Elastic CNAF Datacenter extension via opportunistic resources

INFN-CNAF

Istituto Nazionale di Fisica Nucleare

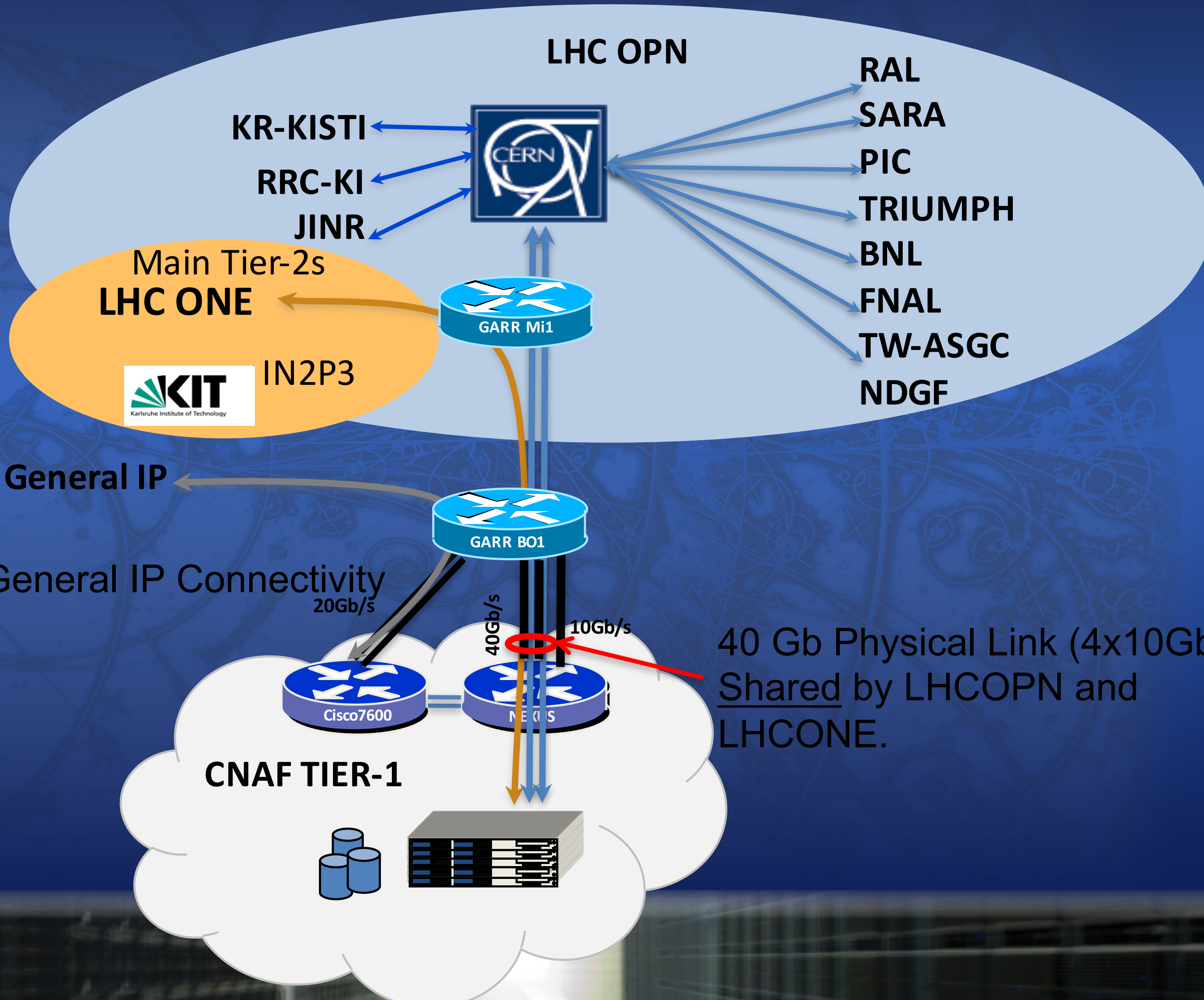
INFN

- **National Institute for Nuclear Physics** (INFN) is a research institute funded by the Italian government
- Composed by several units
  - 20 units dislocated in the main Italian University Physics Departments
  - 4 Laboratories
  - 3 National Centers dedicated to specific tasks
- CNAF is a National Center dedicated to computing applications

