Next Generation of HEP CPU Benchmarks

Domenico Giordano (CERN)
on behalf of the HEPiX CPU Benchmarking WG

CHEP 2018 Conference, Sofia, Bulgaria

Performance measurement

From "*Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling*"

- by **Raj Jain** , Wiley Computer Publishing, John Wiley & Sons, Inc
- 1992 Computer Press Award Winner

"Performance is a key criterion in the design, procurement, and use of computer systems […] to get the highest performance for a given cost."

"The types of applications of computers are so numerous that it is not possible to have a standard measure of performance […] for all cases."

"The first step in performance evaluation is to select the right measures of performance, the right measurement environments, and the right techniques."

Select the right measures

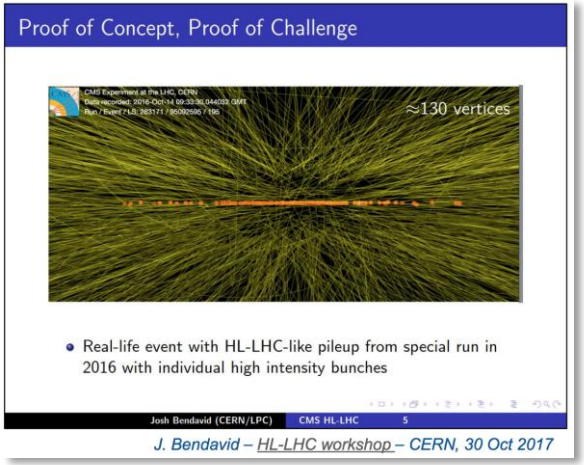
Typical HEP application consists of [CHEP18 Poster 72: Trident]

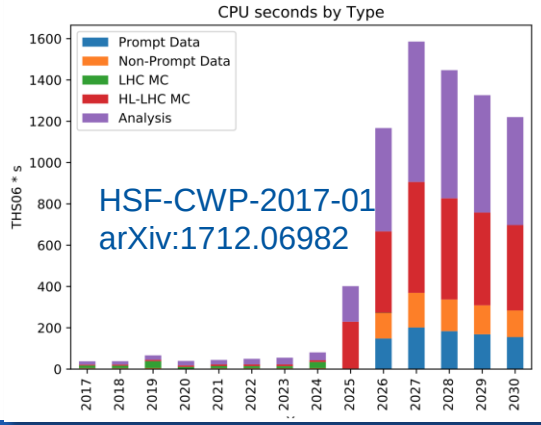
- A cluster of several hundred algorithms
- Complex framework interconnecting these algorithms
- Linear instruction spread (no hotspots)
- Non existence of classical numerical loops
- Average runtime of several hours

From 50 to 80% of WLCG CPU time spent in simulation

- CPU requirements will change and increase at HL-LHC
  - more data to process (higher luminosity)
  - more complex events (higher pileup)

Requirement:
the HEP benchmark must scale with
the average performance of the job mix running in WLCG





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

    <<CPP showed a good match with average lxbatch e.g. for FP+SIMD, Loads and Stores and Mispredicted Branches>> [*]
  - Execution time of the full HS06 suite: O(4h)
- Since 2009 HS06 has been used for
  - Performance studies
  - Procurement procedures
  - Pledges & Accounting

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

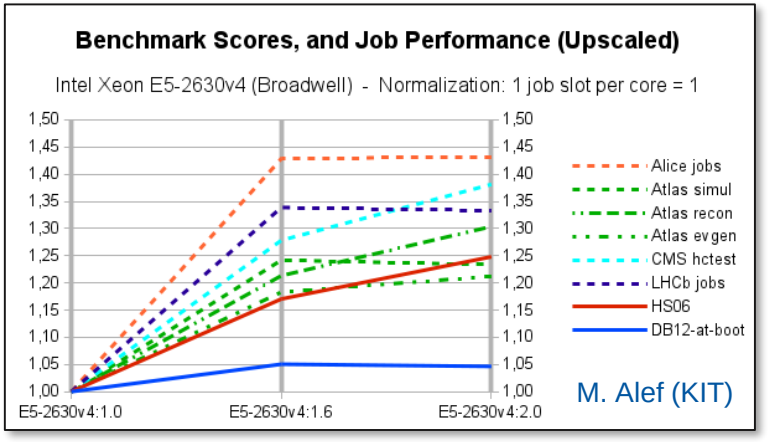
Correlation	Generation	Simulation	Reconstruction	Total
Atlas	0.9969	0.9963	0.9960	0.9968
Alice pp MinBias	0.9994		0.9832	0.9988
Alice PbPb	0.9984		0.9880	0.9996
LhcB	0.9987			
CMS HiggsZZ	0.9982		0.9987	0.9983
CMS MinBias	0.9982		0.9974	0.9974
CMS QCD 80 120	0.9988		0.9987	0.9988
CMS Single Electron	0.9987		0.9942	0.9981
CMS Single MuMinus	0.9986		0.9926	0.9970
CMS Single PiMinus	0.9955		0.9693	0.9955
CMS TTbar	0.9985		0.9589	0.9987

