

## Elastic CNAF DataCenter extension via opportunistic resources

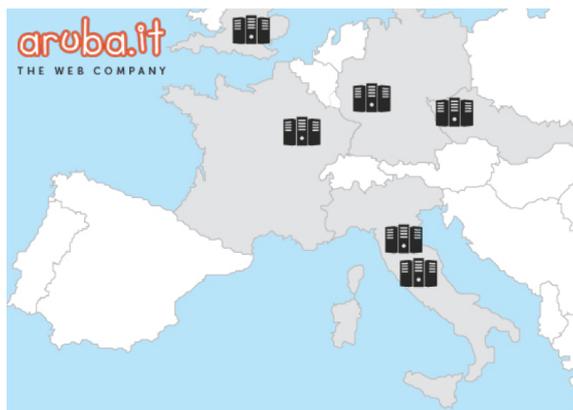
---

**Stefano Dal Pra***INFN-CNAF**E-mail: [stefano.dalpra@cnafe.infn.it](mailto:stefano.dalpra@cnafe.infn.it)***Vincenzo Ciaschini***INFN-CNAF**E-mail: [vincenzo.ciaschini@cnafe.infn.it](mailto:vincenzo.ciaschini@cnafe.infn.it)***Luca Dell'Agnello***INFN-CNAF**E-mail: [luca.dellagnello@cnafe.infn.it](mailto:luca.dellagnello@cnafe.infn.it)***Andrea Chierici***INFN-CNAF**E-mail: [luca.dellagnello@cnafe.infn.it](mailto:luca.dellagnello@cnafe.infn.it)***Donato De Girolamo***INFN-CNAF**E-mail: [donato.degirolamo@cnafe.infn.it](mailto:donato.degirolamo@cnafe.infn.it)***Vladimir Sapunenko***INFN-CNAF**E-mail: [vladimir.sapunenko@cnafe.infn.it](mailto:vladimir.sapunenko@cnafe.infn.it)***Tommaso Boccali***INFN-Pisa**E-mail: [tommaso.boccali@pi.infn.it](mailto:tommaso.boccali@pi.infn.it)***Alessandro Italiano***INFN-Bari**E-mail: [Alessandro.italiano@ba.infn.it](mailto:Alessandro.italiano@ba.infn.it)*

The Computing facility CNAF, in Bologna (Italy), is the biggest WLCG Computing Center in Italy, and serves all WLCG Experiments plus more than 20 non-WLCG Virtual Organizations and currently deploys more than 200 kHS06 of Computing Power and more than 20 PB of Disk and 40 PB of tape via a GPFS SAN. The Center has started a program to evaluate the possibility to extend its resources on external entities, either commercial or opportunistic or simply remote, in order to be prepared for future upgrades or temporary burst in the activity from experiments. The approach followed is meant to be completely transparent to users, with additional external resources directly added to the CNAF LSF batch system; several variants are possible, like the use of VPN tunnels in order to establish LSF communications between hosts, a multi-master LSF approach, or in the longer term the use of HTCCondor. Concerning the storage, the simplest approach is to use Xrootd fallback to CNAF storage, unfortunately viable only for some experiments; a more transparent approach involves the use of GPFS/AFM module in order to cache files directly on the remote facilities. In this paper we focus on the technical aspects of the integration, and assess the difficulties using different remote virtualisation technologies, as made available at different sites. A set of benchmarks is provided in order to allow for an evaluation of the solution for CPU and Data intensive workflows. The evaluation of Aruba as a resource provider for CNAF is under test, with limited available resources; a ramp up to a larger scale is being discussed. On a parallel path, this paper shows a similar attempt of extension using proprietary resources, at ReCaS-Bari; the chosen solution is simpler in the setup, but shares many commonalities.

## 1. Introduction

During spring 2015, the CMS Experiment[2] support and the Tier-1 team at CNAF took early contacts with Aruba[1], a major Italian host, web and cloud provider, at its site in Arezzo (Tuscany, Italy)<sup>1</sup>. The goal was to evaluate the possibility of remote computing resource usage from a commercial provider, in order to meet “elastically” the needs of High Energy Physics (HEP) computing, allowing for opportunistic resource utilisation as a transparent extension of CNAF during time periods when activity from ordinary Aruba users is low. This requires the resources to be given back almost instantaneously when such users need them; the chosen technical solution has been an artificial lowering of the virtual CPU clock in such cases, without the need of ever stopping completely opportunistic virtual machines; this capability is present under the VMware virtualisation system Aruba uses. This is a nice alternative to a more standard opportunistic solution, where low priority virtual machines are shut down when higher priority users claim the resources; in the proposed solution, no checkpointing or failure handling is needed on the user side. The price to pay is a more difficult to predict time to completion for tasks, which is not an issue in the case of High Throughput Computing used by typical High Energy Physics workflows. CMS has been used as initial test case, with the extension to all the LHC Experiments already undergoing.



