HEPscore Candidates

Tristan Sullivan
University of Victoria
September 7/22

Background

- Benchmarking working group tasked with finding replacement for HEPSPEC06 (HS06) based on physics workloads
- HS06: seven benchmarks, three runs each. The median of each benchmark is taken, and the final score is the geometric average of the seven individual benchmark scores
- HEPscore: Take the same approach, potentially with weights

$$\mathrm{HS} = \left(\prod_{i=1}^{n} \left(\frac{s_i}{r_i} \right)^{w_i} \right)^{\left(1/\sum_{i=1}^{n} w^i\right)}$$

s_i = workload score
w_i = workload weight
r_i = score on reference machine

If sum of weights is one, this simplifies to

$$\mathrm{HS} = \prod_{i=1}^{n} \left(\frac{s_i}{r_i} \right)^{w_i}$$

Reference machine is
Intel Xeon E5-2630 v3 @ 2.40 Ghz, hyperthreading (HT) on

Workload score is defined as events processed per second

Example

	AMD_7551P, HT off	Ref. Machine
Physical cores	32	16
Year	2017	2014
Workload avg.	640.1	361.5
HEPScore	1.77	1
$HEPScore_n$	0.88	1
HS06_64	603.8	415.5
HS06_64/ref	1.45	1
$HS06\_64_n$/ref	0.73	1

	AMD_7551P, HT on	Ref. Machine
Physical cores	32	16
Year	2017	2014
Workload avg.	700.7	361.5
HEPScore	1.94	1
$HEPScore_n$	0.97	1
HS06_64	729.9	415.5
HS06_64/ref	1.76	1
$HS06\_64_n$/ref	0.88	1

$HEPScore_n$ and $HS06_n$ include a further normalization by the number of physical cores of the machine (actually $pcores/pcores_{ref}$); this is useful for comparing benchmarks (e.g. HEPScore to HS06)

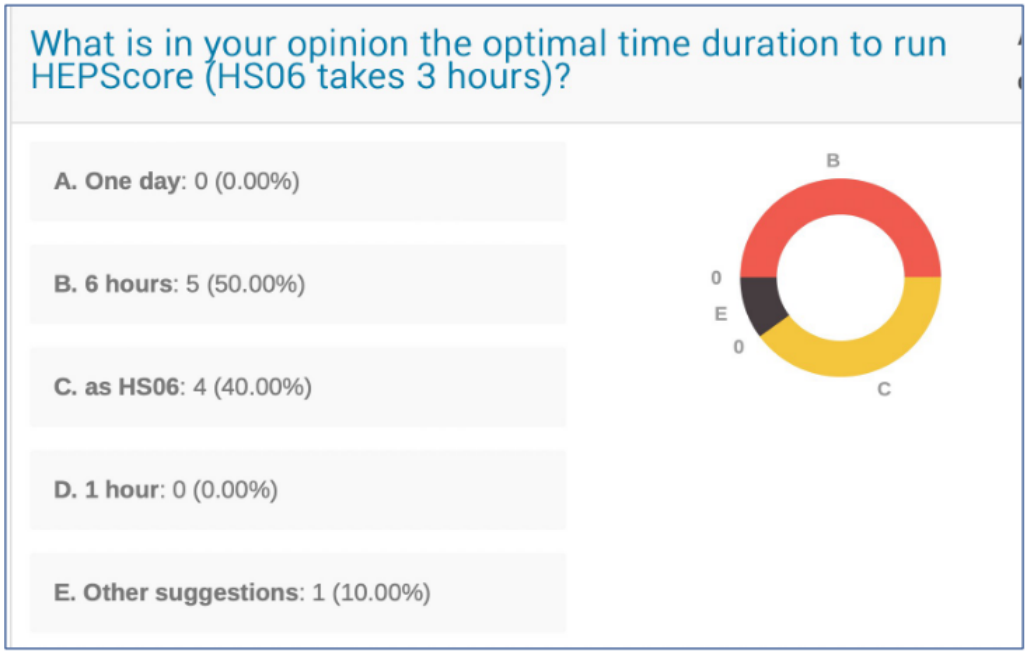
Ratio of ~1.2 between HT on and HT off is typical. Here it is 1.2 for HS06_64, and 1.1 for HEPScore

I haven’t defined HEPScore yet; this is just to illustrate the calculation method. Definition will come in the next slides

Available Workloads

Workload	Running Time (m)
Atlas_gen_sherpa	31
Atlas_reco_mt	69
Atlas_sim_mt	156
CMS_gen_sim	42
CMS_digi	31
CMS_reco	51
Belle2_gen_sim_reco	25
Alice_gen_sim_reco	194
LHCb_gen_sim	104
Juno_gen_sim_reco	67
Gravitational Wave	138
Total	908 (15+ hours)

Times for three runs on reference machine



Runs in a time similar to HEPSpec06 (3 hours)

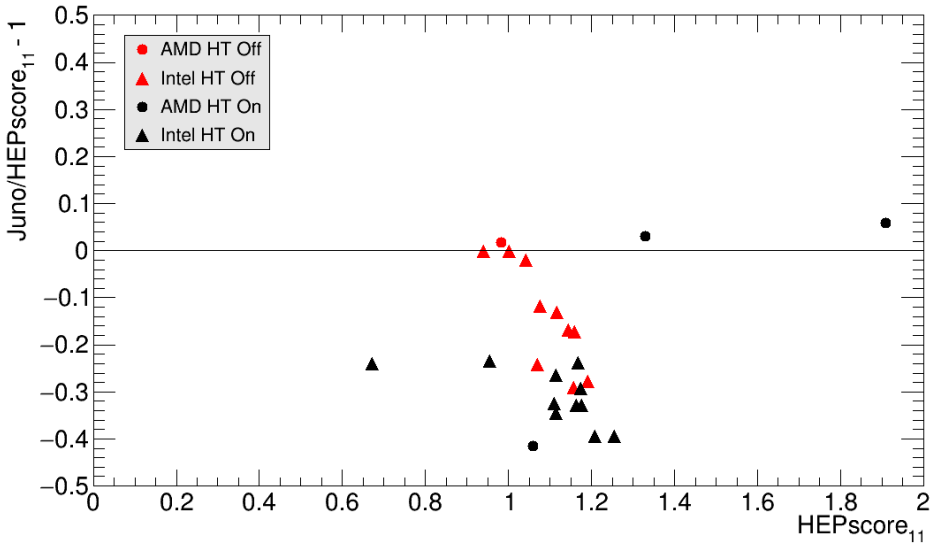
Including all workloads would exceed desired running time

