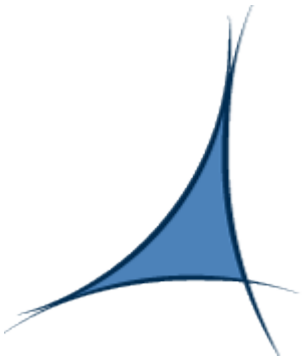
An alternative model to distribute VO specific software to WLCG sites: a prototype at PIC based on CernVM file system

Elisa Lanciotti on behalf of PIC Tier1



PIC
port d'informació
científica

CERN, GDB meeting 08/09/2010

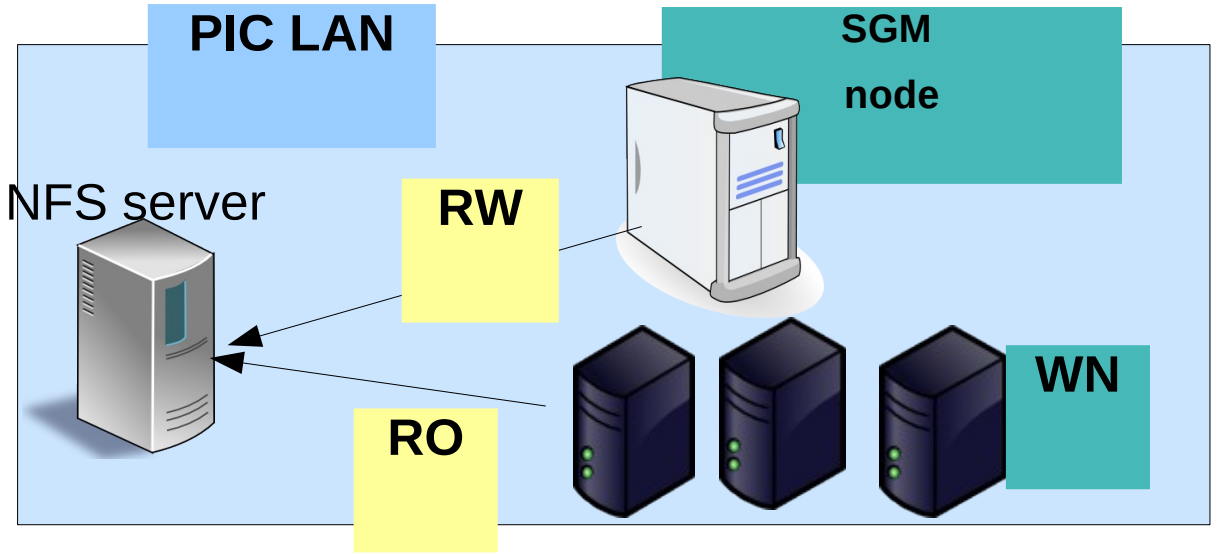
PIC
port d'informació
científica

Contents

- Introduction: current model for software distribution presents several limitations
- A possible alternative: a model based on a new protocol: CernVM file system, a network file system developed in the framework of CernVM project
- Test-bed set up at PIC
- Test description
- Results
- Outlook

Current model for software distribution

- In a distributed computing model as WLCG the software of VO specific applications has to be efficiently distributed to any site of the Grid
- Applications software currently installed in a shared area of the site visible for all worker nodes (WN) of the site (NFS, AFS or other)
- The software is installed by jobs which run on the SGM node (a privileged node of the computing farm where the shared area is mounted in write mode)