The Tier-1 at INFN-CNAF

- WLCG Grid site dedicated to HEP computing for LHC experiments (ATLAS, CMS, LHCb, ALICE) works with ~30 other scientific groups
- 1.000 WNs , 20.000 computing slots, 200k HS06 and counting.
  - LSF as current Batch System, Condor migration foreseen
- 22PB SAN disk (GPFS), 27PB on tape (TSM) integrated as an HSM
  - Also supporting LTDP for CDF experiment
- Dedicated network channel (LHC OPN, 20Gb/s) with CERN Tier-0 and T1s, plus 20GB/s (LHC ONE) with most of the T2s
  - 100Gbps connection in 2017
- Member of HNSciCloud European project for testing hybrid clouds for scientific computing

WAN@CNAF



20 Gb/s For General IP Connectivity

40 Gb Physical Link (4x10Gb) Shared by LHCOPN and LHCONE.

Extension use-cases

- Elastic opportunistic computing with transient **Aruba** resources. CMS selected for test&setup
- **ReCaS/Bari**: extension and management of remote resources
  - These will become pledged resources for CNAF

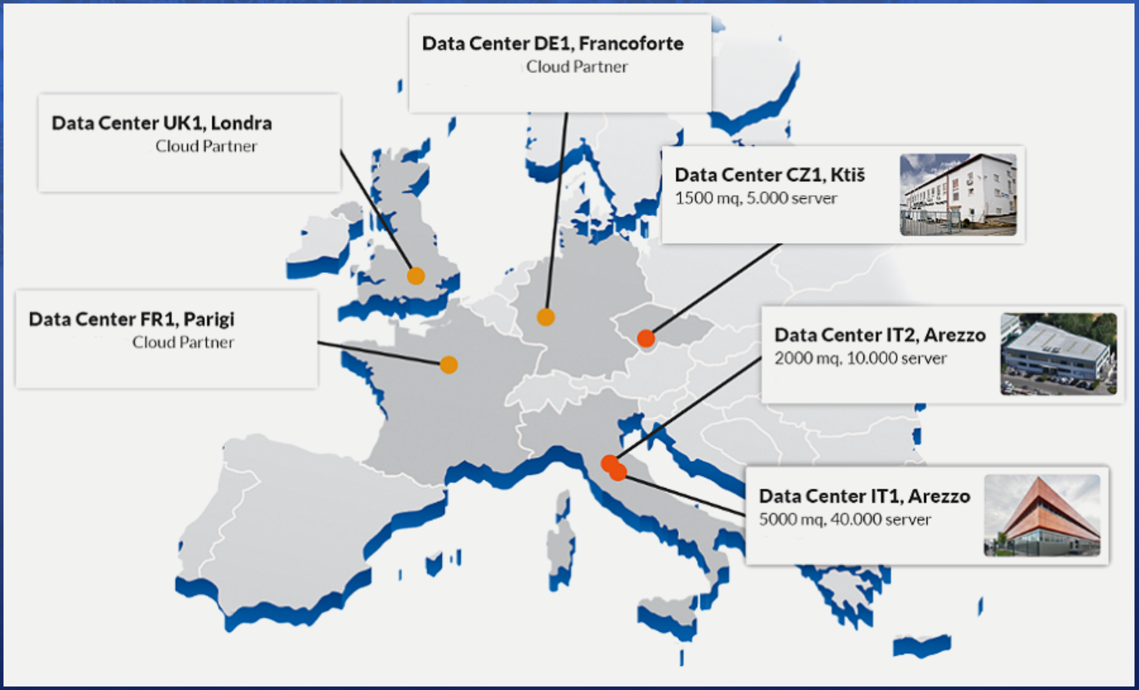
Use-case 1: Aruba

Pros of Opportunistic computing

- CMS
  - Take advantage of (much) more computing resources.
    - CONS: transient availability
- ARUBA
  - Study case in order to provide unused resources to an “always hungry” customer
- INFN-T1
  - Test transparent utilization of remote resources for HEP (proprietary or opportunistic)

Aruba

- One of the main Italian resource providers
  - Web, host, mail, cloud ...
- Main datacenter in Arezzo (near Florence)



