

## A comparison of HEP code with SPEC<sup>1</sup> benchmarks on multi-core worker nodes

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2010 J. Phys.: Conf. Ser. 219 052009

(<http://iopscience.iop.org/1742-6596/219/5/052009>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 194.36.1.2

The article was downloaded on 07/06/2011 at 09:34

Please note that [terms and conditions apply](#).

## A Comparison of HEP code with SPEC<sup>1</sup> benchmarks on multi-core worker nodes

**Michele Michelotto<sup>1</sup>, Manfred Alef<sup>2</sup>, Alejandro Iribarren<sup>3</sup>, Helge Meinhard<sup>3</sup>, Peter Wegner<sup>4</sup>, Martin Bly<sup>5</sup>, Gabriele Benelli<sup>3</sup>, Franco Brasolin<sup>6</sup>, Hubert Degaudenzi<sup>3</sup>, Alessandro De Salvo<sup>7</sup>, Ian Gable<sup>8</sup>, Andreas Hirstius<sup>3</sup>, Peter Hristov<sup>3</sup>**

1) INFN Padova, via F. Marzolo 8 35131 Padova Italy

2) Forschungszentrum Karlsruhe GmbH, Steinbuch Centre for Computing, Hermann-von-Helmholtz-Platz 1, D-76344 Eggenstein-Leopoldshafen

3) CERN CH-1211 Genève, Switzerland

4) DESY Platanenallee 6, D-15738 Zeuthen, Germany

5) RAL Rutherford Appleton Laboratory, Harwell Science and Innovation Campus, Didcot Oxfordshire, OX11 0QX UK

6) INFN Bologna, viale Berti-Pichat 6/2 40127 Bologna, Italy

7) INFN Roma, Piazzale Aldo Moro 2, 00185, ROMA, Italy

8) Dept. of Physics & Astronomy, University of Victoria, PO Box 3055, STN CSC, Victoria, BC, V8W 3P6 Canada

**Abstract.** The SPEC[1] CINT benchmark has been used as a performance reference for computing in the HEP community for the past 20 years. The SPECint\_base2000 (SI2K) unit of performance has been used by the major HEP experiments both in the Computing Technical Design Report for the LHC experiments and in the evaluation of the Computing Centres. At recent HEPiX[3] meetings several HEP sites have reported disagreements between actual machine performances and the scores reported by SPEC.

Our group performed a detailed comparison of Simulation and Reconstruction code performances from the four LHC experiments in order to find a successor to the SI2K benchmark. We analyzed the new benchmarks from SPEC CPU2006 suite, both integer and floating point, in order to find the best agreement with the HEP code behaviour, with particular attention paid to reproducing the actual environment of HEP farm i.e., each job running independently on each core, and matching compiler, optimization, percentage of integer and floating point operations, and ease of use.

---

<sup>1</sup> SPEC is a trademark of the Standard Performance Evaluation Corporation

## 1. Introduction

HEP requires huge quantities of computing power to process, simulate and analyze the vast amount of data produced in the modern experiments. A benchmark is needed to express the computing power requested by the experiments and the computing power offered by the computing centers. In the past HEP used some versions of the SPEC CINT synthetic benchmark.

A good benchmark should be able to predict the performance of HEP code on given machines. The performance when running the benchmark must be correlated to the performance observed on the real HEP application.

HEPiX is an international group, often interested in benchmarking topics, of HEP computing users and system administrators working in cooperating High Energy Physics institutions to share information. HEPiX meets every six months to discuss issues relating the HEP computing community. From approximately 2005, several presenters at HEPiX meetings had noted discrepancies between HEP application performance and the standard benchmark SI2K (SPECint\_base2000) scores. A consensus emerged within HEPiX that SI2K was no longer an appropriate measure of CPU performance for HEP applications.

It was also noted by HEPiX that the definition of SI2K itself was not clear, since some sites made their own measurements in their own environments while others were taking published SI2K scores from the official SPEC site ([www.spec.org/cpu2000/results](http://www.spec.org/cpu2000/results)).

