CS5413: HIGH PERFORMANCE SYSTEMS AND NETWORKING

SUPER CLOUD STORAGE MEASUREMENT STUDY AND OPTIMIZATION

December 23, 2014

Sneha Prasad (sh824@cornell.edu)
Lu Yang (ly77@cornell.edu)
Cornell University

Contents

1 INTRODUCTION

Over the last few years a significant number of organisations have chosen to host their data and services with a few cloud service providers. These cloud providers act in capacity of both producers and distributors of cloud services, meaning that that they control the whole ecosystem of proprietary interfaces that are not compatible across different cloud providers. A customer of one cloud service provider cannot shift vendors without incurring significant expensive downtimes, he/she is said to be 'Locked-In' by the vendor.

In this context, it becomes very important to control and regulate these cloud providers to prevent vendor lock-ins. Supercloud is a system proposed by [1] It decouples providers and distributers by providing a uniform cloud service interface/ layer of abstraction layered on top of resources obtained from several diverse infrastructure-as-a-service (IaaS) cloud resource providers. Top layer of supercloud provides a uniform interface to customers, while the bottom layer talks to different service providers. Decoupling customers from cloud providers provides customers the flexibility to migrate across providers without incurring cost of starting from scratch again and again. Decoupling and layering of OS is acheived primarily through virtualization.

In the next section we will introduce Virtualization, its types, benefits and draw out its advantages that enable Supercloud architecture.

1.1 Virtualization

Virtualization primarily allows multiple OS's to run concurrently on shared physical resources. These OS instances called as virtual machines are managed through a Virtual Machine Manager or hypervisor. Virtualization provides isolation, decentralization, security and efficient utilization of the physical resources.

1.2 Types of Virtualization

Hypervisors can provide a fully emulated version of the hardware called 'full virtualization', but it is extremely complex to get it right. Another approach is for Hypervisor to spawn a copy of the Host OS, this is called 'lightweight virtualization', this approach while simple is constrained by inflexibility. Guest OS must be the same version as that of host OS, other types of of OS are

not supported.

Drawing a middle ground between these two approaches is 'Para Virtualization' where it does not try to emulate the hardware but obtains access to it with help of a slightly altered OS. This approach is leads to a optimized 'full virtualization'.

1.3 Xen

Supercloud leverages the Xen-Blanket, a nested virtualization system that can transform any provider specific virtual machine instance into a unified, distributor-specified one.Xen can be enabled primarily because of Intel Virtualization Technology.

2 BACKGROUND

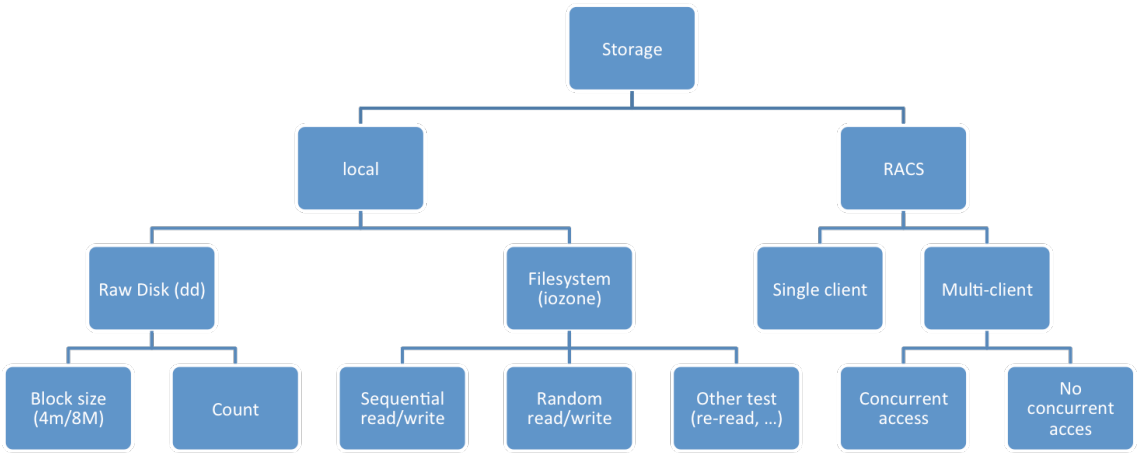
Server virtualization is inefficient without corresponding support from storage. A 2010 study by William Blair and Company, a Chicago-based investment bank, found that companies involved in server virtualization projects typically spend $2 to $3 on storage for every $1 they spend on server virtualization.
Server virtualization decouples a virtual machine (VM) from the physical hardware on which it runs, meaning it decouples the VM from the underlying storage, it is very storage resource hungry as standard images may spin up VM's with far more storage space than needed.
In a datacenter environment as we spin up more VMs, we get an increase in demand for storage capacity, but also as VMs move around a virtualized infrastructure it can make sequential accesses random. Random I/O stresses performance and capacity of storage systems.
Biggest challenges posed by server virtualization is handling the high levels of I/O that multiple VMs running on a single physical host can generate, all going through a single hypervisor running on the host.
The above stated issues get amplified in a nested virtualization environment required to setup Supercloud storage. Now I/O has to pass through 2 layers of Hypervisors running on Host and Guest.Performance is expected to take a hit due to added complexity. Performance that matches existing cloud systems will be a key factor in success of Supercloud.
Therefore, we need a performance benchmarking / measurement study to asses the impact of nested virtualization, to identify bottlenecks if any and design a solution to optimize this performance.

3 Design and Measurement Tools

3.1 Test Plan

We conduct a storage performance measurement study in a nested virtualization environment. Our test cases are carried out at three levels: from bottom to top, there are baremetal server, Guest VM, and Nested Guest VM. At each level, we measure read/write throughput for local and RACS storage. For local storage, we use dd command to measure raw disk performance and IOZone to measure filesystem performance. For RACS, we write a test script to measure PUT/GET performance for both single client and multi-client. See Figure 1 for more details.

Figure 1: Test Plan



3.2 DD command

DD stands for "Data Description"; it is usually used for copying and converting data sources. Caveat for using dd for disk benchmarking is that it only tests filesystem access to get more accurate results we used a disk benchmark is tools specifically geared towards this.

3.3 IOZone tool

Iozone is a filesystem benchmark tool that generates and measures a variety of file operations. Iozone is useful for determining a broad filesystem analysis of a computer platform. The benchmark tests file I/O performance for the following operations. Read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read/ write, pread/pwrite variants, aio_read, aio_write, and mmap.

