Federating Data in the ALICE Experiment

Costin.Grigoras@cern.ch

Outline

- Data access methods in ALICE
- Storage AAA
- Storage monitoring
- SE discovery
- LHC experiments' experience

Data access methods in ALICE

- Central catalogue of logical file names (LFN)
  - With owner:group and unix-style permissions
  - Size, MD5 of files
  - Metadata on subtrees
- Each LFN is associated a GUID that can have any number of replicas (PFNs)
  - root://<redirector>//<HH>/<hhhhh>/<GUID>
    - *HH* and *hhhhh* are hashes of the GUID
  - Same namespace on all storage elements
- Files are immutable on the SEs

Data access methods in ALICE (2)

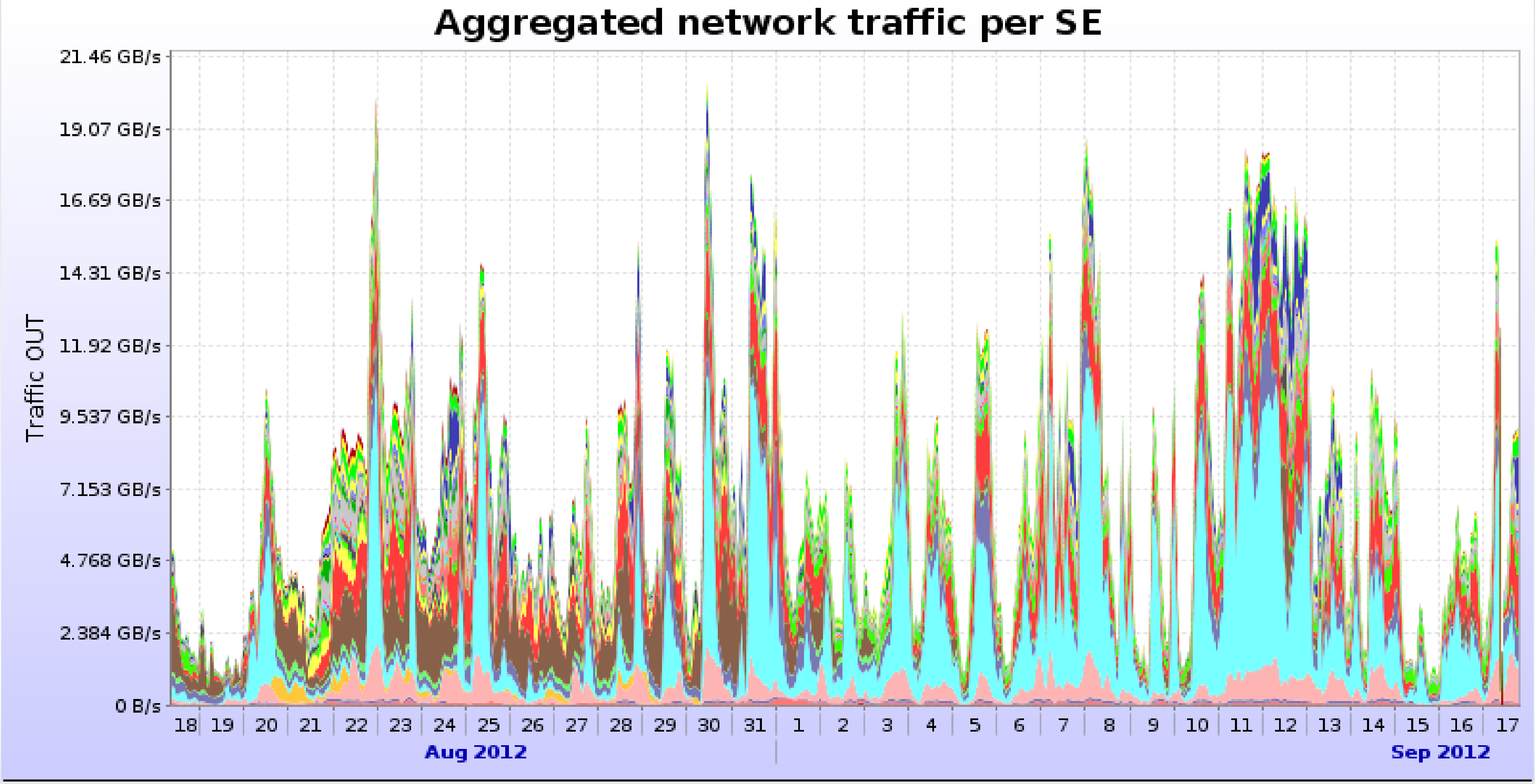
- Data files are accessed remotely
  - From the closest working replica to the job
    - Jobs go to where a copy of the data is, though we are investigating how to combine job priority with lax site match
- Exclusive use of xrootd protocol for remote access
  - Plus http, ftp, torrent for downloading other input files
- At the end of the job N (2..4 typically) replicas are uploaded from the job itself (xrdcp cmd line)
- Scheduled data transfers for raw data, conditions and other on-demand replications (like SE evacuation) using xrd3cp

