Power9 and CMS

(see list of contributors at the last slide)

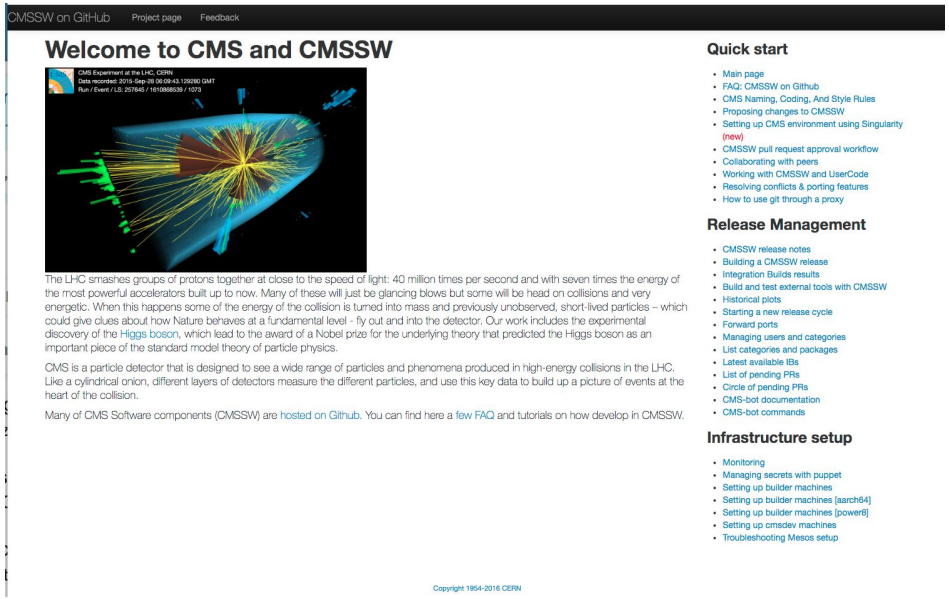
outline

- **Intro**
- **(Where) Power9 at CMS?**
- **(Why) Power9 at CMS?**
- **(How) Power9 at CMS?**

And what will follow from it ....

The CMS Software Stack

- Historically the CMS sw stack (CMSSW now, but ORCA/COBRA before) has been using only Open Source components (apart from one ORACLE lib)
- Initially (early 2000s), there was support for Linux/x86, Solaris and (some) Windows NT
  - Even if Linux/x86 has basically been the only production release since then
- SLOC applied to the master release today ➜




```
Total Physical Source Lines of Code (SLOC)                = 5,677,023
Development Effort Estimate, Person-Years (Person-Months) = 1,749.27 (20,991.23)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 9.14 (109.73)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 191.30
Total Estimated Cost to Develop                           = $ 236,302,497
 (average salary = $56,286/year, overhead = 2.40).
SLOCCount, Copyright (C) 2001-2004 David A. Wheeler
```

\+ ~600 external packages

Variants ...

- In most cases a production version of each release is defined to be used in production (**“slc5_amd64_gcc472” → SLC5 OS + x86/64 Arch + gcc version 4.72**)
- Still, the code is maintained at least buildable on many variants, used
  - For comparison purposes (“what if we change the version of gcc?”)
  - Because different architectures can show / hide “features” (resilience to memory errors, effect of compiler flags, etc)
- For example, typical variants have been
  - Chipset (Intel vs Power vs ARM)
  - Architecture (32/64)
  - OS (macos vs Windows vs Linux)
  - OS family (slc7 vs Alma8)
  - Compiler (gcc vs clang vs icc)
  - Version of external tools (Geant4 10 vs Geant4 9)
  - Memory handling (malloc vs tcmalloc vs jemalloc)
- Typically, no real request nor intention to maintain multiple production versions: compile and (if possible) run some tests

(Where) Power9 at CMS?

- Around 5 years ago some groups in CMS started to have access to HPC systems for testing / integration
- This goes in the direction of trying to offload a considerable part of the HL-LHC processing to these systems
- In the last ~5-10 years many non-x86 systems appeared at the top of top500.org; most notably IBM Power systems
- Go for it!

