Supporting multiple hardware architectures at CMS: the integration and validation of POWER9

Tommaso Boccali[1], Saqib Haleem[2], Dirk Hufnagel[3], Alan Malta Rodrigues[4], Jordan Martins[5], Marco Mascheroni[6], Hasan Ozturk[7], Antonio Pérez-Calero Yzquierdo[8,9], Kirill Skovpen[10], Daniele Spiga[11], Christoph Wissing[12]

[1] INFN Sezione di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy
[2] National Centre for Physics (PK)
[3] Fermi National Accelerator Laboratory (USA)
[4] University of Notre Dame, USA
[5] Universidade do Estado do Rio de Janeiro (BR)
[6] University of California San Diego, La Jolla, CA, USA
[7] CERN (Switzerland)
[8] Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), Madrid, Spain
[9] Port d'Informació Cientifica (PIC), Barcelona, Spain
[10] Ghent University (BE)
[11] INFN Sezione di Perugia, Via A. Pascoli 23c, 06123 Perugia, Italy
[12] Deutsches Elektronen-Synchrotron, Notkestr. 85, 22603 Hamburg, Germany

E-mail: daniele.spiga@pg.infn.it christoph.wissing@desy.de

Abstract. Computing resources in the Worldwide LHC Computing Grid (WLCG) have been based entirely on the x86 architecture for more than two decades. In the near future, however, heterogeneous non-x86 resources, such as ARM, POWER and Risc-V, will become a substantial fraction of the resources that will be provided to the LHC experiments, due to their presence in existing and planned world-class HPC installations. The CMS experiment, one of the four large detectors at the LHC, has started to prepare for this situation, with the CMS software stack (CMSSW) already compiled for multiple architectures. In order to allow for a production use, the tools for workload management and job distribution need to be extended to be able to exploit heterogeneous architectures. Profiting from the opportunity to exploit the first sizable IBM Power9 allocation available on Marconi100 HPC system at CINECA, CMS developed all the needed modifications to the CMS workload management system. After a successful proof of concept, a full physics validation has been performed in order to bring the system in production. The experiences are of very high value, when it comes to commissioning of the similar (even larger) Summit HPC system at Oak Ridge, where CMS is also expecting a resource allocation. Moreover the compute power of those systems is being provided also via GPUs and this represents an extremely valuable opportunity to exploit the offloading capability already implemented in CMSSW. The status of the current integration including the exploitation of the GPUs, the results of the validation as well as the future plans will be shown and discussed.

1. Introduction

Over the last few years the CMS [1] experiment has begun to prepare its computing system to effectively exploit and benefit from a heterogeneous non-x86 resources landscape, where *ARM*, *POWER* and possibly *Risc-V*, will become available. For the moment the main targets are heterogeneous resources available from opportunistic providers such as HPC centres. This preparatory work will certainly be beneficial also in view of a future evolution of the regular grid infrastructure.

While the CMS software stack (CMSSW [2]) is already compiled for multiple architectures since years and all recent CMSSW releases, including nightly, patch- and pre-releases are regularly built for *x86-64, ARM* and *ppc64le* architectures, the computing infrastructure as a whole has been developed to support a single architecture.

Recently, significant effort has been invested by CMS Offline and Computing in order to generalise several assumptions in the services (such as CRAB [3] and WMAgent [4]). The first achievement was an official release of WMAgent, which includes full support for job submission to *ppc64le* based resources. This was achieved by integrating the resources made available at CINECA [5] on Marconi 100, which provided a sizable allocation on Power9 + GPU [6].

After successfully extending the WMAgent system to support more than one architecture, CMS was in the position to enable full support for multiple architectures in the full stack of the computing system. This allowed for the first time the production of official data samples in order to perform the Physics Validation of the PPC architecture and to declare *ppc64le* certified for production purposes. The process and the strategy to achieve this goal is detailed in the following section.

2. Enhancing the CMS Computing system

From the operational perspective, the ultimate goal for the CMS Offline and Computing is to enable the ability to define a list of supported architectures for each injected workflow, labelled e.g. 'slc7_amd64_gcc9', 'slc7_ppc64le_gcc9', 'el8_aarch64_gcc11', and then to select at the time of job scheduling, where to execute each payload depending on either a standard matchmaking process or on site defined rules, also called *custom matching expression*. This approach allows CMS to minimise, and possibly eliminate, any additional burden on the production teams caused by the existence of multiple architectures. Once a workflow is properly described, the underlying computing system needs to be able to transparently handle the various technicalities, such as job pressure on sites via architecture-aware pilots, to ensure the proper matchmaking of resources and payloads and the setup of the runtime environment. In order to properly set up the job runtime environment, the system needs to automatically detect the architecture on which the job has landed on and to adapt accordingly.

