CERN openlab

Benchmarking in HPC

Extending HEP Benchmark Suite

David Southwick, Maria Girone IT-DI-OPL

In collaboration with

WLCG Benchmarking Working Group

CERN, SKAO, GÉANT and PRACE Collaboration

EGI-ACE

9/3/2021

HEP Benchmark Suite

A short history

HEP Benchmarking Suite: A benchmark orchestrator & reporting tool.

Provides an array of benchmarks, including HEPscore – the proposed solution for diverging HEPspec06 scores (over 15+ years use, EOL now)

- Designed for WLCG compute environment
- Intended for procurement teams, site administrators
- First with VM containment, later nested docker images

None of these approaches are compatible with HPC!

- Collaboration with WLCG Benchmarking Group to refactor & re-tool for HPC execution at scale!
- Enables R&D benchmarking, such as ongoing ROOT RNTuple HPC optimization

Components

HEP Benchmark Suite 2.0: Now with 100% more HPC!

Minimal Dependencies
Python3 + container choice

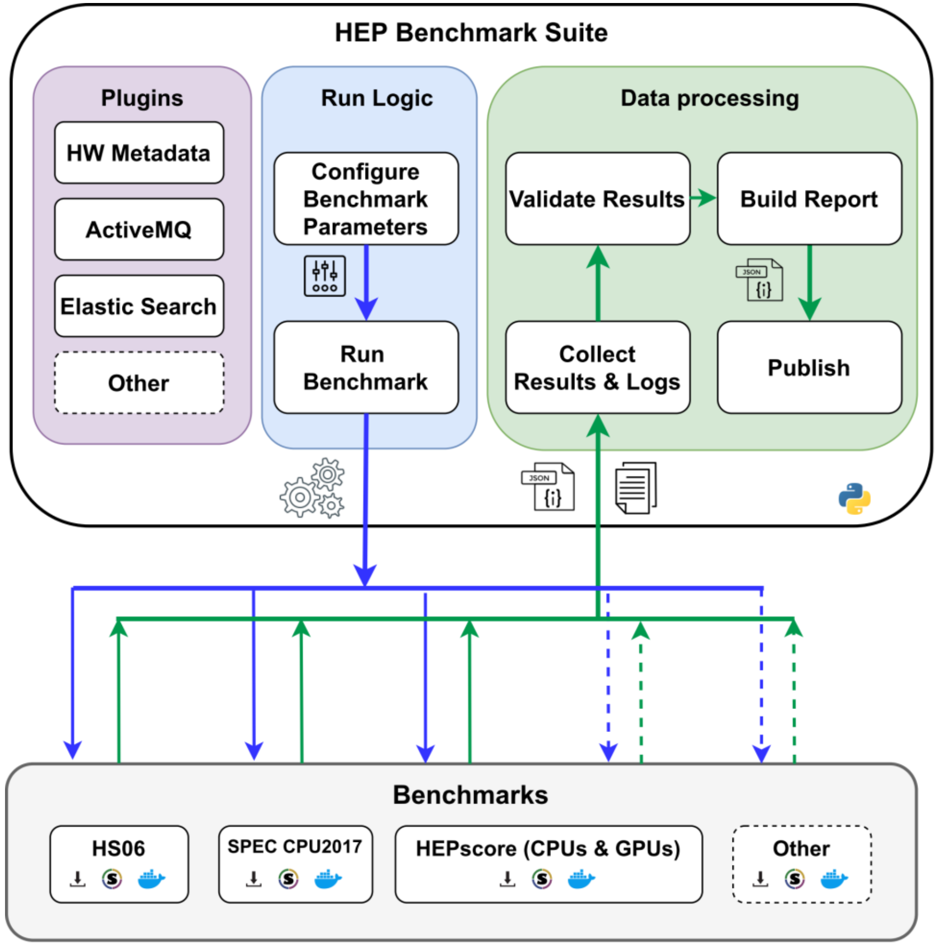
Modular Design
Snap-in workloads & modules