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,299,072	415.53	513.85	28,335
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438
.					
.					
.					
9	**Marconi-100** - IBM Power System AC922, IBM POWER9 16C 3GHz, Nvidia Volta V100, Dual-rail Mellanox EDR Infiniband, IBM CINECA Italy	347,776	21.64	29.35	1,476

Top500.org list June 2020: #2, #3 and #9 systems are IBM Power !

In particular: situation in Italy

- **CHEP 19 (Adelaide)**: Italy presented a contribution on the use of CINECA (Italian Prace Tier-0) KNL with WLCG processing, via an elastic extension of CNAF/Tier-1
- Not an architecture problem (Xeon Phi → x86_64), but an infrastructural one
  - Closed network, low RAM, no Local Disk, …
- 90 McoreH used by the 4 LHC Experiments

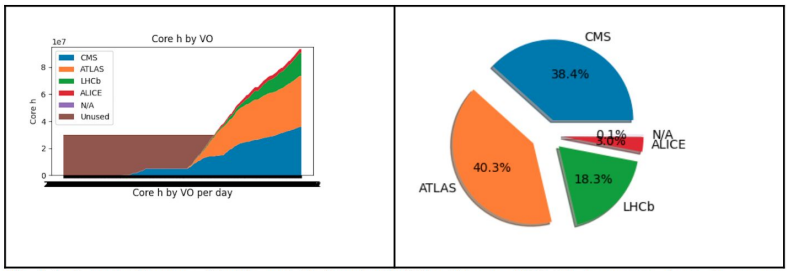


Fig. 7. Left: total utilization of the Marconi A2 from April 2019 to February 2021. The brown area shows the amount of remaining grant (30 Million core hours). (right) utilization by experiments as a fraction of the utilized 93 Million core hours.

The 3 parties in the game

- LHC Experiments
  - The italian groups in ALICE, ATLAS, CMS, LHCb, deliver 10-15% of the experiments' computing pledges.
- CNAF
  - Italian Tier-1 in Bologna. Main INFN data center providing computing and storage to more than 30 experiments
- CINECA
  - HPC center in Bologna. EU/Prace Tier-0.
  - Current top machine is **Marconi**, partially **KNL** and partially **SkyLake** based. At **21st position in top500.org**
  - Selected site for 1 of the 3 EU pre-exascale machines

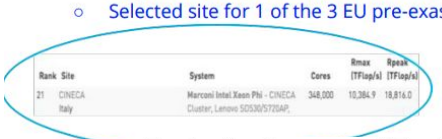
CNAF
- Standard "GRID-like"HTC farm (30k cores, 400 kHS06)
- 38 PB of disk
- 90 PB of tapes on 2 libraries

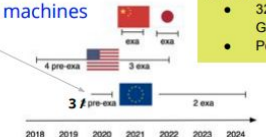
CINECA

Marconi cluster
- Based on Omnipath
- ~19 Pflop/s
- 17 PB of local storage

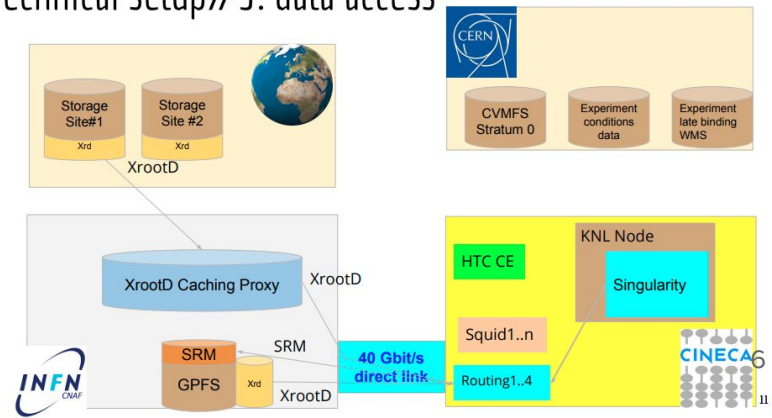
Marconi A2 Partition
- 3600 nodes with 1 Xeon Phi 2750 (KNL) at 1.4 GHz and 96 GB of RAM
- 68 cores/node, 244800 cores
- Peak Performance: ~11 Pflop/s

Marconi A3 Partition
- 3216 nodes with SkyLake at 2.1 GHz
- Peak Performance: ~8 Pflop/s





Technical setup#3: data access



(Why) Power9 at CMS?

