**Abstract**

The High Performance Computing (HPC) domain aims to optimize code in order to use the last multicore and parallel technologies including specific processor instructions. In this computing framework, portability and reproducibility are key concepts. A way to handle these requirements is to use Linux containers. These “light virtual machines” allow to encapsulate applications within its environment in Linux processes. Containers has been recently rediscovered due to their abilities to provide both multi-infrastructure environment for developers and system administrators and reproducibility due to image building file. Two container solutions are emerging: Docker for micro-services and Singularity for computing applications. We present here the ComputeOps project which has the goal to study the benefit of containers for HPC applications.

**ComputeOps: container for High Performance Computing**

C. Cavet, S. Gadrat and the ComputeOps project members
ccavet@apc.in2p3.fr & sebastien.gadrat@cc.in2p3.fr

---

**HPC**

- Portability and reproducibility/repeatability in research.

**French partners**

- ComputeOps partners:
  - Aristote: GENCI, IDRIS (french supercomputers).
  - Institutes and labs.
  - P2IO ACP:
    - GPGPU: two cards of 1GB;
    - CUDA 9.2 and 9.0, last Nvidia driver;
    - TensorFlow library;
    - New: FPGA card (Alvio U280 Xilinx Data Center Accelerator FPGA card.)

**ComputeOps project**

**Goal:** the project [1] studies and provides tools for the community.

- container technology interoperability.
- portability on various architectures and infrastructures.
- repeatability of computing.
- good practices for writing recipes.

**Sharing container image**

- Hubs, registries and Marketplaces: provide catalogues of images.
  - sharing images with tagged versions;
  - giving metadata information: owner, recipe, code repository, etc.

**CI with containers**

**Goal:** a way to automatically build and share container images on registries.

- Commit on GitLab-CI: run the Docker image creation stage and push images on the private registry;
- Singularity in Docker (SinD): automatically build Singularity images with infrastructure specifications (bind volumes) and publish/update them on hubs;
- Singularity Hub (Shub): collections of automatically built images;
- Run in production: on a HPC cluster within the scheduler.

**References**


The project is founded by the CNRS IN2P3 (French Institute for Nuclear Physics and Particle Physics) and is part of the DocaLog Master Project.