**Figure 1:** Aruba Data Center network

## 2. The initial test case

In order to test the elastic setup in a real production environment, the CMS Experiment[2] has been selected. CMS is one of the biggest HEP users at CNAF, with a 2016 resource allocation (from Rebus[3]) of 48 KHS06 of CPU power, 4 PB of online disk storage and 12 PB on tape. CMS was selected as a CNAF test case mainly given the high level of local expertise; extension of the tests to the other LHC experiment is already undergoing, and should present no difficulties given the similarities in workflows and usage patterns.

The computational tasks CMS executes at CNAF are covering various aspects of its activities, from Monte Carlo simulation and Data reconstruction, to end-user analysis; the submission infras-

<sup>1</sup>The Data Center Network of Aruba are shown in Figure 1.

structure uses pilots from glideinWMS[4] with late binding, with the site not being able to predict the type of running job before it starts.

The typical CMS workflow needs some basic requirement in order to succeed and, from the perspective of the Tier-1 it proceeds as follow:

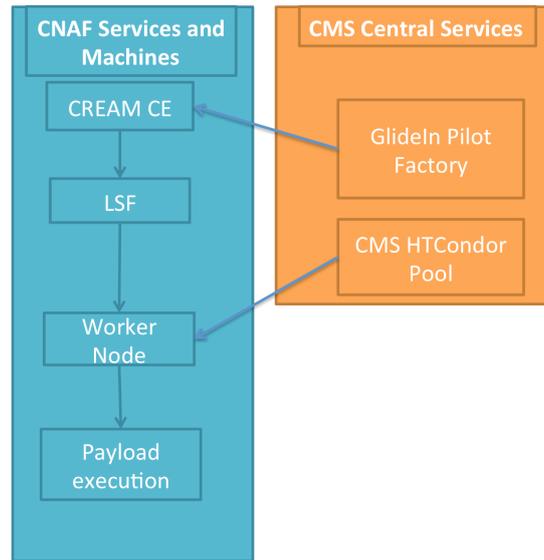
- GlideinWMS submits *Grid pilot jobs* to one or more Computing Elements at the site, which in turn submit them to the Local Resource Manager System (LRMS: the LSF Batch system[9], in our case).
- Each production CMS pilot submitted to INFN-T1 is currently a multi-core one, thus requiring to run on a Worker Node providing eight free slots and eight times more RAM than a single-core one (16 to 20 GB).
- The Batch System dispatches the pilot job to a suitable Worker Node (WN) and after the pre-execution check on the machine succeed, it starts and accesses the actual processing task as dispatched via HTCondor[5]; the executable and all the needed libraries are distributed via CVMFS[6] and available through a local SQUID proxy (this requires approximately up to 2GB per job);
- The batch system accesses the conditions database, also available through a SQUID proxy (approximately 1GB per job);
- The batch system accesses its input ROOT[7] files (file size varying from a few MB to many GB per job). The job looks for them on the local storage; those not locally available are retrieved by network through Xrootd[12].
- stage output files and logs to a SRM enabled storage supporting CMS. File size is generally smaller and spans from 50 to 500MB.

The interactions between the CMS Central Services and CNAF are sketched in Figure 2. Figure 3 shows the interactions needed by a CMS payload when running. ATLAS jobs follow a very similar pattern, and use many of the described services (SQUID, CVMFS, ...); LHCb and ALICE jobs present a number of peculiarities, still should not prevent the usage of the resources after some tweaking.

