HEPscore Candidates

Tristan Sullivan
University of Victoria
September 19/22

Introduction

- As you have seen in previous talks, the benchmarking working group currently has eleven workloads available
- Many combinations were tried; I will show three possibilities
  - HEPScore_{11}: All workloads
  - HEPScore_{9}: Remove Juno and Gravitational Wave
  - HEPScore_{6}: Further remove Alice, Atlas_sim_mt, and CMS_digi
- These combinations will be motivated in the coming slides
- Very little difference between these candidates

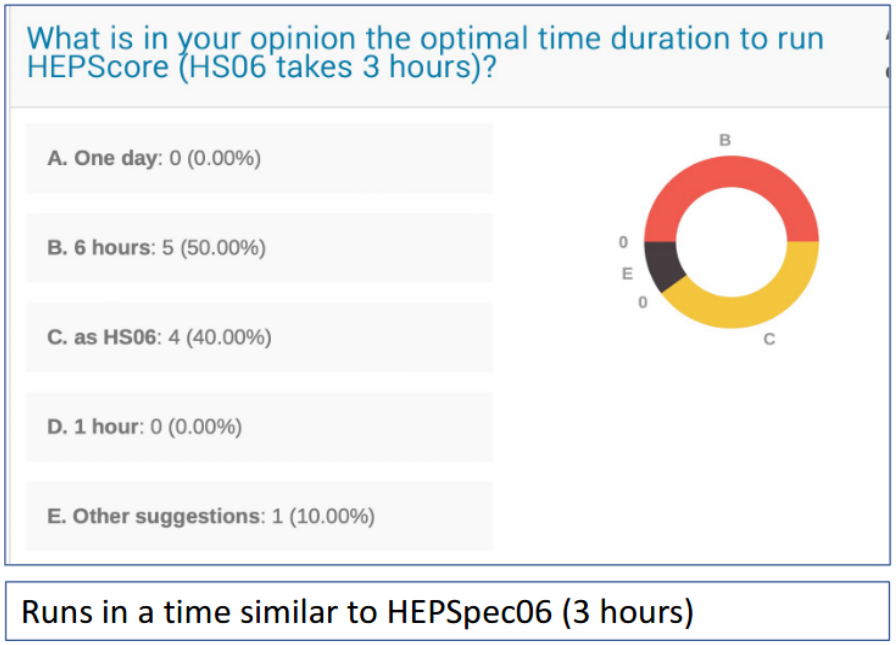
Introduction Continued

- After selecting workloads, there is also the possibility of applying weights
- Again, I will show three possibilities
  - Nominal: Equal weights for all workloads
  - Grid: Weighted by approximate fraction on grid, obtained from WLCG accounting (see Randy's talk, Slide 22)
  - Experiment: Equal weights for each experiment; ATLAS and CMS have multiple workloads, so these get weighted less
- Again, there is very little difference between these scenarios

Available Workloads

Workload	Running Time (m)	# of events * # of threads
Atlas_gen_sherpa	31	200 * 1
Atlas_reco_mt	69	100 * 4
Atlas_sim_mt	156	5 * 4
CMS_gen_sim	42	20 * 4
CMS_digi	31	50 * 4
CMS_reco	51	50 * 4
Belle2_gen_sim_reco	25	50 * 1
Alice_gen_sim_reco	194*	3 * 4
LHCb_gen_sim	104	5 * 1
Juno_gen_sim_reco	67	50 * 1
Gravitational Wave	138	1 * 4
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

