Integration of CINECA HPC to CNAF and putting it in production

S. Dal Pra and D.Spiga on behalf of the team

Quick recap

- Goal: Run typical HEP job workloads on HPC resources, using ordinary Grid Middleware
- Use case: 30M hours on Marconi A2 HPC cluster:
  - INTEL KNL, 68 core x 4 thread = 272 “slot”
  - 96GB RAM, (10 reserved to the OS)
  - Batch System: Slurm
  - Outgoing connectivity only toward CERN, CNAF
- “whole node” jobs only
  - **A singlecore job is accounted 68 times its WCTime.**

Project name	LHC@HPC
Research field	High Energy Physics

Principal Investigator (PI)

Title (Dr., Prof., etc.)	Dr
Last name	Boccali
First name	Tommaso
Organisation name*	Istituto Nazionale di Fisica Nucleare (INFN)
Department*	Sezione di Pisa
Group*	Scientific Computing, CMS
Country	Italy

A typical Marconi A2 node	A typical WLCG node
A KNL CPU: 68 or 272(HT) cores, x86_64, rated at ~¼ the HS06 of a typical Xeon	1 or 2 Xeon-level x86_64 CPUs, 32-64 cores, O(10) HS06/thread with HT enabled
96 GB RAM, with ~10 to be reserved for the OS: 1.3-0.3 GB/thread	2GB/thread, even if setups with 3 or 4 are more and more typical
No outbound connectivity	Full outbound connectivity, sw found on CVMFS mounts, singularity
small local disk (large scratch areas via GPFS/Omnipath)	O(20 GB/thread) local scratch space
Slurm as LRMS; Whole Node only jobs, 24 h lease time, no Grid submission.	Access through a Grid CE. Single slot and 8 slot jobs; 2 to 4 days lease time
Access granted to individuals	Access via pilots and late binding; VOMS AAI for end-user access

Adapting for WLCG requirements

- To meet with our requirements, CINECA agreed to alter their default cluster configuration:
  - **Network**: Enable outbound connectivity from the Compute Nodes to CNAF (storage) and CERN (calibrations et al.) networks.
  - **Network**: Enable Inbound connectivity to a Front End host (with public IP), to be installed and configured as a HTCondor-CE
  - **SW/Storage**: Singularity, CVMFS
  - A set of local users and groups (poolaccount, requested 6 x VO, 2 as a minimum)

Specific setup and workarounds

- Local SSD on HPC nodes is way too small for WLCG jobs → job homedir reconfigured to reside on the shared GPFS area. This is done by configuring the Slurm “prolog”
- A bug in the slurm_submit.sh (blah) converts “requested cores” to “requested nodes” → patch applied
- JumboFrame support problem with GARR router, causing transfer timeout when reading from CNAF storage. → Fixed by reconfiguring the router
- At this point the minimal requirements to run a pilot were satisfied

Installing HTCondor-CE as a frontend for Slurm

- HTC-CE 3.2.2 (coupled with condor-8.8.2) has been installed on a Slurm Submit Node and made reachable at the standard 9619 TCP port.
- Completely manual setup
  - No WLCG environment (yaim, puppet, EGI repos,...)
  - unprivileged only access: rpm install upon request, edit permission granted for a few config files, a few sudo commands for CE service management
- GSI (voms, argus, poolaccount) Argus server installed in the CE host itself.

Key config points for HTCondor-CE: JOB_ROUTER

```
JOB_ROUTER_ENTRIES @=jre
[
        name = "condor_pool_dteam";
        GridResource = "batch slurm";
        TargetUniverse = 9;
        Requirements = (regexp("dteam", TARGET.x509UserProxyVoName));
        MaxJobs = 100;
        MaxIdleJobs = 4;
]
[
        name = "condor_pool_cms";
        GridResource = "batch slurm";
        TargetUniverse = 9;
        Requirements = (target.x509UserProxyVOName =?= "cms") && \
                       (target.x509UserProxyFirstFQAN =!= "/cms/Role=lcgadmin/Capability=NULL");
        MaxJobs = 1200;
        MaxIdleJobs = 200;
]
```

