

Cooperation among ALICE SEs

Current status and directions

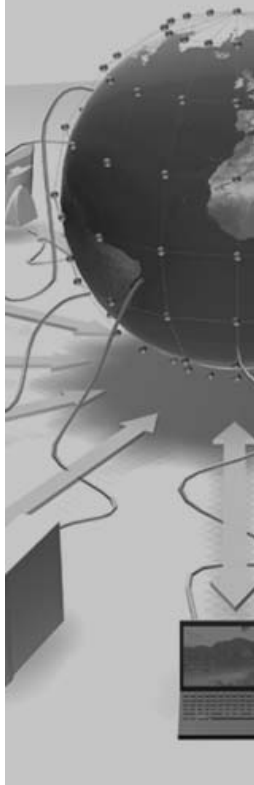
(The ALICE Global Redirector. A step towards real storage robustness.)

Fabrizio Furano

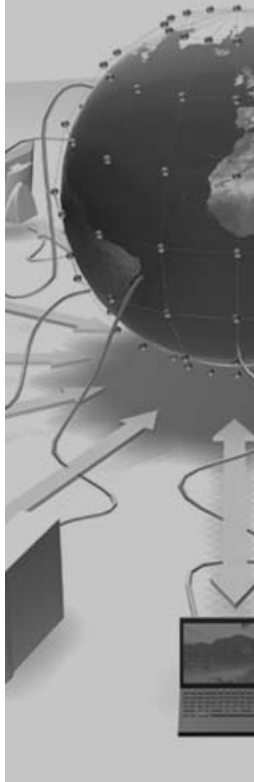
CERN IT/GS

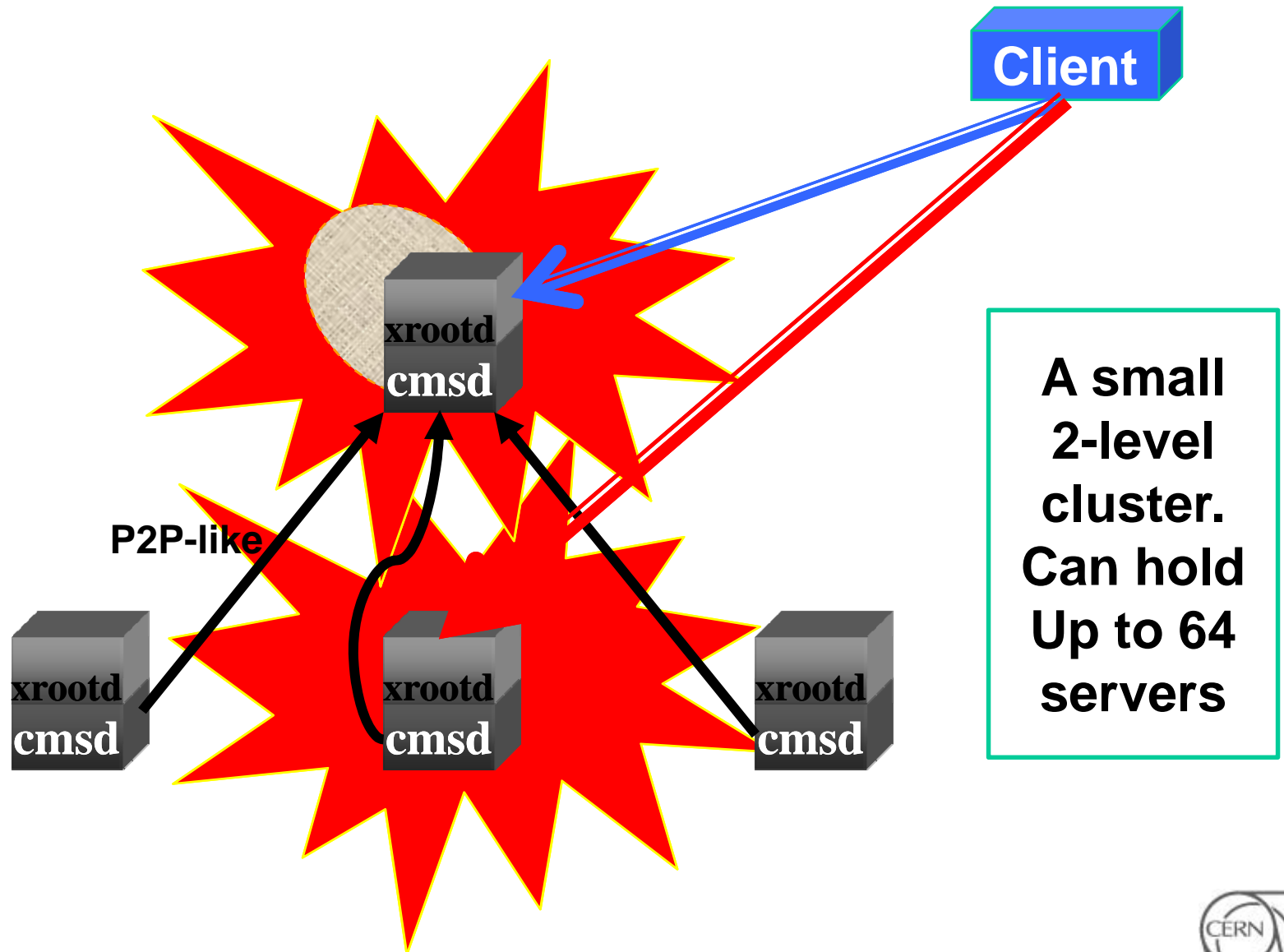
Nov 2008

ACAT '08

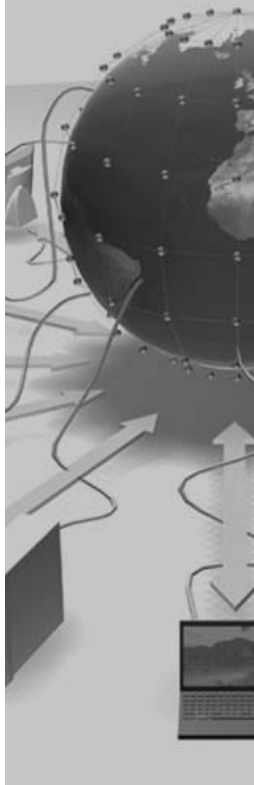


- No weird configuration requirements
 - Scale setup complexity with the requirements' complexity. No strange SW dependencies.
- Highly customizable
- Fault tolerance
- High, scalable transaction rate
 - Open many files per second. Double the system and double the rate.
 - NO DBs for filesystem-like funcs! Would you put one in front of your laptop's file system? How long would the boot take?
 - No known limitations in size and total global throughput for the repo
- Very low CPU usage on servers
- Happy with many clients per server