Repeatable & Verifiable
Declarative YAML config

Designed for Ease-of-Use
Simple integration with any job scheduler

Variety of containment choices
Singularity (incl. CVMFS Unpacked), Docker, Podman**soon

Metadata + Analytics
Automated Reporting via AMQ

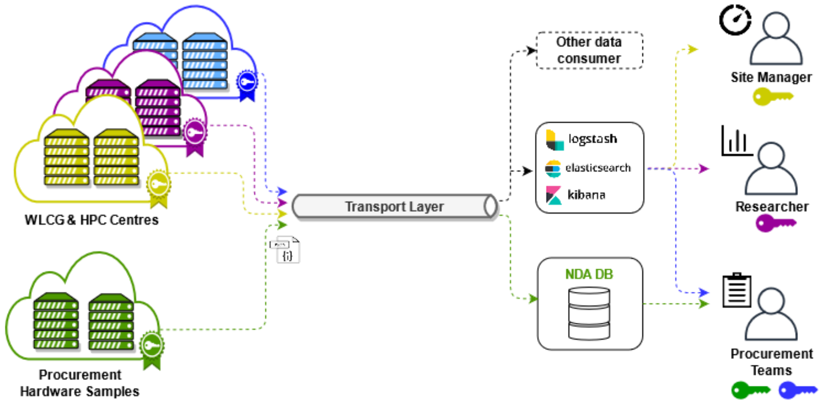


https://gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite/-/tree/qa-v2.0

Features on HPC

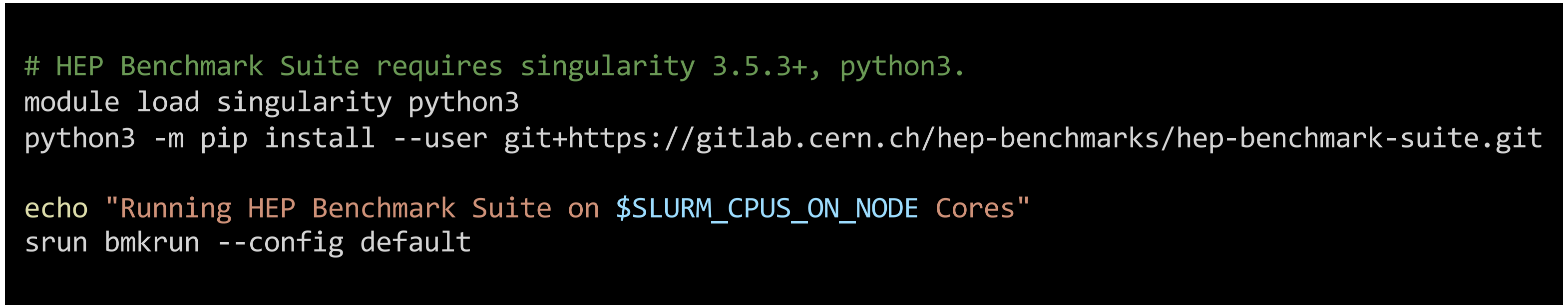
Benchmarking Heterogeneous architectures

- Multi-arch container workloads (x86, IBM Power, ARM …)
- Multi-GPU container workloads (Nvidia, AMD, Intel…)
- Easily extendable to other sciences!



