



CPU Benchmarking in WLCG : HEPscore

Rapport sur les travaux du HEPiX Benchmarking Group [1] et en particulier sur le benchmark candidat successeur de HEP-SPEC06 : HEPscore.

Jean-Michel BARBET, Laboratoire SUBATECH

Credits : Domenico Giordano et tous les membres du « Benchmarking Group »

HEPscore

- Un outil [3] développé par le groupe de travail HEPiX « Benchmarking » (HEPiX Benchmarking Group) suite à la demande de WLCG pour remplacer HEP-SPEC06
- Devrait déboucher sur un nouveau benchmark qui sera nommé « HEPscore202x » (HEPscore2022 ?)
- N'est plus basé sur un sous ensemble de benchmarks de la société SPEC mais sur des « workloads » fournies par les expériences : mesure d'un nombre d'évènements traités par unité de temps

Historique des benchmarks WLCG

- SpecInt2000 (SI2K) : basé sur le sous ensemble des benchmarks sur les entiers de SPEC 2000
- HEP-SPEC06 (HS06) : basé sur un sous-ensemble « all-cpp » de SPEC CPU2006 (entiers et flottants)
- A venir : HEPscore202x : moyenne géométrique (pondérée?) des scores de différentes workloads HEP

Pourquoi un nouveau benchmark ?

- Des déviations entre les prédictions de HS06 et les calculs des expériences ont commencé à apparaître avec les dernière générations de processeurs
- Le benchmark SPEC CPU 2006 n'est plus supporté depuis 2018
- SPEC CPU 2017 a été considéré mais des études portant sur les cycles CPU montre que son comportement est éloigné de celui des applications HEP [2]
- Préférence pour un logiciel disponible dans licence
- Un benchmark basé sur les applications HEP est souhaitable

Travaux HEPiX Benchmarking group

- Comparaison entre HS06 et SPEC CPU 2017 (C++ applications)
- Création de containers comportant une applications HEP « stand-alone » avec des interfaces permettant de spécifier un nombre d'évènements et de threads, de récupérer les résultats dans un format standard (json) en sortie
- Développement d'un outil HEPscore permettant d'exécuter une suite de containers pour calculer un résultat global basé sur une moyenne géométrique pondérée
- Comparaison des résultats de HEPscore(beta) avec HS06 et SPEC CPU2017
- Développement de la « Benchmarking Suite » [4] et outils de reporting

En un coup d'oeil

Diapo extraite de [7]

HEP Benchmarks project

Three main components

- HEP Workloads

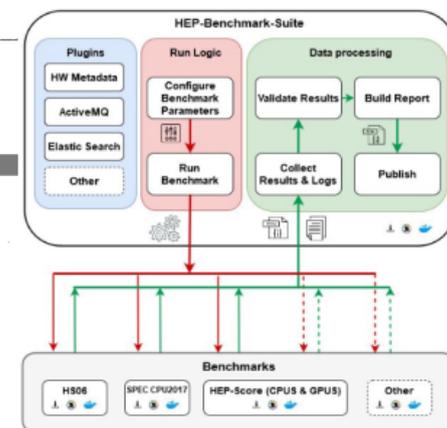
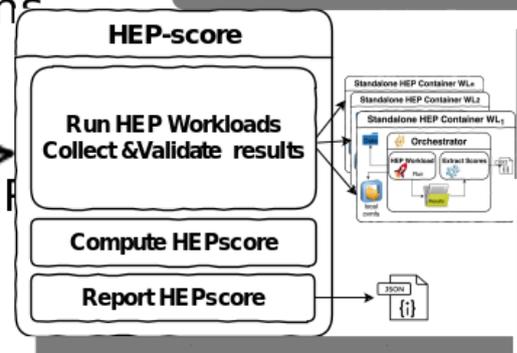
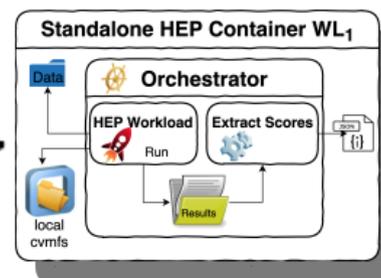
- Individual **reference** HEP applications
- Common build infrastructure

- HEP Score

- Orchestrate the run of a series of HEP workloads
- Compute the **HEPscore** value
- Report whole set of WL results

- HEP Benchmark Suite

- Orchestrator of multiple benchmark suites
 - HEPscore, HS06, SPEC CPU2017...



