TSM
as tape storage backend
for
disk pool managers

Jos van Wezel
Doris Ressmann
GridKa, Karlsruhe

tape backends for dCache

- OSM (DESY)
- Enstore (FNAL)
- HPSS (BNL, CC-IN2P3)
- DMF© (SARA/NIKHEF)
- TSM (FZK/GridKa)

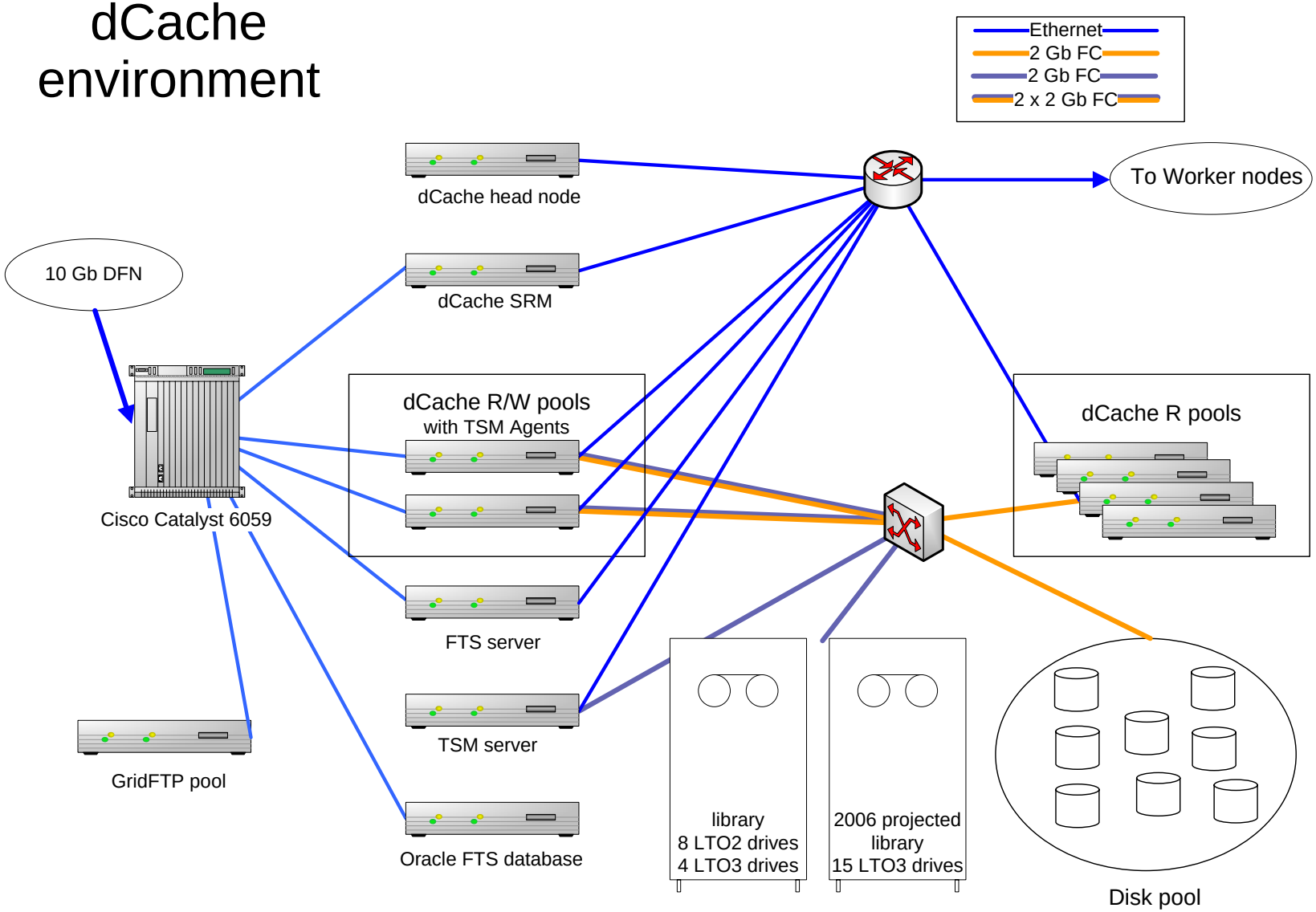
Why TSM

- already in use at FZK
- takes the burden out of tape handling
  - tape/drive replacement
  - generation migration
  - reporting, monitoring
  - etc.
- separates administrator roles
  - storage management
  - tape management
- runs on Linux and i386 HW
- clients (pool nodes) can directly talk to tape via Storage Agents
- documented proven platform
- wide spread use and acceptable price
- application programmers interface

