Performance Benchmark for Cloud Block Storage

J.R. Arredondo

vJune2013
Contents

• **Fundamentals** of performance in block storage
• **Description** of the Performance Benchmark test
• **Cost** of performance comparison
• **Charts and data!**
  – **Charts**: RAX vs. RAX (CBS SATA, CBS SSD, Local Storage)
  – **Charts**: RAX vs. AWS (CBS SATA vs. EBS Standard)
  – **Charts**: RAX vs. AWS (CBS SSD vs. EBS Provisioned IOPS)

• **Appendix:**
  – **Basics** of Cloud Block Storage
The Bottom Line
Things to remember about Cloud Block Storage performance

More throughput

“It can get more work done in a given amount of time”

Less latency

“It does it fast”

More consistency

“It does it consistently fast”

Not just fast, but consistently fast

Pricing is simple

Never pay for I/O
Example: AWS EBS Standard delivers 20% of the performance of RAX CBS SATA, and only 17% of the performance of RAX Local storage.
Example: Rackspace CBS SSD and CBS SATA deliver low variability of performance
Performance Fundamentals
Crash course on **a few** of the things that impact performance

<table>
<thead>
<tr>
<th>DEVICE TYPE</th>
<th>Instance Storage vs. SATA vs. SSD vs. dedicated storage (SAN, NAS, DAS)</th>
</tr>
</thead>
</table>
| IO SCHEDULER | /sys/block/YOUR_Device/queue/scheduler  
Options: noop, anticipatory, deadline, cfq |
| FILE SYSTEM  |  
• **ext2**: no journal  
• **ext3**: journal, online resize  
• **ext4**: extents (contiguous blocks), barriers (writeback cache), checksum, no inode locking, max file size 16TB  
• **xfs**: data=writeback, barriers, delayed allocation  
Others: btrfs, zfs |
| MOUNTING OPTIONS |  
• **noatime**: do not update access times metadata  
• **nobarrier** (xfs) or barrier=0 (ext3, ext4): don’t put barriers to wait for write acknowledgment  
• **logbufs** (xfs): in memory log buffers |
| YOUR APPLICATION! | |
## Caveat: A benchmark is not “real world”

<table>
<thead>
<tr>
<th>What it is</th>
<th>What it is not</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Idealized measurement</td>
<td>• Exactly what the customer or application will see</td>
</tr>
<tr>
<td>• Guideline</td>
<td></td>
</tr>
</tbody>
</table>

**THIS CANNOT BE OVERSTATED**

Always benchmark your own application!
Performance basics
There is a difference between fast and consistently fast

\[ \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \]

\[ SD = \sqrt{\frac{\sum (x - \bar{x})^2}{N - 1}} \]

**AVERAGE LATENCY**
The ORANGE and GREEN dots appear to have somewhat close average latency.

Lower Latency Average (“faster”) is better.

**CONSISTENCY**
The GREEN dots show higher performance consistency (lower variability).

The Standard Deviation is higher for the ORANGE dots (more variability).

Lower Standard Deviation (“more consistent” or “less variable”) is better.

**NOTE:** Not all variability may be “bad.” Some variability is “good” (for example, if your storage volume responds faster for a little while you likely won’t complain).

So in addition to the Standard Deviation we will also calculate the **Semi Standard Deviation** as that “bad” variability (that variability on the “bad” side of the average, or higher than the average).
Performance basics in storage:
Higher throughput (Megabytes/second) is better (as in “it can get more work done”)

Higher throughput (good)

Lower throughput (not as good)
Performance basics in storage: Lower latency (milliseconds) is better (as in “it works fast”)

- Lower latency (good)
- Higher latency (not as good)

Notice Y axis (latency) is inverted (top is better)

X axis is just the time of the test and sample (seconds)

Watch the scale (varies for sequential and random IO)
Performance basics in storage:
Less variability (standard deviation) is better (as in “it is consistent”)

Little variability (good)
Lots of variability (not as good)
The Tests
A single test is a specific combination of…

