Benchmarking Commercial Cloud Resources

D. Giordano
(CERN IT-SDC)

White Areas Lecture
16 December 2015
Acknowledgments

• Work done in collaboration with
  – SDC Cloud Team (Twiki)
  – Vladimir Petrov (Master Thesis student project)

• A similar talk was given at HEPiX Workshop, Fall 2015
  – This is a longer version
    • Including more details and plots
    • Recent findings in benchmarking commercial clouds
Performance Measurement

• “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.”

• “The process of performance comparison for two or more systems by measurements is called benchmarking, and the workloads used in the measurements are called benchmarks.”

– 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
Benchmarking Computers for HEP

Eric McIntosh
CERN, Geneva, Switzerland

Abstract

This report summarises the results of the CERN benchmark tests carried out on a variety of Mainframes and Workstations during the last fifteen years. The tests are a suite of FORTRAN programs used to determine the CPU power of a computer system for running High Energy Physics applications. They are essentially scalar due to the well known difficulties in vectorising this type of application, but a matrix inversion in convenience and continuity I decided to keep the CPU metric, but reduce the number of production codes. I was able to make GABI (now CRN5), JAN (CRN12), and FOWL (CRN3), reasonably easy to port and I added another modern event generator LUND (CRN4) to give a 50/50 distribution between event generation and reconstruction as that was the workload distribution at the time. I supplemented these codes with several “kernel” type applications to at least get a feel for compilation times (CRN4C), vectorisation
Performance in Cloud Environments

• Performance measurement and monitoring are essential

• Additional benefits in a commercial cloud environment (private – public – hybrid clouds)
  – Deal with the intrinsic variability and inhomogeneity
  – Compare the presumed and perceived performance
  – Quickly identify performance issues
In The Next Slides

• Experience gained in procuring commercial cloud resources and connection with benchmarking

• Description of the Benchmark Suite implemented to collect and analyse performance metrics

• Some study results

• NB: Focus on CPU performance metric
Caveat

• Other benchmarking activities not covered in this talk
  – HEP-SPEC 06 benchmarking
    • The official CPU performance metric used by WLCG sites since 2009
      – It meets the HEP requirements:
        » Percentage of floating point operations observed in batch jobs
    • Component of the procurement process for bare-metal servers
    • References: HEPiX Benchmark WG

  – Passive benchmarking
    • CPU power normalization based on multivariate fit
      – Use real workloads (experiments’ jobs) as not-calibrated metrics
    • Christian Nieke (IT-DSS-DT)
    • References: Analytics WG meeting
Cloud Procurement & Benchmarking
CERN Cloud Procurement

- Started in 2011 within the Helix Nebula partnership among leading research organizations and European commercial cloud providers
  - Objective is to support the CERN’s scientific computing programme
    - Integrate commercial cloud IaaS within the experiment frameworks
    - Improve the CERN procurement process for cloud IaaS
    - Evaluate cost and benefit of cloud IaaS
CERN Cloud Procurement Roadmap

- **First Procurement**, March ‘15
  - Target a single VO, run simulation jobs

- **Second Procurement**
  - Production activity currently running (started in November ’15)
  - Target multi VOs, simulation jobs

- **Third Procurement**
  - Production activity to start during Spring ‘16
  - Target multi VOs, full chain processing

- EC co-funded joint Pre-Commercial Procurement (PCP) HNSciCloud project (‘16-’18)
  - More details in R. Jones talk at Nov. GDB
Commoditize Cloud Resources

• Benchmarking needs during the procurement process
  – Define technical specs, adjudication criteria and remediation options
    • Request a VM able to process at least N reference events/sec
    • Verify that resources are delivered according to the specified performance
    • Compare offers based on cost/event
      – NB: Current CERN adjudication rules are based on cheapest compliant price and not best value for money