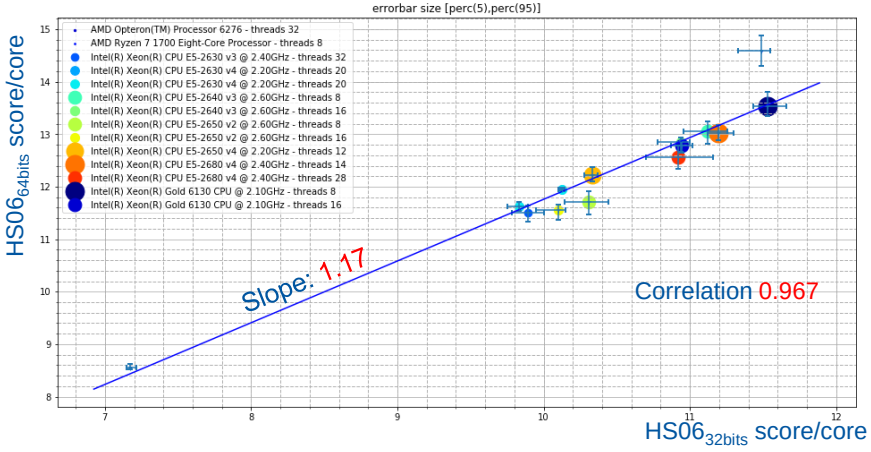
[*] Correlation of HEP-SPEC06 with several kinds of applications and different experiments

[*] "A comparison of HEP code with SPEC benchmarks on multi-core worker nodes"
J. Phys.: Conf. Ser. 219 (2010) 052009
CHEP-09

HS06 & recent CPU models

- Reported by Alice and LHCb that their workloads did not scale anymore with HS06
  - *J. Phys. : Conf. Ser.* 898 (2017) 082011
- Independent studies still show agreement within 10% for Atlas and CMS workloads
  - CPU benchmarking with production jobs
- Some “understood” differences:
  - HS06 compiled at 32 bits, whereas experiment applications at 64 bits (10%-20% effect) *[ref]*





New emerging scenarios

Replace HS06 by an equally accurate benchmark suite

- SPEC CPU® 2006 is being retired by SPEC, replaced by CPU® 2017
- Prepare for the future changes
  - Evolution of experiment software and better usage of the full CPU potential

Use fast benchmarks in some specific cases

- Commercial Cloud and HPC opportunistic resources
  - Assessment of the delivered performance & forecasting the job slot duration
  - Contexts where changing conditions require prompt feedback but not necessary high accuracy

SPEC CPU2017

SPEC releases major new CPU benchmark suite

The SPEC CPU2017 benchmark suite features updated and improved workloads, use of OpenMP to accommodate more cores and threads, and optional metric for measuring power consumption

Gainesville, Va., June 20, 2017 -- The Standard Performance Evaluation Corp. (SPEC) today released the SPEC CPU2017 benchmark suite, an all-new version of the non-profit group's software for evaluating compute-intensive performance across a wide range of hardware systems.

The SPEC CPU2017 benchmark suite is the first major update of the worldwide standard CPU performance evaluation software in more than 10 years. The new suite includes updated and improved workloads with increased size and complexity, the use of OpenMP to allow performance measurement for parallelized systems with multiple cores and threads, and an optional metric for measuring power consumption.

