Storage

In HEP,

on the Grid,

and in the Computing Center

Yves Kemp
GridKa-School, HEP session
Karlsruhe 2.9.2009

HELMHOLTZ | ASSOCIATION

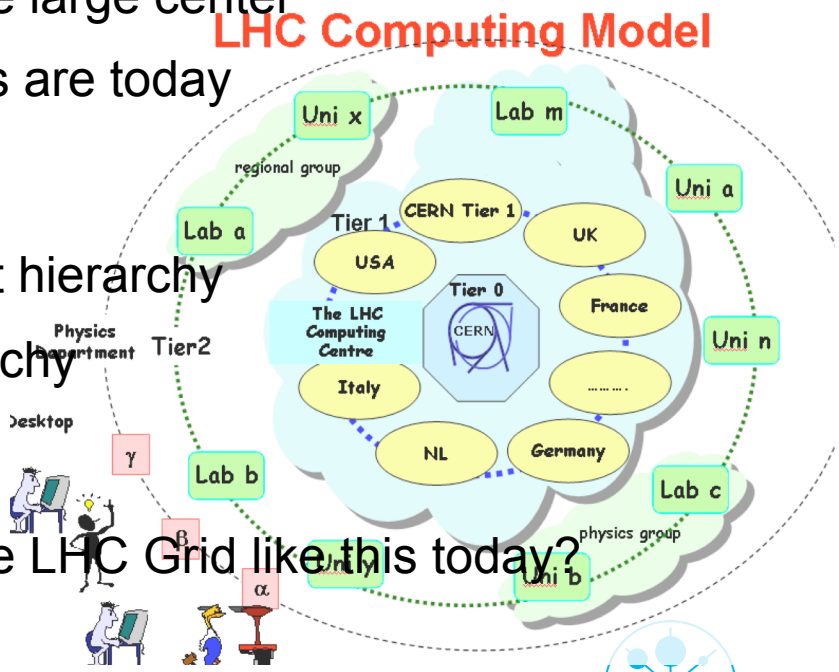
DESY

HEP and storage use

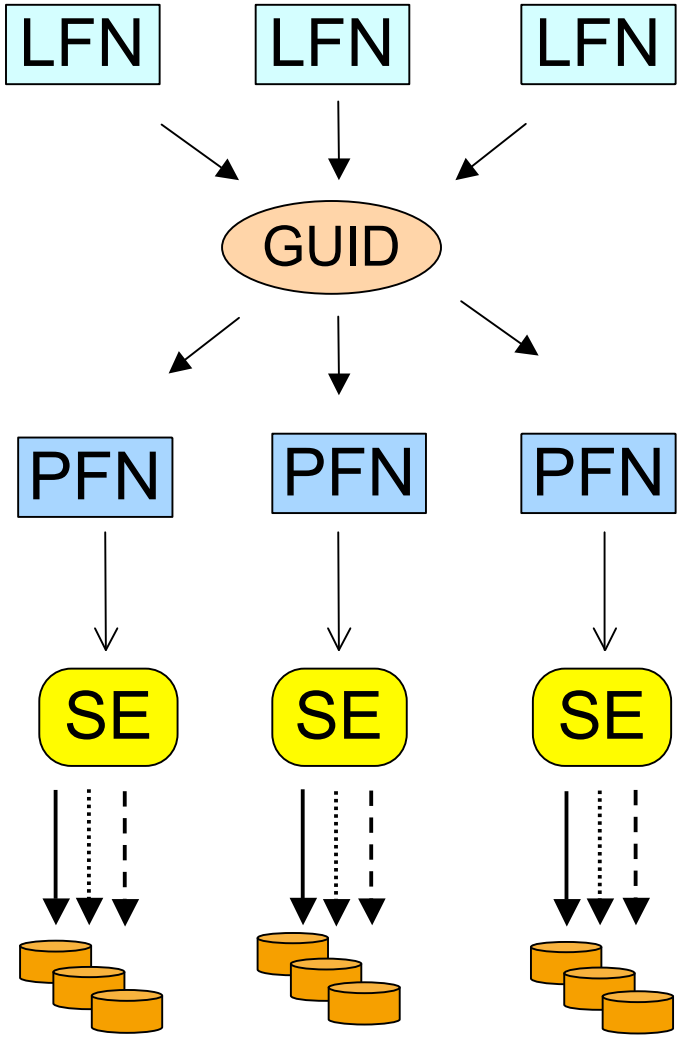
- HEP has always been major storage user
  - Bubble chambers … (HERA, BaBar, Belle, Tevatron, RHIC…)
- With LHC: HEP is going into a new dimension
  - LHC has pushed Data Grid technologies
  - Vendors embracing HEP datacenters, good clients:-)
- Future: Not clear whether HEP still has a key position
  - Data mining, data warehousing, … rapidly growing capacity needs in industry
  - Other science (e.g. XFEL@DESY) similar data rates expected than HEP
  - First lesson: HEP should stick to industry standards!

