HEP Benchmarks updates and demo

D. Giordano (CERN/IT)

on behalf of
HEPiX CPU Benchmarking WG
hepix-cpu-benchmark@hepix.org

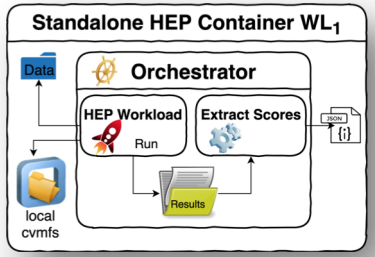
HEPiX Spring 2021 Online workshop
16 March 2021

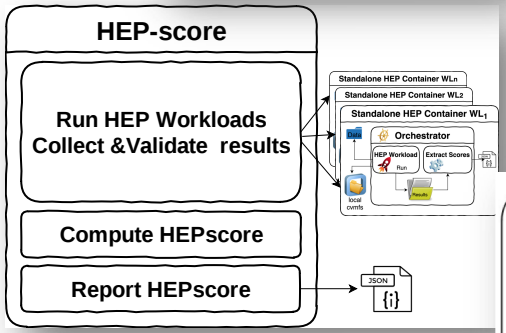
HEP Benchmarks project

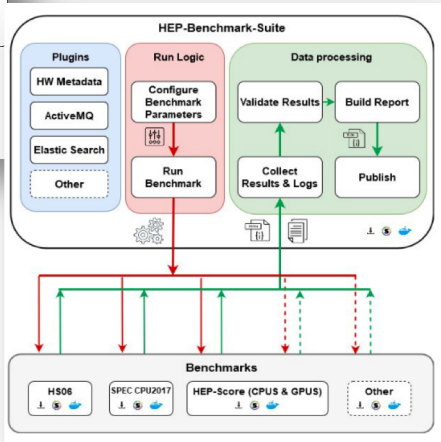
Three main components being developed since ~2 years

!!! Released under GPLv3 licence !!!

- *HEP Workloads*
  - Individual **reference** HEP workloads
  - Common build infrastructure
- *HEP Score*
  - Orchestrate the run of a series of HEP workloads
  - Compute the **HEPscore** value
  - Report whole set of WL results
- *HEP Benchmark Suite*
  - **Meta-orchestrator** of multiple benchmark suites
    - HEPscore, HS06, SPEC CPU2017…







HEP Workloads: Status

- Included new CPU Experiments workloads
  - BelleII (done), Atlas sim MT (in progress)
  - To come: Dune, gravitational waves, WeNMR, …
- First GPU workload containerized
  - SimpleTrack (LHC simulation)
  - Multi-GPU container workloads (Nvidia, AMD, Intel...)
- Plan:
  - Include the Run-3 CPU workloads proposed in the WLCG HEP Score Task Force
    - Support multiple architectures (e.g. ARM) as long as Experiments' software has been ported
  - Integrate other GPU workloads: next CMS Patatrack (HLT Track reco), MC Madgraph

Summary of currently supported HEP workloads

Experiment	Name	Description	Experiment license	Latest Container	Readiness	Pipeline status
Alice	gen-sim	link	GNU GPL v3	docker	w.i.p.	pipeline passed
Atlas	gen	link	Apache v2	docker	Y	pipeline passed
Atlas	sim	link	Apache v2	docker	Y	pipeline passed
Atlas	digi-reco	link	Apache v2	docker	w.i.p.	pipeline passed
CMS	gen-sim	link	Apache v2	docker	Y	pipeline passed
CMS	digi	link	Apache v2	docker	Y	pipeline passed
CMS	reco	link	Apache v2	docker	Y	pipeline passed
LHCb	gen-sim	link	GNU GPL v3	docker	Y	pipeline passed
Belle2	gen-sim-reco	link	GNU GPL v3	docker	Y	pipeline passed

https://gitlab.cern.ch/hep-benchmarks/hep-workloads

HEP Score v1.0 : Status

- Several new features included in this new release
  - **Singularity** and **Docker** engines are both supported, forced user namespace too
  - Access of **cvmfs unpacked** images
  - Better handling of disk space, configurable cleanup of the working directory
  - Optimised the report structure, allows retries in case of run failures
  - Improved CI tests
  - Configurable **weighted geometric** mean for the HEP workloads
  - Python wheels available: useful for installations in sites with limited external connectivity
- To install & Run: documentation at https://gitlab.cern.ch/hep-benchmarks/hep-score
- Main developers: C. Hollowell, C. van der Laan, D. Southwick

HEPscore2X: the configuration

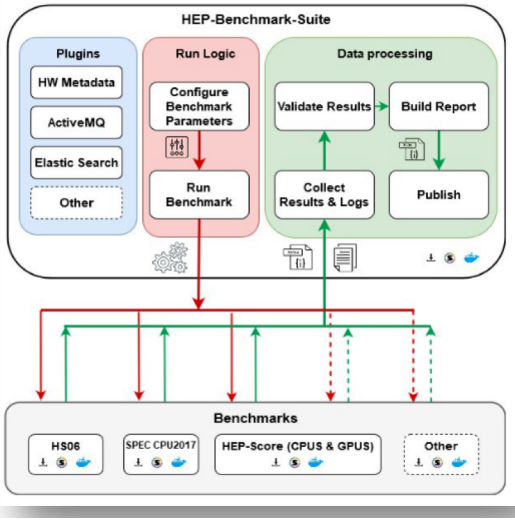
- **HEP Score tool** is configurable
  - Config: list of workloads to run, reference scores & weights, settings
- HEP Score pkg distributed with a default config file: **HEPscore2X**
  - Includes the most stable workloads tested so far on CPUs up to 256 cores
  - "2X" to be replaced by the year of future adoption
- The official config will be defined by the WLCG HEP Score Task Force
  - After inclusion of Run3 workloads and performance study
- NB: Other config files can be used and passed to the tool
  - Convenient to include new workloads and perform studies
  - Each config is associated to an unique ID in the final report

