

# Enabling INFN–T1 to support heterogeneous computing architectures

Stefano Dal Pra<sup>1,\*</sup>, Daniele Spiga<sup>2,\*\*</sup>, Tommaso Boccali<sup>4</sup>, Andrea Chierici<sup>1</sup>, Lucia Morganti<sup>1</sup>, Vladimir Sapunenko<sup>1</sup>, Daniele Cesini<sup>1</sup>, Lorenzo Rinaldi<sup>3</sup>, Daniele Gregori<sup>5</sup>, and Marco Cicala<sup>5</sup>

<sup>1</sup>INFN–CNAF, v.le B. Pichat 6/2 - 40127 Bologna, IT

<sup>2</sup>INFN Sezione di Perugia, Via Alessandro Pascoli, 06123 Perugia, IT

<sup>3</sup>Bologna University and INFN Sezione di Bologna, v.le B. Pichat 6/2 - 40127 Bologna, IT

<sup>4</sup>INFN Sezione di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, IT

<sup>5</sup>E4 Computer Engineering SPA, Viale Martiri della Libertà 66, Scandiano, IT

**Abstract.** The INFN–CNAF Tier-1 located in Bologna (Italy) is a center of the WLCG e-Infrastructure providing computing power to the four major LHC collaborations and also supports the computing needs of about fifty more groups - also from non HEP research domains. The CNAF Tier1 center has been historically very active putting effort in the integration of computing resources, proposing and prototyping solutions both for extension through Cloud resources, public and private, and with remotely owned sites, as well as developing an integrated HTC+HPC system with the PRACE CINECA supercomputer center located 8Km far from the CNAF Tier-1 located in Bologna. In order to meet the requirements for the new Tecnopolo center, where the CNAF Tier-1 will be hosted, the resource integration activities keep progressing. In particular, this contribution will detail the challenges that have recently been addressed, providing opportunistic access to non standard CPU architectures, such as PowerPC and hardware accelerators (GPUs). We explain the approach adopted to both transparently provision x86\_64, ppc64le and NVIDIA V100 GPUs from the Marconi 100 HPC cluster managed by CINECA and to access data from the Tier1 storage system at CNAF. The solution adopted is general enough to enable seamless integration of other computing architectures at the same time from different providers, such as ARM CPUs from the TEXTAROSSA project, and we report about the integration of these within the computing model of the CMS experiment. Finally we will discuss the results of the early experience.

## 1 Introduction

The INFN-CNAF Tier-1 computing center [1], plays an important role in supporting the High Energy Physics (HEP) community. With the continuous growth in the scale and complexity of scientific computing, it has been proactive in exploring ways to integrate diverse computing resources to meet the increasing demand. Over the years, significant progress has been made at integrating various computing resources with the center's infrastructure. These

---

\*e-mail: [dalpra@infn.it](mailto:dalpra@infn.it)

\*\*e-mail: [daniele.spiga@pg.infn.it](mailto:daniele.spiga@pg.infn.it)

resources encompass a wide spectrum, ranging from opportunistic resources deployed on Cloud Instances [2], nodes available on high-performance computing (HPC) systems [3], to extensions within data centers [4]. These works have enabled access for the HEP community to new sets of computing resources, while keeping unaltered the standard WLCG [5] Grid paradigm adopted to access them.

While the HEP computing infrastructure predominantly relies on the x86\_64 architecture, recent collaborations between the WLCG community and HPC centers have introduced computing power via PowerPC architecture-based machines, such as the M100 supercomputer[6] hosted by CINECA[7] and also providing NVIDIA V100 GPUs. In the same period, several providers have begun offering access to novel computing architectures, such as low power ARM CPUs from the TEXTAROSSA [8] European Project (provided by the E4Company), or even traditional x86\_64 CPUs offered by external providers through cloud or traditional “local” access, according to specific agreements or grants with INFN, such as the one with ASI[9] (ASI-INFN Agreement No. 2021-43-HH.0.). All these developments have motivated an exploration into methods for seamlessly incorporating these resources, making them accessible through standard job submission Grid interfaces and local queue management services, such as HTCondor-CEs and HTCondor batch system.