Benchmarking suite

- Un outil [4] permettant d'exécuter des benchmarks et d'écrire les résultats dans une base de données centrale (ElasticSearch) afin de l'exploiter avec des outils de « data-mining » (Kibana, Grafana,...)
- Supporte actuellement : DB12, HS06, SPEC-CPU2017(subset), HEPscore(beta)

HEPscore

- Script écrit en Python(Python3)
- Dépendances limitées
- Installation facile avec pip3
- Configuration par défaut : utilise singularity pour lancer 3 exécutions de chacune des workload actuelles :
 - atlas-gen-bmk
 - belle2-gen-sim-reco-bmk
 - cms-gen-sim-bmk
 - cms-digi-bmk
 - Lhcb-gen-sim-bmk
- Résultat par rapport à une machine référence et normalisé pour être équivalent à HS06

Essayer HEPscore (1)

Pré-requis : singularity (docker), pip3 :

```
sudo yum install singularity python3-pip
```

Installer HEPscore (userland: dans ~/.local) :

```
pip3 install --user git+https://gitlab.cern.ch/hep-benchmarks/hep-score.git
```

Préparation :

```
mkdir /tmp/results  
setenv SINGULARITY_CACHEDIR /dlocal/singularity-cache/
```

Essayer HEPscore (2)

```
~/local/bin/hep-score /tmp/results/
2021-11-08 15:58:25 hepscore [INFO] .atlas-sim-bmk is commented out: Skipping this
benchmark!
2021-11-08 15:58:25 hepscore [INFO] HEPscore20POC Benchmark
2021-11-08 15:58:25 hepscore [INFO] Config Hash:
308789e4115de88571518055c1478875aff38a82dd0a34dc3179ec2a07a0f514
2021-11-08 15:58:25 hepscore [INFO] HEPscore version:      1.2.0
2021-11-08 15:58:25 hepscore [INFO] System:                Linux nanpcw1.in2p3.fr
3.10.0-1160.36.2.el7.x86_64 #1 SMP Wed Jul 21 11:57:15 UTC 2021 x86_64
2021-11-08 15:58:25 hepscore [INFO] Container Execution: singularity
2021-11-08 15:58:25 hepscore [INFO] Registry:           docker://gitlab-
registry.cern.ch/hep-benchmarks/hep-workloads
2021-11-08 15:58:25 hepscore [INFO] Output:
/tmp/results/HEPscore_08Nov2021_155825
2021-11-08 15:58:25 hepscore [INFO] Date:                   Mon Nov  8 15:58:25 2021

2021-11-08 15:58:25 hepscore [INFO] Executing 3 runs of atlas-gen-bmk
2021-11-08 15:58:25 hepscore [INFO] Starting run0
[...]
```

Résultats

```
jq . /tmp/results/HEPscore_10Nov2021_131321/HEPscore20POC.json
{
  "benchmarks": {
    "atlas-gen-bmk": {
      "results_file": "atlas-gen_summary.json",
      "weight": 1,
      "version": "v2.1",
      "args": {
        "threads": 1,
        "events": 200
      },
      "run0": {
[...]
```

```
    "environment": {
      "system": "Linux nanpcw1.in2p3.fr 3.10.0-1160.36.2.el7.x86_64 #1
SMP Wed Jul 21 11:57:15 UTC 2021 x86_64",
      "start_at": "Wed Nov 10 13:13:21 2021",
      "singularity_version": "3.8.0-1.el7",
      "end_at": "Wed Nov 10 18:29:31 2021"
    },
    "wl-scores": {
[...]
```

```
    "score": 93.454,
    "status": "success",
    "score_per_core": 11.682
  }
}
```

Utiliser la Benchmarking suite

- Le plus facile est d'utiliser un des scripts exemples :
`run_HEPscore_default_from_wheels.sh`
- Par défaut la publication dans le dashboard est désactivée
- Pour pouvoir publier, il faut utiliser un certificat et le faire enregistrer par le groupe benchmarking
- On devient ainsi « early adopter » et on enrichit la base de données des résultats par types de CPU :-)

Tester une « workload »

- Il est possible de tester les workloads séparément :

```
singularity run -C -B /tmp:/tmp -B /tmp/results:/results docker://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads/cms-gen-sim-run3-bmk
[...]
```

```
[bmk-driver.sh] spawn 2 processes
```

```
[...]
```

```
[bmk-driver.sh] parse results and generate summary: completed (status=0)
```

```
[bmk-driver.sh] json file validation: starting
```

```
[...]
```

```
[bmk-driver.sh] exiting back to cms-gen-sim-run3-bmk.sh
```

```
[bmk-driver.sh] Wed Nov 10 13:09:17 CET 2021 exiting common benchmark driver (status=0)
```