3.4 RACS

Redundant Array of Cloud Storage is a storage service that stripes data across multiple providers to prevent vendor lock-in and data loss if one provider goes down. RACS exposes an interface with 4 generic operations LIST, DELETE, GET, PUT. We will test throughput on all of these operations with RACS through its Amazon S3 like interface.

4 ENVIRONMENT

In this section we describe our experimental setup to test local storage via DD and Iozone, and test RACS setup on Amazon S3.

4.1 Local Storage

We set up a two-node OpenStack topology to run our tests on cloudlab.us setup at University of Utah. Our baremetal server is Ubuntu 12.04 and we use CentOS 6.5 for guest VMs. On top of the Level 1 Guest VM, we installed Xen-blanket to homogenize cloud infrastructures.

4.2 RACS

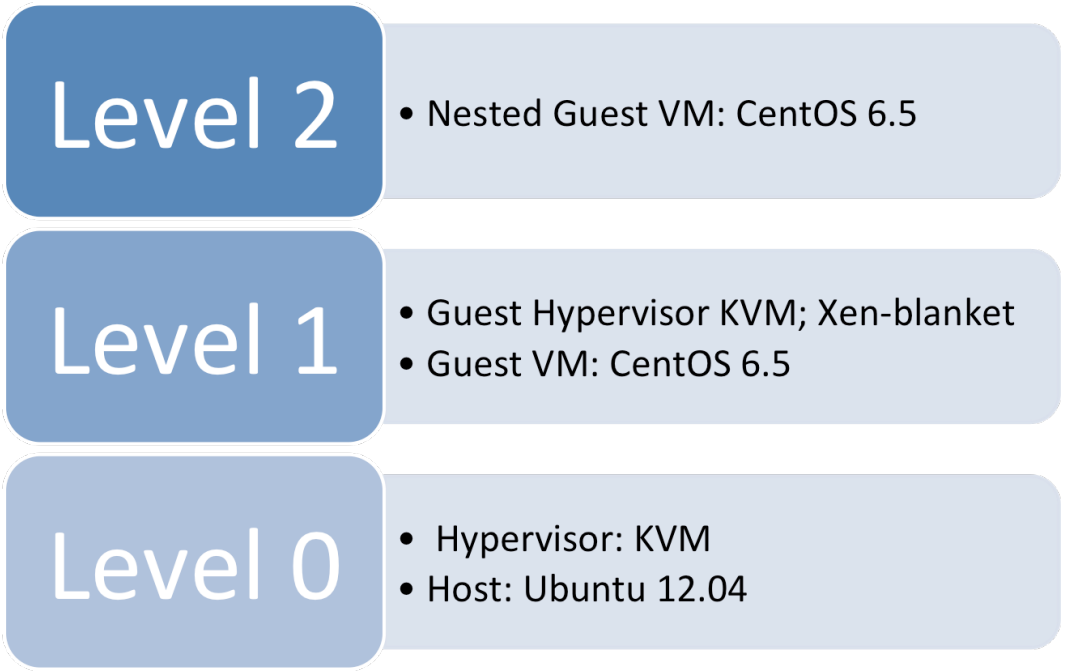
RACS was setup on local Ubuntu connected to Amazon s3 and local filesystem repositories

5 EVALUATION

5.1 Setup

We run 10 dd commands for each level and measure the average throughput for 4M and 8M. The count size for 8M is 250, for 4M is 500. A simple command to do real-world disk write test in linux:

Figure 2: Environment for Level 0, 1 and 2



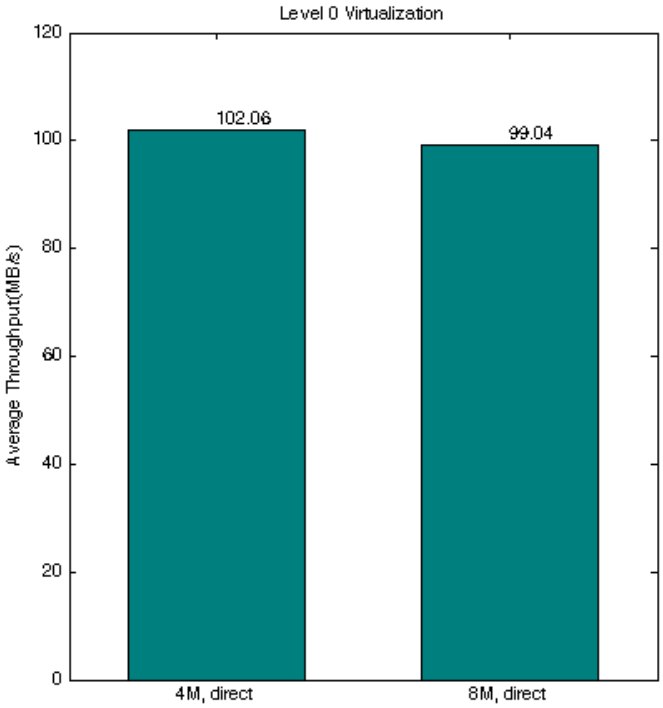

```
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=8M count=250 oflag=direct
250+0 records in
250+0 records out
2097152000 bytes (2.1 GB) copied, 20.9896 s, 99.9 MB/s
```

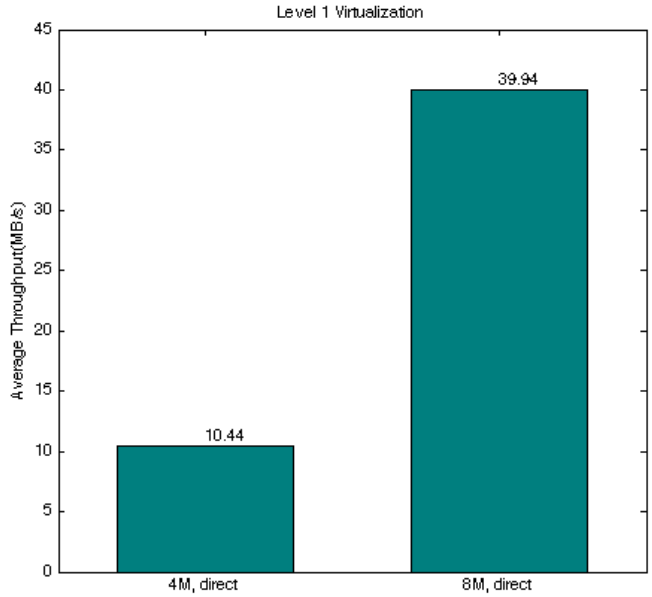
We run all 13 IOZone tests and vary the file size from 64KB to 524288KB and record length from 4KB to 16384KB. `-a` flag allows us to run all 13 tests. We add `-b` flag to write the test output in binary format to a spreadsheet, so that we can construct 3D graphs from the spreadsheet later.