During the fall 2006 HEPiX Meeting at JLAB, the chair of IHEPCCC (now disbanded) requested that HEPiX form a Working Group[6] to search for a standard benchmark suitable to the HEP community. Helge Meinhard of the CERN Laboratory was put forward as chair of the working group, and its members were selected from HEPiX computing experts and representatives from the major HEP experiments. This paper details the work of the HEPiX CPU benchmarking working group to find a standard benchmark appropriate to HEP, and the eventual selection of the HEP-SPEC06[7] benchmark.

## 2. History of HEP benchmarks

In the past several benchmarks have been used in the HEP world: Cern Unit, MIPS, and VUPS first, then several versions of SPEC CPU since about SPEC CPU92.

SPEC (Standard Performance Evaluation Corporation) is a non-profit organization that established, maintains and endorses a set of computer-related benchmarks.

SPEC CPU is designed to provide performance measurements that can be used to compare compute-intensive workloads on different computer systems.

## 3. SI2K

The version of SPEC CPU used in the Computing Technical Design Report (CTDR) of all the LHC experiments and, in general, used by HEP experiments is CINT2000, informally known as SI2K.

CPU2000 consists of two types of tests: INTEGER and FLOATING POINT. Each of these can be run in SPEED mode or RATE mode.

SPEED mode is meant to measure how fast the computer completes a single task whereas the throughput, capacity or RATE mode is meant to measure the performance of the whole machine, running in parallel on all the logical CPUs (a machine can have several processors, each of them can have several cores and each core can run several threads in parallel).

The computing centres also express their CPU power in terms of SI2K but several people used different definitions of SI2K.

Some centres and experiments used the value of SI2K published by SPEC while others took their own measurement, using the compiler and operating systems in production in their computing centre environment.

#### 4. Deviations of measured SI2K from the published results

The HEP applications are usually compiled with the gcc[4] open source compiler on the Scientific Linux[5] operating system. The measurements of SI2K on gcc/Linux are usually much lower than the published value. This would not be a problem if the two values would be proportional on all processor families and generations.

Since about 2005 several presentations at HEPiX conferences underlined that the measured results were lower and lower than the value published. The ratio was different for different families of x86 processors and for different set of compilation switches.

For example, the GridKa site reported that while in 2001 the value measured with gcc optimized with the switches: “-O3 -funroll\_loops -march=\$ARCH” where about 80% of the value published, in 2006 it was lower than 60%.