The LHC data challenge and the LHC Grid

- (you know the numbers, no need to bring them up again)
- Hosting CPUs and storage in one single/very few places impossible
  - Technology: Dimension just far beyond current state of art
  - Network: Is slow and expensive
  - Security: Need at least a second place and copy in case the first one breaks
  - Politics: Did not want to put all money into one large center
  - That was back in 1990ths… guess how things are today
- Two things come together …
  - The Computing Grid with its (theoretically) flat hierarchy
  - A tiered layer of data centers with clear hierarchy
- … to create the LHC Grid
  - An heretical question: Would one still build the LHC Grid like this today?



Coming from the Grid view to the local view



- "Filename" on the Grid
  - Global Unique Identifier (GUID)
  - guid:3a69a819-2023-4400-a2a1-f581ab942044
- Easier with Logical File Name (LFN)
  - lfn:/grid/myexp/kemp/ExitingDataset.dat
  - lfn:/grid/myexp/myboss/DataWithBadDetector.dat
- Physical File Name (PFN)
  - A Path on the SE: also called Storage URL (SURL)
  - /storage/grid/experiments/cms/kemp/ver04/run2342/results/data/file124.dat
  - Files can be replicated to several SEs
- Up to here: Correspondence governed by catalogues
  - Like LFC
- From the PFN to the transport URL (TURL)
  - The transport protocol: (gsi)dcap, gsifp, xrootd
  - The SE (SRM) will tell you
  - … and you will access the data on the hardware….

DESY

The “Ideal” storage: Attributes to storage

- Fast:
  - Fast in getting the meta-data (“ls -l”)
  - Fast in getting the first desired bit / random reads
  - Fast in getting a sustained stream
  - Fast in writing the data
- Huge
- Unique, consistent and easy:
  - Unique namespace (no “/tmp”, “/afs/…”, “/grid/cms/…”, “/home”, “srm://pnfs…”
  - Consistent access methods throughout the whole storage
  - Easy access to the data
- Accessible from everywhere
  - And fast ;-)
- Cheap
  - Purchase
  - Running costs
  - cooling, electricity, space consumption, …

Harry Potter tm Trunk with Dressing Up Set.

Some more attributes…

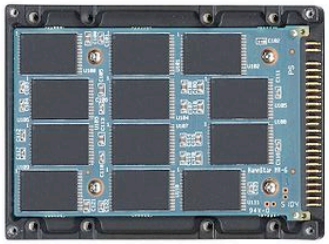
- Secure (data integrity)
  - Authentication and authorization (no one else can temper your data)
  - Backup (even you cannot temper your data by mistake)
  - Robust media/technology & backup (even a disaster cannot temper your data)
- Simple manageability, stable running, good support
  - Little administration costs, good vendor support
  - Little disturbances by downtimes / maintenance
- Migration
  - If a newer / better system becomes available: No vendor lock-in
- Long term availability
  - Of your data
  - Of the storage system
  - Of the protocols
- ….

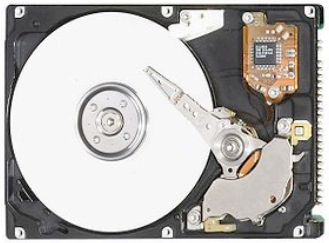
Conclusion:

"The One Ideal" storage does not exist
Compromises, and different products
for different purposes

Some technology: Media

- RAM and NVRAM (e.g. battery powered DRAM (+disks))
  - Yes: RAM-Disks do exist: Databases!
  - Sometimes used as Meta-Data disks for fileservers in HEP
- Solid State Disks
  - Have emerged in the last year, become less and less expensive
  - Serious competitor to Hard-Drives in some future
  - Different access behavior than traditional “spindle disks”
- Hard Disk Drives (with magnetic spindles)
  - Established technology
  - High density, and increasing
  - Streaming performance very good
  - Random access / seek time relatively slow w.r.t. streaming
  - Different connections / qualities: P-ATA, S-ATA, SCSI, SAS, FC, …





