Enabling INFN-T1 to support heterogeneous computing architectures

S Dal Pra, D. Spiga et al.
CNAF Tier-1 (Grid / Batch computing)

- Providing ~ 715KHS06 / 59Kcores, 1010 Compute nodes to ~ 23 WLCG communities (HEP / astroparticle / astrophysics) and ~ 30 local research groups.
- 6 x HTCondor-CE 5.1.6 on top of HTCondor 9.0.17
- Moving to tecnopolo (→ #392, track 7, 9 May 2023, 12:00)
  - Opportunistic usage of Computing resources from Leonardo HPC is foreseen by an early agreement
  - x86_64 CPU, 128 cores, 512GB RAM, 4xAmpere GPU, Slurm batch system
- Initial investigations on: Marconi 100
  - PPC CPU, 192 cores, 256GB RAM, 4xV100 GPU, Slurm batch system

Our Goal: seamless integration of opportunistic and pledged resources.
Marconi 100: Power AC922 “Whiterspoon”

- 980 CN + 3 login nodes
- 2x16 core IBM 8335-GTG@2.6GHz
  - Up to 128 threads
- 4xNVIDIA V100 GPUs
- RAM: 256 GB/node
- Local disk: 1.6TB NVMe
- Shared Disk Space: 8PB (GPFS)
- Slurm scheduler (whole node/24h)
# Pledged vs Opportunistic

<table>
<thead>
<tr>
<th>CNAF (Pledged)</th>
<th>M100 (Opportunistic)</th>
<th>General Case (Opp.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid+local, HTC-CE / HTC</td>
<td>Local, Slurm</td>
<td>Batch, Cloud, K8S, …</td>
</tr>
<tr>
<td>1 or 8 cores, ≥ 3.5GB/core</td>
<td>Whole node, 2GB/core</td>
<td>…</td>
</tr>
<tr>
<td>x86_64</td>
<td>PowerPC, NVIDIA V100</td>
<td>X86_64, ARM, PPC, GPU</td>
</tr>
<tr>
<td>72h or more runtime</td>
<td>24h max runtime</td>
<td>…</td>
</tr>
<tr>
<td>Full network for cluster operation, sw distribution (cvmfs) and data transfer</td>
<td>Agreed Outbound Connectivity toward known networks (CNAF, CERN) + cvmfs</td>
<td>Outbound connectivity is assumed. CVMFS is a strong requirement</td>
</tr>
</tbody>
</table>

**Jobs must be able to run with the available QoS** ➔ 2 problems:
- Experiment side: Have suitable payloads to run on opportunistic resources
- Provider side: steer most suitable jobs there

**Note: Opportunistic != Free**
M100: From Slurm CN to CNAF WN

- Two simple ideas:
  - have a HTCondor STARTD running as Slurm Job
  - Detect pending jobs for ppcml64 @CNAF and trigger resource creation

How to implement it

1. From a M100 Login Node submit a Slurm job
2. At start, it launches a Singularity container which
   - activates an HTCondor STARTD, which
   - authenticates to the Central Manager and join the CNAF pool
     - CCB (Connection Control Broker 9618 port)
     - IDTOKENS for authentication
   - Becomes available to execute jobs submitted to HTCondor-CE at CNAF.
     - StartJobs expression to only accept proper jobs
Main Idea: Make Jobs for M100 identifiable at submit time. Needed information must be available at the JobRouter of the HTC-CE

Different approaches possible, each has pros and cons

1. Set a custom attribute in the Submit File:
   - Example: +WantRoute = “cms_m100”
   - Define a JobRouter entry to add Arch == “ppc64le” to the Requirements of the routed job

2. Set an agreed claim in the access token (SCITOKEN, IAM Token, EGI Check-in Token)
   - Several token claims can be inspected as Classad Attributes by the JobRouter
   - Latest HTCondor versions (10.4.1) makes possible further customization
**M100: Autoscaling availability**

**Goals:**

- Trigger instantiation of M100 resources (i.e. submit Slurm wholenode jobs) only when suitable payloads are queued at CNAF.
- Avoid having unused PPC nodes running without payload.

- **A script running on a M100 login node:**
  - checks CNAF queues to for jobs awaiting PowerPC resources
  - checks Slurm for running and pending wholenode jobs (Slurm_R, Slurm_P).
  - If pend_CNAF > 0, it submits some wholenode jobs to Slurm.
  - If pend_CNAF == 0, Idle Slurm jobs (the hosted STARTD has no jobs) are terminated. This ensures that M100 resources are accounted only upon real need. **Note:** this cannot work with late binding (i.e. pure pilot) model
1. Job submitted to HTCondor-CE defines a custom Attribute +WantRoute in submit file
2. A JobRouter rule matches with the WantRoute custom attribute and
3. Arch=="ppcm164" to routed job requirements
4. The routed job is queued to the SCHEDD
5. The Queue Manager polls CNAF SCHEDD for queued ppcm164 jobs. If there are,
6. A Slurm Job is submitted, depending on number of current pending/running Slurm Jobs
7. At start, the Slurm Job launches a Singularity container which in turn
8. Executes a HTCondor STARTD configured with IDTOKEN credentials and proper StartJobs policies
9. The STARTD authenticates with the CNAF HTCondor pool through the Connection Control Broker

= IDTOKEN
**Generalizations**

The above model is quite generic and can be easily generalized for different computing architectures / infrastructures. This has been actually done:

**Herd**: K8S WN-pod instantiated via Cloud for preliminary tests, preparing for usage on ASI resources

**UniBO - Open Physics Hub**: Slurm Batch, x86_64 Compute Nodes (debian) with containerized STARTD

**Textarossa**: ARM architecture, see next slides
• E4 is a company with historical ties with CNAF and CERN computing

• Using hardware form the TEXTAROSSA EU project (EuroHPC), we had access to 2 Ampere Altra MAX nodes:
  ○ Dual 128 cores Neoverse N-1; the variant with 3.0 GHz (M128-30)
    ■ 256 cores per node
  ○ 2 memory settings: 256 GB and 1 TB
  ○ Fast local disk, poor WAN networking

• Using the experience gained on CINECA systems, we were able to integrate the nodes in CNAF computing (via HTCondor CEs) and run typical workflows on them; we also enabled the manual GlideIn method
  ○ HepScore: 3884 HepScore23
  ○ CMS: we were able to insert the machines in the production system, and start a the standard validation machinery
    ■ Unfortunately, only @ pileup 0, due to the networking
    ■ Amazing results, still: less than 1% errors
    ■ In course of validation (but physics validation not conclusive w/o pileup)
Validation process

Technical integration successfully verified:
CMS sees ARM resources as CNAF local computing capacity

Two campaigns of data generation have been successfully completed
- Run3 scenario
- Phase2 conditions

Samples are injected in the CMS computing infrastructure and ready for the next step
Observations

Dealing with Heterogeneous resources exposes several points **worth considering**

- **IDTOKENS** management: some improvement needed (distribution, revocation)
- **CCB** is a potential bottleneck? (keep sandbox transfer at a minimum!)
- Pilot “late binding” can be “too late”:
  - a precious resource can be dedicated to a pilot who has no suitable payload to run there
  - The QoS at the (opportunistic) resource is unacceptable by the payloads
- **Declaring needs in the Access Token** could help, better than adding custom attributes in the submit file: The “VO manager” can map execution privileges to groups, such as: “GPU_<model>”, “aarch64”, “ppc64lm”, thus controlling who can use special resources.
  - Easy to agree between small experiments and Site Administrators
  - Latest HTCondor (10.4.x) enables custom Token validation PLUGIN (custom token ➔ user mapping)

"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"