St. Lammel, Fermilab-CD, October 7^{th}, 1998

CDF Run II Data Handling
Tape Handling Subsystem

Stephan Lammel

- **Tape Handling Overview**
- **The Basic Ideas**
- **Tape Handling Hardware Components**
- **Tape Handling**
- **Tape Handling Software Components**
- **Enstore**
- **Concluding Remarks**

Tape Handling Overview

We expect to record about 1 PByte of data in the two years of Run II. This is a 20 fold increase in data volume compared to Run I.

Data will be written from the data logger, production, and user selections.

Data will be read by production and analysis. Analysis activities make the largest fraction of tape I/O with over 2/3 of the total tape I/O bandwith.

Tape access will be rather different from what we did in Run I.

The use of Exabyte 8200 and 8500 technology directly from analysis_control caused physicists great pain.

For Run II we plan to eliminate any direct tape access by a user but rather spool any tape resident data to disk first.

This approach is also necessary since ROOT's I/O mechanism requires direct access files.

The Basic Ideas

The data handling software architecture is based on the assumption of a volume based tape subsystem.

The idea is to use a simple staging system based on the mt_tools package (a CDF software package).

The mt_tools package was already used successfully in Run I in both online and production. The package has spooled over 100 TBytes to and from tape. (We spool about 1/2 TByte/day for serial media tests.)

The package has undergone a major overhaul during the summer in preparation for Run II (2 FTE weeks).

The package is a very light-weight layer on top of the Fermilab Tape Tools (FTT) package to write ANSI labeled tapes and to add error recovery and retry.

Tape reading and writing is not rocket science.

Tape Handling Hardware Components

- The Run II tape technology will be selected by the serial media working group in March/April of next year.
- Current thinking of the group is in the direction of reliable but commodity tape technology:
  - tapes with 50 to 100 GBytes capacity and
  - tape drives with read/write speed of order 5 MB/sec.
- All tape drives (inside the library or operator serviced) will be directly attached via SCSI to the compute nodes in the analysis system.

- We have optimized the analysis/storage system and planed the acquisition in such a way that we can put all of the 1 PByte into the automated tape library.
- Shelves and operators are the fallback in case of library overflow.

We have current generation tape drives of the potential Run II candidate technologies.

The EMASS automated tape library is in the process of being setup in FCC. (3 quadrotower system, shared with D0 until second library arrives.) Thanks to Mark Leininger who did a terrific job here!

Tape Handling

The decision for a volume based approach is based on the fact that tape volumes are the most natural unit in dealing with tapes.

Physics datasets (primary, secondary, and even tertiary) are all much larger than the size of individual tapes.

Writing the files of the same dataset to the same physical tape is most natural.

If a dataset is read, all files of that dataset are needed and the whole tape is required.

In most cases where tapes go bad the whole tape becomes unreadable, not individual files. In that case the volume should be flagged unavailable (while it is being recovered), not individual files.