- A storage flavor
- A file systems
- Specific file system mounting options
- An access mode
- Thread count

“RAX CBS SATA”  “XFS”  “defaults”  “Random Read/Write”  32
Storage “flavors” measured

Rackspace

- Cloud Servers
- Local Storage
- CBS SATA
- CBS SSD

AWS

- EBS Standard Volumes
- 500 Provisioned IOPS
- 1000 Provisioned IOPS
- 2000 Provisioned IOPS
- 3000 Provisioned IOPS
- 4000 Provisioned IOPS

In this test we tested standard instances (EBS-Optimized instances will be tested in a next phase)
File Systems measured

- Ext3
- Ext4
- XFS
Mount options

**Ext3**
- defaults
- noatime,barrier=0
  (this is the default mount options of Local Storage in Cloud Servers)
- barrier=1

**Ext4**
- defaults, rw

**XFS**
- noatime
- noatime,nobARRIER

Note that Cloud Servers Local Storage was measured always with these file system and mounting default options of Cloud Servers setup, even as we compare it with other setups (say, ext4 mounted with “defaults”).
Access modes tested
(and some examples of typical applications for each access mode)

- Reporting app: Random Read
- Typical database: Random Read/Write (about 60/40 ratio)
- Batch updates: Random Write
- Import operation: Sequential Read
- Some queries: Sequential Write
- Backup or extract: Sequential Write
For every Flavor, File System, Mount Option and Access mode combination, various thread counts were used.
Only synchronous access was tested

Synchronous

Asynchronous
Here is the full measurements matrix (not all are included in this deck)

<table>
<thead>
<tr>
<th></th>
<th>RAX SATA</th>
<th>RAX SSD</th>
<th>RAX Local</th>
<th>EBSSTD</th>
<th>EBS P500</th>
<th>EBS P1000</th>
<th>EBS P2000</th>
<th>EBS P3000</th>
<th>EBS P4000</th>
</tr>
</thead>
<tbody>
<tr>
<td>XFS (noatime, nobARRIER)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XFS (noatime)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXT4 (defaults, rw)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXT3 (defaults)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXT3 (noatime, barrier=0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXT3 (barrier=1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Server and Storage

### RAX
- 500GB SSD and SATA volumes
- 1.2TB local storage of 30GB Cloud Server (30GB, 8 vCPUs, 600 Mbps internal network)

### AWS
- 500GB volumes for each of
  - EBS Standard and Provisioned IOPS 500, 1000, 2000, 3000 and 4000
- High Memory Double Extra Large (m2.2xlarge) EC2 Server (34.2 GiB, 13 EC2 Compute Units across 4 Virtual Cores)
- This test does not include EBS Optimized instance (future)

---

### CentOS 6.3
- cfq scheduler for all devices
Benchmark tool: SYSBENCH 0.5

- Sysbench (downloaded, compiled, installed latest version 0.5 from Launchpad, not the one you get from apt-get, which is a previous version)
  - FILEIO test
  - 16 files with total of 256GB used
  - 4K block size, 16K operations
  - Across 1 – 2 – 4 – 8 – 16 – 32 – 64 – 128 thread count
  - Synchronous access
  - Random Read, Random Read/Write, Random Write, and Sequential Read, Sequential Write
  - Each individual combination of [Flavor + File System + Mount Option + Access Mode + Synchronous + Thread Count] was tested for 5 minutes (longer tests in future)
    - About 1000 tests
  - 95% latency reported every second
  - Total throughput reported at end of each test
Next is what each chart will consist of
Random Write [CBS SSD, CBS, Local Storage]

This chart will show the 95 percentile latencies across time at 32 threads in milliseconds, with the Y axis inverted. Also, if there were high anomalies but only in 2 or 3 points, the Y axis was shortened to show more detail, so those few points may not be visible. Those points still affected Average and Std Dev calculations.

We will show file system (in parenthesis) and access mode.

Some charts have a “zoom.”

Here we show the average of the 95 percentile latency, as well as standard deviation and semi standard deviation.

This chart will show the throughput in MB / seconds (Megabytes per second).