Wikimedia Commons

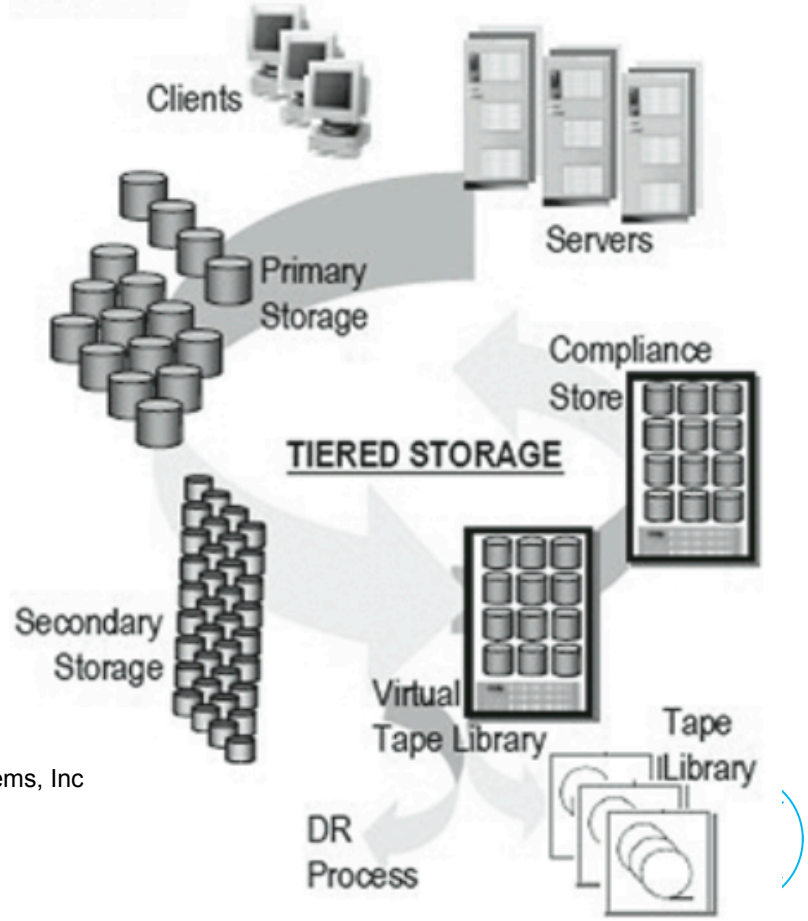
DESY

More technology

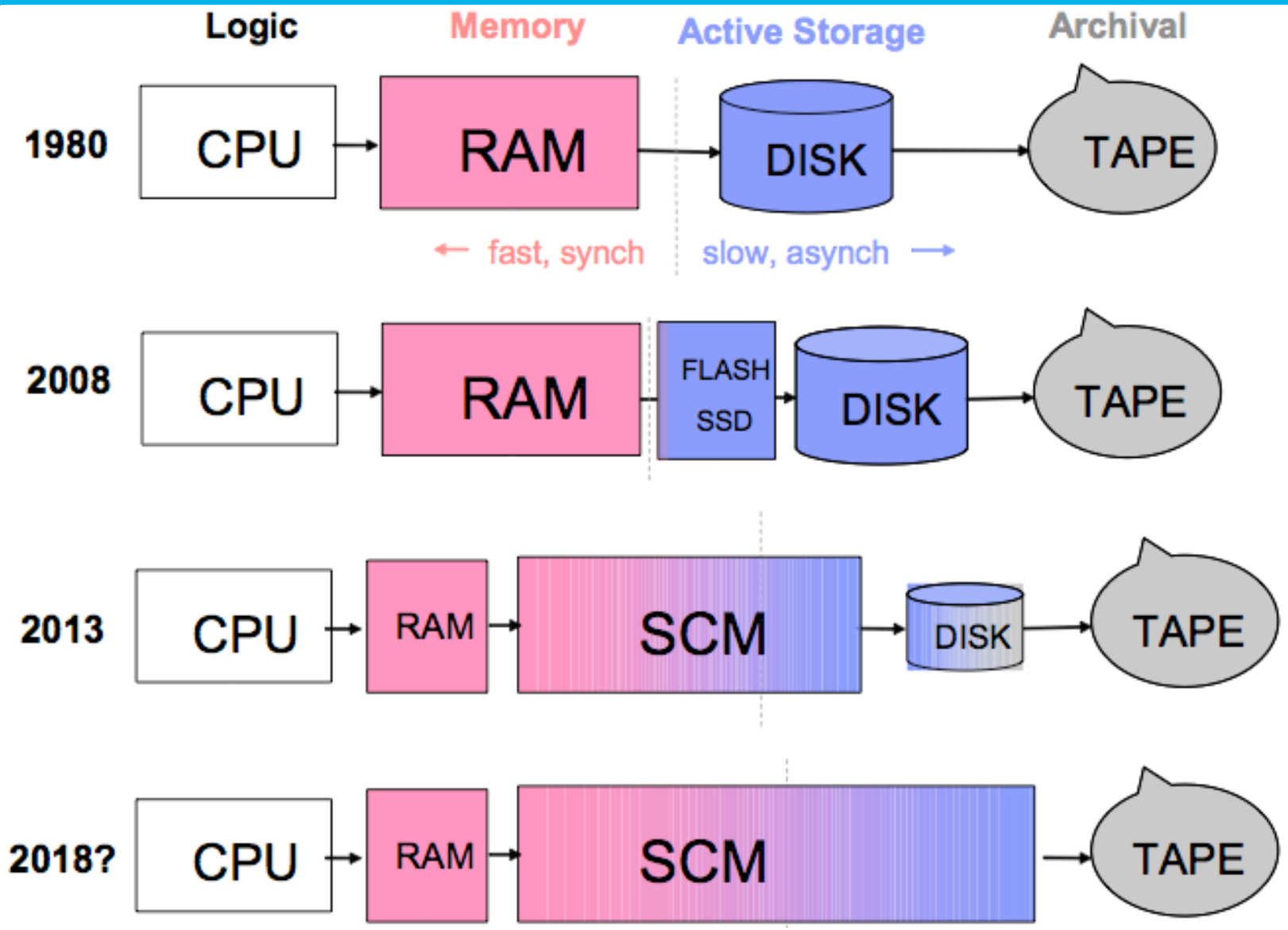
- Tapes
  - “Will disappear soon”: Sentence true since (at least) 10 years :-)
  - And still tape is the working horse for storing data at CERN, FZK, DESY and elsewhere
  - Lowest media cost (~50 EUR / TB), Green-IT (no electricity when not accessed)
  - Best scaling storage system available, difficult to handle (administration, access, ….)
- Optical media (CD, DVD, …)
  - Play only a minor role in large scale data storage

Putting them together:

- **Tiered storage**
  - Migration from active disks to offline storage
  - Automatically, transparent to users



Storage Class Memory



Slide: David A. Pease, IBM

One interesting thing about RAID and data security

- RAID: Redundant Array of Independent|Inexpensive Disks
  - RAID-0: Stripe information among >1 disks
  - RAID-1: Mirror information on 2 disks
  - RAID-5: Capacity of N disks, using N+1 disks, one disk used for checksums
  - RAID-6: Capacity of N disks, using N+2 disks, two disks used for checksums
  - RAID-10/50… combinations of the above
- HDD error rate between 10^{-16} and 10^{-14}
  - Results in reading error every 10-1000 TB
- RAID-5 is secure … unless you store more than ~10 TB:-)
  - RAID-6 is somewhat more secure
- **Absolute security does not exist! Some data will get lost!**
  - Also tapes (I.e. backup) can fail!