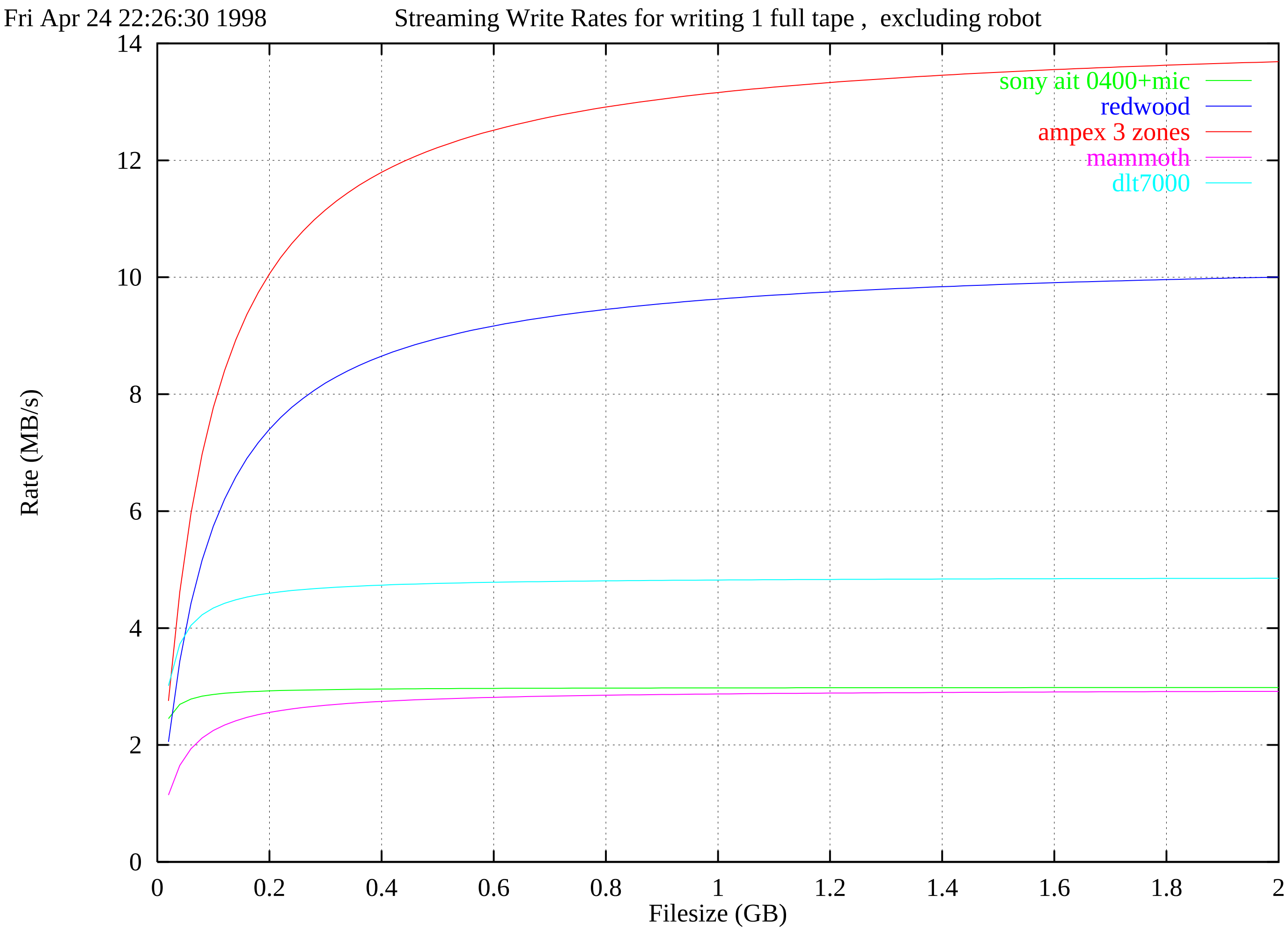
Tape Handling

Spooling full tapes to disk allows an efficient use of the tape drive resources.

The effective I/O bandwidth of a tape drive in single file pick or write mode is degraded by as much as a factor of 2 over a volume based read/write (even with large, 1 GByte files).

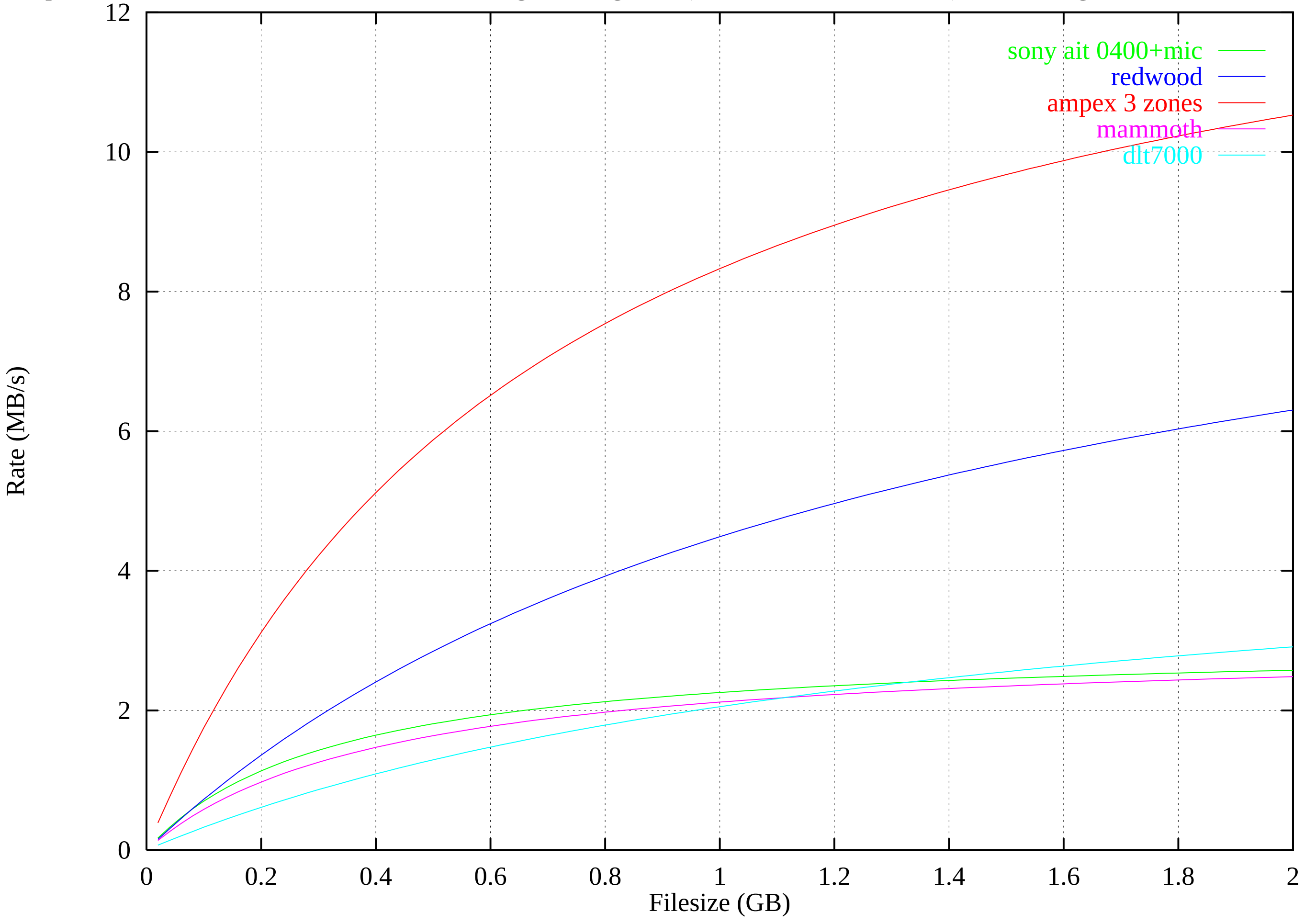
In order to overcome this problem in a file base tape handling system, requests for files on the same tape would have to be clustered together.