Key config point for Slurm **slurm_local_submit_attributes.sh**

```
#!/bin/bash
echo "#SBATCH --time=48:00:00"
echo "#SBATCH --mem=180000"
echo "#SBATCH --partition=knl_usr_prod"
echo "#SBATCH --reservation=s_res_Pra12_1234_0"
### To enable 48h run
echo "#SBATCH --qos=qos_special"
echo "#SBATCH --account=Arp11_1234_0"
echo "#SBATCH -D /scratchdir`echo $HOME |sed 's%^/mrc/home%%'`"
echo ""
echo "#### Loading singularity modulefile"
echo "module load autoload singularity"
echo 'export CINECA_SCRATCH="/scratchdir`echo $HOME |sed 's%^/mrc/home%%'`"'
echo "### Moving to specific job directory ###"
echo 'cd ${CINECA_SCRATCH}/slurm_job.${SLURM_JOB_ID}'
echo 'export TMPDIR="${CINECA_SCRATCH}/slurm_job.${SLURM_JOB_ID}"'
echo "### Setting some required directories ######"
echo 'export X509_CERT_DIR="/cvmfs/grid.cern.ch/etc/grid-security/certificates"'
echo 'export SINGULARITY_BINDPATH="/mrc_work/Arp11_1234/cms/SITECONF"'
-:---  slurm_local_submit_attributes.sh   Top L14    (Shell-script[bash])
```

Default Slurm parameters can be modified by this shell script, executed at each submission.

This is used to override default HOME dir, job lifetime, Memory limit etc. And to set needed env variables (for X509 and singularity)

Integrating with CMS

The CMS approach was to consider the HPC resources as an extension (subsite) of the CPU power provided by T1_IT_CNAF, instead of a distinct site.

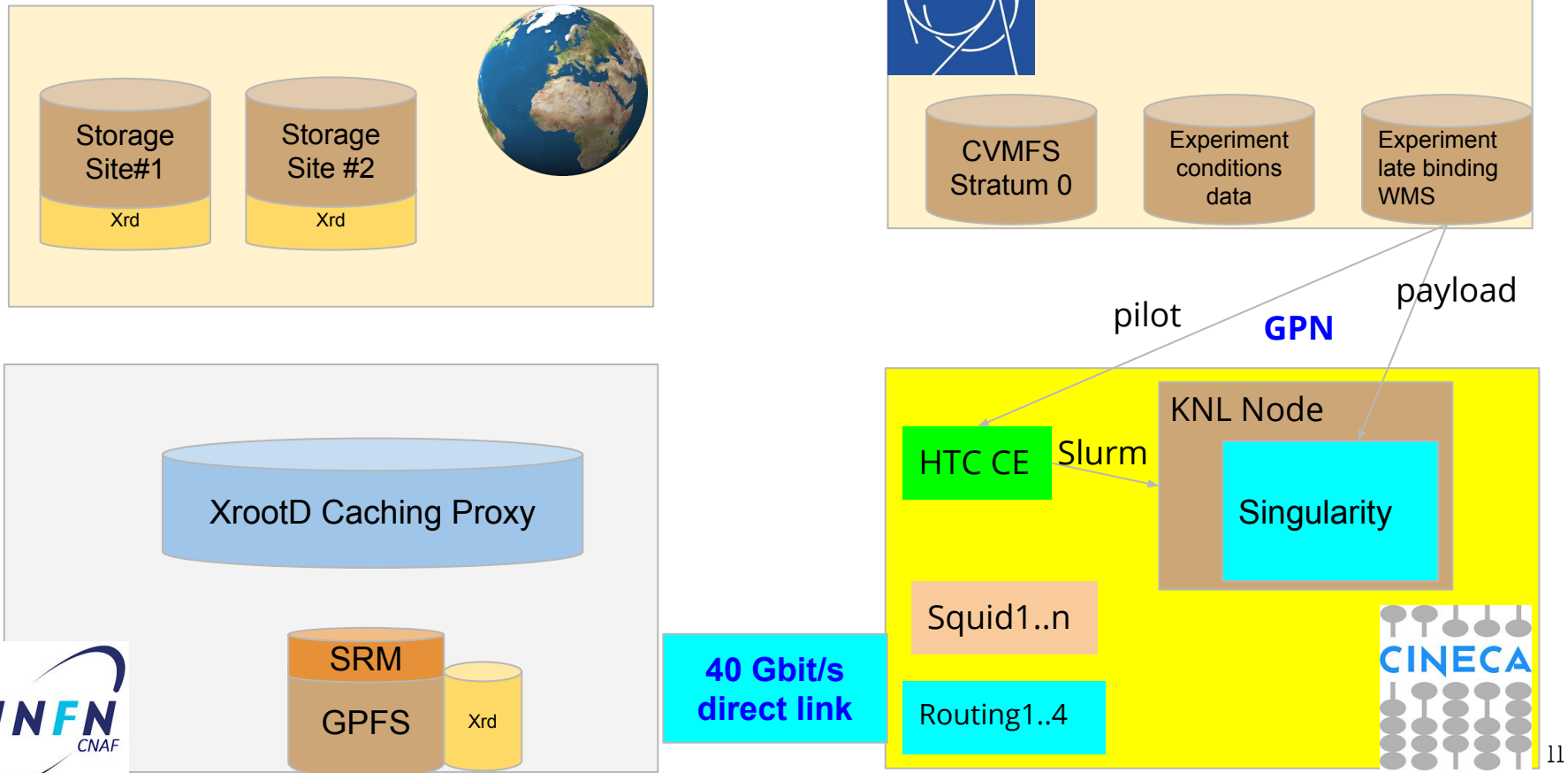
- This way it can rely on CNAF for the other facilities:
  - Output: INFN-T1 storage (via SRM)
  - Input: XRootD proxy cache @ CNAF
- Integrating CINECA with CMS boils down to adding a new CE on the GlideIn WMS Factory, making it a member of T1_IT_CNAF, however serving “different” resources.
- CINECA has been configured as **sub-site** of CNAF
  - <site name="T1_IT_CNAF">
    <subsite name="CNAF-CINECA"/>