Throughput markers at 32 threads and maybe additional ones.

Flavors tested:
Cost of performance comparison
At Rackspace there is never a charge for I/O

- You just pay for the storage you provision
Cost of performance comparison

Let’s calculate the cost of performance (or its inverse, the performance per dollar). We would use the throughput of a specific scenario and the costs.

**RACKSPACE MONTHLY COSTS (SATA or SSD)**

Storage price = [Price of GB] * [volume size]
IO price = 0
Total price = Storage price

**AWS MONTHLY COSTS**

You can use 100 for [IOPS of the application] as an estimate using a medium size app

**EBS Standard Costs**

Storage price = [Price of Standard GB] * [volume size]
IO price = [IOPS of the application] * [Price of 1 Million IO requests] * 60 * 60 * 730 hours / 1 million
Total price = Storage Price + IO Price

**EBS Provisioned IOPS Costs**

Storage price = [Price of PIOPS GB] * [volume size]
IO price = [Price of IOPSMonth] * [provisioned IOPS]
Total Price = Storage Price + IO price

AWS has an additional extra compute fee for EBS-Optimized EC2 instances (dedicated network). We did not use that in these tests (future).
Pricing as of June 2013

**RACKSPACE**

**Standard Volumes**
- $0.15/GB/month
- $15.00/mo for 100GB

**SSD Volumes**
- $0.70/GB/month
- $70.00/mo for 100GB

**AWS**

**Amazon EBS Standard volumes**
- $0.10 per GB-month of provisioned storage
- $0.10 per 1 million I/O requests

**Amazon EBS Provisioned IOPS volumes**
- $0.125 per GB-month of provisioned storage
- $0.10 per provisioned IOPS-month

**Amazon EBS Snapshots to Amazon S3**
- $0.095 per GB-month of data stored
Comparing cost of MB/s (lower is better)

Cost of MB/s
(500 GB of storage, 100 IOPS application,
using the throughput of the ext3 file system test, Random Read/Write operations
at 32 threads)
Comparing MB/s per dollar (higher is better)

MB/s per dollar
(500 GB of storage, 100 IOPS application, using the throughput of the ext3 file system test, Random Read/Write operations at 32 threads)
Charts “Rackspace vs. Rackspace”

CBS SSD vs. CBS SATA vs. Local Storage
SSD delivers high consistency with Semi St Dev of 0.19.

SSD delivers 12x more throughput than SATA and Local Storage.
At 32 threads, SSD delivers 6.5x more throughput than SATA and Local Storage.

SSD delivers lowest latency and small variability.
SSD delivers less latency and a more consistent performance with a Semi Std Dev of only 18.77.

SSD delivers 3.5x the throughput of SATA at 32 threads.

SATA performs 61% better than local storage but only at high thread number.
Local Storage delivers more throughput at all thread counts except 128 (and presumably at higher ones).

Very low latency across the board and similarly low variability, but Local provides lowest latency with just a little more variability.
Local storage delivered triple the performance of both SATA and SSD.

Very low latency and variability across the board, but Local storage edges SATA and SSD.
SSD is 11x faster than SATA

Local storage and SATA deliver comparable throughput.

Very low latency and high consistency of SSD. SATA also with high consistency.
SSD delivers lowest latency. Local Storage delivers less latency than SATA but more variability.

SSD delivers 8x more throughput than SATA.

Very similar throughput of SATA vs. Local Storage (formatted ext3).
Random Write [CBS SSD, SATA, Local Storage]

- Highly consistent SSD and SATA latency.

<table>
<thead>
<tr>
<th>Flavor</th>
<th>LATENCY95_AVG_DISPLAY</th>
<th>LATENCY95_STDEV_DISPLAY</th>
<th>LATENCY95_SEMISTDEV_DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAX-LOCAL</td>
<td>177.97</td>
<td>17.96</td>
<td>19.32</td>
</tr>
<tr>
<td>RAX-SSD</td>
<td>17.96</td>
<td>20.91</td>
<td>245.57</td>
</tr>
<tr>
<td>RAX-SATA</td>
<td>38.98</td>
<td>3.24</td>
<td>2.26</td>
</tr>
</tbody>
</table>

SSD is almost 3x faster than SATA.

