Large DBs on the GRID

Getting performance with wild HEP data distribution

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Outline

- The problem
  - The structured files
- Generic system architectures
  - Offline part, jobs, data, bookkeeping etc.
- Catalogues and metadata repositories
- Performance is important
- Jobs to the data or data to jobs?
- The xrootd way
- Direct access in WAN/LAN
- Storage cooperation

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

The problem

- HEP experiments are very big data producers
- The HEP community is a very big data consumer
  - Analyses rely on statistics on complex data
  - Scheduled (production) processing and user-based unscheduled analysis
- Performance is a primary factor
  - The infrastructure must be able to guarantee access and functionalities
  - The softwares must use the infrastructures well

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Structured files

CERN IT Department

- In the present times the computations are organized around very efficient “structured files”
  - See the presentation by Rene Brun
  - Which contain homogeneous data ready to be analyzed
  - At the various phases of HEP computing
  - Centrally-managed data processing rounds create the “bases of data” to be accessed by the community
  - One site is not enough to host everything and provide sufficient access capabilities to all the users

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

The minimalistic workflow

CERN IT Department

- In general, an user will:
  - Decide which analyses to perform for his new research
  - Write the code which performs it
    - Usually a high level macro or a "simple" plugin of an experiment-based software framework
  - Ask the system about the data requirements
    - Which files contain the needed information
  - Ask another system to process his analysis
  - Collect the results

Typically an experiment-based metadata repository and/or file catalogue

Typically the GRID (WLCG) or PROOF

This will likely become also his own computer as the hw performance increases and the sw allows it efficiently

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

Databases

CERN IT Department

- A DB as a “structured, heterogeneous base of data”
  - From the end of the 90s it became clear that putting **everything** in a relational or object-based orthodox DB was not a good choice
    - Versatility and expressive power comes at the expenses of performance, ease of maintenance and scalability
    - An insufficient performance also can cause big frustration and big system instabilities
    - Difficult to scale them at these extreme levels
      - Also the cost does not scale linearly
  - In practice, HEP data is typically composed by two sections:
    - The structured files, optimized for access performance
    - The metadata about them
      - A searchable repository of descriptions of what they contain
      - Support information

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

Databases and storage

CERN IT Department

- In simple words, the need is:
  - A bookkeeping repository able to do searches
    - With a big granularity, we are interested in big “datasets”, not in single events randomly spread somewhere
    - A relational/OO DB is perfect for that (order of 10^8 files per experiment)
  - An efficient and scalable data access system
    - ALL the performance of the hardware (disk pools) must be reachable
      - And scale linearly with it
    - If an app computes 15MB/s the disk will have to ‘see’ 15MB/s for it, not more.
    - Several processes hitting the same (remote) disk should not make its performance worse
      - In other words: the disk throughput must be exploited and it should scale
    - “Computes” means “computes”, not “transfer” or “consume”. This is very important.

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

The GRID

- In its biggest simplification we can consider the GRID service (WLCG) as a huge batch system
  - Distributed worldwide
  - The user submits a job to process
  - This job is assigned to some worker node in some site
    - Through some mechanisms
    - E.g. nowadays a lot of user groups is using the “pilot job” paradigm
      - What is sent and scheduled is a “job agent”, which checks the environment first and then “pulls” the true job from an experiment-based service