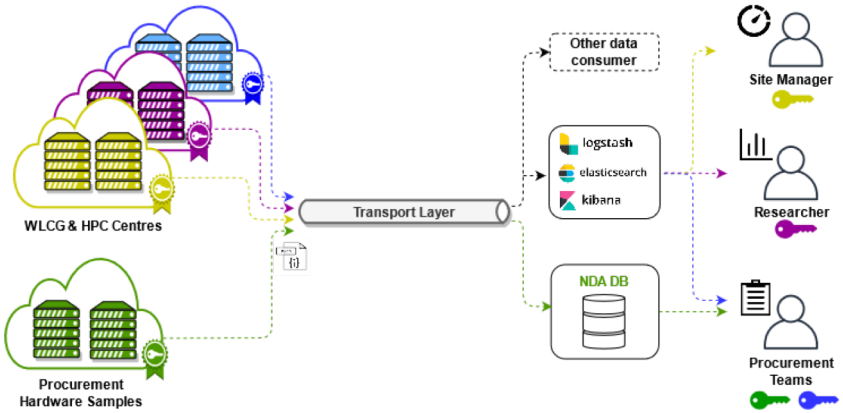
hepscore2X_0.8.yaml 1.45 KB

```yaml
hepscore_benchmark:
  benchmarks:
    atlas-gen-bmk:
      results_file: atlas-gen_summary.json
      ref_scores:
        gen: 384
      weight: 1.0
      version: v2.1
      args:
        threads: 1
        events: 200
    belle2-gen-sim-reco-bmk:
      results_file: belle2-gen-sim-reco_summary.json
      ref_scores:
        gen-sim-reco: 5.44
      weight: 1.0
      version: v2.2
      args:
        threads: 1
        events: 50
    cms-gen-sim-bmk:
      results_file: cms-gen-sim_summary.json
      ref_scores:
        gen-sim: 0.726
      weight: 1.0
      version: v2.1
      args:
        threads: 4
        events: 20
    cms-digi-bmk:
      results_file: cms-digi_summary.json
      ref_scores:
        digi: 3.58
      weight: 1.0
      version: v2.1
      args:
        threads: 4
        events: 50
    cms-reco-bmk:
      results_file: cms-reco_summary.json
      ref_scores:
        reco: 2.196
      weight: 1.0
      version: v2.1
      args:
        threads: 4
        events: 50
    lhcb-gen-sim-bmk:
      results_file: lhcb-gen-sim_summary.json
      ref_scores:
        gen-sim: 90.29
      weight: 1.0
      version: v2.1
      args:
        threads: 1
        events: 5
  settings:
    name: HEPscore2X
    reference_machine: "CPU Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz"
    registry: docker://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads
    method: geometric_mean
    repetitions: 3
    retries: 1
    scaling: 355
    container_exec: singularity
```

HEP Benchmark Suite v2.0

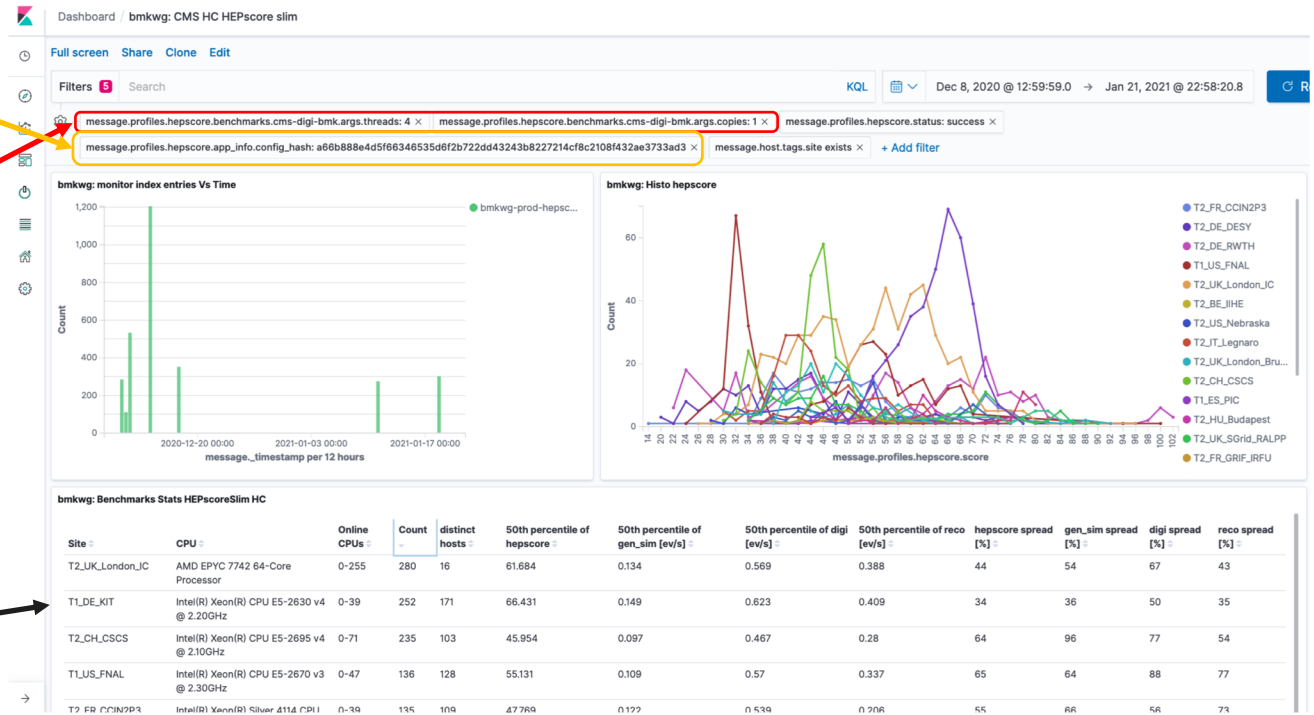
- Meta-orchestrator for the execution of several benchmarks and the publication of the suite's report
  - HS06, HEP Score, SPEC CPU 2017, …
- Features of this new version
  - Modular design, fully rewritten in python3.6+
  - Distributed via pip install, python wheels available
  - Metadata section with detailed HW information
  - Install as unprivileged user
    - HPC compatible: example of SLURM submission
    - Run also on Grid pilot jobs: example
      - see next slide
- Main developers: M. Fontes Medeiros, D. Southwick