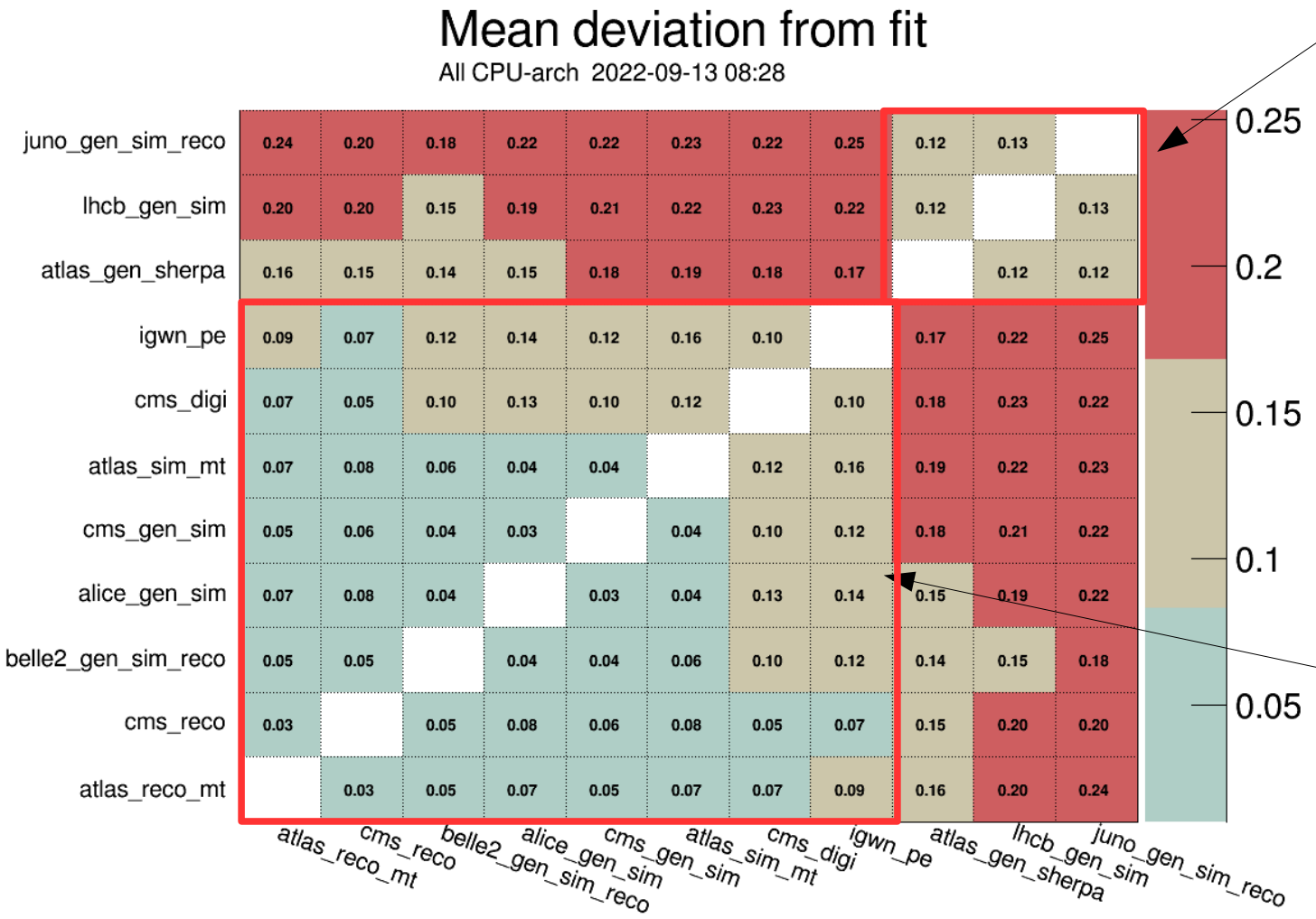
Considerations for removal: length, different running conditions: GW doesn't saturate node, Juno requires > 2GB/core

$HEPScore_{11}$: all workloads

$HEPScore_{9}$: remove GW and Juno

* - Alice reco currently not included in benchmark score, due to technical problems with reco workload. Reco is ~ 50% of running time. Once issue is resolved, could run only reco to shorten workload length.