This endeavor requires a two-fold effort. On one front, experiments must validate their software to ensure they produce reliable results with an alternative computing architecture. On the other front, Grid resource providers like the INFN-CNAF Tier-1 must devise strategies to enable easy access to these heterogeneous resources for the community.

### 1.1 The CNAF Tier-1 Computing Facility

CNAF Tier-1 delivers a computing power of approximately 715,000 HEPSCORE [10] from 59,000 CPU cores on 1010 Compute Nodes. These resources are used by 23 WLCG communities, including those involved in HEP, astroparticle physics, and astrophysics, as well as around 30 local research groups. Grid access to the computing resources is handled by 6 instances of HTCondor-CE 5.1.6, on top of a HTCondor 9.0.17 batch system, serving both Grid and local users.

### 1.2 Moving to Tecnopolo

Transition to the Tecnopolo environment is planned [11], which is expected to further enhance resource utilization. The Tecnopolo environment will enable opportunistic usage of computing resources from the Leonardo High-Performance Computing (HPC) facility [12] through an early agreement. More specifically, this includes x86\_64 Compute Nodes from Leonardo HPC, equipped with Intel Xeon Platinum or Sapphire Rapids CPUs, 512 GB of RAM, and 4 NVIDIA A100 GPUs [13], all managed by the Slurm batch system managed by CINECA [7].

### 1.3 Objective: Seamless Resource Integration

With the primary goal to achieve seamless integration of both opportunistic and pledged computing resources, our initial investigations started with the Marconi 100 [6] machines; these are powered by PPC Power9 CPUs featuring 192 cores per node with 256 GB of RAM, and 4 NVIDIA V100 GPUs as shown in Fig. 1, also managed by the Slurm batch system. The different CPU architecture offers a further opportunity to exploit solutions to provide support for heterogeneous architectures.

In the following sections of this paper, we delve deeper into the technical details and operational aspects. We initially focus with the M100 use case.

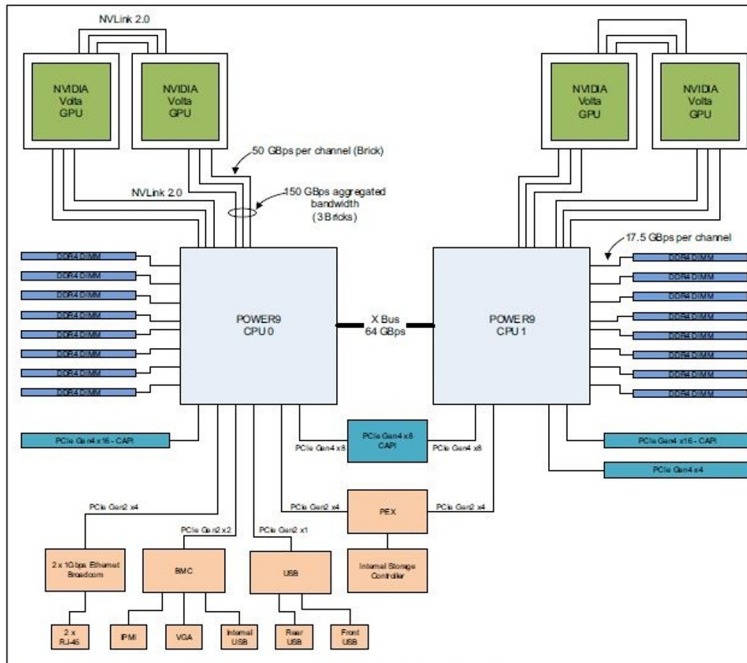


Figure 2-5 The Power AC922 server model GTH logical system diagram

Figure 1: Architectural view of a M100 Compute Node.