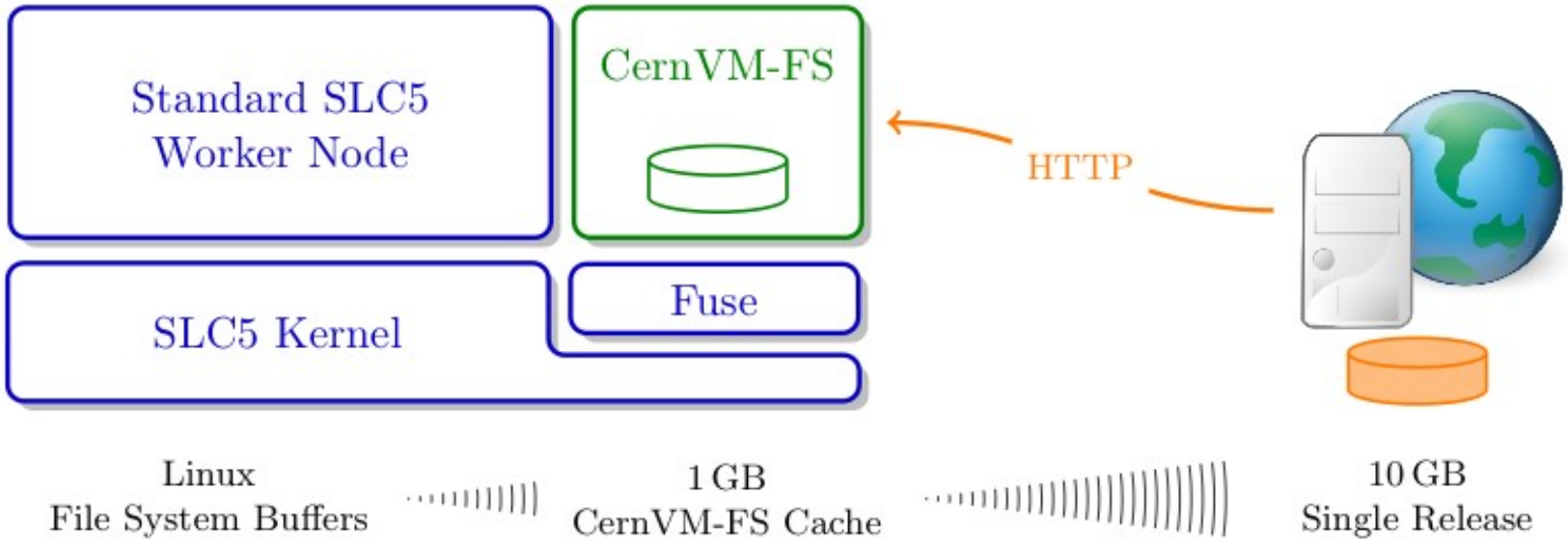


Limitations of the current model for software distribution

- Some issues observed with this model:
  - NFS scalability issues
  - Shared area sometimes not reachable (not properly mounted on the WN, or too loaded NFS server..)
  - NFS locked by SQLite (known bug if NFS is mounted in r-w mode)
  - Software installation in many Grid sites is a tough task (job failures, resubmission, tags publications...)
  - Limited quota per VO in the shared area: if VOs want to install new releases and keep the old ones they have to ask for an increase of quota
- Number of GGUS tickets relative to shared area issues for LHCb: 33 in the last quarter

An alternative model based on CVMFS

- CernVM-FS is a network file system developed in the framework of CernVM project
- Repository of applications: contains the result of a make install
- Can be mounted on the WN and accessed as read-only file system, through http protocol
- Complemented by local site proxy for web caching



CVMFS installation and configuration

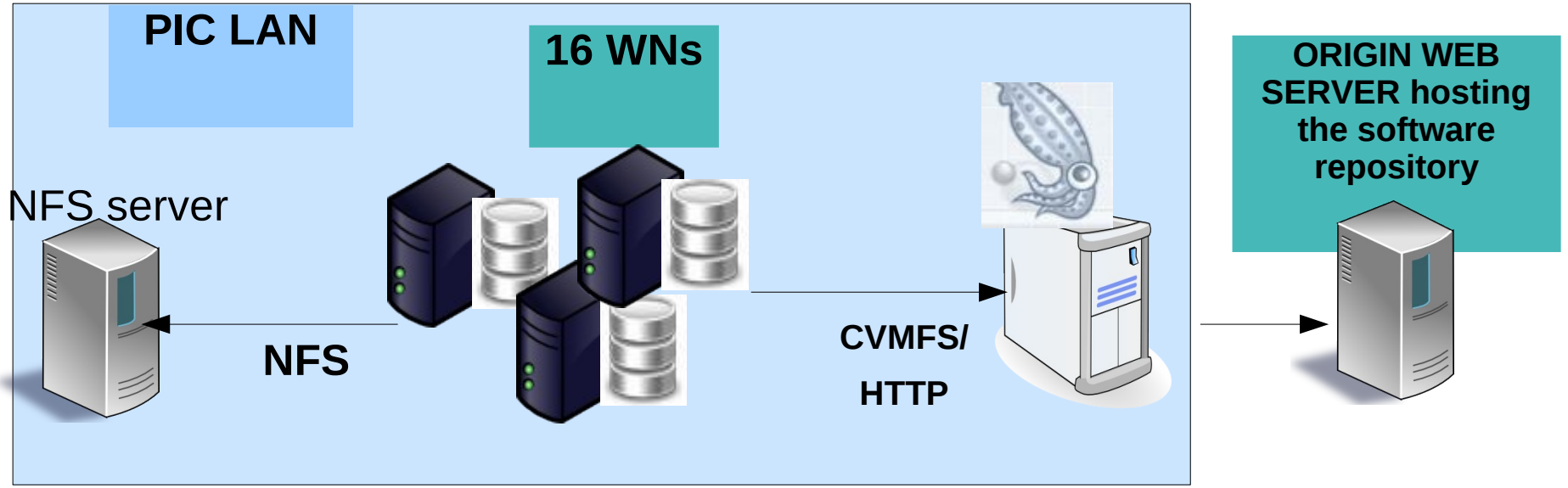
- Installation with yum
- Configuration in one file:

```
Cat /etc/cvmfs/local.d/default.conf
CVMFS_USER=cvmfs
CVMFS_NFILES=32768
CVMFS_CACHE_DIR=/home/cache/cvmfs2
CVMFS_QUOTA_LIMIT=-1
CVMFS_SERVER_URL=http://cernvm-webfs.cern.ch/opt/@org@
CVMFS_OPTIONS=allow_other,remount_sleep=10,entry_timeout=10,attr_timeout=10,timeout=10,negative_timeout=10
CVMFS_REPOSITORIES=lhcb,atlas
CVMFS_HTTP_PROXY="http://squid01-test.pic.es:3128"
```

