



HS06 – The history

Michele Michelotto

Before Y2K

- ▶ HEP application were running on IBM mainframe, VAX and later on Unix workstation
- ▶ Unit to measure the CPU performances
 - ▶ CERN Unit (CU), the power of an IBM 168 mainframe (circa 1980), proposed by Eric McIntosh
 - ▶ VAX 11/780 was described a one-MIPS machine because it was equivalent to IBM System/360 that ran one Mega IPS
 - ▶ Sometimes called VUP (Vax Unit of Performance)
 - ▶ A CU was equivalent to 4 Mips

SPEC

- ▶ SPEC (Standard Performance Evaluation Corporation)
- ▶ The score express how many time the machine is faster than a reference machine
- ▶ SPEC CPU 89
- ▶ SPEC CPU 92
 - ▶ CINT92 and CFP92. No electronic records available
- ▶ SPEC CPU 95
 - ▶ CINT95 and CFP95 – SPECint_rate95 and SPECfp_rate95 also available
- ▶ SPEC CPU 2000
 - ▶ This version was widely used for the LHC experiments Computing TDR and at during the first year of the GRID.

Problems with SI2K

- ▶ You have a computer node and you want to know the throughput in SI2K. How can I know the get it
- ▶ I measure it myself. I need to buy the SPEC suite, know how to run it, measure for each different processor
- ▶ I look up on spec.org site.

Lookup on SPEC site

- ▶ I look on Spec.org the rating of a machine with the same processor.
- ▶ I normalize the score to the clock of my node
- ▶ I take that score and multiply for the number of processor in my node
- ▶ I take that value and add on all the node of the same type
- ▶ Problem: Should I take the base or the optimized value? Does it scale linearly? Should I consider the compiler used? The compiler switches?
- ▶ I still have the problems of linearity
- ▶ Some architectures had a very high score on SPEC that was not reproduces by our application

I measure

- ▶ I buy the SPEC suite. I learn to make it run on my software environment. I measure the SI2K int on a process and I multiply by the number of CPU in the machine.
- ▶ More correct since I use the compiler I'll use for my application but each site has its own set of version of compilers and compilations switches (CERN vs GridKa)
- ▶ I still have the problems of linearity
- ▶ I can't use the rate because it is designed for job syncing at the end, not for batch jobs
- ▶ Some architectures had a very high score on SPEC that was not reproduced by our application. But the effect with gcc is less evidents

Using badly the SI2K

- ▶ This confusion in the definition of SI2K was leading to a confusion in the definition of the CPU power of a center.
- ▶ One can have two farm of 10 nodes each with 2 processor and 2 core each with SI2K scores different by a factor 2
- ▶ If you overestimate your farm you can comply to the pledge but the your site can't process all the HEP events you wanted to process
- ▶ If you underestimate your farm you're buying more CPU power than is necessary

Requirements for HS06

- ▶ Match around 3% of main HEP application (simulation, reconstruction)
- ▶ Footprint around 2GB/core like CPU application
- ▶ Reproduce the batch job environment. Single thread programs on fully loaded worker node
- ▶ Easy to run in order to have a number without any tuning
- ▶ Easy to run for IT companies that participate to tender
- ▶ Use the compiler switches from the architect forum

Bonus

- ▶ We made a fine tuning adding some Floating Point benchmarks taken from the Floating Point suite and added to the Integer Suite.
- ▶ Otherwise the SPEC CPU INT 2006 would have been ok
- ▶ Or a subset of it to keep it shorter

Ageing

- ▶ We imposed the `-m32` switch to force 32 bit code on 64 bit since in that moment we were in early phases of the migration to 64bit
- ▶ This is not a problem if the improvement of performances passing from 32 to 64 bit is the same on HS06 and HSI4