## 2 M100: From Slurm Compute Node to CNAF Execution Point

Transparent integration of compute resources from Marconi 100 (M100) within the CNAF Tier-1 infrastructure can be achieved by implementing two main ideas:

1. **HTCondor STARTD as a Slurm Job:** the M100 Compute Nodes are managed by Slurm, and we are entitled to submit jobs to a dedicated Slurm queue from a Login Node, with no particular privileges. We want this job, once started, to instantiate an HTCondor STARTD which must be able to join the HTCondor pool at CNAF and then run suitable payloads from one of the CNAF HTCondor Access Points
2. **Detecting Pending Jobs for ppcml64:** When pending jobs specifically tailored for the ppcml64 architecture are detected at CNAF, a number of the above described Slurm Jobs are to be submitted, thus triggering the instantiation of the needed resources.

### 2.1 STARTD as a Slurm Job

To obtain a HTCondor STARTD (i.e., Execution points) we submit to Slurm a job that instantiates a Singularity container at startup. This, in turn, instantiates an HTCondor STARTD daemon. This HTCondor version is of course binary compatible with the PPC architecture. Moreover, the STARTD is equipped with IDTOKEN credentials and configured to join the HTCondor Central Manager at CNAF, through the CCB [14] component.

### 3 CNAF: Steering Jobs to M100

We consider now the problem to efficiently route Grid jobs submitted to the HTCondor-CEs at CNAF toward M100 resources. The main problem is to identify when a job is intended for a specific set of resources. To that extent, a number of possible solutions are theoretically possible:

1. **Set a Custom Attribute in the Submit File:** In this approach, a custom attribute is defined in the job's submit file to specify the destination as M100. For example, a custom attribute such as `+WantRoute = "cms_m100"` can be included. Subsequently, a JobRouter entry can be configured to match by that specific `+WantRoute` value and inject the requirement `Arch == "ppc64le"` to the routed job. This method provides explicit control over job routing based on architecture.
2. **Set an Agreed Claim in the Access Token:** Alternatively, an agreed-upon claim can be embedded within the access token used during job submission. This claim can be inspected by the JobRouter as a Classad Attribute. Various types of tokens, such as SCITOKEN, IAM Token, or EGI Check-in Token, can be used for this purpose. Recent versions of HTCondor (10.4.1 at the time of this work) offer enhanced customization possibilities for processing these claims.
3. **Late binding, as usual.** While the previous methods require that the submitting user declare at submit time the preferred resource set, this method follows the usual way of the "pilot mode": at start time in the Execution Point, a script detects the specifics of its running environment and then connects to its "payload factory", waiting for a suitable payload to run. If there are none, the pilot script gives up after some time (usually on the order of 20 minutes).

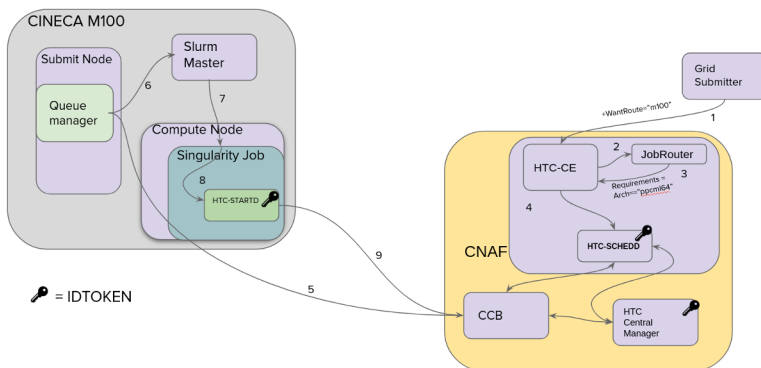


Figure 2: external PowerPC Computing Nodes from the CINECA M100 cluster join the CNAF HTCondor pool. Other groups of resources can also join the main CNAF pool in similar way.

The latter method would be preferable from the submitter point of view, since this is already the standard adopted model, with no changes. However, a few drawbacks come into play with a new scenario where opportunistic and heterogeneous resource sets become available at the same time. For example, allowing a random pilot to start on a precious

resource (such as a GPU) when there are no payloads to run, is inefficient. It is worth noting that these methods can coexist, because each user community can potentially agree upon a preferred method with the site administrators. For the testing performed in the present work, we have adopted the first method and a simplified version of the second (check for specific submitter token subject, to ensure that no unintended job from regular users is accepted).