The CDF Data Catalogue has to keep track of the file layout on tape so that the staging jobs can be organized apppropriately and their execution and completion be managed by the batch system.



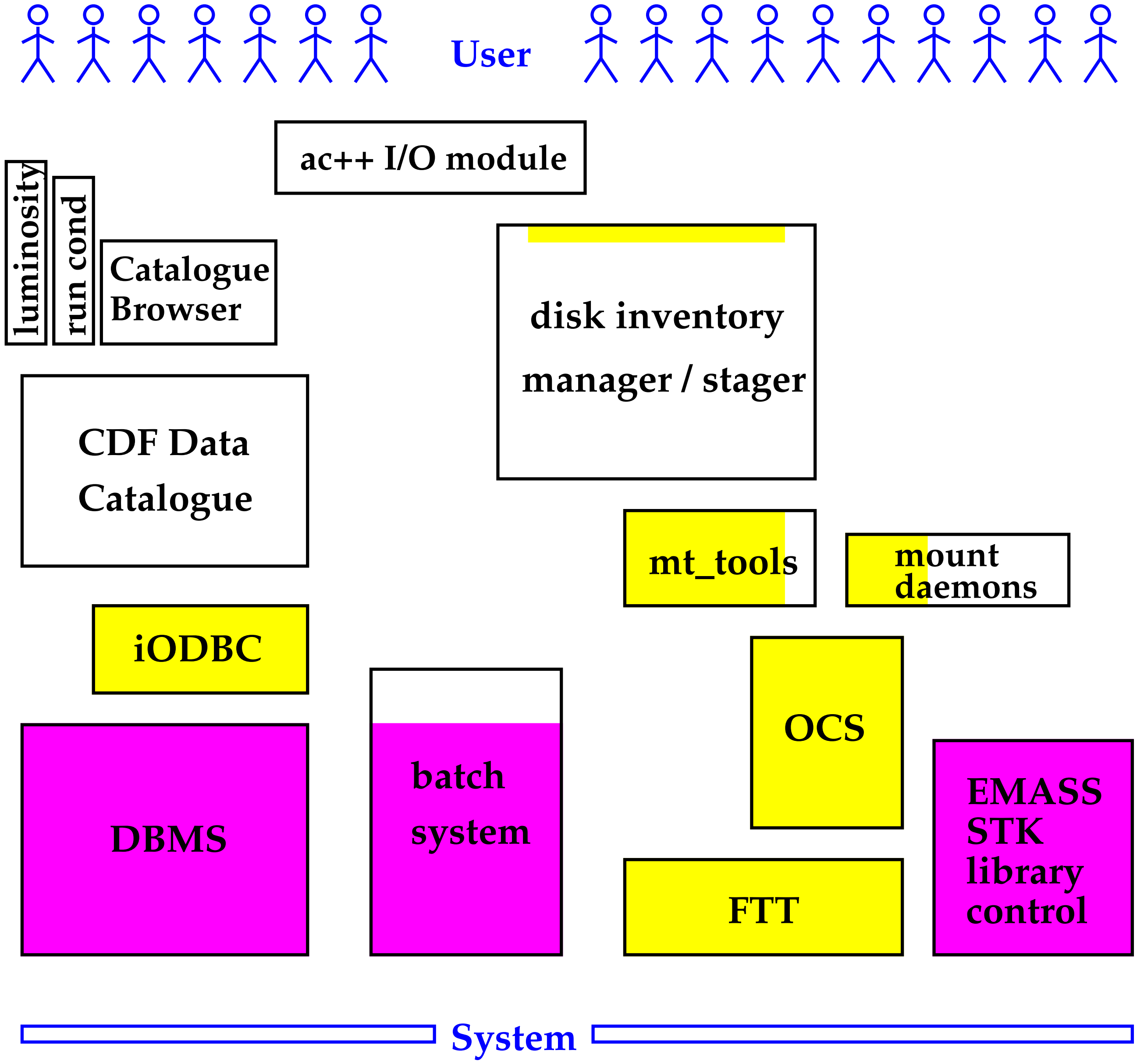
Fri Apr 24 22:26:30 1998
Streaming Write Rates for writing 1 full tape , excluding robot
Rate (MB/s)
Filesize (GB)
sony ait 0400+mic
redwood
ampex 3 zones
mammoth
dlt7000
0
2
4
6
8
10
12
14
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Fri Apr 24 22:26:21 1998 Read Rates for reading 1 average file (successive random reads) , excluding robot