Workload Correlations



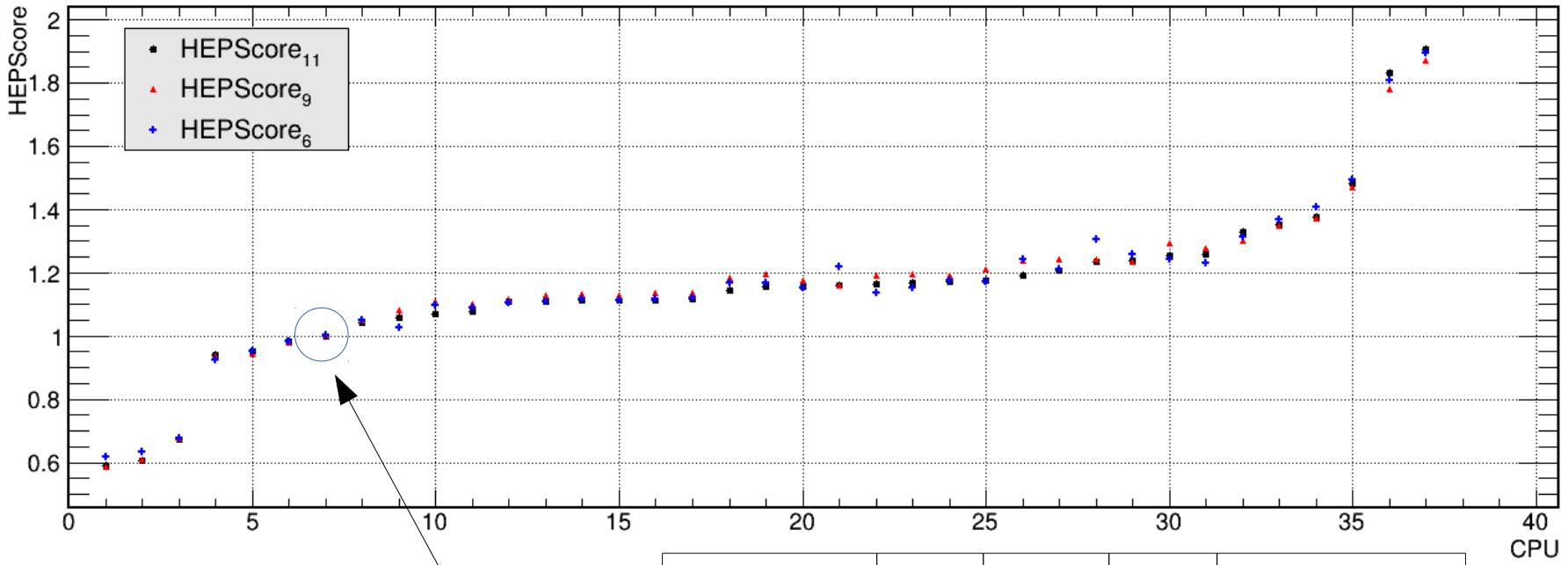
Pick two from here

Pick four from here

Many workloads highly correlated with each other: no need to include all

HEPScore$_6$: LHCb_gen_sim, ATLAS_gen_sherpa, CMS_gen_sim, CMS_reco, ATLAS_reco, Belle2_gen_sim_reco

HEPScore Candidates



	Min	Mean	Max	Run time (m)
$HEPScore_9$	0.97	1.01	1.04	703
$HEPScore_6$	0.97	1.01	1.06	322

CPUs ordered by increasing $HEPScore_{11}$ score

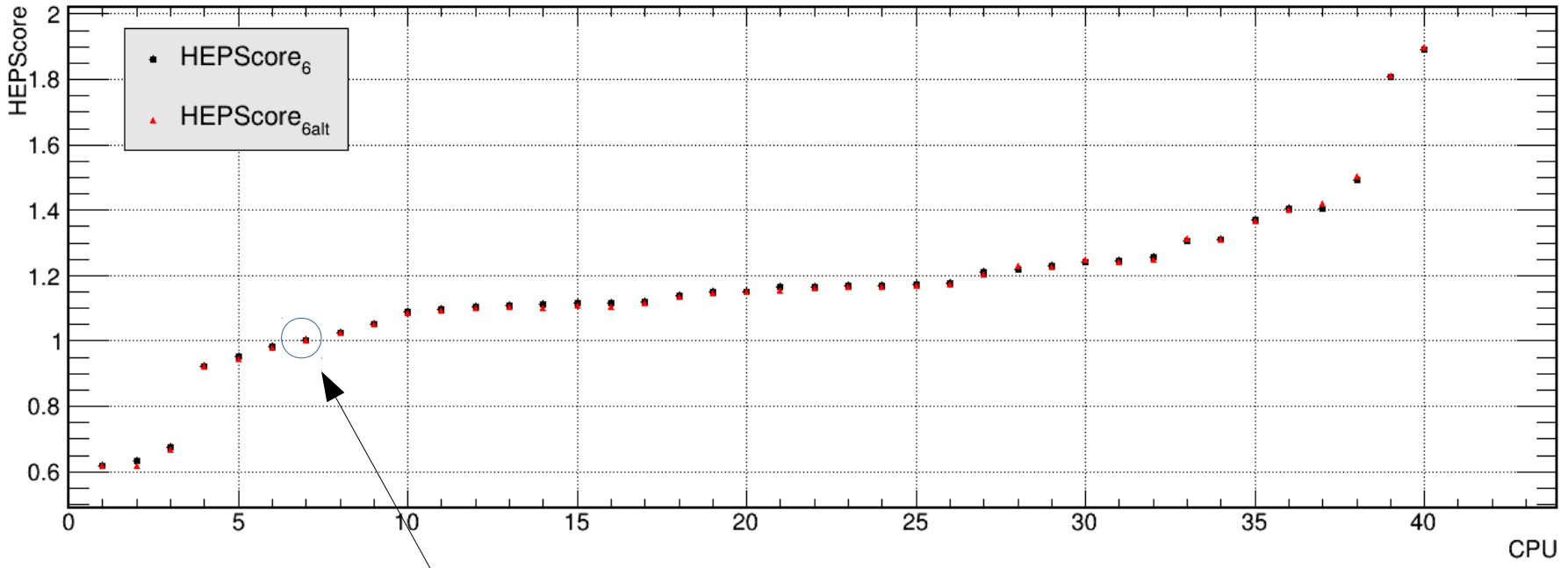
Difference between HEPScore candidates is very small

