Next Generation of HEP CPU Benchmarks

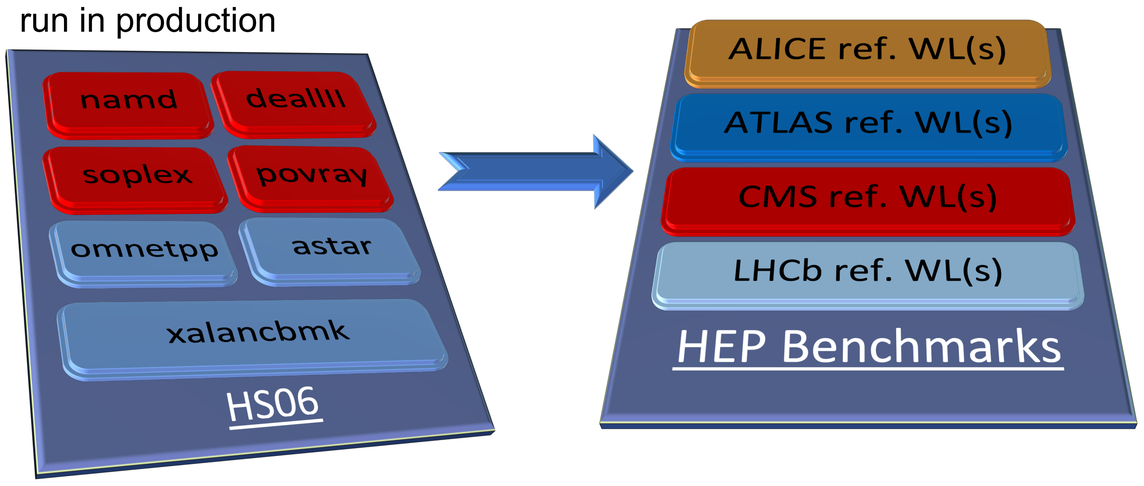
D. Giordano[1], E. Santorinaiou[1]

On behalf of the HEPiX Benchmarking Working Group

[1] CERN IT Department

Motivations

- HEPSPEC-06 (**HS06**) [1] is a decade-old CPU benchmark suite used to procure, pledge and account the **WLCG** compute resources. Its adoption spans from the hardware vendors, to the site managers, funding agencies and software experts
- Recent studies [2] show
  - **lack of correlations** with some of the HEP
  - no advantage in adopting **SPEC CPU 2017**, the industrial standard successor of HS06
- We propose a **suite alternative** to the industrial benchmarks, that adopts directly the workloads (Gen, Sim, Reco, Analysis) that HEP Experiments run in production



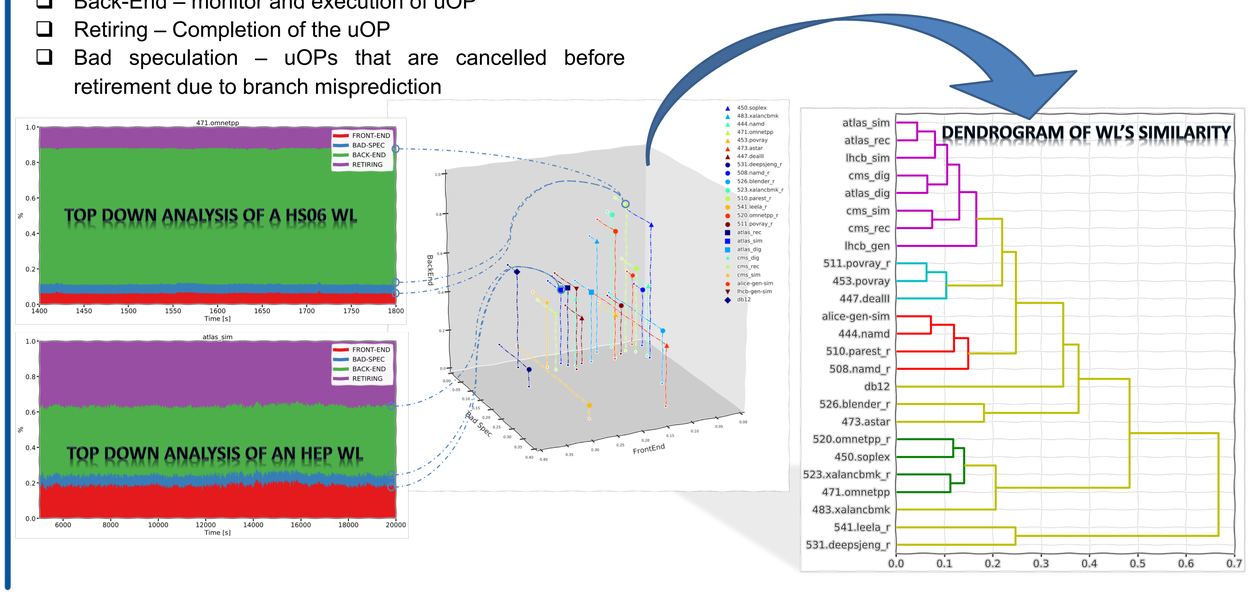
Requirements for a HEP Benchmark Suite

- Running Experiment's workloads requires knowledge of several components that make hard a **generalized adoption** as **benchmarks**
  - E.g.: find software, define exp. environment, run appropriate configuration file, have the correct input data and conditions, understand which metrics to extract
- Our approach:
  build **standalone containers** encapsulating **all and only** the dependencies needed to run the benchmarks
- The **HEP benchmark suite** will be adopted also by experts external to the Experiments, therefore shall have free license and long-term support
- The workloads must be easy to use, be w/o remote data access, guarantee the result reproducibility and error handling