<table>
<thead>
<tr>
<th></th>
<th>cost ratio</th>
<th>cpu/evt</th>
<th>cpu/evt ratio</th>
<th>cost/evt (ratio)</th>
<th>ratio_(cost/h *s/evt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudA</td>
<td>1.00</td>
<td>0.94</td>
<td>1.00</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>CloudB</td>
<td>3.04</td>
<td>1.13</td>
<td>1.20</td>
<td></td>
<td>3.65</td>
</tr>
<tr>
<td>CloudC</td>
<td>2.04</td>
<td>3.36</td>
<td>3.57</td>
<td></td>
<td>7.31</td>
</tr>
</tbody>
</table>

• Enable a “cloud” commodity exchange based on a measurable value
  – See Deutsche Boerse Cloud Exchange approach
  • “Benchmarks are used to define a PU. The Compute Product and Memory Product are currently only available as Performance level regular”

(http://cloud.exchange/Product-Concept/DBCE-Product-Concept.pdf)
Benchmarking in the Procurement Phase

- Started ~1 year ago for the preparation of the first CERN cloud procurement (Production in March ’15)

- Evaluated different alternatives based on the following requirements
  - Open source
    - Share it easily with cloud providers and let them run it
      - Crucial in tender phase to allow proper choice of VM configuration
  - Light weight installation, reasonably fast running time
    - Submission of bmks on many VM instances for fine-grained probing approach
      - Possibly getting code from remote repository like cvmfs
  - Reproducible
    - If random generation is used, fix random seed in order to have always the same sequence of events
  - Functional relationship with experiment workloads
    - To extrapolate expectations on job duration
ATLAS Kit Validation tool

- Considered ATLAS tools
  - Being the ATLAS MC production the targeted flagship use case for the March ‘15 production
  - Build on past experience

- ATLAS Kit Validation (KV)
  - Well known tool used by the ATLAS community
  - Framework essentially independent from the underlying tests
  - It’s mainly bash and python based wrapper
    - ATLAS code accessed from CVMFS
  - Comparison with HEP-SPEC06 already studied in the past
KV Reference Workload

- Which workload to use for benchmarking?
  - CPU time/event is different for each workload
  - Measured that within ~10% the relative CPU/event performance doesn’t depend on specific workloads
    - Confirmed also using a different approach: HammerCloud jobs

- Preferred workload: **G4 single muon**: faster running time O(few mins)
  - NB: the CPU time/event doesn’t include the first event, to avoid bias due to the initialization process
Benchmarking During March ’15 Production

- Up to 3,000 concurrent running VMs
  - ~1.2 million CPU hours of processing

- Each provisioned VM has been benchmarked
  - ~30,000 VM benchmarks performed
  - KV benchmark: 100 Single Muon events simulated (~2 min to run)

- Results
  - CPU performance uniform within 15% spread
  - Benchmark profile consistent over time
  - Consistent job CPU performance and benchmark
    - Prompt identification of outliers

More details in GDB of June ’15

KV CPU time/evt: daily distributions

Job CPU time/evt Vs KV CPU time/evt

Outliers
Benchmark in Current Procurement (Oct. ‘15)

- Building on the March ‘15 experience, CERN has launched a second larger procurement for commercial cloud resources.

- In this case benchmarking is used to
  - Fix limits: min. desired (KV 1.2 s/evt) and tolerated performance (KV 1.5 s/evt).
    - Reminder: adjudication is on cheapest compliant bid.
  - Define service credits for poor performance.
    - All provisioned VMs are systematically profiled (shown in a dedicated slide).