To achieve this goal, a number of changes had to be made to two main areas of CMS Computing, namely: Workload Management and Submission Infrastructure [7]. The key interactions between these components are shown in figure 1.

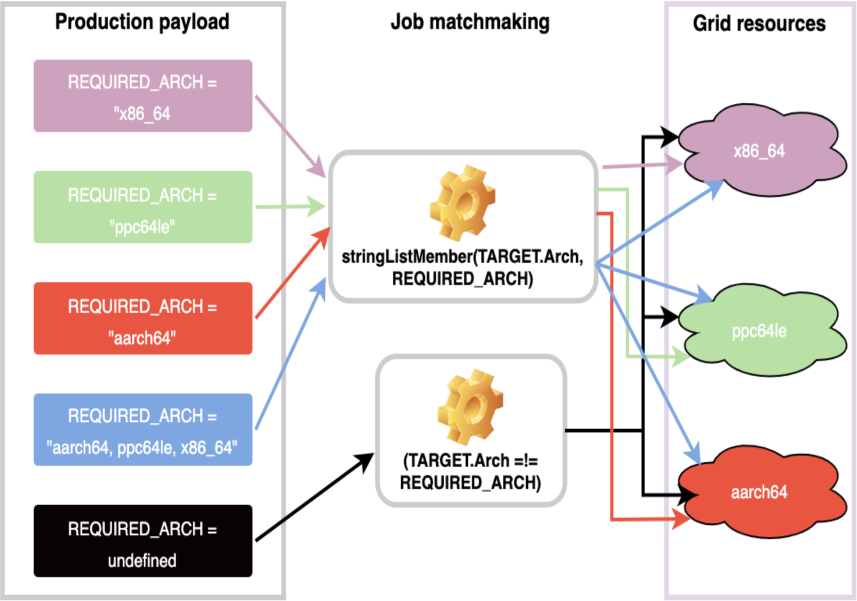

Production payload
Job matchmaking
Grid resources
REQUIRED_ARCH = "x86_64
REQUIRED_ARCH = "ppc64le"
REQUIRED_ARCH = "aarch64"
REQUIRED_ARCH = "aarch64, ppc64le, x86_64"
REQUIRED_ARCH = undefined
stringListMember(TARGET.Arch, REQUIRED_ARCH)
(TARGET.Arch =!= REQUIRED_ARCH)
x86_64
ppc64le
aarch64

Figure 1. Relationship between payload description, the matchmaking expression and the resources

2.1. Workload Management

The Workload Management system (WM) represents the bridge between workflow requestors, typically expert users from the physics community, and the computing operations team. Requestors are responsible for the workflow definition, including the definition of resource requirements. The WM layer needs to know where to find these resources in order to target them with payloads based on constraints, such as data locality, CPU architecture, sufficient RAM available etc.

Requestors define which architectures are allowed to be used in a given workflow or a task of the workflow, based on the availability of the CMSSW software.

The WMAgent, a core component of the WM system, propagates the architecture requirements from the workflow description to the actual job description via HTCondor [8] job ClassAds. The latter will then be used by the Submission Infrastructure services as explained in the section 2.2.

In order to enhance the WM system to fully support the multi-architecture work, several changes were required at different layers of the WM system:

1. Requests Manager (ReqMgr2) has been enabled to accept several architecture labels in the workflow description
   - "ScramArch": ["slc7_amd64_gcc900","el8_ppc64le_gcc11"]
2. WMAgent has been extended to propagate the workflow description all the way to the HTCondor job classad definition
   - REQUIRED_ARCH = “x86_64,ppc64le”
   - Requirements = 'stringListMember(TARGET.Arch, REQUIRED_ARCH)'
3. The job runtime library has been equipped to perform an auto-discovery of the platform
   - machine = platform.machine()
4. The runtime job wrapper has been evolved to configure the batch slot based on the architecture previously detected

2.2. Submission Infrastructure

The Submission Infrastructure (SI) manages the resources from geographically distributed providers that are joined into a dynamically sized and centrally managed HTCondor pool, the CMS Global Pool. It uses the GlideinWMS system [8], whose task is to submit pilot jobs (glideins) to the Grid resources of (WLCG [9]) sites. It also incorporates other types of resources such as computing resources allocated at HPC centres or commercial and academic clouds. The GlideinWMS performs the resource provisioning based on requirements of the production payload. Once computing resources are available, there is a job matchmaking between the host and the payload. The described system also supports the so-called *manually launched glideins*. In this case, as the name implies, the *glideins* are started manually, e.g. by the resource provider, and not via GlideinWMS. However, once bootstrapped they also join the Global Pool.

