

HEPscore Development Update

Chris Hollowell <hollowec@bnl.gov> - Scientific Data and Computing Center (SDCC)
On behalf of the HEPiX Benchmarking Working Group

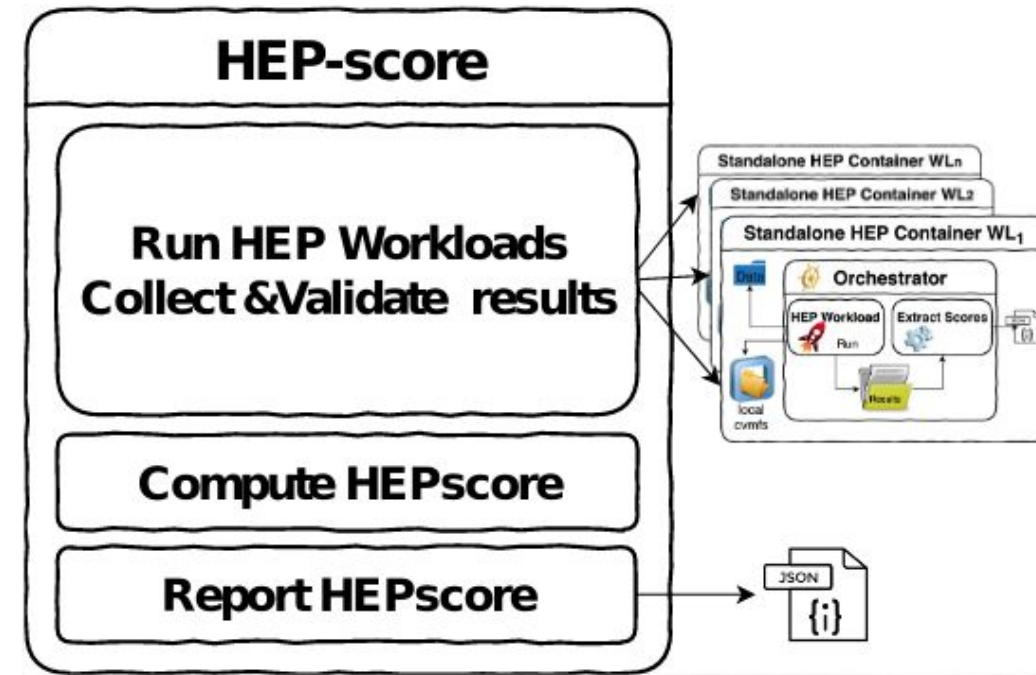
HEPscore Workshop - 9/19/2022



@BrookhavenLab

HEPscore Application

- HEPscore application - [HEP Benchmarks Project](https://gitlab.cern.ch/hep-benchmarks/hep-score) software which orchestrates the execution of multiple benchmark containers and derives a final overall numeric benchmark score
 - Developed in Python
 - Freely available under GPLv3: <https://gitlab.cern.ch/hep-benchmarks/hep-score>
- HEPscore is a standalone application, but is often run as part of the [HEP Benchmark Suite](#)
 - Running with the suite allows for automated reporting/publishing of results via ActiveMQ
- Shares some design elements with HEPspec06
 - Supports geometric mean for overall score calculation
 - Individual benchmarks can be run multiple times, with median score taken, to reduce the effect of outliers



HEPscore Application (Cont.)

- HEPscore benchmark sets are fully configurable via YAML
 - Builtin/default configuration executes the “HEPscoreBeta” benchmark
 - Utilizes CERN’s GitLab Docker registry to download benchmark container images
 - A [configuration](#) which utilizes unpacked containers in CVMFS is also available
 - Currently consists of a number of Run2 workloads
 - Task force deciding on Run3 workload set, parameters, and overall benchmark name (i.e. “HEPscore22”)
- Besides simply printing the overall numeric score, detailed JSON (or YAML, if ‘-y’ is passed) output is also generated
 - Contains information about system environment, configuration and individual sub-benchmark scores

```
48 cms-reco-bmk:
49   results_file: cms-reco_summary.json
50   ref_scores:
51     reco: 2.196
52   weight: 1.0
53   version: v2.1
54   args:
55     threads: 4
56     events: 50
57 lhcb-gen-sim-bmk:
58   results_file: lhcb-gen-sim_summary.json
59   ref_scores:
60     gen-sim: 90.29
61   weight: 1.0
62   version: v2.1
63   args:
64     threads: 1
65     events: 5
66 settings:
67   name: HEPscoreBeta
68   reference_machine: "CPU Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz"
69   registry: docker://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads
70   method: geometric_mean
71   repetitions: 3
72   retries: 1
73   scaling: 355
74   container_exec: singularity
```