Current SPEC CPU subcommittee members include AMD, ARM, Dell, Fujitsu, HPE, IBM, Inspur, Intel, Nvidia and Oracle.

https://www.spec.org/cpu2017/press/release.html

https://www.spec.org/cpu2017/Docs/index.html#benchmarks

The Benchmarks

SPEC CPU2017 has 43 benchmarks, organized into 4 suites:

SPECrate 2017 Integer — SPECspeed 2017 Integer

SPECrate 2017 Floating Point — SPECspeed 2017 Floating Point

Benchmark pairs shown as:

`5nn.benchmark_r / 6nn.benchmark_s`

are similar to each other. Differences include: compile flags; workload sizes; and run rules. See: [OpenMP] [memory] [rules]

SPECrate 2017 Integer	SPECspeed 2017 Integer	Language [1]	KLOC [2]	Application Area
500.perlbench_r	600.perlbench_s	C	362	Perl interpreter
502.gcc_r	602.gcc_s	C	1,304	GNU C compiler
505.mcf_r	605.mcf_s	C	3	Route planning
520.omnetpp_r	620.omnetpp_s	C++	134	Discrete Event simulation - computer network
523.xalancbmk_r	623.xalancbmk_s	C++	520	XML to HTML conversion via XSLT
525.x264_r	625.x264_s	C	96	Video compression
531.deepsjeng_r	631.deepsjeng_s	C++	10	Artificial Intelligence: alpha-beta tree search (Chess)
541.leela_r	641.leela_s	C++	21	Artificial Intelligence: Monte Carlo tree search (Go)
548.exchange2_r	648.exchange2_s	Fortran	1	Artificial Intelligence: recursive solution generator (Sudoku)
557.xz_r	657.xz_s	C	33	General data compression

SPECrate 2017 Floating Point	SPECspeed 2017 Floating Point	Language [1]	KLOC [2]	Application Area
503.bwaves_r	603.bwaves_s	Fortran	1	Explosion modeling
507.cactuBSSN_r	607.cactuBSSN_s	C++, C, Fortran	257	Physics: relativity
508.namd_r		C++	8	Molecular dynamics
510.parest_r		C++	427	Biomedical imaging: optical tomography with finite elements
511.povray_r		C++, C	170	Ray tracing
519.lbm_r	619.lbm_s	C	1	Fluid dynamics
521.wrf_r	621.wrf_s	Fortran, C	991	Weather forecasting
526.blender_r		C++, C	1,577	3D rendering and animation
527.cam4_r	627.cam4_s	Fortran, C	407	Atmosphere modeling
	628.pop2_s	Fortran, C	338	Wide-scale ocean modeling (climate level)
538.imagick_r	638.imagick_s	C	259	Image manipulation
544.nab_r	644.nab_s	C	24	Molecular dynamics
549.fotonik3d_r	649.fotonik3d_s	Fortran	14	Computational Electromagnetics
554.roms_r	654.roms_s	Fortran	210	Regional ocean modeling

[1] For multi-language benchmarks, the first one listed determines library and link options (details)

[2] KLOC = line count (including comments/whitespace) for source files used in a build / 1000

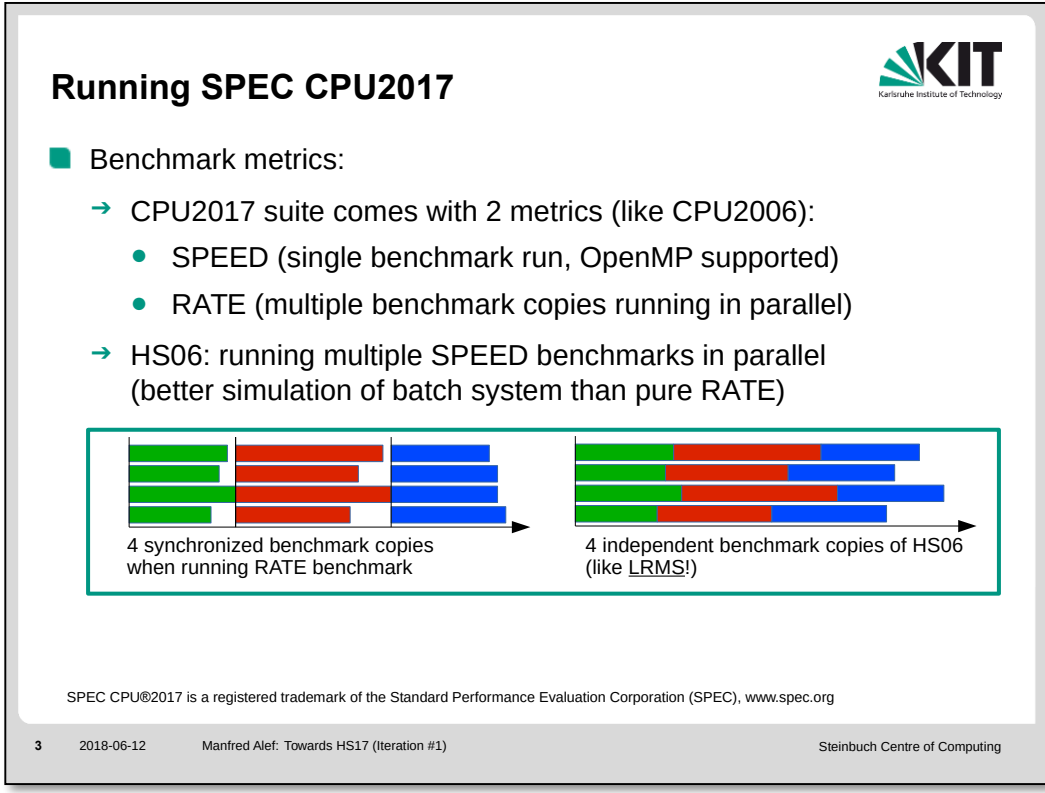
Larger suite, more complex code, shaped for multi-core and multi-threads