The Runtime environment for CMS

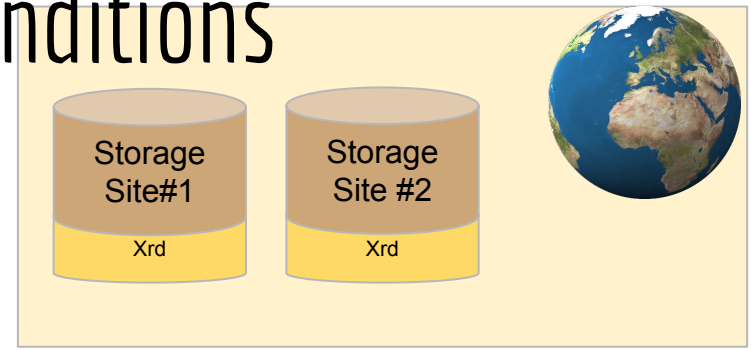
- CVMFS & Singularity
  - Ok after some troubleshooting with CINECA Admin
- Squids for CMS Frontier
  - Managed by CINECA; squids CNAF for fallback
- CMS SITECONF (Trivial File Catalog)
  - Repository for configuration files describing the CMS site: I/O parameters, squid etc.
- A set of needed environment variables:
  - X509 certificates, BIND-PATH for singularity, runtime PATH etc.

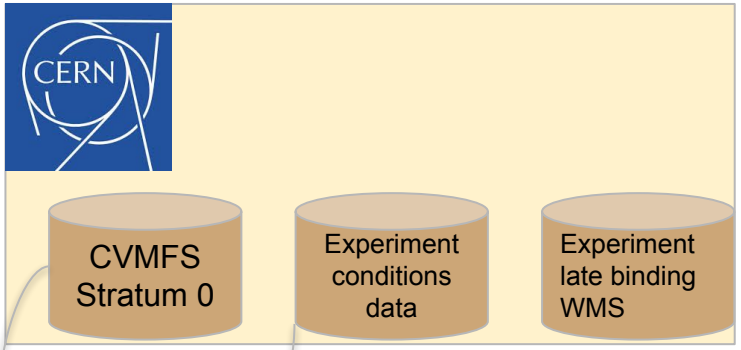
Final correct setup obtained after manually executing the glidein wrapper script on the node, then activating submission from ITB-DEV, then ITB, and finally PROD

