UBUNTU MEMORY I/O BENCHMARK TEST RESULTS

FOR JOYENT
Revision 5

January 21st, 2010

Scope:

This report summarizes the Memory Input Output (IO) benchmark testing performed in December of 2010 for Joyent Ubuntu Linux cloud servers.

References:

[1]: http://blog.cloudharmony.com/2010/06/cloud-server-benchmarking-part-4-memory.html

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

[3]: Raw test data: MEMORY-IO_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://www.alasir.com/software/ramspeed/ramspeed-2.6.0.tar.gz

[6]: http://www.cs.virginia.edu/stream/ref.html

[7]: http://www.streambench.org

Joyent Memory IO Benchmark Testing Report

Introduction

The Memory IO testing was performed as part of a larger benchmark effort intended to provide a basis for comparison between the Joyent Ubuntu 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 the Joyent Ubuntu servers. Our testing procedures are intended to follow CloudHarmony's efforts as closely as possible and extend benchmarking for the Joyent servers.

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 Ubuntu servers 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 calculations for the baseline numbers and server instances are however identical.

CloudHarmony standardized on CentOS 64bit as the operating system used for baseline tests except where it was unavailable. The Joyent Ubuntu servers run version 10.04 of the operating system.

The Joyent servers provide a "bursting" capability that allows a service to use more processor resources on a temporary basis than the guaranteed minimum. This differs from nearly all other cloud providers that provide a fixed processor configuration. While bursting capability can be a tremendous advantage to an operational system, it can complicate benchmark testing which will stress the system to its maximum capacity. On the Joyent Ubuntu servers the bursting capability allows a process on even the smallest server to potentially use nearly the entire compute capability of the underlying hardware.

The Joyent Ubuntu servers use large commodity servers with an available 4 hyper-threaded processors that effectively yield 8 processor cores. This means that Joyent's smallest server, the 1 GB, can in some cases outperform Amazon's largest EC2 instance. Due to this bursting capability, side-by-side comparisons may not be identical in nature between the Joyent Ubuntu servers and other cloud providers. Our conclusions outline the similar comparison between Joyent's 8GB Linux server and Amazon's EC2 c1.xlarge instance which both yield 8 total cores.

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 Ubuntu servers included: 1GB, 2GB, 4GB, 8GB, 16GB.

To run the majority of benchmark tests, CloudHarmony made use of the Phoronix Test Suite ref[1]. Version 2.6.1 was used for compatibility and comparison with our benchmarks performed on the Joyent Ubuntu Servers ref[2]. 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. Our tests however used version 2.6.1 on all servers including the baselines.

The Joyent Ubuntu servers utilize version 10.04 and only required minor tweaks to the phoronix test suite.

Benchmark Tests
There are 7 benchmarks CloudHarmony used to compute the CCU comparison metrics. All of the tests ran properly on the Joyent Ubuntu servers, but hdparm cached read was excluded due to it's disk IO functionality and not memory IO. The 6 remaining tests included in this report are:

> *Unixbench 5.1.2, redis-benchmark 2.0.1, ramspeed 2.6.0, cachebench, geekbench, and stream*
>
> *The default versions contained in Phoronix-Test-Suite 2.6.1 were used.*

Testing Procedures
The Phoronix Test Suite 2.6.1 was setup on each server to run cachebench, redis-benchmark, ramspeed, and stream. Unixbench, required manual installation on the

test machines. Phoronix compiles their results in xml files to be displayed in a web browser. The suite also creates image graphs for visual comparison. When not using Phoronix for testing, the output was saved to a flat-file for record keeping.

In order to reproduce our testing procedures on the Joyent Ubuntu servers see ref[2]: joyent_ubuntu_memory_test.sh, phoronix-test-suite-JoyentUbuntu-2.6.1.tar.gz. 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 Ubuntu are included ref[2]: phoronix-test-suite-JoyentUbuntu-2.6.1.tar.gz. This tar file includes the small tweaks required for Joyent's Ubuntu servers. If using this tar file, extract into the user directory and skip to step 5.

2. If installing the default Phoronix Test Suite 2.6.1 ref[4], apply the patch file ref[2] phoronix-suite-2.6.1-ubuntu.patch to the test suite. This patch makes changes to the installation files and performs the minor alterations necessary for the Ubuntu servers.