The overall implementation is represented in Figure 2, and it operates according to the following points:

1. A job submitted to a HTCondor-CE defines a custom `+WantRoute` attribute within its submit file.
2. A JobRouter rule matches the `WantRoute` custom attribute and apply proper job transform rules:
- 3,4. The JobRouter rule adds the `Arch=="ppcml64"` requirement to the routed job; then, it is queued to the SCHEDD.
5. In the Slurm Login Node at CINECA, the Queue Manager script regularly checks the CNAF queues for jobs specifically requiring the `ppcml64` architecture.
6. When the Queue Manager detects such `ppcml64` jobs at CNAF, it submits a number of Slurm jobs, depending on the current number of already pending and running Slurm jobs
7. At the start of the Slurm job, a Singularity container is launched.
8. Within the Singularity container, an HTCondor STARTD is activated. This is equipped with IDTOKEN credentials and an appropriate `StartJobs` policy to only accept jobs with proper requirements.
9. The HTCondor STARTD authenticates itself with the CNAF HTCondor pool through the Connection Control Broker (CCB), establishing a secure connection for job execution.

An example submit file for testing is provided in Listing 1 and two example JobRouter rules in Listing 2, one for access to M100 resources by CMS and one for access by ATLAS to a local cluster [15] of the Physics and Astronomy department of the Bologna University. For the ATLAS use case, we run a ROOT-based analysis code, independent of the ATLAS Athena framework.

```
[sdalpra@ui-htc ~]$ cat ce_m100_p9.sub
universe = vanilla
use_scitokens = true
+Owner = undefined
+WantRoute = "cms_m100"
executable = p308/htcp308.p9
output = htcp308_$(ClusterId).$(ProcId).out
error = htcp308_$(ClusterId).$(ProcId).err
log = htcp308_$(ClusterId).$(ProcId).log
arguments = "0_0_1_2001"
queue 1
```

Listing 1: Example submit file for M100 resources

```

JOB_ROUTER_ROUTE_cms_m100 @=jrt
REQUIREMENTS (WantRoute =?= "cms_m100") &&\
  (AuthTokenIssuer =?= "https://cms-auth.web.cern.ch/" &&\
    AuthTokenSubject =?= "78f275d5-bb1a-4b2d-9956-f82316a8482e")
UNIVERSE VANILLA
SET Requirements (TARGET.Arch =?= "ppc64le")
@jrt

JOB_ROUTER_ROUTE_atlas_unibo @=jrt
REQUIREMENTS (WantRoute =?= "atlas_unibo") &&\
  (AuthTokenIssuer =?= "https://atlas-auth.web.cern.ch/" &&\
    AuthTokenSubject =?= "ccff569b-bda5-45ac-91bb-44c8e198a385")
UNIVERSE VANILLA
SET Requirements (TARGET.Arch =?= "x86_64")
@jrt

```

Listing 2: Example JobRouter rules for M100 and UniBO cluster

## 4 Autoscaling

The primary goal of autoscaling is to only instantiate M100 resources when there are suitable payloads queued at CNAF specifically requiring these kinds of resources, and have resources dynamically allocated based on actual demand. It is important to prevent having active PPC nodes running without payloads, since the time spent on PPC nodes is accounted to the Slurm queue dedicated to CNAF.

To achieve the above goals, a script is deployed on a Marconi 100 (M100) login node, which monitors the state of CNAF queues and the status of Slurm jobs at regular times, and takes action if needed to allocate or deallocate M100 resources, according to the following logic:

- If there are pending CNAF jobs, the script submits additional Slurm whole node jobs to meet the demand.
- If there are no pending CNAF jobs, the script terminates idle Slurm jobs. These correspond to the hosted HTCondor STARTDs having no active jobs to execute. This ensures that M100 resources are allocated only when there is a genuine need.

It's important to note that this approach aligns with a workload-driven model, ensuring that resources are allocated based on the actual demand from CNAF. However, in a "late binding" scenario (i.e., the submitted job never declares its needs because these are only defined after the job actually starts in an Execution Point) this autoscaling logic fails because there would always be some pilot job willing to run on a M100 machine, even when no payloads are queued in the payload factories managed by the submitting experiment for those machines.