Technical setup#1: jobs

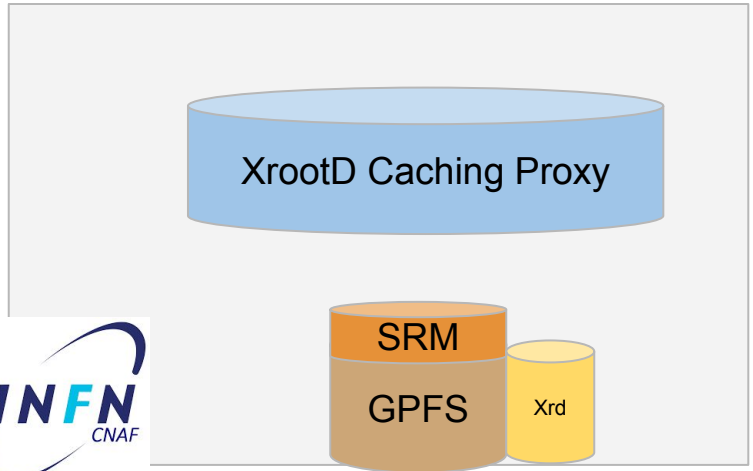


Technical setup#2: sw and conditions

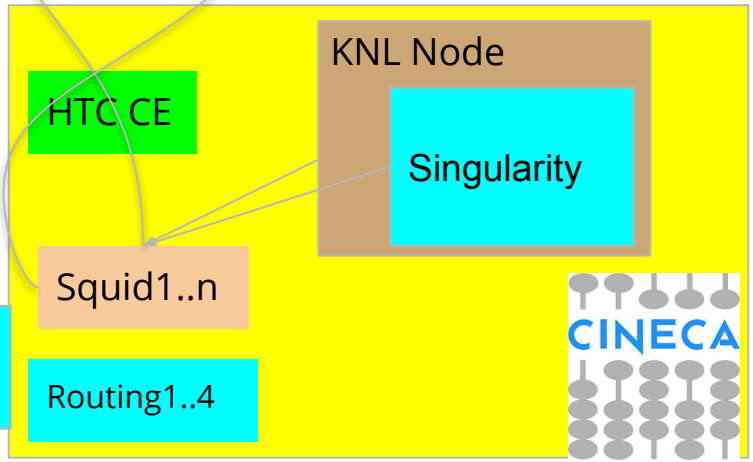




GPN

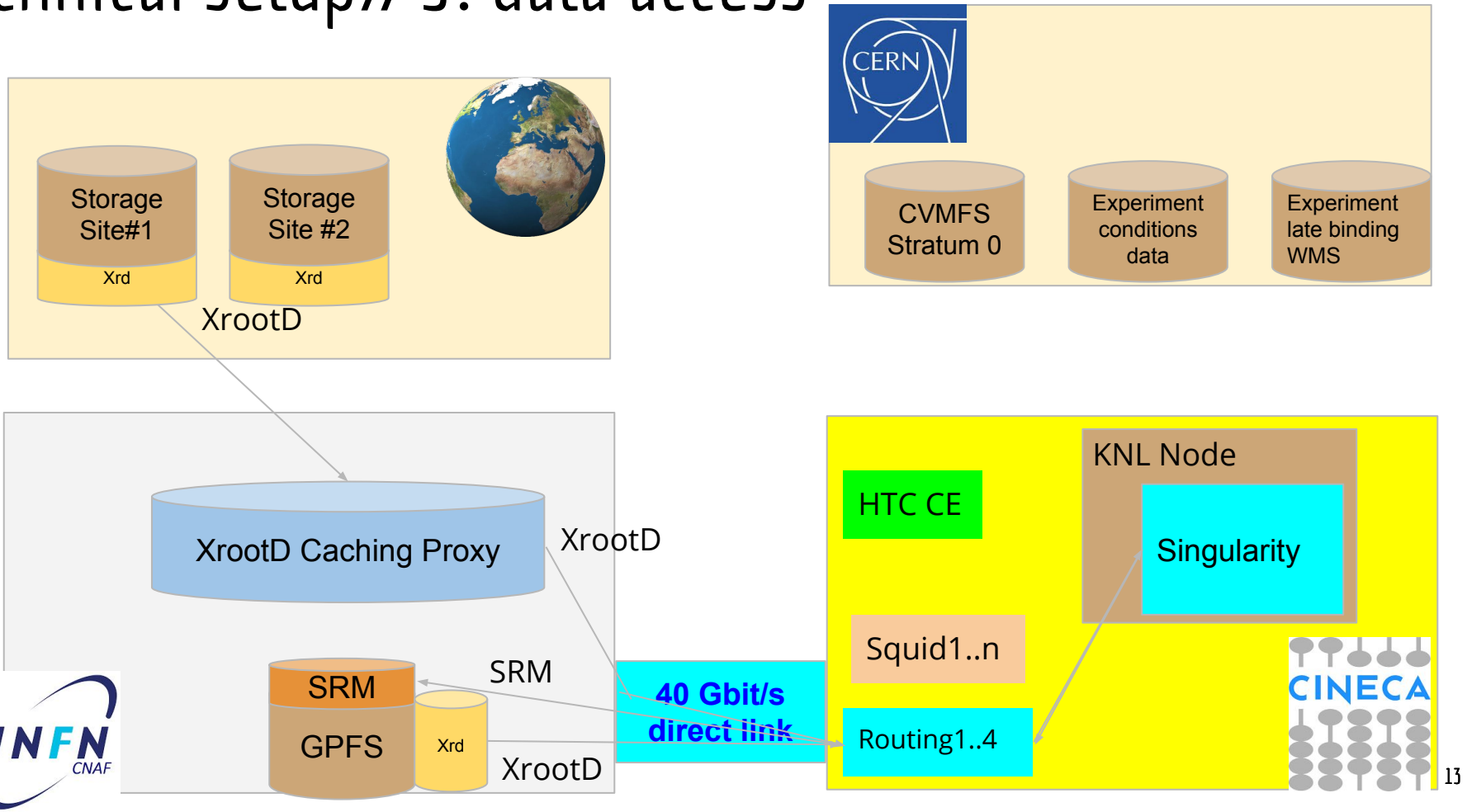


40 Gbit/s direct link



INFN CNAF

Technical setup#3: data access



CINECA integration @ CMS

- **From CMS computing perspectives**, the integration is based on the implementation of a **fully transparent extension of T1@CNAF** with the HPC
  - Transparent for operations, avoiding any custom workflow setup and ad hoc solutions
  - “CINECA does not exist” for CMS operators
- Although both Analysis and Production workflows have been **successfully tested**, we restricted CINECA to only run production/processing workflows (see later), because at least initially, it is more “under control”