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

The Data Management

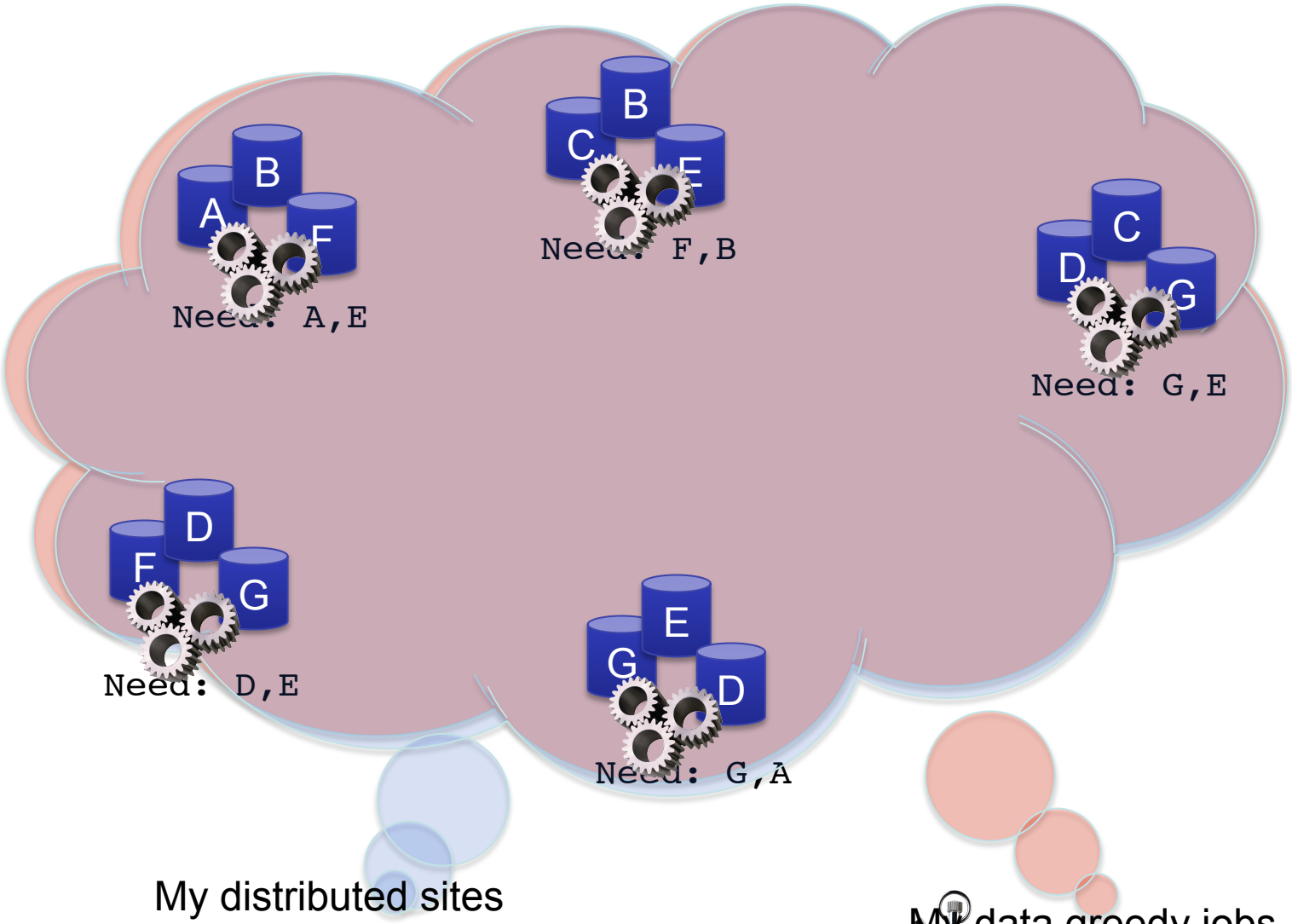
CERN IT Department

- Files and datasets are stored into Storage Elements, hosted by sites
  - The decision is often taken when they are produced
- Processing jobs are very greedy
  - Up to 15-20 MB/s
- The GRID machinery (ev. Together with some service of the experiment) decides where to run a job
  - The service can also be human-based (!)
- Matching the locations of the data with the available computing resources is known as the “GRID Data Management Problem”.

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

An example



My distributed sites
With pre-filled storage
With computing farms

My data greedy jobs

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Where to put the data?

CERN IT Department

- Structuring the production/processing is a very hard problem
- There are basically 2 philosophies, plus a third addition:
  - Send the job in the same place as the data files it needs
    - Easy if there's only one very big input file for a job
    - Otherwise it degrades to a complex matching problem
  - Send the data where the job goes
    - Practicable if moving data around is simple and efficient (The GRID officially uses FTS/SRM for this)
    - Eventually doing static data/site partitions, or putting in a site everything which will be needed by the next processing round made local
      - Quite unpractical but inelegantly doable
    - In its extreme form the data files are copied locally to the WN
- Or:
  - Send the jobs to the data if possible
    - Without fully enforcing the proximity
    - The bulk of the load will be in the LAN
  - And then access directly the data, Web-like.
    - Very recent technological possibility (historically was quite inefficient for random data processing through WAN)
    - Better for common data sets accessed by many jobs
      - In general, for things which are unpractical to replicate everywhere

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

Where to put the data?

CERN IT Department

- As an example, ALICE chose to:
  - The data access uses a unique file-system-like technology, called XROOTD
  - (Sub)jobs are sent (automatically) to the site which hosts the main data file(s) they need in its Storage Element (SE)
  - Conditions data (many complex files needed by many jobs, always the same for a run) are accessed via WAN from 1-2 SEs accessible from everywhere

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

LFNs and PFNs

- Logical File Name: a filename as it is seen by the application
- Physical File Name: the filename as it is physically stored
  - E.g. with the mount point prefix before
  - Or a completely different file name
- It's a powerful idea to implement location transparency
- This distinction can make things easier or difficult.
  - A simple rule (e.g. a local directory prefix) can make it easier to aggregate servers into clusters, exposing a common name space
  - An older idea was to completely detach them, and assign a number as a PFN
  - Doing so, the association pieces must be kept in a DB

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

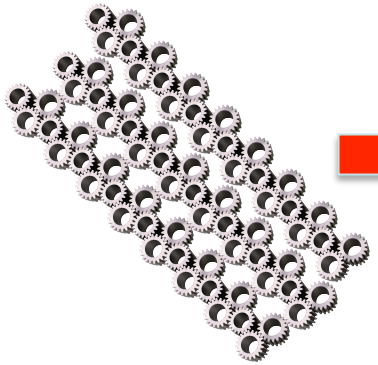
The "Line of Fire"

- A very common pitfall: "we can translate all the requests towards the SE as they come, so we can implement a relational DB-based system which spreads the load through N data servers"
  - In practice, it stores the exact location(s) of each file
  - It may work in principle, but it may be as demanding as serving the data. Very difficult to accommodate in sites with varying service levels.
  - Also, such an external system cannot reflect unexpected changes (e.g. a broken disk)

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

The "Line of Fire"

- Designs, informally, the architectural position in the front of a storage element, directly exposed to the load coming from the processing jobs
  - Very delicate position where to put any system
  - The transaction rate (open) can get up to 2-3K per second per site
  - Eventually, the load will accumulate until…