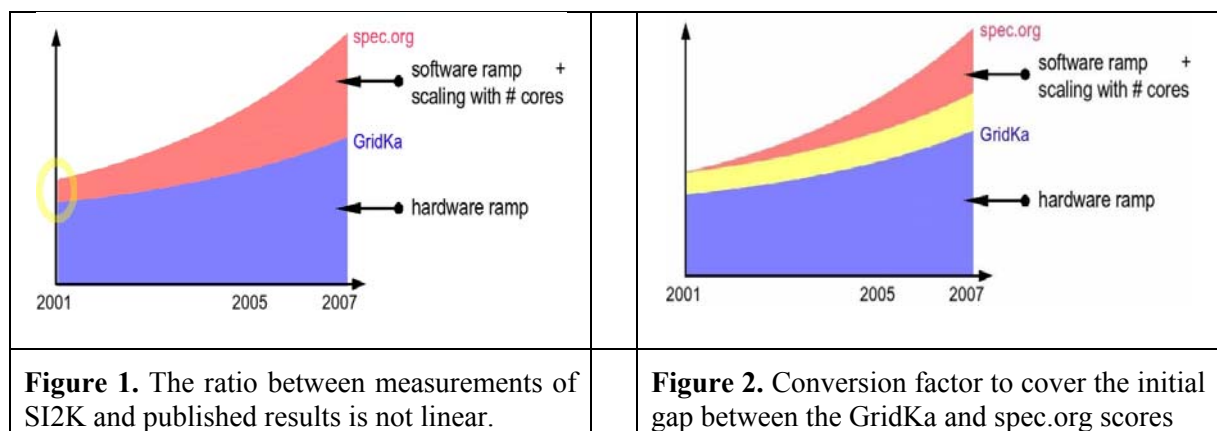


Figure 1 outlines the increasing gap between the benchmark scores. The blue curve shows the evolution according to the SPEC CPU2000 benchmarks which were run in a fixed operating environment. Multiple SPEED runs (see section 6) were performed in order to simulate the behavior of the batch system. The red curve is based on corresponding SPEED performance scores (SPECint\_base2000) published at [www.spec.org](http://www.spec.org). The hardware vendors may use the most efficient operating system, compiler, and optimizing flags which are available at the time when they run the benchmarks.

A comparison with performances of real HEP applications showed that the correct value to use was the gcc one.

This means that if one considers that the value in the CTDR is taken from the SPEC site, one should take the value measured and add 25% to have a real evaluation of the computing power of a machine. This is indicated by the yellow area in figure 2. WLCG decided to take the more conservative set of optimization switches mandated by the LCG Architects Forum: “-O2 -fPIC -pthread”, that gave roughly two thirds of the published results and proposed hence to increase that value by 50% to obtain the real computing power of a HEP worker node

This interim solution permitted a clear statement of a method to compute SI2K in the same way for the whole WLCG collaboration.

In 2006, HEPiX prompted the creation of a working group of experts from the major HEP Labs to propose a long term solution.

The working group, chaired by Helge Meinhard, worked in close collaboration with the WLCG[9] Grid Deployment Board (GDB), with support from the four LHC experiments.

The group decided to compare the performance of the CPU intensive HEP applications with available benchmarks, with a strong focus on SPEC CPU2006, the new release of the SPEC CPU suite that replaced SPEC CPU2000 in 2006.

## 5. SPEC CPU2006

The major difference between the SPEC CPU2000 and CPU2006 suites is that the new CPU2006 has a larger memory footprint, around 1GB for 32-bit applications, while CPU2000 was around 200MB.

This matches more closely the typical memory occupancy of a modern HEP application and the experiment requirement of at least 2GB RAM per core for LHC worker nodes.

Using CPU2000 in these systems would not exercise enough of the memory subsystem.

SPEC designed CPU2006 to provide a comparative measure of compute-intensive performance on a wide range of hardware using workloads developed from real user applications.

The current version of the benchmark suite is V1.1, released in June 2008.

The following tables describes the single benchmark in the SPEC CINT2006 suite and SPEC CFP2006 suite:

Test name	Language	Description
400.perlbench	C	Programming Language Derived from Perl
410.bzip2	C	Compression
403.gcc	C	C Compiler
429.mcf	C	Network simplex algorithm
445.gobmk	C	Artificial Intelligence, the game Go
456.hmmcr	C	Search Gene Sequence Protein
458.sjeng	C	Artificial Intelligence: Chess game
462.libquantum	C	Physics, Quantum Computing Simulation
464.h264ref	C	Video Compression with H.264 codec
471.omnetpp	C++	Discrete Event Simulation
473.astar	C++	Path Finding Algorithms
483.xalancbmk	C++	XML document processing

The benchmarks in the Integer Suite, the source language and a brief description

Test name	Language	Description
410.bwaves	Fortran	Fluid Dynamics – 3D viscous flow simulation
416.gamess	Fortran	Quantum Chemistry
433.milc	C	Physics and Quantum Chromodynamics
434.zeusmp	Fortran	Physics: Computational Fluid Dynamics
435.gromacs	Fortran	Biochemistry, Molecular Dynamics
436.cactusADM	C, Fortran	General Relativity
437.leslie3d	Fortran	Computational Fluid Dynamics
444.namd	C++	Biology: Molecular Dynamics
447.dealII	C++	Finite Element Analysis
450.soplex	C++	Linear Programming, Optimization