HEPscore Execution Example

- Example of installing and running hep-score on RHEL7:

```
$ pip3 install --user git+https://gitlab.cern.ch/hep-benchmarks/hep-score.git
Collecting git+https://gitlab.cern.ch/hep-benchmarks/hep-score.git
  Cloning https://gitlab.cern.ch/hep-benchmarks/hep-score.git to /home/tmp/pip-g4bj963w-build
...
Installing collected packages: pbr, pyyaml, hep-score
  Running setup.py install for hep-score ... done
Successfully installed hep-score-1.4.0 pbr-5.10.0 pyyaml-6.0

$ hep-score /tmp
2022-09-12 14:02:45 hepscore [INFO] HEPscoreBeta Benchmark
2022-09-12 14:02:45 hepscore [INFO] Config Hash:          e13a5f3325137011f79a49cae50ab4ebf48b27bebe7b70fa3d115f7984998a98
2022-09-12 14:02:45 hepscore [INFO] HEPscore version:    1.4.0
2022-09-12 14:02:45 hepscore [INFO] System:              Linux spool0901.sdcc.bnl.gov 3.10.0-1160.71.1.el7.x86_64
2022-09-12 14:02:45 hepscore [INFO] Container Execution: singularity
2022-09-12 14:02:45 hepscore [INFO] Registry:            docker://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads
2022-09-12 14:02:45 hepscore [INFO] Output:              /tmp/HEPscore_12Sep2022_140245
2022-09-12 14:02:45 hepscore [INFO] Date:                Mon Sep 12 14:02:45 2022

2022-09-12 14:02:45 hepscore [INFO] Executing 3 runs of atlas-gen-bmk
2022-09-12 14:02:45 hepscore [INFO] Starting run0
...
2022-09-08 20:25:43 hepscore.gen_score() [INFO] Final result: 878.6845
```

HEPscore Execution Example (Cont.)

```
$ hep-score -h
usage: hep-score [-h] [-m [{singularity,docker}]] [-S] [-c] [-C]
                [-f [CONFFILE]] [-r] [-o [OUTFILE]] [-y] [-p] [-V] [-v]
                [OUTDIR]

-----
HEPscore Benchmark Execution
-----

This utility orchestrates several benchmarks

Additional Information:
  https://gitlab.cern.ch/hep-benchmarks/hep-score
Contact:
  https://wlcg-discourse.web.cern.ch/c/hep-benchmarks

positional arguments:
  OUTDIR                Base output directory.

optional arguments:
  -h, --help            show this help message and exit
  -m [{singularity,docker}], --container_exec [{singularity,docker}]
                        specify container platform for benchmark execution
                        (singularity [default], or docker).
  -S, --usersns        enable user namespace for Singularity, if supported.
  -c, --clean          clean residual container images from system after run.
  -C, --clean_files    clean residual files & directories after execution.
                        Tar results.
  -f [CONFFILE], --conffile [CONFFILE]
                        custom config yaml to use instead of default.
  -r, --replay         replay output using existing results directory OUTDIR.
  -o [OUTFILE], --outfile [OUTFILE]
                        specify summary output file path/name.
  -y, --yaml          create YAML summary output instead of JSON.
  -p, --print          print configuration and exit.
  -V, --version        show program's version number and exit
  -v, --verbose        enables verbose mode. Display debug messages.
```

Changes In The Latest HEPscore Release

- HEPscore v1.4 released on 4/22/2022
- **Renamed default benchmark to HEPscoreBeta**
 - Clarifies the benchmark set/configuration is in a development state and being reviewed by the task force
- **Documented Podman usage**
 - Podman not yet officially supported, but various tests have shown it is usable
- **No longer contain PID namespace with Singularity (MR#134)**
 - Resolves issues running via Singularity in k8s pods
- **Add support for oras:// and https:// registry URIs**
 - Allows the use of SIF images in OCI container registries
 - Reduces initial execution time by eliminating Singularity's automated SIF image build from Docker layers
- **Full release notes:** <https://gitlab.cern.ch/hep-benchmarks/hep-score/-/releases/v1.4>

Ongoing Development Activities

- **Support for multiple weights per container**
 - HEPscore utilizes a weighted geometric mean for overall score calculation, allowing selected individual container scores to have more of an impact on the overall score
 - Currently only supports a single weight per workload container
 - Developing functionality that will allow multiple weights per container for multi-workload containers
- **Support for nested container argument dictionaries in the YAML configuration**
 - One of the new GPU workload containers being developed requires support for nested arguments such as: `--args '--cpu 2'`
- **Implementing the use of `--unsquash` when running workloads with Apptainer**
 - Additional details in the next few slides

Evolving Linux Container Engine Landscape

- HEPscore was designed to work with both Singularity and Docker
- [Podman](#), an open-source Docker-alternative, is shipped as the stock/default container engine with RHEL/Rocky/Alma Linux 8+
 - Provides daemonless execution via OCI runc and user namespaces
 - Highly compatible with Docker CLI: can create docker -> podman symlink for most uses
 - Compared to Singularity, still not heavily utilized in WLCG or at HPC centers
 - But gaining traction in some communities, particularly due its inclusion in RHEL
- In Nov 2021, it was announced that Singularity development was moving to the Linux Foundation and being renamed [Apptainer](#)
 - SingularityCE/Pro from Sylabs also continue to exist
 - Starting with the upcoming Apptainer 1.1.0 release, Apptainer will be the default “singularity” package in EPEL
 - This release will also disable privileged (setuid) execution by default
 - Will utilize user namespace-based execution unless a separate ‘apptainer-suid’ package is also installed