CVMFS v2.47: added multi VO support, two repositories (ATLAS, LHCb mounted on the same system)

Test-bed setup at PIC

- A dedicated blade of 16 Wns, 8 cores each, configured in a test queue
- On each node, software area mounted through NFS (production NFS server) and software repository of CernVM mounted through CernVM-FS
- One Squid server at the site as **http proxy** and **web cache**: necessary to reduce network latency and reduce the load on the origin web server of CernVM



Test description

- A test job which sets the environment and runs the application for analysis (DaVinci)
- No development required: only difference is an environment variable (path to the software area)

export MYSITEROOT=/software/lhcb/slc4/lib
export VO_LHCB_SW_DIR=/software/lhcb/slc4
Source $MYSITEROOT/Lblogin.sh
SetupProject DaVinci v...
gaudirun.py $DAVINCIROOT/options/DaVinci.py

Software area accessed through NFS

export MYSITEROOT=/opt/lhcb
Source $MYSITEROOT/Lblogin.sh
SetupProject DaVinci v...
gaudirun.py $DAVINCIROOT/options/DaVinci.py

Repository of CernVM accessed through CernVM-FS

Metrics to measure

- Execution time for SetupProject - the most demanding phase of the job for the software area (huge amount of stat() and open() calls)
- Execution time for DaVinci
- Dependence with the number of concurrent jobs

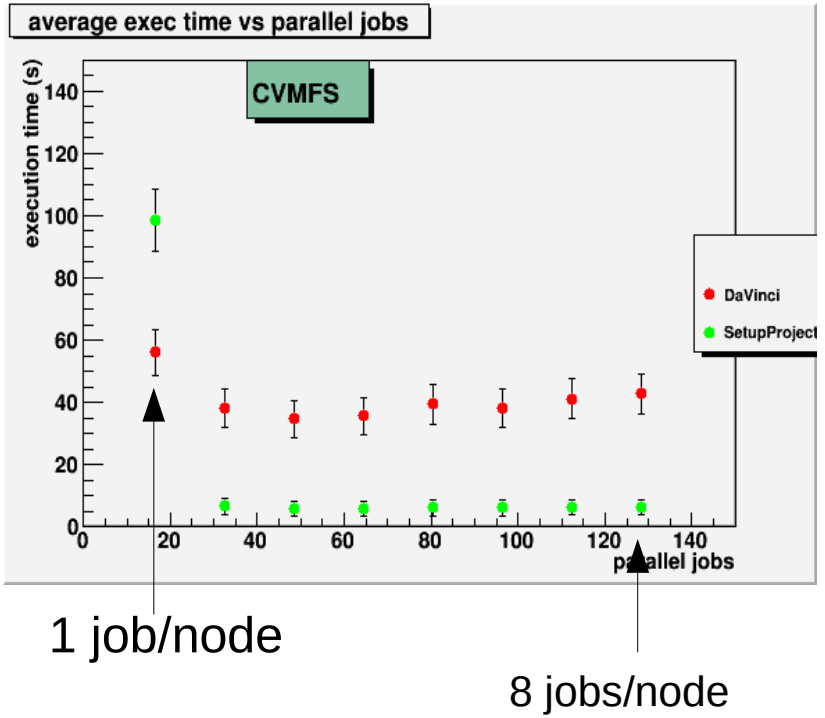
A run of the test script consists of a set of consecutive job submissions:

JobsPerNode=1

while jobsPerNode<=Ncore(=8)