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Xrootd and Scalla

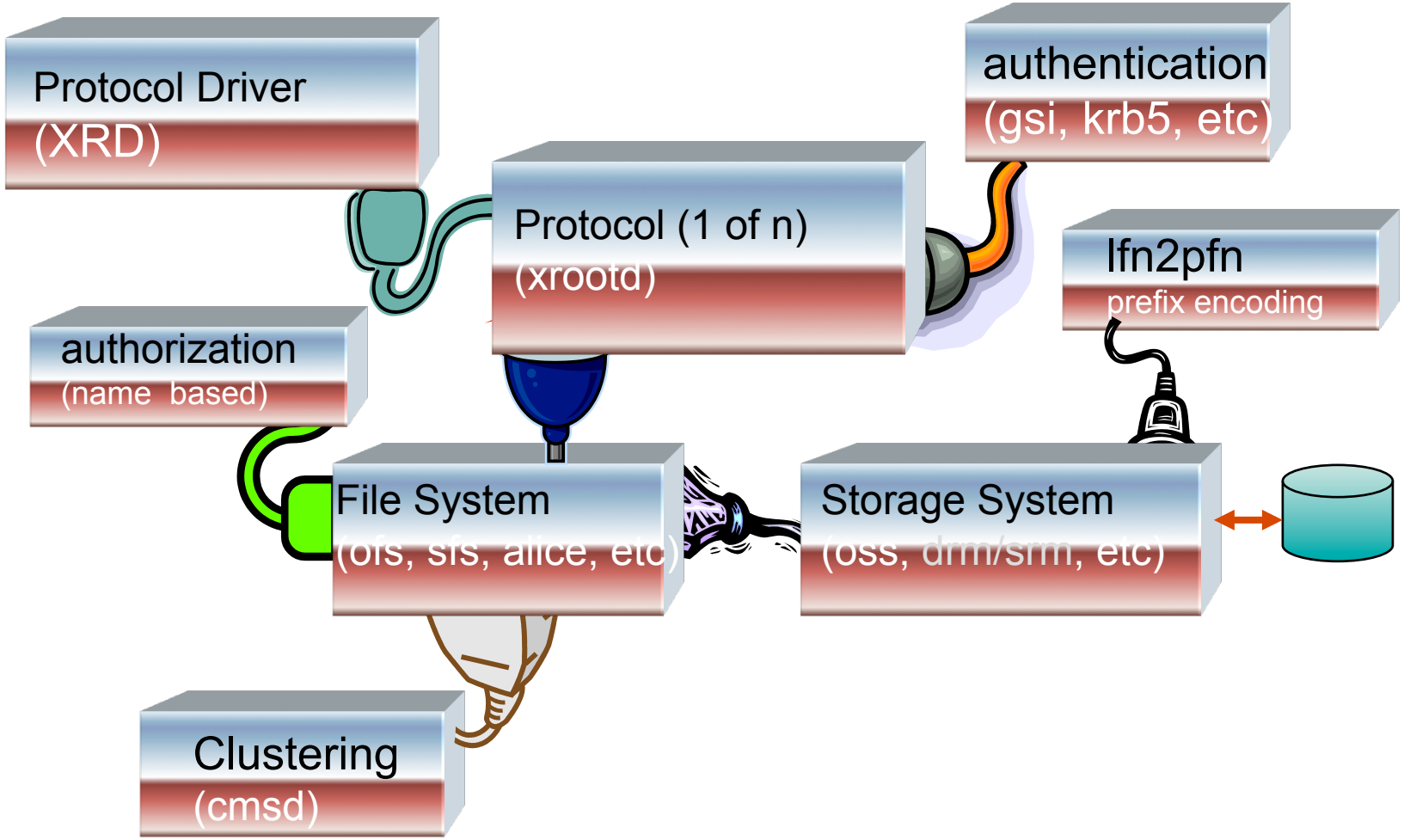
CERN IT Department

- In 2002 there was the need of a data access system providing basically:
  - "Indefinite" scaling possibility
  - Maniacally efficient use of the hardware
  - Accommodate thousands of clients per server
  - Great interoperability/customization possibilities
- In the default config it implements a non-transactional distributed file system
  - P2P elements to aggregate up to 262K servers through LAN/WAN
  - Sacrifies a few POSIX constraints (e.g. distributed locks) to get scalability and performance
  - Protocol and client/server designed for high performance in data access
  - Not linked to a particular data format
    - Thus matching very well the ROOT requirements