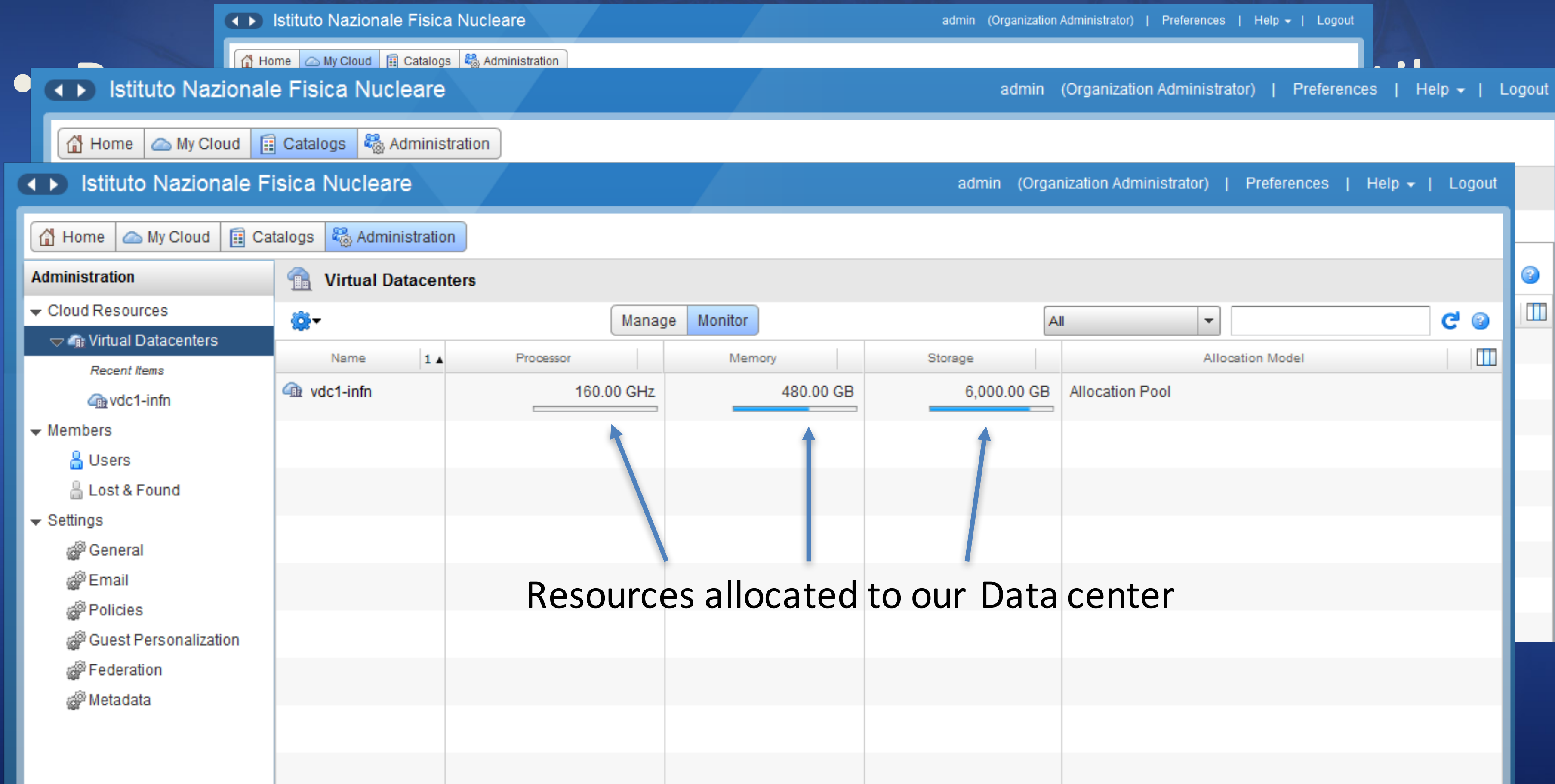
The CMS Experiment at INFN-T1

- 48k HS06 of CPU power, 4PB of online Disk storage and 12PB of tape
- Implemented all majors computing activities
  - Monte Carlo simulations
  - Reconstruction
  - End-user analysis
- The 4 LHC experiments are close enough in requests / workflows
  - extension to the other 3 under development

The use-case

- Early agreement CNAF - Aruba
  - ARUBA provides an amount of Virtual resources (CPU cycles, RAM, DISK) to deploy a remote testbed
    - VMWare dashboard
  - When Aruba customers require more resources, the CPU Freq. of the provided VMs in the testbed is lowered down to a few MHz **(not destroyed!)**
- Goal
  - Transparently join these external resources “as if they were” in the local cluster, and have LSF dispatching jobs there when available
  - Tied to CMS-only specifications for the moment
  - Once fully tested and verified, extension to other experiments is
    - Trivial for other LHC experiments
    - To be studied for non-LHC VOs

