Benchmarking and accounting for the (private) cloud

2015 J. Phys.: Conf. Ser. 664 022035

(http://iopscience.iop.org/1742-6596/664/2/022035)

Journal of Physics: Conference Series **664** (2015) 022035 doi:10.1088/1742-6596/664/2/022035

Benchmarking and accounting for the (private) cloud

J Belleman and U Schwickerath

European Organization for Nuclear Research (CERN), 1211 Geneva 23, Switzerland

E-mail: Ulrich.Schwickerath@cern.ch

Abstract. During the past two years large parts of the CERN batch farm have been moved to virtual machines running on the CERN internal cloud. During this process a large fraction of the resources, which had previously been used as physical batch worker nodes, were converted into hypervisors. Due to the large spread of the per-core performance in the farm, caused by its heterogenous nature, it is necessary to have a good knowledge of the performance of the virtual machines. This information is used both for scheduling in the batch system and for accounting. While in the previous setup worker nodes were classified and benchmarked based on the purchase order number, for virtual batch worker nodes this is no longer possible; the information is now either hidden or hard to retrieve. Therefore we developed a new scheme to classify worker nodes according to their performance. The new scheme is flexible enough to be usable both for virtual and physical machines in the batch farm. With the new classification it is possible to have an estimation of the performance of worker nodes also in a very dynamic farm with worker nodes coming and going at a high rate, without the need to benchmark each new node again. An extension to public cloud resources is possible if all conditions under which the benchmark numbers have been obtained are fulfilled.

1 Introduction

Encouraged by the experiences from a small prototype for a private cloud running both direct experiment payloads and parts of the CERN batch farm at CERN [1], the CERN batch farm has been virtualised to more then 90% during the past two years. Virtualised batch resources are run on specific tenants on the CERN OpenStack [2] driven Infrastructure as a Service (IaaS) infrastructure [3]. Batch service managers are by construction ordinary users of this service and have no access to the hypervisors themselves which host the resources. Performance evaluations are done using the SPEC CPU2006 [4] with specific tunings for high energy physics applications, the so called HS06 benchmark [5]. HS06 benchmarking results are needed both for scheduling and accounting in the batch service. Benchmarking individual worker nodes when they are created is not an option because such virtual machines in general don't fill up a full hypervisor which can yield to over-optimistic benchmarking results. To work around this issue we have introduced a new scheme to classify worker nodes which allows to benchmark only a subset of nodes and generalise the results to all hosts of the same category. The resulting performance is then used both for scheduling and accounting via APEL.

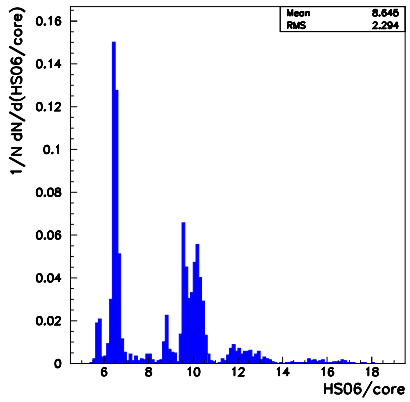

Figure 1. Normalised probability distribution of the per-core HS06 performance for virtual batch worker nodes at CERN.

A caveat of this classification is that it requires detailed information about the worker node which is easy to retrieve from the worker node itself. While this is fine for a traditional batch farm it would be nice to be able to use the same classification for nodes which belong to other users of the IaaS infrastructure for accounting purpose. Options to allow for this are discussed in the second part of this paper.

2 Virtualisation of batch resources

2.1 The CERN batch farm

The batch farm at CERN is a dynamic and heterogenous installation which changes effectively every day. At the time of the conference about 4300 machines were known to the batch farm, 3700 of them in public resources and available for WLCG processing. The remaining resources are dedicated for special applications, like HPC-like engineering applications or T0 processing. 93% of the batch nodes were virtual machines. The batch farm is currently managed by LSF, see for example [6] and [7].