HEPscore “slim” config on grid sites

- A “slim” version of the **HEPscore2X** config, including **only** CMS gen-sim, digi, reco
- Running on 4-cores job slots
- User job submission, glide-in singularity pilot runs
  - Requires singularity-in-singularity (i.e. user namespaces) enabled on the grid site
- Results collected in central DB (Elasticserach) and monitored via Kibana dashboards
- Enables sampling of grid nodes’ real performance



HS06 and SPEC CPU 2017

- Make sure that HS06 and SPEC CPU 2017 can run via the Suite
  - Orchestrator scripts and libraries available in a container image, built at https://gitlab.cern.ch/hep-benchmarks/hep-spec/
  - NB: the SPEC suites are not included for license reasons, have to be pre-installed on the host
- HS06
  - Config is linux_gcc_cern.cfg used by HS06 in the last decade, with few adaptations
- SPEC CPU 2017
  - Default “HEP” benchmark set to mimic HS06: Cpp-rate set of benchmarks
    - Benchmark set can be reconfigured. Eg. –b intrate will run SPEC INT 2017 rate
    - NB: All configuration changes are tracked in the reported results
  - Config similar to linux_gcc_cern.cfg, distinct for x86 and ARM

HS06 for ARM CPUs

```
We do not appear to have working vendor-supplied binaries for your
architecture.  You will have to compile the tool binaries by
yourself.  Please read the file
   SPEC_CPU2006_v1.2/Docs/tools-build.html
for instructions on how you might be able to build them.
Please only attempt this as a last resort.
```

- Enable support for ARM
  - Multi-architecture container
    - gitlab-registry.cern.ch/hep-benchmarks/hep-spec/hepspec-cc7-multiarch:v2.0
  - SPEC CPU 2017 already supports ARM cpus. Only CPU model needs to be changed when running on ARM
  - HS06 too old to support natively ARM cpus: SPEC 2006 tooklit needed to be built to work
    - Build the toolkit following instructions https://www.spec.org/cpu2006/Docs/tools-build.html
      after patching some old code
      - Patch procedure available https://gitlab.cern.ch/hep-benchmarks/hep-spec/-/tree/master/patch_SPEC2006

- NB: patching the toolkit is one time operation
  - The toolkit is then included in the *tool/bin* area
    Re-creating an archive for SPEC CPU 2006 allows to use it in any other aarch64 machine

- NB: HS06 for ARM only supported at **64 bits**

Running HS06 on ARM

AWS Graviton2 bare-metal server benchmarked using the hepspec multi-architecture container (see previous slide)

```
$ lscpu
Architecture:                    aarch64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
CPU(s):                          64
On-line CPU(s) list:             0-63
Thread(s) per core:              1
Core(s) per socket:              64
Socket(s):                       1
NUMA node(s):                    1
Vendor ID:                       ARM
Model:                           1
Model name:                      Neoverse-N1
Stepping:                        r3p1
BogoMIPS:                        243.75
L1d cache:                       4 MiB
L1i cache:                       4 MiB
L2 cache:                        64 MiB
L3 cache:                        32 MiB
NUMA node0 CPU(s):               0-63
```

```
$ echo ${SECRET_URL} | /hep-spec/scripts/hep-spec.sh -w $BMK_RUNDIR -b $BMK -m $BMK_OPTION -p ${SPEC_DIR} -u -
############################
         CERN HEPSPEC
Sat Feb 27 17:16:00 UTC 2021
############################
2021-02-27T17:16:00 [/hep-spec/scripts/hep-spec.sh] Variable values:
       HEPSPEC_SOURCEDIR=/hep-spec/scripts
       HEPSPEC_BMK=hs06
       HEPSPEC_NUMPROC=64
       HEPSPEC_PATH=/scratch/HEPSPEC/CI_hs06_ext_g2_bare_12384842
       HEPSPEC_SET=all_cpp
       HEPSPEC_MACHINE_OPTION=default
       HEPSPEC_ITERATIONS=3
       HEPSPEC_WORKDIR=/scratch/jobs/hs06_ext_g2_bare_12384842/hep-spec
       HEPSPEC_DEBUG=0
```


```
{"hs06":{"start":"Sat Feb 27 17:19:26 UTC 2021", "end":"Sat Feb 27 20:10:46 UTC 2021", "copies":64,
            "runcpu_args":"1 runspec: runspec --define machine_option:64 --config=linux_gcc_cern.cfg --action=build all_cpp;64 runs
pec: runspec --define machine_option:64 --config=linux_gcc_cern.cfg --nobuild --noreportable --iterations=3 all_cpp", "bset":"all
_cpp", "LINK":"  6  g++ -O2 -fPIC -pthread  -DSPEC_CPU_LP64    <objects>     -o options;  1  g++ -O2 -fPIC -pthread  -DSPEC_CPU_LP
64 -DSPEC_CPU_LINUX   <objects>     -o options;", "hash":"7b84bb375cee11731a958a26d6fc155d",
          "score":1170.998, "avg_core_score" : 18.296, "num_bmks":7 ,"bmks":{ "444.namd":[  23.5, 23.5, 23.5, 23.5, 23.5, 23.5, 2
```