Local storage (formatted ext3) delivers the least throughput.
(ext4) Sequential Read [CBS SSD, SATA, Local Storage]

Very low latency and variability across the board, but SATA shines with minimum variability.

Local storage (formatted ext3) delivers double the throughput of SSD and SATA (formatted ext4).
(ext4) Sequential Write [CBS SSD, SATA, Local Storage]

Very low latency and low variability across the board.

SATA and SSD perform more or less the same, but Local Storage (formatted ext3) is 3x faster.
(xfs)
SSD delivered lowest latency at about 6.8ms and the lowest variability.

SSD is 15x faster than SATA, and has very low latency profile. SATA (formatted xfs) has very similar throughput as Local Storage (formatted ext3).
SSD delivered lowest latency and variability. Slightly less latency seen in Local Storage when compared to SATA.

SSD delivered 7x more throughput than SATA.

SATA (formatted xfs) with very similar throughput as Local Storage (formatted ext3).
In XFS, SSD saw a little more latency and variability than SATA. Local Storage saw a lot more of both latency and variability.

SSD is from 50% to 150% faster than SATA.

SATA outperforms Local Storage (formatted ext3) by 60%.
SSD is from 18% to 32% faster than SATA, but Local Storage (formatted ext3) delivers twice the throughput of SSD (except at 128 threads where it ties with SSD).

Very low latency across the board though SATA saw just a little more than Local Storage and SSD, as well as more variability.
Very low latency of Local Storage. SSD sees a little more latency and variability.

Very similar performance between SSD and SATA, but Local Storage (formatted ext3) delivers 2x more throughput.
Charts “Rackspace vs. AWS”

RAX CBS SATA vs. RAX Local Storage vs. AWS EBS Standard
Higher variability of EBS Standard (2-3 times that of Local and 16x that of CBS SATA).

AWS EBS Standard delivers only 50% of the throughput of RAX Cloud Block Storage and RAX Cloud Servers local storage.
AWS EBS Standard delivers 20% of the performance of RAX CBS SATA, and only 17% of the performance of RAX Local storage.

Average 95 percentile latency of AWS EBS is about 4x that of CBS SATA, with about 46x more variability.
AWS EBS Standard saw about 20x the average 95 percentile latency of CBS SATA, with about 8 times the variability.

RAX SATA delivers 22x and RAX Local Storage 14x the throughput of AWS EBS Standard.
AWS EBS Standard saw 5x the latency of CBS SATA and about 240x the variability.

AWS EBS Standard delivers only 50% of the throughput of RAX CBS SATA and 24% of the throughput of RAX Local storage.
Very low latency and variability across the board.

About 24% more throughput with CBS SATA than with AWS EBS Standard, but local storage delivers the most throughput.
(ext4)
AWS EBS Standard shows more latency (818ms at 95 percentile on average compares to 143ms) and more than 44x the variability.

RAX CBS SATA delivers more than 5x the throughput of AWS EBS Standard volumes.
AWS EBS Standard shows about 6x the average 95 percent latency of CBS SATA and about 26x the variability.

RAX CBS SATA delivers more than 5x the throughput of AWS EBS Standard volumes.
RAX CBS SATA delivers 24x and Local Storage 15x the performance of AWS EBS Standard volumes.

AWS EBS Standard’s average 95 percentile latency is over 1.3 seconds compares to 39ms of CBS SATA. Large variability in EBS.
RAX CBS SATA delivers about 2x the performance of AWS EBS Standard, but local storage (formatted ext3) delivers 5x.

100x more latency seen in AWS EBS Standard, with a lot of variability compared with the steady CBS SATA.
Very little latency across the board, though AWS EBS Standard sees more latency and variability than CBS SATA and Cloud Servers Local Storage.

