HEP Score and HEP Benchmark Suite
“How to use HEP Workloads as benchmarks”

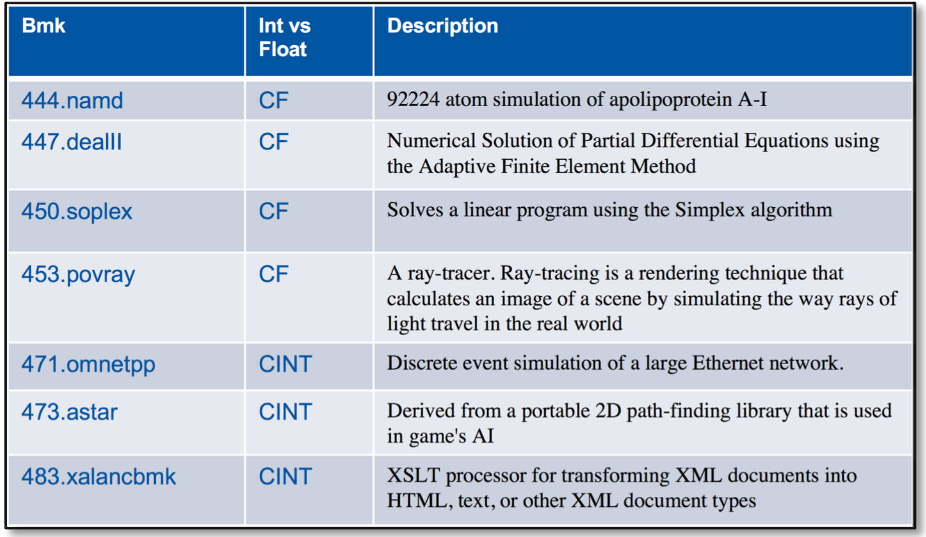
D. Giordano (CERN IT-CM)

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

pre-GDB - Benchmarking
8 Oct 2019

A reminder about HS06

- Subset of SPEC CPU® 2006 benchmark
  - SPEC's industry-standardized, CPU-intensive benchmark suite, stressing a system’s processor, memory subsystem and compiler



Bmk	Int vs Float	Description
444.namd	CF	92224 atom simulation of apolipoprotein A-I
447.dealII	CF	Numerical Solution of Partial Differential Equations using the Adaptive Finite Element Method
450.soplex	CF	Solves a linear program using the Simplex algorithm
453.povray	CF	A ray-tracer. Ray-tracing is a rendering technique that calculates an image of a scene by simulating the way rays of light travel in the real world
471.omnetpp	CINT	Discrete event simulation of a large Ethernet network.
473.astar	CINT	Derived from a portable 2D path-finding library that is used in game's AI
483.xalancbmk	CINT	XSLT processor for transforming XML documents into HTML, text, or other XML document types

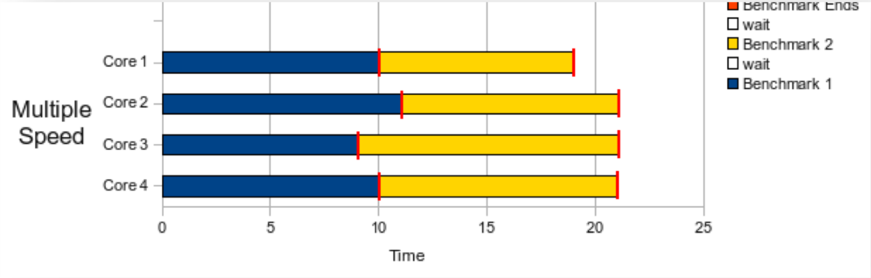
- HS06 is suite of 7 C++ benchmarks
  - In **2009,** proven **high correlation** with experiment workloads

- The overall HS06 score is the **geometric mean** of the 7 individual benchmark scores

HS06 score computation

- ❑ For each available core the sequence of benchmarks runs **3 times**
  - Each core sequence is independent (potential **time misalignment**)
  - Multiple-Speed approach

- ❑ For each core and benchmark, the **median** value of the **3 measurements** is taken, and a ratio respect to a **reference value** is computed



- ❑ Compute the **geometric mean** of the ratio values (per core)

- ❑ HS06 score = **sum of the geometric means** across cores

Why the geometric mean?