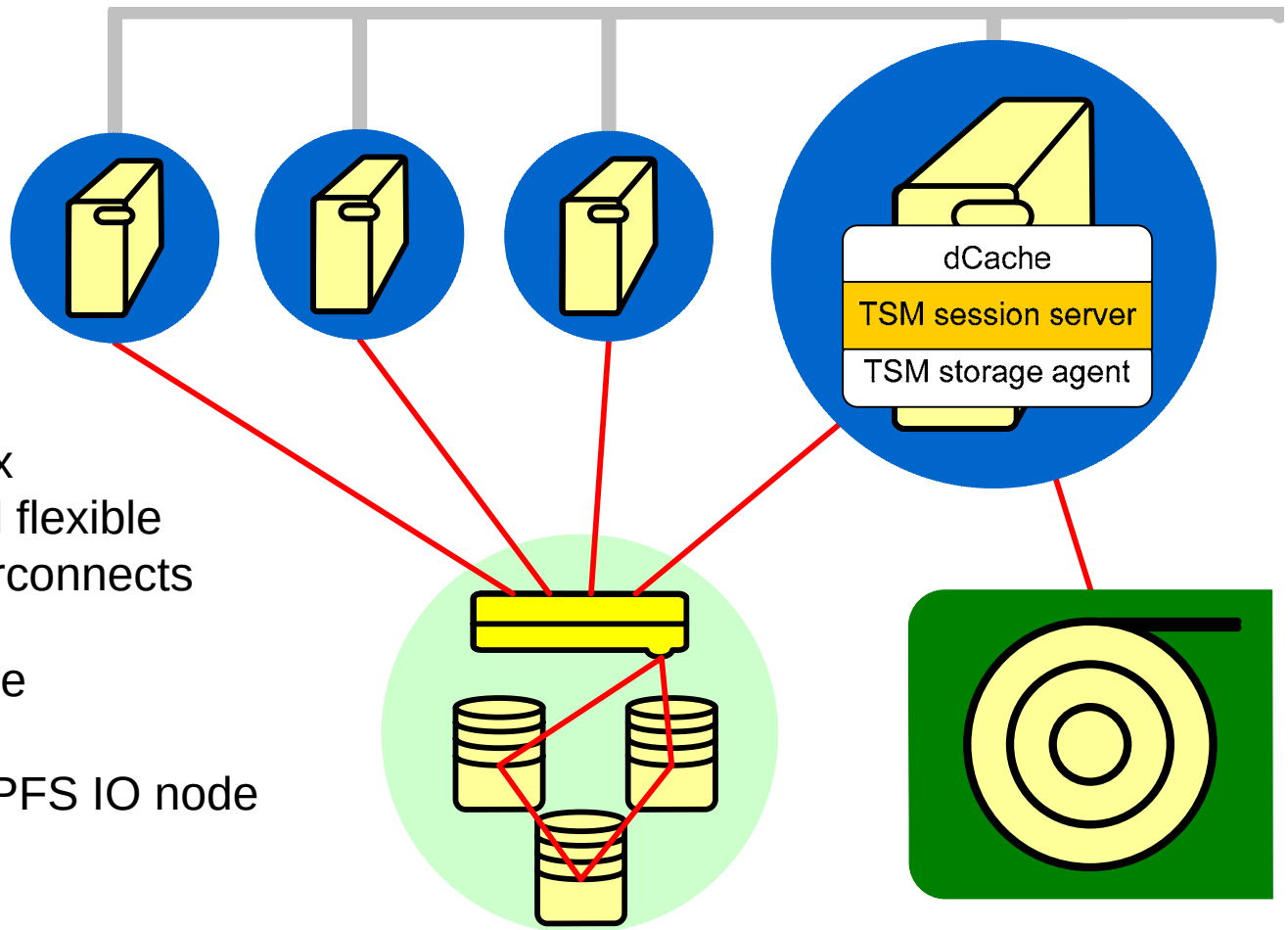
dcache as tape front end

- Fresh data is collected per storage class
- Each storage class queue has parameters to direct the flush-to-tape operation
  - max time between flushes-to-tape
  - max number of bytes not written to tape
  - max number of files not written to tape
  - max number of concurrent writes to tape
- At flush a user defined backend is called
  - tsmcp
  - tss

dCache environment



dCache pools



Consolidated NAS box

- No SAN fabric, still flexible
- Enables other interconnects
  - Iban, 10 GE
- Easier maintenance
- Lower costs
- Can function as GPFS IO node