![KV bmk results as performed by cloud providers](chart.png)

**Rejection threshold**
Supporting Diverse WMS Approaches

- 4 single-core or 1 multi-core job per VM (4 vCPU)
- Different VM lifecycles
- **Common benchmarking approach:**
  - Profile each VM at the beginning of its lifetime and at each pilot cycle
Cloud Benchmark Suite
Cloud Benchmark Suite

- Be able to run several benchmarks on the same cloud resource

- Strategy
  - Allow collection of a configurable number of benchmarks
    - Compare the benchmark outcome under similar conditions
  - Mimic the usage of cloud resources for experiment workloads
    - Benchmark VMs of the same size used by VOs (1 vCPU, 4 vCPUs, etc)
    - Probe randomly assigned slots in a cloud cluster
      - Not knowing what the neighbor is doing
  - Generalize the contextualization to run the benchmark suite in any cloud
  - Have a prompt feedback about executed benchmarks
    - In production can suggest deletion and re-provisioning of underperforming VMs
  - Ease data analysis and resource accounting
Benchmark Suite Architecture

- A configurable sequence of benchmarks to run
- Results are collected in Elasticsearch cluster & monitored with Kibana
  - **Metadata**: VM UID, CPU architecture, OS, Cloud name, IP address, …
- Detailed analysis performed with IPython analysis tools
Benchmarks Approach

- For each benchmark run in parallel as many threads as the number of vCPUs

- Two running modes
  - Sequential
  - Synchronized
6 Benchmarks Used (so far)

- HEP related
  - LHCb Fast Benchmark (**fastBmk**)
    - Original python code modified by A. Wiebalck to run `python.multiprocessing`
    - Very fast, gaussian random generator
  - ATLAS KV
    - **NB: other benchmark candidates can be included**

- Open-source Phoronix benchmarks adopted by DBCE to "commoditize" resources

<table>
<thead>
<tr>
<th>Category</th>
<th>Benchmark Tool (B.)</th>
<th>Threshold (t.)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application/CPU</td>
<td>7-Zip Compression</td>
<td>1600 MIPS</td>
<td>Number of MIPS of (virtual) CPU when compressing a file with 7zip</td>
</tr>
<tr>
<td>Application/CPU</td>
<td>LAME MP3 Audio Encoding</td>
<td>36 s</td>
<td>Time to convert a WAV into an MP3 file with LAME (CPU only)</td>
</tr>
<tr>
<td>Application/CPU</td>
<td>Linux Kernel Compiler3</td>
<td>1480 s</td>
<td>Time to compile a Linux kernel. (CPU only)</td>
</tr>
<tr>
<td>Application/CPU</td>
<td>x264 Video Encoding4</td>
<td>14.5 frames/s</td>
<td>Time to convert video from MPEG2 to MPEG4</td>
</tr>
</tbody>
</table>
Amount of Data Collected

- ~870k benchmark suites executed
- Various providers tested
  - Including CERN OpenStack
  - Different data centres (when possible)
    - Azure NorthEU, WestEU, CentralUS
- Different OS for the VM image
  - SLC6, CernVM, CentOS 6.*
- Different CPU models