prmon

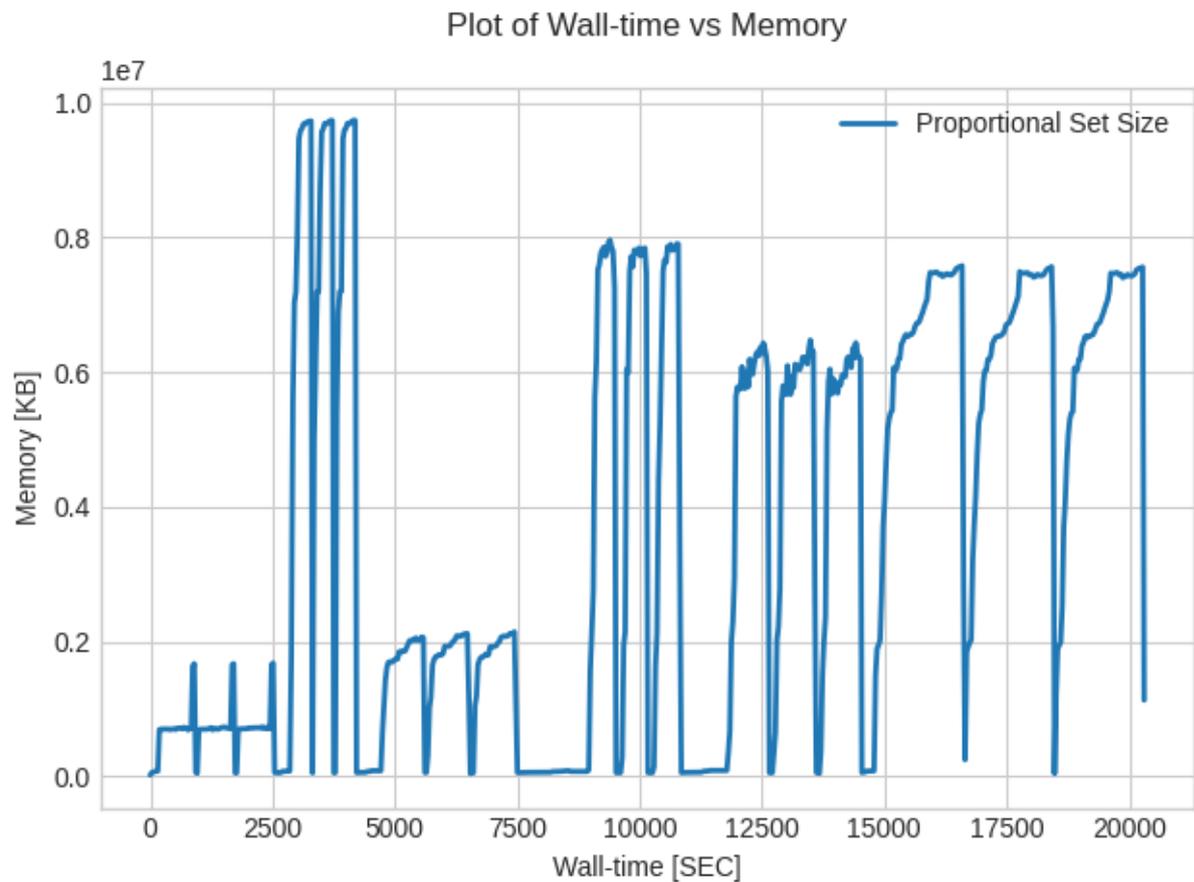
- L'outil prmon [5] permet de recueillir des métriques système collectées pendant l'exécution.
- Il suffit de lancer prmon avec le processus courant dans le script qui lance les benchmarks :

```
/dlocal/prmon_2.2.0_x86_64-static-gnu93-opt/bin/prmon -p $$&
```

- Les métriques sont disponible dans un fichier et on peut créer des graphiques :

```
pip3 install numpy pandas matplotlib  
/dlocal/prmon_2.2.0_x86_64-static-gnu93-opt/b  
in/prmon_plot.py  
./prmon_plot.py  
INFO      :: Saved output into PrMon_wtime_vs_pss.png
```

Prmon : exemple



Travaux en cours et perspectives

- Le Benchmarking Group valide actuellement les workloads « run3 » candidates à l'intégration dans le nouveau benchmark HEPscore 202x

Vous pouvez participer !

- La « taskforce » WLCG « HEP-SCORE deployment TF » travaille à définir quel sera la combinaison de tests qui composera le futur benchmark.
- Des travaux sont en cours pour valider des workloads utilisant les GPUs ou mixtes GPU/CPU
- Des travaux sont en cours pour pouvoir faire tourner les benchmarks sur des plateformes ARM

Références

[1] HEPiX Benchmarking Group

<https://twiki.cern.ch/twiki/bin/view/HEPIX/CpuBenchmark>

[2] Next Generation of HEP CPU benchmarks

<https://doi.org/10.1051/epjconf/201921408011>

[3] HEPscore

<https://gitlab.cern.ch/hep-benchmarks/hep-score>

[4] HEP Benchmarking suite

<https://gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite/>

[5] prmon :

https://indico.cern.ch/event/813751/contributions/3991506/attachments/2099462/3529369/prmon_mete.pdf

[6] HEPiX spring 2021 : HEP Benchmarks: updates and demo

<https://indico.cern.ch/event/995485/contributions/4257475/>

[7] HEPiX autumn 2021 : HEPiX Benchmarking WG update

<https://indico.cern.ch/event/1078853/contributions/4576275/>

[8] HEPiX benchmarking solution for WLCG computing resources

(A paraître dans *Computing and Software for Big Science*, Springer)