CPU resources are bought in chunks of equal hardware and kept for at least three years. Each machine is benchmarked as part of the burn-in phase before it goes into production, and the results are stored in a hardware inventory database, along with features and purchase order IDs. Most of these resources are nowadays used as hypervisors in the CERN OpenStack-driven IaaS infrastructure. The batch service is only one of the customers of these resources. Users of the IaaS infrastructure have little or no influence on which hypervisor their new batch nodes will be dispatched, nor do they have access to the hypervisor or their performance numbers. Therefore, the performance is not known a priori and must be estimated when the host is up. This is very different from the old times when the batch farm consisted of physical nodes only. Those were classified and grouped by purchase ID and the performance was estimated per each such group using the HS06 benchmark.

2.2 Classification of worker nodes

Grouping worker nodes by their performance is still a requirement in a virtualised environment. The performance numbers are not only used for scheduling decisions by the batch system but also for accounting purpose. Simply benchmarking each VM once it has been created is not a solution either: unless the hypervisor is fully loaded the benchmarking results may end up being

Table 1. Worker node features which have an impact on the benchmark results.

feature	encoding	remarks
CPU brand	*i* for Intel *a* for AMD *o* anything else, unknown	encoded as single character
operating system	*6* for SL(C)6	major release version of OS
number of cores	integer number	logical cores
CPU vendor, type and generation	CPU type	from */proc/cpuinfo*
CPU clock speed	integer, in MHz	
memory speed	two integers	eg. 26 for 2.6GHz

too optimistic causing jobs to fail because LSF may think they had already used up their CPU allocation.

The first step is to define a way to classify worker nodes after they've been started. The performance evaluation is done with HS06[1]. Table 1 summarises the worker node features which are currently taken into account, along with the way they are encoded in the classification: For the operating system we concentrate on Scientific Linux brands only. Hypervisor BIOS settings enabling things like *virtualisation support* [8] or *Turbo mode* [9] are not accessible by virtual machines and are assumed to be tuned to maximise the performance of the machine. The CPU type is endcoded following a recipe given in [10]. CPUfamily, CPUmodel and CPUstepping values are retrieved from */proc/cpuinfo*, converted into hexadecimal and concatenated.

Putting it all together identifies the hardware reasonably well.

A caveat is that current version of KVM does not properly pass the memory speed to the virtual machines. Therefore, a conservative default of 266MHz is assumed if the memory speed information cannot be read from *dmidecode*. To make the described classification useful it is important that CPU passthrough is enabled on the hypervisors because CPU emulation has the side effect that different hardware types with potentially different performace will be merged in to the same class.

2.2.1 Example

Let's take a simple example. The string *a6_8_1512h23_266* means:

- AMD-based host
- SL(C)6 based operating system
- 8 cores
- CPU-ID 1512h
  - CPU family = *15h*
  - CPU model = *1h*
  - CPU stepping = *2h*
- (maximum) CPU speed is 2300 Hz

[1] Run in 32bit mode for historical reasons

- the memory speed could not be read so a default of 266 MHz is used [2]

The classification works in the same way for physical and virtual resources. There is no difference.