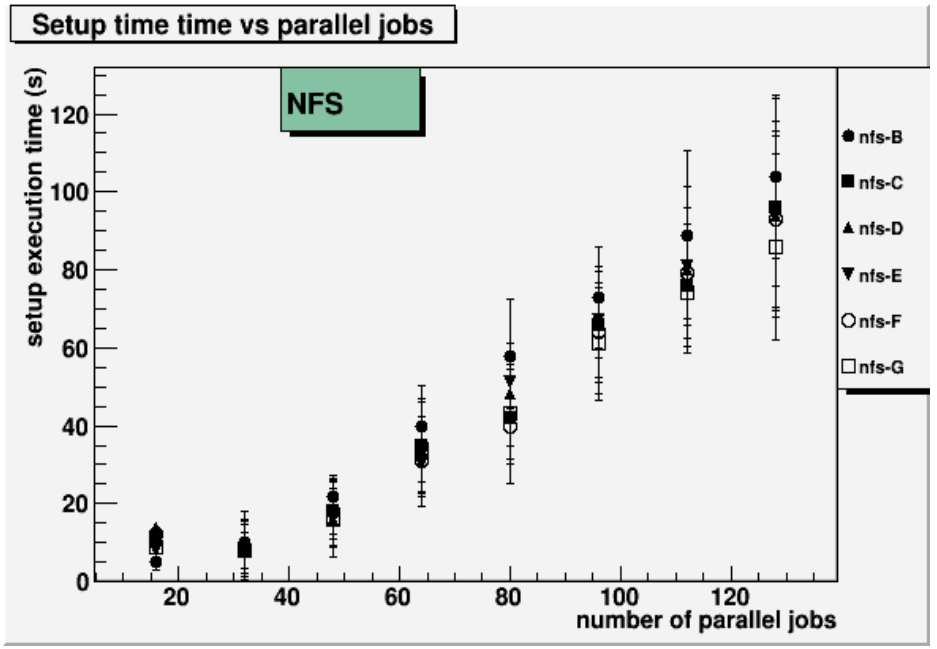
qsub a bunch of 16 jobs => all jobs start execution at the same time

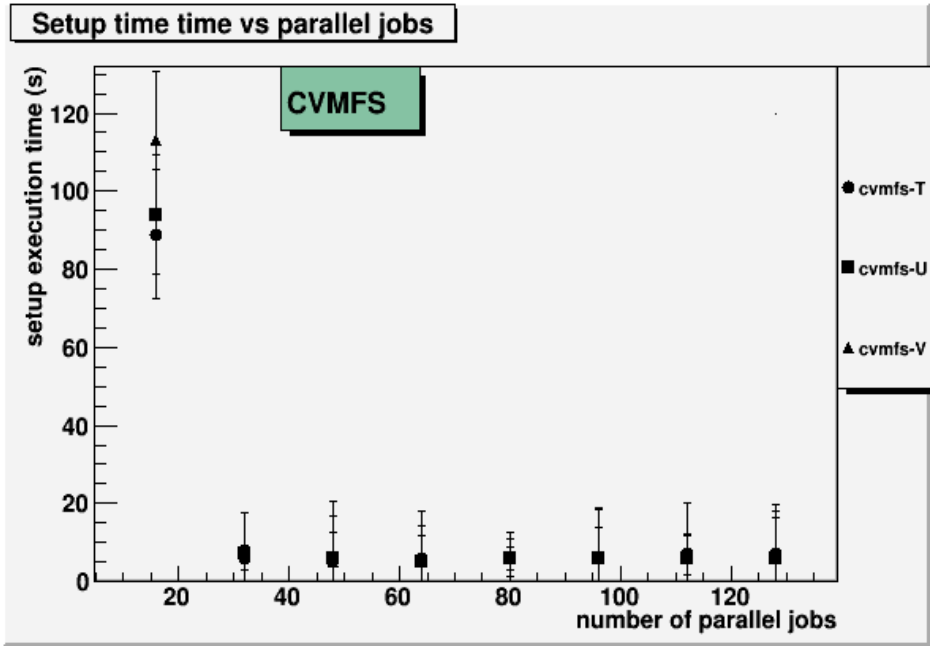
- First point: 1 job/node → total jobs 16
- ...
- Last point: 8 jobs/node ==> total jobs 128



Results: LHCb SetupProject execution time

- Clear dependence of the execution time with the number of jobs per node with NFS protocol. Mount options are ro,noatime,nolock
  - Effect on client side: 8 jobs on one node gives an execution time of 100s
- Very low execution time, and no dependence with the number of jobs per node for CernVM-FS