 - Thousands. But check their bw consumption vs the disk/net performance!
- WAN friendly (client+protocol+server)
 - Enable efficient remote POSIX-like direct data access through WAN
 - Read and write
- WAN friendly (server clusters)
 - Can set up WAN-wide huge repositories by aggregating remote clusters
 - Or making them cooperate





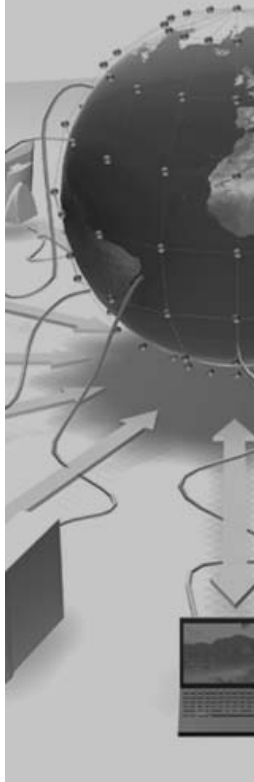
- Technically, various things can be done with WANs and the xrootd
 - The client can read/write efficiently
 - We can group remote clusters into one meta-cluster
 - Remote xrootd-based clusters can fetch files between them
- In ALICE these possibilities are going to be exploited in a coordinated way to:
 - Give robustness to tier-2 data access
 - Without breaking the computing model
 - Suggest new ideas/tricks
 - For quick and robust data movement (Tier-2s)
 - For maintenance, SE backups,