Demo

- Login to newly created VM
- Run HEP Score example
  - Install singularity, python3-pip, wheels of HEP Benchmark Suite
  - Run and collect results in Elasticsearch
- Run HS06 example
  - install singularity, python3-pip, git & HEP Benchmark Suite
  - Run and collect results in Elasticsearch

Demo: install

- Grid certificate to authenticate to the AMQ broker

```
[root@bmk16-cc7-ad4orebalr ~]# ls
anaconda-ks.cfg  original-ks.cfg  usercert.pem  userkey.pem
[root@bmk16-cc7-ad4orebalr ~]#
```

- Install singularity and python3-pip

```
[root@bmk16-cc7-ad4orebalr ~]# yum install -y singularity python3-pip
Loaded plugins: changelog, fastestmirror, kernel-module, ovl, protectbase, tsflags,
              : versionlock
Loading mirror speeds from cached hostfile
214 packages excluded due to repository protections
```

- Download example script

```
[root@bmk16-cc7-ad4orebalr ~]# curl -O https://gitlab.cern.ch/hep-benchmarks/hep-benchmark-s
uite/-/raw/master/examples/hepscore/run_HEPscore_default_from_wheels.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1957  100  1957    0     0  10166      0 --:--:-- --:--:-- --:--:-- 10246
[root@bmk16-cc7-ad4orebalr ~]# vi run_HEPscore_default_from_wheels.sh
```

Demo: edit

- Edit the script
  - Change site name
  - User certificate path, for data publication
  - Enable publishing to AMQ broker
  - Add a custom tag (not mandatory)

```
#-----------------------------------------------
# Replace somesite with a meaningful site name
SITE=CERN
#-----------------------------------------------


echo "Running script: $0"
cd $( dirname $0)

WORKDIR=`pwd`/workdir

mkdir -p $WORKDIR
chmod a+rw -R $WORKDIR

cat > $WORKDIR/bmkrun_config.yml <<EOF2
activemq:
  server: dashb-mb.cern.ch
  topic: /topic/vm.spec
  port: 61123  # Port used for certificate
  ## include the certificate full path (see documentation)
  key: '/root/userkey.pem'
  cert: '/root/usercert.pem'

global:
  benchmarks:
  - hepscore
  mode: singularity
  publish: true
  rundir: ./suite_results
  show: true
  tags:
    site: $SITE
    description: "HEPiX_Demo"

hepscore:
  version: v1.0
  config: default
  options:
      userns: True
      clean: True
EOF2

cd $WORKDIR
export MYENV="env_bmk"        # Define the name of the environment.
python3 -m venv $MYENV        # Create a directory with the virtual environment.
source $MYENV/bin/activate    # Activate the environment.
wheels_version=hep-benchmark-suite-wheels-v2.0.tar
curl -O https://hep-benchmarks.web.cern.ch/hep-benchmark-suite/releases/${wheels_version}
tar xvf ${wheels_version}
python3 -m pip install suite_wheels/*.whl
cat bmkrun_config.yml
bmkrun -c bmkrun_config.yml
```

Demo: run (1)

- Downloads wheels
- Install packages
- …

```
[root@bmk16-cc7-ad4orebalr ~]# ./run_HEPscore_default_from_wheels.sh
Running script: ./run_HEPscore_default_from_wheels.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1480k  100 1480k    0     0   982k      0  0:00:01  0:00:01 --:--:--  982k
suite_wheels/
suite_wheels/pbr-5.5.1-py2.py3-none-any.whl
suite_wheels/typing_extensions-3.7.4.3-py3-none-any.whl
suite_wheels/importlib_metadata-3.7.2-py3-none-any.whl
suite_wheels/stomp.py-6.1.0-py3-none-any.whl
suite_wheels/PyYAML-5.4.1-cp36-cp36m-manylinux1_x86_64.whl
suite_wheels/zipp-3.4.1-py3-none-any.whl
suite_wheels/hep_benchmark_suite-2.0-py3-none-any.whl
suite_wheels/hep_score-1.0.0-py3-none-any.whl
suite_wheels/setuptools-44.1.1-py2.py3-none-any.whl
suite_wheels/docopt-0.6.2-py2.py3-none-any.whl
Processing ./suite_wheels/PyYAML-5.4.1-cp36-cp36m-manylinux1_x86_64.whl
Processing ./suite_wheels/docopt-0.6.2-py2.py3-none-any.whl
Processing ./suite_wheels/hep_benchmark_suite-2.0-py3-none-any.whl
Processing ./suite_wheels/hep_score-1.0.0-py3-none-any.whl
Processing ./suite_wheels/importlib_metadata-3.7.2-py3-none-any.whl
Processing ./suite_wheels/pbr-5.5.1-py2.py3-none-any.whl
Processing ./suite_wheels/setuptools-44.1.1-py2.py3-none-any.whl
Processing ./suite_wheels/stomp.py-6.1.0-py3-none-any.whl
Processing ./suite_wheels/typing_extensions-3.7.4.3-py3-none-any.whl
Processing ./suite_wheels/zipp-3.4.1-py3-none-any.whl
Installing collected packages: PyYAML, docopt, typing-extensions, zipp, importlib-metadata, stomp.py, hep-benchmark-suite, setuptools, pbr,
 hep-score
  Found existing installation: setuptools 39.2.0
    Uninstalling setuptools-39.2.0:
      Successfully uninstalled setuptools-39.2.0
Successfully installed PyYAML-5.4.1 docopt-0.6.2 hep-benchmark-suite-2.0 hep-score-1.0.0 importlib-metadata-3.7.2 pbr-5.5.1 setuptools-44.1
.1 stomp.py-6.1.0 typing-extensions-3.7.4.3 zipp-3.4.1
```