Space needed on the WN local disk

Space needed for the CernVM-FS cache: software releases + file catalog overhead

- LHCb:
  - One version of DaVinci (analysis package): the software takes 300 MB + CernVM-FS catalogue overhead, total space: 900 MB
  - The catalog contains file metadata for all LHCb releases
  - Downloaded once (for every new release) and then kept in cache
  - The execution of any additional version of DaVinci adds 100 MB of data
- ATLAS:
  - one release of Athena: 224MB of data + catalog files: 375MB on disk
  - The overhead is less since the catalog has been designed for ATLAS software structure – first release and then packages

About scalability

- Easily scalable adding a list of Squid servers in the CernVM-FS configuration file:

Currently in the CVMFS configuration file:

CVMFS_HTTP_PROXY="http://squid01-test.pic.es:3128"

But multiple Squids can be added:

CVMFS_HTTP_PROXY="http://squid01-test.pic.es:3128|
http://squid02.pic.es:3128|http://squid03.pic.es:3128".....

CVMFS chooses one random proxy at mount, and then automatically fails over

To be tested

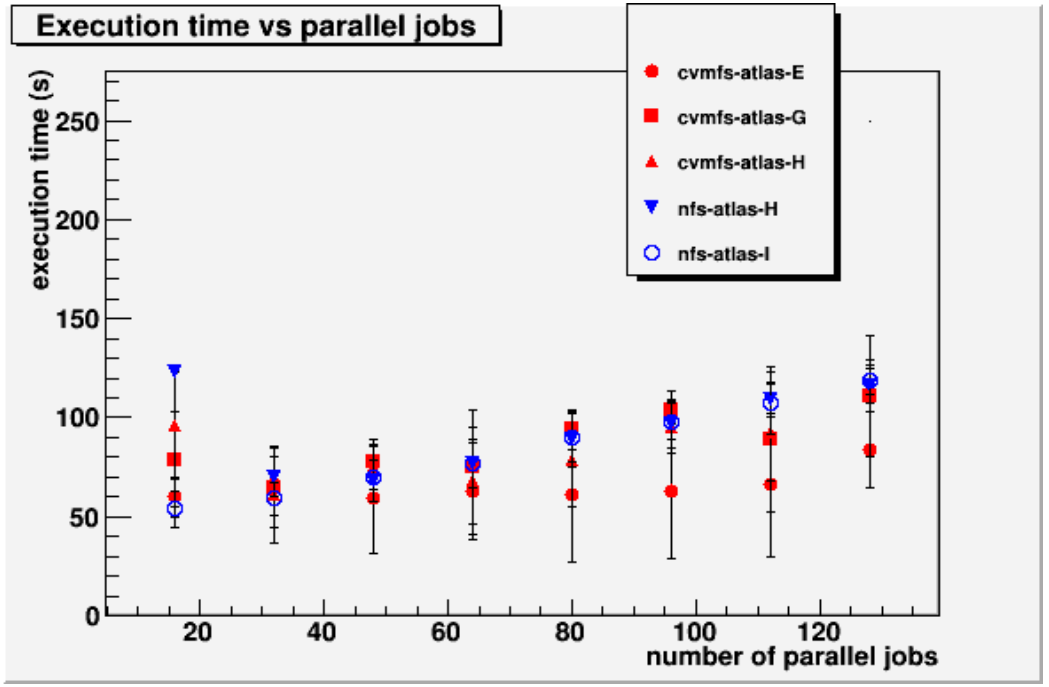
Tests for ATLAS

- Same test-bed and test script than LHCb, different executable
- Executable is an Athena test job: MC generation (1 evt) + reconstruction + analysis
- Only the total execution time is measured

Very slight difference between CernVM-FS and NFS.

- This ATLAS job accesses less data in the software area. Not possible to compare with the LHCb SetupProject

CernVM-FS performs equally or slightly better than NFS for a typical ATLAS test job



Summary and outlook

- Problems with the current model for software distribution
- Possible alternative model based on CernVM-FS: First tests very promising
  - No change required for the application or job submission (only an environment variable has to be changed)
  - Very fast execution when software cached locally
- Next to do:
  - Test scalability adding a list of Squid servers
  - Larger scale tests
  - Repeat the test with jobs especially stressing for the NFS shared area (ATLAS jobs which compile on the WN)

Acknowledgements

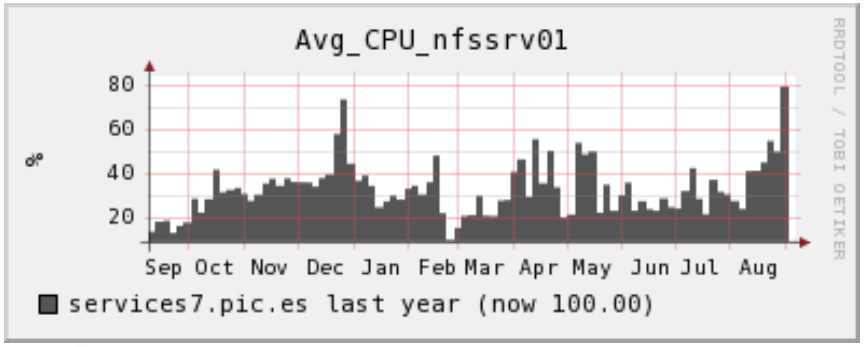
Thanks to Jakob Blomer, developer of CernVM-FS