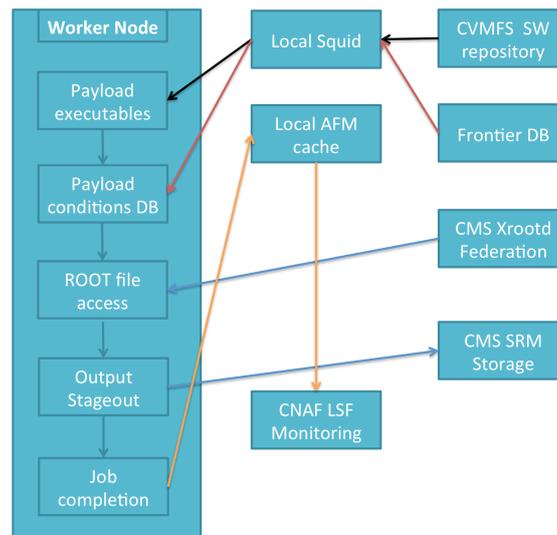
CMS differs from other experiments running at CNAF in that it exclusively submits multi-core pilots. The INFN-T1 provides resources for both the single and multi-core demand. This however may lead the centre to risk of underusing its resources: a fixed subset of WNs dedicated to multi-core only would be undersized during periods of peak activity, and almost unused in case of reduced submission rate.

To address this problem and reduce underutilisation of resources, a mechanism to implement dynamic partitioning for multi-core resources[14] have been implemented and deployed in production since August 2014 (see Fig. 4). Every CMS job runs on machines belonging to this partition.

When a standard CMS job is run at CNAF, the prerequisites are satisfied by on-site services; namely, SQUIDs, LSF and Grid Computing Elements (CE) are accessible through the LAN, and the input files are stored on the locally mounted GPFS[8] filesystem. However, when the job runs on a remote instance, it is preferable to have some services (SQUIDs for Condition Database access and



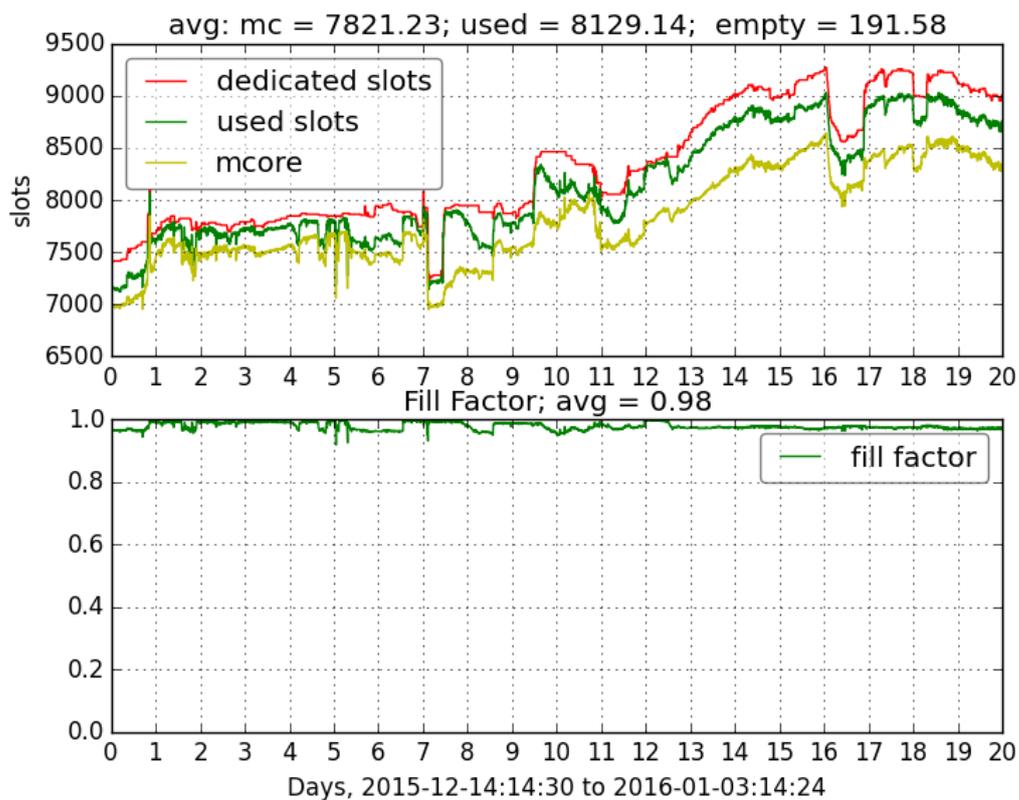
**Figure 2:** Interactions between CMS Central Services and CNAF Site when a pilot job arrives.