Simple integration with SLRUM & other job orchestrators

- Single dependency on Python3.6 + container of your choice

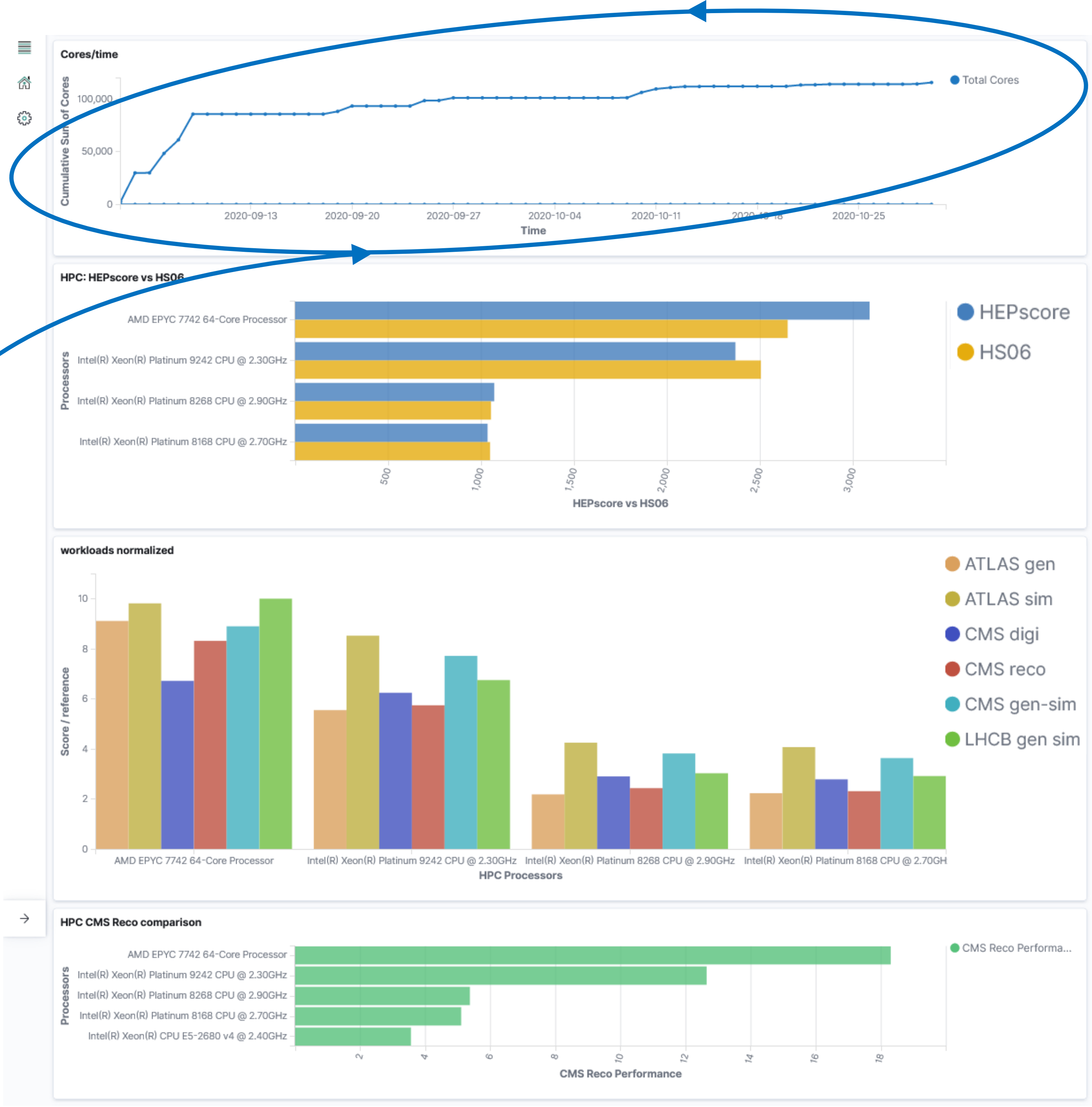


```
# HEP Benchmark Suite requires singularity 3.5.3+, python3.
module load singularity python3
python3 -m pip install --user git+https://gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite.git

echo "Running HEP Benchmark Suite on $SLURM_CPUS_ON_NODE Cores"
srun bmkrun --config default
```

Aggregation

Analysis via Kibana

Short Benchmarking Campaign
~120,000 cores
(Sept – Oct)