VM Management via VMWare

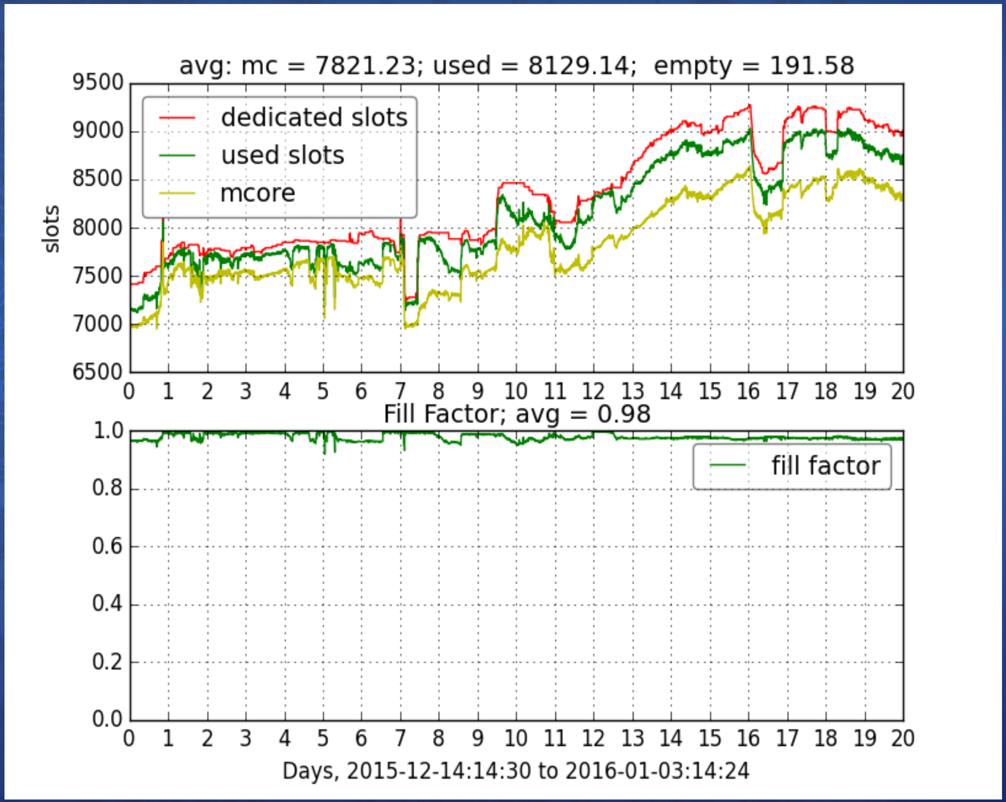


The CMS workflow at CNAF

- Grid pilot jobs submitted to CREAM CEs
  - Late binding: we cannot know in advance what kind of activity it's going to perform
- Multicore only
  - 8 core (or 8 slot) jobs: CNAF dedicates a dynamic partition of WNs to such jobs
- SQUID proxy for Software and Condition DB
- Input files on local GPFS disk, fallback via Xrootd, O(GB) file size
- Output file staged through SRM (StoRM) at CNAF.

The dynamic Multicore partition

- CMS jobs run in a dynamic subset of hosts dedicated to multicore-only jobs.
- Elastic resources shall be member of this subset.



Adapting CMS for Aruba

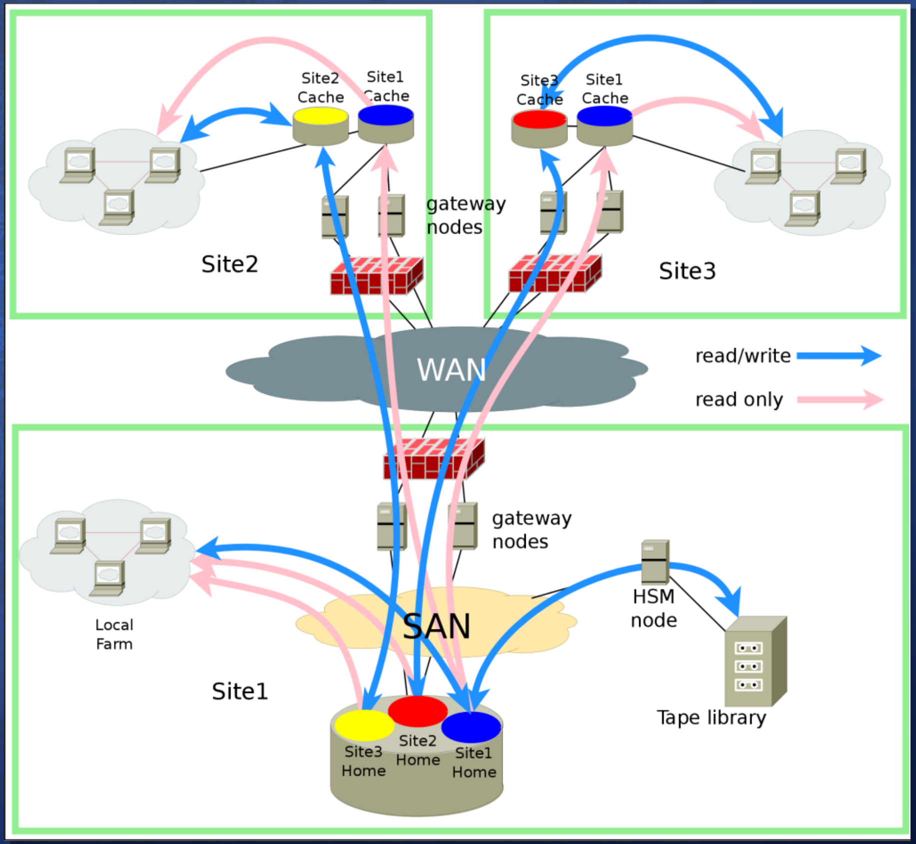
- Main idea: transparent extension
  - Remote WN join the LSF cluster at boot “as if” local to the cluster
- Problems:
  - Remote Virtual WN need read-only access to the cluster shared fs (/usr/share/lsf)
  - VMs have private IP, are behind NAT & FW, outbound connectivity only, but have to be reachable by LSF
  - LSF needs host resolution (IP ↔ hostname) but no DNS available for such hosts