CINECA Resources configuration for CMS

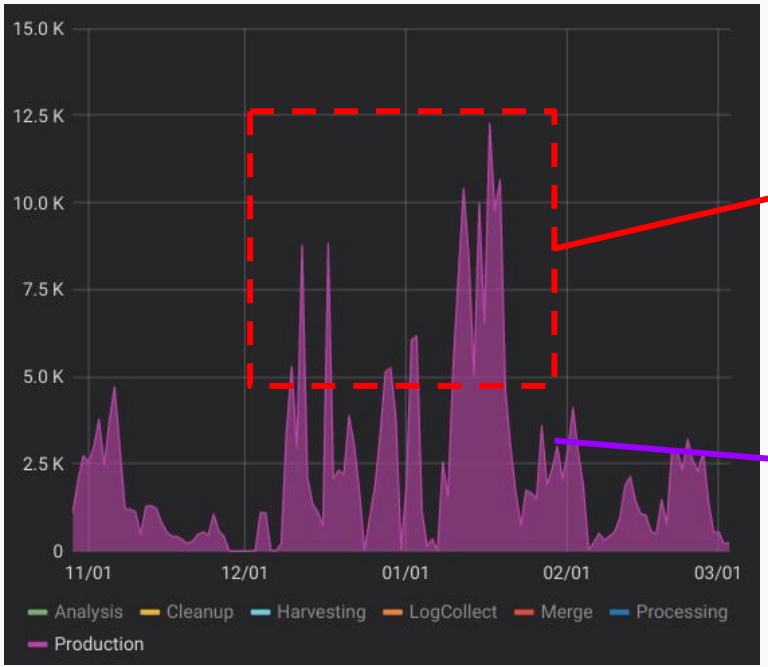
- The dedicated HTCondor-CE has been added to the Glidein Frontend within the pool of T1-CNAF CEs
- CINECA has been configured as **sub-site** of CNAF
  - <site name="T1_IT_CNAF">
    <subsite name="CNAF-CINECA"/>
- Job selection is done using the **custom-pilot approach**
  - The WN only matches with low-memory jobs, (lower RAM/core on KNL) and specific workloads (low I/O), no user analysis
  - This mechanism is versatile and can be adapted for more special nodes (GPUs, I/O devices or networking, ...)