Cores/time
Total Cores
Cumulative Sum of Cores
100,000
50,000
0
2020-09-13
2020-09-20
2020-09-27
2020-10-04
2020-10-11
2020-10-18
2020-10-25
Time
HPC: HEPscore vs HS06
AMD EPYC 7742 64-Core Processor
Intel(R) Xeon(R) Platinum 9242 CPU @ 2.30GHz
Intel(R) Xeon(R) Platinum 8268 CPU @ 2.90GHz
Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
Processors
500
1,000
1,500
2,000
2,500
3,000
HEPscore vs HS06
HEPscore
HS06
workloads normalized
Score / reference
10
8
6
4
2
0
AMD EPYC 7742 64-Core Processor
Intel(R) Xeon(R) Platinum 9242 CPU @ 2.30GHz
Intel(R) Xeon(R) Platinum 8268 CPU @ 2.90GHz
Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GH
HPC Processors
ATLAS gen
ATLAS sim
CMS digi
CMS reco
CMS gen-sim
LHCB gen sim
HPC CMS Reco comparison
AMD EPYC 7742 64-Core Processor
Intel(R) Xeon(R) Platinum 9242 CPU @ 2.30GHz
Intel(R) Xeon(R) Platinum 8268 CPU @ 2.90GHz
Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz
Processors
2
4
6
8
10
12
14
16
18
CMS Reco Performance
CMS Reco Performa...

Summary

HEP Benchmark Suite: Extended to benchmark HPC

- **Designed for HPC job submission - Flexible, modular, and unprivileged**
- **Heterogeneous support included (multi-GPU and multi-arch containers)**
- **Easily extendable (Belle2 production workload for HEPscore recently added!)**
- **Fault-tolerant, Repeatable & Verifiable**
- **Enables statistical analysis via reporting**
- **Verified on several HPC sites, soon on PRACE HPC testbeds**

Interested? Full source and examples at:
gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite

THANK YOU

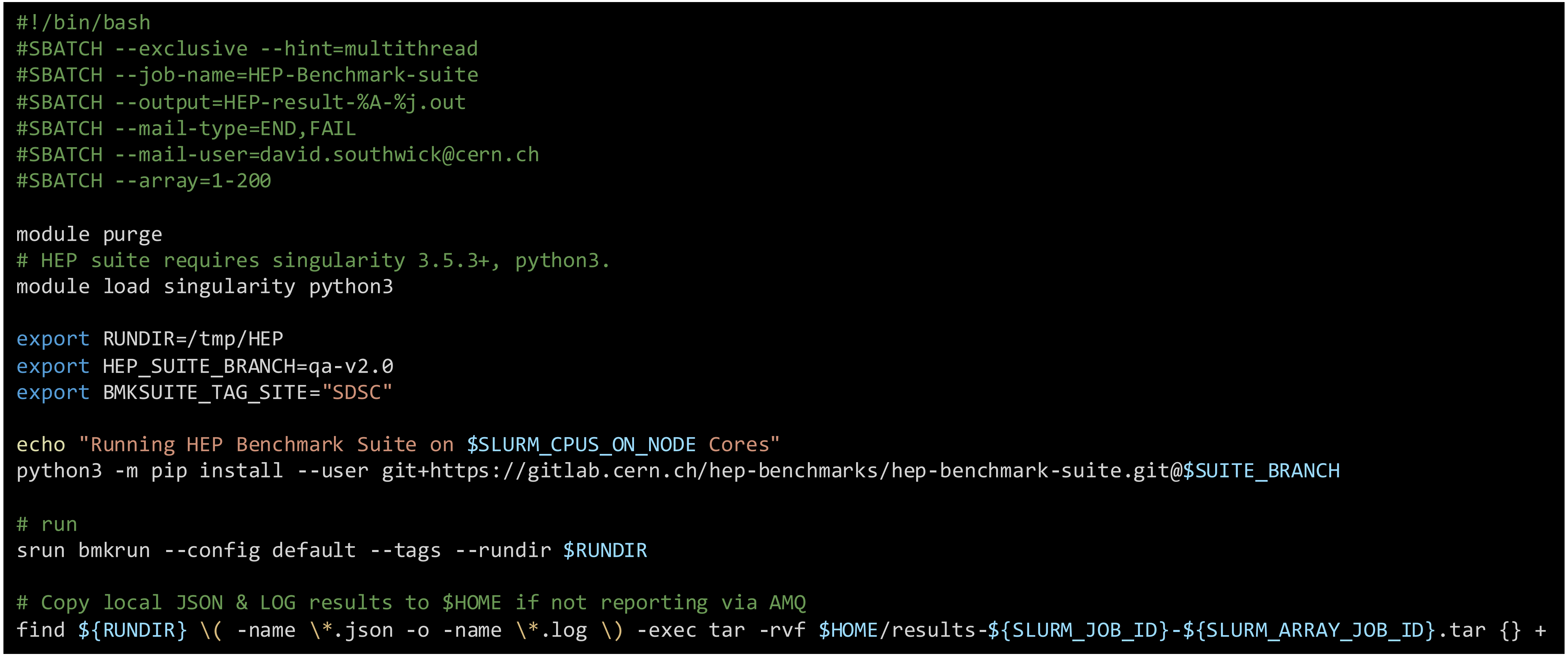
EGI-ACE

SKA SQUARE KILOMETRE ARRAY

GÉANT

PRACE

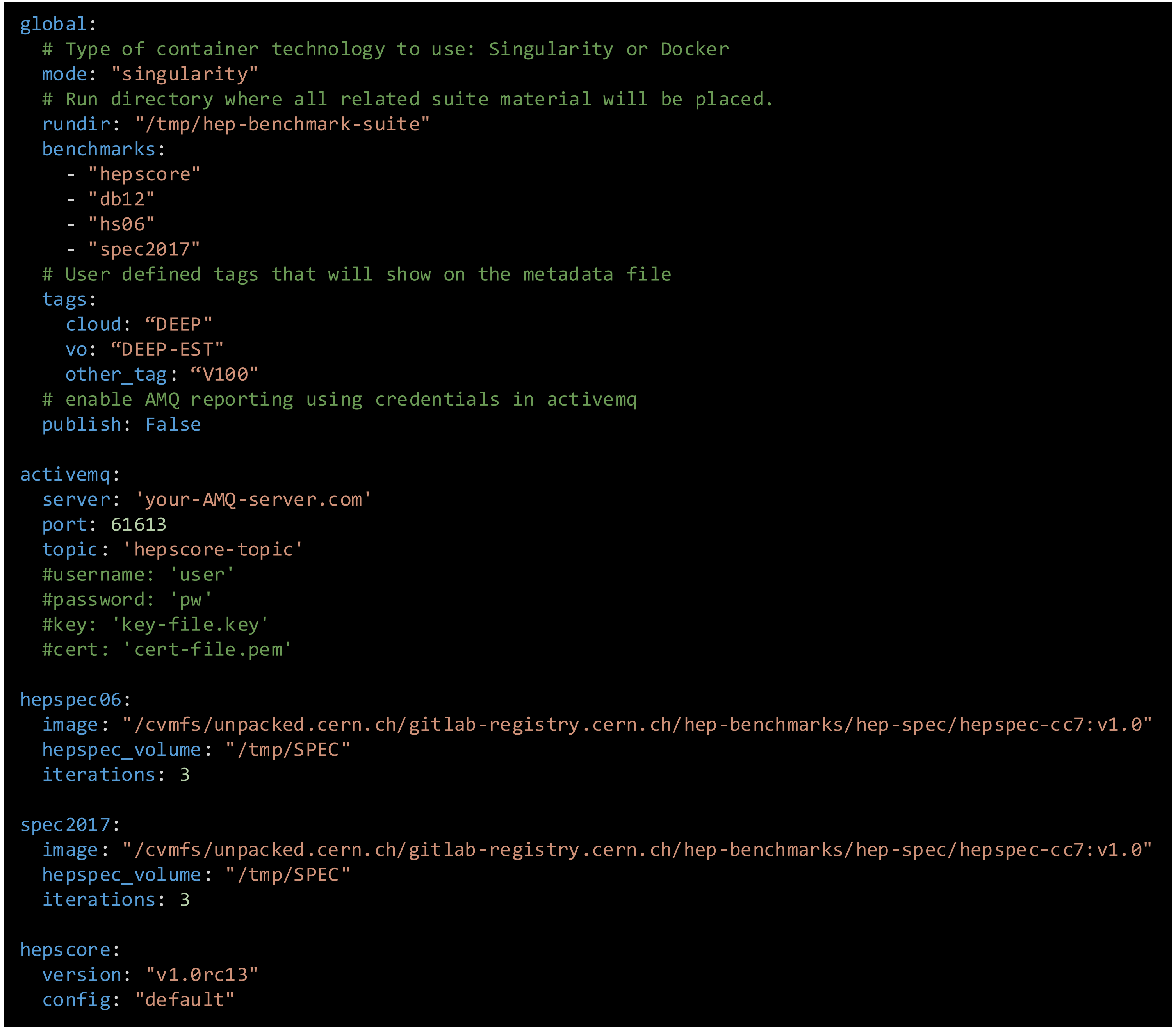
SLURM batch example



```
#!/bin/bash
#SBATCH --exclusive --hint=multithread
#SBATCH --job-name=HEP-Benchmark-suite
#SBATCH --output=HEP-result-%A-%j.out
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user=david.southwick@cern.ch
#SBATCH --array=1-200

module purge
# HEP suite requires singularity 3.5.3+, python3.
module load singularity python3

export RUNDIR=/tmp/HEP
export HEP_SUITE_BRANCH=qa-v2.0
export BMKSUITE_TAG_SITE="SDSC"

echo "Running HEP Benchmark Suite on $SLURM_CPUS_ON_NODE Cores"
python3 -m pip install --user git+https://gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite.git@$SUITE_BRANCH

# run
srun bmkrun --config default --tags --rundir $RUNDIR

# Copy local JSON & LOG results to $HOME if not reporting via AMQ
find ${RUNDIR} \( -name \*.json -o -name \*.log \) -exec tar -rvf $HOME/results-${SLURM_JOB_ID}-${SLURM_ARRAY_JOB_ID}.tar {} +
```

