CPU BENCHMARK TEST RESULTS

FOR JOYENT
Revision 9

October 13th, 2010

Scope:

This report summarizes the CPU benchmark testing performed in August and September of 2010.

References:

[1]: http://blog.cloudharmony.com/2010/05/what-is-ecu-cpu-benchmarking-in-cloud.html

[2]: Svn repository: https://svn.codespaces.com/ims/joyent-benchmarks
username: joyent **password:** joyent

[3]: Raw test data: CPU_Final_Results.xlsx

[4]: Phoronix Test Suite 2.6.1:
http://www.phoronix-test-suite.com/download.php?file=phoronix-test-suite-2.6.1

[5]: http://byte-unixbench.googlecode.com/files/unixbench-5.1.2.tar.gz

Joyent CPU Benchmark Testing Report

Introduction

The CPU testing was performed as part of a larger benchmark effort intended to provide a basis for comparison between Joyent SmartMachines and other virtual servers offered by cloud service providers.

Earlier in 2010, CloudHarmony engaged in an extensive benchmarking effort intended to provide "information and analysis to enable educated decisions pertaining the adoption of, and migration to cloud services". Their results and analysis are presented in a series of articles published online ref[1]. The CloudHarmony blog does not contain results for Joyent SmartMachines. Our testing procedures are intended to follow CloudHarmony's efforts as closely as possible and extend benchmarking for Joyent SmartMachines.

Instead of trying to reproduce all of the CloudHarmony results, we focused on those outlined for the Amazon EC2 servers used in their benchmark tests ref[1]. Our tests closely approximate the methods from CloudHarmony in regards to calculations and tests used. Figures for the Joyent SmartMachines should be a useful addition to the other benchmarks included in CloudHarmony's blog. It should be noted that not all test executables and versions contained in this report are identical to those of CloudHarmony due to differences in operating systems. These results should not be compared side-by-side to those of CloudHarmony. Our mathematical baselines and server instances are however identical.

CloudHarmony standardized on CentOS 64bit as the operating system used for test servers except where it was unavailable. Joyent SmartMachines run an OpenSolaris based operating system. Due to the original tests being Linux based, modifications were necessary to execute similar binaries on the Solaris based system.

SmartMachines provide a "bursting" capability enabled by their OS that allows a service to use more processor and memory resources on a temporary basis than the guaranteed minimum. This differs from nearly all other cloud providers that

provide a fixed processor/memory configuration. While bursting capability can be a tremendous advantage to an operational system, it can complicate benchmark testing by stressing the system to its maximum capacity. On the Joyent SmartMachines the bursting capability allows a process on even the smallest server to potentially use nearly the entire compute and memory capability of the underlying hardware.

Joyent uses large commodity servers with 8 hyper-threaded processors that effectively yield 16 processor cores. This means that Joyent's smallest server, the 1 GB SmartMachine, can in some cases outperform Amazon's largest EC2 instance, m2.4xLarge. Due to this bursting capability, side-by-side comparisons may not be identical in nature between Joyent and other cloud providers.

Benchmark Setup

Amazon EC2 was used as our primary baseline benchmark for all CPU tests. The servers used consist of: m1.small, c1.medium, m1.large, m1.xlarge, m2.xlarge, c1.xlarge, m2.2xlarge, m2.4xlarge. All Amazon servers – 8 servers in 4 regions, were configured identically in terms of OS, CentOS 5.4 64-bit (or 32-bit in the case of EC2 m1.small and c1.medium where 64-bit is not supported). Joyent servers included: 1GB, 2GB, 4GB, 8GB, 16GB, 32GB.

All SmartMachines come with Joyent's SmartOS based on OpenSolaris - SunOS 5.11 snv_121. On the Joyent servers, the default gcc compiler is version 3.4.3 and the Amazon instances come equipped with gcc version 4.1.2. To provide comparison with an out-of-box install, the default software versions were used.

To run the majority of benchmark tests, CloudHarmony made use of the Phoronix Test Suite ref[1]. Version 2.6.1 was used for Joyent's OpenSolaris compatibility. There are several differences between version 2.2.0 used by CloudHarmony and 2.6.1 used in this report. These include test versions, source code, and executables.

The Phoronix Test Suite open source framework was originally a Linux specific tool and streamlines the testing procedures. Many of the included tests run natively under Linux, but not on Solaris as noted in the Phoronix documentation. Since the test suite makes use of shell scripts to download, unpack, build, install, and run the various tests, some modifications were necessary to run identical tests on the Joyent SmartMachines. Scripts within the Phoronix Test Suite make use of the Bourne Shell by specifying the #!/bin/sh command interpreter. Normally including #!/bin/sh works fine on most Linux systems, but under Solaris this is not the case.