Considerations for removal: length, dissimilarity to other workloads. Also different running conditions: GW doesn't saturate node, Juno requires > 2GB/core

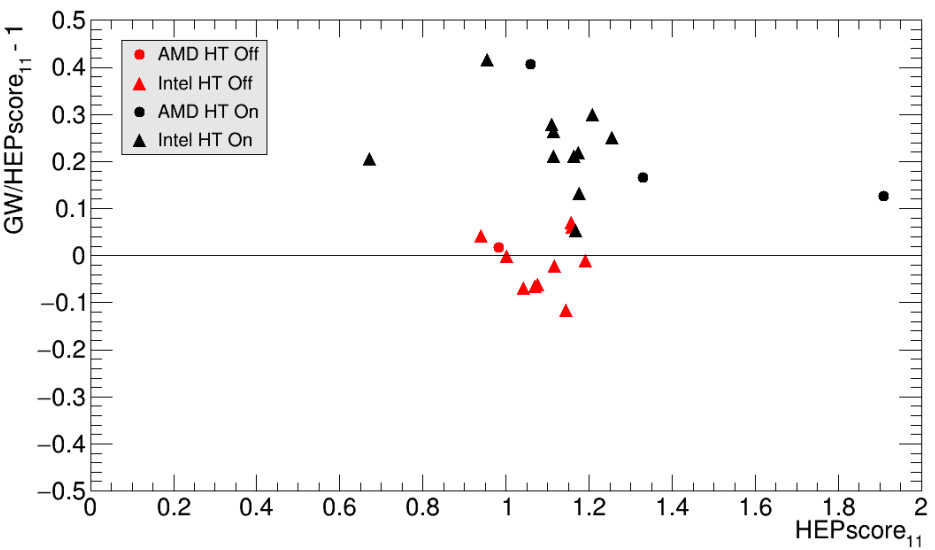
For now, define $HEPScore_{11}$ as average of all eleven workloads, for comparative purposes

Individual WL Deviations

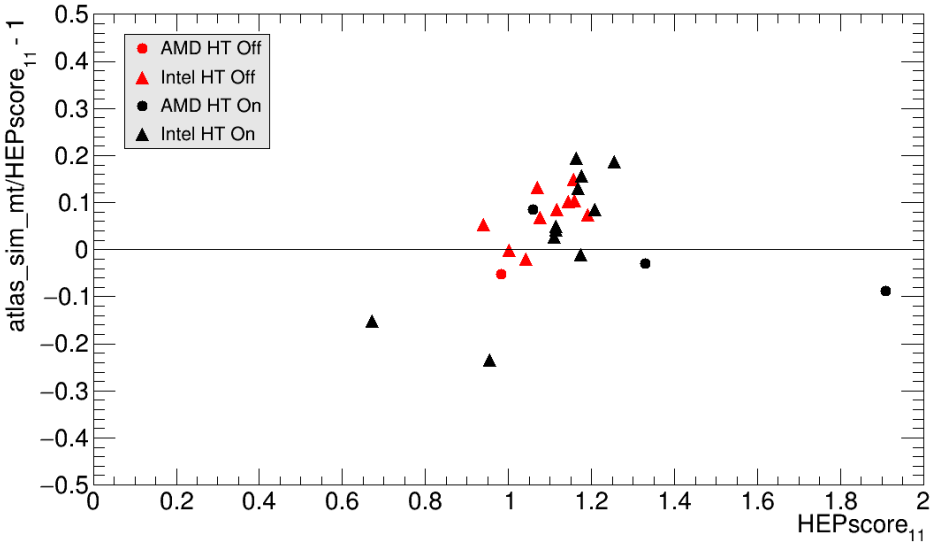
Juno



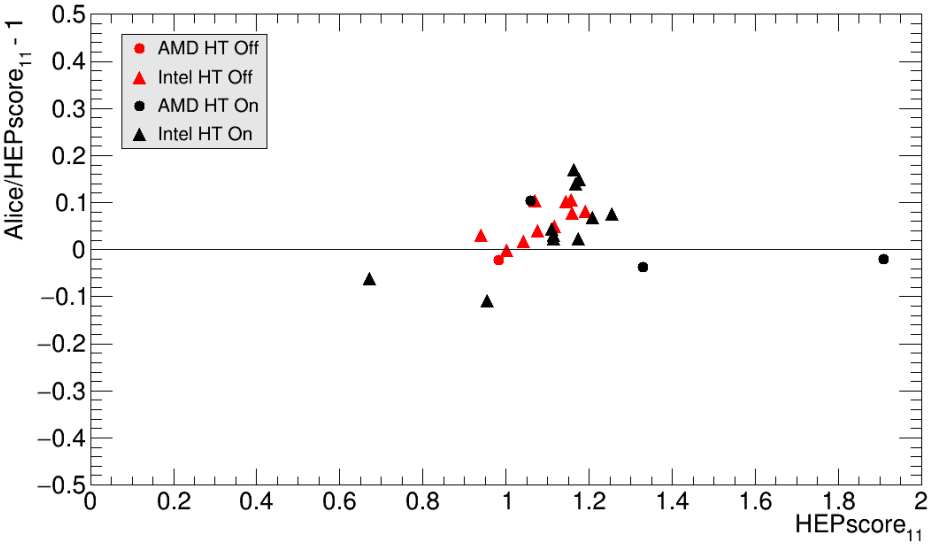
GW



Atlas_sim_mt



Alice



Tristan Sullivan

Available Workloads

Workload	Running Time	FOM
Atlas_gen_sherpa	31	0.123
Atlas_reco_mt	69	0.083
Atlas_sim_mt	156	0.092
CMS_gen_sim	42	0.077
CMS_digi	31	0.073
CMS_reco	51	0.086
Belle2_gen_sim_reco	25	0.044
Alice_gen_sim_reco	194	0.067
LHCb_gen_sim	104	0.124
Juno_gen_sim_reco	67	0.213
Gravitational Wave	138	0.151
Total	908 (~15 hours)	