Tape Handling Software Components

- The FTT package handles all low level tape I/O and device communication. The package exists and is well tested on a large number of platforms and peripherals.
- The OCS package handles tape drive status tracking, that is allocation, mount status, etc. and tape mount requests. The package exists and meets our requirements. However, the software is hard to install and needs an overhaul to reduce some of its complexity and to streamline the implementation.
- We have library control software for both the STK and EMASS library. The software is provided by the library vendor and updates are available at regular intervals. The software provides very high level user APIs.
- We need to interface the library control software with the OCS package so our mount requests are forwarded to the library controller. There exists an interface for stacker and small scale libraries.
- The mt_tools package is a thin layer on top of FTT and OCS to write, read, label, list, or dump ANSI standard compliant tapes.


User
ac++ I/O module
luminosity
run cond
Catalogue Browser
disk inventory manager / stager
CDF Data Catalogue
mt_tools
mount daemons
iODBC
OCS
batch system
DBMS
EMASS STK library control
FTT
System

Enstore

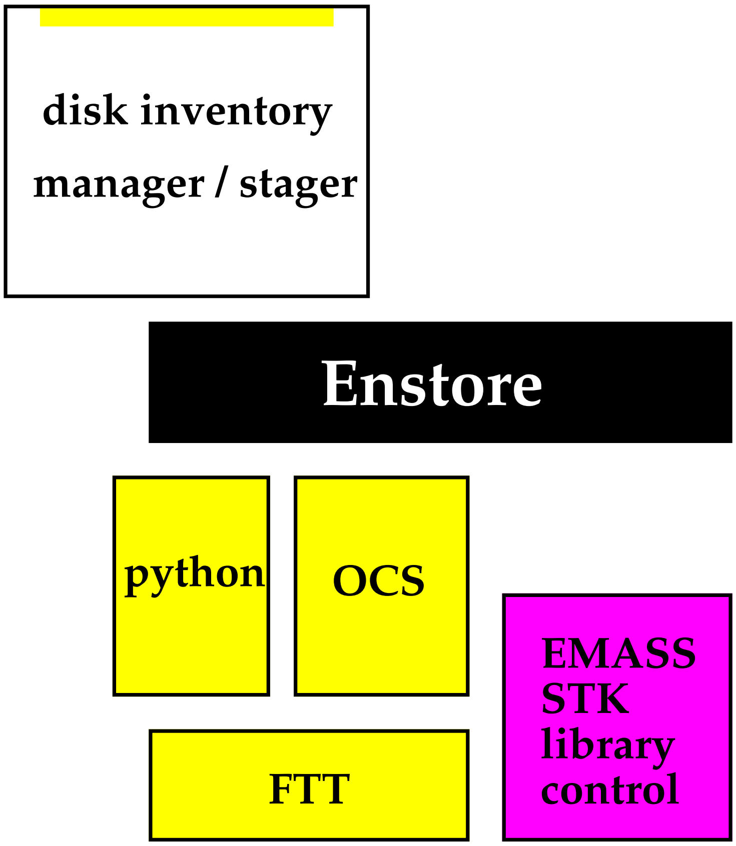
Enstore is being developed for Run II by the High Performance and Parallel Computing department as a replacement for the High Performance Storage System (HPSS) that is installed at Fermilab.