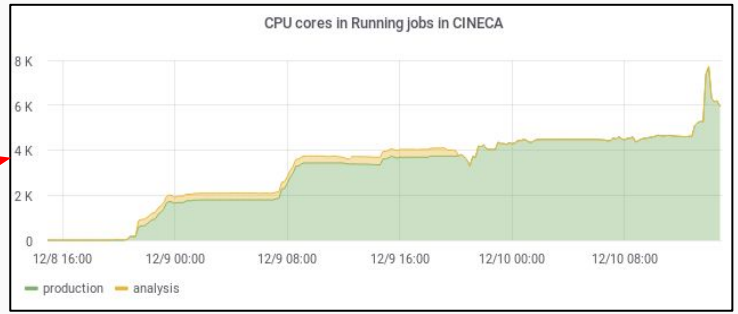
Input and Output data flow

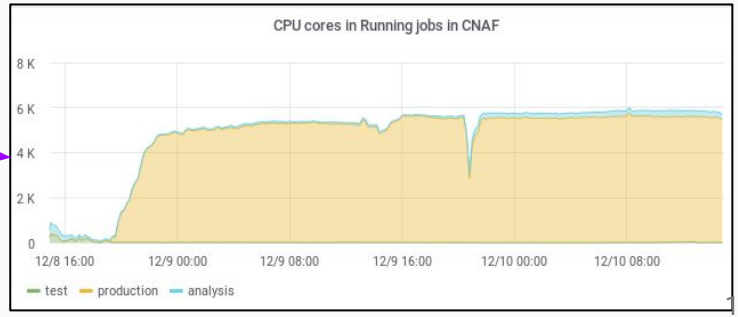
- Inputs and outputs follow different paths
  - **Inputs** are accessed via Xrootd protocol directly from CNAF if there; otherwise through **Xrootd cache** deployed at CNAF, in order to overcome the routing limitation
    - <lfn-to-pfn protocol="xrootd" destination-match=".*" path-match="/+store/(.*)" result="root://xcache.cr.cnaf.infn.it:31094//store/$1"/>
    - From the jobs point of view, CNAF “has all CMS files”
  - **Outputs** are copied to CNAF storage via SRM
    - And in turn registered/handled by the Data Management