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

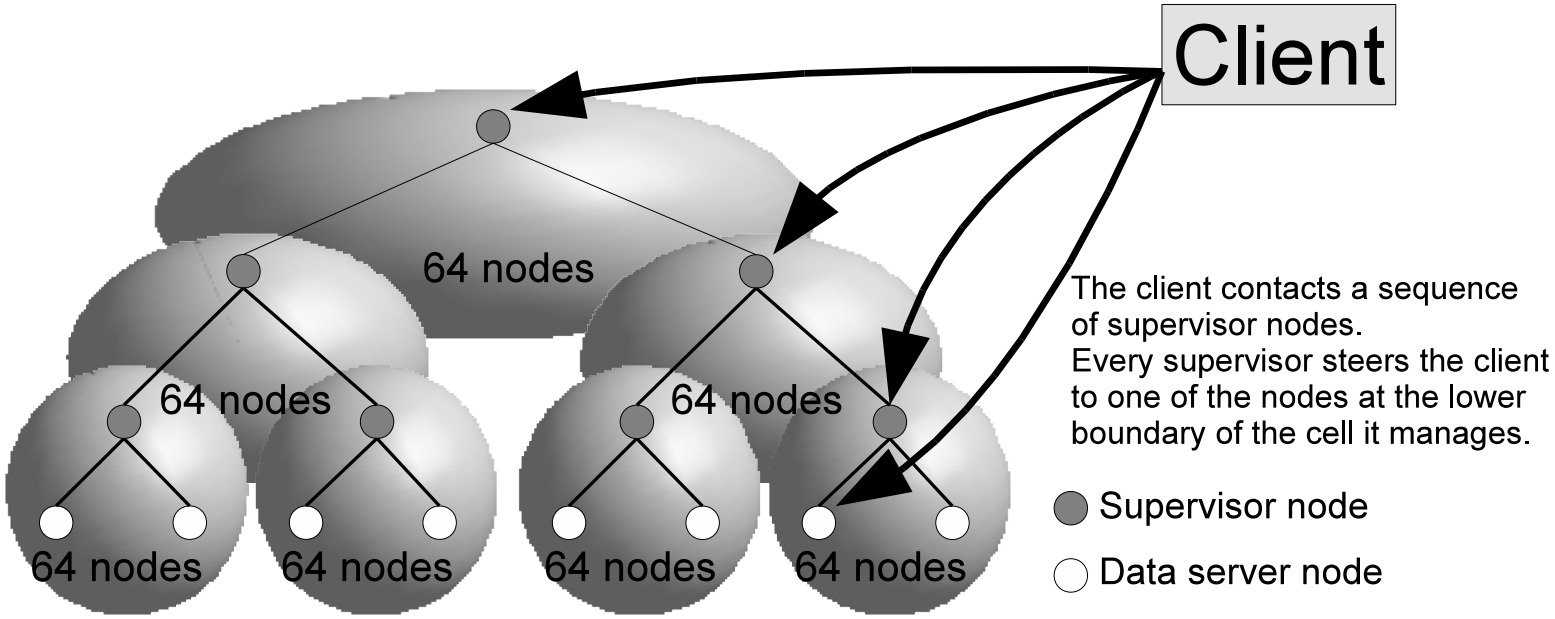
CERN

Plugins



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Cells



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Performance in data access

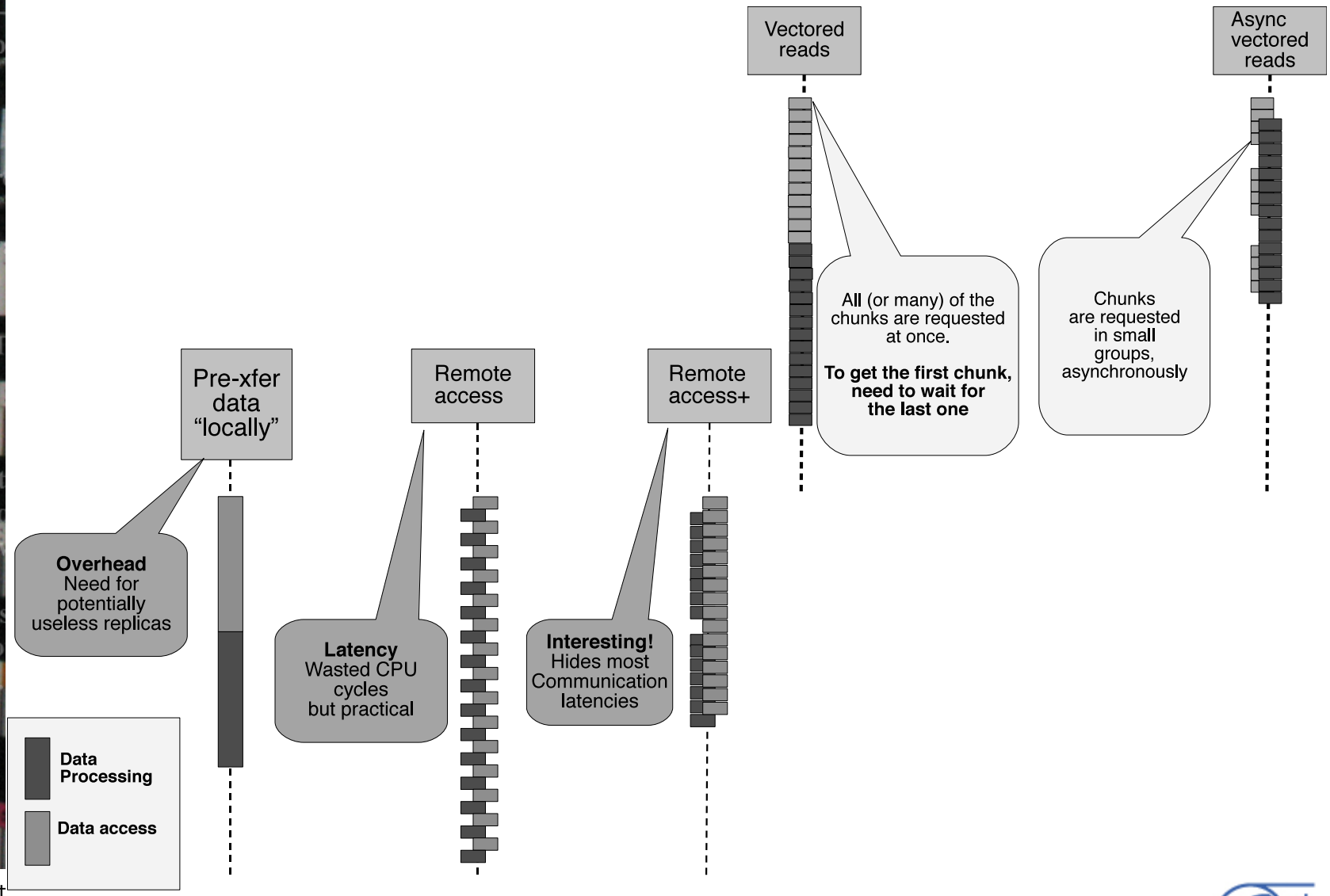
CERN IT Department

- The Scalla/xrootd project puts great emphasis in performance. Some items:
  - Asynchronous requests (can transfer while the app computes)
  - Optimized vectored reads support (can aggregate efficiently many chunks in one interaction)
  - Exploits the ‘hints’ of the analysis framework to annihilate the network latency
    - And reduce the impact of the disks’ one by a big factor
  - Allows random-access-based data access through high latency WANs

