THE DEPENDENCE OF SITES SPEED FACTOR FROM THE DECLARED HS06

V. A. Matskovskaya [1], A. Sciabà [2]

[1] *Plekhanov Russian University of Economics, 36 Stremyanny per., Moscow, 117997, Russia*

[2] *CERN, CH-1211 Geneva 23 Switzerland*

E-mail: viktoriyams@mail.ru

The work started with the existing data analysis for the ATLAS experiment, designed to measure the processing speed of various ATLAS sites. The main task was to adapt the analysis so that it was fully automated and could be integrated into the ATLAS monitoring system. Another goal was to check whether the processing speeds declared by sites as “corepower” (i.e. the HEPSpec06 benchmark score divided by “corepower”) are well correlated with the rates established by this method and measure this correlation quantitatively. As a result, it was decided to continue the study, repeat the analysis and test individual sites for which there is a significant discrepancy from the reference values, measure and adjust the speed of various CPU models with their HS06 estimates.

Keywords: HepSPEC06, Grid sites, ATLAS, Kibana, speed factor, corepower.

1. Background

ATLAS developed a monitoring system based on Elasticsearch for storing data on all jobs it runs at different sites worldwide. Kibana, its graphical user interface, is available for data exploration and rapid prototyping of analyses and it allows to easily create complex queries.

In our analysis, data is aggregated in different "buckets" using these variables

- JEDI task ID (a task is a collection of similar jobs, each one running on a fraction of a dataset)
- Site
- CPU model
- Processing type (which describes the type of job)

Inside each bucket, numerical metrics (sums, averages, standard deviations) are calculated for the: CPU time, wallclock time, number of events, number of jobs, number of cores etc. Given a set of jobs running on a set of sites (and each one on a certain CPU), we can assume that if the jobs belong to the same JEDI task, their average CPU time per event is inversely proportional to the speed of the CPU. We call speed factor a dimensionless number proportional to the speed of the CPU. If our set of jobs includes several JEDI tasks, one can think of using simultaneously these tasks to fit the same set of speed factors.

2. Speed factor calculation

In the following, α indicates an index running over tasks and i an index running over sites or CPU times. The population average of the CPU time per event for task α at site or CPU i is $\mu_{i\alpha}$ and the average we measure is $a_{i\alpha}$. Of course, not all sites or CPU types appear on all tasks, so sums over i run only on sites or CPU types where the task runs.

It is assumed that the $a_{i\alpha}$ values we measure are Gaussian-distributed around $\mu_{i\alpha}$ with standard deviation $\sigma_{i\alpha}$. It is also assumed that the speed factors k_i do not depend on α, and in particular that $\mu_{i\alpha}k_i = A_\alpha$, where A_α is a constant with respect to i. It is assumed that also $\sigma_{i\alpha} = S_\alpha\mu_{i\alpha}, \forall i,\alpha$.

To make the analysis easier, we choose to use $\sigma_{i\alpha}$ as errors on $a_{i\alpha}$, while it would be more correct to use $\sigma_{i\alpha}/\sqrt{N_{i\alpha}}$, where $N_{i\alpha}$ is the number of jobs run at site or CPU type i for task α. We also assume that S_α=S, where S is constant for all tasks.

The χ^2 we want to minimise is the following:

$$\chi^2 = \sum_{i\alpha}\left(\frac{a_{i\alpha}-\mu_{i\alpha}}{\sigma_{i\alpha}}\right)^2 = \sum_{i\alpha}\left(\frac{a_{i\alpha}-\mu_{i\alpha}}{S_\alpha\mu_{i\alpha}}\right)^2 = \sum_{\propto}\frac{1}{{S_\alpha}^2}\sum_{i}\left(\frac{a_{i\alpha}}{\mu_{i\alpha}}-1\right)^2 =$$

$$= \sum_{\propto}\frac{1}{{S_\alpha}^2}\sum_{i}\left(\frac{a_{i\alpha}k_i}{A_\alpha}-1\right)^2 = \frac{1}{S^2}\sum_{\alpha}\frac{1}{A_\alpha^2}\sum_{i}(a_{i\alpha}k_i - A_\alpha)^2$$

As S is a constant, we can minimise a function $f = S^2\chi^2$ and the free parameters are A_α and k_i. We use as initial values $k_i = 1$ and $A_\alpha = (\sum_i a_{i\alpha})/n_\alpha$, where n_α is the number of sites or CPU types in task α.