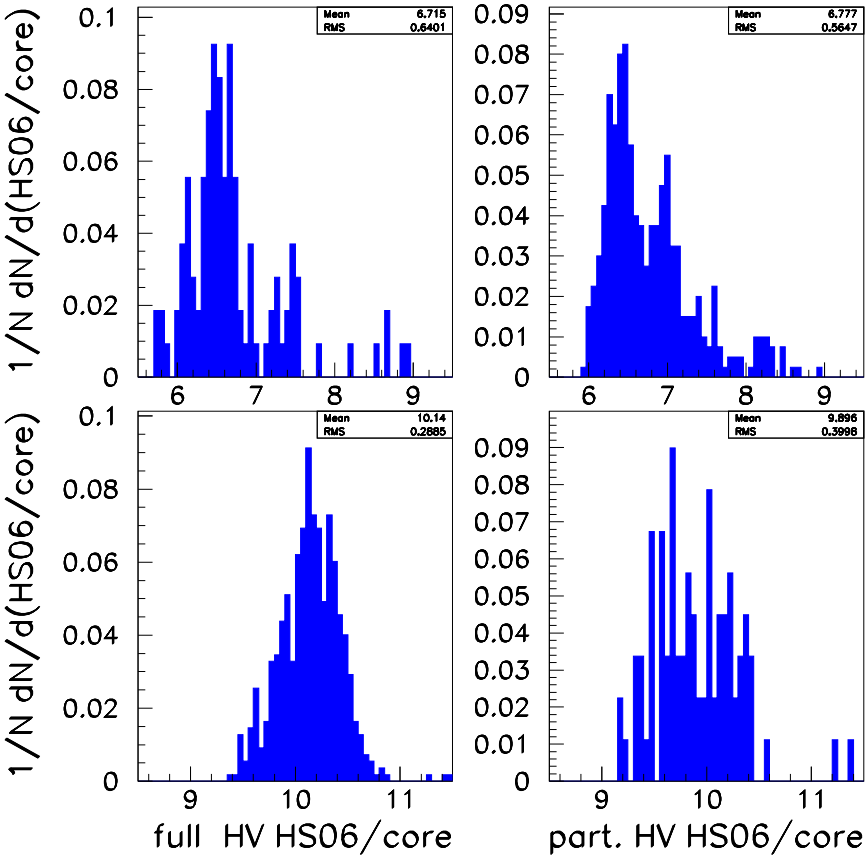

Figure 2. Benchmarking results for two different hardware types, namely a6_8_1512h23_266 (upper row) and i6_8_63e4h26_266 (lower row). On the left it was ensured that the hypervisors were fully loaded with VMs all doing the benchmark at the same time, on the right some VMs on the hypervisors may have been busy with something else.

3 Benchmarking

The benchmarking is done following the procedures documented in [11]. Results are returned via e-mail and parsed by analysing the subject lines of the received e-mails. A small modification was applied to ensure that the subject of the e-mails contained both the host name and the classification of the host name, along with the final benchmarking result. During the measurements it has been ensured that the underlying hypervisors were fully loaded with other batch nodes being benchmarked at the same time. Fig. 1 shows the per-core performance in HS06 for the virtual batch worker nodes in the CERN batch farm. The values vary by almost

[2] This is in fact a virtual batch worker node

a factor three so a proper schema for classifying different worker nodes is needed. The different peaks correspond to different hardware types which result in different performances. The values vary from 5 to up to 18. Thus a good classification of the nodes is needed to ensure proper normalization of CPU times for scheduling of batch jobs as well as for accounting purposes.

As already mentioned, when doing the benchmarking it is essential to ensure that the hypervisors get fully filled up. This is the only way to ensure that the results are representative for the whole class being benchmarked. Fig. 2 shows what happens if this requirement is not fulfilled. While for the VMs on the left hand side precautions were taken to ensure that the underlying hypervisor was running batch VMs which were all being benchmarked at the same time, for the plots on the right this requirement was not necessarily fulfilled and some VMs may have been busy with unpredictable I/O or CPU intensive payloads, or even no work load at all. While for shown two cases the mean value does not shift significantly within the available statistics, the distributions on the right hand side have a larger width, significantly deviate from a gaussian shape and additional peaks at higher values become visible.

Also, it must be ensured that the hypervisors are set up in their final configuration as worker nodes, including all the necessary daemons up and running. Only this way it can be assured that the benchmarking result is representative for the final user experience. It is assumed that all the hypervisors are configued in the same way, in the sense that all use the same settings and optimizations. If hypervisors have different tunings and settings this would be visible in the distribution of the HS06 values measured on the samples (provided the sample of virtual machines is representative). When interpreting the results one needs to be pessimistic because optmistic results are dangerous for jobs. If a job ends up on a worker node which has a performance which is lower than what was measured for the class it belongs to, there is a risk that the batch system will kill the job prematurely with a CPU time limit.

When starting the benchmark itself it is important to synchronise the startup of the benchmarks on the different independent virtual machines. This problem was addressed by starting the benchmark into background using parallel ssh on all candidate hosts simultanously. There is still a small error introduced by this approach due to synchronisation issues which can result in individual benchmarking results to look too optimistic. Since the benchmark runs for several hours though the impact of this is small and can be further reduced by running the benchmark in a row several times on each worker node.