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

Access strategies



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

WANs are difficult

CERN IT Department

- In WANs each client/server response comes much later
  - E.g. 180ms later
- With well tuned WANs one needs apps and tools built with WANs in mind
  - Otherwise they are walls impossible to climb
    - I.e. VERY bad performance… unusable
  - Bulk xfer apps are easy (gridftp, xrdcp, fdt, etc.)
  - There are more interesting use cases, and much more benefit to get
- ROOT has the right things in it
    - If used in the right way

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

What can we do

CERN IT Department

- Basically, with an XROOTD-based frontend we can do 2 things via WAN:
  - Access remote data
  - Aggregate remote storages
    - Build an unique storage pool with subclusters in different sites
    - No practical size limits, up to 262K servers in theory
    - No third-party SW needed
    - So, we don't need to know in advance where a file is…
    - We just need to know which is the file we need
- There are pitfalls and things to consider
  - But a great benefit to get as well
  - Let's see what's possible and some of the new ideas

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

Exercise

- Caltech machinery: 10Gb network
- Client and server (super-well tuned)
  - Selectable latency:
    - ~0.1ms = super-fast LAN
    - ~180ms = client here, server in California
      - (almost a worst case for WAN access)
- Various tests:
  - Populate a 30GB repo, read it back
  - Draw various histograms
    - Much heavier than the normal, to make it measurable
    - From a minimal access to the whole files
    - Putting heavy calcs on the read data
    - Up to reading and computing everything
      - Analysis-like behaviour
  - Write a big output (~600M) from ROOT