As such Enstore is a network oriented and file based Mass Storage System (MSS). It is not a Hierarchical Storage Management (HSM) System. The system archives files to tape and retrieves them upon request.

It keeps track of its file inventory and allows the user convenient access to this inventory as an NFS directory tree via the Perfectly Normal Filesystem (PNFS) developed at DESY.

Enstore is a large monolithic package. This makes it hard to debug especially since it is written in scripting language.

Enstore writes tapes according to its own standard. Files are encapsulated into cpio archives and written to tape, separated by tape marks. We believe tapes should be written in ANSI standard.


disk inventory
manager / stager
Enstore
python
OCS
EMASS
STK
library
control
FTT

Enstore

A file based tape subsystem is not desirable for data that have a size much larger than that of a single file (TBytes versus GByte).
We can combine files (for instance through tar or cpio) into large files simulating our filesets and thus reduce Enstore to a volume based system.

But keep in mind the reason for our 1 GByte file size choice: the 32-bit filesystem/operating system limits corresponds to 4 GBytes (2 GB in case of ssize_t).

Enstore is being developed around the idea of network attached tape drives. Reliability, performance, and cost factors make direct attached tape drives desirable.

Sending data via IP over the network is costly: about 20% of a Pentium II CPU is required to drive Fast Ethernet (10 MB/s) at full speed. We need 150 MB/s total tape I/O. A high performance network between all the mover and analysis nodes is required.

The developers of Enstore have offered the option of direct attached tape drives but no promise was made to use a direct copy and not IP over the loopback adapter.

Enstore

Enstore would need to be integrated into the CDF data handling system. If we want to manage computing resources well, we need to know the available resources and must be able to control them.

Enstore is designed with its own resource management strategy. Its resource scheduling does not match well our batch based approach and an integration will require significant extra CDF manpower and result in needless complexity.

Two examples:

- For the batch system to schedule tape I/O it needs to know the available tape drives. What was before a static one-to-one configuration of the I/O batch queues on a node is now a dynamic cluster wide coordination for each stageing job.
  User file requests on the command line and the out-of-order processing of Enstore to optimize tape access could make it rather difficult to predict drive availability.

- Volume information is required to organize jobs according to the file layout on tape. This requires an additional interface to the Enstore catalogue (and query of each job) or the volume information of each file needs to be fed back into the CDF Data catalogue.
  This duplicate information has to be kept synchronized between the two databases.

Enstore, Global View

Enstore is a million dollar project. We should all be aware of this.

Enstore is still in the development phase.

Significant involvement from CDF is needed to push for additional features outside the current design that would make it a more usable system for us.

If we don't want to give up tape based analysis at remote institutions we need an mt_tools based system anyhow. According to the Enstore developers it is not intended to be used at remote institutions. Thus it would not eliminate any work/effort but add a second, more complex tape handling system.

Instead of writing a generic tool, Enstore should be customized to the DØ needs and thus reduced in scope.
$\longrightarrow$ save several FTE years.

In the case of a failure of Enstore both Run II experiments will not be compromised.

Enstore, Summary

- **Enstore doesn't fit well into our Run II data handling design.**
- **It is very complex.**
- **It does not offer desired features beyond the mt_tools approach.**
- **It requires additional manpower on CDF side.**
- **It presents a significant risk.**
- **We should not use it to store raw or production output in Run II.**

If you are convinced we should adopt Enstore we should revise the data handling design to a file based tape handling system.

Concluding Remarks

- We expect to record about 1 PByte of data starting in spring of 2000.
- The hardware procurement and commissioning is on schedule.
- We plan to avoid all direct tape drive access by users but spool tape resident data to disk first.
- The data handling architecture is based on a volume based tape subsystem.
- The plan is to use a simple staging system based on the mt_tools package.
- The package has been overhauled in preparation for Run II.
- The mt_tools based tape I/O approach is well tested, with over 100 TBytes of tape I/O (over 20,000 tapes).
- The incomplete component in the tape susbsystem is the interface between OCS and library control software, which we don't have yet.
- I recommend we initiate a re-thinking of the scope of the Enstore project.