Some figures

- 58 disk SEs, 9 tape SEs (T0 and T1s)
  - 57x xrootd, 1x EOS, 1x DPM, 4x CASTOR, 4x dCache
- 17PB in 200M files on disk SEs
- Average replication factor is 3
- 2 copies of the raw data on MSS:
  - Full copy at CERN T0
  - One distributed copy at T1s (full runs)

More figures

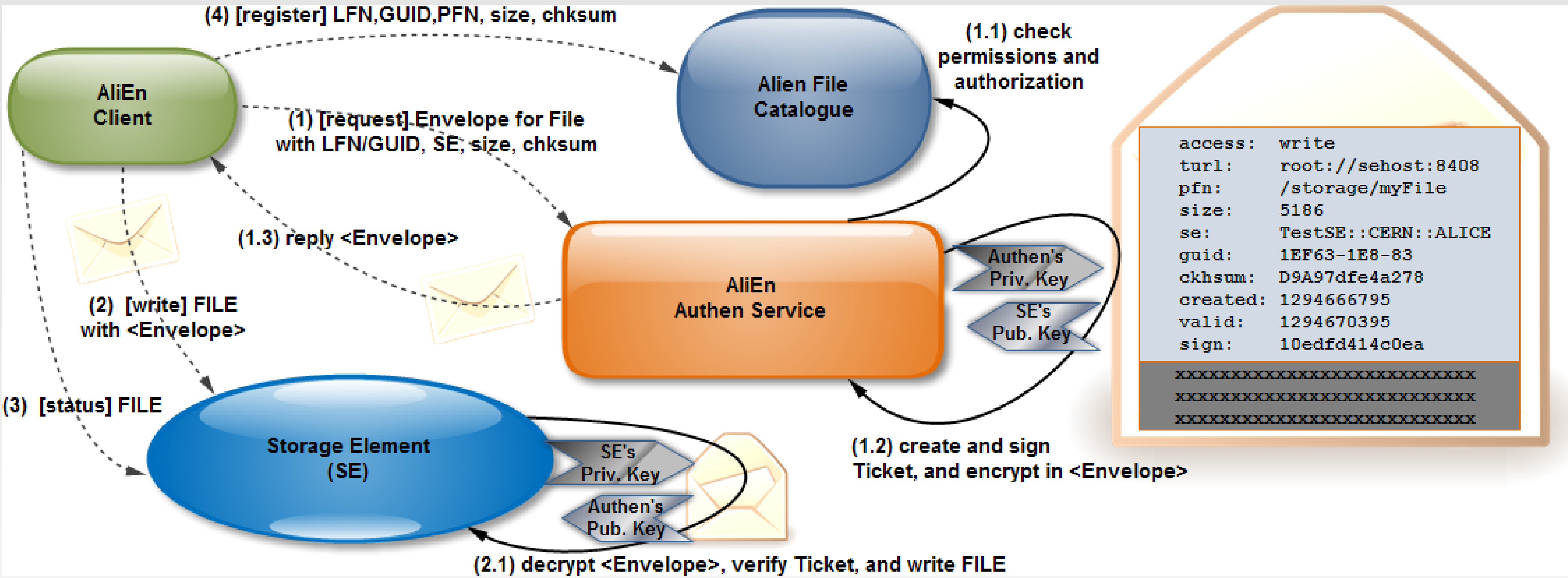
- Writing at 1GB/s avg, 4GB/s max (2.3PB/mo)
- Reading at 7.4GB/s avg, 20GB/s max (18.5PB/mo)