Thanks to Iosif Legrand and Ramiro Voicu

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Exercise

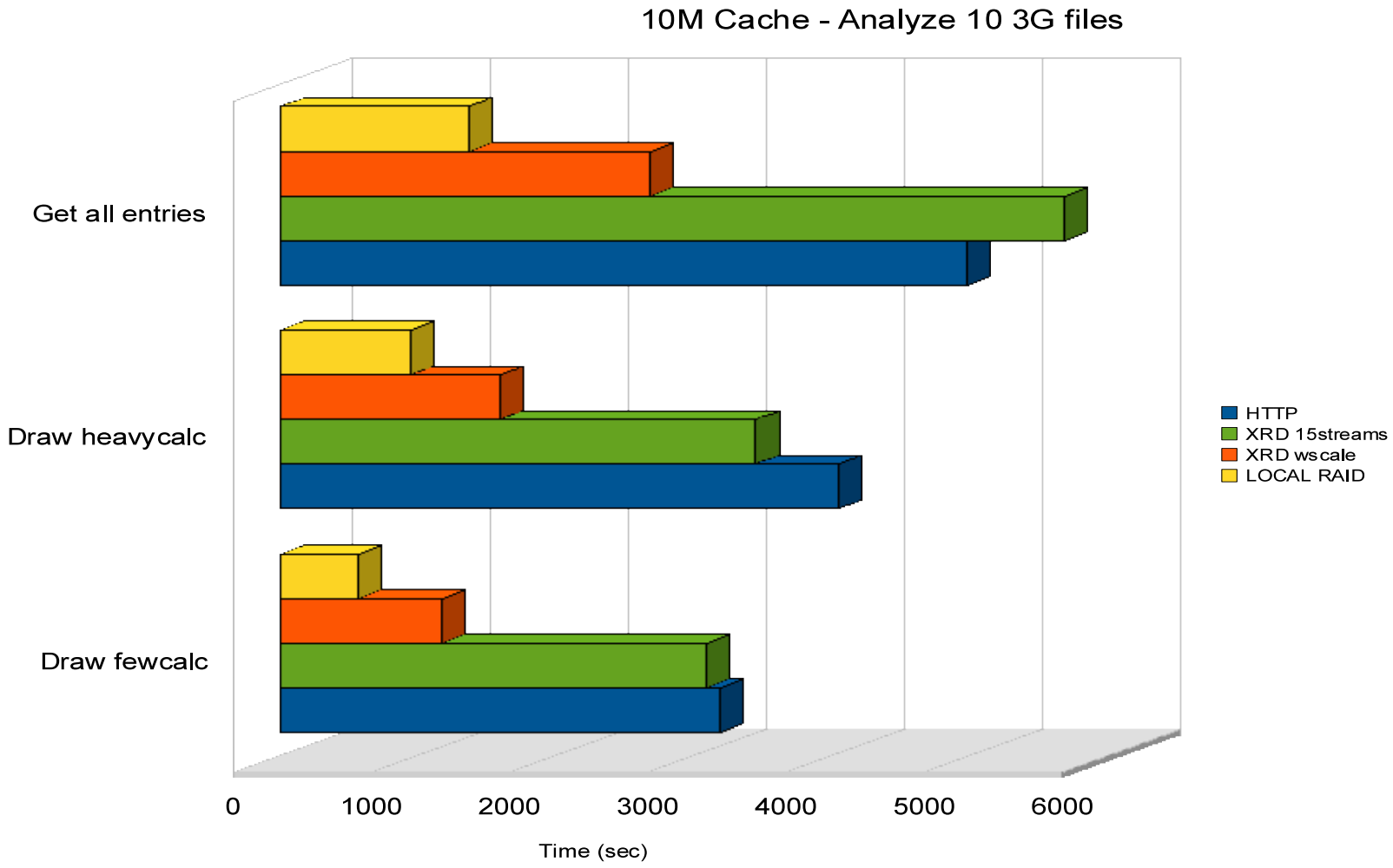
CERN IT Department

- This is not a “Bandwidth race”
  - The goal is not to fill the 10Gb bandwidth
    - Others are interested in that, and do it very well
- We wanted to see:
  - Can we use all this to live better with data?
  - How does a normal task perform in LAN/WAN?
    - In a measurable and stable WAN environment
- Local disk vs XROOTD vs HTTP (Apache2)
  - Why HTTP? Because it is just the most difficult opponent:
    - Efficient (LAN+WAN) and lightweight
    - No bandwidth waste *
    - Very robust server (but not enough OK for HEP data mgmt)
    - Well integrated in ROOT, works well (except writes, not supported)

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

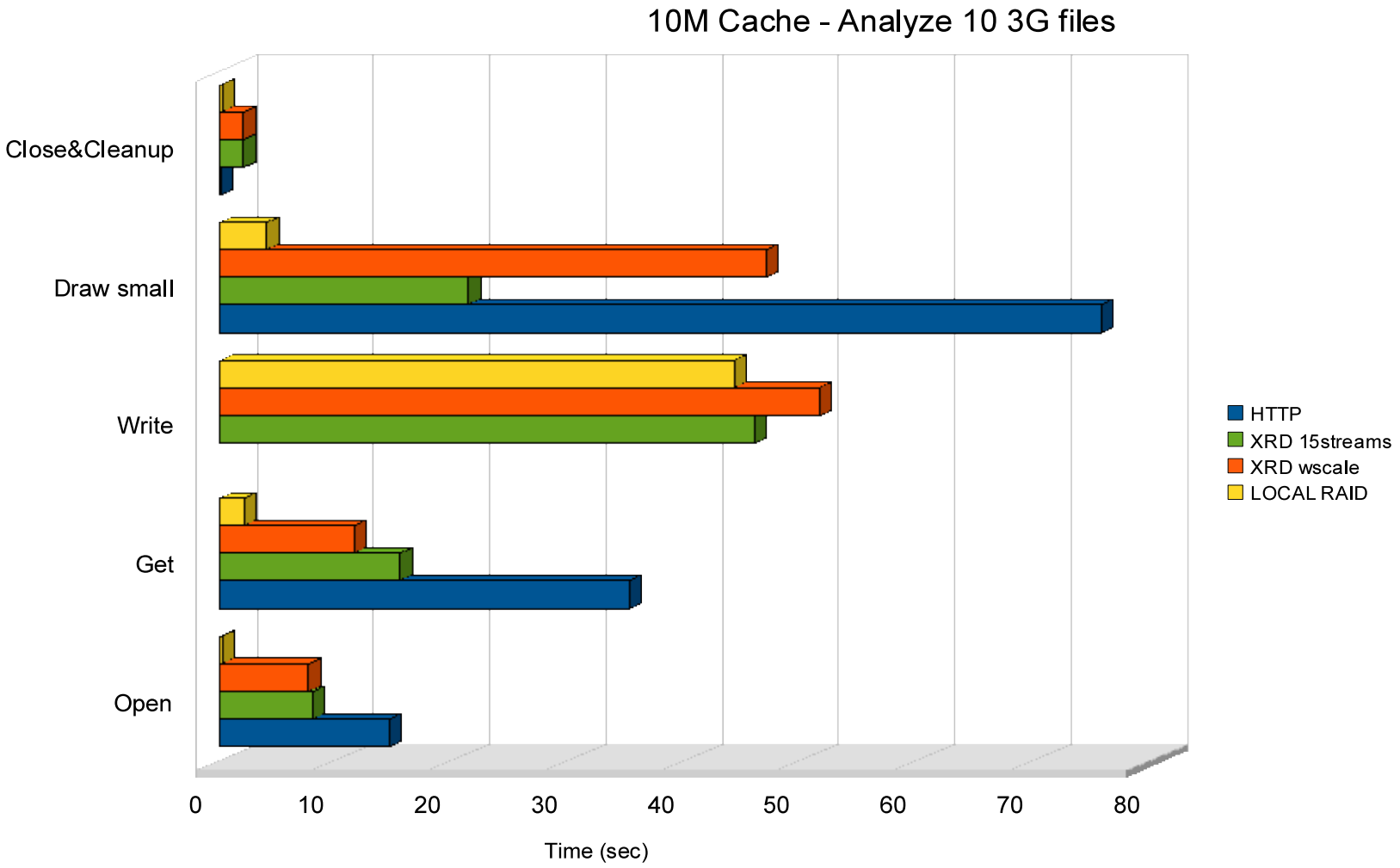
** See the talk about gpfs/xrootd (on Thu) and the Lustre analysis by A.Peters (ACAT08)*