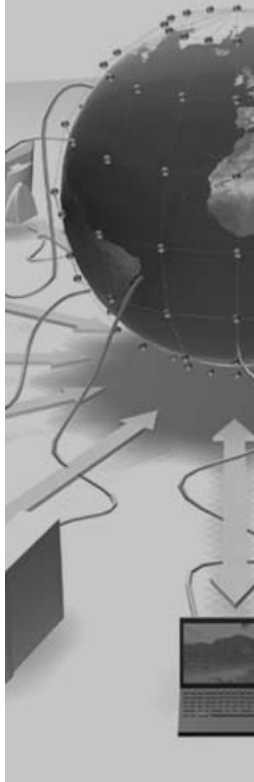
- Up to now, xrootd clusters could be populated
 - With xrdcp from an external machine
 - Writing to the backend store (e.g. CASTOR/DPM/HPSS etc.)
- E.g. FTD in ALICE now uses the first
 - Load and resources problems
 - All the external traffic of the site goes through one machine
 - Close to the dest cluster
- If a file is missing or lost
 - For disk and/or catalog screwup
 - Job failure
 - ... manual intervention needed
 - With 10^7 online files finding the source of a trouble can be VERY tricky

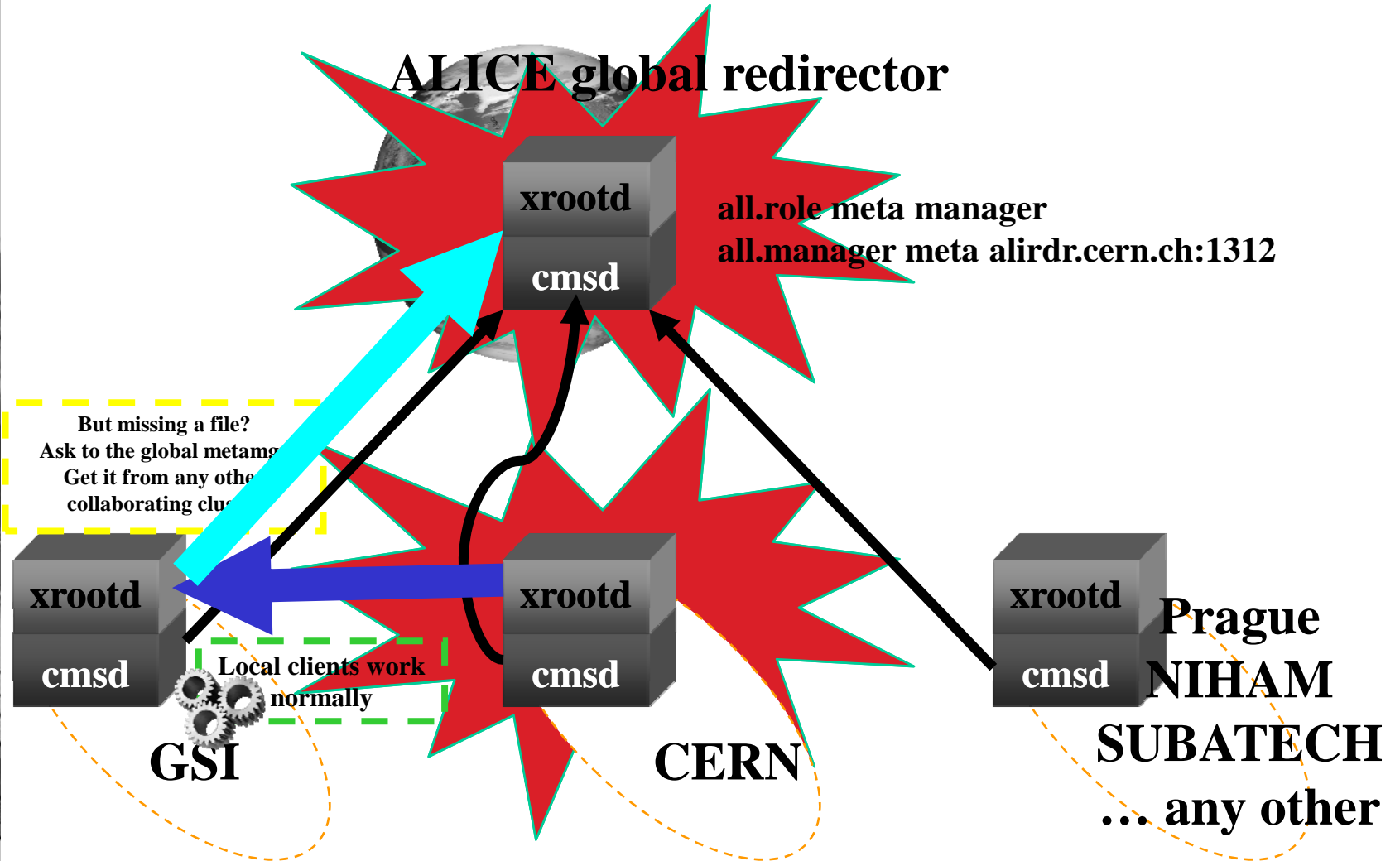


- But, what if the missing file is available in a neighbor cluster?
- What if the cluster was able to detect it (in a few millisecs) and fetch it at max speed?
 - After all, it was a “hole” in the repo