Different storage places in the NAF

- OK, what is the optimal workplace on the NAF?
  - No single answer, but you get my personal recommendations for free :-)

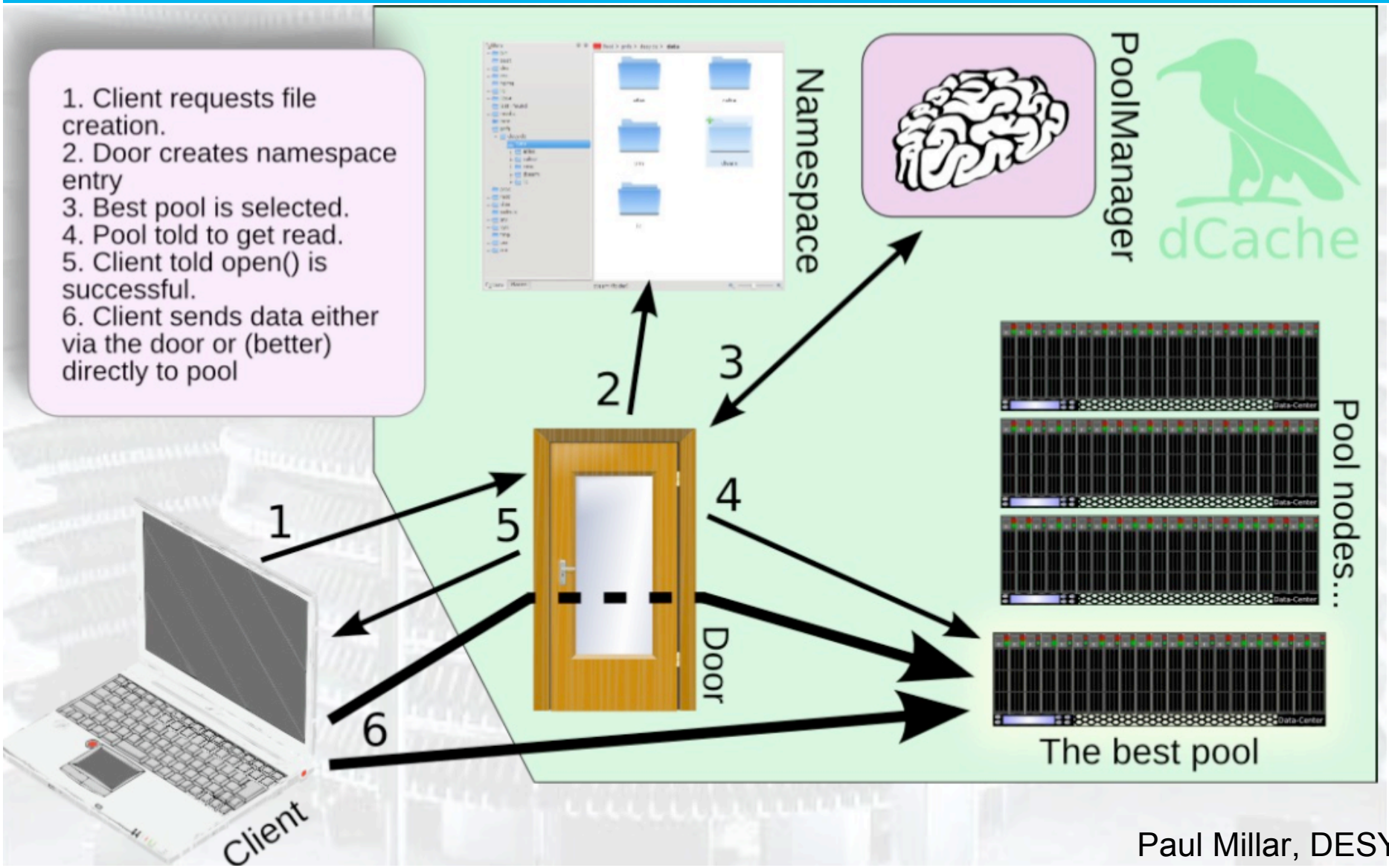
- AFS /afs/naf.desy.de/…: Network file system
  - Login files, small data amounts (like plot.root) (total <1 GB)
  - Source files for code (exclude libs or bins: Check HowTo with your VO admins)
  - There are group volumes for SW releases. Check with your VO admins
  - Compilation can be slow (Atlas-CMT problem), usually OK
  - Not available on Grid-WNs (not directly, and I will not tell you how)
  - Backup
- /tmp: Local file system
  - Is local, quite large, no quota, somewhat fast
  - Cleaned up every 10 days
- /scratch/… (Currently Lustre file system)
  - Fast cluster file system, available everywhere in NAF but not externally, no backup, (currently) no quota or ACLs
  - Using InfiniBand as interconnect, low latency and high bandwidth
  - Currently optimized for large files, bad for unpacking source code and compilation
  - Useful for temporary storage of “hot data”
  - Storage of often-used personal NTuples
- dCache / SE
  - Central (Grid) Import/Export system, well integrated into experiment’s workflows
  - Large data sets, shared by many people and accessible from everywhere
  - Not “filesystem-like”: No compilation etc.
  - Backup / Archive possible (not done yet)