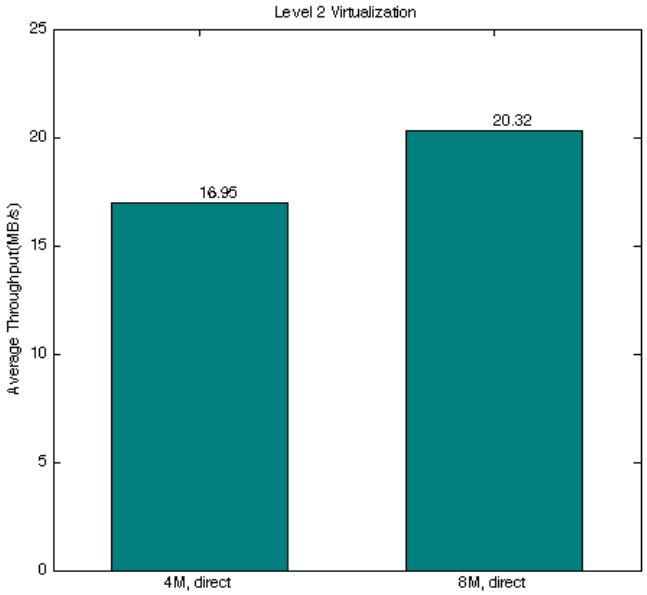
```
./iozone -a -b test.xls
```

5.2 Result

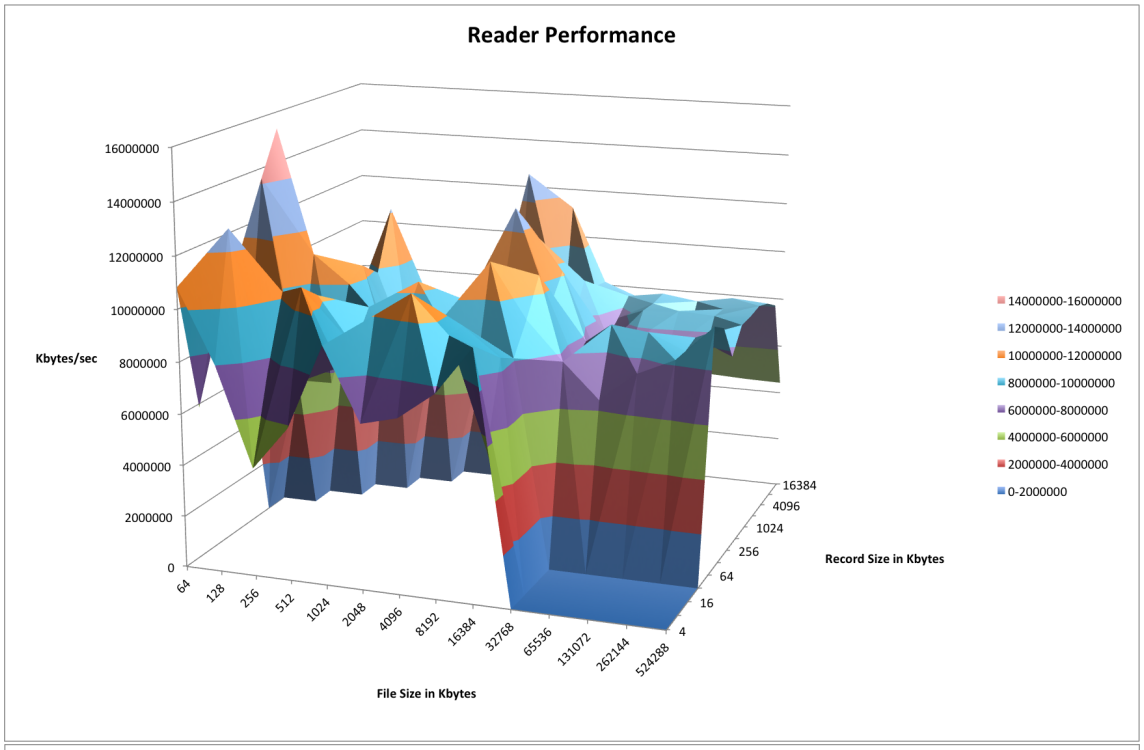
5.2.1 DD

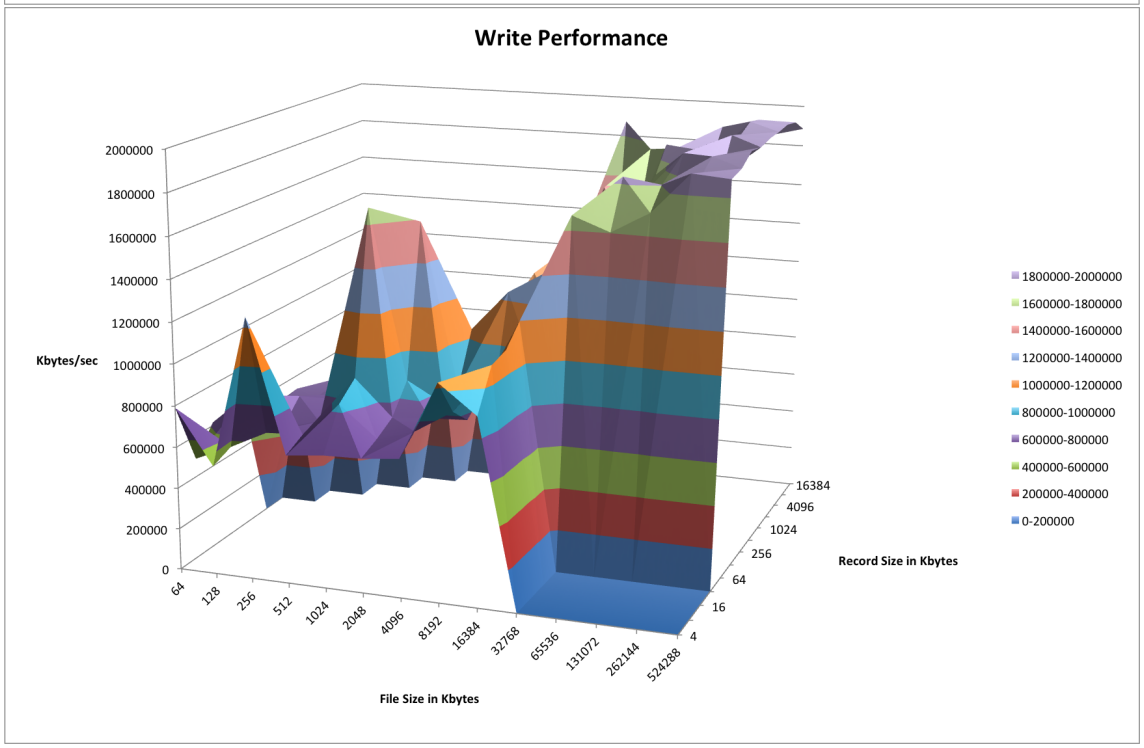