Making the minor change from /bin/sh to /bin/ksh on Solaris can be made easily with a global search and replace to make all scripts compatible. Some other modifications require manual inspection and correction – versions and packages of required software are different between Solaris and Linux. There were also path changes needed in the test suite code to link to the appropriate executables in the Joyent SmartMachine OS. All changes to the test suite have been configuration controlled and is available to Joyent via subversion ref[2].

Benchmark Tests

There are 19 benchmarks CloudHarmony used to compute the CCU comparison

metrics. Of the 19 benchmarks, 6 required Linux specific drivers or executables and were excluded from this report. Our testing results do not include:

espeak, mafft, nero2d, opstone-svd, opstone-svsp, and opstone-vsp

The following 13 tests were run and compared on both Amazon EC2 and Joyent SmartMachines:

c-ray, crafty, dcraw, geekbench, graphics-magick, hmmer, john-the-ripper-blowfish, john-the-ripper-des, john-the-ripper-md5, openssl, sudokut, tscp, unixbench

As noted, the benchmark tests on Joyent required script customizations or source code modifications: ref[2] phoronix-test-suite-2.6.1-SolarisPatched.tgz. The benchmark executables test similar functionality on Joyent to the tests CloudHarmony ran on other servers. Calculations from the test results to derive our baselines are of an identical nature between the different Operating Systems.

Testing Procedures

The Phoronix Test Suite 2.6.1 was setup on each Joyent and Amazon server to run all the CPU benchmarks except unixbench. Phoronix compiles their results in xml files to be displayed in a web browser. The suite also creates image graphs for visual comparison. Unixbench was run independently from the others with output saved to a flat-file for record keeping.

In order to reproduce our testing procedures on the Joyent SmartMachines see ref[2]: joyent_tests_directory.tgz, phoronix-user-directory.tgz, phoronix-suite-2.6.1-solaris.patch. The following guidelines should produce similar or identical test results:

1. Install the Phoronix Test Suite into a local directory within the user's folder on each server. Tar files for Joyent are included ref[2]: joyent_tests_directory.tgz, phoronix-user-directory.tgz. These two files contain all tests and executables that run on Joyent. If using these tar files, extract them into the user directory and skip to step 6.

2. If installing the default Phoronix Test Suite 2.6.1 ref[4], apply the patch file ref[2] phoronix-suite-2.6.1-solaris.patch to the test suite. This patch makes changes to the installation files and performs all alterations necessary for the Solaris platform. The test suite should be installed to a local user directory labeled *'Tests'*.

3. Install the tests via Phoronix. The tests should fail to install by default for tscp and geekbench. Perform these additional modifications necessary for the two benchmark tests.

   For tscp, manually apply a patch to the source file main.c at line 74 >> add:

   computer_side = EMPTY;
   bench();

break;

Once the additional code is added, compile the program using:

*gcc *.c -o tscp*

The remaining executable needs to be created from the install.sh file code located in the phoronix-test-suite_solaris\pts\test-resources\tscp\ directory. This file could also have been taken from another successful installation on a different operating system and change the interpreter from /bin/sh to /bin/ksh. Example of the executable is:

```
#!/bin/ksh
cd tscp181/
./tscp $@ > $LOG_FILE 2>&1
echo $? > ~/test-exit-status
```

Make the *tscp* file executable >>

chmod +x ~/.phoronix-test-suite/installed-tests/tscp/tscp

Phoronix requires another file, pts-install.xml, to also be put into the tscp directory. This tells the test suite the program has been successfully installed. Simply copy this file from a successfully installed test.

4. Geekbench needs to be manually extracted from the tarball and put into its own directory within the installed-tests/geekbench folder:

   >> ./phoronix-test-suite/installed-tests/geekbench/geekbench-solaris/geekbench_x86_32 (executable)

   Modify the Phoronix executable – geekbench – in the geekbench folder:

   >> ./phoronix-test-suite/installed-tests/geekbench/geekbench:

   ```
   #!/bin/ksh
   cd geekbench-solaris/
   ./geekbench_x86_32 > $LOG_FILE 2>&1
   echo $? > ~/test-exit-status
   ```

   If missing, place a similar pts-install.xml file in the geekbench folder.

5. Download and extract the Unixbench source ref[5] into the folder Tests/unixbench-5.1.2. Follow the directions contained in the Unixbench source code to compile and run.