- **Not** really because we expected to increase substantially the available CPU resources for CMS
- **Not** because we think Power is a relevant platform for the next years (indeed the current HPC systems in operations may well be the last)
- **But**:
  - Because it was the first non x86 platform available at scale
  - Because it is a large scale attempt of integration WLCG-HPC/PRACE

MARCONI - 100

Nodes: 980

Processors: 2x16 cores IBM POWER9 AC922 at 3.1 GHz

Accelerators: 4 x NVIDIA Volta V100 GPUs, Nvlink 2.0, 16GB

Cores: 32 cores/node

RAM: 256 GB/node

Peak Performance: ~32 PFlop/s

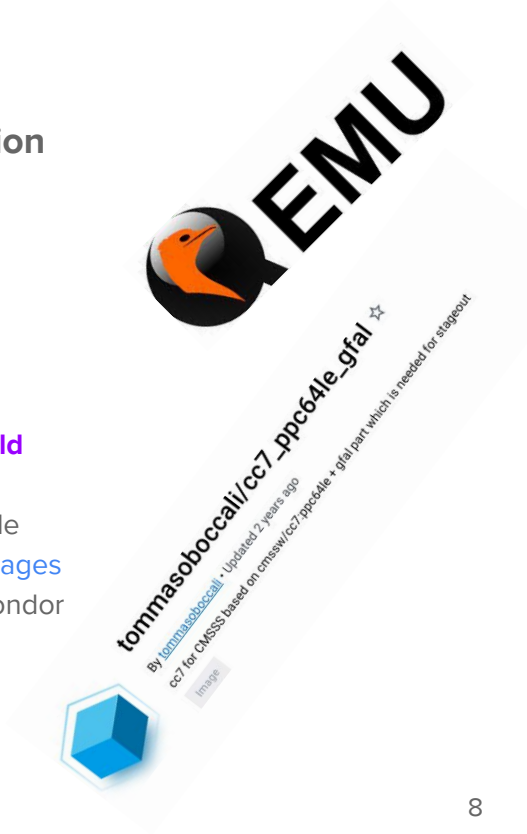
O(10^{6-7}) Core Hours available via 2 projects:

- Standard Allocation for LHC tests
- PRACE Project “PowerAtCMS”
  - (Boccali, Pierini, Spiga)

(How) Power9 at CMS?

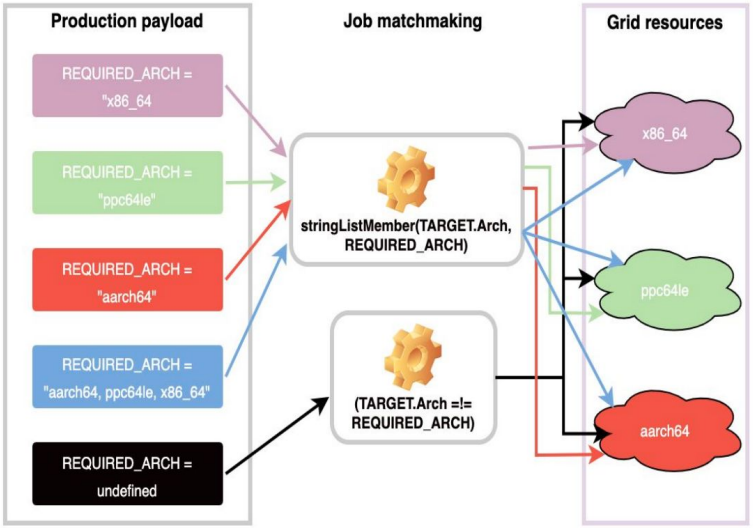
- As shown in the initial slides, CMS does routinely compile CMSSW for various platforms, among which ppc64le (Power8/9) and aarch64 (ARM 64bit) ➜ **no action needed on this side**
  a. **Thanks to the Openlab for the availability of machines at CERN**