Demo: run (2)

- Downloads wheels
- Install packages
- Execute the Suite (bmkrun), reading the config bmkrun_config.yml
- Perform checks
  - Singularity version, disk space, HEP Score version
- Run HEP Score
- …

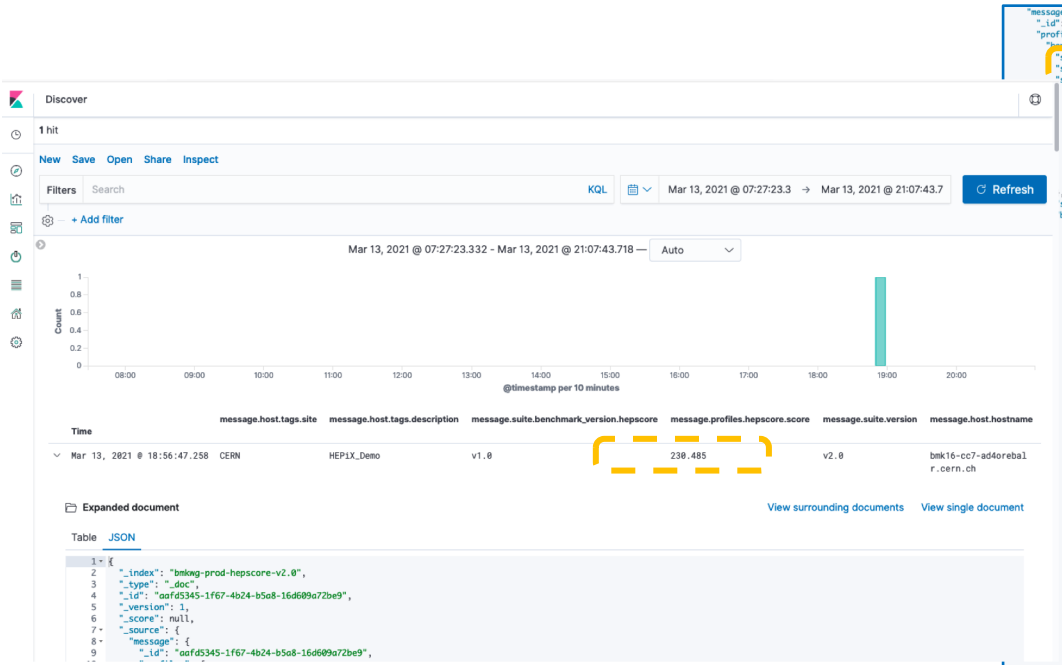
```
# The following configuration was loaded: bmkrun_config.yml
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:start [INFO] Starting HEP Benchmark Suite
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:preflight [INFO] Running pre-flight checks
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:preflight [INFO]  - Checking if selected run mode exists...
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:preflight [INFO]    - singularity executable found: /usr/bin/singularity.
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:preflight [INFO]    - singularity version: 3.7.1-1.el7.
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:preflight [INFO]  - Checking provided work dirs exist...
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:preflight [INFO]  - Checking for a valid configuration...
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:preflight [INFO]  - Checking if rundir has enough space...
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:start [INFO] Pre-flight checks passed successfully.
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:run [INFO] Benchmarks left to run: ['hepscore']
2021-03-13 15:06:34, hepbenchmarksuite.hepbenchmarksuite:run [INFO] Running benchmark: hepscore
2021-03-13 15:06:34, hepbenchmarksuite.benchmarks:prep_hepscore [INFO] Checking if hep-score is installed.
2021-03-13 15:06:34, hepbenchmarksuite.benchmarks:prep_hepscore [INFO] Found existing installation of hep-score in the system: v1.0.0
2021-03-13 15:06:34, hepbenchmarksuite.benchmarks:prep_hepscore [INFO] Installation matches requested version in the config file: v1.0
2021-03-13 15:06:34, hepbenchmarksuite.benchmarks:run_hepscore [INFO] Attempting to import hepscore
2021-03-13 15:06:34, hepbenchmarksuite.benchmarks:run_hepscore [INFO] Successfully imported hepscore
2021-03-13 15:06:34, hepbenchmarksuite.benchmarks:run_hepscore [INFO] Using default config provided by hepscore.
2021-03-13 15:06:34, hepbenchmarksuite.benchmarks:run_hepscore [INFO] Starting hepscore
2021-03-13 15:06:34, hepscore.hepscore:run [INFO] HEPscore2X Benchmark
2021-03-13 15:06:34, hepscore.hepscore:run [INFO] Config Hash:         430b0c7e62755bffa1495348e21e4c43e54a22067ef349acdcd593fb55b8dbb2
2021-03-13 15:06:34, hepscore.hepscore:run [INFO] System:              Linux bmk16-cc7-ad4orebalr.cern.ch 3.10.0-1160.15.2.el7.x86_64 #1 SM
P Wed Feb 3 15:06:38 UTC 2021 x86_64
2021-03-13 15:06:34, hepscore.hepscore:run [INFO] Container Execution: singularity
2021-03-13 15:06:34, hepscore.hepscore:run [INFO] Registry:            docker://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads
2021-03-13 15:06:34, hepscore.hepscore:run [INFO] Output:              ./suite_results/run_2021-03-13_1406/HEPSCORE
2021-03-13 15:06:34, hepscore.hepscore:run [INFO] Date:                Sat Mar 13 15:06:34 2021

2021-03-13 15:06:34, hepscore.hepscore:_run_benchmark [INFO] Executing 3 runs of atlas-gen-bmk
2021-03-13 15:06:34, hepscore.hepscore:_run_benchmark [INFO] Creating singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 15:06:34, hepscore.hepscore:_run_benchmark [INFO] Starting run0
```