dCache in a nutshell

- dCache was introduced as a disk cache for tapes
  - Combine large but slow tape systems with small but fast file servers
  - Remember the “tiered-storage” plot???
- Today can manage up to 10 PB of data
  - But also “Tier-3-like” installations, with O(100 TB) of data
- Speaks many languages
  - (grid-)ftp
  - dcap, gsidcap
  - Xrootd protocol
  - http
  - SRM as a meta-language

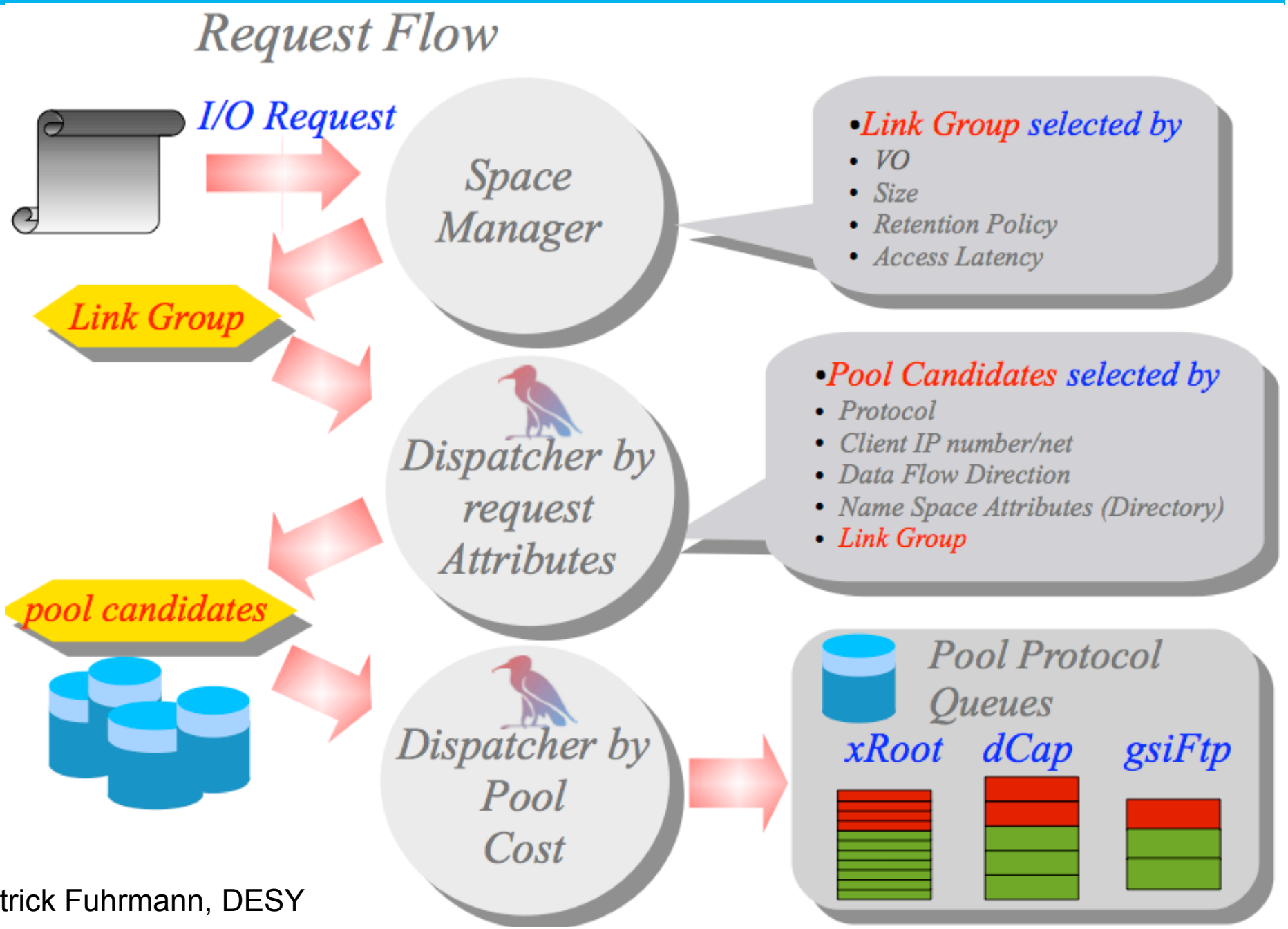
- Other systems (CASTOR, DPM, …) have similar setup

