Benchmarking in Cloud Environments

D. Giordano, C. Cordeiro
CERN

pre-GDB Benchmarking
7 February 2017
Experience

Started in 2015 within the Helix Nebula project
Running in a number of cloud providers
Collected ~4 M tests
  – Focus on CPU performance
Adopted for
  – CERN Commercial Cloud activities
  – Azure / IBM evaluations
  – CERN OpenStack tests
"Commoditize" Cloud Resources

Benchmarking needs during the procurement process

– Define technical specs, adjudication criteria and remediation options
– Provide Cloud Providers with easy to use recipes to run benchmarks specific to our field, resulting in a clear measurement

<table>
<thead>
<tr>
<th></th>
<th>cost ratio</th>
<th>cpu/evt [s]</th>
<th>cpu/evt ratio</th>
<th>cost/evt (ratio) 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>1.00</td>
</tr>
<tr>
<td>CloudB</td>
<td>3.04</td>
<td>1.13</td>
<td>1.20</td>
<td>3.65</td>
</tr>
<tr>
<td>CloudC</td>
<td>2.04</td>
<td>3.36</td>
<td>3.57</td>
<td>7.31</td>
</tr>
</tbody>
</table>

Benchmarking needs during the operation

– Measure the effective performance
  • And compare it to the declared performance
– Quickly identify performance issues
Benchmarking & Service Credit Compensation

Poor performance gives rise to Service Credit Compensation

- Fix limits: min. desired and tolerated performance Compensation $\propto$ lost performance · penalty

![Poorly performing cloud](chart1)

![Well performing cloud](chart2)
Studied Benchmarks

Fast (much faster than HS06) and representative of the HEP workloads

- Convergence on KV, DB12 (Dirac Benchmark 12)
- Whetstone also included (mainly for legacy reasons)

Also evaluated other benchmarks: Phoronix CPU benchmark

- Dropped because not showing linearity respect to the HEP workloads

<table>
<thead>
<tr>
<th>Benchmark Tool (Bₙ)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-Zip Compression</td>
<td>Number of MIPS of (virtual) CPU when compressing a file with 7zip</td>
</tr>
<tr>
<td>LAME MP3 Audio Encoding</td>
<td>Time to convert a WAV into an MP3 file with LAME (CPU only)</td>
</tr>
<tr>
<td>Linux Kernel Compiler3</td>
<td>Time to compile a Linux kernel. (CPU only)</td>
</tr>
<tr>
<td>x264 Video Encoding4</td>
<td>Time to convert video from MPEG2 to MPEG4</td>
</tr>
</tbody>
</table>
Correlation KV vs (some) HEP workloads

**KV**

- Seen that within ~10% the relative CPU/event performance doesn’t depend on specific CPU intensive workloads
  - Even if CPU time/event is different for each workload
- Preferred workload: **G4 single muon**: faster running time \( O(\text{few mins}) \)
- Linearity KV vs Athena WrapCPU/evt/core Sim jobs

---

**Details of the Analysis**

- 3 CPU models
  - 'AMD 4171', Intel 'E5-2673 v3' and 'E5-2660 0'
- 3 Azure DCs (EU north and west, US center)
- Simulation jobs: EVNT input collections
Correlation DB12 vs KV

DB12 Vs KV

- Good Linearity at first order proved across different CPU models
  - But discrepancies measured at finer-grained level (see next slides)

- Azure A1 series
- Azure A3 and D3 series
Systematic Studies Under Controlled Conditions
240 Servers Profiled @ CERN

Servers installed in the Wigner Data Centre have been benchmarked when still under commissioning (June/July ‘16)

– Now the servers are in production, assigned to the batch resources
– Each server consists of
  • 2 Intel Xeon Processor E5-2630 v3 CPUs @2.4GHz (Haswell);
  • 8 cores (16 threads) per CPUset ⇔ 32 vCPUs available when SMT enabled
  • RAM 64 GB, SSD disk

Server CPU performance benchmarked
– physical hardware & virtualized resources
– Benchmarks: HS06, KV, DB12, Whetstone (wsn)
  • Objective: assess the scale factors among configurations
Benchmarking Workflow

Benchmarking approach
- Run benchmarks in each VM
- Synchronization among VMs by cronjobs
- Pile-up of consecutive benchmarks avoided with lock files

HS06 not integrated in the suite
- Adopted similar approach but bmk run done with custom scripts

Test duration:
- wsn :<< 1min ; DB12: ~1 min ; KV: ~7 min
- HS06: ~6h

Example:
- wsn – 0,30 * * * *
- DB12– 5,35 * * * *
- evt_kvcpu – 15,45 * * * *

For more details about the Benchmarking Suite see the C. Cordeiro talk
Configuration Tested

Several configurations adopted
- Partitioning the available logical cores across VMs of different sizes

- Benchmarks per VM always running a # parallel workloads == # vCPUs

- Naming convention
  - VMn (n=32,16,8,4,1) identifies a VM with n vCPU

- Number of VMn per server == 32/n

- A VM uses a single NUMA node
  - Exception: VM$_{32}$

- VM image:
  - Scientific Linux CERN SLC release 6.8 (Carbon)

- VMs are re-provisioned when configuration changes
  - VM resize is not used

NB: those are just simplified pictures to describe the partitioning
In Addition 2 VM\textsubscript{8} Configurations