RAX AWS SATA delivers 50% more throughput than AWS EBS Standard, but Local storage delivers almost 4x more.
Random Read [CBS SATA, AWS EBS Std, Local Storage]

AWS EBS Standard shows double the 95 percentile average latency and about 5x the variability.

RAX CBS SATA outperforms AWS EBS Standard by about 67%.
AWS EBS Standard shows about 6x the 95 percentile average of SATA, and a lot of variability (857 compared to 52).

RAX CBS SATA is about 6x faster than AWS EBS Standard, while Local Storage (formatted ext3) is about 8x faster.
Random Write [CBS SATA, AWS EBS Std, Local Storage]

AWS EBS Standard shows over a second of 95 Percentile Average Latency and large variability of performance.

RAX CBS SATA throughput is 30x that of AWS EBS Standard volumes, while Local Storage (formatted ext3) is about 18x faster that EBS Standard.
Very low latency across the board, though Local Storage is fastest. Still, more variability from AWS EBS Standard.

RAX SATA is about 60% faster than EBS Standard, while Local Storage (formatted ext3) can be about 300% faster.
Cloud Servers Local Storage shows lowest latency and variability, with AWS EBS Standard showing the largest latency and variability of performance.

RAX CBS SATA is about 70% faster than AWS EBS Standard, and Local Storage (formatted ext3) can be 3x faster.
Charts “Rackspace vs. AWS”

RAX CBS SSD vs. AWS EBS Provisioned IOPS
SSD shows the lowest latency average (95 Percentile) and lowest variability.

SSD delivered the most throughput, including more than AWS EBS with 4000 Provisioned IOPS, which cost $400 / month.

### Random Read [CBS SSD, AWS EBS Provisioned IOPS]

<table>
<thead>
<tr>
<th>Flavor</th>
<th>LATENCY95_AVG_DISPLAY</th>
<th>LATENCY95_STDEV_DISPLAY</th>
<th>LATENCY95_SEMISTDEV_DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAX-SSD</td>
<td>6.61</td>
<td>0.20</td>
<td>0.19</td>
</tr>
<tr>
<td>AWS-EBSP4000</td>
<td>11.59</td>
<td>4.07</td>
<td>3.80</td>
</tr>
<tr>
<td>AWS-EBSP3000</td>
<td>11.90</td>
<td>0.42</td>
<td>0.34</td>
</tr>
<tr>
<td>AWS-EBSP2000</td>
<td>16.98</td>
<td>0.32</td>
<td>0.29</td>
</tr>
<tr>
<td>AWS-EBSP1000</td>
<td>33.38</td>
<td>0.53</td>
<td>0.34</td>
</tr>
<tr>
<td>AWS-EBSP0500</td>
<td>66.27</td>
<td>1.25</td>
<td>0.57</td>
</tr>
</tbody>
</table>
CBS SSD showed more variability but very similar latency averages (95 percentile) as EBS Provisioned IOPS 3000 and 4000.

CBS SSD showed the highest throughput, 10% more than AWS EBS Provisioned IOPS 4000 up to 750% more than IOPS 500.
(ext3) Random Write [CBS SSD, AWS EBS Provisioned IOPS]

CBS SSD showed the smallest latency average (95 percentile), with a little variability.

CBS SSD delivers about 160% more throughput than 4000 Provisioned IOPS and lower latency and consistency (about 11x the throughput of 500 IOPS).
CBS SSD showed the lowest 95 percentile latency average with little variability.

CBS SSD delivered the highest throughput, about 70% higher than AWS EBS 4000 Provisioned IOPS, and about 13x more than AWS 500 EBS Provision IOPS.
SSD delivers 132% more throughput than 500 Provisioned IOPS.

Low latency and variability across the board.
CBS SSD showed the lowest 95 percentile latency average with low variability. AWS EBS 4000 provisioned IOPS showed the most variability.

CBS SSD delivered the highest throughput, showing about 6x more than AWS EBS 1000 Provisioned IOPS.
CBS SSD showed highest throughput except at 128 threads where AWS Provisioned IOPS delivered about 8% more.