Times for three runs on reference machine

FOM is the average distance from the x-axis in the plots of WL vs. $HEPScore_{11}$ (see previous page)

FOM is a measure of how different each workload is from the average of all workloads

Available Workloads

Workload	Running Time	FOM
Atlas_gen_sherpa	31	0.123
Atlas_reco_mt	69	0.083
~~Atlas_sim_mt~~	~~156~~	~~0.092~~
CMS_gen_sim	42	0.077
CMS_digi	31	0.073
CMS_reco	51	0.086
Belle2_gen_sim_reco	25	0.044
~~Alice_gen_sim_reco~~	~~194~~	~~0.067~~
LHCb_gen_sim	104	0.124
~~Juno_gen_sim_reco~~	~~67~~	~~0.213~~
~~Gravitational Wave~~	~~138~~	~~0.151~~
Total	353 (~6 hours)	

Times for three runs on reference machine

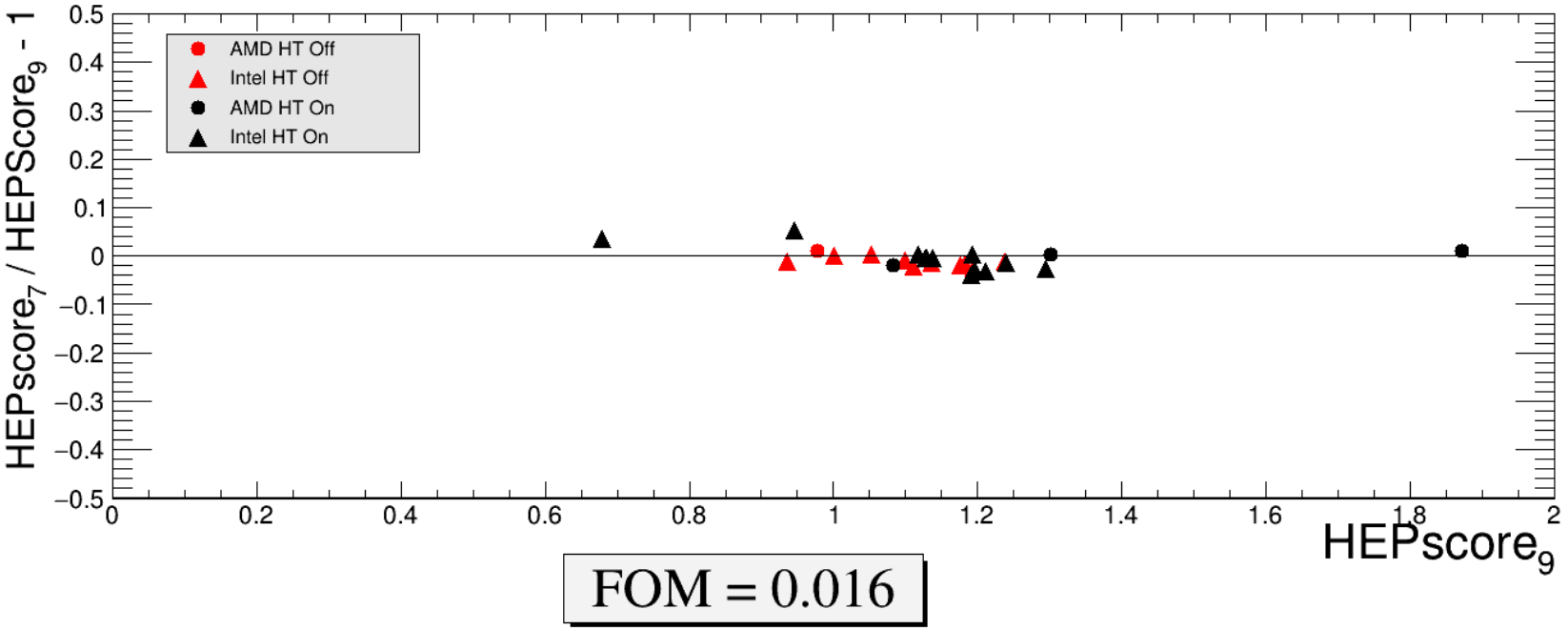
FOM is the average distance from the x-axis in the plots of WL vs. HEPScore (see previous page)

FOM is a measure of how different each workload is from the average of all workloads

Proposal: discard Alice and Atlas_sim based on length; GW and Juno for FOM and reasons discussed earlier (see Slide 4) to form nominal HEPScore

Total running time just under six hours

Impact of Removing Alice and Atlas_sim_mt



HEPScore$_9$ is the average of all workloads besides GW and Juno

HEPScore$_7$ additionally removes Alice and Atlas_sim_mt

Excluding these two workloads has minimal impact on HEPScore, saves almost six hours of running time