Table shows ratio to $HEPScore_{11}$

Time is on reference machine

Reference machine
(Intel Xeon E5-2650 v4)

Alternative HEPScore$_6$



HEPScore$_{6alt}$ includes Alice_gen_sim instead of CMS_gen_sim

Table shows ratio to regular HEPScore$_6$

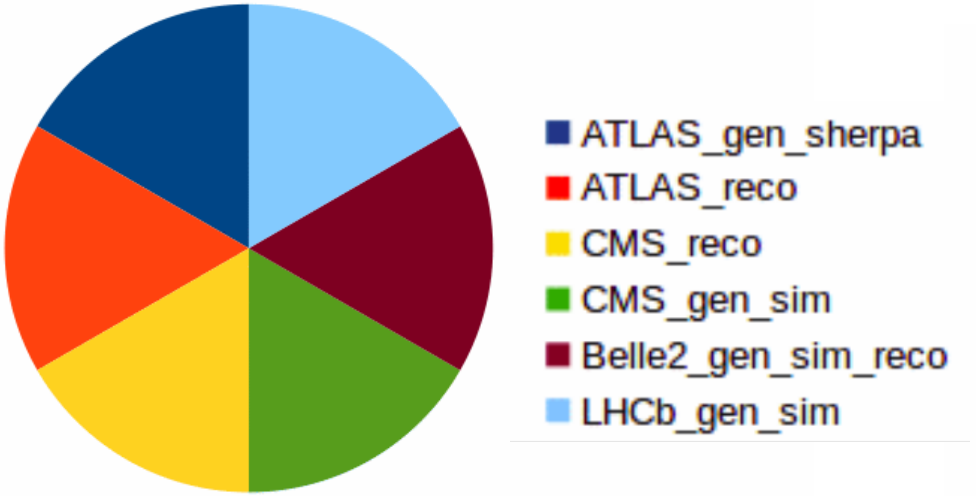
Very little impact, large increase in running time

	Min	Mean	Max	Run time (m)
HEPScore$_{6alt}$	0.98	1.0	1.01	474

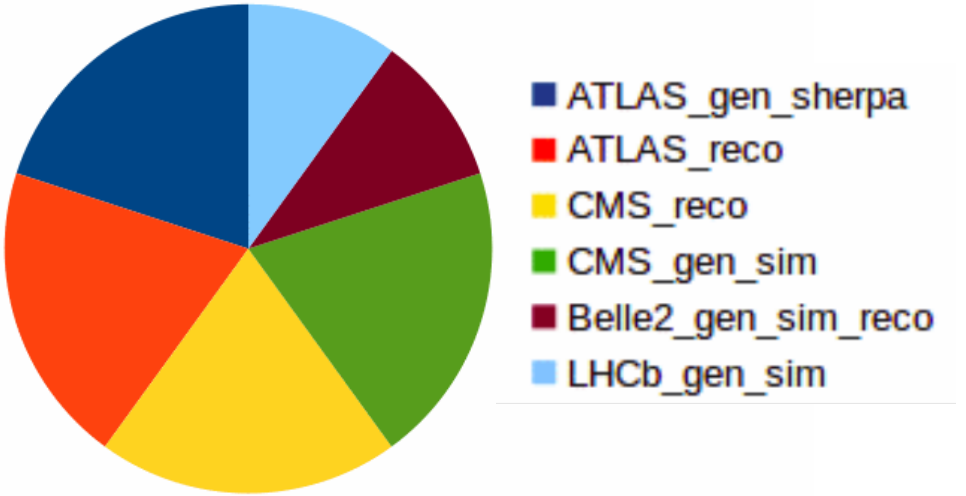
Reference machine
(Intel Xeon E5-2650 v4)

Weighting Candidates

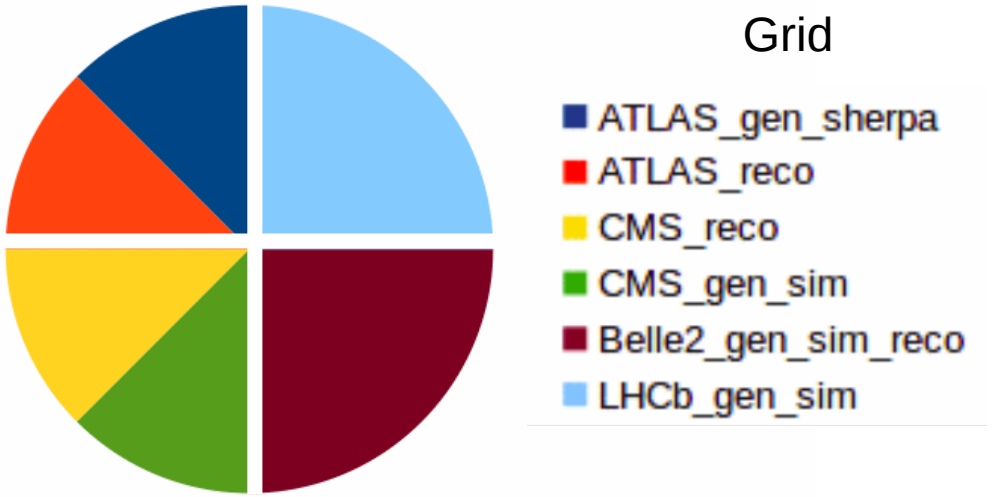
After removing GW, Juno, Atlas_sim_mt, Alice, CMS_digi