same application area as in HS06

Defining the working point

Different configurations to be investigated

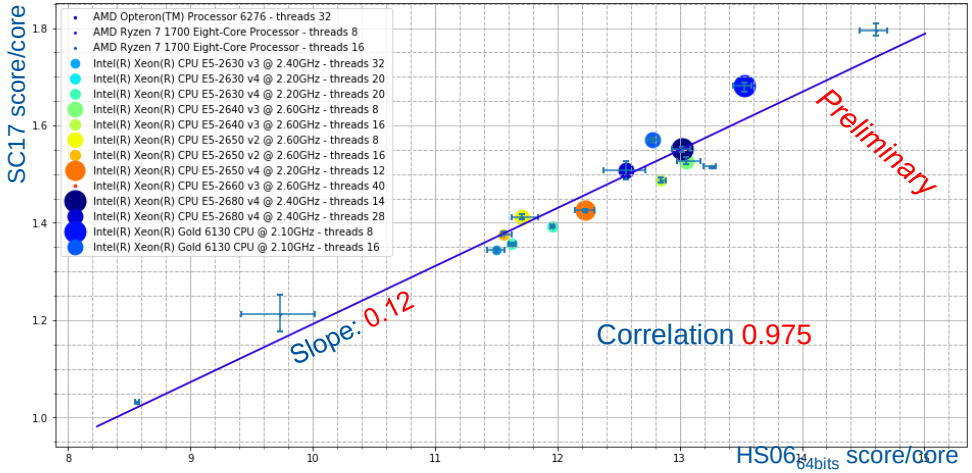
- Running mode: as in HS06 or as multiple ***rate*** runs
- List of benchmarks
  - All (23)
  - Only C++ benchmarks (8)
  - Selecting the sub-set best matching the HEP workloads
- Compiler flags
  - So far: -g -O3 -fPIC -pthread



Are SPEC CPU®2017 and HS06 correlated?

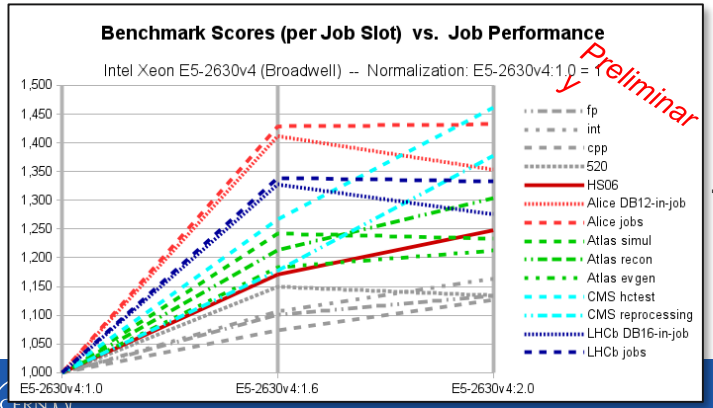
Very high correlation (0.975) between SPEC CPU®2017 score and HS06 (64 bits)

- Measured on 7 different Intel CPU models, 1 AMD Opteron and 1 (Desktop) AMD Ryzen
- SMT on and off
- NB in the plot
  - error bars are [5%,95%] values
  - Marker size ⇔ amount of data collected



Residuals ratio of the linear fit

- |extr./meas.-1| < 5%



- Scaling factor respect to the number of running slots/core is less representative of the HEP workloads
  - Caveat: study (M. Alef KIT) done on few CPU models so far

Are the individual benchmarks independent?