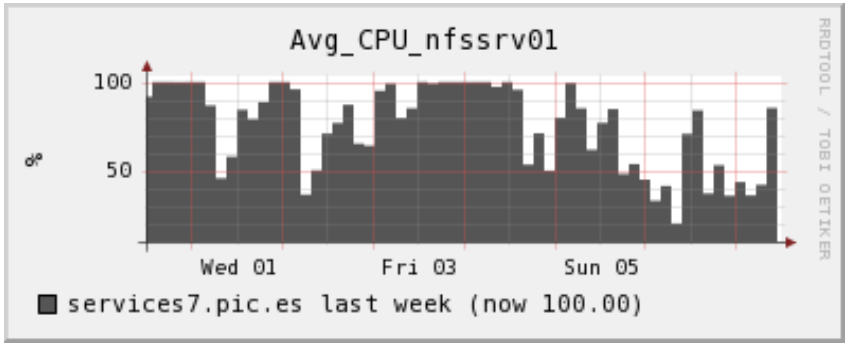
Thank you for your attention!
¿Questions?
Feedback to lanciotti@pic.es

Backup slides

Backup: NFS server setup at PIC

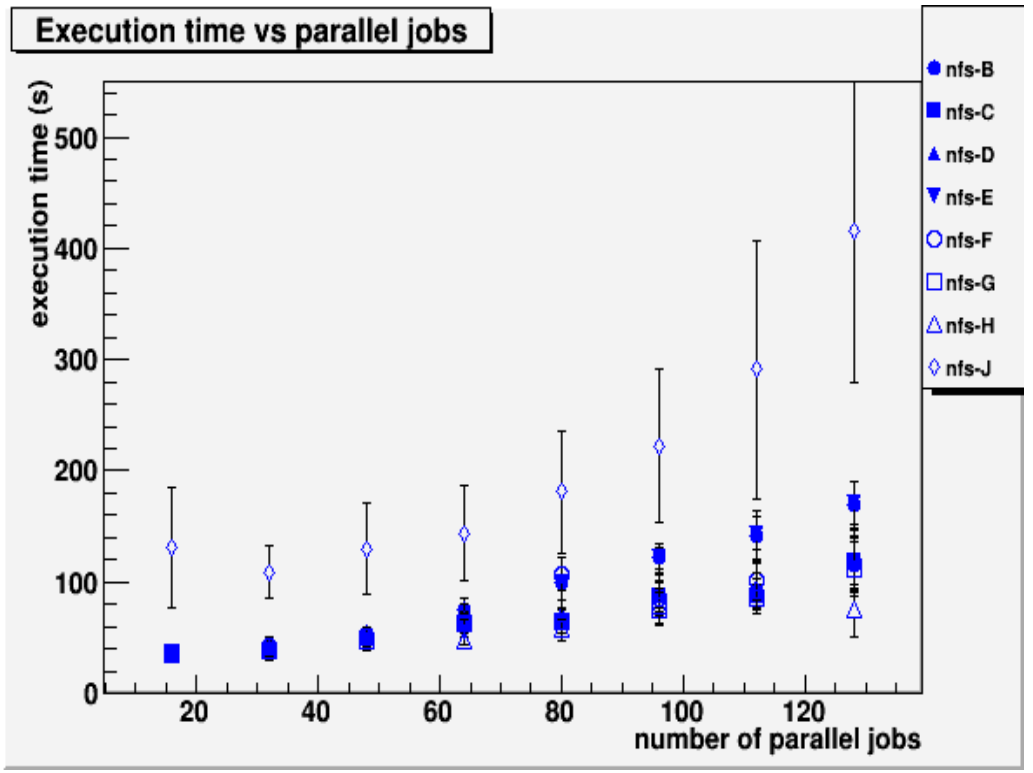
The software area is installed on a FAS2020 NetApp cabinet
http://www.netapp.com/us/products/storage-systems/fas2000/fas2000-tech-specs.html
the cabinet has:
2 controllers, in active-passive configuration (if one fails, the other takes the control of the disks).
2 disk trays. One of them has 12 SAS disks in RAID 6 configuration and hosts the software area.
Setup is scaled accordingly to PIC computing farm (2300 job slots), only last week some problem: many LHCB jobs failed with SetupProject timeout