## 5 ARM@CNAF

The integration of ARM-based computing resources at CNAF represents a significant milestone, enabled by the collaboration with E4[16], a company with historical ties to both CNAF and CERN computing. In this section, we delve into the specifics of the ARM architecture deployment at CNAF and the notable outcomes achieved.

## 5.1 Hardware Configuration

E4 leveraged hardware resources from the TEXTAROSSA EU project, which is part of the EuroHPC initiative [17], to introduce ARM-based computing nodes at CNAF. The hardware included two Ampere Altra MAX nodes, each equipped with the following specifications:

- Dual 128 cores Neoverse N1, 3.0 GHz (M128-30);
- 256 cores per node;
- up to 1 TB RAM.

The nodes available for early testing were equipped with fast local storage, although their wide area network (WAN) connectivity was limited.

## 5.2 Integration and Usage

Leveraging the experience gained from working with CINECA systems, INFN team seamlessly integrated these ARM-based nodes into the CNAF computing environment, using the support from the E4 sysadmins. The integration in CNAF's HTCondor queues was achieved by adding a specific JobRouter rule, similar to those in Listing 2, forging a proper `StartJobs` expression for the `STARTD` and issuing a dedicated `IDTOKEN` for this specific set of resources.

The ARM nodes were used for HEPSCORE calculations, achieving a benchmark of 3884, indicating their suitability for high-performance scientific computing tasks; to-date, the result is the third-highest ever registered in the HEPSCORE DB ([18]). The ARM-based machines were successfully incorporated into the CMS production system, accepting standard production jobs. Standard validation procedures were initiated, although limitations in networking resources restricted their use to pileup level 0. Nevertheless, results were remarkable, with less than 1% error rates observed. It's important to note that while substantial progress was made in the validation process, comprehensive physics validation, especially for scenarios involving pileup, remains ongoing. These results pave the way for further exploration of ARM-based architectures in high-energy physics.

The introduction of ARM-based resources at CNAF showcases the adaptability of the facility to diverse hardware architectures and the potential for enhancing computational capabilities. Despite networking limitations, the initial outcomes have been promising, setting the stage for continued exploration and utilization of ARM-based computing within the realm of high-energy physics. In order to study power utilization, CNAF performed measurements using meter-equipped PDUs at E4; these data were shared with other teams collecting similar data ([19]).

## 6 Validation Process

The validation process for the integration of ARM-based computing resources at CNAF has been a pivotal step in ensuring the seamless operation and reliability of these resources within the CMS computing infrastructure. This section summarizes the key aspects of the validation process, which has demonstrated the successful integration of ARM resources into the local computing capacity at CNAF. This activity proceeded independently with respect to the validation of the physics results, which is beyond the scope of this work.

## 6.1 Technical Integration Verification

The initial phase of the validation process focused on verifying the technical integration of ARM resources. This verification was crucial to ensure that CMS could recognize and utilize ARM-based resources as part of the CNAF local computing capacity. Notably, this integration was successful, and CMS now perceives ARM resources as an integral part of its computational infrastructure. At the same time, CNAF has decided to procure a small ARM cluster, to allow for tests with high bandwidth access to the storage system.

## 6.2 Data Generation Campaigns

Two comprehensive campaigns of data generation were executed to validate the performance and reliability of ARM-based resources. These campaigns were conducted under the following conditions:

1. **Run3 Scenario:** The first campaign was conducted under the Run3 scenario, simulating the operational conditions expected during the upcoming Run3 phase of experiments. This scenario allowed for testing the ARM resources under conditions that closely mirror the actual production environment.
2. **Phase2 Conditions:** The second campaign was executed under Phase2 conditions, which are representative of the expected computing conditions during the Phase2 upgrade of CMS experiments. This phase is characterized by increased data volumes and computational demands.