453.povray	C++	Image Ray Tracing, Image Rendering
454.calculix	C, Fortran	Structural Mechanics Finite element code
459.GemsFDTD	Fortran	Computational Electromagnetic
465.tonto	Fortran	Quantum Chemistry: Hartree-Fock
470.lbm	C	Fluid Dynamics: Lattice Boltzmann
481.wrf	C	Weather Modelling
482.sphinx3	C	Speech Recognition

The benchmarks in the Floating Point Suite, the source language and a brief description

### 6. Speed, Rate and Multiple Speed

The SPEC CPU suite is designed to measure the performances of a single CPU and the performances of a whole multi-CPU computer.

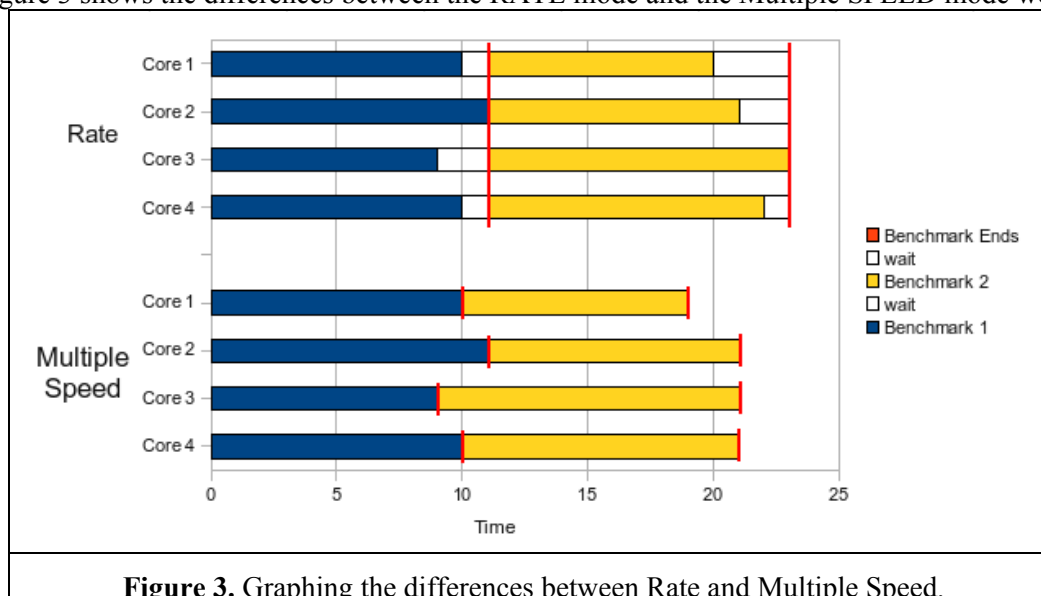
The SPEED (default) test runs each sub benchmark three times, takes the median result and finally computes the geometric average of all the benchmark scores.

The RATE test runs each benchmark in parallel on the desired number of CPUs. When all CPUs complete their workload for a benchmark the time is taken. If there are several CPUs in a worker node, and today it is common to have many cores in a system, when each core finishes its task it stays idle waiting for the slowest one.

So when one measures the RATE of N-core machines one gets less than N times the score obtained with SPEED, not only because of the completion time of the N processes on the single machine, but also because at the end of each run some cores are idle waiting for the last one.

In the HEP environment each job has a small bunch of independent events to process. In a typical batch system a job runs to completion on a CPU or core, and as soon as it finishes a new job is launched. This model is matched by the Multiple SPEED method of benchmarking, so we chose to run N instances of SPEED in parallel, instead of the RATE mode, to compute the throughput performance of a worker node.

Figure 3 shows the differences between the RATE mode and the Multiple SPEED mode we chose.