The χ^2 does not change if we rescale all k_i and A_α by the same factor. Just for esthetic reasons we normalise the speed factors so that their sum is equal to their number (so to be varying around 1); therefore there is, no absolute scale and only ratios between speed factors are quantitavely meaningful.[1]

3. HepSPEC06

PanDA is the ATLAS workload management system, and in its architecture it has the concept of Panda queue (PQ) as an entry point to a computing resource. A PQ has an attribute called corepower, which is supposed to be the average HS06 score per core for the CPUs in the worker nodes accessed by that PQ. For each site, HEPSPEC06 was obtained from AGIS, the ATLAS database containing all information on the computing infrastructure. We know that these numbers have a low "trust level", as they are not accurately validated.[2]

For each CPU, the HS06 score is obtained from the HEPiX benchmarking working group. It's calculated in "ideal" conditions, e.g. HS06 run at boot time on the physical node. CPUs seen by real jobs can be different: virtual machine overhead, overcommitting, and most importantly, not knowing if the site enabled hyperthreading (which gives a 1.6 factor of difference in the "corepower"). It is fair to assume that in most cases HT is enabled.[3][4]

4. Stages of work

Firstly we do data aggregation from ES. Then we do the above speed factor calculations, then for each job we look at sites which have outstanding speed factor (very high or very low). Then we made plots for each site with abnormal CorePowers which differ from the most common values by more than 20**%**. We gave a closer look to sites that have a speed factor very far from the value that would correspond to its corepower according to the linear fit. The linear function was chosen with one parameter so that it passes through the point (0,0), because an site with corepower equal to zero can have only zero speed factor. For each processing type (job) we identify a few sites which have more than two errors to make linear fit and count correlation coefficient. So the correlation plots show how speed factors change with different CPUs.

For example you can see on Figure 1 that Intel Xeon E5-2650 v2 with Ivy Bridge architecture gets a very low speed factor compared to Haswell and Broadwell architectures. However, on Figure 2, you can see that the same CPU looks much faster than others. From this we can conclude that the difference may be due to factors different from the architecture.

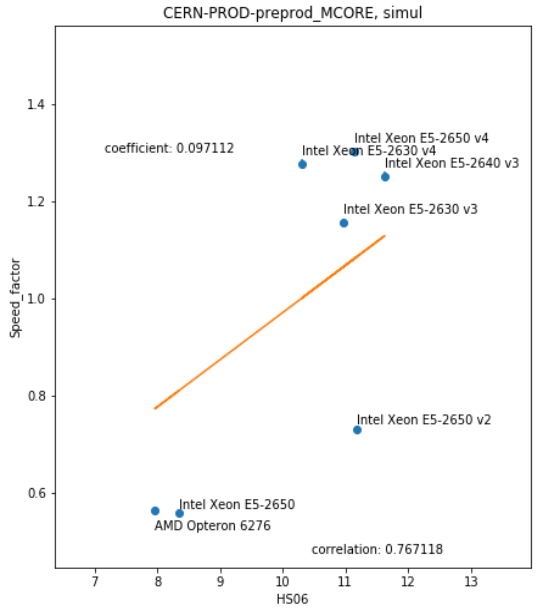

Figure 1. An example for job simul and site CERN-PROD-preprod_MCORE

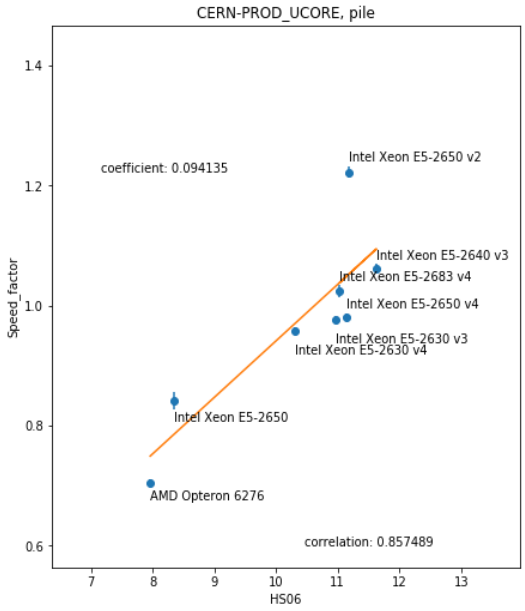

Figure 2. An example for job pile and site CERN-PROD_UCORE

5. Conclusion

The accuracy of the speed coefficients for sites compared to the main power is about 20%. this may occur due to many systematic uncertainties and inaccuracies of the main indicators. This is due to the variability of the core power of sites, for which the speed factor is checked, which affects the linear fit. However, with respect to speed factors for processors, the picture is less consistent.

Correlation graphs for different types of processing on the same sites look very similar to each other, which indicates that the analysis does not produce “random” numbers. The points of older CPUs tend to lie below points of newer CPUs on the correlation graphs. But this is not always the case, indicating the differences between the performance observed by the application and the “theoretical” performance of HS06 due to other factors.

This analysis can also be used for other Grid sites, but with adaptation to the site specifics.

References

[1] Sciabà A. et al. CPU benchmarking with production jobs //https://indico.cern.ch/event/578967/contributions/2455901/attachments/1408462/2153742/CPU_benchmarking_with_production_jobs.pdf

[2] Benchmarking WG // https://w3.hepix.org/benchmarking.html

[3] Sciabà A. et al. Passive benchmarking of ATLAS Tier-0 CPUs // https://indico.cern.ch/event/614359/contributions/2510792/attachments/1425728/2187149/Passive_benchmarking_of_ATLAS_Tier-0_CPUs.pdf

[4] A comparison of HEP code with SPEC1 benchmarks on multi-core worker nodes // http://iopscience.iop.org/article/10.1088/1742-6596/219/5/052009/meta