6. Use the test executable script ref[2] *setup_joyent.sh* to run all tests with specific parameters and execute each CPU benchmark. It automatically packs the output files from both Phoronix Test Suite and unixbench into one tar file. Modify the last line to match an available ftp server.

Note: If tests fail to run, make the following modifications to the test suite core files to see the full executable outputs for troubleshooting:

phoronix-test-suite/pts-core/library/pts-functions_shell.php

At line 110 add:

```
echo pts_variables_export_string($extra_vars) . "\n\n";
echo $exec . "\n\n";
```

This will output the Phoronix variables and executable to the command line.

Baselines

A cumulative baseline was taken from all Amazon results and calculated based on the methodology from CloudHarmony. It should be noted that while only 13 benchmarks were used to calculate the CPU comparison score (CCS) and CloudHarmony Compute Unit (CCU) values in this test, CCS and CCU results for the Amazon baseline servers were comparable to the corresponding results posted on the CloudHarmony blog.

Test Results

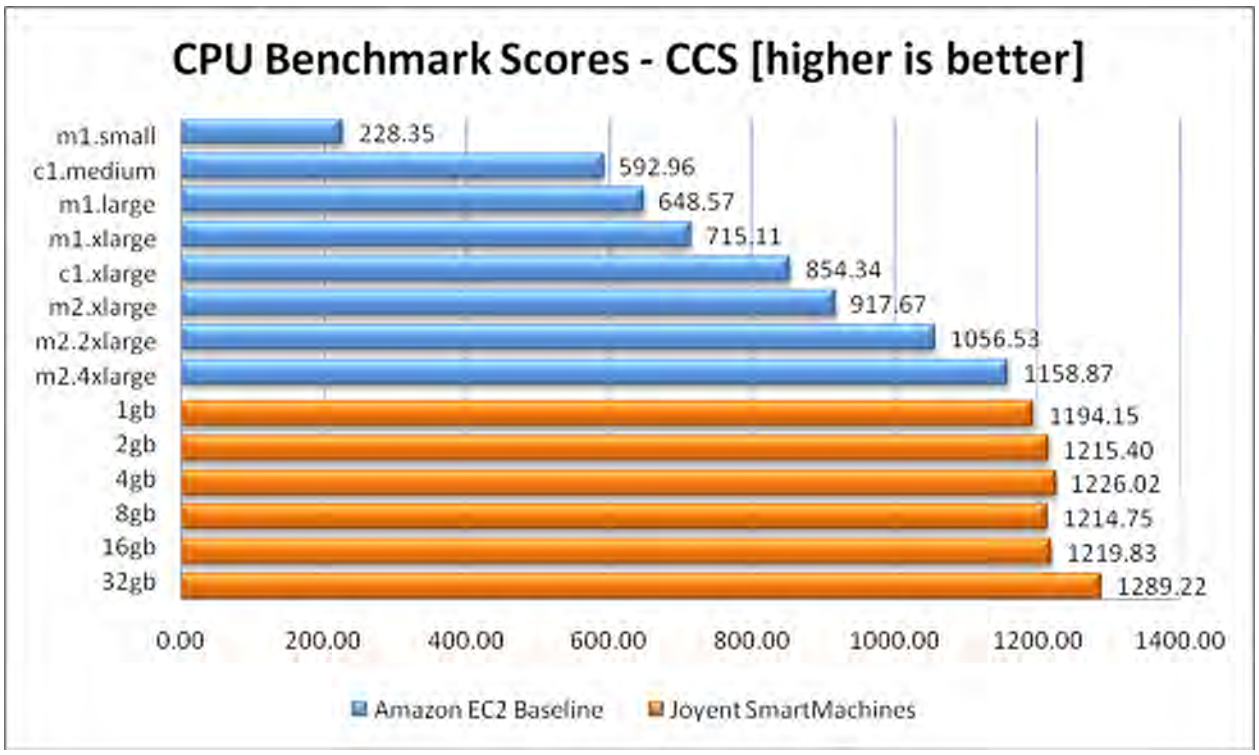
For the full raw test data and calculations see the spreadsheet ref[3]. Datasets and breakdowns from each benchmark test from the different servers are presented in several spreadsheets. Tests requiring specific Linux drivers or executables have been excluded from our results.