HEPScore Candidates

After removing GW, Juno, Atlas_sim_mt, and Alice

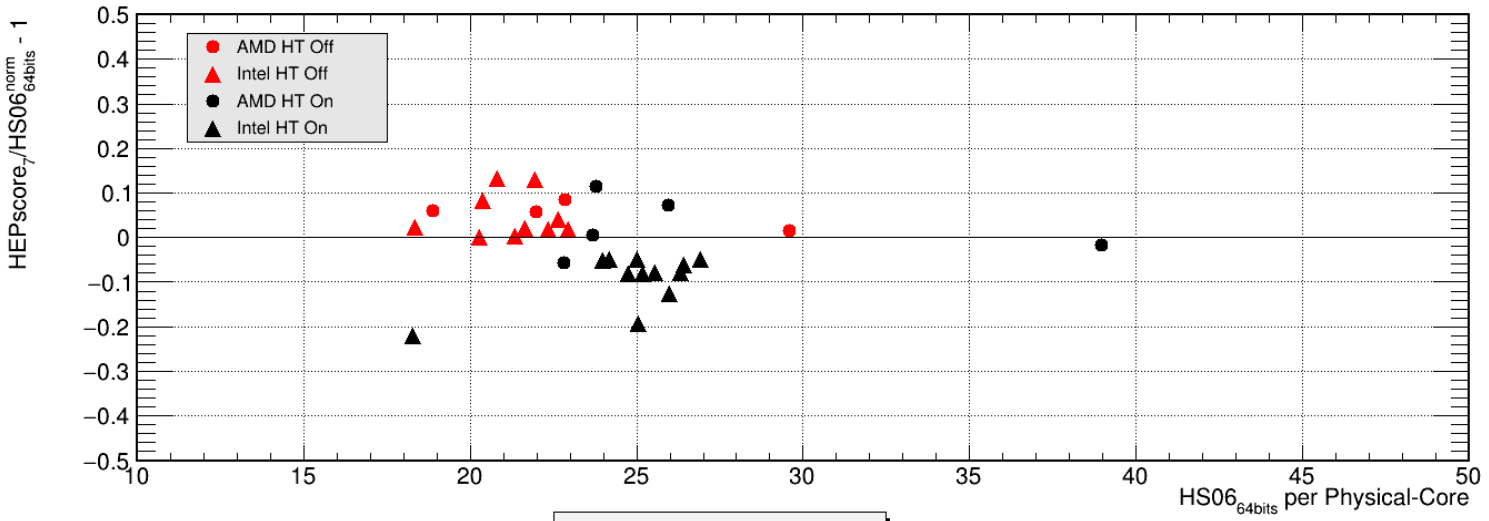
- “Nominal”: weight workloads equally
  - All workloads get weight 1/N, where N is the number of workloads (N = 7)
- “Experiment”: Weight experiments equally
  - Atlas workloads get 0.5/N, CMS get 0.333/N, LHCb and B2 get 1/N, where N is the number of experiments (N = 4)
- “Grid”: Weight by approximate fraction on grid, obtained from CPU usage in WLCG accounting
  - Atlas workloads get 0.4 total, CMS get 0.4 total, LHCb and B2 get 0.1

CPU Models

Architecture	Year of Release
NehalemEP	2009
IvyBridge	2013.5
Haswell	2014.5
Broadwell	2016
Naples	2017.25
Skylake	2017.5
Cascade Lake	2019.25
Rome	2019.5
Milan	2021
Ice Lake	2021.25

These are the CPU models for which data exist for all workloads

Nominal



FOM = 0.067

Workload	Weight
Atlas_gen	1/7
Atlas_reco	1/7
CMS_gen_sim	1/7
CMS_digi	1/7
CMS_reco	1/7
Belle2	1/7
LHCb	1/7

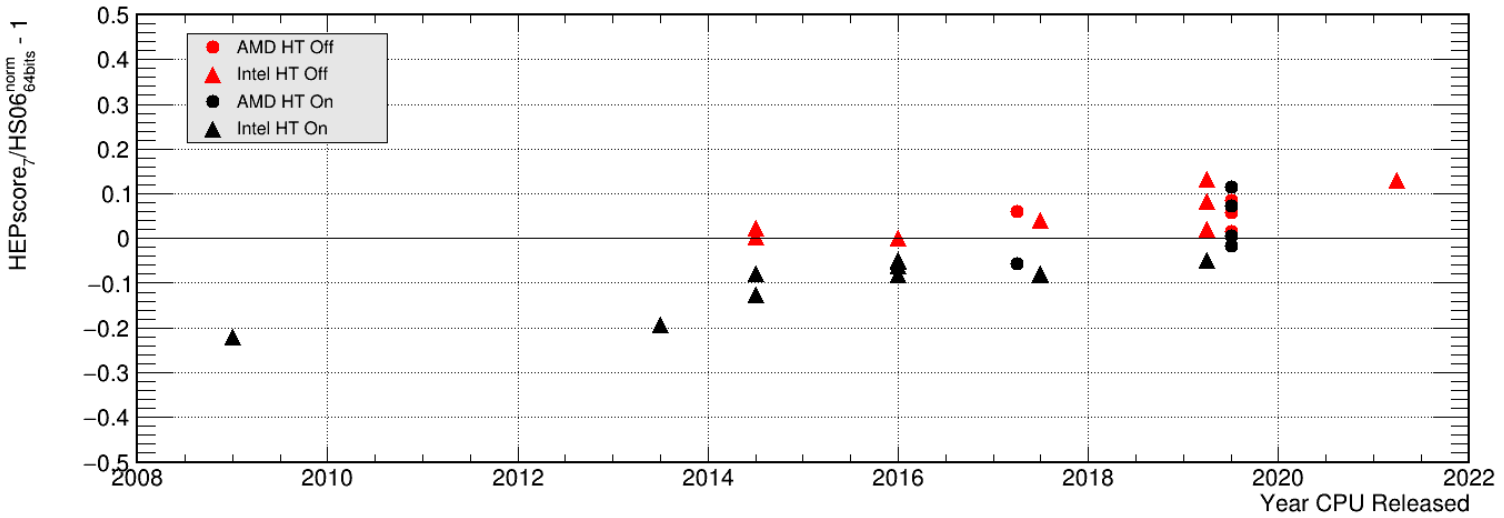
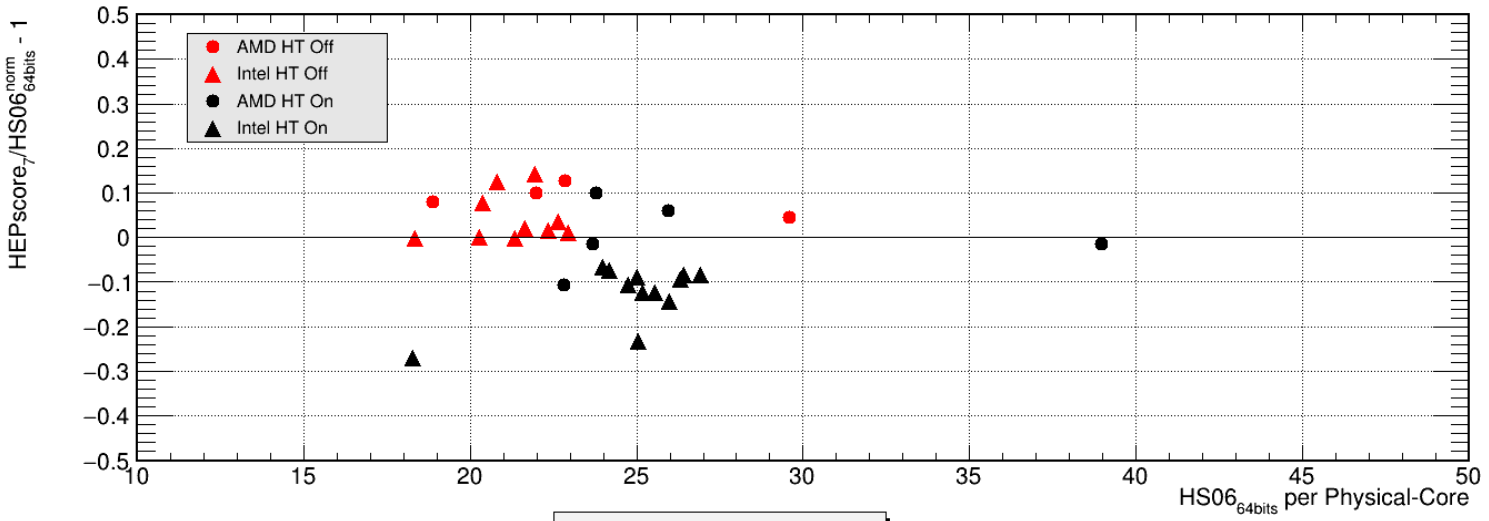