- Ratio of scores ⇔ average relative performance
  - Geometric mean of ratios == ratio of geometric means

- Example:
  - T_i^r : is the speed of running WL (i) in reference compute node (r)
  - T_i^C : is the speed of running WL (i) in a compute node (C)
  - T_i^G : is the speed of running WL (i) in a compute node (G)
  - λ_i is the relative weight of a given WL
    - If all are equal it can be 1 for all

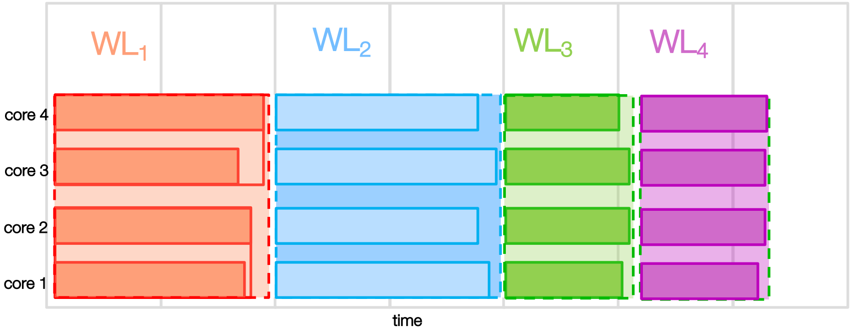
$$S^{C,r} = e^{\frac{\sum \lambda_i \ln({T_i^C}/{T_i^r})}{\sum \lambda_i}}$$

$$S^{G,r} = e^{\frac{\sum \lambda_i \ln({T_i^G}/{T_i^r})}{\sum \lambda_i}}$$

$$S^{G,C} = e^{\frac{\sum \lambda_i \ln({T_i^G}/{T_i^C})}{\sum \lambda_i}} = S^{G,r}/S^{C,r}$$

- N.B.: that would be valid also with (G) being a node with GPU to be compared with a node with CPU only

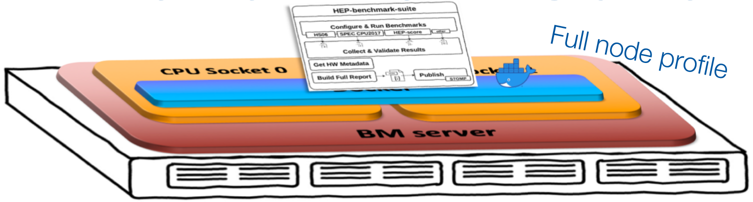
HEP score concepts



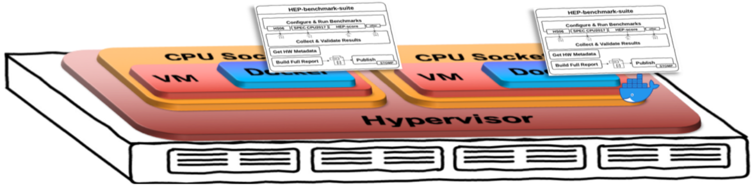
- ❑ HEP-score triggers HEP Workloads' runs in sequence
  - A **container** per WL
  - **3** times per WL, in sequence, and the **median** WL score is retained
- ❑ Each container runs the Experiment executable with a configurable number of threads (MT) or processes (MP)
- ❑ The available cores are saturated spawning a ***computed*** number of parallel copies
- ❑ The **score** of each WL is the cumulative event throughput of the running copies
  - When possible the initialization and finalization phases are excluded from the computation
  - Otherwise a long enough sequence of events is used
- ❑ A WL ***speed factor*** is computed as ratio of the WL score on a given machine w.r.t. the WL score obtained on a fixed reference machine
- ❑ **HEPscore** is the **geometric mean** of the WLs' ***speed factor***