**Figure 3:** Interactions between the Worker Node and CMS services when an executable is started.

CVMFS) replicated and made locally available, in order to prevent excessive latencies and large geographical network I/O.

For what concerns the input ROOT files, LHC Experiments implement (with some differences which can be overlooked here) an internal mechanism which falls back to remote access using Xrootd Federations in case the local access fails. Using that, the input file is searched in all the CMS computing centres[11], and when found, a direct remote read is performed. This failover mechanism has been optimised for the elastic extensions: the Xrootd services at CNAF are queried before the other ones in the federation. Doing so, the data access is maintained as much as possible



**Figure 4:** Multi-core jobs run in a dynamic partition of dedicated nodes, whose size auto adjusts to satisfy the multi-core submission flow. The chart displays how the partition size grows and shrinks over a time window of 20 days, the number of running multi-core jobs, the sum of multi and single-core jobs, still running on recently dedicated nodes. The space between the red and the green line is the amount of unused slots in the partition, mainly due to nodes having less than eight free slots. The fill factor represents how fully the partition is being used.

within the monitored CNAF–Aruba link. External accesses are still possible via the federation, but are only used as a second level fall back.

### 3. Approach to opportunistic remote extension

The simplest extension of CNAF Computing Center requires remote resources to be seen as local, i.e. join the local LSF batch system and be seen on ar of local machines.

In doing this, there were several problems to overcome:

- The IPs of the remote VMs only have a private IP and are behind NAT and Firewall systems.
- LSF needs direct and reverse resolution (map hostname to IP and vice-versa) of its clients;
- All client nodes in a cluster need read access to a shared filesystem, which has to be provided over the general network, resulting in all the associated network latency problems. This

filesystem host LSF work area, and some experiment specific directories (those hosting site specific configuration for jobs).

In order to solve these problems, and to automate the process of deploying and configuring virtual worker nodes on remote machines a CNAF developed service, *dynfarm*, has been used.

In its configuration at Aruba, the virtual machines, upon startup, connect to the *dynfarm* server at CNAF. It authenticates connection requests coming from remote hosts and delivers the information needed in order to establish a VPN tunnel, used to communicate with a small subset of the local cluster in Bologna: the LSF batch system, the CREAM Computing Elements and the Authorization (Argus) service. This solves the first problem, since from the VM side the tunnel is an outbound connection that passes through the firewall, and the fact that the remote hosts get new addresses through the VPN makes them distinguishable even through a NAT.

A *full* VPN is not required since not all the remote hosts traffic is routed through the VPN server, but only the one required to have a functional worker node (traffic with the LSF server, CE and Argus). All other traffic still goes through its default outgoing route.

In order to solve the second problem (the fact that remote hosts need a hostname discoverable by the LSF master), the `$LSF_ENVDIR/hosts` file is used. This file works just like the well known `/etc/hosts` but is exclusive to LSF batch system. By configuring that, we can easily map in advance a superset of known private IPs and hostnames: done that, we can configure hostnames as regular WNs into other mandatory LSF configuration files.

Finally, in order to give each node access to the required shared filesystem, a read-only GPFS cache using the AFM plugin is deployed on a dedicate host at the Aruba computing centre, and all remote hosts are configured to access the area via that. The schematic of AFM is shown in Figure 5.

Figure 6 shows how *dynfarm* interacts with local and remote service when setting up the VPN.

#### 4. A more standard remote setup: the case of Bari-ReCaS

The CNAF Tier-1, at the same time as the test with Aruba, has also worked in the direction of the utilisation of remote resources at ReCaS[13], in BARI. The ReCaS project, financed by the MIUR<sup>2</sup> in the “PON Research and Competitività”<sup>3</sup>, has deployed a large resource base in a computing center at the Physics Department in Bari, very close to the INFN Tier-2. The utilisation pattern is simpler than the one tested with Aruba:

- The machines are not virtualised. Once the bare metal is correctly installed, connected to the network and powered, it is directly managed by CNAF staff, both via IPMI and ssh connection. The installed software is the same as for the machines in Bologna;
- The machines have public and private (used for ipmi) IPs belonging to CNAF, and all the outgoing traffic is routed to CNAF via a Level 3 circuit, with a reserved bandwidth of 20 Gbit/s. no host-level VPN needs to be set-up;