Adapting CMS for Aruba

- Solutions:
- Read-only access to the cluster shared fs
  - Provided through GPFS/AFM
- Host resolution
  - LSF has his own version of /etc/hosts
  - This requires to declare a fixed set of Virtual nodes
- Networking problems solved using **dynfarm**:
  - *Service developed at CNAF to provide integration between LSF and virtualized computing resources.*

Remote data access via GPFS AFM

- GPFS AFM
  - A cache providing geographic replica of a file system
  - manages RW access to cache
- Two sides
  - Home - where the information lives
  - Cache
  - Data written to the cache is copied back to home as quickly as possible
  - Data is copied to the cache when requested
- Configured as Read-only for site extension



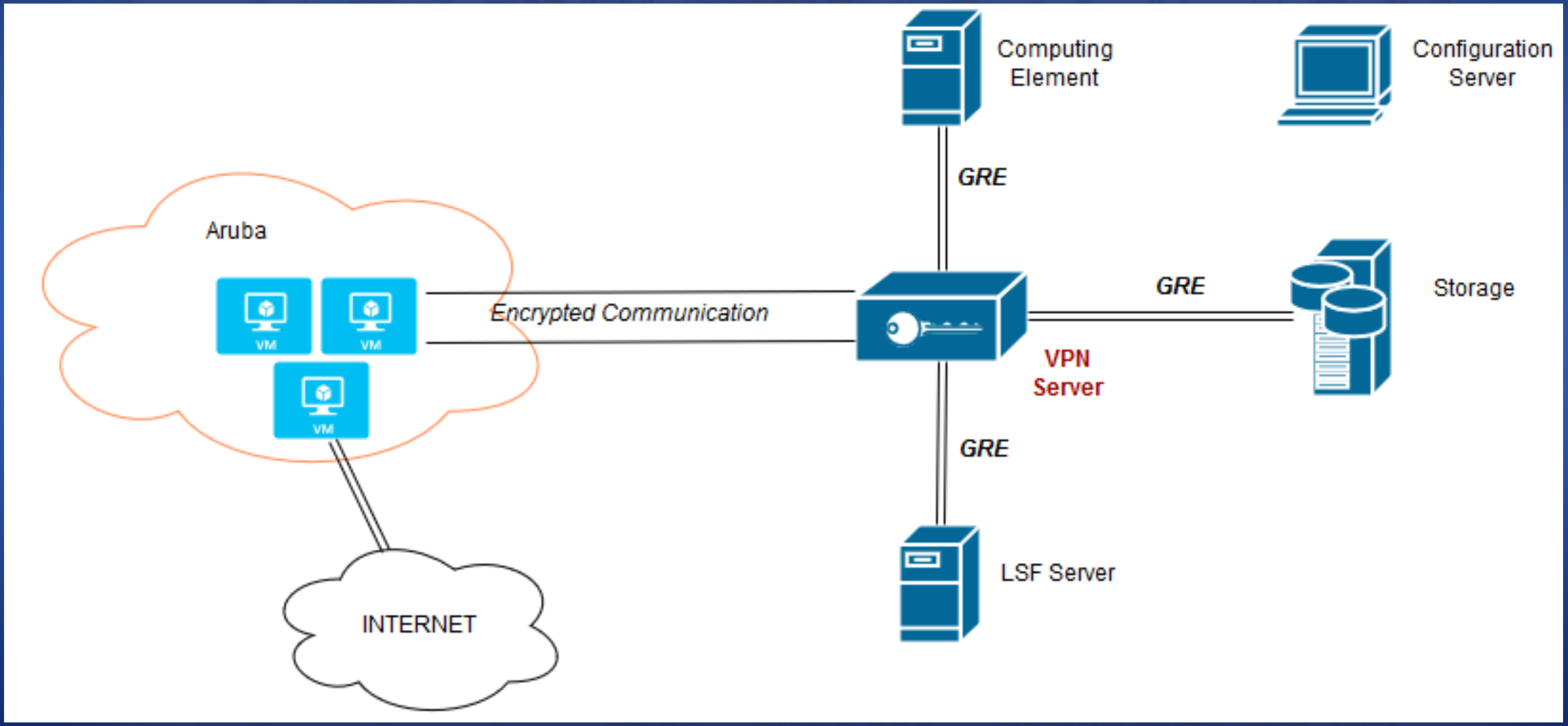