Select the right benchmark

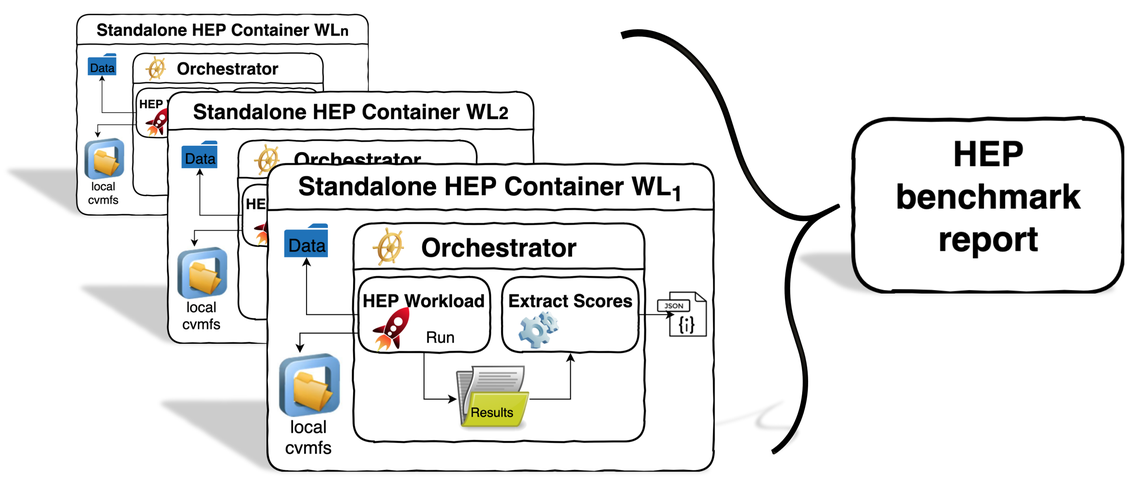
- **Trident** [3], an analysis tool of the hardware performance counters, has unveiled the **dissimilarities** between HEP workloads and the SPEC CPU benchmarks

Characterization of the resources utilised by a given workload
- Front-End – fetch and decode program code
- Back-End – monitor and execution of uOP
- Retiring – Completion of the uOP
- Bad speculation – uOPs that are cancelled before retirement due to branch misprediction



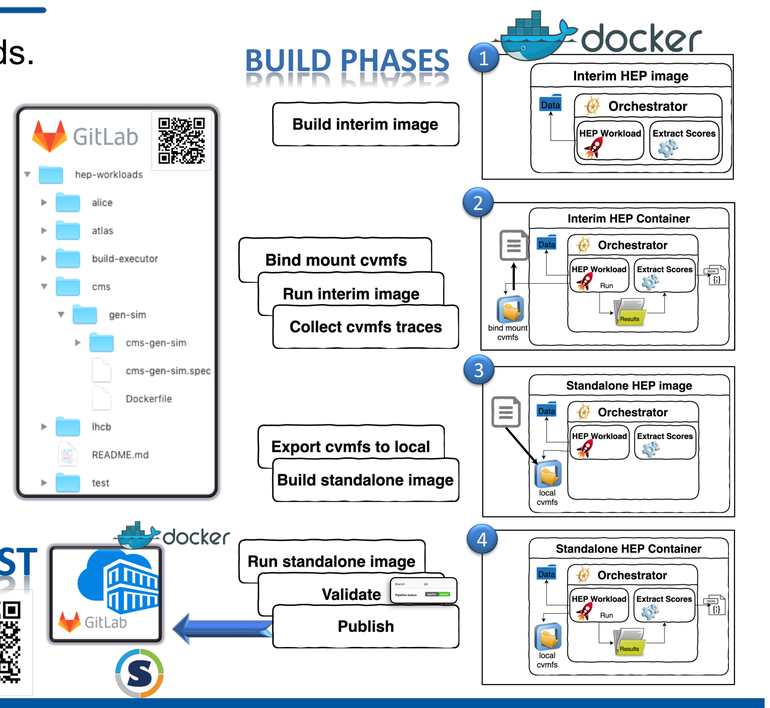
Components of an HEP Workload

- SW repository, typically distributed via CernVM File System (**cvmfs**) [4]
- **Input data**: Event data and Condition data
- An **Orchestrator** script per workload
  - Configure the environment and run the HEP application
  - Handle errors
  - Parse the produced output and create **score results**
- For each experiments' workload used as benchmark, the **standalone container** includes the local copy of the needed software and data



A Repository of HEP Benchmarks

- A suite of HEP benchmarks requires stable procedures to build and distribute the benchmarking workloads. We have realized an effective and user-friendly infrastructure, leveraging
  - CVMFS **Trace** and **Export** [5] utilities to export the workloads' software from cvmfs to local
  - GitLab CI/CD for fully automated **continuous integration**
  - GitLab Registry for container distribution
- Experts from the Experiments focus on providing the HEP workloads: software, data, result parser
- Experts on benchmarking focus on running the containers and profiling the compute resources
- Two container solutions offered: **Docker** and **Singularity**
- Simple instructions to run the benchmark and produce the results in JSON format
  - docker run -v /some_path:/results $IMAGE
  - singularity run -B /some_path:/results docker://$IMAGE
- Coming soon: workloads for GPU profiling



References & Pictures' credits

Contact: domenico.giordano@cern.ch