**Figure 3.** Graphing the differences between Rate and Multiple Speed.

The Multiple SPEED run finishes earlier so it gives a score slightly higher than the RATE test. The effect is more evident when running in a worker node with several cores.

## 7. Methodology

Our goal was to measure the performance of CPU-intensive HEP applications on a wide range of x86 worker nodes and find the best match with a standard suite of benchmarks.

The requirements for the suite were

- Easy to procure, not too expensive
- Maintained and supported by a wide community or a stable organization
- Widely recognized in the academic and industrial world

We began by studying and understanding the SPEC CPU2006 to assess its suitability as a benchmark, and we held weekly phone meetings with people from the four HEP experiments: Atlas, CMS, Alice and LHCb.

With the aid of the experiments, we selected a stable set of applications to run. The same fixed version of each application was to be run with identical event data on each test machine.

We also needed a dedicated set of worker nodes to run the benchmark and the applications.

## 8. The lxbench cluster

CERN provided a set of machines to run tests on, ranging from the old single core Intel Nocona to the modern quad core Intel Harpertown. We also had worker nodes with dual core AMD processors.

Other machines were made available for short periods from INFN, GridKa and RAL as a crosscheck but due to their limited availability we were unable to run a complete set of tests on them. These results are not reported.

Host name	RAM Size	CPU Speed (GHz)	Processors x Cores	CPU Architecture / Cache size
lxbench01	2x1 GB	2.8	2x1	Intel Nocona / 1MB
lxbench02	4x1 GB	2.8	2x1	Intel Nocona / 2MB
lxbench03	4x1 GB	2.2	2x1	AMD Opteron 275 / 2MB
lxbench04	8x1 GB	2.66	2x2	Intel Woodcrest 5150/ 4MB
lxbench05	8x1 GB	3.00	2x2	Intel Woodcrest 5160/ 4MB
lxbench06	8x1 GB	2.66	2x2	AMD Opteron 2218 / 2MB
lxbench07	8x2 GB	2.33	2x4	Intel Clovertown E5345 / 4MB
lxbench08	8x2 GB	2.33	2x4	Intel Harpertwon E5410 / 6MB

Table describing the hardware characteristics of the lxbench farm

## 9. The HEP applications

Alice provided three types of event processing for two types of events: pp Minimum Bias, and Pb-Pb.

- Generation plus Simulation
- Digitization
- Reconstruction

Atlas provided four kinds of event processing:

- Event Generation
- Simulation
- Digitization
- Reconstruction

CMS provided three types of event processing with CMS SW 2.0.0pre5:

- Generation plus Simulation
- Digitization
- Reconstruction

for several physics channels to test if some of them would give different results:

Higgs in four leptons, Minimum Bias, QCD jets, T-Tbar, and single particle (electron pion and muon) gun events

LHCb provided Generation + Simulation on pp Minimum Bias events.

### 10. The first results

Since all the experiments were running 32 bit applications, some of them on 64 bit operating systems, we chose to benchmark using this environment and made comparisons with SPEC CINT2006 and CFP2006 compiled in 32 bit mode.

Our goal was to find a benchmark that could predict the performance within a few percentage points. We found very good correlation of all HEP applications with both the Integer and the Floating Point benchmarks of the SPEC CPU2006 suite.

### 11. Perfmon

Helge Meinhard drew our attention to the work of Andreas Hirstius at the CERN Openlab on performance monitoring using the Perfmon tool. The Perfmon monitoring tools permits to measure with great details some internal counters on x86 cpu.

We analyzed the counters of a small subset of the lxbatch cluster at CERN where many different types of HEP application are usually running.

We noticed that, while the INT suite was in good correlation with our HEP applications, it does not match the percentage of Floating Point operations present in the typical HEP applications load.

We tried to see if some mixing of Integer and Floating Point benchmark could give a better match. Thus, if a future processor has improved integer or floating point performance but not both at the same time, we will track those improvements.