CMS @ CNAF == CPUs of (INFN-T1 + CINECA), Storage of INFN-T1

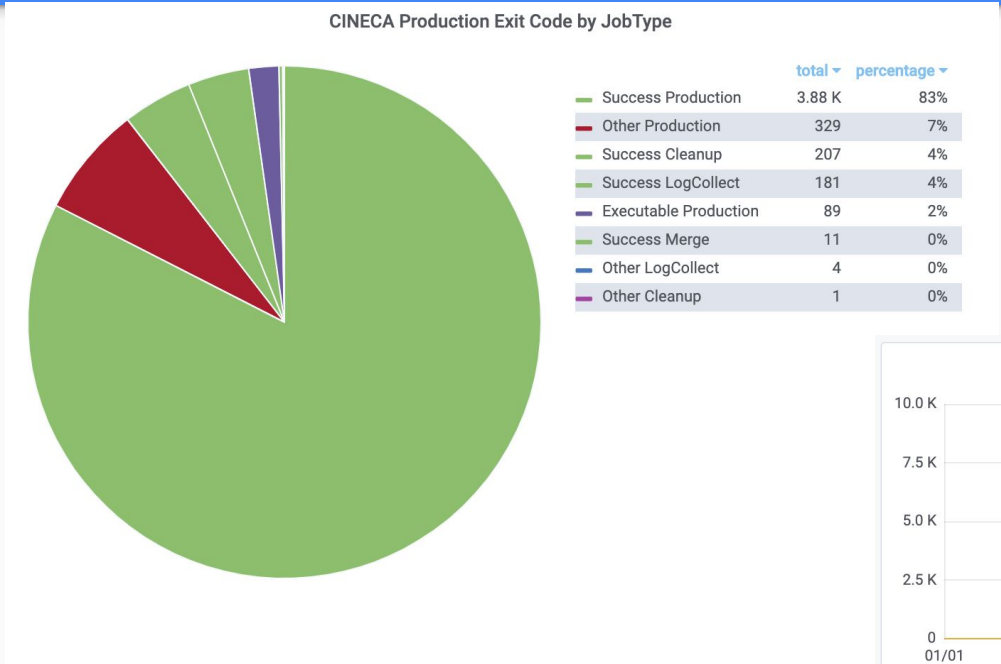
Production jobs Running at T1_IT_CNAF



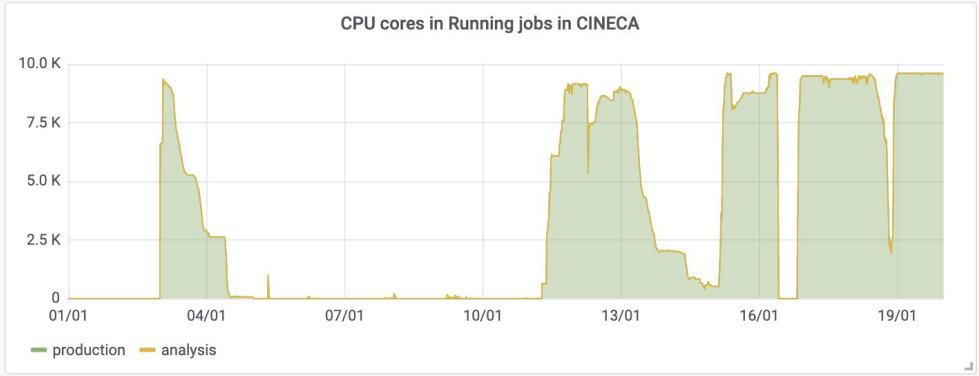




The CINECA Integration at CMS is Production Ready



- 92% of successful jobs is compatible with what CMS sees at more standard sites.
- CMS sees CNAF as a processing site able to serve > 17k threads (~7k at CNAF and 10k at CINECA).