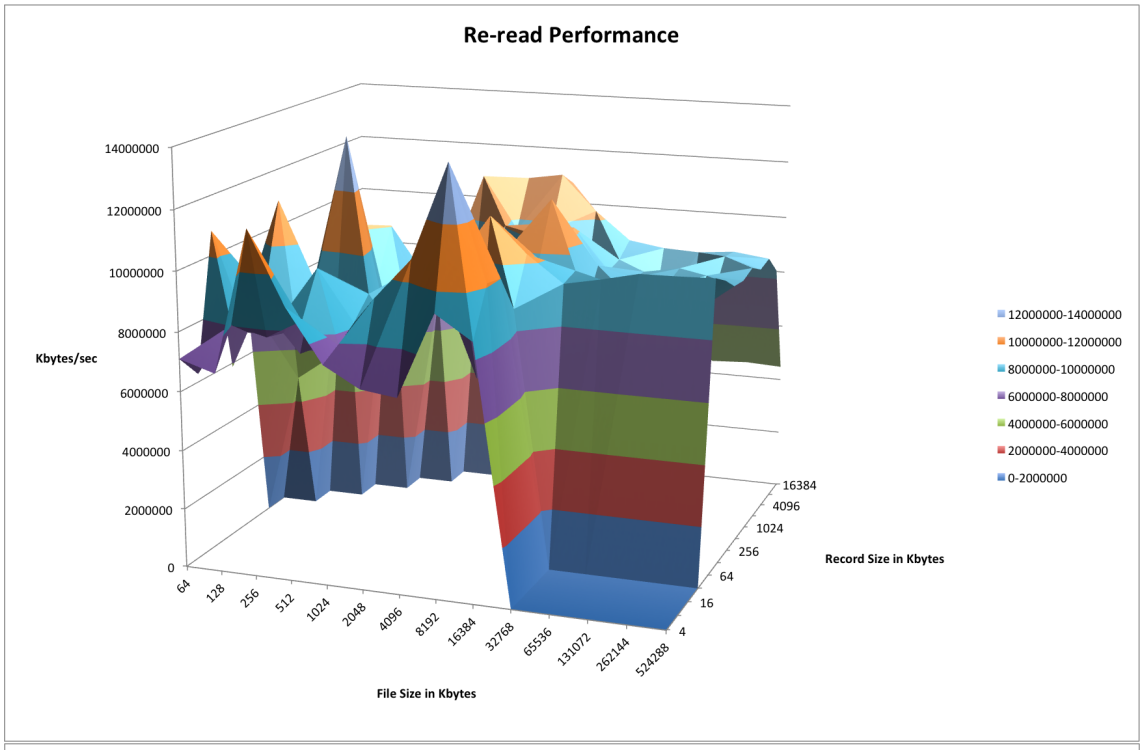


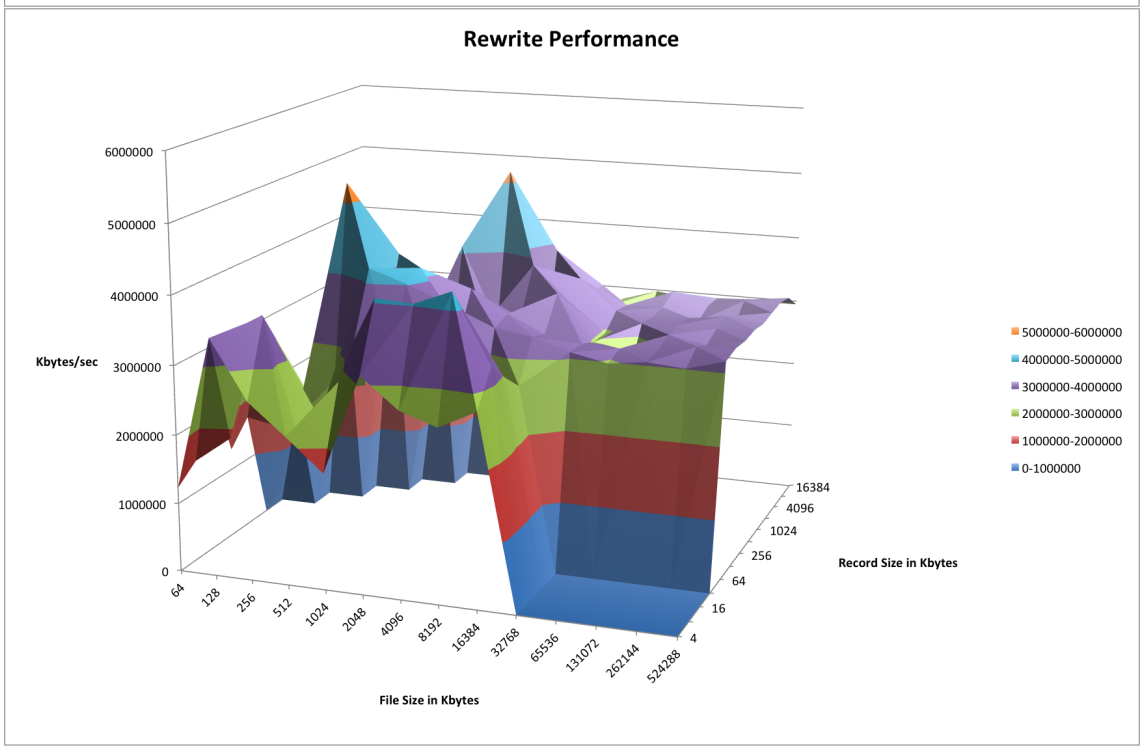

Level 2 Virtualization
Average Throughput(MB/s)
25
20
15
10
5
0
16.95
20.32
4M, direct
8M, direct