---

<sup>2</sup>Italian Ministry for Education, University and Research

<sup>3</sup>Ricerca e Competitività 2007–2013 – Azione I – Interventi di rafforzamento strutturale PONa3\_00052, Avviso 254/Ric

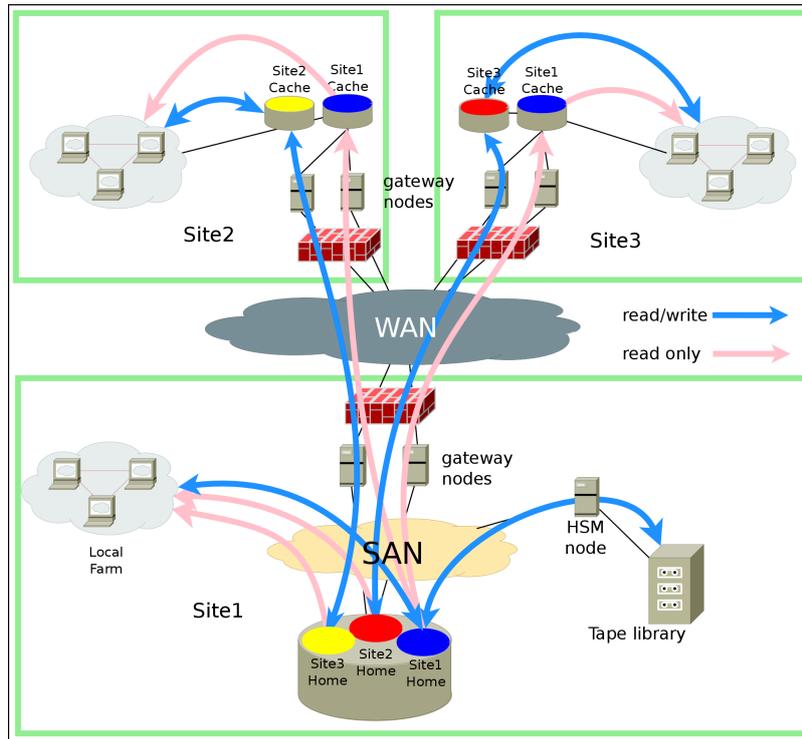


Figure 5: GPFS AFM setup.

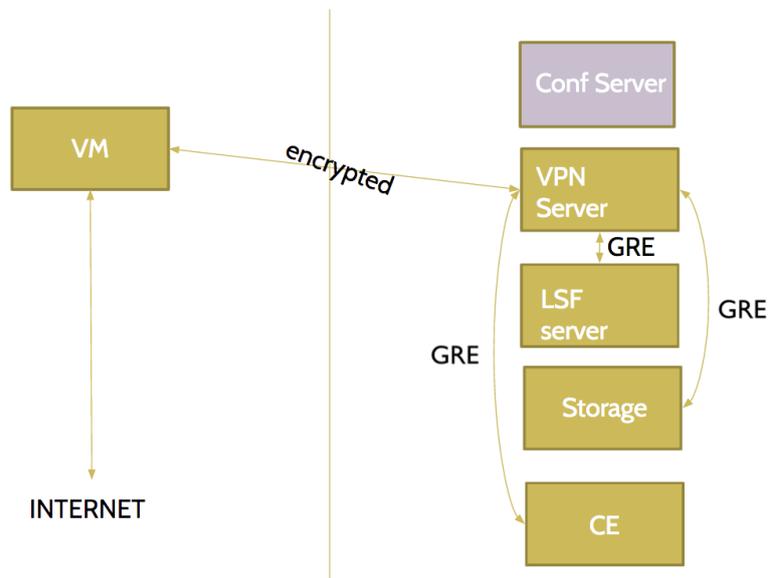


Figure 6: The dynfarm communication flow.

- local squids are present on-site and managed by CNAF staff;
- online read access to GPFS disk available through AFM replica (Figure 5). This has been set up for three LHC Experiments: ATLAS, CMS, LHCb with an overall disk size of 330TB. ALICE does not need it since it directly accesses via Xrootd;
- The cluster shared filesystem (hosting configurations, LSF work area, ...) is also replicated via AFM and locally made available for read-only access to the WNs via NFS. Even though this is a small filesystem with low traffic, care has to be taken to keep its access latencies at a minimum: heavy data access should not interfere with the cluster management traffic, which depends on the readiness of the shared filesystem.