3. One additional modification is needed for the phoronix suite to properly access the root system. Run these commands:

   cd ~/phoronix-test-suite/pts-core/test-libraries/
   mv root-access.sh root-access.sh.orig
   sed 's/\\n/\n/g' root-access.sh.orig > root-access.sh
   chmod +x root-access.sh

4. After extracting the tar file in the user's local directory, run the following commands to install the phoronix tests to run:

   cd ~/phoronix-test-suite
   ./phoronix-test-suite install cachebench ramspeed stream

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.

6. Run the benchmarks manually via phoronix and UnixBench or use the test executable script ref[2] joyent_ubuntu_memory_test.sh to run all the tests. The script automatically packs the output files from each test result. Modify the last line to match an available server for uploading or download the tar file manually. Memory IO tests on each server take approximately 4 hours to complete.

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
The NewServers Jumbo server was assigned a score of 100. All other servers were assigned a score proportional to the performance of that server, where greater than 100 represents better results and less than 100 represents poorer results. For example, a server with a score of 50 scored 50% lower than the baseline server overall, while a server with a score of 125, scored 25% higher. It should be noted that hdparm-cached-read benchmarks was not used to calculate the MIOP score in this test because it was not run on the Joyent Ubuntu Servers. The MIOP were comparable to the same results posted on the CloudHarmony blog.

Test Results
The test scores of MIOP were compared to those of the NewServers baseline and a ratio was created based on a percentage of each result to that of the baseline.

Overall tests show the Joyent Ubuntu Servers to be the highest, even beating the NewServers baseline. These results are surprising due to the NewServers Jumbo server running without virtualization and direct access to the physical hardware.

Joyent Ubuntu Servers have a bursting capability which shines in the comparison of the Amazon EC2 m1.small server and the 1GB Ubuntu Server. The most comparable servers are the Amazon EC2 c1.xlarge with 7GB and Joyent 8GB.

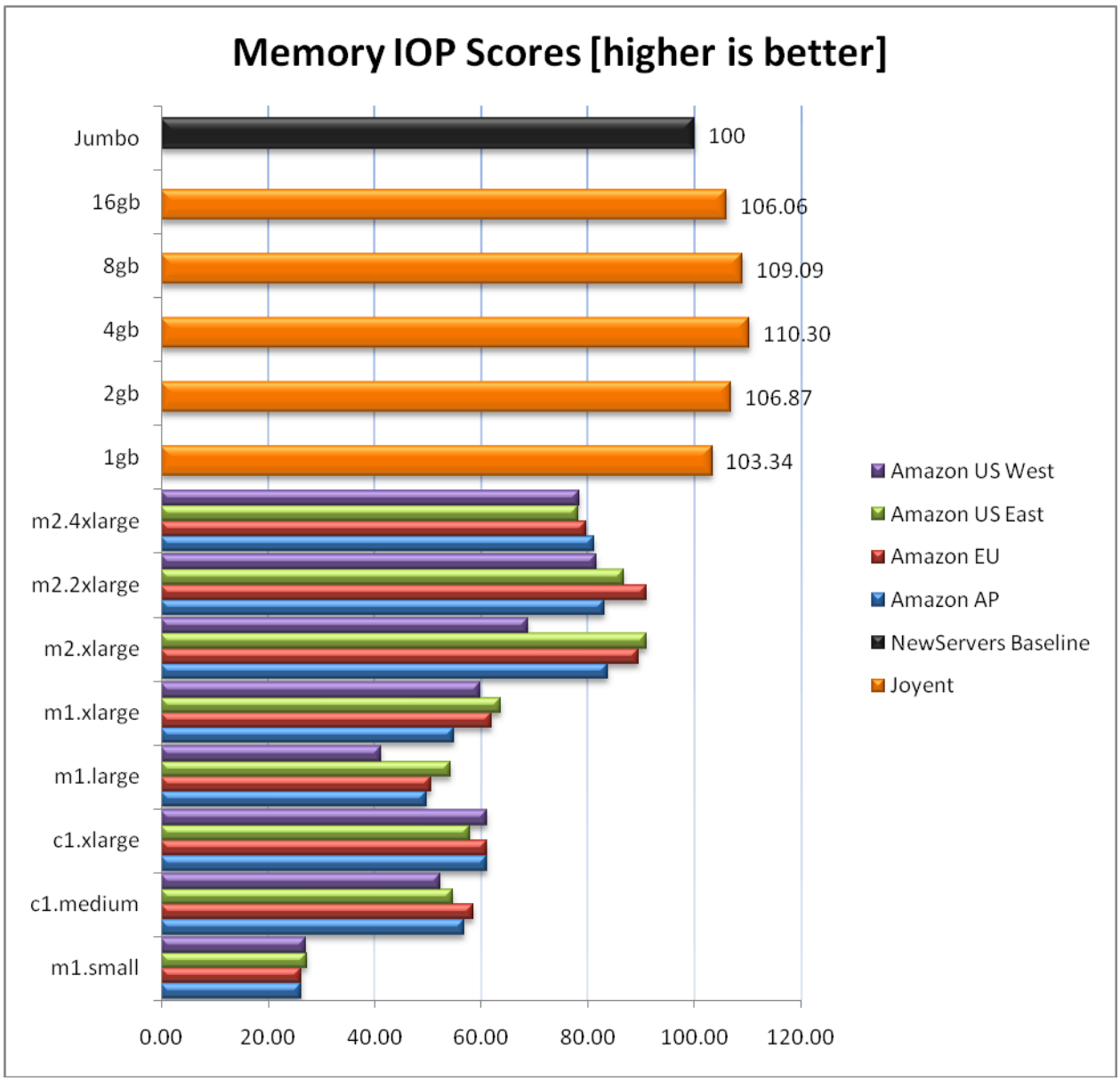
The two following graphs and data illustrate the above calculations and results:

Memory IO, calculated as MIOP ref[1], for the Joyent Ubuntu Servers:

	m1.small	c1.medium	m1.large	m1.xlarge
Amazon AP	26.19	56.72	49.77	54.94
Amazon EU	26.28	58.44	50.57	61.99
Amazon US East	27.38	54.66	54.31	63.65
Amazon US West	27.11	52.21	41.31	59.76

	m2.xlarge	c1.xlarge	m2.2xlarge	m2.4xlarge
Amazon AP	83.81	61.03	83.19	81.23
Amazon EU	89.48	61.04	91.09	79.71
Amazon US East	91.08	57.85	86.78	78.22
Amazon US West	68.79	61.19	81.63	78.35

	1GB	2GB	4GB	8GB	16GB
Joyent Ubuntu	103.34	106.87	110.30	109.09	106.06



Conclusion

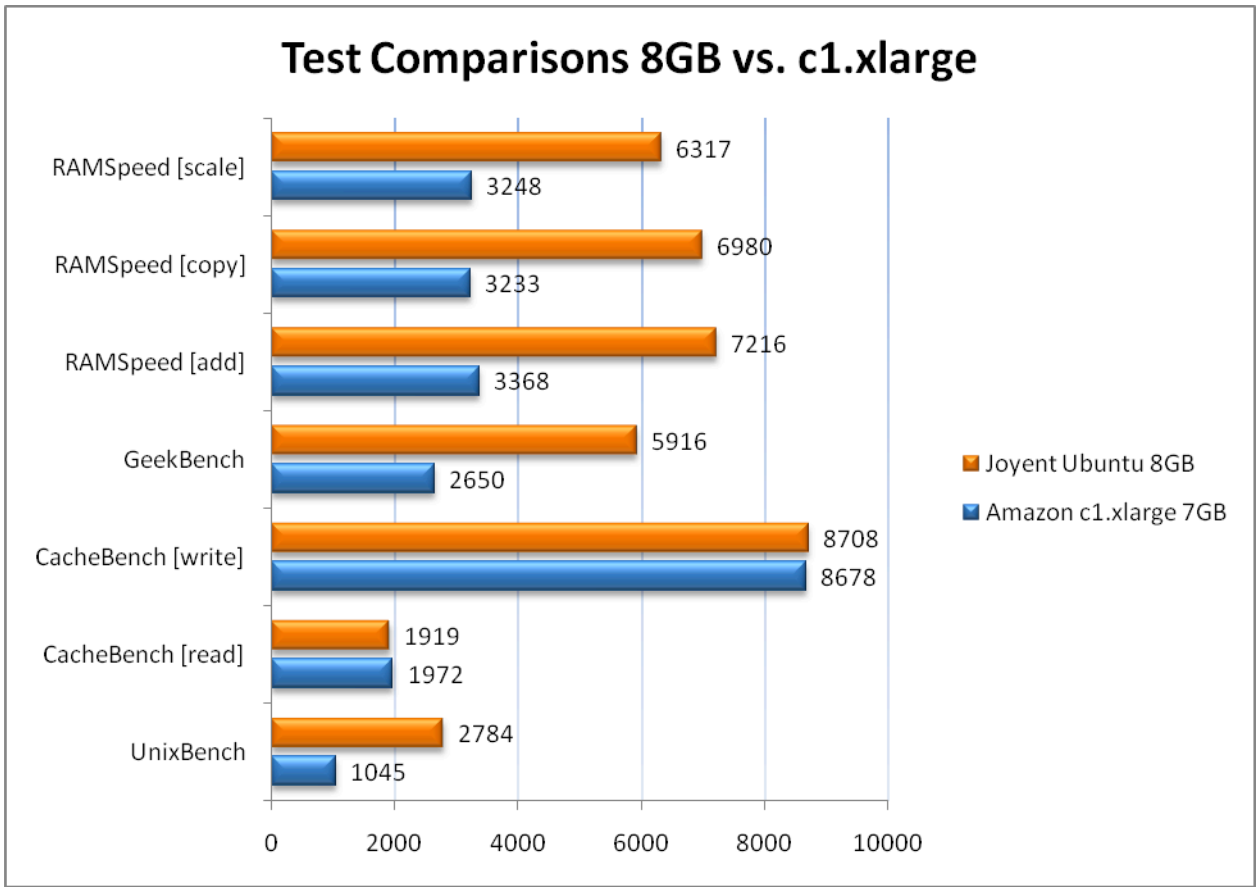
The variation in MIOP values between the 1GB and 16GB Joyent Ubuntu Servers did not show a strong linear increase in Memory IO performance with the increase in server capacity. Overall, the Joyent servers outperformed all the Amazon server instances in our comparisons.