Apptainer Support

- Tested Apptainer 1.1.0rc2 release with HEPscore v1.4
 - Apptainer includes singularity -> apptainer symlink: no need to change execution command
- An issue was found with container execution time during our tests
 - Workloads were taking considerably longer to execute
 - *For example, on a 64-core (total) 2x AMD EPYC 7351-based host, atlas-gen-bmk execution time went from 25 minutes with Singularity, to nearly 3 hours with Apptainer (unprivileged)*
 - Saw significant CPU time being used by a single 'squashfuse' process
 - Caused by move of 1.1.0rc to a FUSE mount of SIF images, instead of unpacking them to a temporary sandbox when run unprivileged
 - squashfuse design is not multithreaded/multiprocess so could not performantly handle many processes/threads performing I/O
 - Likely other multiprocess/multithreaded containers would be affected, not just the HEP workloads containers
 - Apptainer issue reported in Github: <https://github.com/apptainer/apptainer/issues/665>

Apptainer Support (Cont.)

- Developers resolved by including a patched/multithread squashfuse_II binary in the Apptainer packaging
 - Change integrated into the 1.1.0rc3 release
- Also requested a flag to revert to the old Singularity behavior for unprivileged (user namespace) execution where SIF images are unpacked to temporary sandbox
 - Preserving the original behavior would ensure no impact on benchmark runtime/performance
 - Concerns about the stability of a development release of squashfuse_II
 - `--unsquash` flag was also integrated in the 1.1.0rc3 release
- Tests indicate the the use of squashfuse_II or ‘--unsquash’ resolve the execution time issue
 - Apptainer 1.1.0rc3 runtimes and scores within normal variations when compared to Singularity execution
 - However, we are in the process of modifying the HEPscore application to call Apptainer with `--unsquash` to most closely replicate Singularity behavior

Podman Support

- As reported in the Fall 2020 HEPiX, we were able to successfully run HEPscore with Podman
 - subprocess.Popen() call with shell=False used in the hep-score code
 - Therefore setting 'docker' alias was not sufficient
 - Had to create a docker -> podman symlink in /usr/bin
 - Also ran into an issue with the open file descriptor limit - needed to increase
 - With podman, limit appears to apply to the entire container, not per process
 - Additional testing is needed to verify score consistency with Docker

```
$ hep-score -v -m docker /tmp
```

```
...
```

```
2022-09-16 08:46:23 hepscore.run() [INFO] HEPscoreBeta Benchmark
2022-09-16 08:46:23 hepscore.run() [INFO] Config Hash:          5fd1d63488af9d1ad43abb2fececb8ec3cb66c5b25b5307cfccf50465e0ee835
2022-09-16 08:46:23 hepscore.run() [INFO] HEPscore version:    1.4.0
2022-09-16 08:46:23 hepscore.run() [INFO] System:              Linux spool0901.sdcc.bnl.gov 3.10.0-1160.71.1.el7.x86_64
2022-09-16 08:46:23 hepscore.run() [INFO] Container Execution:  docker
2022-09-16 08:46:23 hepscore.run() [INFO] Registry:             docker://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads
2022-09-16 08:46:23 hepscore.run() [INFO] Output:               /tmp/HEPscore_16Sep2022_084623
2022-09-16 08:46:23 hepscore.run() [INFO] Date:                Fri Sep 16 08:46:23 2022
```

```
2022-09-16 08:46:23 hepscore._run_benchmark() [INFO] Executing 3 runs of atlas-gen-bmk
```

```
2022-09-16 08:46:23 hepscore._run_benchmark() [INFO] Starting run0
```

```
2022-09-16 08:46:23 hepscore._run_benchmark() [DEBUG] Running ['docker', 'run', '--rm', '--network=host', '-v',
'/tmp/HEPscore_16Sep2022_084623/atlas-gen-bmk/run0:/results', 'gitlab-registry.cern.ch/hep-benchmarks/hep-workloads/atlas-gen-bmk:v2.1', '-W',
'--threads', '1', '--events', '200']
```

```
...
```

```
2022-09-16 13:23:01 hepscore.gen_score() [INFO] Final result: 871.8371
```

Conclusions

- HEPscore is the HEP Benchmarks Project application which orchestrates the execution of the workload benchmark containers
 - Computes an overall score using weighted geometric mean
 - Benchmark set and parameters highly configurable
 - Current default benchmark is HEPscoreBeta
 - Final benchmark set/configuration being determined by the WLCG HEPscore Task Force
- While fully functional and stable, HEPscore continues to be actively developed
 - A number of new features were added in the recent HEPscore v1.4 release
 - Support for SIF images in OCI registries, Podman documentation, etc.
 - Several additional features planned for the next release
 - Support for Apptainer's `--unsquash` flag, multiple weights per container, etc.
- Over time, Apptainer and Podman will likely replace the use of Singularity and Docker in our community
 - Work has been completed to ensure these will be fully supported in HEPscore