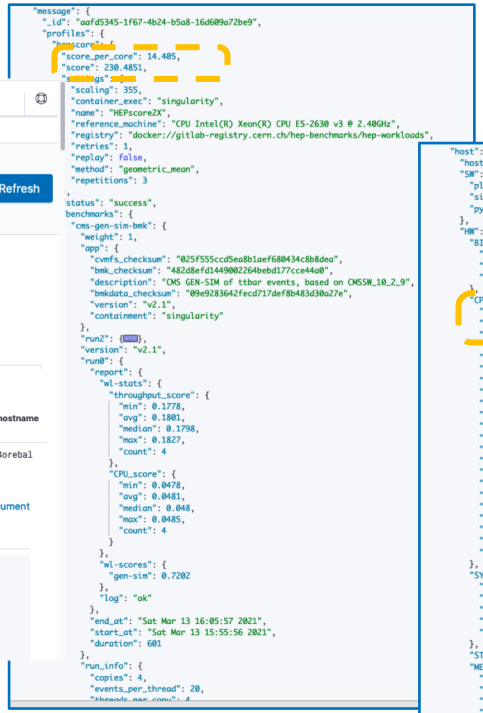
Demo: run (3)

- Downloads wheels
- Install packages
- Execute the Suite (bmkrun), reading the config bmkrun_config.yml
- Perform checks
  - Singularity version, disk space, HEP Score version
- Run HEP Score
  - Notice the execution of the individual workloads
  - Overall duration: 3h50’ for 3 runs of each workload
- Retrieve metadata
- …

```
2021-03-13 15:06:34, hepscore.hepscore:_run_benchmark [INFO] Executing 3 runs of atlas-gen-bmk
2021-03-13 15:06:34, hepscore.hepscore:_run_benchmark [INFO] Creating singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 15:06:34, hepscore.hepscore:_run_benchmark [INFO] Starting run0
2021-03-13 15:15:35, hepscore.hepscore:_run_benchmark [INFO] Starting run1
2021-03-13 15:23:43, hepscore.hepscore:_run_benchmark [INFO] Starting run2
2021-03-13 15:31:50, hepscore.hepscore:_container_rm [INFO] Removing temporary singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 15:31:50, hepscore.hepscore:_run_benchmark [INFO]
2021-03-13 15:31:50, hepscore.hepscore:_run_benchmark [INFO] Executing 3 runs of belle2-gen-sim-reco-bmk
2021-03-13 15:31:50, hepscore.hepscore:_run_benchmark [INFO] Creating singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 15:31:50, hepscore.hepscore:_run_benchmark [INFO] Starting run0
2021-03-13 15:40:56, hepscore.hepscore:_run_benchmark [INFO] Starting run1
2021-03-13 15:48:26, hepscore.hepscore:_run_benchmark [INFO] Starting run2
2021-03-13 15:55:55, hepscore.hepscore:_container_rm [INFO] Removing temporary singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 15:55:56, hepscore.hepscore:_run_benchmark [INFO]
2021-03-13 15:55:56, hepscore.hepscore:_run_benchmark [INFO] Executing 3 runs of cms-gen-sim-bmk
2021-03-13 15:55:56, hepscore.hepscore:_run_benchmark [INFO] Creating singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 15:55:56, hepscore.hepscore:_run_benchmark [INFO] Starting run0
2021-03-13 16:05:57, hepscore.hepscore:_run_benchmark [INFO] Starting run1
2021-03-13 16:13:42, hepscore.hepscore:_run_benchmark [INFO] Starting run2
2021-03-13 16:21:30, hepscore.hepscore:_container_rm [INFO] Removing temporary singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 16:21:31, hepscore.hepscore:_run_benchmark [INFO]
2021-03-13 16:21:31, hepscore.hepscore:_run_benchmark [INFO] Executing 3 runs of cms-digi-bmk
2021-03-13 16:21:31, hepscore.hepscore:_run_benchmark [INFO] Creating singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 16:21:31, hepscore.hepscore:_run_benchmark [INFO] Starting run0
2021-03-13 16:32:29, hepscore.hepscore:_run_benchmark [INFO] Starting run1
2021-03-13 16:38:29, hepscore.hepscore:_run_benchmark [INFO] Starting run2
2021-03-13 16:44:29, hepscore.hepscore:_container_rm [INFO] Removing temporary singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 16:44:30, hepscore.hepscore:_run_benchmark [INFO]
2021-03-13 16:44:30, hepscore.hepscore:_run_benchmark [INFO] Executing 3 runs of cms-reco-bmk
2021-03-13 16:44:30, hepscore.hepscore:_run_benchmark [INFO] Creating singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 16:44:30, hepscore.hepscore:_run_benchmark [INFO] Starting run0
2021-03-13 17:04:12, hepscore.hepscore:_run_benchmark [INFO] Starting run1
2021-03-13 17:20:19, hepscore.hepscore:_run_benchmark [INFO] Starting run2
2021-03-13 17:36:23, hepscore.hepscore:_container_rm [INFO] Removing temporary singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 17:36:24, hepscore.hepscore:_run_benchmark [INFO]
2021-03-13 17:36:24, hepscore.hepscore:_run_benchmark [INFO] Executing 3 runs of lhcb-gen-sim-bmk
2021-03-13 17:36:24, hepscore.hepscore:_run_benchmark [INFO] Creating singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 17:36:24, hepscore.hepscore:_run_benchmark [INFO] Starting run0
2021-03-13 18:04:10, hepscore.hepscore:_run_benchmark [INFO] Starting run1
2021-03-13 18:30:27, hepscore.hepscore:_run_benchmark [INFO] Starting run2
2021-03-13 18:56:41, hepscore.hepscore:_container_rm [INFO] Removing temporary singularity cache ./suite_results/run_2021-03-13_1406/HEPSCORE/scache
2021-03-13 18:56:41, hepscore.hepscore:_run_benchmark [INFO]
2021-03-13 18:56:41, hepscore.hepscore:gen_score [INFO] Final result: 230.4851
2021-03-13 18:56:41, hepbenchmarksuite.hepbenchmarksuite:run [INFO] Completed hepscore with return code 0
2021-03-13 18:56:41, hepbenchmarksuite.plugins.extractor:collect_sw [INFO] Collecting SW information.
2021-03-13 18:56:41, hepbenchmarksuite.plugins.extractor:collect_hw [INFO] Collecting HW information.
2021-03-13 18:56:41, hepbenchmarksuite.plugins.extractor:collect_cpu [INFO] Collecting CPU information.
2021-03-13 18:56:41, hepbenchmarksuite.plugins.extractor:collect_bios [INFO] Collecting BIOS information.
2021-03-13 18:56:41, hepbenchmarksuite.plugins.extractor:collect_system [INFO] Collecting system information.
2021-03-13 18:56:41, hepbenchmarksuite.plugins.extractor:collect_memory [INFO] Collecting system memory.
2021-03-13 18:56:41, hepbenchmarksuite.plugins.extractor:collect_storage [INFO] Collecting system storage.
2021-03-13 18:56:42, hepbenchmarksuite.hepbenchmarksuite:cleanup [INFO] Reading result file: ./suite_results/run_2021-03-13_1406/HEPSCORE/hepscore_result.json
2021-03-13 18:56:42, hepbenchmarksuite.hepbenchmarksuite:cleanup [INFO] Successfully completed all requested benchmarks
```