Figure of merit is average distance from X axis

Experiment



FOM = 0.083

Workload	Weight
Atlas_gen	0.125
Atlas_reco	0.125
CMS_gen_sim	0.0833
CMS_digi	0.0833
CMS_reco	0.0833
Belle2	0.25
LHCb	0.25

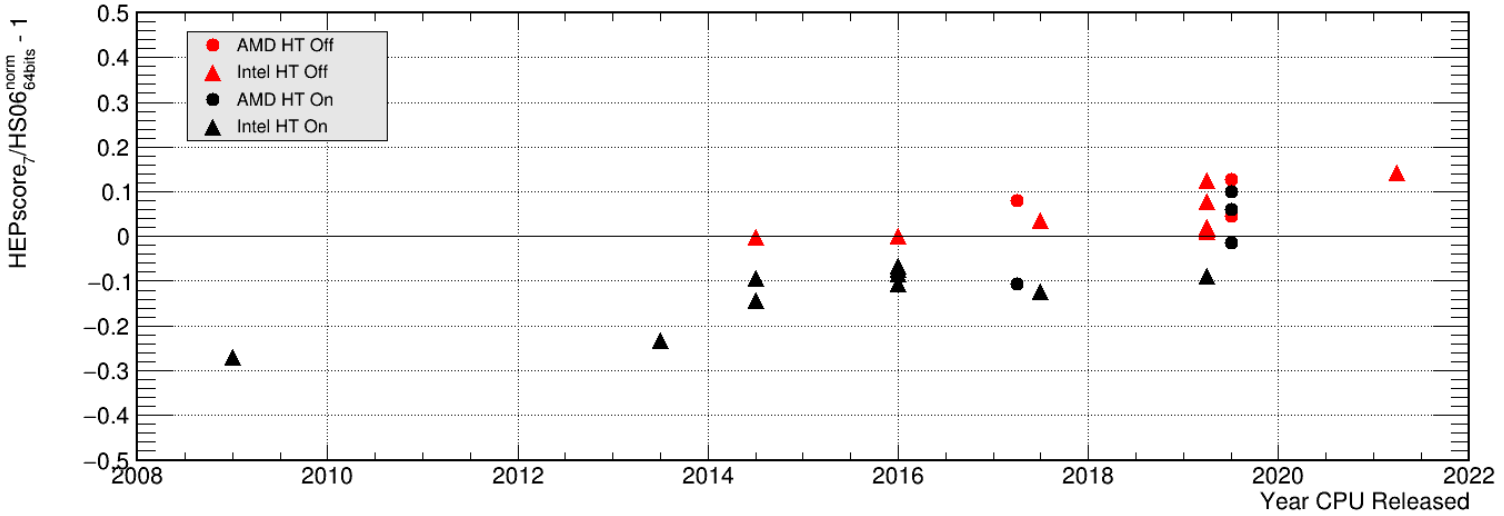
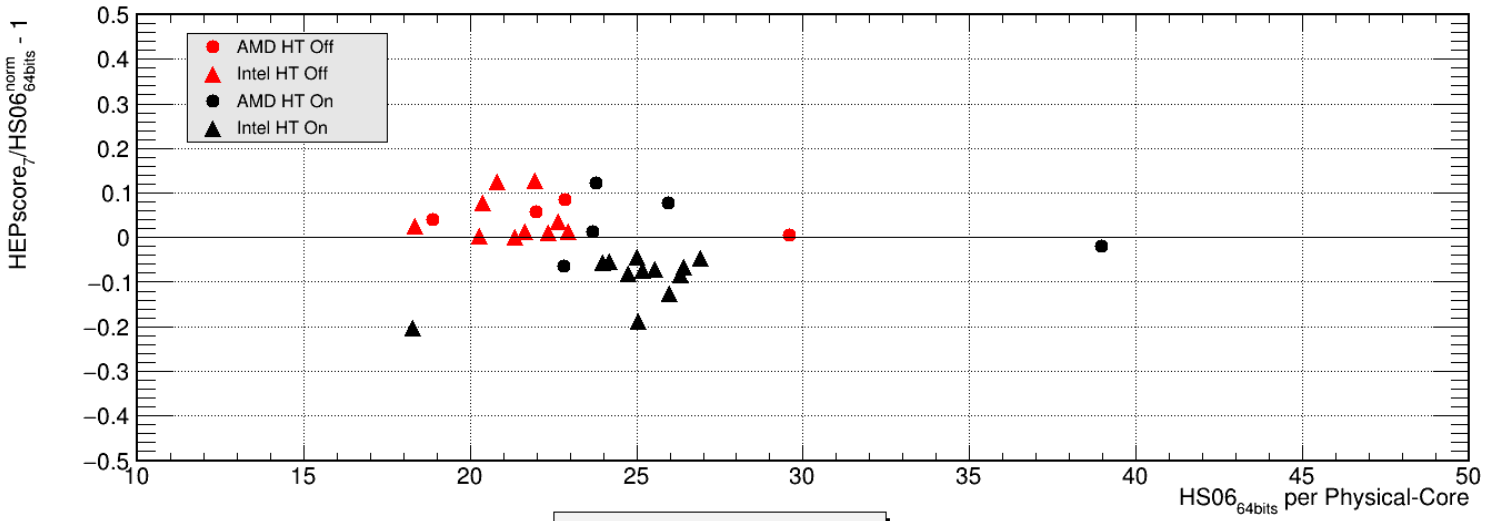