10Gb WAN 180ms Analysis



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

10Gb WAN 180ms Analysis



An estimation of Overheads and write performance

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Comments

- Things look quite interesting
  - BTW same order of magnitude than a local RAID disk (and who has a RAID in the laptop?)
  - Writing gets really a boost
    - Aren't job outputs written that way sometimes?
    - Even with Tfile::Cp
- We have to remember that it's a worst-case
  - Very far repository
  - Much more data than a personal histo or an analysis debug (who's drawing 30GB personal histograms? If you do, then the grid is probably a better choice.)
  - Also, since then, the performance of the sw increased further by a big factor

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Comments

- As always, this is not supposed to substitute a good local storage cluster
  - But can be a good thing for:
    - Interactive life, multicore laptops
    - Saving the life of a job landed in a place where its input is not present
    - Federating relatively close sites…
      - E.g. one has the storage, the other has the WNs
    - An user willing to debug its analysis code locally
      - Without copying all the repo locally
    - Whatever could come to the mind of a WWW user

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

A nice example

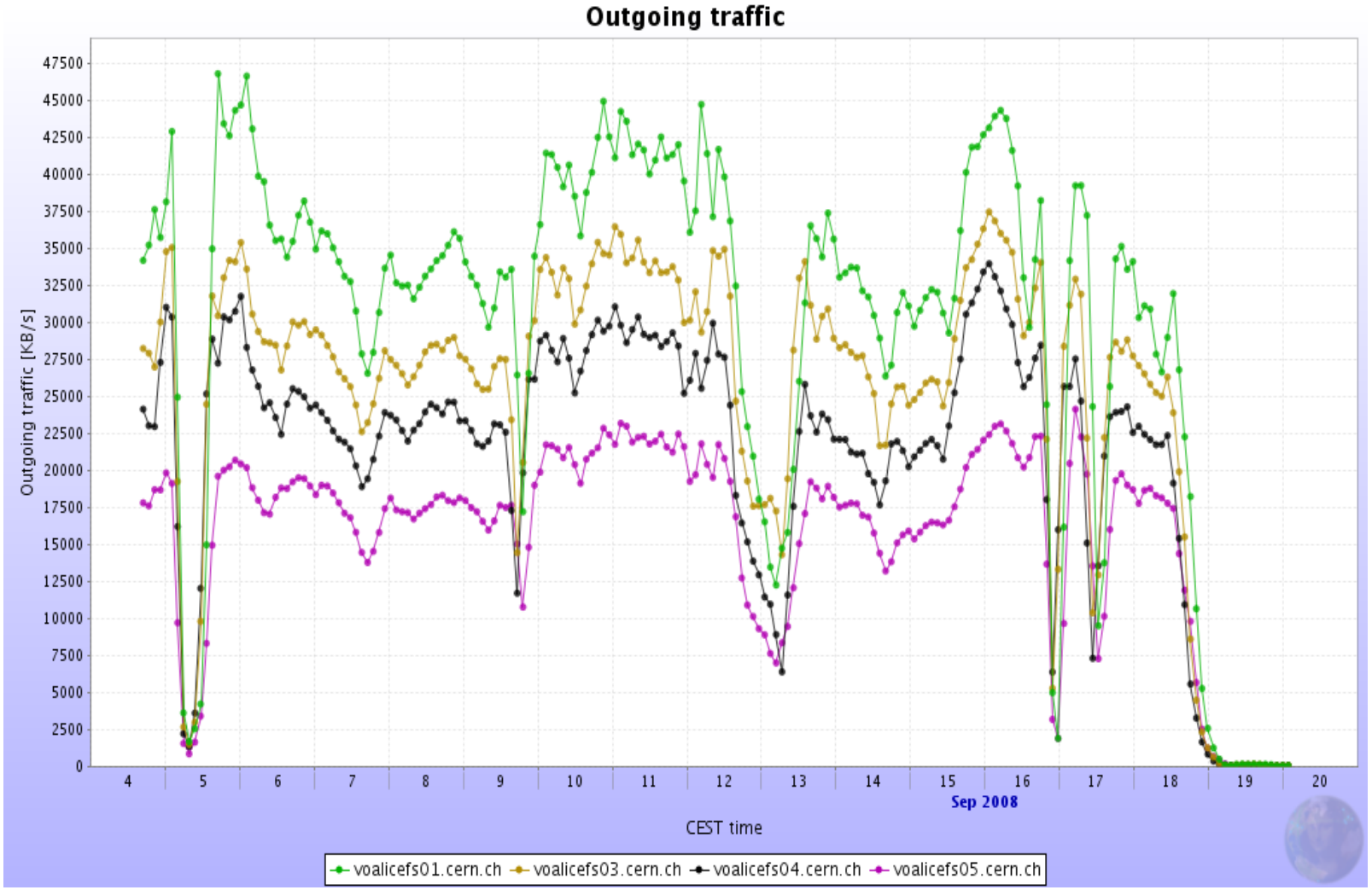
CERN IT Department

- ALICE conditions data repository
  - Regular ROOT files annotated in the AliEn catalogue
    - Populated from various online DBs and runtime detector tasks
    - Nothing strange
  - Primary copy on xrootd storage servers at CERN (5x, 30 TB total)
  - Accessed directly by all MC and reconstruction jobs on the Grid
    - Up to 12Kjobs, Up to 6-8K constant connections
    - Directly means no pre-copy, i.e. very byte-efficient