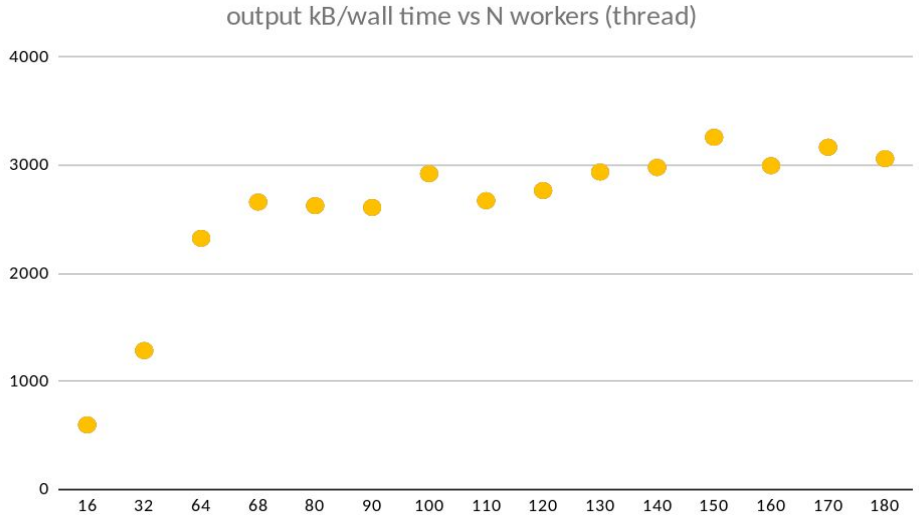


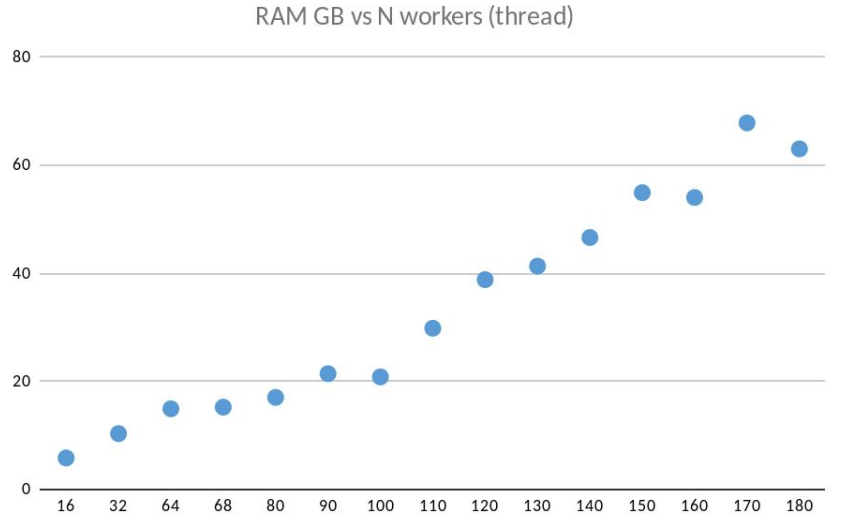
Alice

- The ALICE approach was to test the multithread sw to be used for RUN-3, available via CVMFS, and to be executed in singularity / Co7.
- Early tests via direct manual submission to the slurm cluster
  - Multithread Pb Pb, some sw bugfix on the way, different number of threads
  - Optimal job size: 180 thread --> (73GB RSS max, 0.4GB/thread)
- Next tests performed via Grid submission
- Output written via Xrootd @ CNAF or CERN

Alice

Several tests performed, to find the “best fit” for wholenode jobs





ATLAS, LHCb

- Pilot submission ok
- Work in progress for best workload tuning

Current status, next steps

- Marconi A2 unavailable from January 16, with only 18% of the 30M hours used so far; it will be available from April.
- 2 Nodes available for test purposes in the meanwhile
- Bring to prod level the other experiments
- Use the remaining 82% of the remaining time.

Next Steps: Site Accounting

- WallClockTime Accounting per VO can be performed at HTC-CE level

[sdalpra0@r000u11l06 ~]$ condor_ce_history -cons '((GridJobStatus =?= "COMPLETED") && (JobStartDate =!= undefined) && (RoutedToJobId =?= undefined))' -af x509UserProxyVOName ClusterId '((CompletionDate - JobStartDate))' '((CompletionDate - JobStartDate) * 68)/3600.0' Owner JobStartDate CompletionDate

cms 255723 170296 3216.702222222222 a07cms02 1583488889 1583659185

atlas 253973 65 1.227777777777778 a07atl00 1583488841 1583488906

atlas 253856 70 1.322222222222222 a07atl00 1583488728 1583488798

cms 252042 155394 2935.22 a07cms02 1583333394 1583488788

- Complete accounting data remain at Batch System level (sacct)
- Some effort needed to also integrate accounting

Conclusions

- Standard Grid Middleware was successfully exploited to run HEP payloads on HPC resources
- Payloads should be carefully selected to fully exploit whole node resources, having unusual RAM/core ratio.
- Adopted strategies so far proved flexible enough and can be adapted reasonably well to newer scenarios.
- Much better infrastructural integration between CNAF and CINECA is expected after the migration to the new Tecnopolo will be completed