In order to enable multi-architecture workloads, WMAgent propagates the architecture ClassAd parameter, which must then be used to match to an appropriate slot. If a job does not require any architecture, then any grid resource can be used.

During the initial integration phase of *ppc64le* at M100, resources were joined into the Global Pool via manual glideins. Later, CMS enhanced the GlideinWMS system with a generic mechanism that allows submission of pilots to regular Grid resources, depending on the payload architecture. With this the first matchmaking became architecture aware, moving away from the assumption that all resources are “*x86_64*”. The Glidein Factory evolved to support the capability to specify the architecture through a new attribute in the entry description: *GLIDEIN_REQUIRED_ARCH***.** As a result, pilots will be submitted according to this value in order to put pressure on the right providers.

Once the pool of resources with heterogeneous architectures is built, the most suitable slot for the job execution can be determined via the so-called second matchmaking step. The latter will also honour a list of target architectures that the WMAgent has specified in the requirements i.e *Requirements = (stringListMember(TARGET.Arch,"x86_64,ppc64le")).*

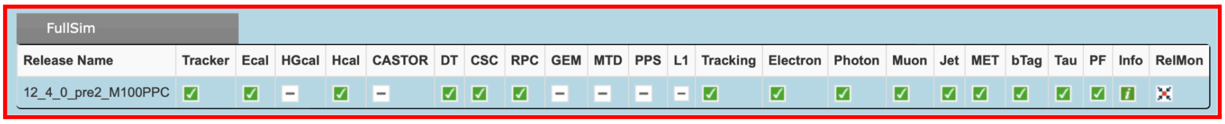
Once the pilot lands on the node, HTCondor automatically adds an attribute in the machine ClassAd that identifies the platform of the machine.

3. Validation strategy

Regarding the physics validation, the ultimate goal of CMS is to allow the use of the samples produced on *ppc64le* (i.e. on Marconi100) for CMS approved Physics papers, ensuring that any scientific measurement remains independent of the architecture used to process the data and Monte Carlo simulation samples.

Taking advantage of the first sizable IBM Power9 allocation available on the Marconi100 HPC system at CINECA, CMS was able to perform such validation procedure. The strategy was to run established Release Validation Monte Carlo workflows [10] on *ppc64le*, both with and without Pileup, to carefully compare them with the samples from the same workflow executed on *x86* resources at CERN, using the same CMSSW release and workflow configurations. The production of large samples allowed a study of the distributions of physics observables regarding any significant deviations.

Technically, the validation was performed using the regular production system. The operations team introduced a new feature to identify such workflows and assign them to the site where the Power machines are available. Employing the established production pipeline [11], all existing features such as automatic input and output placement, monitoring and integration with Data Management enabled smooth production. The use of RelMon [12], the official CMS tool for performing automated comparisons of two root files containing histograms and profiles, was particularly beneficial. For example, it allows to make regressions of CMSSW releases by evaluating the produced Data Quality Monitoring (DQM) histograms. Its primary use case is the systematic validation of a 'test CMSSW release' against a 'reference CMSSW release'. Figure 2 shows a RelMon screenshot.



Release Name	Tracker	Ecal	HGcal	Hcal	CASTOR	DT	CSC	RPC	GEM	MTD	PPS	L1	Tracking	Electron	Photon	Muon	Jet	MET	bTag	Tau	PF	Info	RelMon
12_4_0_pre2_M100PPC	☑	☑	−	☑	−	☑	☑	☑	−	−	−	−	☑	☑	☑	☑	☑	☑	☑	☑	☑	i	☒

Figure 2. Screenshot of the RelMon GUI showing the successful tests

Several physics processes have been generated, such as top quark pair production, Drell-Yan, W+jets, SUSY, etc. The results of the comparisons are reported by various subsystem experts from sub detector groups and physics object groups. Figure 3 shows some example distributions.