- But at least 3 problems arise:
  a. **(more a feature): there was no Power9 HTCondor available, nor a singularity image we could use with some bells and whistles (VOMS, GFAL, ...)**
     - To create a viable image repository, we needed access as ***root*** to CINECA, not available
     - We resorted to using QEMU to emulate a ppc64le on a x86_64 ➜ prepared a list of images
     - No more a problem today; HTCondor + images are promptly available at CERN + HTCondor
     - TODAY: we can use directly **/cvmfs/unpacked.cern.ch/registry.hub.docker.com/cmssw/el8:ppc64le**



Job distribution / pilot provisioning

b. **The central and local WMS software stacks (mostly Python, which is good!) were written assuming the payloads are x86_64 ➔ we need to make the architecture a configurable parameter ➔ see also Antonio's talk today**



- Jobs insertion in the condor pool (WMAgent): prepare a HTCondor requirements matching multiple architectures
  - Requirement ➔ ['slc7_amd64_gcc820','slc7_ppc64le_gcc820']

Requirements = (stringListMember(TARGET.Arch,"ppc64le,X86_64")) && (TARGET.OpSys == "LINUX") && (TARGET.Disk >= RequestDisk) && (TARGET.Memory >= RequestMemory) && (TARGET.Cpus >= RequestCpus) && (TARGET.HasFileTransfer)

- Provisioning of the resource: GlideinWMS sends the proper pilot (via the factory) via a HTCondor CE **- OR -** we can use a standalone factory connected to CINECA's Slurm that maintains a "pressure" of "Manual Glideins" on the site

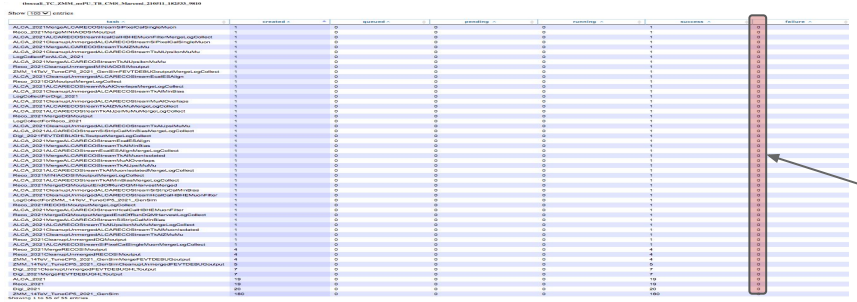
```
(base) [dspiga00@login03 slurm_glidein_spiga]$ python slurm_queue_manager_spiga_nogpu.py
===== Configured with:
Slurm command squeue -l -n CMS_VAL --me
Max_Running 3
Max_Pending 2
Job Executor ./slurm_glidein_nogpu.job
Log Prefix logs/job_log_nogpu
SLURM Job Name CMS_VAL
=== Date: 2023-04-21 17:21:01.281879  1682090461
(running, pending) = (0,0)
Submitting 2 jobs.
Submitted job logs/job_log_nogpu1682090461
Submitted job logs/job_log_nogpu1682090465
=== Date: 2023-04-21 17:21:14.308249  1682090474
(running, pending) = (0,2)
=== Date: 2023-04-21 17:21:19.326226  1682090479
(running, pending) = (0,2)
=== Date: 2023-04-21 17:21:24.338158  1682090484
(running, pending) = (0,2)
=== Date: 2023-04-21 17:21:29.354929  1682090489
(running, pending) = (0,2)
```

The physics validation process

c) **Even if it runs (== does not crash), it is not immediately usable for Physics production unless a proper Physics Validation happens**

CMS uses the Release Validation framework to routinely assess the physics quality of software releases (== *bugs are not introduced*) AND to perform differential analyses (== *a change of compiler does not introduce features*)