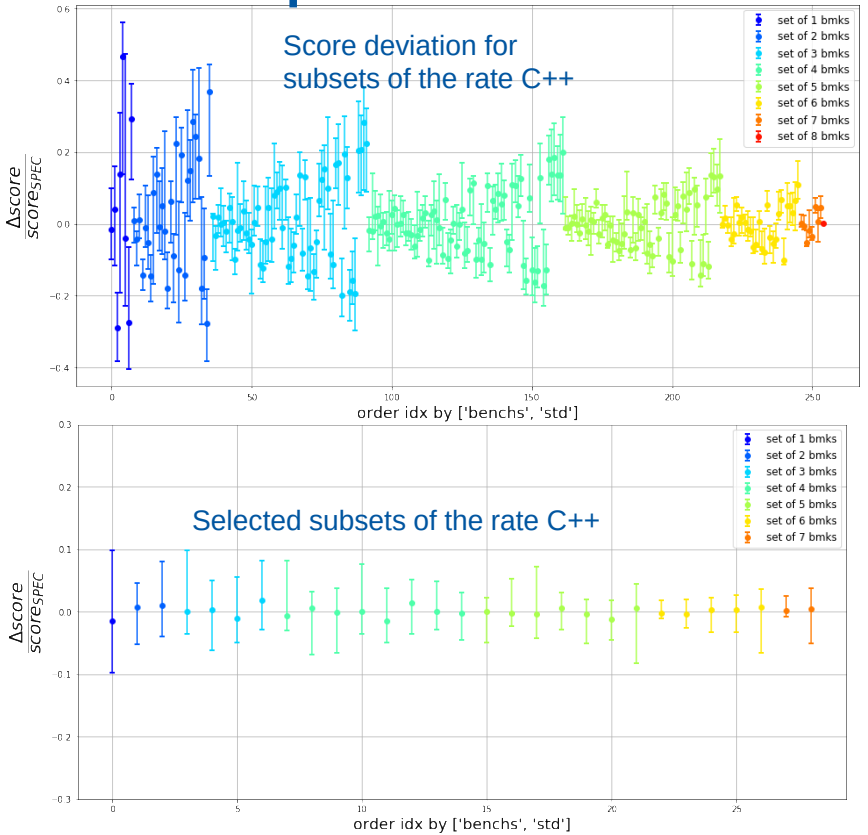
Are all SPEC CPU® 2017 benchmarks needed?

- Less benchmarks => Shorter runtime
  - Currently a run of the 8 C++ benchmarks takes <2.5hour/iteration> in the 7 tested CPU models
- Better control of benchmark score Vs HEP job mix

Found subsets of the **rate** C++ still well representative of the full performance score

- Example:
  - 508.namd_r , 520.omnetpp_r, 526.blender_r Discrepancy
    - » max= 0.06
    - » mean= -0.003 ± 0.004