Storage AAA

- Storage-independent
- Handled centrally by the Authen AliEn service
- Checks client credentials and catalogue permissions and issues access tickets
  - XML block signed and encrypted by Authen
- The client hands these tickets to the respective storage and (for writes) notifies the catalogue of the successful operation
- Implemented in xrootd (EOS, Castor and EOS are using it) and dCache

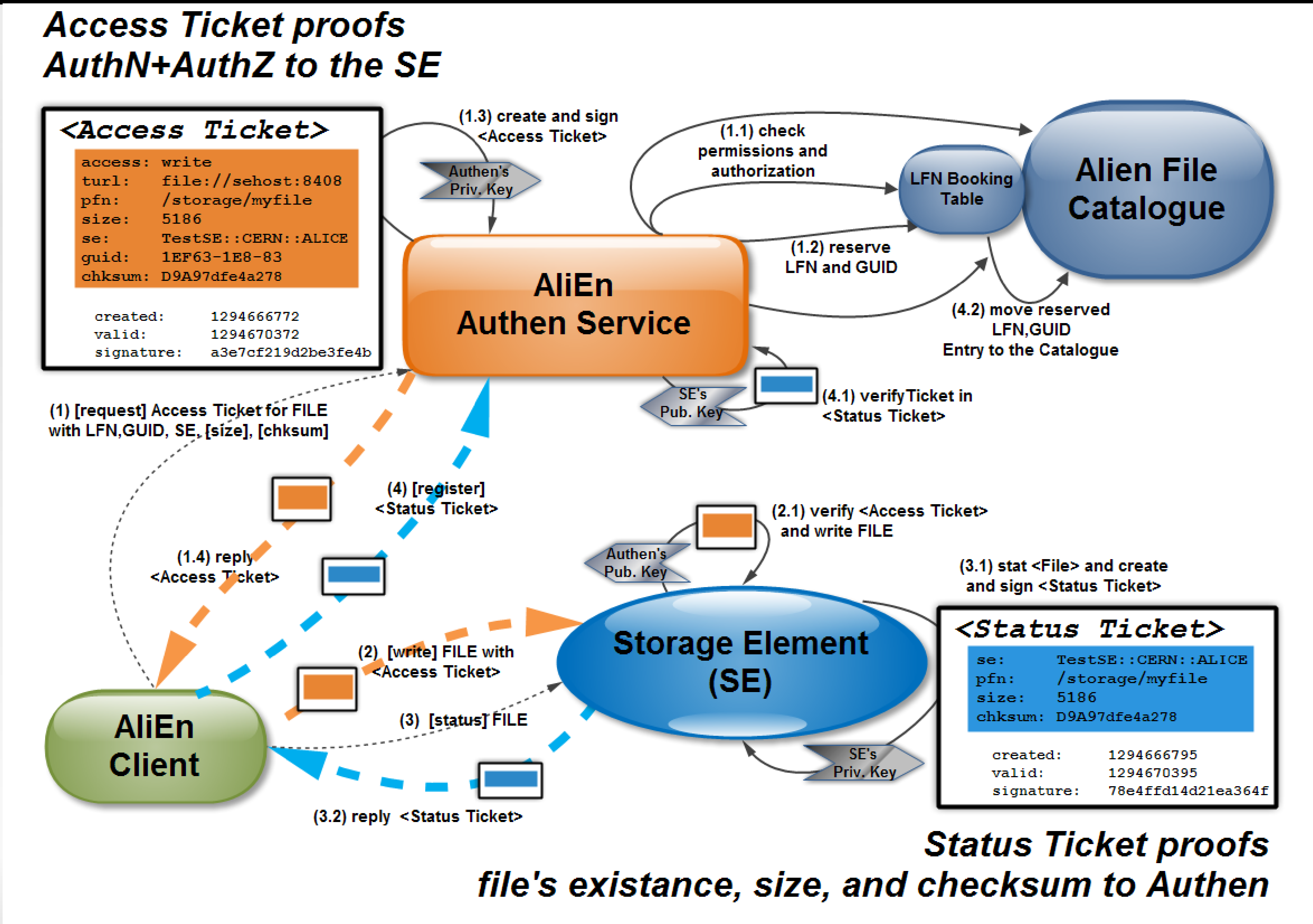
Storage AAA (2)