Nominal

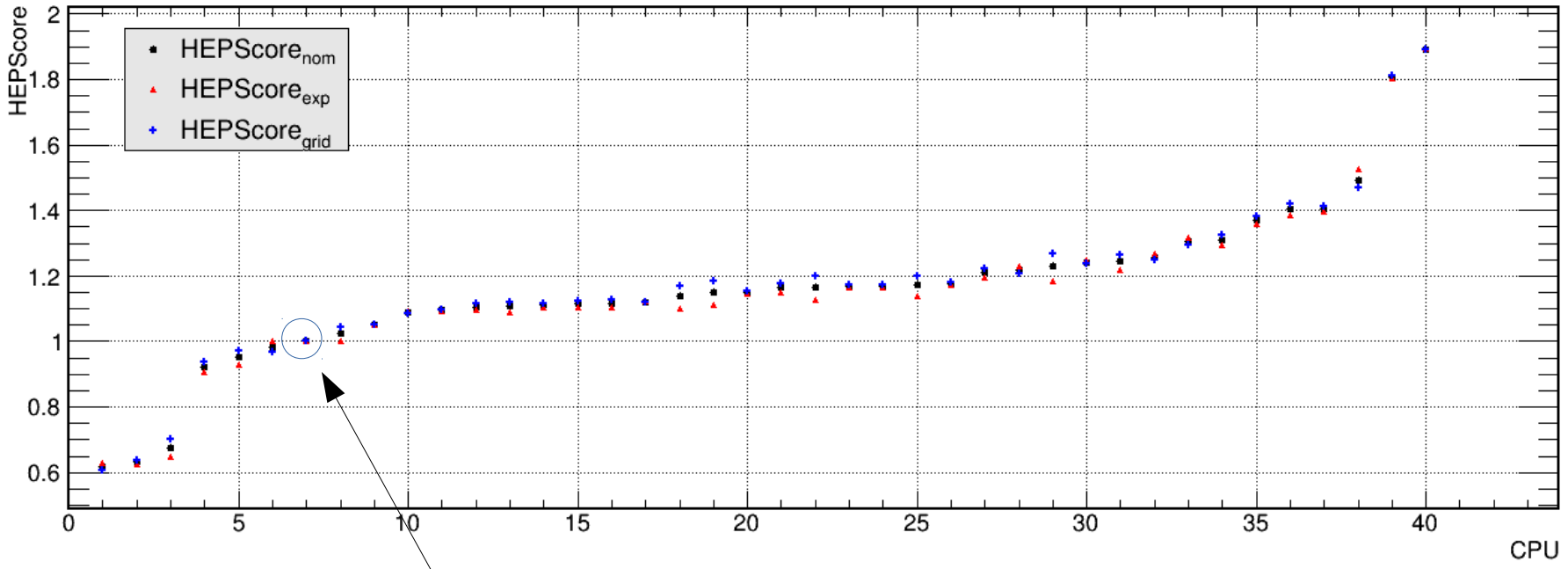


Grid



Experiment

Weighting Candidates



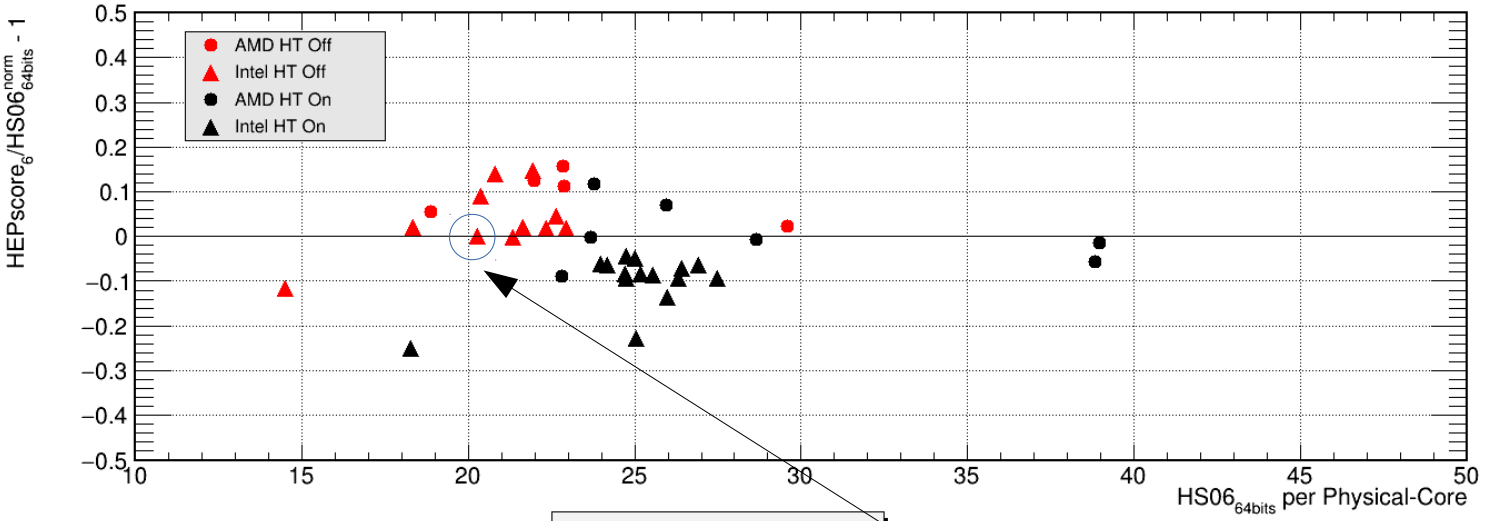
Again, difference between candidates is very small

Table shows ratio to HEPScore$_6$ with nominal weights

	Min	Mean	Max
Grid	0.98	1.01	1.04
Experiment	0.96	0.99	1.02