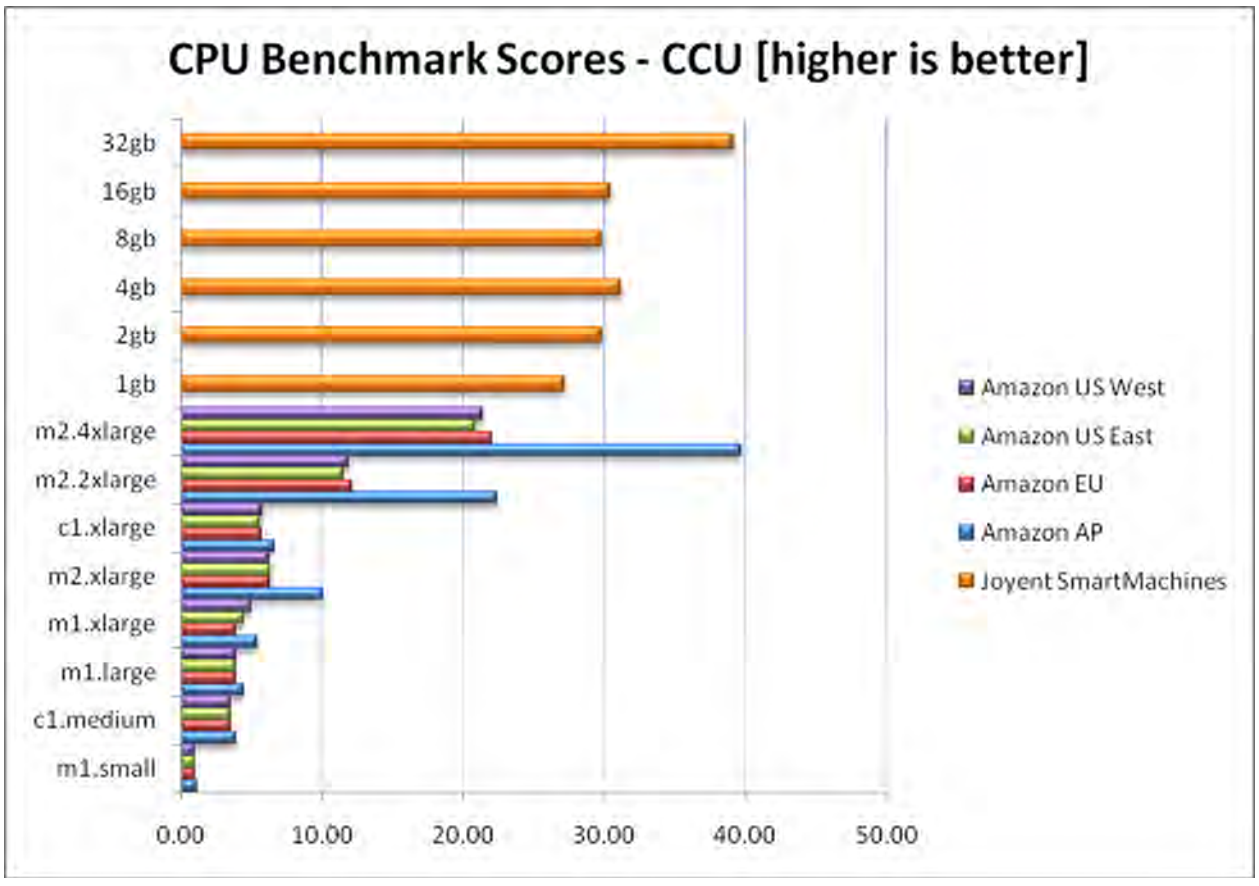
To calculate CCS and CCU please refer to the ref[3] CPU_Final_Results.xlsx document. The CCS scores are compared and calculated against an average of the Amazon server tests. Our calculations are based on and have been verified against those found on the CloudHarmony blog.

As shown in the following graphs, Joyent outperforms the Amazon EC2 servers, which may be a result of the Joyent architecture and the OpenSolaris based operating system.

The graphs below contain similar results for CCU and also outlines the scores from each individual Amazon region. A table lists each server's exact calculations.

Each metric and comparison for CCS and CCU is shown in the following graphs:


CPU Benchmark Scores - CCS [higher is better]
m1.small
c1.medium
m1.large
m1.xlarge
c1.xlarge
m2.xlarge
m2.2xlarge
m2.4xlarge
1gb
2gb
4gb
8gb
16gb
32gb
228.35
592.96
648.57
715.11
854.34
917.67
1056.53
1158.87
1194.15
1215.40
1226.02
1214.75
1219.83
1289.22
0.00
200.00
400.00
600.00
800.00
1000.00
1200.00
1400.00
Amazon EC2 Baseline
Joyent SmartMachines