- In the near future, this highly robust way to move files could be used as a transport mechanism
 - Always coordinated by FTD
 - Or any other external system as well



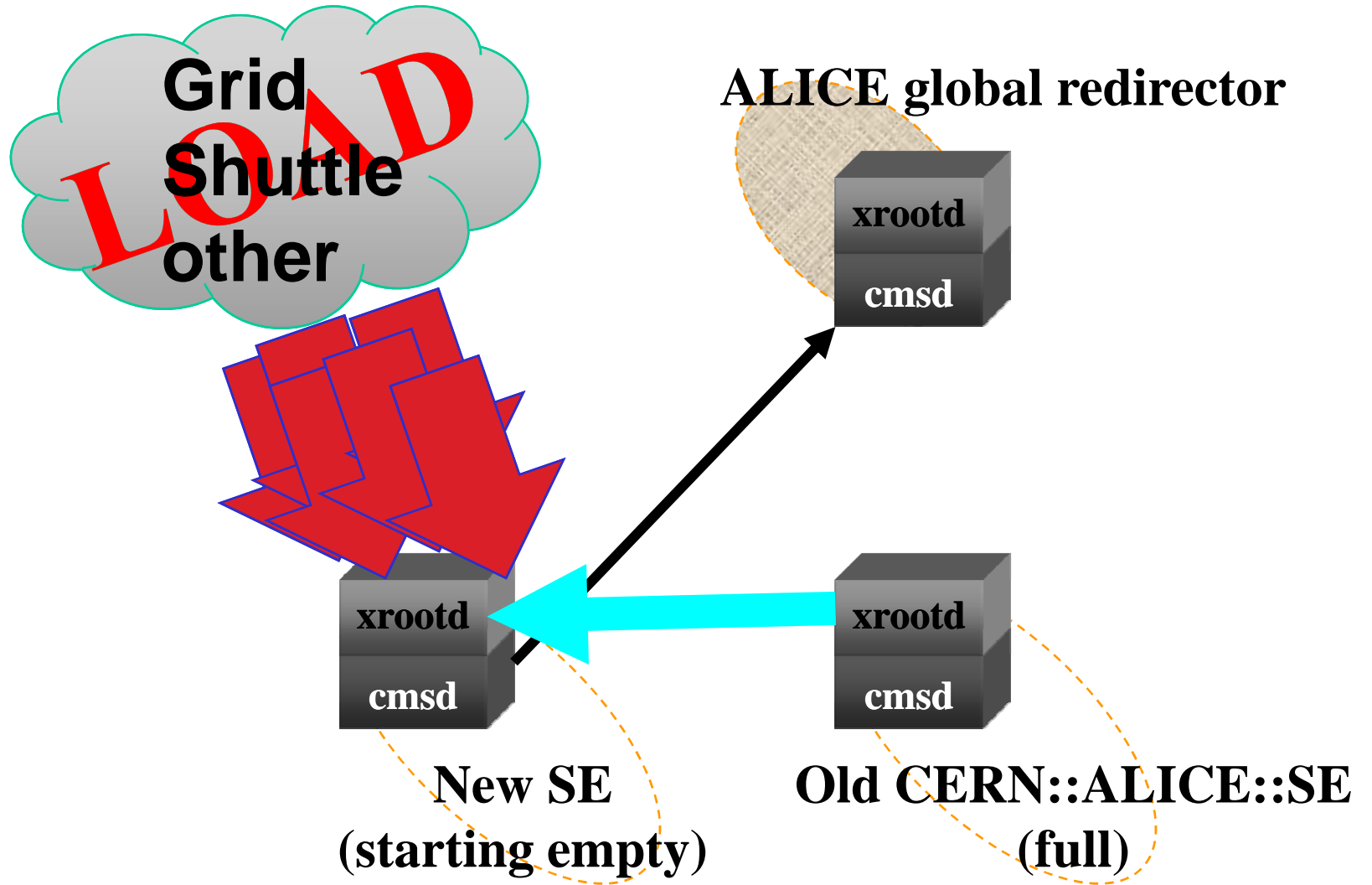
- **Basic idea:**
 - A request for a missing file comes at cluster X,
 - X assumes that the file ought to be there
 - And tries to get it from the collaborating clusters, from the fastest one
 - The MPS (MSS intf) layer can do that in a very robust way
- **Note that X itself is part of the game**
 - And it's composed by many servers
- **In practice**
 - Each cluster considers the set of ALL the others like a very big online MSS
 - This is much easier than what it seems
 - Slowly Into production for ALICE in tier-2s
- **NOTE:** it is up to the computing model to decide where to send jobs (which read files)