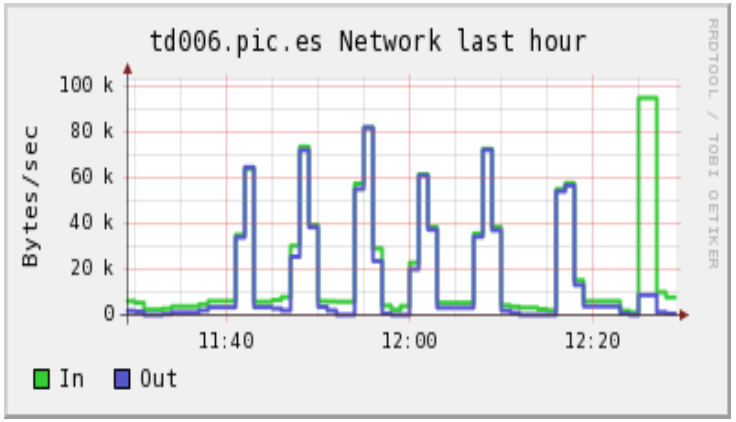
Backup: runs with NFS

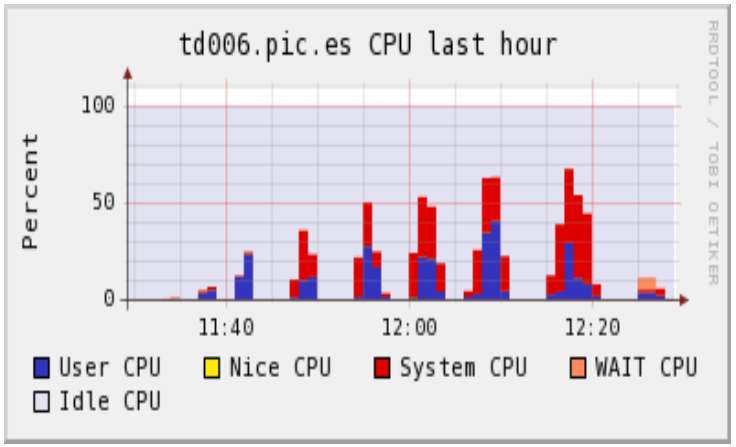
- Runs taken in different moments, corresponding to different levels of load of the NFS server. Almost no load (nfs-B) , 100% CPU usage (nfs-J)



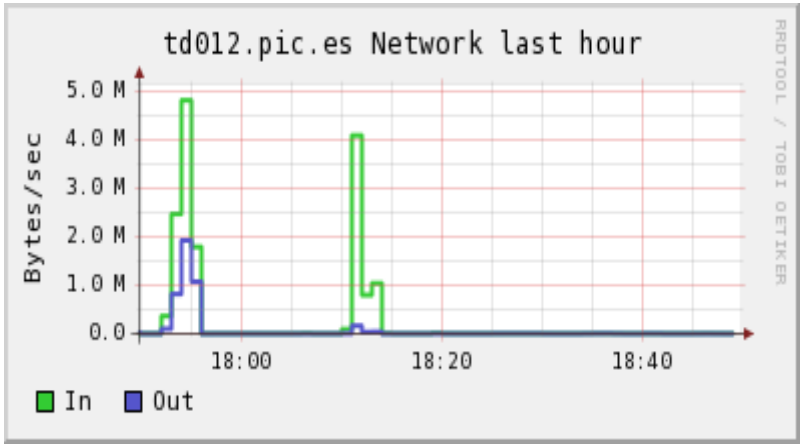
Backup: details of a run with local caching

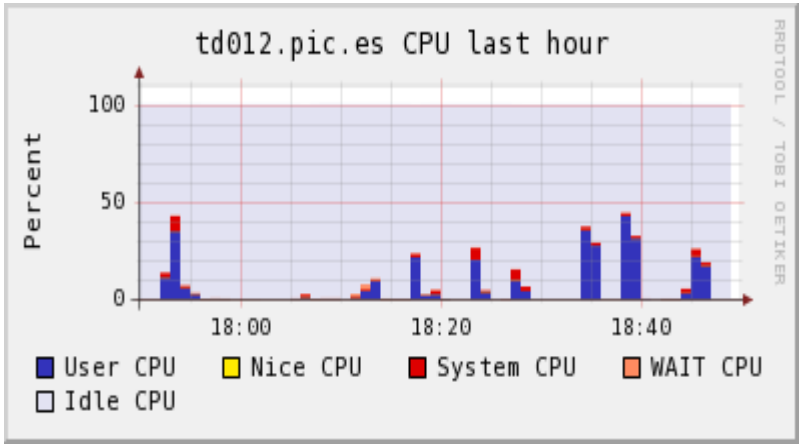
NFS protocol. Even if data are not transferred, still some network traffic between client and server (files metadata)