CPU Benchmark Scores - CCU [higher is better]
32gb
16gb
8gb
4gb
2gb
1gb
m2.4xlarge
m2.2xlarge
c1.xlarge
m2.xlarge
m1.xlarge
m1.large
c1.medium
m1.small
0.00
10.00
20.00
30.00
40.00
50.00
Amazon US West
Amazon US East
Amazon EU
Amazon AP
Joyent SmartMachines

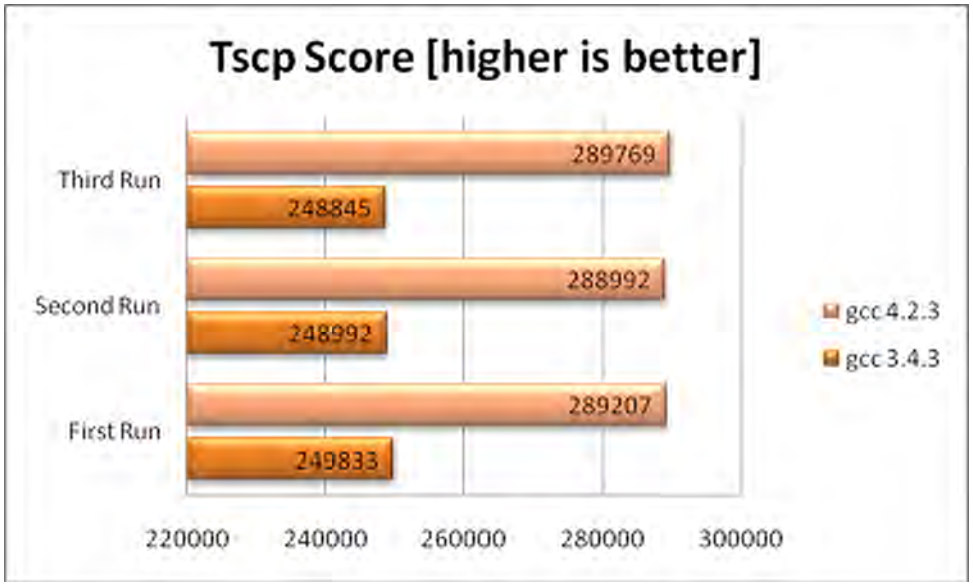
	m1.small	c1.medium	m1.large
Amazon AP	1.15	3.92	4.41
Amazon EU	0.98	3.52	3.85
Amazon US East	0.97	3.49	3.91
Amazon US West	0.96	3.53	3.92

	m1.xlarge	m2.xlarge	c1.xlarge	m2.2xlarge	m2.4xlarge
Amazon AP	5.40	10.05	6.60	22.40	39.69
Amazon EU	3.92	6.30	5.67	12.19	22.02
Amazon US East	4.43	6.30	5.57	11.49	20.89
Amazon US West	4.96	6.21	5.76	11.86	21.39

	1gb	2gb	4gb	8gb	16gb	32gb
Joyent SmartMachines	27.20	29.85	31.24	29.85	30.48	39.19

Since the majority of our tests are compiled using Gcc, we needed to compare the impact of each operating system's default version. The Gcc version packaged with the Joyent SmartOS is 3.4.3, while the default Amazon CentOS version is 4.1.2. We compiled and ran our Tscp benchmark spot tests with different versions of Gcc and recorded the results. The comparison test scores between Gcc 3.4.3 and 4.2.3 show an approximate 14% increase in performance on both Joyent and Amazon EC2. Gcc 4.1 and 4.2 showed no difference in the results. Should the default Gcc version be upgraded to 4.x, the Joyent SmartMachine scores should be significantly higher.

The following graph outlines our scores from the GCC comparison test:



Conclusion

From our calculations and test results, Joyent outperforms the established baselines across the Amazon instance sizes in terms of CPU performance. Since Joyent's bursting capability allows a process to potentially use nearly the entire CPU and memory capacity of the underlying hardware, Joyent's smallest server, the 1 GB SmartMachine considerably outperforms Amazon's smallest, m1.small, server.

The most equivalent comparison among the results is Joyent's largest server, the 32 GB SmartMachine and Amazon's EC2 m2.4xlarge instance. Both cloud servers have the same effective 16 processor cores and a comparable amount of memory. While analyzing these two similar servers, the Joyent SmartMachine easily outperformed the average of Amazon instances. It should be noted that an Amazon m2.4xlarge instance from the Asia Pacific region produced higher than normal results.

While comparing the Joyent SmartMachine CCU values, the 1GB to 32GB server sizes show little variation in CPU performance with an increase in server capacity. Amazon servers however showed a consistent increase in the CCU values in the test results from small to large based on server capacity. The Joyent SmartMachine test scores were expected due to their CPU bursting capability and unique architecture.