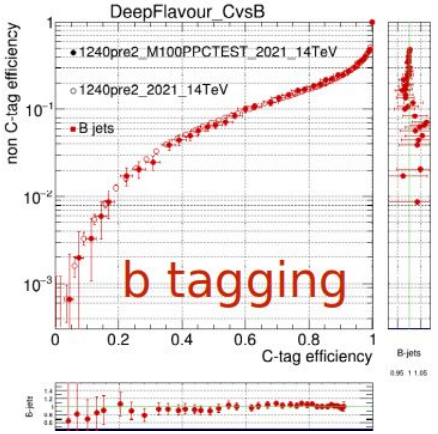
- Release Validation uses various processes (ttbar, DY, W+jets, SUSY) in various scenarios (Phase-1, Phase-2, Heavy Ions, various PU settings, single particle samples) to make sure all CMS Physics use cases are covered
- Validation happens at the level of detector response AND at the level of Physics Objects
- RelMon performs automated differential comparisons, but the final "stamp" is still human driven

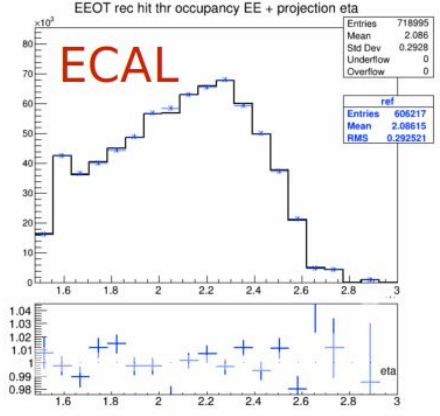


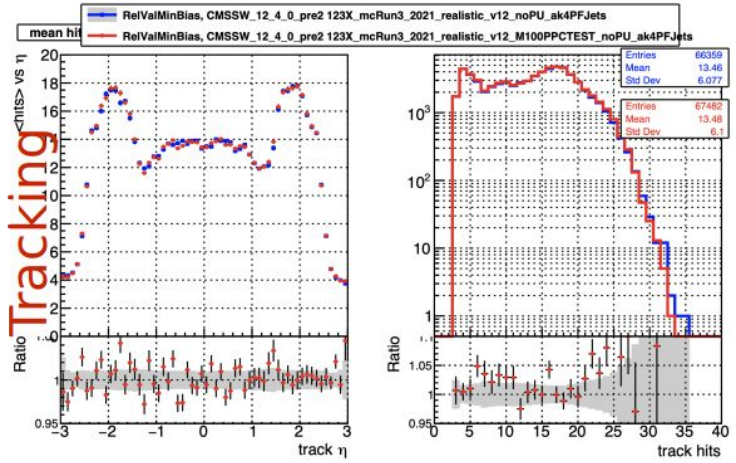
Unreadable but "no failures"

Validating Power8/9 for CMS Physics

- In the end, Power vs x86 is different from comparing plts between G4.9 and G4.10: use the exact same code base + externals and search for sizeable differences with only arch changed
- O(10 M Events) needed to perform a full validation in all the relevant scenarios