From the management point of view, these resources appear as local machines, inside the LAN and seamlessly connected to LSF. There are no problems with firewalls, nor any requirements for host-level VPNs.

Currently the “remote” cluster has roughly 2000 cores, and accepts jobs only by LHC Experiments, since only these have an Xrootd fallback mechanism in place.

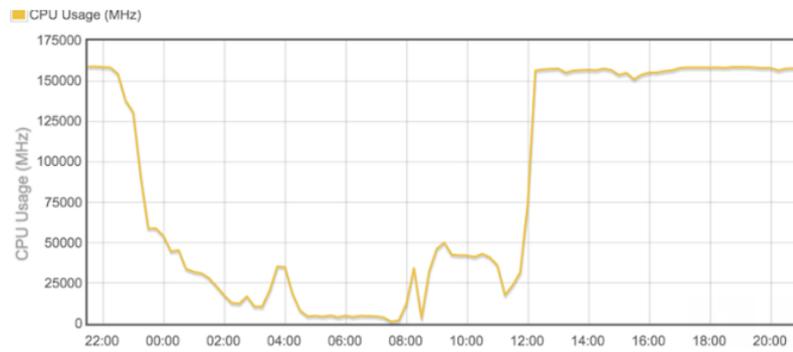
## 5. Results

Results presented here for the Aruba testbed are the results of a few weeks of running.

<i>mese</i>	<i>queue</i>	<i>site</i>	<i>njobs</i>	<i>avg_eff</i>	<i>max_eff</i>	<i>avg_wct</i>	<i>avg_cpt</i>
2015-10	cms_mc	AR	2793	0.609	0.912	205.823	134.606
2015-10	cms_mc	T1	34342	0.713	0.926	120.886	96.147
2015-11	cms_mc	AR	232	0.752	0.871	320.182	241.498
2015-11	cms_mc	T1	22151	0.805	0.886	177.937	147.926
2016-01	cms_mc	AR	459	0.212	0.656	50.865	10.125
2016-01	cms_mc	T1	18577	0.774	0.852	122.470	96.862
2016-02	cms_mc	AR	2698	0.438	0.713	163.304	67.627
2016-02	cms_mc	T1	16367	0.793	0.868	146.713	117.404

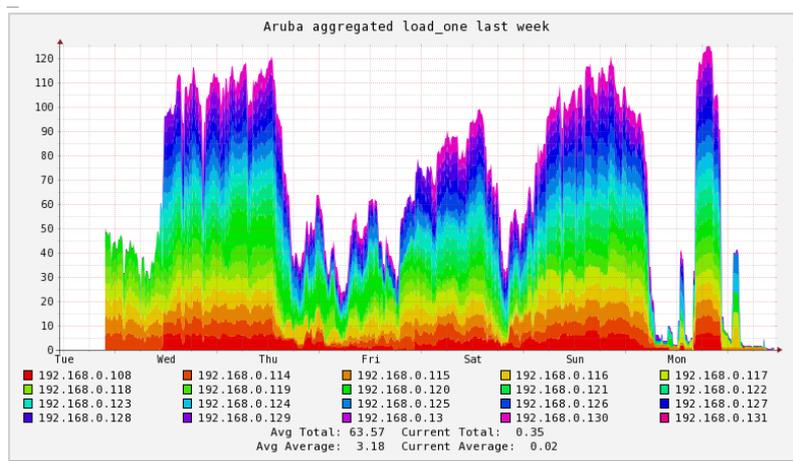
**Table 1:** Comparative efficiency and duration of CMS jobs per site, by month. The average efficiency tends to be higher with longer jobs

The testbed Aruba, prepared for INFN, is in the form of a VMware Virtual Data Center, where a given amount of RAM, Disk and CPU is offered. Its partitioning within different hosts is completely under CNAF control; in particular, the testbed allowed us to use 160 GHz of CPU (Intel 2697-v3). Assuming 2 GHz cores, this enables in principle 80 active cores; in reality, more can be obtained, since the cap is not on allocated cores, but on used CPU. So, a larger WN base can be installed, and if at some point the total CPU utilization would surpass the cap, an automatic clock slowdown would happen, with the same mechanism used when opportunistic usage must be reduced. Figure 7 shows a situation where cap is reached, and total CPU utilization is kept under 160 GHz.