<table>
<thead>
<tr>
<th># tests per Cloud laaS &amp; OS</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>AZURE</em></td>
</tr>
<tr>
<td>Scientific Linux release 6.6 (Carbon)</td>
</tr>
<tr>
<td><em>AZURE</em></td>
</tr>
<tr>
<td>CentOS release 6.7 (Final)</td>
</tr>
<tr>
<td>CERN-OPENSTACK*</td>
</tr>
<tr>
<td>Scientific Linux CERN SLC release 6.6 (Carbon)</td>
</tr>
<tr>
<td>CERN-OPENSTACK*</td>
</tr>
<tr>
<td>Scientific Linux release 6.6 (Carbon)</td>
</tr>
<tr>
<td>CERN-OPENSTACK*</td>
</tr>
<tr>
<td>Scientific Linux release 6.7 (Carbon)</td>
</tr>
<tr>
<td>CERN-OPENSTACK*</td>
</tr>
<tr>
<td>CentOS release 6.6 (Final)</td>
</tr>
<tr>
<td>ULTIMUM</td>
</tr>
<tr>
<td>CentOS release 6.6 (Final)</td>
</tr>
<tr>
<td>ULTIMUM</td>
</tr>
<tr>
<td>Scientific Linux release 6.6 (Carbon)</td>
</tr>
<tr>
<td>INNOVO</td>
</tr>
<tr>
<td>Scientific Linux release 6.6 (Carbon)</td>
</tr>
<tr>
<td>INNOVO</td>
</tr>
<tr>
<td>CentOS release 6.5 (Final)</td>
</tr>
<tr>
<td>CLOUDATA</td>
</tr>
<tr>
<td>Scientific Linux release 6.6 (Carbon)</td>
</tr>
<tr>
<td>CLOUDATA</td>
</tr>
<tr>
<td>CentOS release 6.6 (Final)</td>
</tr>
<tr>
<td>DARZ</td>
</tr>
<tr>
<td>Scientific Linux release 6.6 (Carbon)</td>
</tr>
<tr>
<td>DARZ</td>
</tr>
<tr>
<td>CentOS release 6.6 (Final)</td>
</tr>
<tr>
<td>CLOUDHEAT</td>
</tr>
<tr>
<td>Scientific Linux release 6.6 (Carbon)</td>
</tr>
<tr>
<td>CLOUDHEAT</td>
</tr>
<tr>
<td>CentOS release 6.6 (Final)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># tests per Cloud laaS &amp; CPU model</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>AZURE</em></td>
</tr>
<tr>
<td>Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz</td>
</tr>
<tr>
<td><em>AZURE</em></td>
</tr>
<tr>
<td>Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz</td>
</tr>
<tr>
<td><em>AZURE</em></td>
</tr>
<tr>
<td>AMD Opteron(tm) Processor 4171 HE</td>
</tr>
<tr>
<td><em>AZURE</em></td>
</tr>
<tr>
<td>Intel Core i7 9xx (Nehalem Class Core (7)</td>
</tr>
<tr>
<td>CERN-OPENSTACK*</td>
</tr>
<tr>
<td>Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz</td>
</tr>
<tr>
<td>CERN-OPENSTACK*</td>
</tr>
<tr>
<td>Intel Core i7 9xx (Nehalem Class Core (7)</td>
</tr>
<tr>
<td>CERN-OPENSTACK*</td>
</tr>
<tr>
<td>Intel Xeon E312xx (Sandy Bridge)</td>
</tr>
<tr>
<td>CERN-OPENSTACK*</td>
</tr>
<tr>
<td>AMD Opteron 62xx class CPU</td>
</tr>
<tr>
<td>CERN-OPENSTACK*</td>
</tr>
<tr>
<td>Westmere E56xx/L56xx/X56xx (Nehalem-C)</td>
</tr>
<tr>
<td>ULTIMUM</td>
</tr>
<tr>
<td>Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz</td>
</tr>
<tr>
<td>ULTIMUM</td>
</tr>
<tr>
<td>AMD Opteron 62xx class CPU</td>
</tr>
<tr>
<td>INNOVO</td>
</tr>
<tr>
<td>Intel Core i7 9xx (Nehalem Class Core (7)</td>
</tr>
<tr>
<td>INNOVO</td>
</tr>
<tr>
<td>Intel Xeon E312xx (Sandy Bridge)</td>
</tr>
<tr>
<td>CLOUDATA</td>
</tr>
<tr>
<td>Westmere E56xx/L56xx/X56xx (Nehalem-C)</td>
</tr>
<tr>
<td>DARZ</td>
</tr>
<tr>
<td>Intel Xeon E312xx (Sandy Bridge)</td>
</tr>
<tr>
<td>CLOUDHEAT</td>
</tr>
<tr>
<td>AMD Opteron 63xx class CPU</td>
</tr>
</tbody>
</table>

Number of tests per day
Benchmark Results at a Glance

- Metric ~ [1/s]; each point ↔ 10 min average; Colour ↔ Cloud
Benchmark Results at a Glance

- Metric ~ [s]; each point ⇔ 10 min average; Colour ⇔ Cloud
Case of Study: OpenStack at CERN