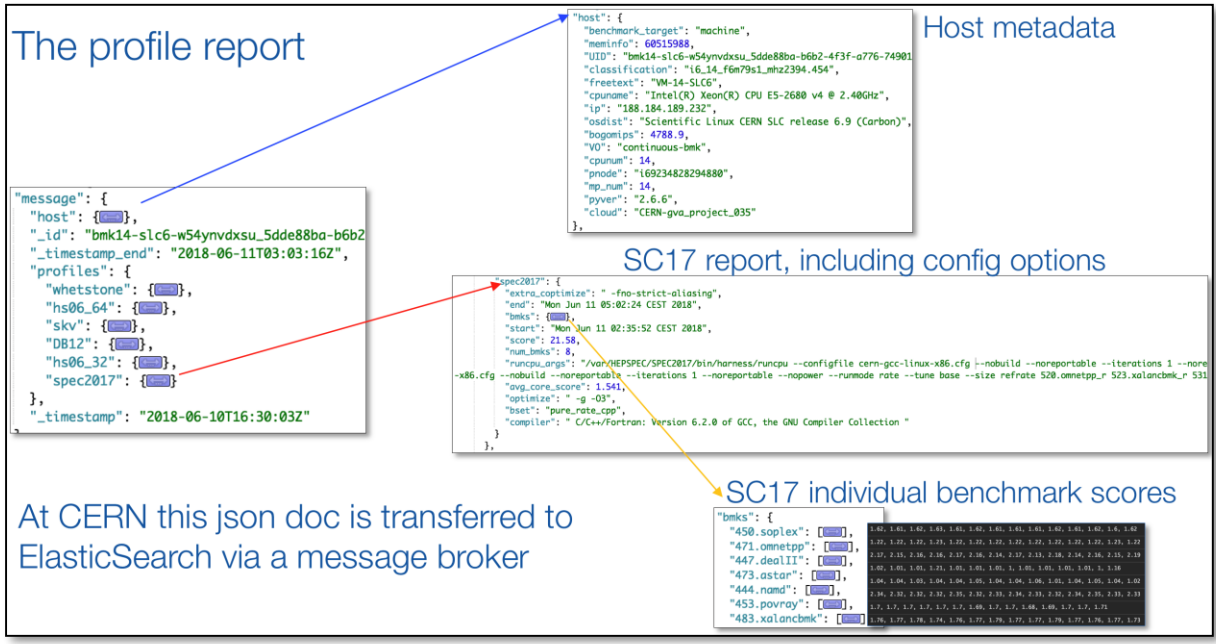
Limitation of the study: focusing on x86 arch, mainly Intel CPUs



Toward a common tool to run SPEC CPU2017

Script to trigger SC17

- Very similar to the HS06 script (runspec.sh)
- Produces json output with results of each running benchmark
  - Includes configuration information
- Enables the sharing/comparison of the measurements

