parallelize transfers gStore <-> lustre

• lustre: powerfull client system

no. of parallel processes limited by
• no. of available data movers
• no. of available tape drives

writing to lustre: effective lustre load balancing

reading from lustre: file distribution on lustre OSTs depends on history
future projects:

2. HSM for lustre
   - future of EOFS and lustre GPL?
   - not yet under investigation
future projects:

3. preparation for FAIR (Start 2018)
   - storage situation 2018?
   - data growth: 33 PB/year (2018)
   - current ATL: expandable to 50 PB (E07)
   - with next gen. (E08): expect >=100 PB
     - E06->E07: was factor 4!
   - data bandwidth: need factor >10
In the past 15 years at GSI we mastered similar increases in data capacity and bandwidth.

Technical progress helped in the past and will help also in the future.

gStore is designed for scalability.
gStore: how it works

Diagram showing the components of the gStore system, including entry server, client, data mover, data server, disk cache, TSM server, TSM storage, client disk, control data, and mass data.
gStore: how it works

client

entry server

query server

cache mgr

tsm server

data mover

data server API

tsm storage agent

client disk

disk cache

tsm storage

control data

mass data

SAN 250 MB/s
gStore: how it works
gStore: Usage Profiles

mainly three use cases:

1. **transfer large amounts of data**, e.g. between
   - lustre
   - group/file servers
   - local disks
   and gStore read/write cache, e.g.
   - stage actions to prepare a data analysis
   - archive actions after a data analysis
   - to handle lack of lustre space
2. parallel transfer of many single files between farm nodes
   - local disks
   - local memory (API)

and gStore read/write cache:

• for data analysis
• smaller GSI experiments not needing lustre
3. On-line data storage: constant, continuous data streams from running experiments over long time ranges

- many data streams in parallel
  - e.g. Hades experiment: 16 data streams
- distribution to DMs: load balancing
- undisturbed from offline business
- fast data availability in lustre
  - for online monitoring & analysis
gStore: online data storage

some more features of online lustre copy:
- **fraction of files**: selectable by user (0 to all)
- optionally: **new lustre subdir** after n files
- **subdir naming conventions**