There are two cases which will now be described.

3.1 Deployment of new resources

Whenever new resources are made available, eg to replace old resources, it is ensured that full blocks of new hypervisors are deployed and given to the batch service managers. The new resources are fully filled up and the HS06 benchmark is started on all new worker nodes at the same time. The results are then interpreted on a statistical basis, taking the mean value of the lowest peak (if there are several in the distribution).

3.2 Re-benchmarking of already deployed resources

Under certain circumstances it can become necessary to re-benchmark existing resources. This can become necessary in case of a major change on the hypervisors, for example additional performance enhancements. As in the case mentioned above it is necessary to identify and benchmark all VMs which run on the same hypervisors and start the benchmark at the same time. This use case requires knowledge about the link between the virtual machine and the

hypervisor, as well as the resources the hypervisor provides. Once VMs have been grouped by hypervisor, the benchmarking can be done as described above.

4 Accounting

CPU requirements and pledges for the experiments are expressed in terms of HS06 ratings. In order to be abled to check if all experiments got their pledges at the end of the month it is important to have reliable performance ratings of the worker nodes they have used. This is true both for traditional batch processing on a shared batch farm and for direct access to IaaS resources by the experiments. Doing the accounting from the perspective of the resource provider in a virtualised environment can be a challenge. Two different cases are discussed here: a (virtualised) batch farm where the worker nodes are owned by the same team doing the accounting and the IaaS case where the owners of the worker nodes differ from the team doing the accounting (cloud accounting case).

4.1 Batch Accounting

Accounting in the CERN batch farm is done via APEL. The described classification schema allows for a standard way of dealing with this and no special measures are to be taken. Fig. 3 shows how it works. Thanks to the classification schema described above and the fact that the worker nodes are centrally managed via Puppet the performance of each worker node is known to the batch system as well as to the worker node itself. In particular, there is no need to make a difference between physical and virtual worker nodes.

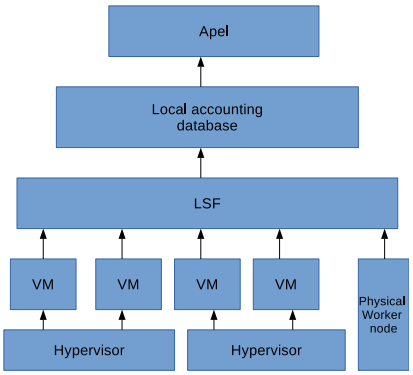

Figure 3. APEL based accounting the CERN batch farm which mixes virtual and physical resources.

Since the classification of worker nodes only requires information which is available on the worker node it is much more portable than the previous convention which was based on the purchase of the hardware. The benchmarking results should be portable to other clouds under the condition that the settings on the hypervisors are the same as on the systems where the benchmarking was done.

4.2 Cloud accounting

As described previously the virtual batch worker nodes at CERN run on the same infrastructure as experiment-owned resources serving the same purpose, namely number crunching. By construction the resource provider has no influence on what is run on these tenants and does not have access to these experiment-owned resources. Therefore, the classification of these virtual

machines as described above cannot be easily retrieved. Nevertheless these resources are part of the pledges and need to be accounted for. Instead of going for the accurate solution found for the batch farm, the performance of the underlying hypervisor is used instead to estimate the performance of the guests running on it.

Cloud accounting at CERN is based on OpenStack Ceilometer. Fig. 4 shows how this is currently implemented. The host name is retrieved from Ceilometer. From the site networking database

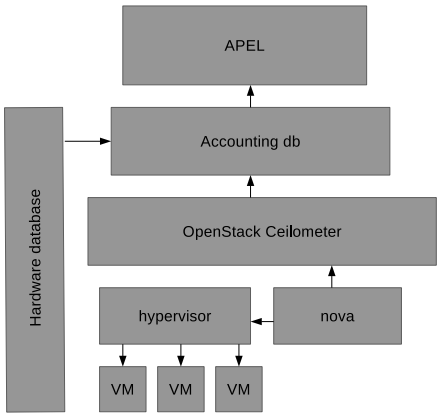

Figure 4. General case: current CERN implementation.

