Efficient Access to Massive Amounts of Tape-Resident Data

David Yu
Jérôme Lauret

CHEP 2016
San Francisco
Managed by High Performance Storage System (HPSS)

~90 PB of data on tapes

~65K+ tapes, mix of LTO 4,5,6 and T10KD technologies
Why are we using tapes?

Tape is reliable, energy efficient, low cost (per GB), and fast!

One LTO-7 can reach 300 MB/s, Bit Error Rate $1 \times 10^{19}$, last for 30 years

We just restored some 15 years tapes (9940B), 100% successful.

Tape is great for data archiving, but it’s sequential access!

Randomly restoring files from massive amount of tapes degrades the read performance primarily due to frequent tape mounts, forwards and rewinds

We have an in house developed system, called ERADAT, to optimize the tape mounts, tape reads, and resource control.

It also provides performance monitoring as well as statistics.
A scheduler system, originally based on a software from Oak Ridge National Lab, developed in the early 2000s.

After some major modifications and enhancements, ERADAT now provides advanced HPSS resource management, priority queuing, resource sharing, web-browser visibility of real-time staging activities and advanced real-time statistics and graphs.

An interface between HPSS and other applications such as the locally developed Data Carousel providing fair resource-sharing policies and related capabilities.
Unmanaged Resources (Tape and Drives)

- Tape drives are first come first serve…
- Users will fight for tape drives
- Waiting time becomes unpredictable
- Unable to prioritize tasks
How to address the problems: Repeat mounts, forwards, rewinds…

Optimize the performance
• Reduce Tape Mounts
• Reduce Rewinds
• Processing Order (FIFO…)

Resource Management
• Resource Allocations
• Dedicated resources
• Shared resources
ERADAT Features

- Staging optimization
- Tape selection orders: FIFO, LIFO, and “By Demand”
- Priority Staging
- Resource Management
  - Resource guaranteed
  - Resource sharing
- Resource Allocation Oversubscription
- Drive-generation Oversubscription

- Multi-level real-time debug log on/off switch
- Sync or async (callback) option
- Real-time configurable auto-retry
- Advanced Thread control
Real-time Monitoring Tools:
- Web-based Control Panel
- Performance Graphs/Reports
  - Staging Activity Graph
  - Tape staging performance report
  - Drive staging performance report
- Staging suspension/resume control
  - Drive-generation level
  - Global level (lock all)
- Resource Lock: tape and drive (HPSS level)
- Auto-detect LSM down, bypassing offline LSM
All interfaces, graphs and reports are HTML based, works on any web-browsers (Any OS, any newer web-browsers)

Tested Environment:
- Mac OSX
- Linux
- Windows

Tested Web-Browsers:
- Safari
- Firefox
- Chrome
- IE (PC only)
Requests are first aggregated by tape-id – reduce mounts
Sorted by position on the tape – reduce rewinds.
If “By Demand”, sort the tape list by # of reqs.

By FIFO: Tape mount order: E, A, B
By Demand: Tape mount order: A, E, B
FIFO vs “By Demand”
Which one is more efficient?
File 0009 (Tape E, pos:10)
File 0002 (Tape A, pos:105)
File 0016 (Tape B, pos:8)
File 0008 (Tape A, pos:605)
File 0004 (Tape A, pos:1)
File 0005 (Tape E, pos:8)
File 0001 (Tape A, pos:40)
File 0022 (Tape C, pos:48)
File 0021 (Tape C, pos:40)

By Demand: Tape mount order: C, A, E, B
By FIFO: Tape mount order: C, E, A, B
Resource Management

Tape drive resource allocation should be under total control, to avoid a service interruption from drive being taken by other process.