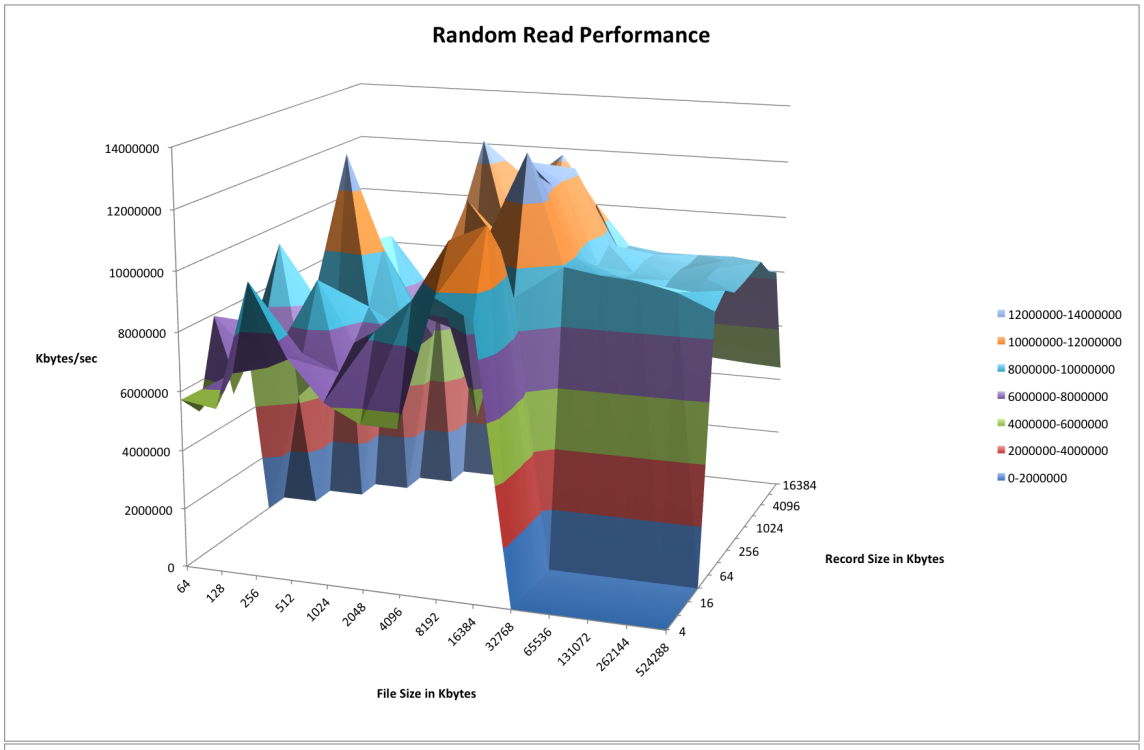
5.2.2 IOZone

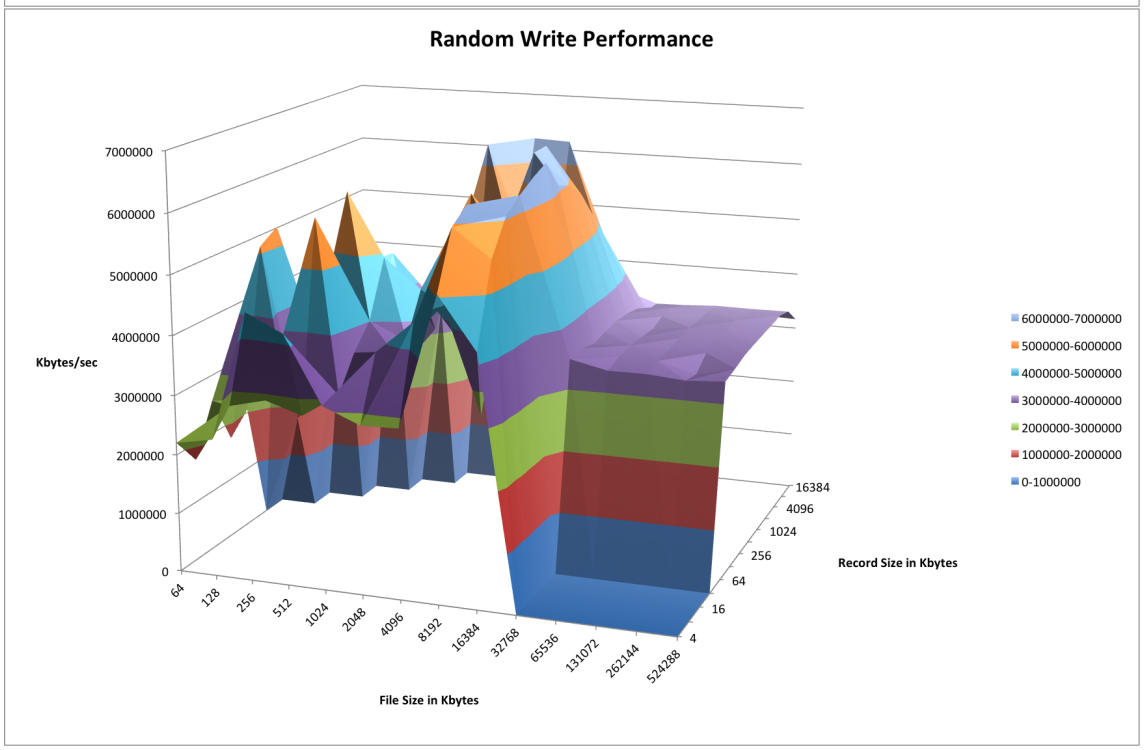

Reader Performance
Kbytes/sec
16000000
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
14000000-16000000
12000000-14000000
10000000-12000000
8000000-10000000
6000000-8000000
4000000-6000000
2000000-4000000
0-2000000