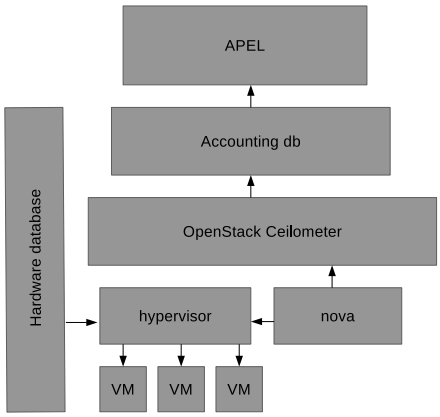

Figure 5. General case: possible portable solution.

the name of the hypervisor can be found and an additional lookup into the hardware database is needed to get the performance of the hypervisor. This solution has two main caveats:

- the implemenation is site-specific and not portable because of the use of additional site specific databases within the stream
- for short-lived VMs it often happens that at the time the performance lookup has to be done the VM has already been destroyed and disappeared from the network database. In this case the link to the hypervisor is broken and no performance value can be taken[3]

A possible improvement is shown in Fig. 5. In this scenario the hypervisor knows its performance. The number can be provided for example with Puppet [12] and is stored somewhere on the node or exported as a fact. A Ceilometer pipe is then used to add this information into Ceilometer which can then be queried when the accounting process jump in. This approach is much more portable than the current way of doing things because the only thing the service provider needs to do is to make the rating of the hypervisor available to the hypervisor itself. The rest of the code is generic and can be ported to other sites. This new approach is currently under development at CERN.

5 Outlook

Depending on the future experiences with Ceilometer pipes further improvements may be possible. If the operating system run by the guests can be guessed from the image name it should be possible to guess the hardware type of the guests in the way described above. The

[3] As a default value in that case the mean value shown in Fig. 1 is used.

hypervisor could then feed either the class or directly the corresponding HS06 rating per VM into Ceilometer.

6 Conclusions

We've developed a new classification schema which can be applied to virtual batch nodes running on the CERN internal cloud infrastructure. Benchmarking is done for a sub-class of worker nodes ensuring that the underlying hypervisors are fully loaded during the benchmark run. This classification works fine and is site independent. It cannot be direclty used though for doing cloud accounting on resources to which the site has no direct access. Therefore, for cloud accounting the performance of the hypervisor is used. This schema is site specific and has issues with short lived VMs. A better solution based on Ceilometer pipes is under development. Ceilometer pipes may offer more possibilities for a more accurate cloud accounting in the future, using the schema in use for CERNs batch farm.

References

[1] Goasguen S, Moreira B, Roche E and Schwickerath U, *J.Phys.Conf.Ser.* **396** (2012) 032098
[2] Open source software for creating private and public clouds, `http://www.openstack.org`
[3] Andrade P, Bell T, van Eldik J, McCance G, Panzer-Steindel B, Coelho dos Santos M, Traylen S and Schwickerath U, *J.Phys.Conf.Ser.* **396** (2012) 042002
[4] Henning J, *Computer Architecture News*, Volume **34**, No. 4 and references therein
[5] HEP-SPEC06 (HS06) Benchmark, `http://w3.hepix.org/benchmarks/doku.php`
[6] Lefebure V and Schwickerath U, *J.Phys.Conf.Ser.* **119** (2008) 042025
[7] IBM Platform LSF, `http://www-03.ibm.com/systems/platformcomputing/products/lsf`
[8] Intel® Corporation 2006, Intel® Virtualization Technology and Intel® Active Management Technology in Retail Infrastructure, Intel® white paper November **2006**
[9] Intel® Corporation 2008, Intel® Turbo Boost Technology in Intel® Core™ Microarchitecture (Nehalem) Based Processors, Intel® white paper November **2008**
[10] How to find out which (Intel) CPU you have, `http://world.std.com/~swmcd/steven/tech/cpu.html`
[11] http://w3.hepix.org/benchmarks/doku.php?id=bench:howto
[12] IT infrastructure management system Puppet, `https://puppetlabs.com`