Storage AAA – in deployment

- Similar to what is in production now
- Simplified tickets
  - Less text, just signed (no encryption any more)
- Introducing storage reply envelopes
  - Size and checksum of what the server got
    - Signed by the storage and returned by xrdcp, xrdstat
    - Very important for data integrity
  - When committing a write the above must match what was booked
  - Can later recheck the files for consistency directly on the servers

Storage AAA – in deployment (2)



Monitoring – host parameters

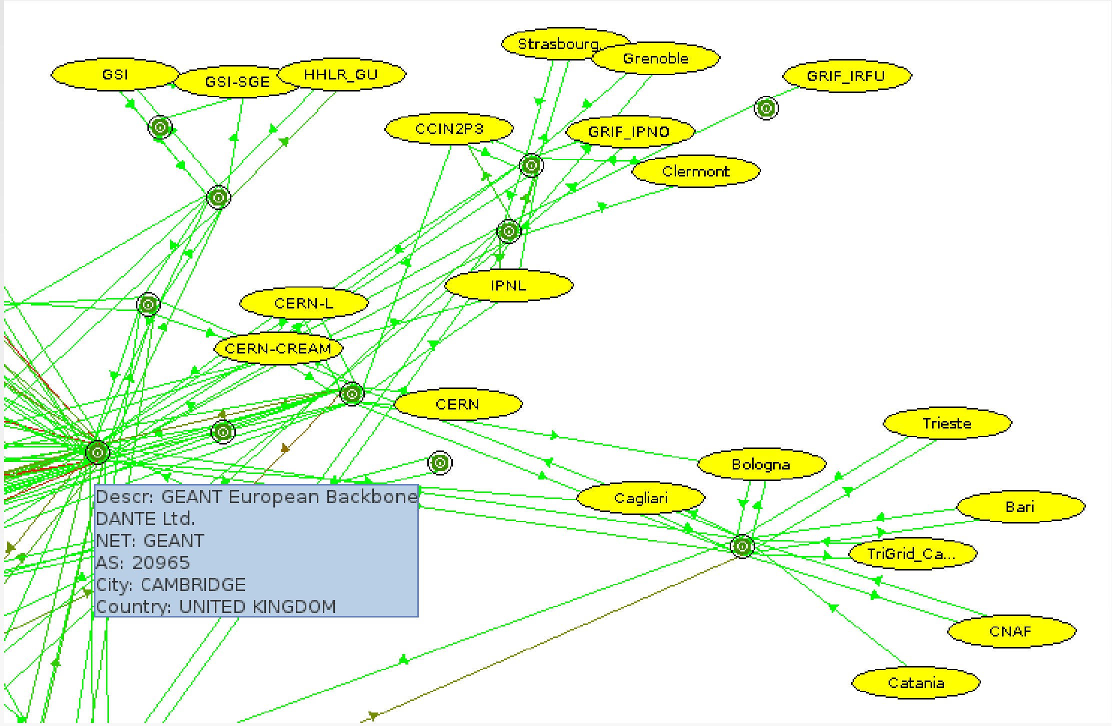
- Integrated in the overall monitoring of ALICE
- xrootd plugin package also brings a host and service monitoring daemon
- Monitoring data from xrootd and the daemon is sent to the site MonALISA instance
- Collected by the central repository and aggregated per cluster
  - http://alimonitor.cern.ch?571
- Under deployment: xrootd 3.2.2 with extended monitoring information

Storage monitoring – functional tests

- add / get / delete performed every 2h
  - From a central location
  - Using the full AliEn suite (like any user or job)
- Results archived for a “reliability” metric
  - Last week * 25% + last day * 75%
- Separate metrics for read and write

Network topology discovery

- Site MonALISA instances perform between each pair of them
  - Traceroute / tracepath
  - Bandwidth estimation
- Recording all details we get a good and complete picture of the network topology



AS view of the topology

SE discovery

- Based on a dynamic “distance” metric from an IP address to a SE
  - Starting from the network topology
    - Same site, same AS, same country, continent...
    - RTT where known, at least to the AS
  - Last functional test excludes non-working SEs
  - Altered by
    - Reliability
    - Remaining free space
    - A random factor to assure 'democratic' data distribution