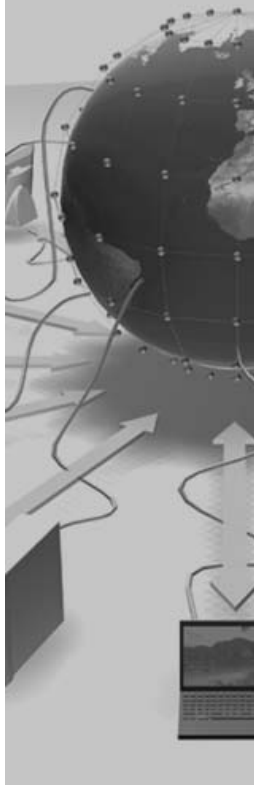


- All of the ALICE cond data is in ALICE:CERN::SE
 - 5 machines pure xrootd, and all the jobs access it, from everywhere
 - There was the need to refurbish that old cluster (2 yrs old) with completely new hw
 - VERY critical and VERY stable service. Stop it and every job stops.
- A particular way to use the same pieces of the vMSS
- In order to phase out an old SE
 - Moving its content to the new one
 - Can be many TBs
 - rsync cannot sync 3 servers into 5 or fix the organization of the files
- Advantages
 - Files are spread evenly → load balancing is effective
 - More used files are fetched typically first
 - No service downtime, the jobs did not stop
 - Server downtime of 10-20min
 - The client side fault tolerance made the jobs retry with no troubles