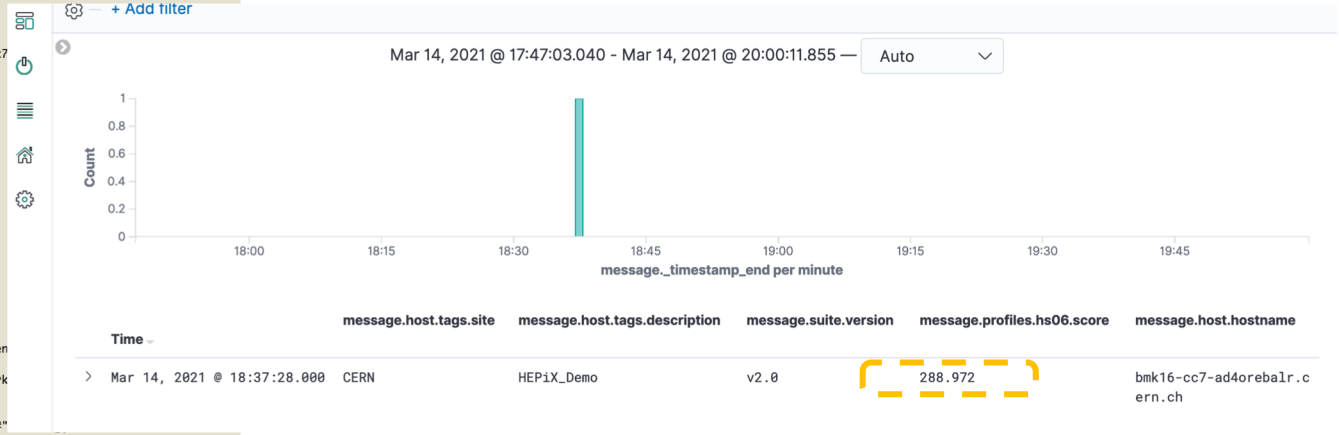
Demo: results in the central DB @ CERN





Demo: run HS06

```
cat > $WORKDIR/bmkrun_config.yml <<EOF2
activemq:
  server: dashb-mb.cern.ch
  topic: /topic/vm.spec
  port: 61123  # Port used for certificate
  ## include the certificate full path (see documentation)
  key: '/root/userkey.pem'
  cert: '/root/usercert.pem'

global:
  benchmarks:
  - hs06
  mode: singularity
  publish: true
  rundir: ${WORKDIR}/suite_results
  show: true
  tags:
    site: CERN
    description: "HEPiX_Demo"

hepspec06:
  # Use the docker registry
  image: "docker://gitlab-registry.cern.ch/hep-benchmarks/hep-spec/hepspec-cc7
  # URL to fetch the hepspec06. It will only be used if the software
  # is  not found under hepspec_volume.
  url_tarball: [redacted]

  # Define the location on where hepspec06 should be found
  # If hepspec06 is not present, the directory should be writeable
  # to allow the installation via the url_tarball
  hepspec_volume: "/tmp/SPEC"

  ## Number of iterations to run the benchmark
  iterations: 3
  ## Specifies if benchmark is run on 32 or 64 bit mode
  ## Default is 64-bit
  mode: 32
EOF2

cd $WORKDIR
export MYENV="env_bmk"        # Define the name of the environment.
python3 -m venv $MYENV        # Create a directory with the virtual environmen
source $MYENV/bin/activate    # Activate the environment.
python3 -m pip install git+https://gitlab.cern.ch/hep-benchmarks/hep-benchmark
cat bmkrun_config.yml

if [ `cat bmkrun_config.yml | grep "this_is_dummy_replace_me" | grep -c -v "#"`
then
  echo -e "\nERROR. You are using the url_tarball parameter. Please replace the dummy url with a real one"
  exit 1
fi
bmkrun -c bmkrun_config.yml
```



- Similarly to run HEP Score
  - Download the example
  - Modify few parameters & run
- SPEC CPU 2017 is similar (read the doc)

Credits

- Collective effort of several member of the HEPiX Benchmarking WG
- CHEP21 abstract

HEPiX benchmarking solution for WLCG computing resources

Miguel F. Medeiros[1,*], *Manfred* Alef[2], *Luca* Atzori[1], *Jean-Michel* Barbet[3], *Ingvild* Brevik Høgstøyl[4], *Olga* Datskova[1], *Riccardo* De Maria[1], *Domenico* Giordano[1], *Maria* Girone[1], *Christopher* Hollowell[5], *Michele* Michelotto[6], *Andrea* Sciabà[1], *Tristan* Sullivan[7], *Randal* Sobie[7], *David* Southwick[1,8], and *Andrea* Valassi[1]