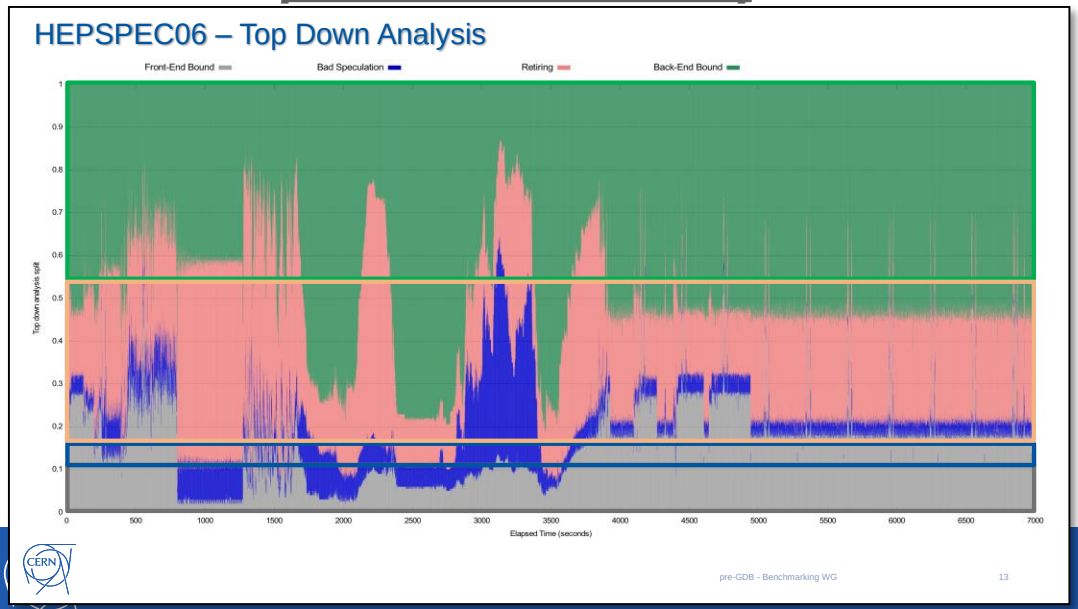


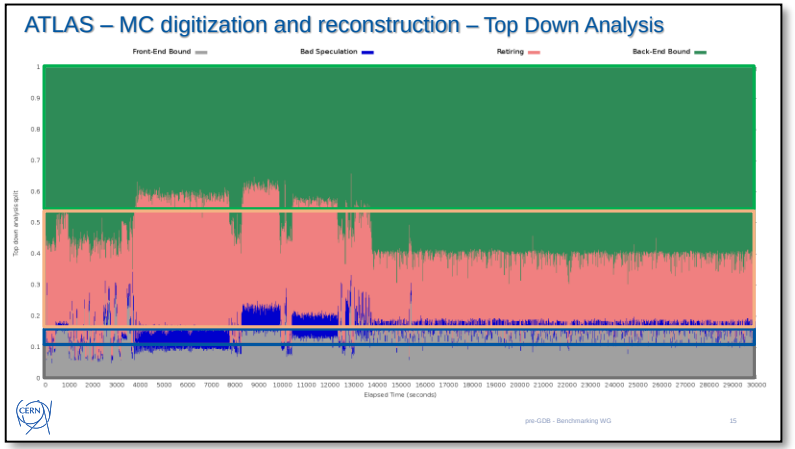
Compare applications by IPC and memory usage

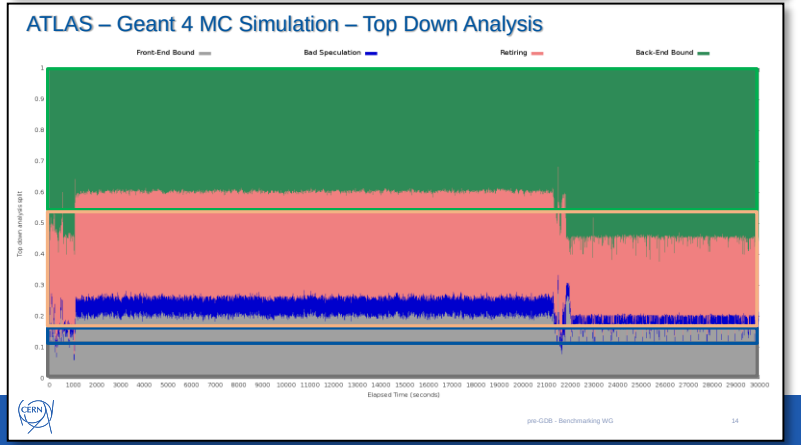
Unveil the (dis)-similarities between HEP workloads and proposed benchmarks (HS06, SPEC CPU2017,…)

- Percentage of time spent in Front-End Vs Back-End Vs Bad speculation (Instructions per Cycle)
- Memory transactions and bandwidth usage

More details in [CHEP18 Poster 72: Trident]







Fast benchmarks investigations

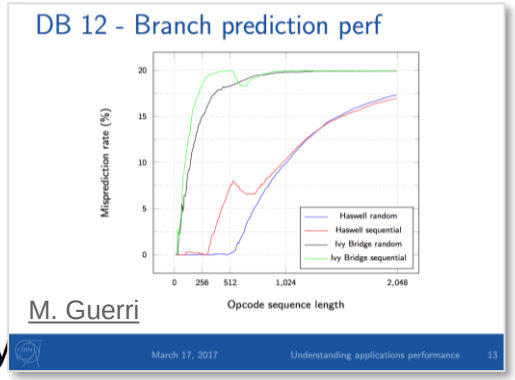
Among several applications proposed and studied, two are the preferred by the WLCG collaborations

- ATLAS KV (KitValidation)
  - Mainly GEANT4. Default workload: 100 single muon event simulation
  - Well in agreement with Atlas and CMS simulation *[ref]*

- DIRAC Benchmark 2012 (DB12)
  - In agreement with Alice and LHCb job performance when DB12 runs at job time (single slot)

Not robust enough to replace long-running benchmarks

- DB12 dominated by the front-end call and branch prediction unit
- SMT is not beneficial at all for DB12 when loading all CPU threads
- KV shortness and event simplicity affected by systematics
  - Found in the performance studies of Meltdown/Spectre patches
  - Can be improved extending the test duration and event complexity



Conclusions

HS06 is a decade old suite used for benchmarking

- Well know by vendors, site managers, funding agencies
- Stable, reproducible, accurate
- But is reaching end of life

Looking for future alternatives in HEP

- SPEC CPU 2017
  - Preliminary studies do not show much advantage respect to the HS06
  - Suite of C++ benchmarks score highly correlated with HS06 score
- Fast benchmarks can play a role in cloud contexts, where re-benchmarking is needed
  - But the current fast benchmark cannot replace HS06 in procurement and accounting tasks

A suite of HEP workloads could be an alternative to industrial standard benchmarks

- Provided that distribution, maintenance and license issues are properly sorted out
- Work in progress