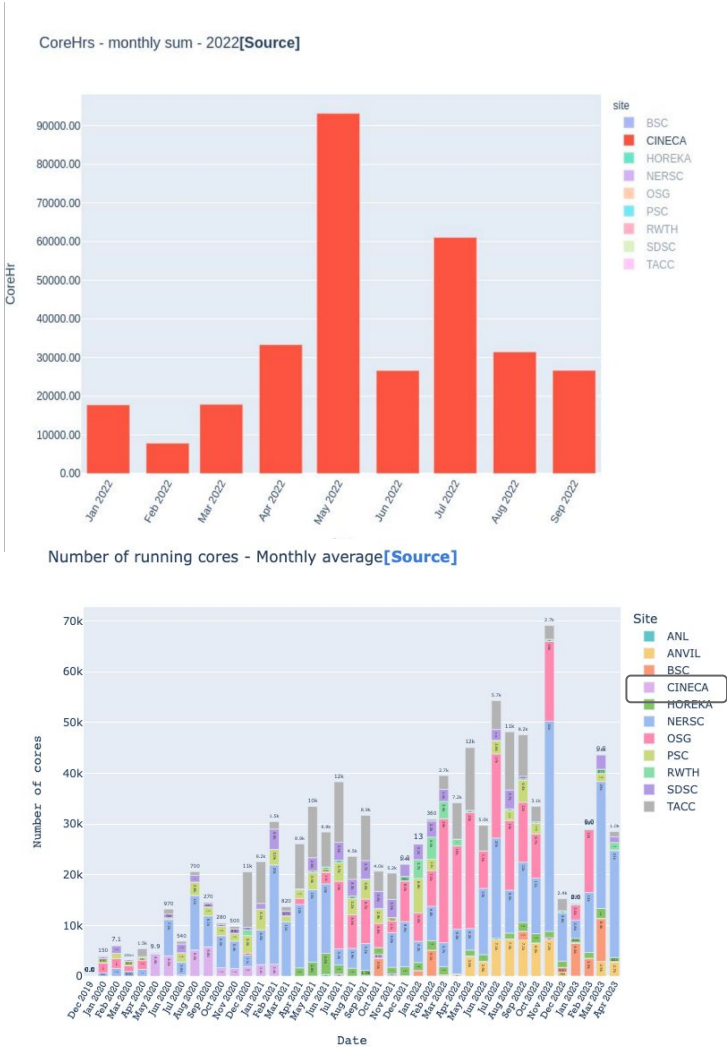
The required subsystem level checks are ok → validated!

FullSim

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	✖

What now?

- IBMPower is a production arcbhitecture for CMS. It is not an overall sizeable contribution, but constantly available @ CINECA
- 2023:
  - **CINECA: will be terminating Marconi 100 by the year's end**
  - **SUMMIT: small grant available and under testing; there will most probably not be a follow-up**
- Still, the important gain from the story is the new capability of our WMS to handle multiarch workflows. As an example, current physics validation tests with ARMv64 (again at CNAF/Tier-1) were "completely transparent" from operational and code points of view