Dynfarm concepts

- The VM at boot connects to a OpenVPN based service at CNAF
  - It authenticates the connection (X.509)
  - Delivers parameters to setup a tunnel with (only) the required services at CNAF (LSF, CEs, Argus)
  - Routes are defined on each server to the private IPs of the VMs (GRE Tunnels)
  - Other traffic flows through general network

Dynfarm deployment

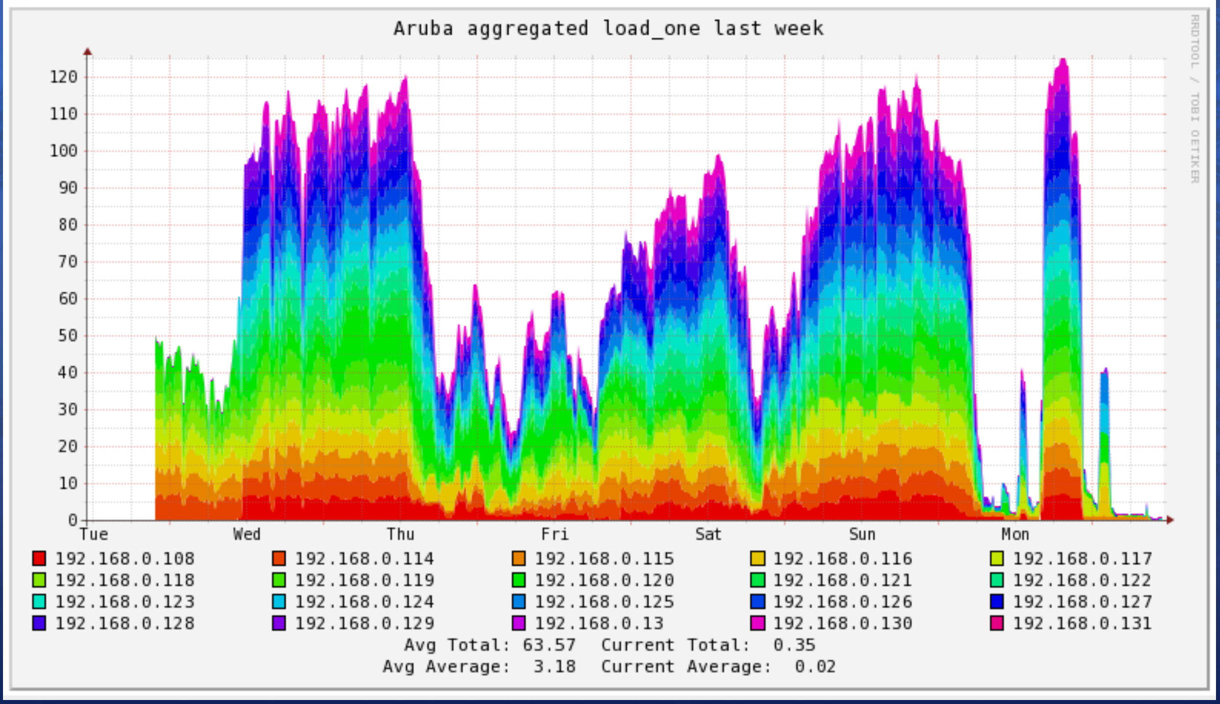
- VPN Server side, two RPMs:
  - dynfarm-server, dynfarm-client-server
    - In the VPN server at CNAF. First install creates one dynfarm_cred.rpm which must be present in the VMs
- VM side, two RPMs:
  - dynfarm_client, dynfarm_cred (contains CA certificate used by VPN server and a key used by dynfarm-server)
- Management: remote_control <cmd> <args>