SE discovery (2)

- Reading from the closest working replica
  - Simply sorting by the distance metric, including the non-working SEs, as last resort
- Writing to the closest working SEs
  - Each SE is associated a tag (“disk”, “tape”, “paper”)
  - Users indicate the number of replicas of each type
    - Default is “disk=2”
  - Not excluding the option of specific target SEs
  - Keep asking until the requirements are met or no more SEs left to try

Remote access impact on efficiency

SSD 266 MB/s	**Access time 0.2 ms**	**Read size 270 MB AOD PbPb**
Job time 39.5 sec	**Throughput 6.83 MB/s**	**Job efficiency 94.1 %**

Spinning 50 MB/s	**Access time 13 ms**	**Read size 270 MB AOD PbPb**
Job time 45.5 sec	**Throughput 5.93 MB/s**	**Job efficiency 86.5 %**

Inter site 7.4 MB/s (JINR)	**Access time = RTT 63 ms + local disk access time (?)**	**Read size 21.53 MB AOD PbPb**
Load=200, Job time 258 sec	**Throughput 0.083 MB/s**	**Job efficiency 2.5 %**
Load=5, Job time 46.8 sec	**Throughput 0.46 MB/s**	**Job efficiency 13.4 %**

I/O latency is a killer for events with many branches

Credit: Andrei Gheata

US ATLAS efficiency tests

- Investigate efficiency varying %events read and TTreeCache size
- Steady improvement with buffer size
- With large enough buffers **80% to 50% wall time efficiency**

Client: *.uchicago.edu				
	% events read (30MB buffer)			100 MB buffer
Server	10%	50%	100%	100%
SLAC	WALLTIME=35.8	WALLTIME=74.5	WALLTIME=105.9	WALLTIME=76.0
	CPUTIME=11.9	CPUTIME=25.12	CPUTIME=41.57	CPUTIME=41.78
BNL	WALLTIME=28.2	WALLTIME=61.6	WALLTIME=87.8	WALLTIME=62.3
	CPUTIME=12.01	CPUTIME=25.27	CPUTIME=45.66	CPUTIME=41.69
SWT2-UTA	WALLTIME=28.1	WALLTIME=40.9	WALLTIME=66.78	WALLTIME=56.4
	CPUTIME=12.06	CPUTIME=22.6	CPUTIME=41.69	CPUTIME=41.78
AGLT2	WALLTIME=25.4	WALLTIME=45.0	WALLTIME=58.5	WALLTIME=49.5
	CPUTIME=11.9	CPUTIME=25.3	CPUTIME=44	CPUTIME=41.65
MWT2	WALLTIME=18.8	WALLTIME=29.4	WALLTIME=48.6	WALLTIME=46.2
	CPUTIME=11.93	CPUTIME=25.2	CPUTIME=44	CPUTIME=42.11

Credit: Rob Gardner

Federating storages as seen by the rest of the LHC experiments

- Optimization of direct access to data is the main goal of all experiments
- Coherent file naming with access to everything
  - Users should be oblivious to the physical storage layout
- WAN direct access is the ultimate wish
- Give more importance to the chaotic, Web-like user activity
- Keep the official data processing (jobs, MC, reco, etc.) as it is, if possible enhance

Conclusions of the Storage Federations WG @ CERN

Federated storage use cases

- Fail over for jobs, with redirection in the client and/or the server
  - In CMS and ATLAS the fallback is predetermined (eg to the US redirector or the EU redirector)
- Self healing (hooks on missing files from the local cluster)
  - CMS investigates dynamic caching of (parts of) files by the local storage
  - ALICE AFs use this method to populate the cluster
- Even full remote access for jobs of certain classes

Conclusions

- ALICE distributed storage infrastructure is transparent to the users
  - Automatically managed
  - ROOT support as TAlienFile (working with LFNs)
- All experiments are aggregating their storages in federations (one or more...)
  - With different technologies
  - ALICE has a central catalogue and the redirection is done via a location-aware central service, automatically managed
- Network latency is (still) the critical factor
  - Because the remote replicas are used only as fallback we haven't seen the network throughput limitations yet

Thank you!