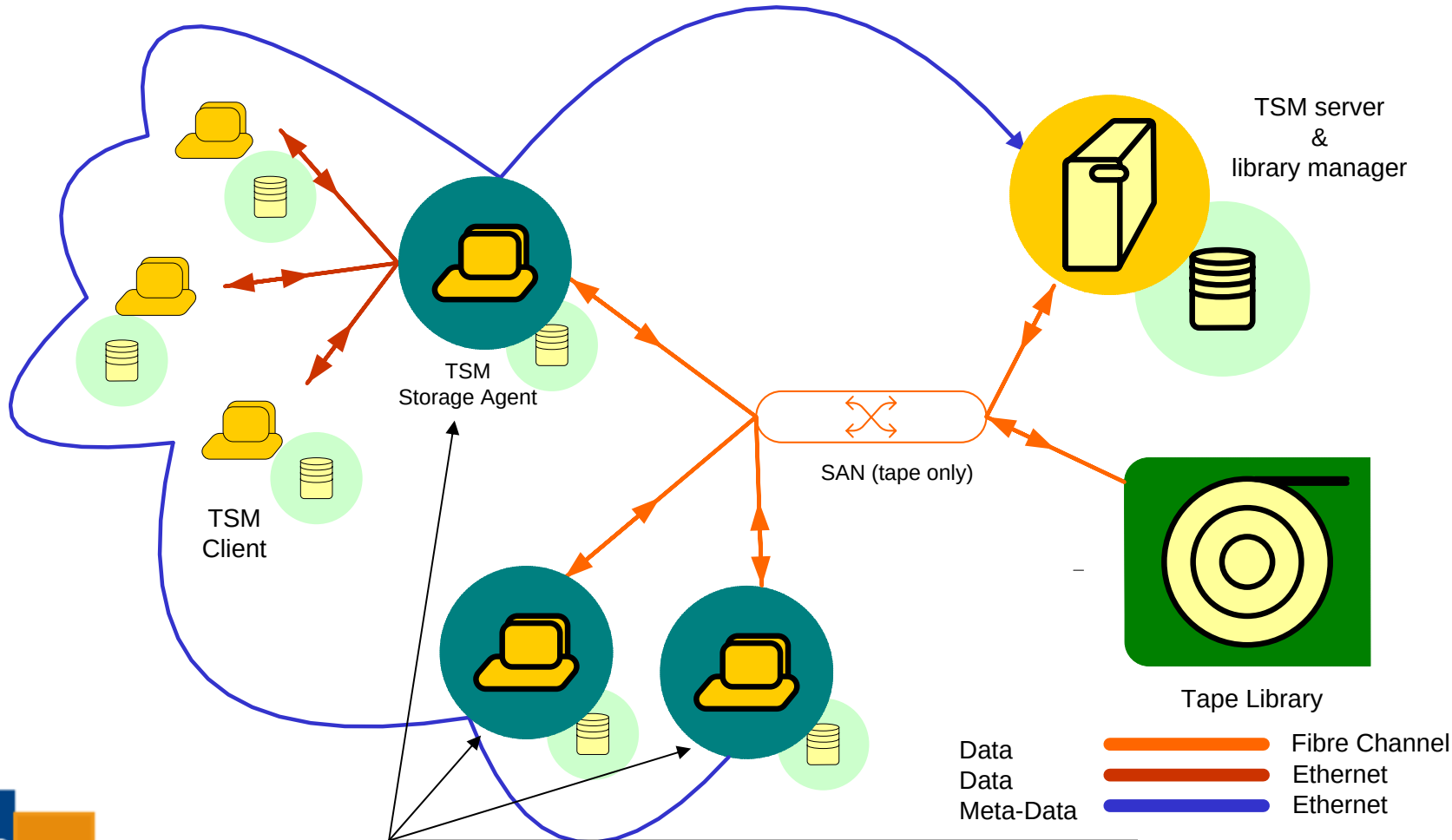
tsmcp

- uses the TSM API
- starts and closes a session for each store to or retrieve from tape. (could be handled in a script that calls the TSM cli)
- Problem with this approach
  - session startup time takes inordinate amount of time
  - On stores: TSM volume selection algorithm starts cartridge juggle. Efficiency nears zero.
  - On retrieves: no control over tape file order

TSM Session Server properties

- Interfaces directly with TSM via its API
  - the API libs come with the TSM software
- Single executable, documentation ‘tss –help’
- Fan out for all dpm to tape activities
  - single session to the TSM server
  - multiple tape flush/retrieve/rename/log/queries
- Runs on the TSM clients, storage agent or on the server proper
- Almost plug-in replacement for the TSM backend that comes with dCache
- Sends different type of data to different tape sets
  - if known from dcache ‘tag’
  - groups data that are likely to be retrieved together
- Queues multiple requests (no state is kept, dpm must re-queue if needed)
- Work in progress (in cooperation with dcache developers)
- Allows to store an exact image of the global name space on tape
  - store the ‘site file name’
  - decoupling of disk pool manager and tape backend
  - needs ‘rename’ support of the dpm

Data flow


TSM server
&
library manager
TSM
Storage Agent
TSM
Client
SAN (tape only)
Tape Library
Data
Data
Meta-Data
Fibre Channel
Ethernet
Ethernet
Storage Agent > TSS > dCache

TSM and TSS in use

- TSM is a viable tape handling system for GridKa
- Promising TSS tests results
  - up to 150 TSM unary database ops/s
  - no cartridge juggling
  - keeps a drive streaming (SAIT at 27 MB/s)
- deployment for SC4 tape challenge
  - 8 LTO3 drives, (8 LTO1 drives)
  - 10 dCache write pools/nodes
  - combined target for this 06/06:
    - to disk: 300 - 500 MB/s
    - to tape: 100 - 150 MB/s
- no known bottleneck in sight
  - clearly the meta data handling at the server does not scale indef.
- clear cut between online and offline storage operations

Future enhancements

Reading

- Sort retrieve order on tape file sequence
  - needs support of the storage manager
  - announced for dCache