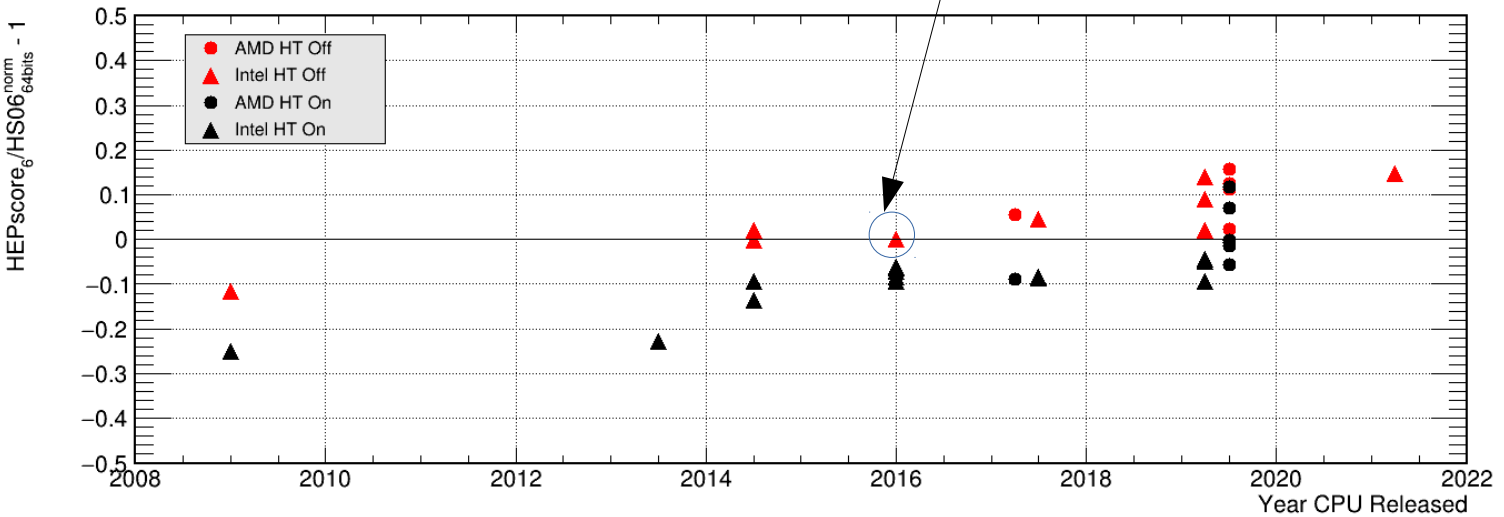
Reference machine
(Intel Xeon E5-2650 v4)

HS06 Comparison



FOM = 0.078

Reference machine
(Intel Xeon E5-2650 v4)



FOM is average distance from X axis

Nominal weights used

Increasing performance of HEPScore compared to HS06 highlights the need for a new benchmark!