Declarative YAML configuration

Hashed & included in report



```
global:
  # Type of container technology to use: Singularity or Docker
  mode: "singularity"
  # Run directory where all related suite material will be placed.
  rundir: "/tmp/hep-benchmark-suite"
  benchmarks:
    - "hepscore"
    - "db12"
    - "hs06"
    - "spec2017"
  # User defined tags that will show on the metadata file
  tags:
    cloud: “DEEP"
    vo: “DEEP-EST"
    other_tag: “V100"
  # enable AMQ reporting using credentials in activemq
  publish: False

activemq:
  server: 'your-AMQ-server.com'
  port: 61613
  topic: 'hepscore-topic'
  #username: 'user'
  #password: 'pw'
  #key: 'key-file.key'
  #cert: 'cert-file.pem'

hepspec06:
  image: "/cvmfs/unpacked.cern.ch/gitlab-registry.cern.ch/hep-benchmarks/hep-spec/hepspec-cc7:v1.0"
  hepspec_volume: "/tmp/SPEC"
  iterations: 3

spec2017:
  image: "/cvmfs/unpacked.cern.ch/gitlab-registry.cern.ch/hep-benchmarks/hep-spec/hepspec-cc7:v1.0"
  hepspec_volume: "/tmp/SPEC"
  iterations: 3

hepscore:
  version: "v1.0rc13"
  config: "default"
```

Collaborating HPC Sites

...and their hardware

San Diego Supercomputer Center

- Intel Skylake 8168 @ 2.8GHz
- Intel Cascade Lake 8268 @ 2.9GHz
- Intel Skylake 6148 @ 2.4GHz
- AMD EPYC 7742 @ 2.25GHz

Advania (now atNorth) - Iceland

- Intel Cascade Lake 9242 @ 2.3GHZ

SDSC SAN DIEGO SUPERCOMPUTER CENTER

atnorth

advania

We thank you for your support!