<table>
<thead>
<tr>
<th>Drive Info</th>
<th>Total Drives</th>
<th>Allocated Drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Raw LTO-5</td>
<td>18</td>
<td>14</td>
</tr>
<tr>
<td>Star Raw LTO-6</td>
<td>21</td>
<td>4</td>
</tr>
</tbody>
</table>

Efficient Access to Massive Amounts of Tape-Resident Data

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Tape Info</th>
<th>Tape ID</th>
<th>Files</th>
<th>Avg size</th>
<th>Status</th>
<th>Staged</th>
<th>Failed</th>
<th>Last Staged</th>
<th>Mount Time</th>
<th>Dry Add</th>
<th>Dry Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Raw LTO-5</td>
<td>S67520</td>
<td>1/1</td>
<td>5,000,419,328 Mounted</td>
<td>0</td>
<td>8-13 14:25:33</td>
<td>0</td>
<td>9-13 14:25:17 (55 sec)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67532</td>
<td>2/4</td>
<td>5,000,366,432 Reading</td>
<td>1</td>
<td>8-13 14:25:36</td>
<td>1</td>
<td>9-13 14:24:18 (00:01:54)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67533</td>
<td>2/5</td>
<td>4,331,375,618 Reading</td>
<td>0</td>
<td>8-13 14:25:15</td>
<td>0</td>
<td>9-13 14:25:15 (57 sec)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67534</td>
<td>1/1</td>
<td>5,000,685,760 Mounted</td>
<td>2</td>
<td>8-13 14:25:53</td>
<td>2</td>
<td>9-13 14:23:01 (00:03:11)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67535</td>
<td>2/2</td>
<td>5,000,240,064 Reading</td>
<td>1</td>
<td>8-13 14:25:00</td>
<td>1</td>
<td>9-13 14:23:19 (00:05:54)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67537</td>
<td>2/1</td>
<td>5,001,083,138 Reading</td>
<td>3</td>
<td>8-13 14:26:03</td>
<td>3</td>
<td>9-13 14:21:40 (00:04:32)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67538</td>
<td>2/5</td>
<td>2,548,830,680 Reading</td>
<td>3</td>
<td>8-13 14:26:09</td>
<td>3</td>
<td>9-13 14:22:59 (00:03:33)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67539</td>
<td>1/1</td>
<td>5,000,431,618 Reading</td>
<td>0</td>
<td>8-13 14:25:21 (48 sec)</td>
<td>0</td>
<td>9-13 14:24:46 (00:01:20)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67540</td>
<td>1/1</td>
<td>5,001,710,680 Mounted</td>
<td>0</td>
<td>8-13 14:24:10</td>
<td>0</td>
<td>9-13 14:23:06 (00:04:04)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67541</td>
<td>2/2</td>
<td>4,318,434,990 Mounted</td>
<td>0</td>
<td>8-13 14:24:53</td>
<td>0</td>
<td>9-13 14:23:19 (00:02:54)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67542</td>
<td>1/1</td>
<td>5,001,859,136 Mounted</td>
<td>0</td>
<td>8-13 14:25:07 (00:01:05)</td>
<td>0</td>
<td>9-13 14:23:19 (00:01:20)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67543</td>
<td>2/3</td>
<td>5,000,589,592 Reading</td>
<td>0</td>
<td>8-13 14:26:10</td>
<td>0</td>
<td>9-13 14:21:32 (00:04:40)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S67607</td>
<td>2/5</td>
<td>3,643,325,888 Reading</td>
<td>4</td>
<td>8-13 14:26:06</td>
<td>4</td>
<td>9-13 14:20:17 (00:05:55)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S66000</td>
<td>1/1</td>
<td>5,000,390,560 Reading</td>
<td>0</td>
<td>8-13 14:25:41</td>
<td>0</td>
<td>9-13 14:24:16 (00:01:67)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S66100</td>
<td>1/1</td>
<td>1,963,038,388 Reading</td>
<td>2</td>
<td>8-13 14:25:55</td>
<td>2</td>
<td>9-13 14:23:38 (00:02:34)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S66125</td>
<td>2/2</td>
<td>2,067,861,312 Reading</td>
<td>1</td>
<td>8-13 14:26:08</td>
<td>1</td>
<td>9-13 14:25:01 (00:01:11)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S66129</td>
<td>2/3</td>
<td>5,001,518,876 Reading</td>
<td>3</td>
<td>8-13 14:26:10</td>
<td>3</td>
<td>9-13 14:21:47 (00:04:25)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Raw LTO-5</td>
<td>S66917</td>
<td>1/1</td>
<td>5,015,422,404 Reading</td>
<td>3</td>
<td>8-13 14:25:56</td>
<td>3</td>
<td>9-13 14:22:09 (00:04:03)</td>
<td>1</td>
<td>IBM LT05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TOTAL: 10 Tapes 28 Files
Multiple Users, multiple policies
Customized resource allocation for each user

Resource sharing:
Adjust resource allocation on demand.