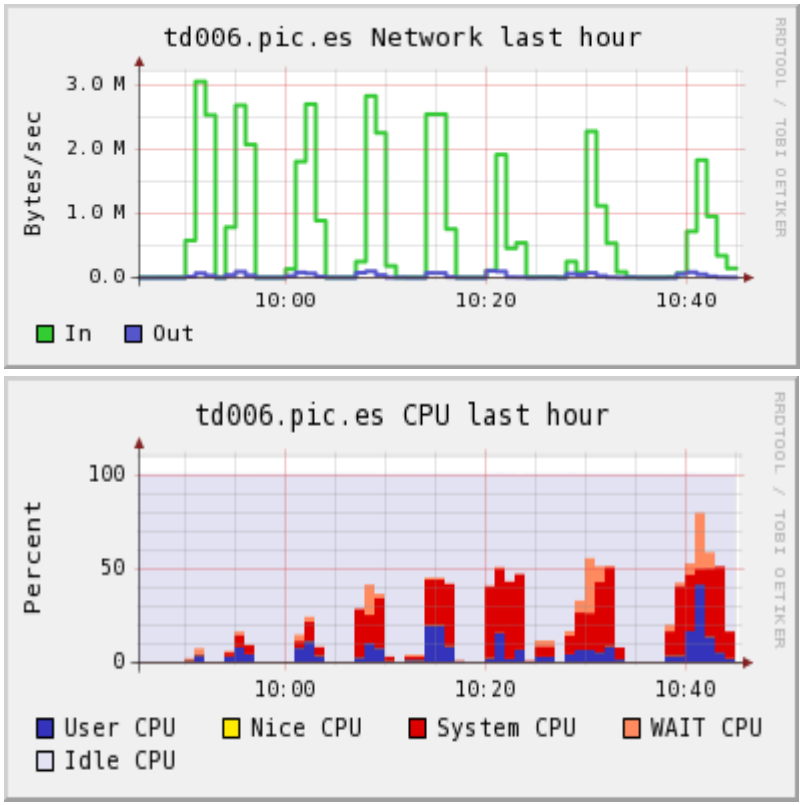
CernVM-FS protocol. First job execution triggers the software download from Squid. After, no network traffic.



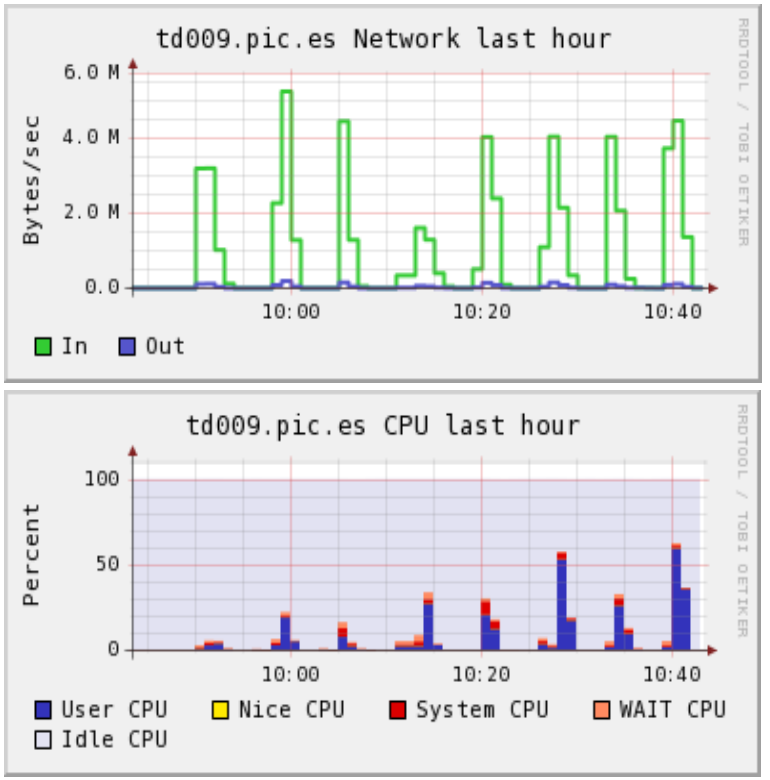


Backup: details of a run without local caching

NFS protocol. NFS cache cleaned at every job submission. Considerable use of system CPU



CernVM-FS protocol. Job execution triggers the software download from Squid at every job submission



About security

- Security is ensured in signed catalogs
- In production only for ALICE