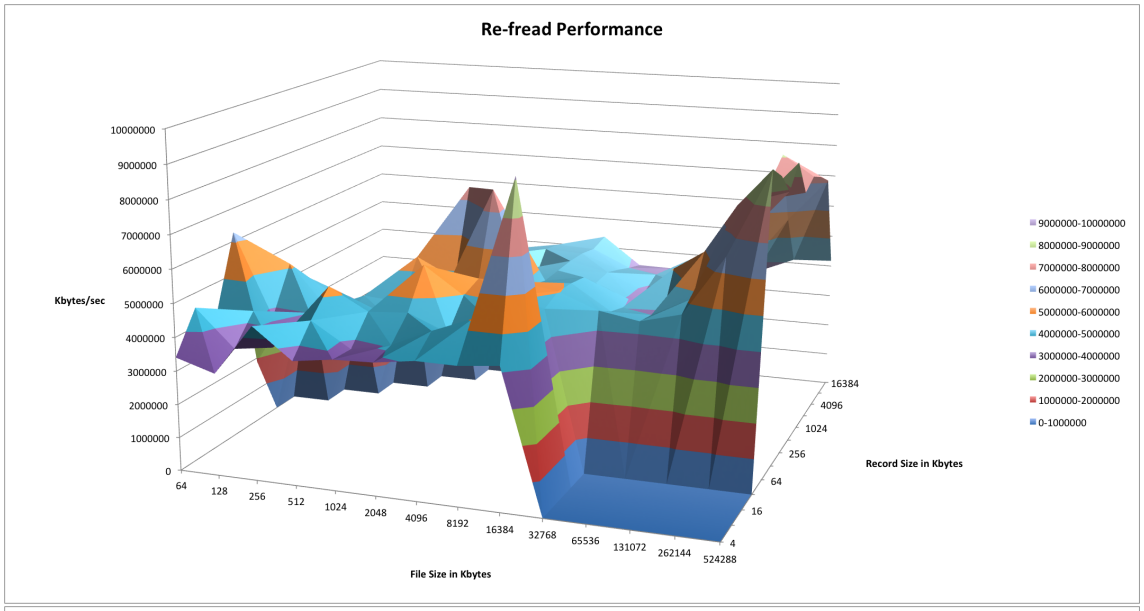

Write Performance
Kbytes/sec
2000000
1800000
1600000
1400000
1200000
1000000
800000
600000
400000
200000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
1800000-2000000
1600000-1800000
1400000-1600000
1200000-1400000
1000000-1200000
800000-1000000
600000-800000
400000-600000
200000-400000
0-200000

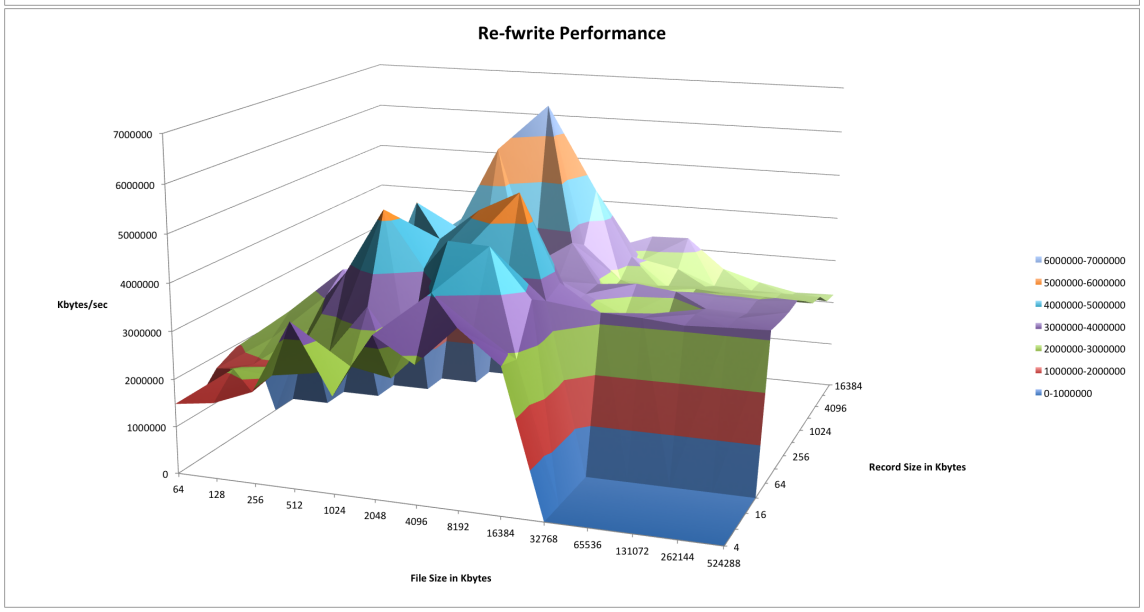

Re-read Performance
Kbytes/sec
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
12000000-14000000
10000000-12000000
8000000-10000000
6000000-8000000
4000000-6000000
2000000-4000000
0-2000000

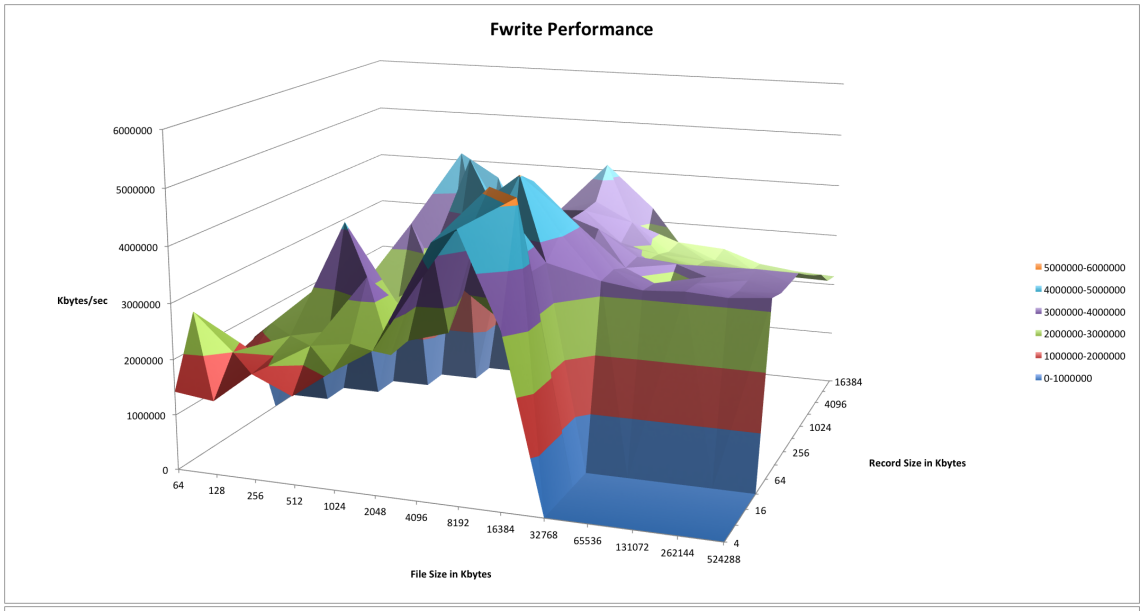

Rewrite Performance
Kbytes/sec
6000000
5000000
4000000
3000000
2000000
1000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
5000000-6000000
4000000-5000000
3000000-4000000
2000000-3000000
1000000-2000000
0-1000000