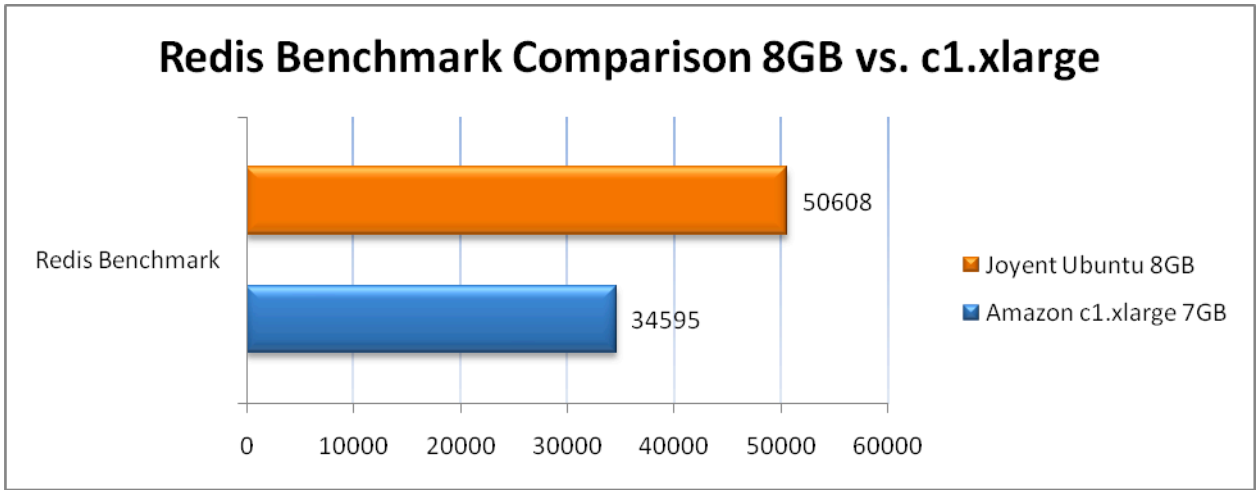
While comparing memory throughput to the NewServers baseline, the Joyent 16GB server still outperforms the baseline which is surprising due to the Jumbo server having an available 48GB of memoy and improved hardware.

The two most comparable server instances are the Joyent 8GB and Amazon c1.xlarge. We compared the c1.xlarge highest score from the US West region of Amazon. Both servers have an available 8 CPU cores and 8GB or 7GB of available

memory. Unfortunately, Amazon does not offer a server instance with 8GB of memory, but this small 1GB difference should not effect Memory IO throughput.