Figure of merit is average distance from X axis

Grid



FOM = 0.065

Workload	Weight
Atlas_gen	0.2
Atlas_reco	0.2
CMS_gen_sim	0.1333
CMS_digi	0.1333
CMS_reco	0.1333
Belle2	0.1
LHCb	0.1

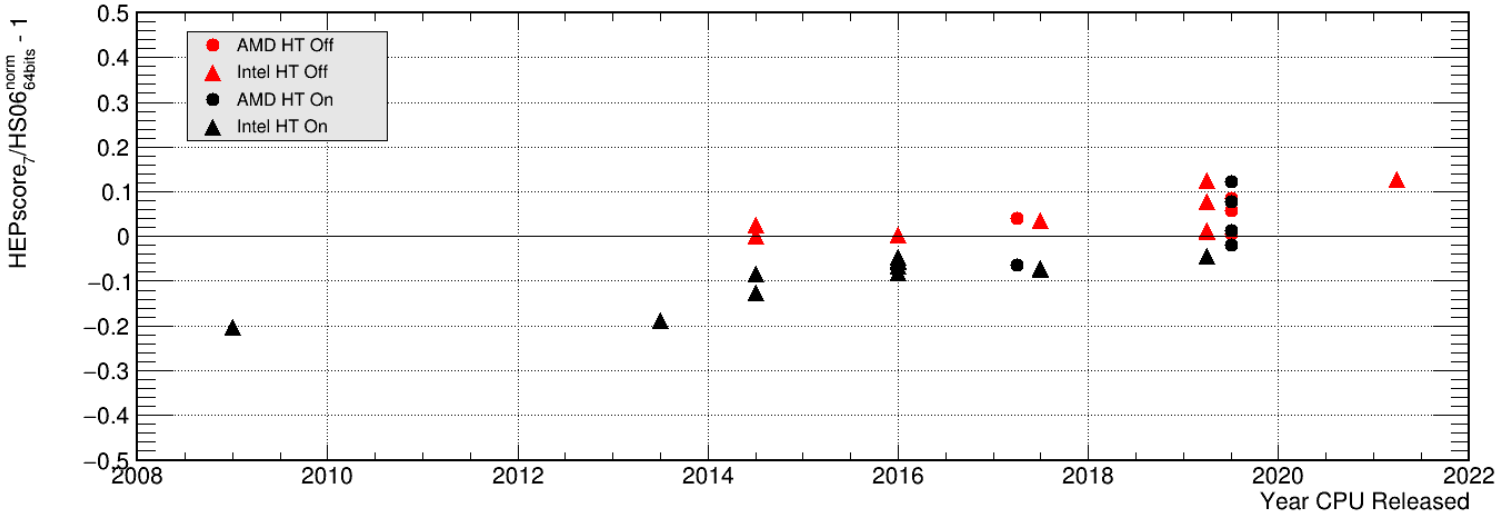
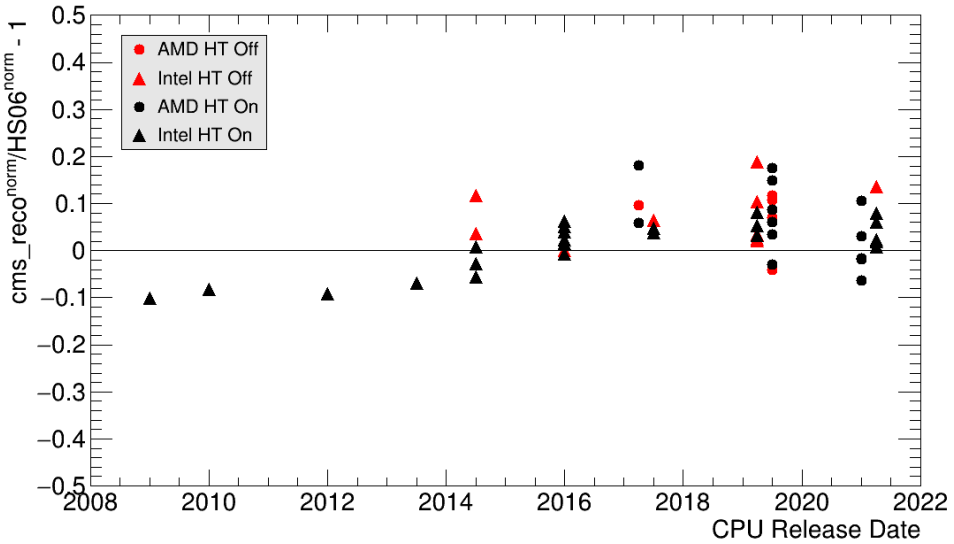