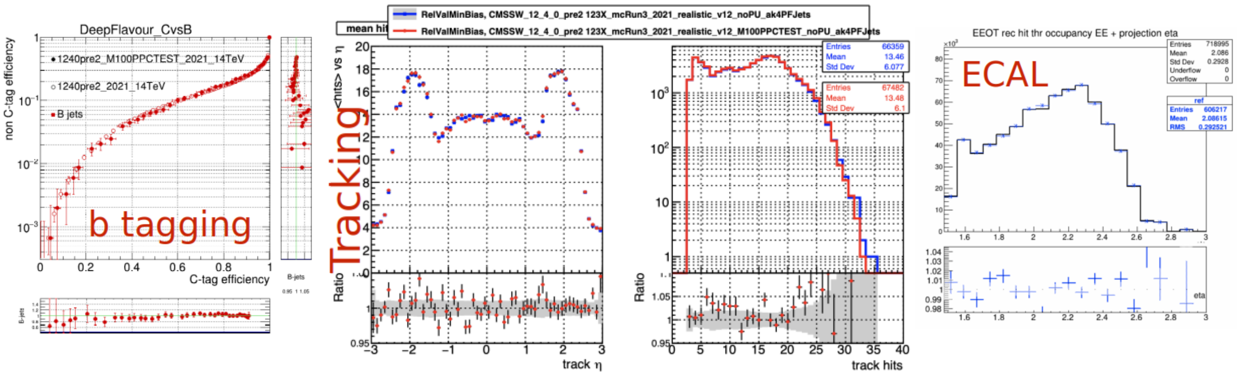


Figure 3. Example plots form the the Physics Validation procedure

4. Summary and future work

The Power9 CPU architecture has been validated by CMS for generation and processing of physics data and is there declared ready for production usage. The experiences gained during the successful integration of Marconi 100 at CINECA will also be beneficial for the integration of other POWER based HPCs, such as the OLCF Summit machine in the USA. On the technical side, the production and processing software stack of the experiment has been enabled, for the first time, to transparently handle heterogeneous non-x86 resources. Support for multiple architectures, in principle any, has been enabled in the computing stack of the experiment. Together with the already implemented support for the ARM architecture in the CMSSW application and the readiness of the distributed computing stack for multi-architecture enables CMS to fully exploit the ARM resources should they become available. Due to its superior power efficiency an increasing availability of ARM resources appears to be a very likely scenario in the near future.

References

[1] S. Chatrchyan et al. “The CMS Experiment at the CERN LHC”. In: JINST 3 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004

[2] https://cms-sw.github.io/

[3] “G Boudoul” et.al “Monte Carlo Production Management at CMS”, Journal of Physics: Conference Series, Volume 664

[4] D.Spiga.,et al. “A CMS application for distributed analysis”, (2009) IEEE Transactions on Nuclear Science, 56 (5), art. no. 5280527, pp. 2850-2858. DOI: 10.1109/TNS.2009.2028076

[5] CINECA Homepage: https://www.cineca.it/en

[6] T. Boccali, D Spiga, Alan Malta Rodrigues, M. Mascheroni, F.Fanzago, “Enabling CMS Experiment to the utilization of multiple hardware architectures: a Power9 Testbed at CINECA”. ACAT-2021 Journal of Physics: Conference Series 2438 (2023) 012031
IOP Publishing doi:10.1088/1742-6596/2438/1/012031

[7] Antonio Pérez-Calero Yzquierdo et al. “Evolution of the CMS Global Submission Infrastructure for the HL-LHC Era”, EPJ Web of Conferences 245, 03016 (2020)
https://doi.org/10.1051/epjconf/202024503016

[8] https://zenodo.org/record/7439153#.ZAC59kjMJhE

[9] HTCondor: https://research.cs.wisc.edu/htcondor

[10] Oliver Gutsche and (on behalf of the Cms Computing and Offline Projects) 2010 J. Phys.: Conf. Ser. 219 042040, DOI 10.1088/1742-6596/219/4/042040

[11] Virginia Azzolini, ”The Data Quality Monitoring Software for the CMS experiment at the LHC: past, present and future”, EPJ Web of Conferences 214, 02003 (2019),
https://doi.org/10.1051/epjconf/201921402003

[12] Danilo Piparo, “RelMon: A General Approach to QA, Validation and Physics Analysis through Comparison of large Sets of Histograms”, 2012 J. Phys.: Conf. Ser. 396 022011

Acknowledgments

The M100 integration was possible also thanks to the support made available by INFN-CNAF. The authors and the CMS collaboration as a whole would like to thank CINECA for the opportunity.