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

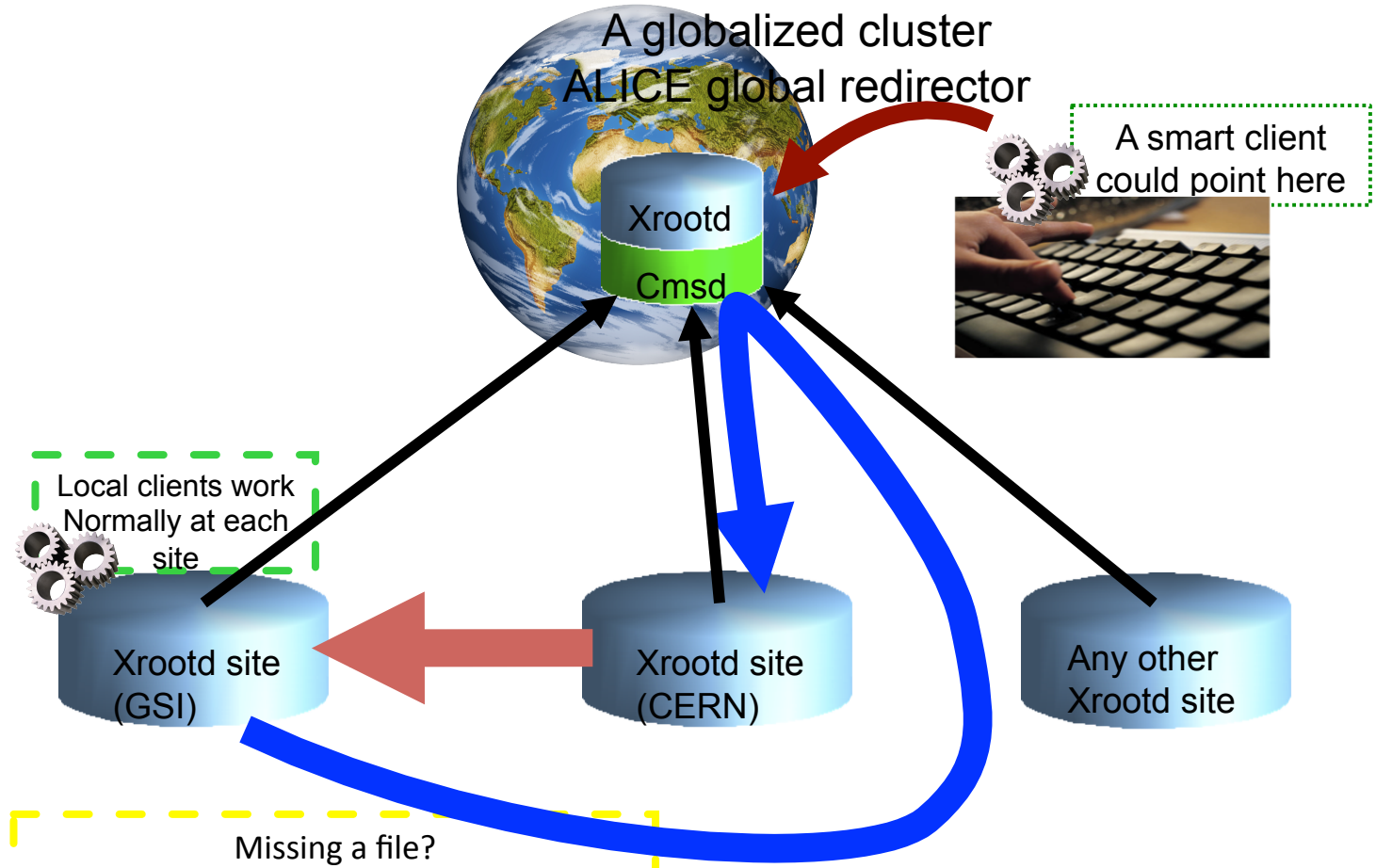
CERN

A nice example



CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

More than Globalization: The VMSS



Virtual
Mass
Storage
System
… built on data Globalization

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Conclusion (1/3)

CERN IT Department

- Many things are possible
  - E.g. solving the “conditions data” problem was a breakthrough for Alice
    - It would be nice to use the globalization to lighten the File Catalog
      - And use it more as a metadata catalog
- Technologically it’s satisfactory
  - But not ended here, there are new possible things, like e.g.
    - Torrent-like *Extreme copy*
      - To boost data movements also in difficult sites
    - Mounting locally a globalized WAN Xrootd metacluster
      - As a local file system using the XCFS tools

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

Conclusion (2/3)

- On the level of the data access framework (ROOT) many things improve every day
  - E.g. how to generate a more “disk-friendly” load to avoid inefficiencies
  - All the improvements bring us closer to the possibility of exploiting forms of “interactive access” to the applications
    - That would mean an evolution from the current “batch system” approach
    - Some projects are aiming for that (e.g. PROOF)
    - That means also having interactive access to the data
    - The massive deployment of these newer technologies could be the real challenge for the next years

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Conclusion (3/3)

CERN IT Department

- "Large DBs on the GRID" …
  - It should be more clear that we are talking about complex distributed systems which interleave:
    - Big and structured high-perf data repositories
    - Complex metadata
    - Distributed computing infrastructures
  - Hence their characteristics interleave
    - Right now the best answer IN HEP is to complement them
    - Use conventional DBs for the metadata and ev. For the coordination
    - Use Web-like direct access for the data, with efficient protocols
    - Make the applications able to exploit the modern HW
    - Organise storage and job submission in order to reduce the distance between a job and its primary source of data
      - Without heading for super-complex solutions just in order to enforce it
    - Organize the effort in order to contain the overall complexity

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

CERN

Thank you!

QUESTIONS?

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it