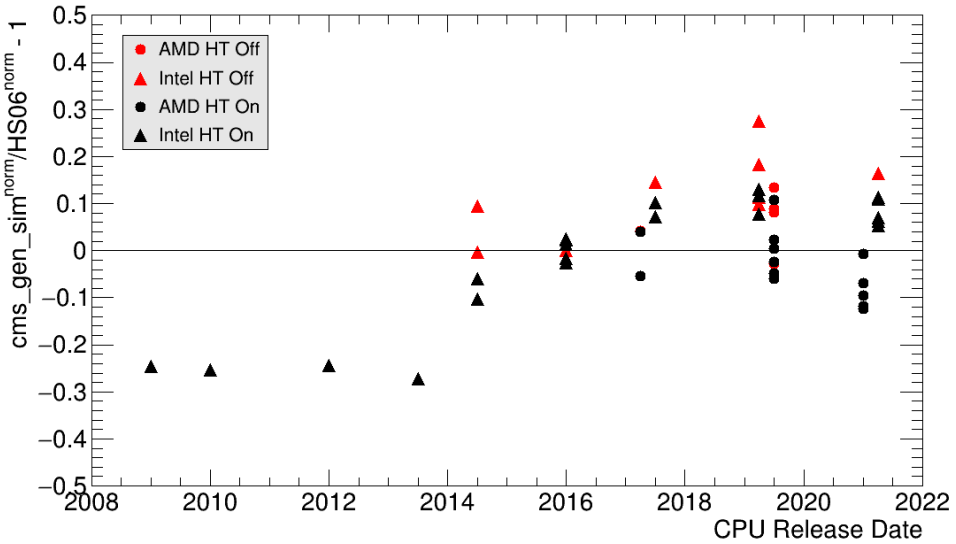
Figure of merit is average distance from X axis

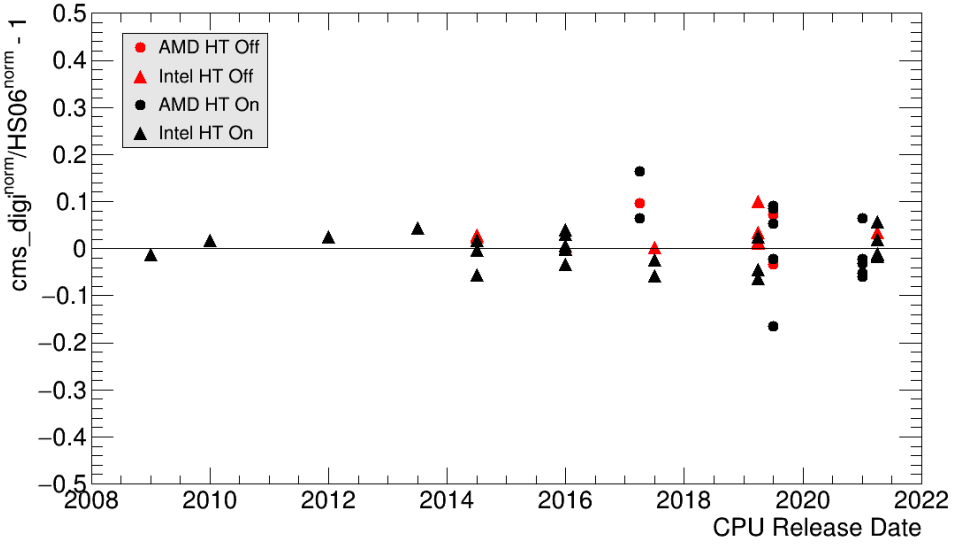
Conclusion

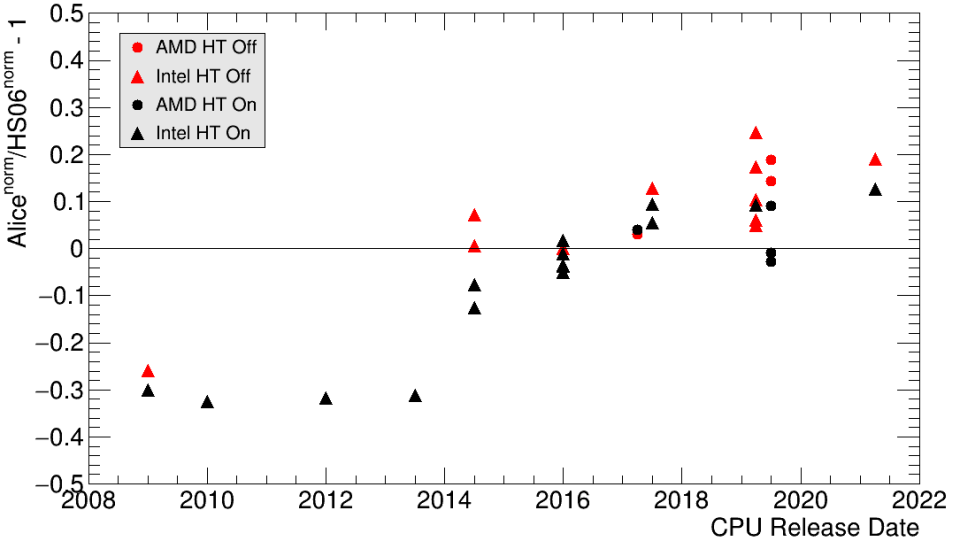
- Proposal: exclude Juno and Gravitational Wave workloads by first-principles arguments (see slide 4)
- Proposal: exclude Atlas_sim_mt and Alice workloads due to long running time, lack of impact on HEPScore (see slides 4 and 8)
- Workload weighting does not change qualitative behaviour of HEPScore
- Another candidate: weight by job type (sim, reco, etc.). Where to get numbers for fraction of each type on grid?