**Figure 7:** CPU capping at 160 GHz, when high load is present.

Figure 8 shows a typical 1-week load result, where each different color represents a virtual host. Figures 9 and 10 show the total outgoing and ingoing network to the Worker Nodes. Finally, Table 2 displays comparative information about efficiency (defined as  $\sum \frac{CPUtime}{NUMcores \times RUNtime}$ ) and duration of finished jobs at the different *sub-sites* over a time period of a month.

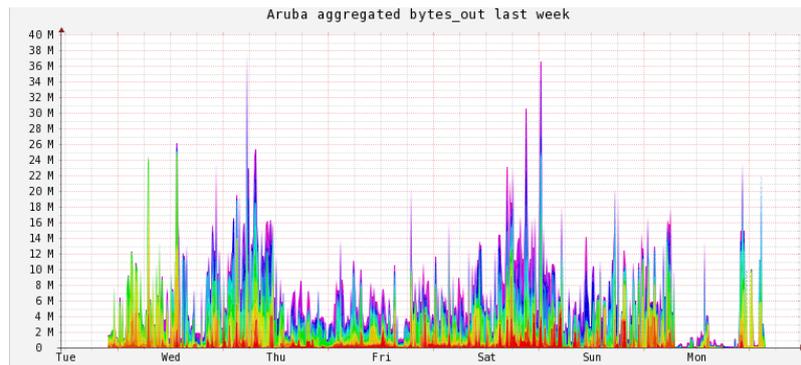


**Figure 8:** Load, one week. Each color represents a single Virtual Worker Node.

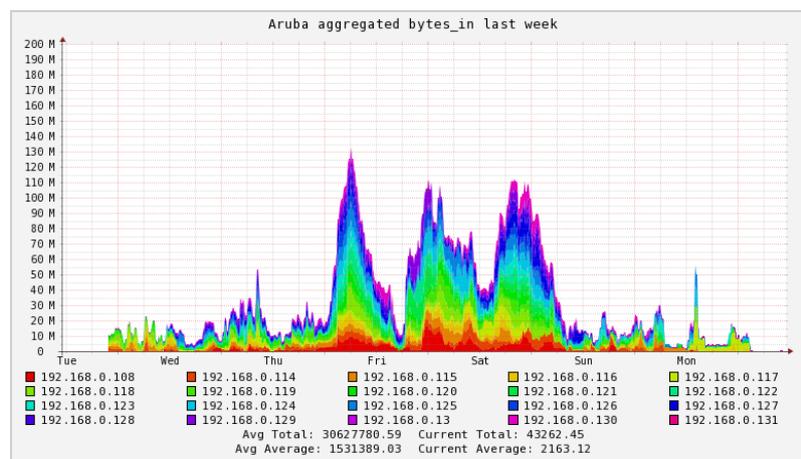
## 6. Problems and deployed solutions

Even if the initial tests have been successful and promising, a number of minor issues are still present and need improvement. Some of them are being addressed, other have known solutions.

- Heterogeneity of management and monitoring tools: deployment of VM is performed through the VMware dashboard, and monitoring facilities are provided by ganglia[10]. The connection with the LSF cluster is maintained through the dynfarm system. This introduces a complexity layer on the overall administration of the whole cluster. In order to limit its impact, a few simple and well known management tool, such as `wassh`, are being made available in the remote cluster too.



**Figure 9:** Network Output from Worker Nodes.



**Figure 10:** Network Input from Worker Nodes.

- Bulk VM instantiation from the VMware dashboard: VMs are currently started one by one. This has been good for testing purposes; scripting a tool for deployment through the VMware API is however mandatory in case of actual usage in production. However, since VMs are eternal (they are never shut down, just slowed down), starting and stopping machines is not a frequent operation.
- Multiple identities for the same VM instance: once created and started through the VMware dashboard, there is no way to map the hostname and IP in the Aruba domain to the hostname and IP in the LSF domain. Because of this, should a remote node become unreachable to LSF, there is no easy way to know which is the corresponding hostname or IP in the Aruba cloud.
- Volatile LSF hostnames and IPs: upon reboot a VM rejoins the LSF cluster with a different hostname and IP. Hostname and IP resilience would be desirable, however there is currently no known way to configure the DHCP component in openvpn in order to achieve this.
- Late binding of pilot jobs: opportunistic resources are bound to a number of constraints and some categories of jobs show lower efficiency, typically those needing little CPUtime