- 4x VM\textsubscript{8}
- 3x VM\textsubscript{8}
  - Easy to obtain just destroying a VM\textsubscript{8} per pnode
  - NUMA tuning (1 NUMA node per VM) results in 1 VM in a processor and 2 VMs in the other processor
  - VMs identified with labels
    - VM\textsubscript{8A}: the single VM in one processor
    - VM\textsubscript{8T}: the two VMs in the other processor
Three classes of results
- KV
- DB12
- wsn

<table>
<thead>
<tr>
<th></th>
<th>Bare Metal</th>
<th>VM_{32}</th>
<th>VM_{16}</th>
<th>VM_{8}</th>
<th>VM_{4}</th>
<th>VM_{1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running period</td>
<td></td>
<td>27 h</td>
<td></td>
<td>93 h</td>
<td>44 h</td>
<td>147 h</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57 h</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>245 h</td>
</tr>
<tr>
<td></td>
<td>~1.4M</td>
<td>~580k</td>
<td>~230k</td>
<td>~78k</td>
<td>~62k</td>
<td>~140k</td>
</tr>
</tbody>
</table>

Number of results for each configuration
Scale Factors Across Configurations

Reference value: Average performance of VM\textsubscript{8A}

\[ \text{Ratio}_{\text{bmk}_a} = \frac{\mu(\text{bmk}_a, \text{VM}_a)}{\mu(\text{bmk}_a, \text{VM}_8)} \]

Double peaks in KV distributions discussed in next slides
Scale Factors Across Configurations

Reference value: Average performance of VM\(_{8A}\)

\[
\text{Ratio}_{\text{bmk}_a} = \frac{\mu_{(\text{bmk}_a,\text{VM}_x)}}{\mu_{(\text{bmk}_a,\text{VM}_{8A})}}
\]

- NB: error bars are included, but small because represent error on the average
- Double peaks in KV distributions discussed in next slides
HS06 measurements

Studied three configurations:

- HS06 for each server (32 logical core) from procurement DB
- HS06 for VM\(_8\) and VM\(_{16}\): the two VM sizes mainly used for batch resources
  - VM results aggregated by physical node (server)

Scaling factor within expectations:

- HS06 for VM\(_8\) is \(\frac{1}{4}\) of HS06 full node
- HS06 for VM\(_{16}\) is \(\frac{1}{2}\) of HS06 full node
Synthetic Benchmarks: A Closer Look
Correlation plots among different benchmarks

- For each VM measurements within the same ½ hour are correlated
  - So 1 KV, 1 fbmk, 1 wsn

How to read the plots:
- 2D scatter plot
- 1D projections

How to explain the 2 peaks in KV?

KV Vs DB12
VM\textsubscript{16}: two peaks in KV results

- Verified that systematically 1 of the 2 VMs performs better than the other
  - This behavior is true for mainly all nodes
- Found that the second CPUset is systematically the slowest one

\% of tests (for each pnode) with a given VM performing faster than the other

- Quantified the \textit{<difference>} of performance among VMs in the two peaks: \textasciitilde2\%
What about DB12 and wsn

The same behavior is not seen in the other two benchmarks

% of tests (for each pnode) with a given VM performing faster than the other

$\text{<difference>}: < 0.4\%$

$\text{<difference>}: < 0.8\%$

% of tests (for each pnode) with a given VM performing faster than the other
Explanation of the Double Peak

Interesting findings from M. Guerri

- More details in his talk.

Understood that the second CPUset has the first thread (physical core) systematically slower than all the others

- This is thread 8, counting the threads of the first CPUset from 0 to 7
- When SMT is enabled, also thread 24 shows the same effect
- The difference in performance when running KV separately on each thread is at the level of 16%
  - Measured on VMs, pinning the vCPUs to the threads of the two CPUsets

- Considering that in standard operation
  - the vCPUs of the VMs are not pinned to any thread,
  - VMs are “confined” to a NUMA node
the effect on a VM_{4,8,16} is weighted by the other 7 physical cores: 2% (VM) = 16% (thread) / 8
Repeated measurements

Investigated if repeated measurements are compatible with a single average behavior

Deviations from the average behavior sign of
  – a change of external conditions in the VM (and/or server)
  – due to non-gaussian tails

– Main question: resolution expected in the benchmark measurement:
  • 2% vs 20% ?

![Graph showing time sequence and 95% C.I. for VM measurements]
Summary

• In cloud environment the VM performance is **highly variable**
  – Changes with the environment (load on the IaaS) or benchmark characteristics

• At first order KV and DB12 show good linearity, as well as KV and Sim jobs of ATLAS
  – Not necessarily true with less CPU intensive workloads
  – Fine-grained resolution effects (~2%) seen in KV and not in the other synthetic benchmarks

• HS06 running on VMs (8 and 16) measures the expected performance factor (bias <5%) respect to the HS06 measurement on bare-metal

• Scale factors across VM sizes within expectations for KV and DB12
  – It is not the case for Whetstone (see VM8_A configuration)
VM$_8$: KV vs fbmk

Two main peaks seen in both plots

- The fastest VM in each pnode is the one in a single processor
- At first order the two benchmarks scale of a factor ~2

• Fine-grained structures appear too
  - In particular for KV

Peak positions
KV VM$_{8T}$: 0.73
KV VM$_{8A1}$: 1.40
KV VM$_{8A2}$: 1.38

fbmk VM$_{8T}$: 9.8
fbmk VM$_{8T}$: 21.2