Comparable latency and variability across the board.
Random Write [CBS SSD, AWS EBS Provisioned IOPS]

CBS SSD delivered about 3x more throughput than AWS EBS 4000 provisioned IOPS and about 6x the throughput of AWS EBS 1000 Provisioned IOPS.

SSD showed the lowest latency 95 percentile average but with a little more variability.
CBS SSD delivered the highest throughput, 7x that of AWS EBS with 1000 Provisioned IOPS.

SSD showed the lowest 95 percentile latency average. There was low variability across all flavors.
CBS SSD showed the highest throughput. Interestingly, AWS EBS 4000 Provisioned IOPS showed less throughput than 1000, 2000, and 3000 Provisioned IOPS.

Very low latencies and variability across the board.
Random Read [CBS SSD, AWS EBS Provisioned IOPS]

Low latencies and variability in general but CBS SSD shows the lowest 95 percentile average.

<table>
<thead>
<tr>
<th>Flavor</th>
<th>LATENCY95_AVG_DISPLAY</th>
<th>LATENCY95_STDEV_DISPLAY</th>
<th>LATENCY95_SEMISTDEV_DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-EBSP1000</td>
<td>32.17</td>
<td>0.15</td>
<td>0.14</td>
</tr>
<tr>
<td>AWS-EBSP2000</td>
<td>16.57</td>
<td>0.29</td>
<td>0.22</td>
</tr>
<tr>
<td>AWS-EBSP3000</td>
<td>11.69</td>
<td>0.34</td>
<td>0.24</td>
</tr>
<tr>
<td>AWS-EBSP4000</td>
<td>19.57</td>
<td>7.33</td>
<td>5.63</td>
</tr>
<tr>
<td>RAX-SSD</td>
<td>6.84</td>
<td>2.34</td>
<td>2.33</td>
</tr>
</tbody>
</table>

CBS SSD showed the highest throughput, about 7x that of AWS EBS 1000 Provisioned IOPS.
Random Read/Write [CBS SSD, AWS EBS Provisioned IOPS]

CBS SSD showed the highest throughput, with a tie with AWS EBS 4000 Provisioned IOPS at 128 threads.

Comparable 95 percentile average latencies and variability across the board, but AWS EBS 3000 Provisioned IOPS showed less variability.
Random Write [CBS SSD, AWS EBS Provisioned IOPS]

CBS SSD showed slightly more latency and more variability in this test.

CBS SSD delivered the highest throughput, though we did see a dip at 32 threads.
CBS SSD showed almost 7x the throughput of AWS EBS 1000 Provisioned IOPS and almost double that of 4000 Provisioned IOPS.

CBS SSD showed lower 95 percentile latency average.

Low variability across the board.
(xfs) Sequential Write [CBS SSD, AWS EBS Provisioned IOPS]

Comparably throughputs at higher thread count, but CBS SSD delivered more throughput at lower thread count.
Interestingly, AWS 4000 Provisioned IOPS was lowest at higher threads.

CBS SSD showed lower 95 percentile average latency with a little more variability.

<table>
<thead>
<tr>
<th>Flavor</th>
<th>LATENCY95_AVG_DISPLAY</th>
<th>LATENCY95_STDEV_DISPLAY</th>
<th>LATENCY95_SEMISTDEV_DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-EBS1000</td>
<td>56.43</td>
<td>19.25</td>
<td>14.08</td>
</tr>
<tr>
<td>AWS-EBS2000</td>
<td>43.26</td>
<td>13.73</td>
<td>10.30</td>
</tr>
<tr>
<td>AWS-EBS3000</td>
<td>64.22</td>
<td>12.01</td>
<td>6.48</td>
</tr>
<tr>
<td>AWS-EBS4000</td>
<td>114.43</td>
<td>29.58</td>
<td>22.96</td>
</tr>
<tr>
<td>RAX-SSD</td>
<td>52.13</td>
<td>40.91</td>
<td>39.72</td>
</tr>
</tbody>
</table>
Basics of Cloud Block Storage
THE RACKSPACE PORTFOLIO