<i>month</i>	<i>queue</i>	<i>site</i>	<i>njobs</i>	<i>avg_eff</i>	<i>max_eff</i>	<i>avg_wct</i>	<i>avg_cpt</i>
2016-02	alice	BA	23153	0.725	0.966	16.071	11.765
2016-02	atlas	BA	451	0.942	0.999	2.833	2.721
2016-02	cms_mc	BA	287	0.516	0.865	388.130	194.689
2016-02	lhcb	BA	11269	0.916	0.967	15.319	13.082
2016-02	mcore	BA	23441	0.674	0.851	155.448	36.344
2016-02	cms_mc	AR	2698	0.438	0.713	163.304	67.627
2016-02	alice	T1	184292	0.697	0.958	15.936	11.284
2016-02	atlas	T1	1182474	0.823	0.999	2.219	1.936
2016-02	cms_mc	T1	16367	0.793	0.868	146.713	117.404
2016-02	lhcb	T1	78769	0.970	0.989	18.230	17.664
2016-02	mcore	T1	13857	0.649	0.972	22.071	16.776

**Table 2:** Comparative information about efficiency and duration at the different sub-clusters over one month.

to process large datafiles. This kind of jobs may have negative impact with scalability of CPU resources, if the available network bandwidth does not scale accordingly. Late binding prevents the batch system from dispatching jobs to the more suitable sub-cluster depending on their activity. It is however possible to configure a specialised entry point queue on LSF and CEs, dedicated to a given sub-cluster, and the pilot factory configured to submit the appropriate kind of jobs only there. This is expected to allow for a more efficient usage of both Aruba and Bari resources.

## 7. Conclusions

The studies presented in this paper, even if at a preliminary level, show that an extension of a large WLCG Data Center like CNAF with remote resources is possible, either via proprietary resources (like in Bari, where computing for HEP has local expertise), or via commercial providers on a standard Cloud infrastructure. In the latter (and more interesting) case, we have proved that a fully working setup, completely transparent to the rest of the CNAF infrastructure, can be obtained via a non intrusive VPN setup, automatised via the dynfarm tool. The setup is easily configurable, and offers a solution for fast-turnaround resource utilisation, even on the short time scale.

## References

- [1] <https://www.aruba.it/en/home.aspx>
- [2] S. Chatrchyan et al. CMS Collaboration 2008 “The CMS experiment at the CERN LHC” J. Inst. 3 S08004.
- [3] <https://wlcg-rebus.cern.ch/apps/pledges/summary/>
- [4] Sfiligoi I, Bradley D C, Holzman B, Mhashilkar P, Padhi S and Würthwein F 2010 “The pilot way to grid resources using glideinwms” WRI World Congress 2 428-432.
- [5] Thain D, Tannenbaum T and Livny M 2004 “Distributed computing in practice: the condor experience Concurrency” Pract. Exper. 17 323-356.

- [6] Blomer J, Buncic P, Charalampidis I, Harutyunyan A, Larsen D, and Meusel R 2012 “Status and future perspectives of CernVM-FS” J. Phys.: Conf. Ser. 396 052013.
- [7] Brun R and Rademakers F 1997 “ROOT - An Object Oriented Data Analysis Framework” Nucl. Instr. Meth. A389 pp 81-86.
- [8] [https://www.ibm.com/support/knowledgecenter/SSFKCN/gpfs\\_welcome.html](https://www.ibm.com/support/knowledgecenter/SSFKCN/gpfs_welcome.html)
- [9] <http://www-03.ibm.com/systems/platformcomputing/products/lfs/>
- [10] <http://ganglia.info/>
- [11] Kenneth Bloom and the Cms Collaboration “CMS Use of a Data Federation” Journal of Physics: Conference Series, Volume 513, Track 4.
- [12] Dorigo A, Elmer P, Furano F and Hanushevsky A 2005 “Xrootd - A highly scalable architecture for data access” WSEAS Transactions on Computers (2005).
- [13] <http://www.recas-bari.it/index.php/en/>
- [14] S. Dal Pra “Efficient provisioning for multi-core applications with LSF” 2015 J. Phys.: Conf. Ser. 664 052008 doi:10.1088/1742-6596/664/5/052008.