The outcomes of these data generation campaigns have been highly promising. Data samples generated during these campaigns have been seamlessly injected into the CMS computing infrastructure. These results signify that the ARM resources are not only technically integrated, but also fully operational and ready to contribute to the computational needs of CMS.

## 7 Conclusions

In this paper, a solution to enable seamless opportunistic Grid access to heterogeneous computing resources provisioned by third party centers has been presented. This method has been tested at CNAF and allowed us to add PowerPC machines hosted at CINECA as Compute Nodes at the INFN-T1 HTCondor pool. Acquisition of such external resources is driven by demand from the Grid jobs submitted at CNAF, and they are automatically released when unneeded.

The model has been adapted to leverage integration with other external providers, with different Local Resource Managers (such as OpenStack or Slurm) and computing architectures, including GPUs. CMS, ATLAS, and other smaller experiments were successful at executing Grid jobs submitted to the HTCondor-CEs at CNAF on these external machines. CMS has successfully validated the integration of its computing workflow to use PowerPC and ARM machines as opportunistic resources provided by an ordinary WLCG site, such as INFN-T1.

At the time of this writing, ARM based machine have been deployed at CNAF and have been made available to user communities. We expect more ARM machines to come in the near future.

The availability of heterogeneous and/or opportunistic computing resources is expected to grow in the future, and results of this work pave the way to include them for usage within the well established Grid paradigm without major changes to the computing model of the HEP communities.



## 8 Acknowledgements

This work is supported by ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU.

## References

- [1] L. Dell’Agnello, T. Boccali, D. Cesini, L. Chiarelli, A. Chierici, S. Dal Pra, D. De Girolamo, A. Falabella, E. Fattibene, G. Maron et al., *INFN Tier-1: a distributed site*, in *EPJ Web of Conferences* (EDP Sciences, 2019), Vol. 214, p. 08002
- [2] S. Dal Pra, V. Ciaschini, L. Dell’Agnello, A. Chierici, D. De Girolamo, V. Sapunenko, T. Boccali, A. Italiano, *Elastic CNAF Data Center extension via opportunistic resources*, in *International Symposium on Grids and Clouds* (2016), Vol. 13
- [3] T. Boccali, V. Sapunenko, D. Cesini, A. Valassi, S. Dal Pra, F. Stagni, D. Spiga, C. Bozzi, S. Zani, A. Doria et al., PoS p. 003 (2021)
- [4] T. Boccali, S. Dal Pra, D. Spiga, D. Ciangottini, S. Zani, C. Bozzi, A. De Salvo, A. Valassi, F. Noferini, L. dell’Agnello et al., *Extension of the INFN Tier-1 on a HPC system*, in *EPJ Web of Conferences* (EDP Sciences, 2020), Vol. 245, p. 09009
- [5] J. Shiers, *Computer physics communications* **177**, 219 (2007)
- [6] <https://www.hpc.cineca.it/hardware/marconi100>
- [7] <https://www.cineca.it/en>
- [8] <https://textarossa.eu/>
- [9] <https://www.asi.it/en/>
- [10] <https://gitlab.cern.ch/hep-benchmarks/hep-score>
- [11] <https://indico.jlab.org/event/459/contributions/11643/>
- [12] (2022), <https://leonardo-supercomputer.cineca.eu>
- [13] (2022), <https://wiki.u-gov.it/confluence/display/SCAIUS/UG3.2%3A+LEONARDO+UserGuide>
- [14] [https://htcondor.readthedocs.io/en/v10\\_0/admin-manual/networking.html#networking-includes-sections-on-port-usage-and-ccb](https://htcondor.readthedocs.io/en/v10_0/admin-manual/networking.html#networking-includes-sections-on-port-usage-and-ccb)
- [15] <https://site.unibo.it/openphysicshub/en>
- [16] <https://www.e4company.com/>
- [17] <https://eurohpc-ju.europa.eu>
- [18] [https://w3.hepik.org/benchmarking/scores\\_HS23.html](https://w3.hepik.org/benchmarking/scores_HS23.html)
- [19] E. Simili, *Energy Efficiency in H.E.P. (ARM vs. x86 and beyond)*, in *CHEP2023, in course of publication* (2023)