Work done in collaboration with J. Van Eldik

• Evaluate the effect of hypervisor load on the performance of single vCPU VMs
  – Extracted 5 nodes from pool of computing nodes
    • Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz
  – Load phases: create a targeted number of VMs per hypervisor
    • 1 VM per HyperV
    • 16 VMs per HyperV
    • 30 VMs per HyperV
  – VM image: Scientific Linux CERN SLC release 6.6 (Carbon)
  – Run sequence of benchmarks

• Used Phoronix open source benchmarks to produce load
Qualitative Look at Data

- Identifiable **transition** of CPU performance when load changes
  - Seen in all benchmark measurements. Performance **recovers** scaling down
Qualitative Look at Data

- Larger dispersion in KV and FastBmk values in the highest-load region
More Quantitative Analysis: FastBmk Vs KV

- Correlation study in the region 16 and 30 VMs
  - NB: *FastBmk metric transformed into value\(^{-1}\) [s]
  - The average performance degradation differs per Hypervisor and Bmk used

Ratio mean(30VMs)/mean(16 VMs)

Evolution of a single VM in the parameter space
FastBmk Vs KV

Profile-X

Projection-Y
Aggr. x hypervisor

2D plot

Projection-X
Aggr. x hypervisor

A single Hyperv.!!
And the Other Benchmarks?

- Ability to discriminate different hypervisor performance depends on the specific test.
Probing the OpenStack Compute Environment

- Idea: probe performance of VMs in OpenStack Compute Environment
  - Where resources are assigned to the experiments for CERN cloud activities
  - Tenant with ~200 single-core VMs
    - Make sure VMs are provisioned in different Hypervisors
    - Run synchronized benchmarking suite

Num VMs per pnode
Preliminary Profiling Results

**Average KV performance Vs Time**

- Mean (green line)
- 5th percentile (blue line)
- 95th percentile (red line)

**KV performance per pnode**

- 5th percentile (green line)
- 95th percentile (blue line)
Microsoft Azure Evaluation
Objective

• Joint collaboration between CERN Openlab and Microsoft Azure
  – Evaluation of Azure platform and integration with existing WLCG tools

• Achievements
  – Evaluated two different provisioning models
    • Azure Service Manager (old), Azure Resource Manager (new)
  – Adopted CernVM image
  – Azure resources integrated in WMS of ATLAS, CMS, LHCb
  – Scale tests
    • Performed in 3 DCs (2 EU, 1 US)
    • Reached ~4800 vCPUs provisioned
  – Performance evaluation
    • Benchmarks and cost
KV Vs fastBmk Performance

- Good Linearity among two independent benchmarks: KV and fastBmk
- The measured effect does not depend on specific compiler flags

A3 and D3 series

A1 series
To Summarise

- In cloud environment the VM performance is highly variable
  - Changes with load on the IaaS
  - It is differently measured by different benchmark tests

- Synthetic, fast benchmarks running in each VM, iteratively along the VM lifetime, allow to spot performance changes
  - Can consolidate accounting of resources, supplying normalization factors
  - Choice of a valid benchmark to translate in job expectation is not trivial
    - Identify reliable benchmark for multi-core applications
What Next?

• Discussion about synthetic benchmarks is ongoing in WLCG and within experiments
  – Several good candidates available
  – Interesting GDB talks on HEP-SPEC06 scalability

• Correlation studies with job performance are ongoing
  – Still preliminary to be discussed here

• In addition to the measurement of CPU performance
  – Profile network and storage access
  – Network is a crucial component of a distributed system
    • Measure LAN and WAN performance
    • Monitoring experience already established in WLCG
      – Could be exported to cloud environments
Conclusions

• Benchmarking is an important aspect of the production process
  – Component of the procurement procedures
  – Monitor delivery on specifications
  – Support activities on performance improvement

• Cloud Benchmark Suite available
  – Configurable. Can include additional benchmarks
  – Tested in several IaaS. Data available through ES-Kibana