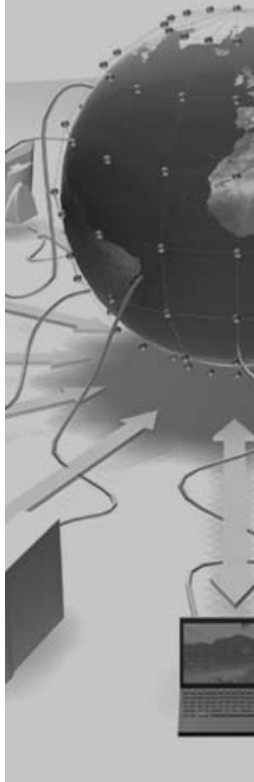




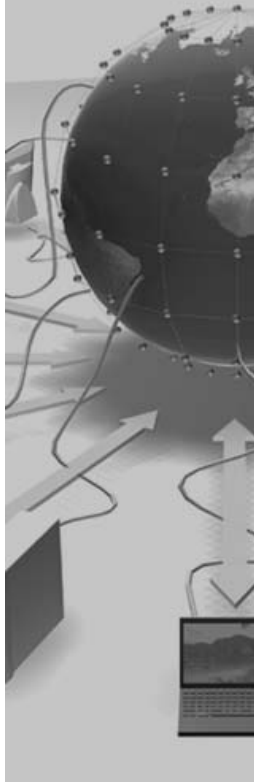
- The mechanism is there, fully “boxed”
 - The new setup does everything it’s needed
- A side effect:
 - Pointing an app to the “area” global redirector gives complete, load-balanced, low latency view of all the repository
 - An app using the “smart” WAN mode can just run
 - Probably now a full scale production/analysis won’t
 - But not sure... maybe yes as well
 - Anyway, what about a small debug analysis on a laptop?
 - After all, HEP sometimes just copies everything, useful and not
 - I cannot say that in some years we will not have a more powerful WAN infrastructure
 - And using it to copy more useless data looks just ugly
 - If a web browser can do it, why not a HEP app? Looks just a little more difficult.
- Better if used with a clear design in mind



- A small test to evaluate:
 - The robustness of the xfer mechanism
 - The illusion of having a unique huge repository
 - NB From Alien it is not available... you must use the Alien tools
 - The automated fetching from one site to another
- So, a functional test among “friend” sites
 - But with the system in production
 - BTW it “knows” all the ALICE OCDB storage

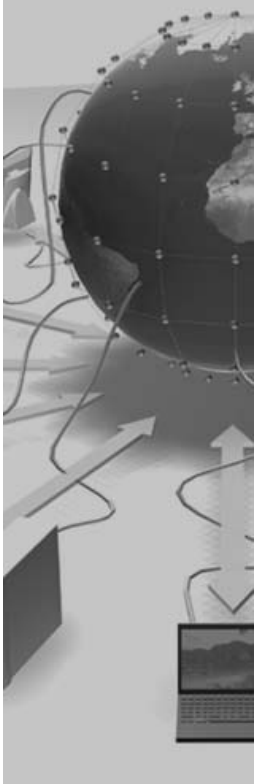
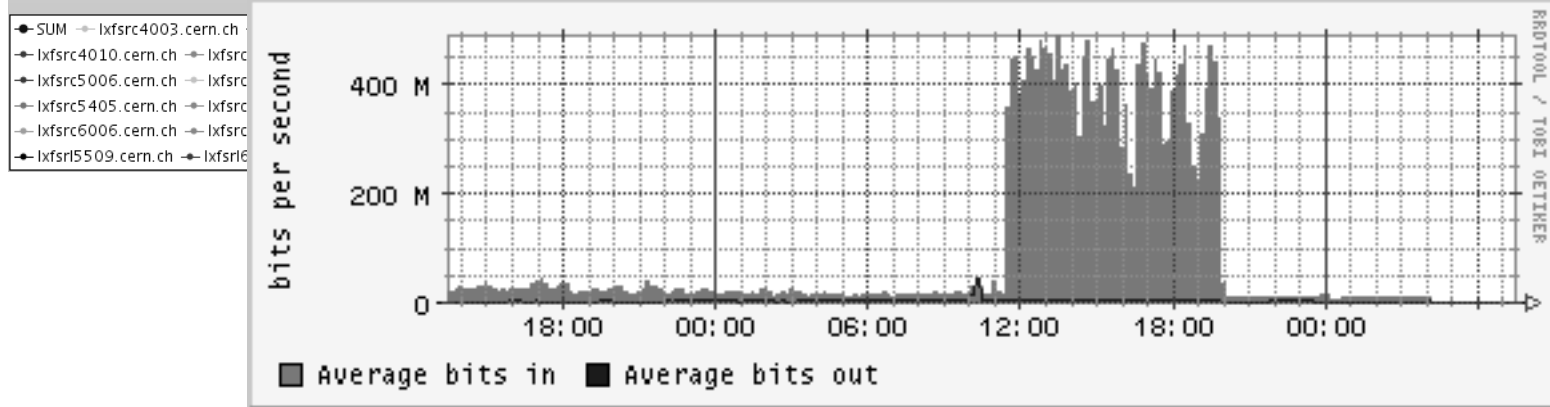
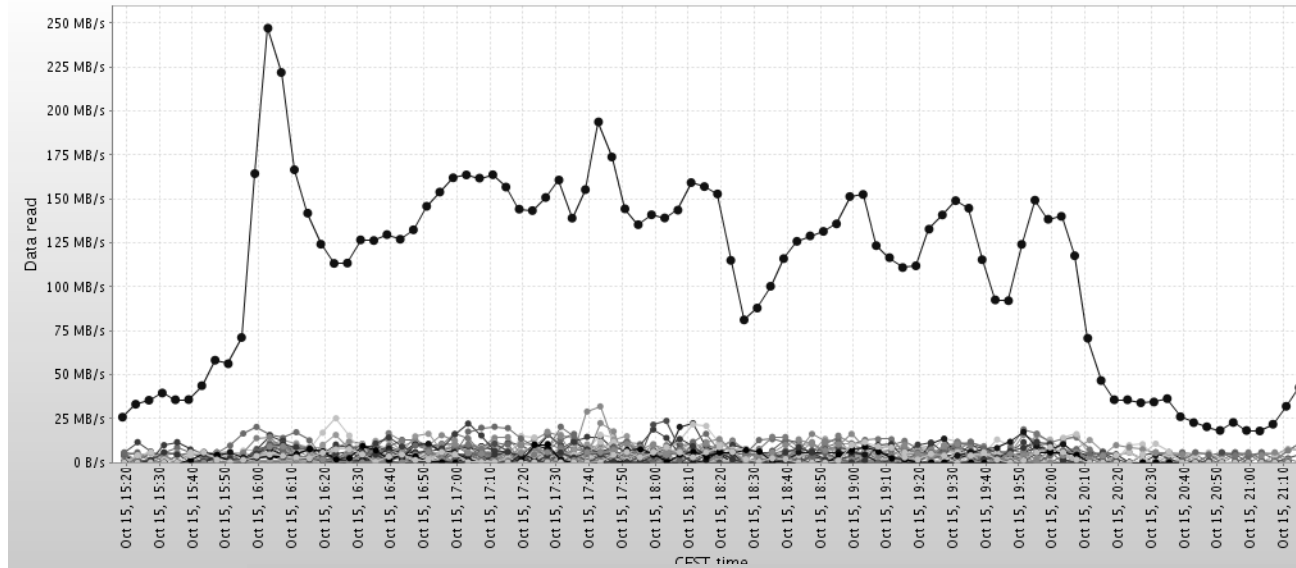