Workflow : writing files



Paul Millar, DESY

dCache a coarse view inside



SRM: Storage Resource Management

- Network protocol providing abstraction layer

- What SRM does:
  - Negotiates transfer proto
  - File pinning / unpinning
  - Space management
  - Name-space operations
  - Permission management

- What SRM does not do:
  - Data transfer
  - Configuring data placement / policy engines
  - Provisioning

http://iris-ict.eu/joomla/images/stories/storage7.jpg

- Two file attributes:
  - Access Latency is: online, nearline, (offline)
  - Retention Policy is: replica, (output), custodial.

dCache: Pools and Doors

- Pools: Hold the data
  - Poolgroup: Group of pools:-) (e.g. all pools from mcdisk, data09disk,…)
  - Poolnode: A computer with pools on it (usually fileserver, or attached to SAN)
  - Replicates: One file can have several copies on different pools (of the same poolgroup). Can be done automatically, useful for increasing performance
- Doors: Connect you with your data
  - Several doors, for each protocol: (gsi)dcap, (gsi)ftp, http(s), xrootd, …
  - Can have more than one / protocol
  - Ask SRM for the door with protocol X → You will get the best matching door
  - You communicate with this best matching door …
- Getting/Putting the data
  - Either your client is redirected to the pool, and gets/puts data directly
  - ((Or the access goes via the door, good for firewalls, bad for speed))

DESY

Authentication/Authorization

SRM & gridftp
Using Certificate

Desktop/DESY-WGS
Dcap: UID2/GID2

Grid WN
Gridftp: Certificate
Dcap: UID1/GID1

NAF WGS&Batch
Dcap: UID3/GID3

This will fail: Four different identifies!
One single authentication/authorization:
Based on Certificate/VOMS

- Protocols
  - Gridftp (same as before)
  - Gsidcap: Same as dcap, but with GSI authz
- E.g. ROOT supports gsidcap
- Meta-Data handling (e.g. file browsing)
  - /pnfs/ needs dcap (not gsidcap!)
  - dcTools developed at DESY by summer student Malte Nuhn
  - On NAF: ini dctools -> dcls -l /pnfs/desy.de/ilc
- SRM tools also OK, but slower
- http based solution under development

Speed discussion

- Hepix storage task force: dCache similar speed than other products
  - DPM, XROOTD server and dCache: No seizable difference in performance
  - Will take dCache as an example in the following
- dCache has movers, and they might get queued if too many
  - To protect the system against overload
  - You might have to wait :-) (We see this, and can optimize things up to some point)
- SRM is slow. People know this. Unfortunately, there is not much we can do about it…
- Communication overhead (doors, pools,…)
  - (~0.5 s communication / file open) + (~0.5 s GSI security / session)
  - dCache developers try to lower both overheads
- Data transfer is very fast
  - In streaming mode, near to wire speed ((gsi)dcap and gsiftp)

How fast a system must be?

- Example math:
  - One job: 10 events/s
  - One event (AOD): 150 kByte
  - →1,5 MByte/s/job (CPU limited….)
  - 5 million events per dataset
  - Want to compute this in 1 hour
- $5*10^7$ / 10 / 3600 = 140 jobs in parallel
- 140 jobs * 1,5 MByte = 210 MByte/s aggregate bandwidth
  - We have tested Lustre with 500 MByte/s (one server!)
  - dCache: Data distributed over different pools. Now: ~100 MByte/s/poolnode, soon: up to 1 GByte/s/poolnode
- Numbers can/will/should change!

M. Schumacher on Ferrari, 2005
Wikimedia commons

The Last Slide

No conclusion here (except that this is an incomplete talk with a lot of personal opinions)

Any questions? Suggestions?

One question to you: We are always looking for benchmark applications. If you got a physics analysis application and are willing to spend some time rerunning your app against several storage technologies and different configurations, please contact us!

Yves.Kemp@desy.de