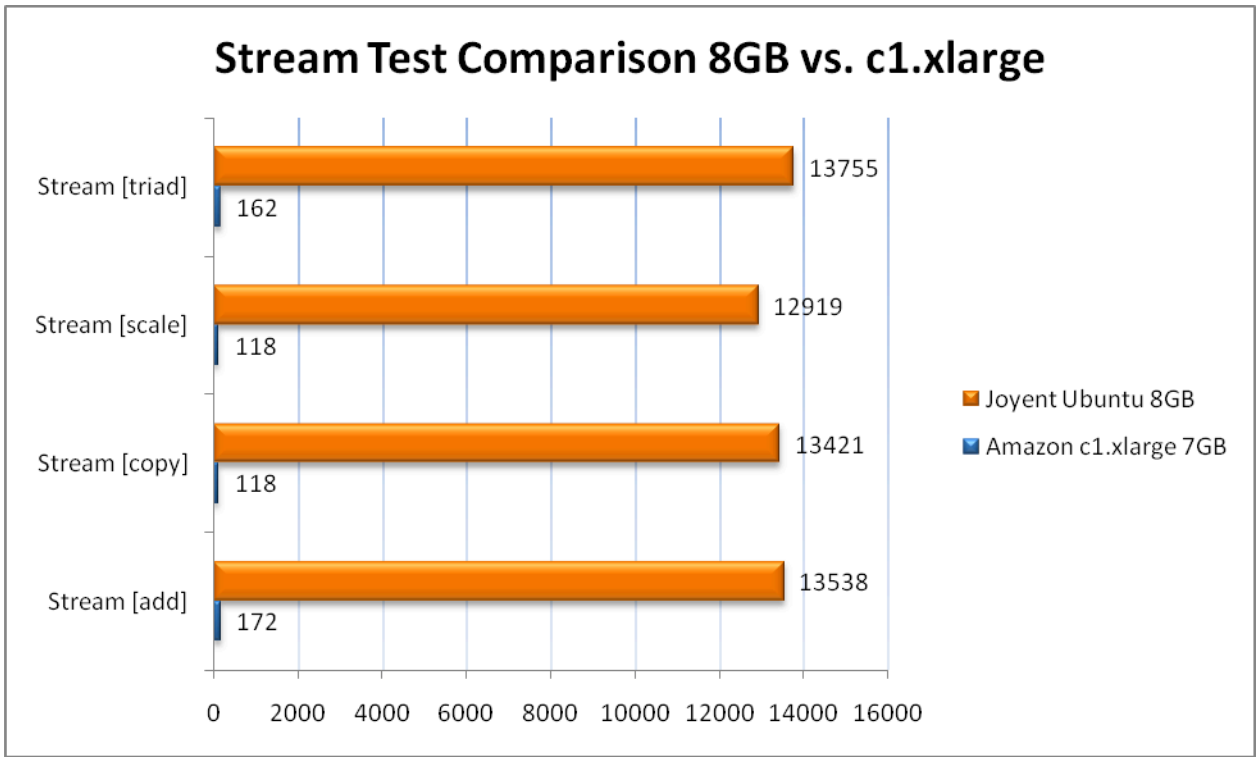
Upon analyzing the difference in scores, the Joyent Ubuntu 8GB server outperformed the Amazon c1.xlarge except with CacheBench, which the two servers scored almost identical. The following graphs give a visual representation of each test performed:





The most surprising result comparison is with the Stream benchmark. From the developer website, this benchmark is described as:

> *The STREAM benchmark is a simple synthetic benchmark program that measures sustainable memory bandwidth (in MB/s) and the corresponding computation rate for simple vector kernels. Ref[7].*
>
> *The STREAM Benchmark is the de facto industry standard benchmark for the measurement of computer memory bandwidth. The STREAM benchmark measures "real world" bandwidth sustainable from ordinary user programs -- not the theoretical "peak bandwidth" provided by most vendors. The STREAM benchmark archive currently has more than 1000 results, dating from 1991 to the present. Ref[8].*



It is clear to see how the difference in hardware and available operating system technologies put the Joyent Ubuntu Servers above the comparisons to Amazon EC2 default instances.

Additionally, Amazon servers, as expected, showed a consistent increase in the MIOP values in the test results from small to large based on server capacity. Individual memory tests however showed variance between the servers which did not follow a linear pattern. These results are comparable to those found on the CloudHarmony blog. The Joyent Ubuntu servers show a small variation between servers, but the throughput of memory IO is consistent with their bursting technology.