Conclusion

- HEPScore is not sensitive to reasonable choices of workloads, weights
- Several considerations: keep the number of benchmarks as small as possible (Domenico's talk), ensure sufficient coverage of workload behaviour, keep running time reasonable
- HEPScore_6 fulfills these requirements
- Weighting workloads equally is simplest and not significantly different from other choices
- Other candidates can be tried, but differences likely to be small
- Could add Alice_reco workload when it is ready

Backup Slides

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 E5-2650 v4, hyperthreading (HT) off

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

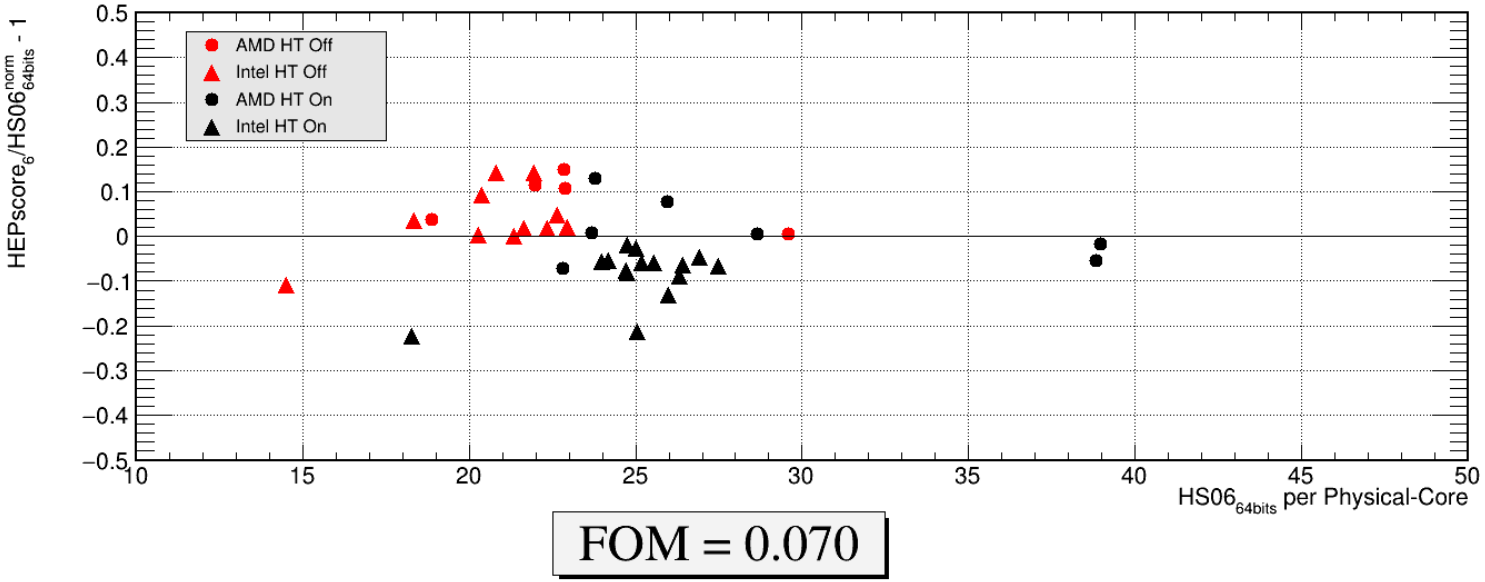
	AMD_7551P, HT on	Ref. Machine
Physical cores	32	16
Year	2017	2014
Workload avg.	700.7	361.5
HEPScore	1.94	1
$HEPScore_n$	0.97	1
HS06_64	729.9	415.5
HS06_64/ref	1.76	1
$HS06\_64_n$/ref	0.88	1

$HEPScore_n$ 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)