We found that a particular 'bset' (a set of benchmarks predefined by SPEC) had a very good agreement with the statistics obtained by the lxbatch cluster. This bset is called all\_cpp and consists of seven benchmarks, three from the CINT suite and four from the CFP suite. These are all the benchmarks written in C++ (471.omnetpp, 473.astar, 483.xalancmmk, 444.namd, 447.dealII, 450.soplex and 453.povray). The following figure shows that SPEC CINT2000 and CINT2006 have a negligible percentage of floating point instructions, SPEC CFP2000 and CFP2006 have around 30% and the all\_cpp bset of SPEC CPU2006 has around 10% of floating point instructions, as observed on lxbatch.

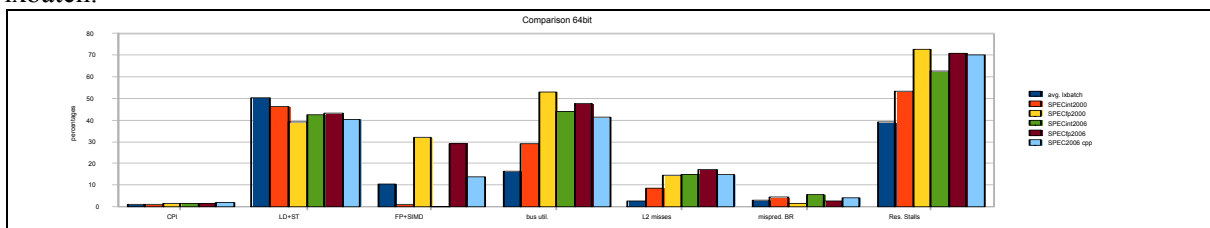


Figure 4. Perfmon Comparison on lxbatch and some SPEC benchmarks

The following table gives the measurements of SPECint2000, SPECint2006 and SPECfp2006 and the new all\_cpp for the systems in the lxbench cluster.

Machine	SPEC2000	SPECint2006 32	SPECfp2006 32	SPEC CPU2006	
				all_cpp	32
lxbench01	1501	11.06	9.5		10.24
lxbench02	1495	10.09	7.7		9.63
lxbench03	4133	28.76	25.23		28.03
lxbench04	5675	36.77	27.85		35.28
lxbench05	6181	39.39	29.72		38.21
lxbench06	4569	31.44	27.82		31.67
lxbench07	9462	60.89	43.47		57.52
lxbench08	10556	64.78	46.48		60.76

We cross-checked our suite of HEP applications with this new benchmark and found very good correlation as expected.

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

## 12. Conclusions

The working group identified a subset of the SPEC CPU2006 as the ideal reference benchmark for HEP. The subset consist of the cpp\_all bset, which demonstrated a good correlation with all the major HEP experiments' applications and a good balancing of INTEGER and FLOATING POINT instructions, similar to the patterns observed on the lxbatch cluster at CERN.

We propose a new name for the benchmark: HEP-SPEC06[7]. This acknowledges the fact that we use a benchmark derived from the SPEC CPU2006 benchmark suite, but with a clearly defined way of running it (tuned for HEP: on Linux, with gcc compiler, with the optimization switches we defined and in Multiple speed mode), and underlines the difference between it and the other benchmarks of the SPEC family.

- [1] SPEC <http://www.spec.org>
- [2] SPEC CPU 2006 <http://www.spec.org/cpu2006/>
- [3] HEPiX <https://www.hepix.org>
- [4] GCC <http://www.gnu.org/software/gcc/releases.html>

- [5] Scientific Linux <https://www.scientificlinux.org>
- [6] HEPiX processors web page <https://hepix.caspur.it/processors>
- [7] HEP-SPEC06 web page <https://hepix.caspur.it/benchmarks>
- [8] Wiki: <https://twiki.cern.ch/twiki/bin/view/FIOgroup/TsiBenchHEPSPECWlcg>
- [9] WLCG: <http://lcg.web.cern.ch/LCG/>