from HEPiX Benchmarking Working Group

[1]CERN, Geneva, Switzerland
[2]KIT, Karlsruhe, Germany
[3]Laboratoire SUBATECH, CNRS-IN2P3, Nantes, France
[4]Norwegian University of Science and Technology, Norway
[5]Brookhaven National Laboratory, USA
[6]INFN, Istituto Nazionale di Fisica Nucleare, Padova, Italy
[7]University of Victoria, Canada
[8]University of Iowa, USA

Abstract. The HEPiX Benchmarking Working Group has been developing a

Conclusions

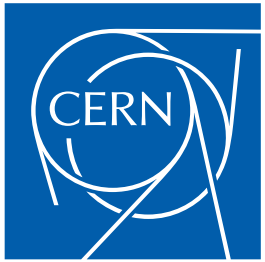
- HEP Benchmark Suite and HEP Score are ready to be tested by our community
  - New HEP Workloads (mainly LHC Run3) will be made available during 2021
- Need volunteer sites to run the Suite and benchmark several CPU models
  - This will permit the studies recommended by the WLCG HEP Score Task Force
- Feedback and support questions are welcome in the HEP Benchmarks Project Discourse Forum

Useful links

Recent publication (CHEP 2019)

Project repository

Documentation: HEP Score, HEP Benchmark Suite


CERN

Benchmark comparing “speed factors”

- In order to compare servers **HS06** and **HEP-Score** implement the **geometric mean** approach. Needs:
  - a set of reference workloads (WLs)
  - a measure of performance per WL (m_i), that typically goes as [1/s] (eg. can be the event throughput)
  - a reference machine

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \cdots x_n}$$

https://en.wikipedia.org/wiki/Geometric_mean

- The score S of a server (**srv**) is defined as the **geometric mean** of the **speed factors** x_i(**srv**,ref) = m_i(**srv**)/m_i(**ref**) respect to the reference machine (**ref**)
  - i.e. “speed” is *normalised* respect to the reference machine “speed”

- The relative score between srv_A and srv_B is the ratio of the scores S(srv,ref) , this is still a geometric mean of speed factors

	WL_1		WL_2		WL_n		Score	S(A,B)
Ref. Srv 	m_1(ref)	1 (by def)	m_2(ref)	1 (by def)	m_n(ref)	1 (by def)	$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$	
Srv A 	m_1(A)	x_1(A,ref)	m_2(A)	x_2(A,ref)	m_n(A)	x_n(A,ref)	S(A,ref)	$\frac{S(A,ref)}{S(B,ref)}$
Srv B 	m_1(B)	x_1(B,ref)	m_2(B)	x_2(B,ref)	m_n(B)	x_n(B,ref)	S(B,ref)	

"File:201912 Rack-optimised servers.svg" by DataBase Center for Life Science (DBCLS) is licensed under CC BY 4.0