- Test: a small cluster at GSI and one at Subatech join the global game
 - Plus a list of files (816, 1.5T in total)
 - Ask those clusters to fetch the files from CERN
 - Delete some files from one of them
 - Verify that the files are still globally accessible
 - Pointing an app to the global redirector
 - Access the files
 - In a local cluster, like a job
 - and verify that they are immediately restored if absent
 - Because they “ought to be there” by definition
 - Basically it’s a functional test

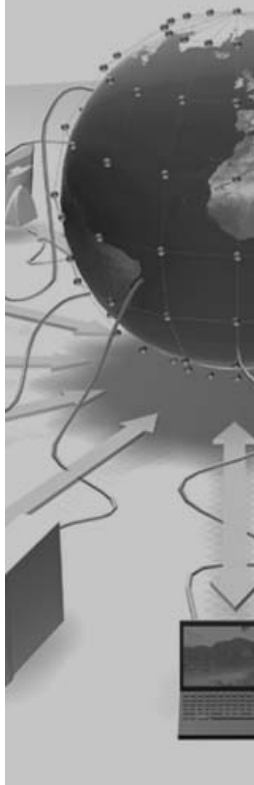


- Very exciting results. No xfer failures reported.
- Some minor setup issues to be properly fixed
 - Root-made setup, combinations of trailing slashes
- An app pointed to the global redirector just works
 - Immediately, but this was limited to online files
 - Some issues on the meta-manager to be investigated
- An app pointed where the file “disappeared”
 - Works after the delay of fetching the file from the neighbor site
- Performance
 - Just a max of 4 concurrent xfers brung about
 - 80MB/s at GSI (2 data servers)
 - 40-45MB/s at Subatech (1 data server)
 - 10 xfers at GSI brung more than 100MB/s
 - Both have a Gb WAN
 - Completely filling it up does not look like a problem
 - It looks that we saturated the disks