Writing

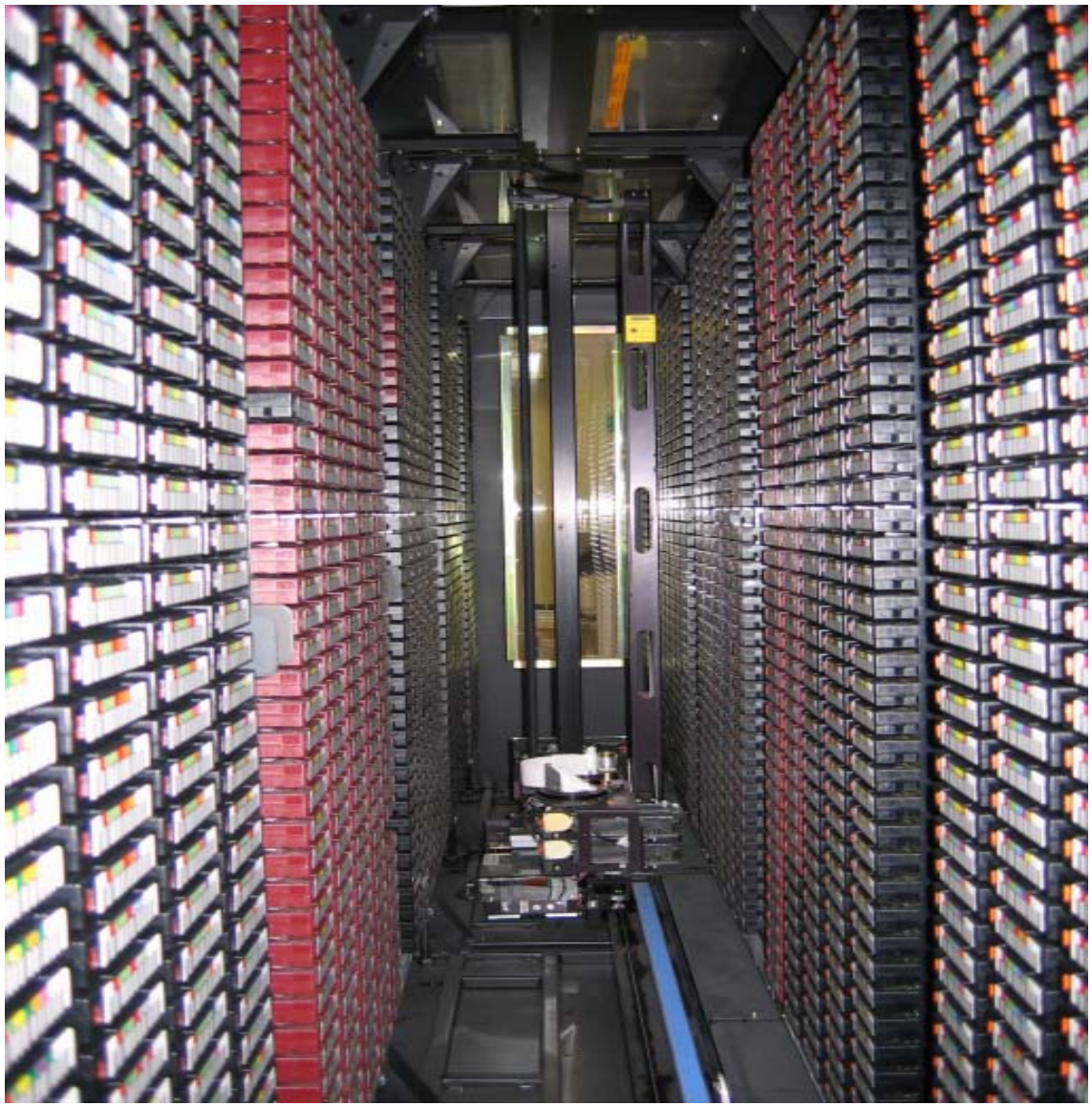
- Improve throughput (LTO3/LTO4)
  - decoupling reads and writes
  - Include sizing estimates on write
  - throttle or stop writes based on node IO load

Support for xrootd

  - can use the same interface

10 Gb networking

  - may use the Ethernet again for tape operations
  - TSS to TSS communication needed



Jos van Wezel April 3,2006