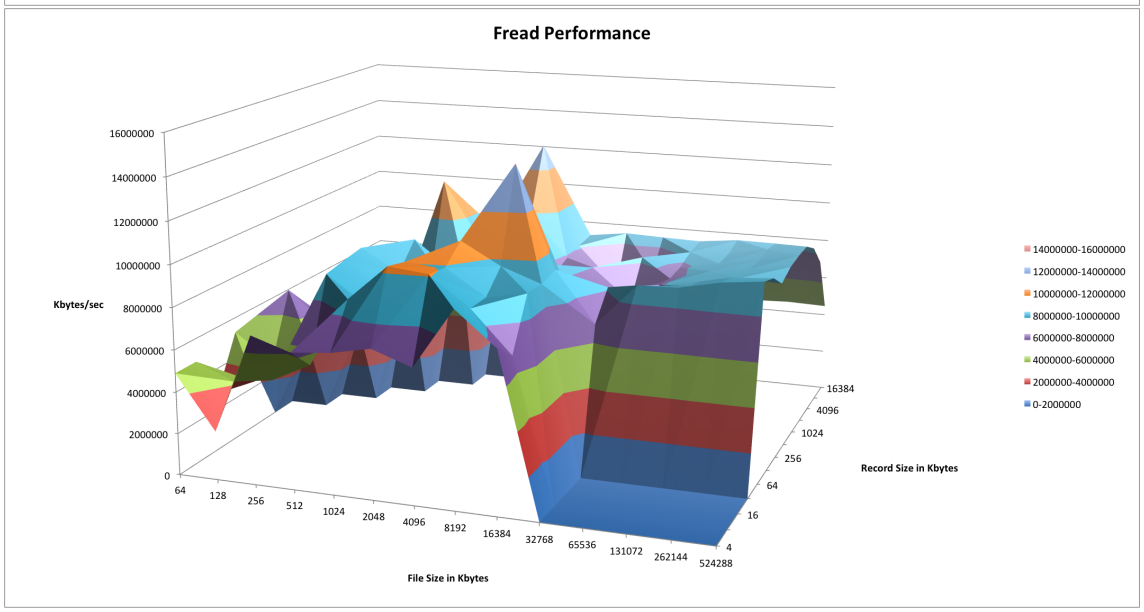

Random Read Performance
Kbytes/sec
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
12000000-14000000
10000000-12000000
8000000-10000000
6000000-8000000
4000000-6000000
2000000-4000000
0-2000000

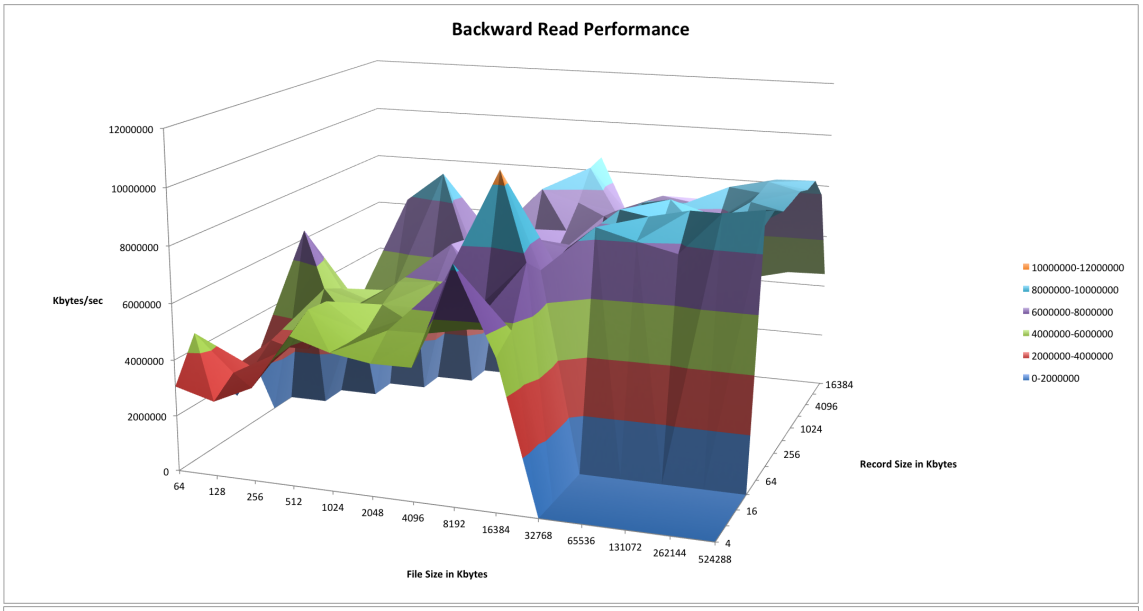

Random Write Performance
Kbytes/sec
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
6000000-7000000
5000000-6000000
4000000-5000000
3000000-4000000
2000000-3000000
1000000-2000000
0-1000000