THE RIGHT TOOL FOR EACH APPLICATION, ALL BACKED BY FANATICAL SUPPORT

BUSINESS APPLICATIONS
- Rackspace Email
- Microsoft® SharePoint and Exchange
- Databases
- Hosted Virtual Desktops
- Cloud Drive

COMPUTE
- Cloud Servers
  - POWERED BY OPENSTACK®
- Cloud Sites (PaaS)
- Managed Dedicated Servers
- Private Clouds
- Rackspace Private Cloud
- Managed Virtualization

STORAGE & DATA
- Cloud Files
  - POWERED BY OPENSTACK®
- Cloud Databases
  - POWERED BY OPENSTACK®
- Cloud Block Storage
  - POWERED BY OPENSTACK®
- Managed Storage (NAS, SAN, DAS)

PLATFORM & NETWORKING
- Monitoring
- Load Balancers
- Private Networks
- Content Delivery Network
- Backup & Replication
- Domain Name System
- Control Panel

ALL BACKED BY FANATICAL SUPPORT®
Block Storage
The most fundamental and versatile storage layer (think of it as a hard drive)
Attachable and scalable storage for Cloud Servers
Scales and lives independently of compute nodes

Grow storage over time based on evolving need without changing compute

- **t0**
  - Cloud Server A
  - Instance Storage

- **t1**
  - Cloud Server A
  - Cloud Block Storage volume 1

- **t2**
  - Cloud Server A
  - Cloud Block Storage volume 1
  - Cloud Block Storage volume 2
  - Cloud Block Storage volume 3

Detach and attach storage to other nodes

- **t0**
  - Cloud Server B
  - Instance Storage

- **t1**
  - Cloud Server B
  - Instance Storage (detached)

- **t2**
  - Cloud Server B
  - Instance Storage
  - Cloud Server C
  - Cloud Block Storage volume 1 (attached to C)
Cloud Block Storage
Two sets of use cases with common themes

I need large amounts of scalable storage, cheap
But I want the performance to be consistent
And I don’t want pricing to feel like the tax code

I need scalable storage for an important [database | content | app] that requires higher levels of performance
But I want the performance to be consistent
And I don’t want pricing to feel like the tax code

Common themes

SATA volumes
SSD volumes
Cloud Block Storage
Consistent and reliable storage performance in the cloud

Reliable I/O
for your cloud applications

Choice
of SSD (high performance) or Standard volumes

Simple pricing
with free I/O (only pay for the storage you provision)

No lock-in
powered by OpenStack

For system administrators, DBAs, IT operations, and architects building:

- Application storage
- Scalable CMS systems
- Self-managed relational databases
- Email archives
- NoSQL stores
- Rich media mobile applications
- SaaS applications
- PaaS solutions
# Cloud Block Storage

Pay only for the storage provisioned, never pay for I/O

<table>
<thead>
<tr>
<th>Pricing</th>
<th>STANDARD VOLUMES: $0.15/GB/month (£0.12/GB/month in UK)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSD VOLUMES: $0.70/GB/month (£0.56/GB/month in UK)</td>
</tr>
<tr>
<td>Snapshots</td>
<td>$0.10/GB/month in Cloud Files storage (£0.07/GB/month in UK)</td>
</tr>
<tr>
<td>I/O</td>
<td>Free of charge</td>
</tr>
</tbody>
</table>

Volumes start at 100GB  
Volumes up to 1TB (larger sets with striping)  
99.9% SLA
Cloud Block Storage
Some Details

• **Performance expectation:**
  – SATA: IOPS in the hundreds
  – SSD: IOPS in the thousands

• **Durability:**
  – Volumes employ hardware-based RAID 10 (stripe of mirrors). You are encouraged to do RAID 1 to prevent loss in the event of a single node failure. Volumes annual failure rate is standard for SATA and SSD.

• **Snapshots:**
  – Created on Cloud Files, available in-region (in the future with Multi-region support)
  – Time for snapshot operation to finish depends on size and network conditions

• **Data Protection:**
  – Drives are wiped/zeroed before returning to available pool