CERN openlab

HEP Analysis workloads for the benchmarking suite

Dominika Kankowska
Supervisors: Domenico Giordano, Xavier Valls Pla

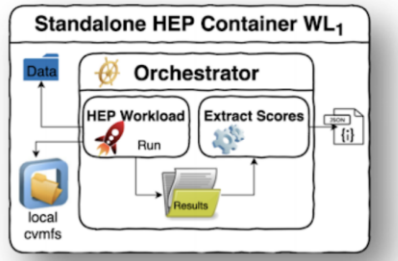
24.09.2020

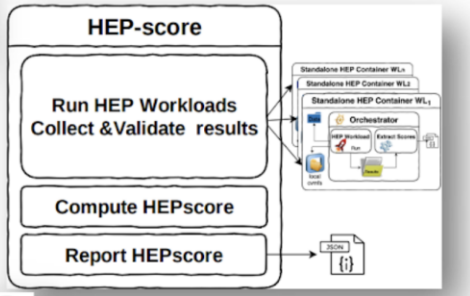
Introduction to HEP Benchmarks

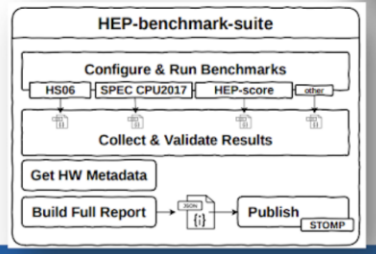
HEP Benchmarks project

Three components https://gitlab.cern.ch/hep-benchmarks

- *HEP Workloads*, *HEP Workloads GPU (new)*
  - Common build infrastructure
  - Individual HEP workloads
- *HEP Score*
  - Orchestrate the run of a series of HEP workloads
  - Compute & Report the **HEPscore** value
    - "Single-number" benchmark score
- *HEP Benchmark Suite*
  - Automate execution of multiple benchmarks
    - HEPscore, SPEC CPU2017, HS06, …
  - Publish results
    - Simplify the sharing, tracking and comparison of results







Tools

https://github.com/root-project/root
https://github.com/root-project/rootbench

Implementation

A CC7 container application where we parse rootbench's output and adapt it to the format expected by the BWG benchmarking suite

- Code in https://gitlab.cern.ch/xvallspl/hep-analysis-workloads
- Gathering execution time
- Single score computed as a weighted sum of the execution times
- Benchmarks to run and weight of each one specified in benchmarks.txt
- Running a subset of relevant CPU benchmarks, curated by the ROOT Team
- GPU benchmarks soon
- Benchmarks maintained by the ROOT team

Implementation (II)

GitLab CI jobs

- Build Docker image containing ROOT
  Build Docker image containing rootbench
- Run container using Docker
- Run container using Singularity

Standalone container

- ROOT dependencies
- rootbench dependencies
- ROOT
- rootbench
- Prefetched files
- hep-analysis-workloads repository

Steps performed in container

- Container entrypoint: /hep-analysis-workloads/root/root/run.sh
  - Read benchmarks.txt file containing benchmarks to run and assigned weights
    - optional: bind-mount a custom benchmarks.txt file at /root/root/benchmark.txt
  - Run benchmarks using rootbench, for example:
    - ctest -R gbenchmark-df102 –VV
  - Parse generated CSV files and generate a JSON summary:
    - names of benchmarks and their durations
    - status of each benchmark (passed/failed)
    - workload score

Results

- Average of 5 runs on a dual-socket 64-core(128 SMT) AMD EPYC 7702 node: 303.87 s => ~ 5 min
- Length of benchmarks tunable by increasing the size of the input data
- Waiting for rootbench:
  - Not all benchmarks available yet
  - MT benchmarks (<50% of the total) are fixed to 8 threads, parameterizable in the future

Useful links

- Project description:
  https://codimd.web.cern.ch/2v70XlE6QWW1lJy6psKCkQ
- Benchmarks compiled by the ROOT team:
  https://codimd.web.cern.ch/fqKN6AJQSeet0senInCIVg
- JIRA Epic:
  https://its.cern.ch/jira/browse/BMK-472
- Relevant JIRA issue:
  https://its.cern.ch/jira/browse/BMK-482
- Analysis workloads repository:
  https://gitlab.cern.ch/xvallspl/hep-analysis-workloads
- rootbench:
  https://github.com/root-project/rootbench

QUESTIONS?

dominika.kankowska@outlook.com

https://www.linkedin.com/in/dkankowska/