Resource sharing

USER A
6 -2 Drives

USER B
4 +2 Drives

0 jobs

800 jobs
Resource Allocation Oversubscription

Let user borrow (fight for) drives on demand

Resource sharing

USER A
6 Drives

USER B
6 Drives

100 jobs
800 jobs

0 jobs
100 Waiting
Drive-generation Oversubscription

Use later gen drive to read prev gen tapes
Use 2 LTO-6 drives (virtual LTO-5) to read LTO-5 tapes

- 800 LTO-5 Reqs
- 20 LTO-6 Reqs

USER A
6-4 LTO-5 Drives
4-6 LTO-6 Drives
Randomly restoring 704 x 10 GB files out of 21 tapes, with 15 available drives.

- **Direct submission**: Using 15 job-queues, it took 270 mins to complete. Average ~444 MB/s. Used 34 mounts.

- **Using ERADAT**: Using 15 job-queues, it took only 70 mins to complete. Average ~1.7 GB/s. Used 21 mounts.
Use case (real)

In production system, used ERADAT to stage files using 8 LTO-6 drives

- But something happened, we lost the disk cache copies (purged)
- All users were pulling files directly from tapes in random order… used all 16 drives unmanaged!

<table>
<thead>
<tr>
<th>Date</th>
<th>Mounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>09-28</td>
<td>67</td>
</tr>
<tr>
<td>09-29</td>
<td>92</td>
</tr>
<tr>
<td>09-30</td>
<td>68</td>
</tr>
<tr>
<td>10-01</td>
<td>140</td>
</tr>
<tr>
<td>10-02</td>
<td>507</td>
</tr>
<tr>
<td>10-03</td>
<td>393</td>
</tr>
<tr>
<td>10-04</td>
<td>518</td>
</tr>
<tr>
<td>10-05</td>
<td>709</td>
</tr>
</tbody>
</table>

Daily Data Transfer Volume (TB) into HPSS in last 8 days

- Used ERADAT
  - 8 drives, up to 150 TB/day
- Lost cache copies
- Unmanaged restore from tapes…
  - 16 drives, ~2 TB/day
Massive Staging
Tapes: LTO-5
Avg File Size: 10 GB
16 LTO-5 Drives
Delivered ~2.0 GB/s
Or 128 MB/s/drive
All overhead included
LTO-5 max transfer rate: 140 MB/s

8/30 – 9/16 (18 days)
242,679 file
2.3 PB
1,708 LTO-5
ERADAT Staging Activity Graph

~245,000 files

Queue goes down at constant rate

Staging performance sustain
ERADAT and other systems

Data Carousel

dCache

ERADAT

CRS

HPSS

High Performance Storage System
Why the carousel
In a sustained request and multi-user environment, ultimate performance may not be the best choice - 1 user requesting 1 file from 1 tape may wait "forever" if ERADAT's restore policy is strictly "high demand"

The long tail

![Files per tape diagram](chart)

- High demand – high priority
- Low demand – long wait time
The Carousel provides ways to achieve fair-shareness by switching between ERADAT policies (high demand, FIFO, low demand) and allows sparse requests to be satisfied in a reasonable time.
Implements:

- **SHARE**: user and group based sharing policies
  - **EQUAL**: all users get equal share
  - **GROUP**: all groups get equal share
  - **GRPW**: group are weighted, equal share within group
- **ORDER**: sorting of requests ahead of ERADAT
  - By time files were requested
  - By tapeID (strict tape ordering, or hybrid approach i.e. submits all files requested from a given tape at the same time to balance fair-share and optimization)

The system balances between sharing (the bandwidth) and sorting (optimization) and switches between ERADAT high demand and FIFO to achieve goals.

Flexible framework - can be extended by any custom SHARE or ORDER policies of your own.
**Summary**

**ERADAT** is a file retrieval scheduler for general use, it is designed to optimize the tape mount and read, and provides resource management for multi-user and multi-purpose use.

**Data Carousel** is designed for further optimization for STAR’s environment.

- The DataCarousel is an extendable and fault tolerant policy driven framework in a multi-user environment.
- For collaboration to make file retrieval requests to ERADAT.
- The DataCarousel allows extending the SHARE policies using a simplistic yet very flexible mechanism.
Questions?

Thank you!