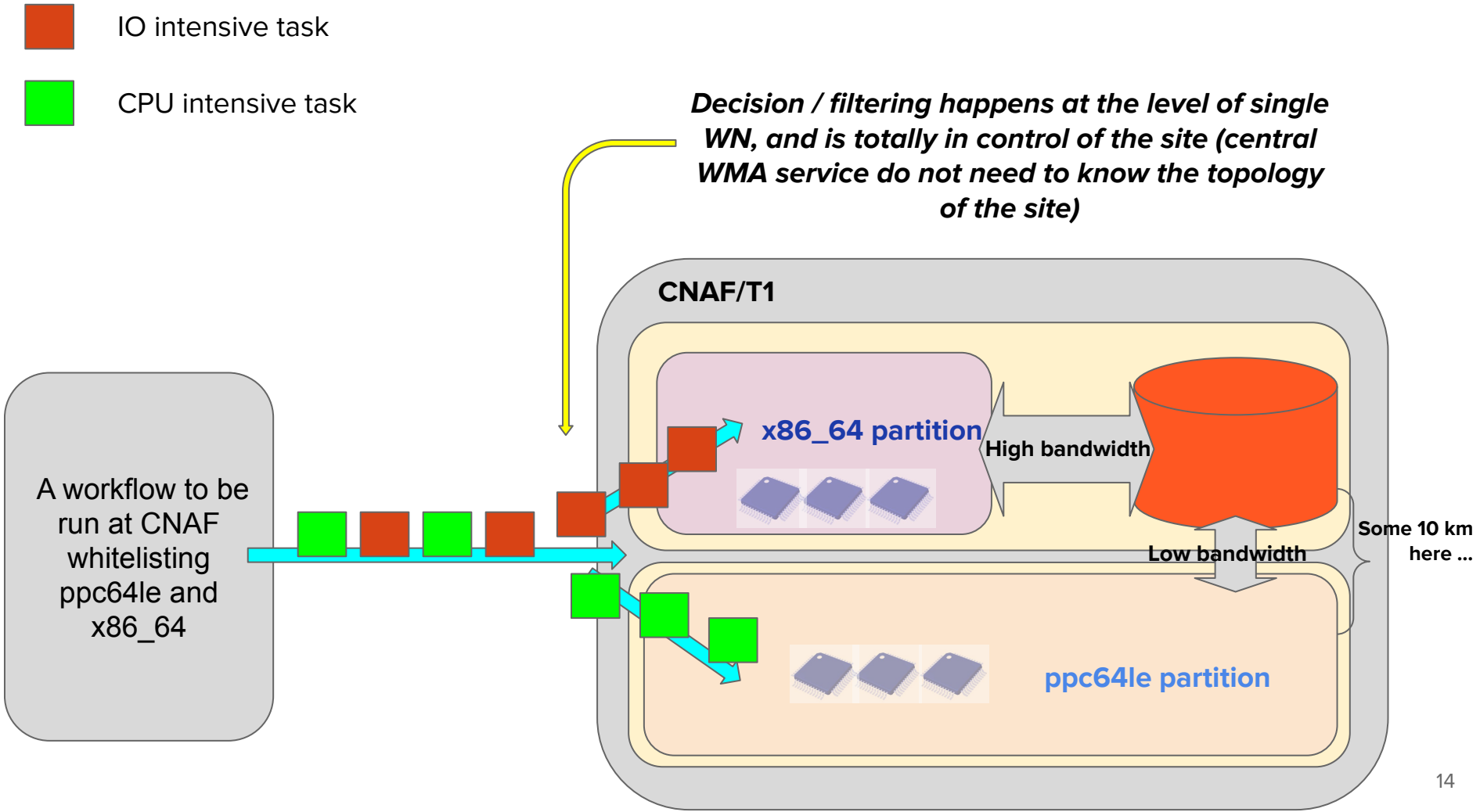


(an intermezzo: fruitful interference multiarch + extension mechanism + local glideIn match)

- It is not immediate to assume CMS will provision non x86_64 systems in owned centers; these resources will come mostly from external / HPC systems
- These are usually “problematic”; for example, network bandwidth to WLCG storage can be scarce
- The simultaneous utilization of Multiarch, Elastic extension and local GlideIn match offers interesting possibilities!

- If a site has 2 subsites:
  - A real one, x86_64, close to the storage (the standard T1)
  - A remote one, with Power, with a less performant connection to the storage (the remote HPC used as elastic extension)

➔ the **local GlideIn match mechanism** can be used to process on the remote site only CPU intensive tasks in a workflow, leaving the IO intensive on the standard site


IO intensive task
CPU intensive task
Decision / filtering happens at the level of single WN, and is totally in control of the site (central WMA service do not need to know the topology of the site)
CNAF/T1
x86_64 partition
High bandwidth
Low bandwidth
Some 10 km here ...
ppc64le partition
A workflow to be run at CNAF whitelisting ppc64le and x86_64

Conclusions

- The availability of sizeable resources of Power9 resources initially at CINECA has served as the push to adapt the WMAgent code to multiarch situations
  - Not only supporting workflows designed for different archs, but also supporting more than 1 arch within a single workflow
- USCMS with SUMMIT is performing a similar validation on a much larger system (SUMMIT ~ 10x CINECA);
- While the total impact of current non x86_64 resource is still negligible, once you have done the step from the first to the second architecture, next steps are transparent
  - Currently using the same mechanism out-of-the-box for ARM64 validation again as an extension of CNAF

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

Contributors

- The CMS Offline and Computing Coordination:
  - Saqib Haleem, Dirk Hufnagel, Alan Malta Rodrigues, Jordan Martins, Marco Mascheroni, Hasan Ozturk, Antonio Pérez-Calero Yzquierdo, Kirill Skovpen, Todor Trendafilov Ivanov, Christoph Wissing, Edita Kizinevic, Nikos Tsipinakis
- The (CMS+CNAF)-Computing team in Italy:
  - Daniele Spiga, Stefano Dal Pra, Lucia Morganti, Diego Ciangottini, Tommaso Boccali