Dynfarm workflow



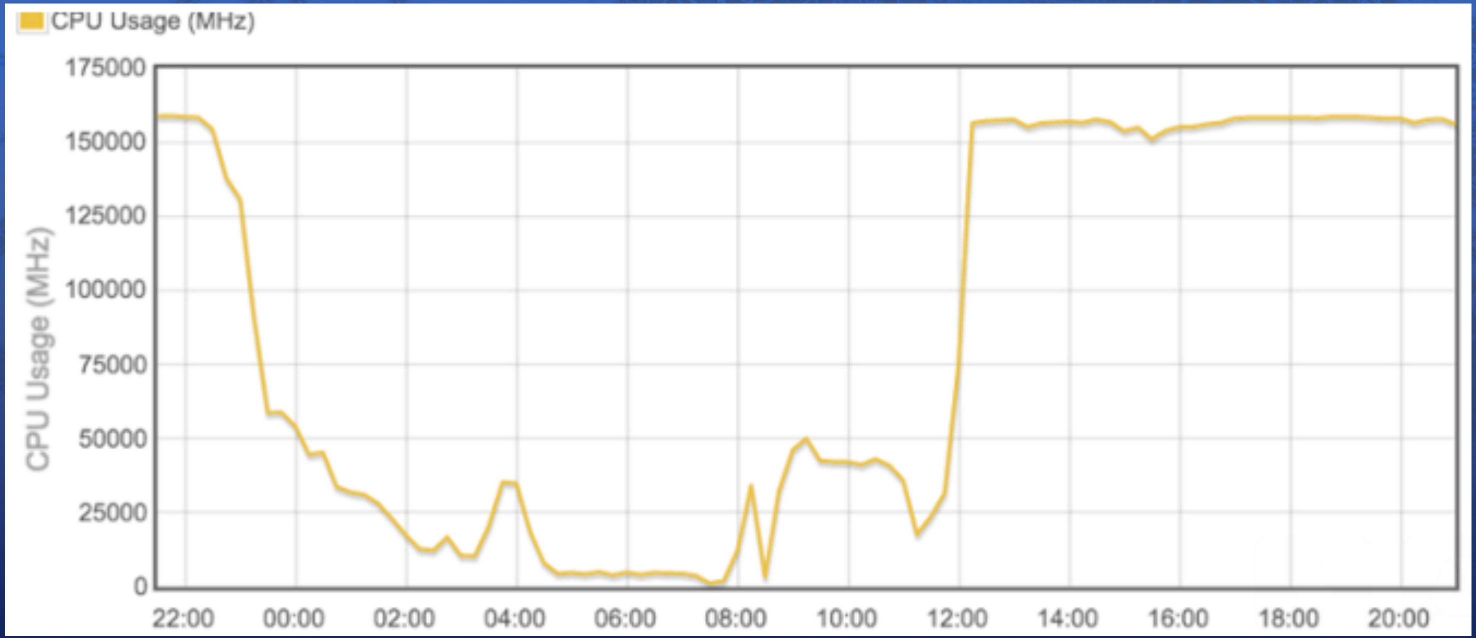
Results

- Early successful attempts from Jun 2015
- Different configurations (tuning) have followed



Results

- 160GHz total amount of CPU (Intel 2697-v3).
  - Assuming 2GHz/core → 10 x 8-cores VMs (possible overbooking)



Results

- Currently the remote VM run the very same jobs delivered to CNAF by GlideinWMS
- Job efficiency on elastic resources can be very good for certain type of jobs (MC)
- Special configuration at GlideIN can specialize delivery for these resources.

Queue	Site	Njobs	Avg_eff	Max_eff	Avc_wct	Avg_cpt
CMS_mc	AR	2984	0.602	0.912	199.805	130.482
CMS_mc	T1	41412	0.707	0.926	117.296	93.203