xrootd IO Summary



- Setting all this manually up can be VERY tricky
- The xrd-installer setup has been refurbished to include all the discussed items
 - Originally developed by A.Peters and D.Feichtinger
- Usage: same of before
 - Login as the xrootd user (being root is not necessary and adds complexity)
 - Create a directory in \$HOME
 - Download the installer script
 - `wget http://project-arda-dev.web.cern.ch/project-arda-dev/xrootd/tarballs/installbox/xrd-installer`
 - Run it
 - `xrd-installer -install -prefix /home/xrduser/myxrootdinstall`
- Then, there are 2 small files containing the parameters
 - For the token authorization library (if used)
 - For the rest (about 10 parameters)
 - Geeks can also use the internal xrootd config file template
 - » And have access to really everything
 - » Needed only for **very** particular things



VMSS_SOURCE: where this cluster tries to fetch files from, in the case they are absent.

MANAGERHOST: the redirector's name

LOCALPATHPFX: the prefix of the namespace which has to be made "facultative" (not needed by everybody)

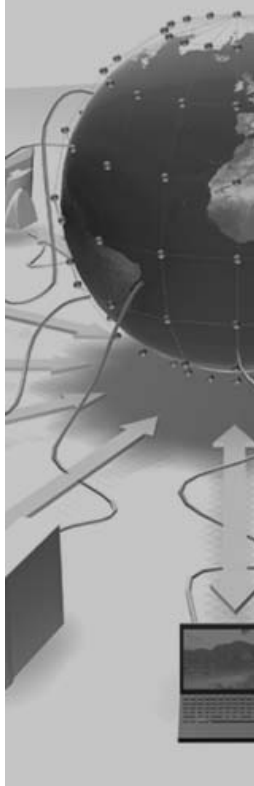
SERVERONREDIRECTOR: is this machine both a server and redirector?

LOCALROOT is the local (relative to the mounted disks) place where all the data is put/kept by the xrootd server.

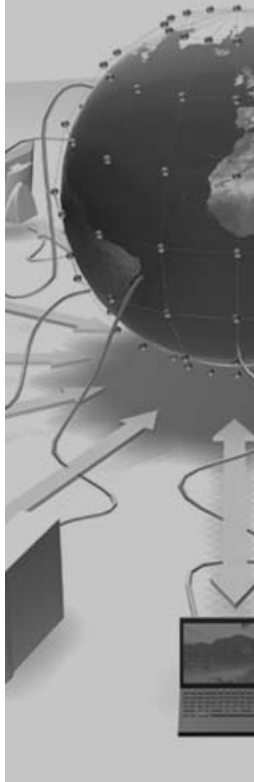
OFSLIB: are you using the ALICE security model?

OSSCACHE: probably your server has more than one disk to use to store data. Here you list the mountpoints to aggregate.

METAMGRHOST,
METAMGRPORT: host and port number of the meta-manager (default: ALICE global redirector)



- We spoke about the status of
 - WAN access and WAN-wide clusters
 - But the WAN could be a generic ADSL (which has quite a high latency)
 - We like the idea of being able to access data efficiently and without pain, no matter from where
 - And this is quite difficult to achieve with typical distributed FSs
 - WAN-wide clusters and inter-cluster cooperation
 - With reference to the ALICE data model
- A lot of work going on
 - Assistance, integration, debug, deployments
 - Setup robustness to config “styles” and combinations
 - Different persons suppose different things (slashes, spaces, ...)
 - New fixes/features
 - Documentation to update on the website (!)



- Old and new Collaborators
 - Andy Hanushevsky, Fabrizio Furano
 - Root: Fons Rademakers, Gerri Ganis (security), Bertrand Bellenot (MS Windows porting)
 - Derek Feichtinger, Andreas Peters, Guenter Kickingger
 - STAR/BNL: Pavel Jakl, Jerome Lauret
 - Cornell: Gregory Sharp
 - Subatech: Jean-Michel Barbet
 - GSI: Kilian Schwartz
 - SLAC: Jacek Becla, Tofigh Azemmoon, Wilko Kroeger, Bill Weeks
 - Alvise Dorigo, Peter Elmer
- Operational collaborators
 - BNL, CERN, CNAF, FZK, INFN, IN2P3, RAL, SLAC