Appendix: Workloads vs. Year

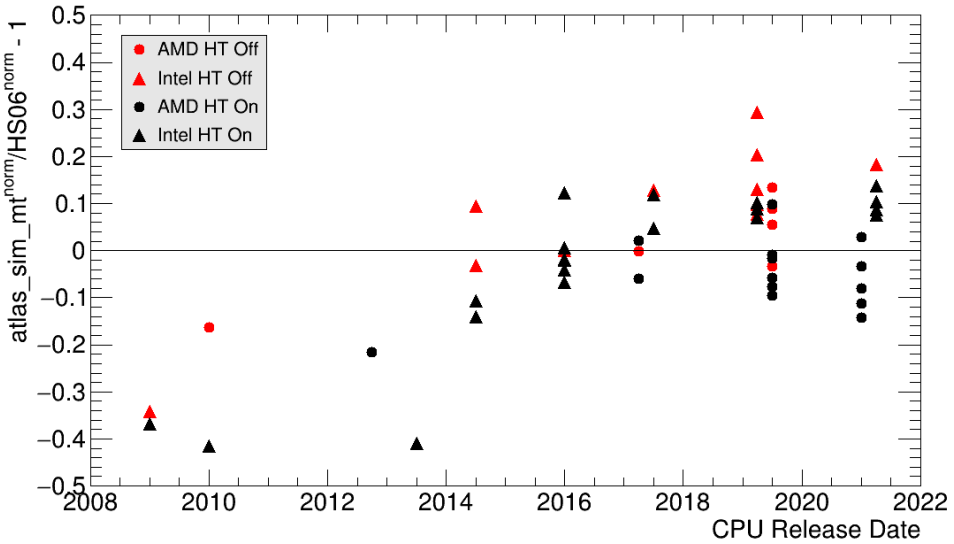


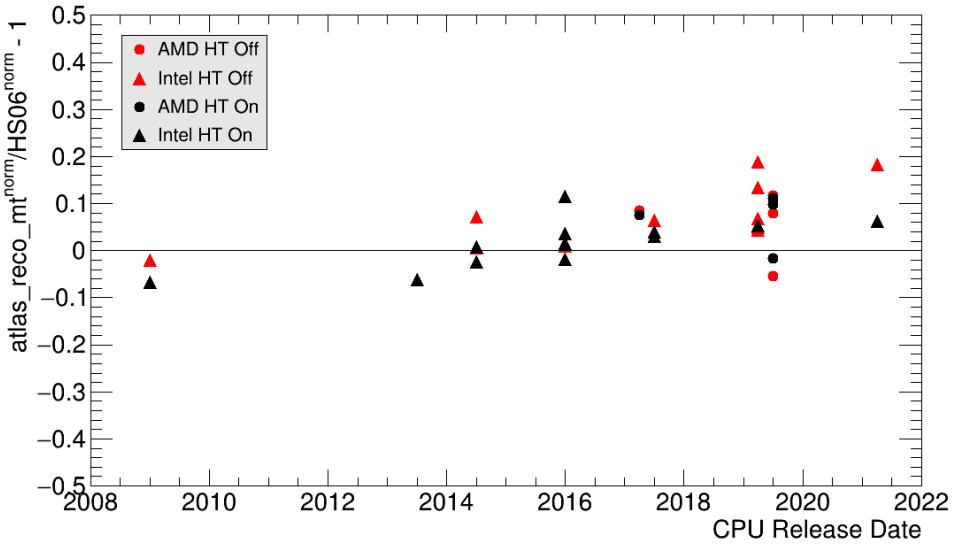


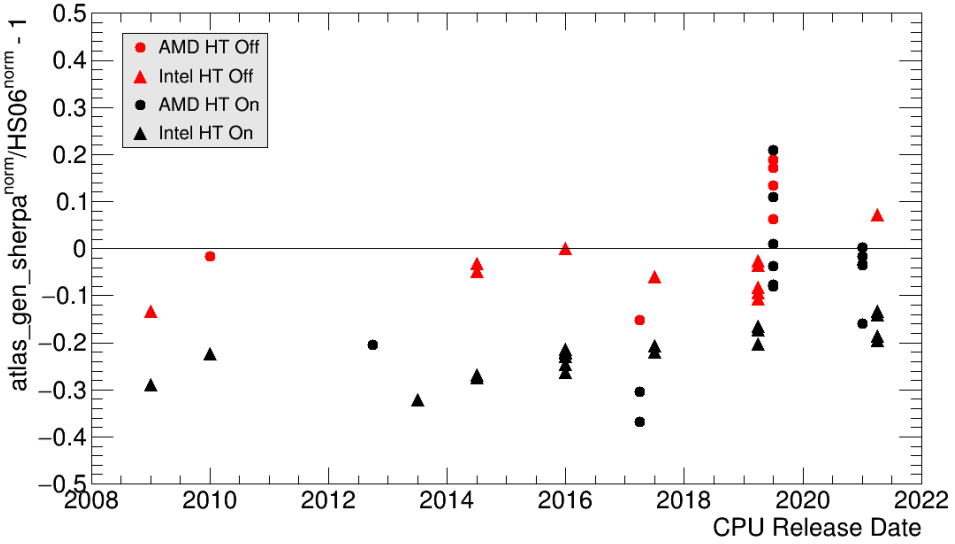


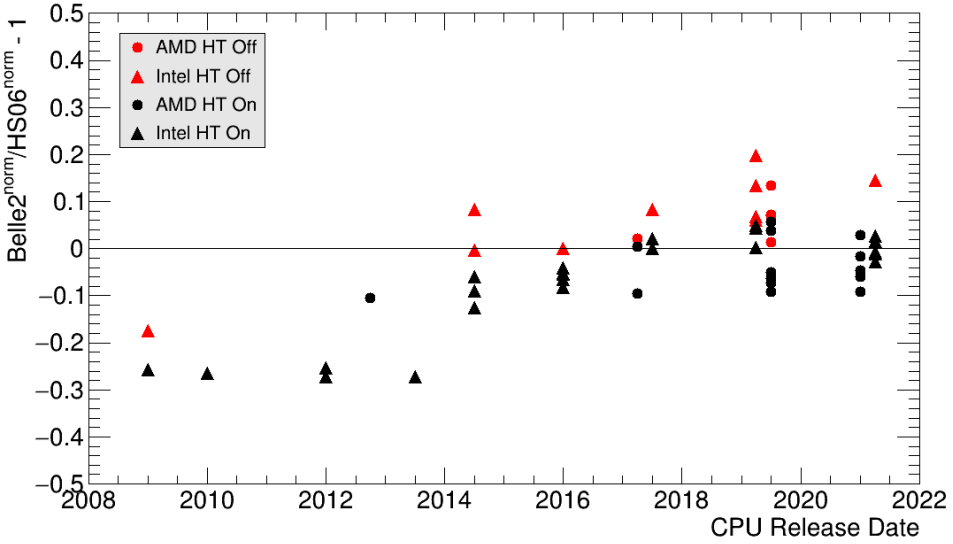


Appendix: Workloads vs. Year









Appendix: Workloads vs. Year