Use-case 2: ReCaS/Bari

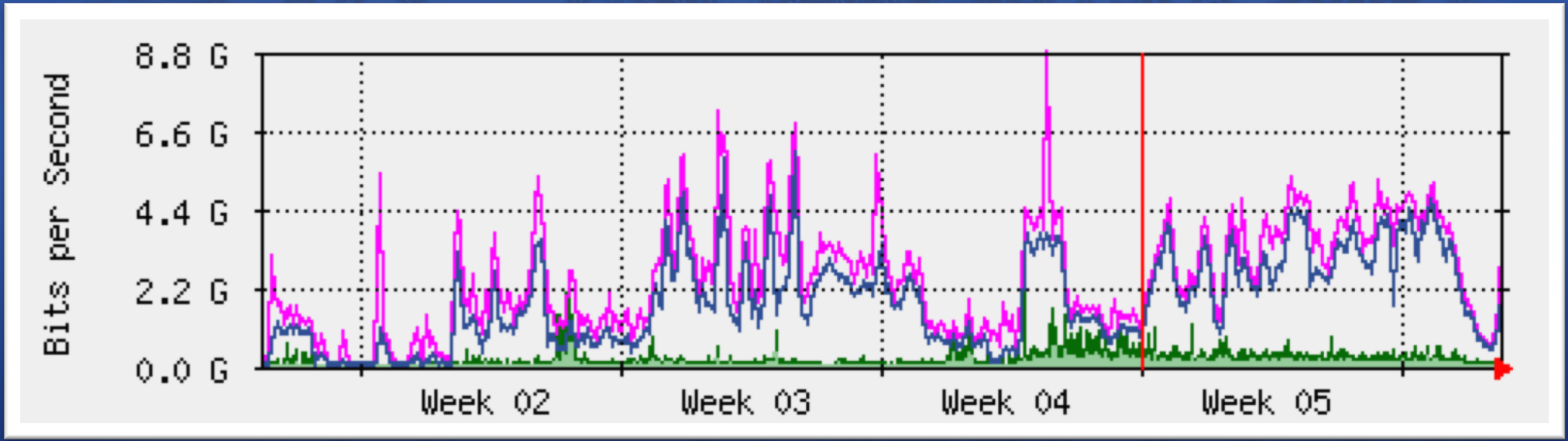
Remote extension to ReCaS/Bari

- ~17.5k HS06, ~30WN, 64 core, 256GB RAM
  - 1 core / 1 slot, 4GB/slot, 8,53 HS06/slot (546HS06/WN)
- Dedicated network connection with CNAF:
  - VPN lev. 3, 20Gb/s
  - Routing through CNAF, IP of remote hosts in the same network range (plus 10.10.x.y for ipmi access)
  - Similar to CERN/Wigner extension
- *Direct and transparent access from CNAF*

Deployment

- Two infrastructure VMs to offload network link:
  - CVMFS and Frontier SQUID (used by ATLAS and CMS)
    - SQUID requests are redirected to the local VMs
- Cache storage GPFS/AFM
  - 2 server, 10 Gbit
  - 330TB (Atlas, CMS, LHCb)
  - LSF shared file system also replicated

Network traffic (4 weeks)



Current issues and tuning

- Latencies in the shared fs can cause troubles
  - Intense I/O can lead to timeout :

ba-3-x-y: Feb 8 22:56:51 ba-3-9-18 kernel: nfs: server nfs-ba.cr.cnaf.infn.it not responding, timed out

- CMS: fallback to Xrootd (excessive load on the AFM cache)

Comparative Results

Queue	Nodetype	Njobs	Avg_eff	Max_eff	Avg_wct	Avg_cpt
Cms_mc	AR	2984	0.602	0.912	199.805	130.482
Alice	T1	98451	0.848	0.953	16.433	13.942
Atlas_sc	T1	1211890	0.922	0.972	1.247	1.153
Cms_mc	T1	41412	0.707	0.926	117.296	93.203
Lhcb	T1	102008	0.960	0.985	23.593	22.631
Atlas_mc	T1	38157	0.803	0.988	19.289	18.239
Alice	BA	25492	0.725	0.966	14.446	10.592
Atlas	BA	15263	0.738	0.979	1.439	1.077
Cms_mcore	BA	2261	0.444	0.805	146.952	69.735
Lhcb	BA	13873	0.916	0.967	12.998	11.013
Atlas_sc	BA	20268	0.685	0.878	24.378	15.658

Conclusions

Aruba

- Got the opportunity to test our setup on a pure commercial cloud provider
  - Developed dynfarm to extend our network setup
    - Core dynfarm concept should be adaptable to other Batch Systems
  - Gained experience on yet another Cloud Infrastructure: Vmware
- Job efficiency encouraging
  - Even better when we will be able to forward to Aruba only non-IO intensive jobs
- Scale of the test quite small, did not reach any bottleneck
- Tested with CMS, other LHC experiments may join in future
- Accounting problematic due to possible GHz reduction
- Good exercise for HNSciCloud too

ReCaS/Bari

- T1-Bari farm extension “similar” to CERN-Wigner
- Job efficiency (compared to native T1) highly depending on storage usage
  - Better efficiency means job on WN is mainly CPU bound (or input file already in cache before start)
- General scalability limited by the width of dedicated T1→BA link (20Gb/s)
- Assistance on faulty nodes somehow problematic