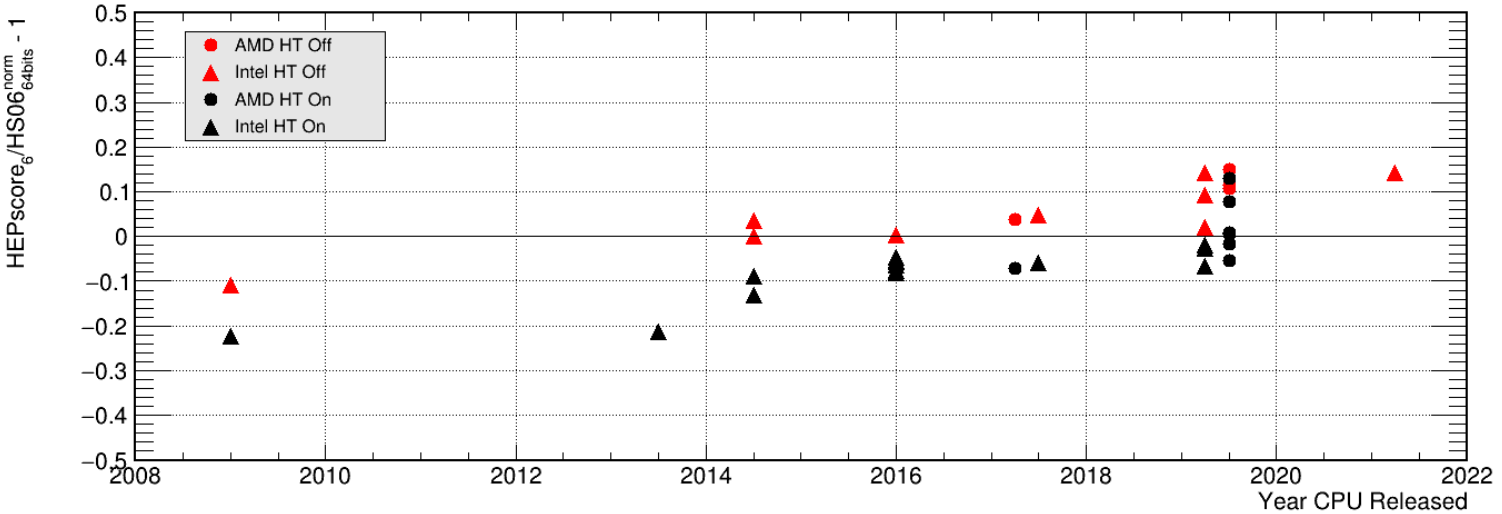
Ratio of ~1.2 between HT on and HT off is typical. Here it is 1.2 for HS06_64, and 1.1 for HEPScore

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

HS06 Comparison, Grid Weighting

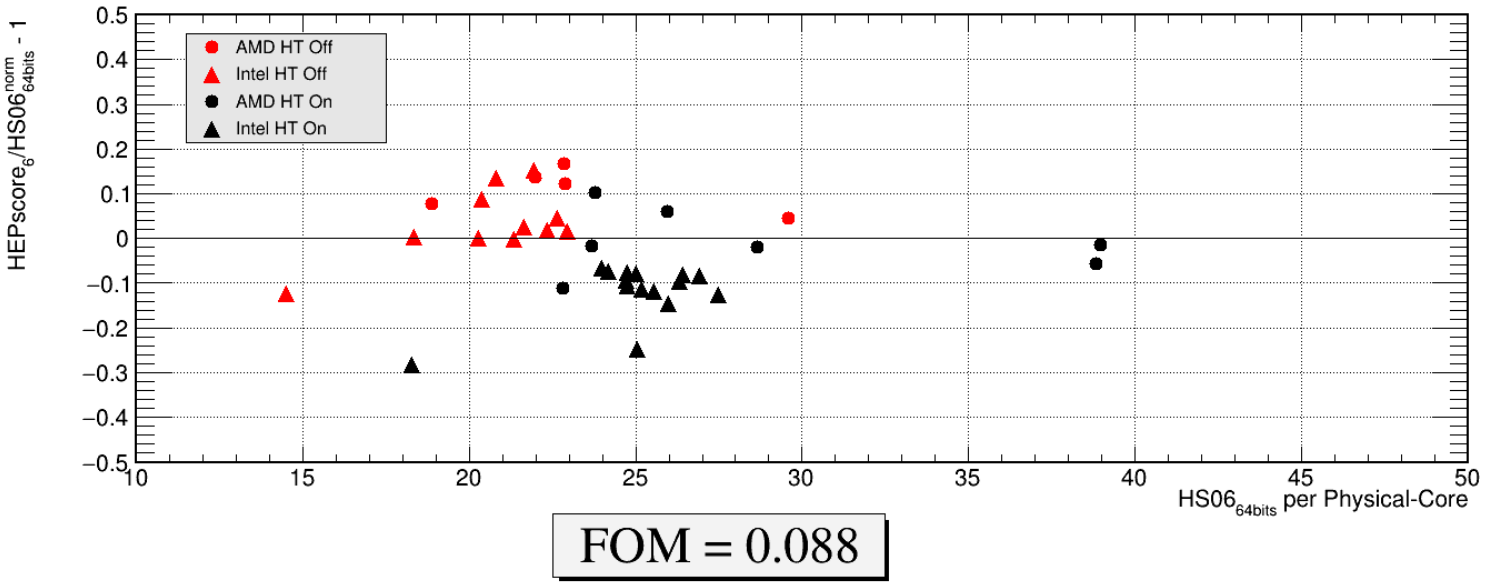


FOM = 0.070



FOM is average distance from X axis

HS06 Comparison, Exp. Weighting



FOM is average distance from X axis