Collecting HEPscores with HEP Benchmark Suite

- ❑ Resource partitioning done ‘*a priori*’
  - – By default profile all the available resources of the (virtual) machine

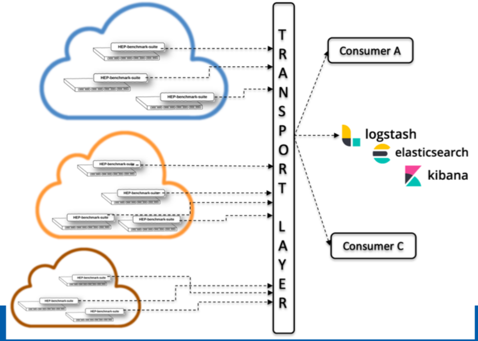


Full node profile



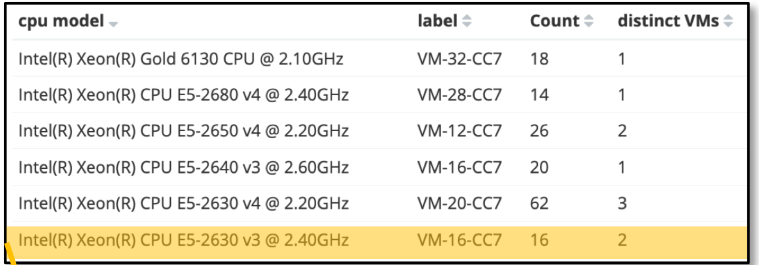
VMs with reserved CPU socket

- ❑ Data collection at CERN
  - – Single sockets of each CPU model are profiled
  - – VMs with fully reserved CPU sockets (½ node)

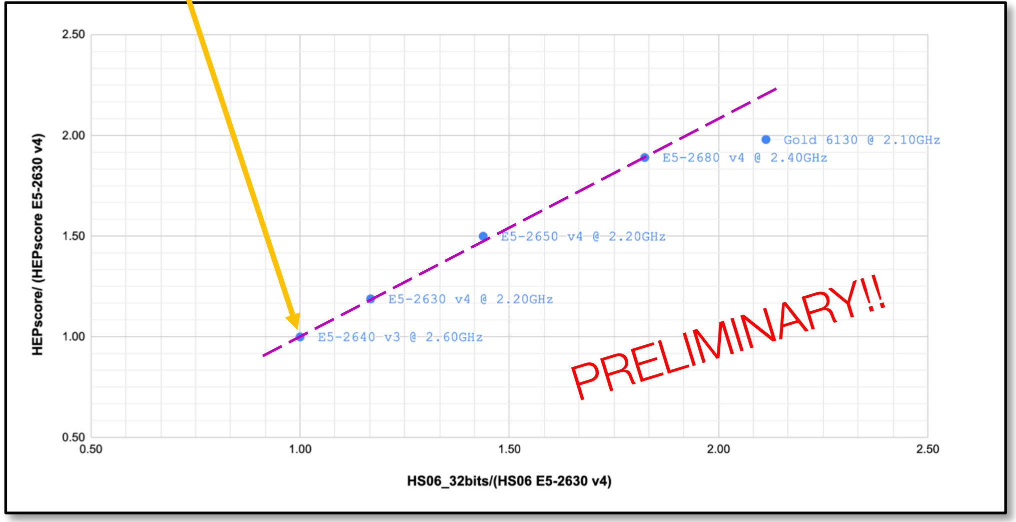


Very preliminary comparison with HS06

- ❑ Compare performance of a single CPU socket
- ❑ Take as reference
  - Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
- ❑ HEP-score consists of
  - CMS: gen-sim, digi, reco
  - Atlas: gen, sim
  - LHCb: gen-sim
  - Still does NOT include all WLs!!
- ❑ Hint of discrepancy for Skylake Gold?
  - To be further investigated



cpu model	label	Count	distinct VMs
Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz	VM-32-CC7	18	1
Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz	VM-28-CC7	14	1
Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz	VM-12-CC7	26	2
Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz	VM-16-CC7	20	1
Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz	VM-20-CC7	62	3
Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz	VM-16-CC7	16	2



CERN