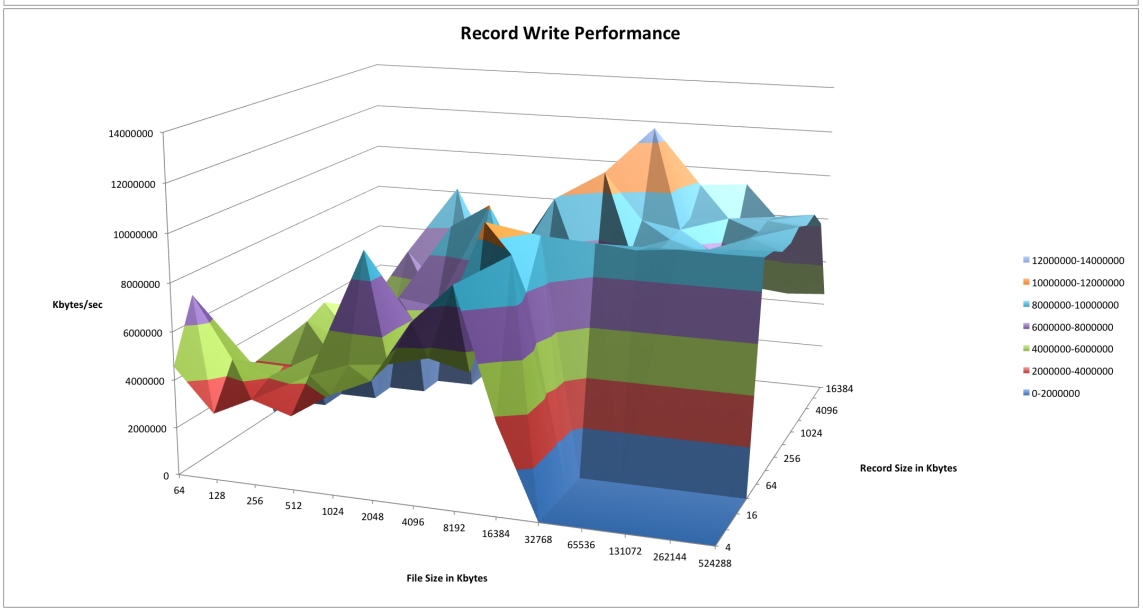

Re-fread Performance
Kbytes/sec
10000000
9000000
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
9000000-10000000
8000000-9000000
7000000-8000000
6000000-7000000
5000000-6000000
4000000-5000000
3000000-4000000
2000000-3000000
1000000-2000000
0-1000000

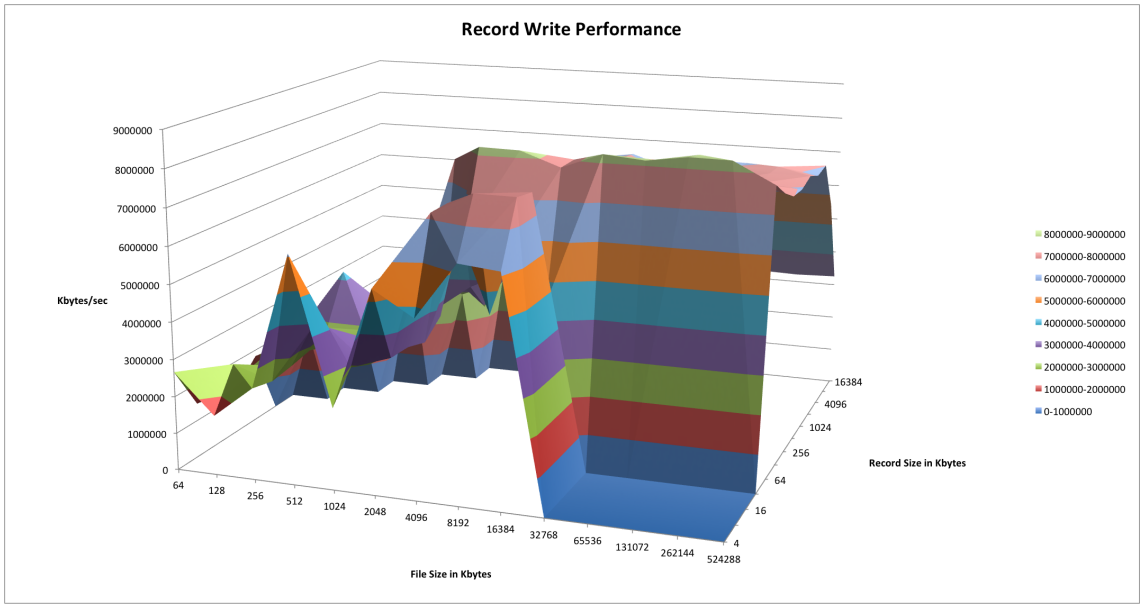

Re-fwrite Performance
Kbytes/sec
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
6000000-7000000
5000000-6000000
4000000-5000000
3000000-4000000
2000000-3000000
1000000-2000000
0-1000000


Fwrite Performance
Kbytes/sec
6000000
5000000
4000000
3000000
2000000
1000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
5000000-6000000
4000000-5000000
3000000-4000000
2000000-3000000
1000000-2000000
0-1000000


Fread Performance
Kbytes/sec
16000000
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
14000000-16000000
12000000-14000000
10000000-12000000
8000000-10000000
6000000-8000000
4000000-6000000
2000000-4000000
0-2000000


Backward Read Performance
Kbytes/sec
12000000
10000000
8000000
6000000
4000000
2000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
10000000-12000000
8000000-10000000
6000000-8000000
4000000-6000000
2000000-4000000
0-2000000


Record Write Performance
Kbytes/sec
14000000
12000000
10000000
8000000
6000000
4000000
2000000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
File Size in Kbytes
16384
4096
1024
256
64
16
4
Record Size in Kbytes
12000000-14000000
10000000-12000000
8000000-10000000
6000000-8000000
4000000-6000000
2000000-4000000
0-2000000



5.2.3 RACS

On RACS we tested following for 10 iterations.This test was constructed as part of Unit testing in RACS setup provided by Ji Yong Shin. fs exists 49 fs listdir 6 fs mkdir 1 fs open 26 fs remove 16 fs rmdir 2 fsrepo deletebucket 3 fsrepo deleteobject 10 fsrepo getallbuckets 4 fsrepo getbucketcontents 2 fsrepo getobject 3 fsrepo head 1 fsrepo putobject 9 s3repo wrapper 68 fsrepo createbucket 16

5.3 Observation and Interpretation

5.3.1 DD

- DD for block size of 8M from level 0 to level 1 throughput drops by 2.5 times, from level 1 to 2 drops by 1/2, from level 0 to level 2 drops by 1/5.
- For block size of 4M, results are not stable

5.3.2 IOZone

- The area that touches the floor of the graph(with record size from 4KB to 32KB and file size from 32768KB to 524288KB) is not measured, because it is very time consuming to test this area.

- Write, re-write, and random write performance tend to increase steadily as record size increases, while read, re-read and random read performance increase from 64KB and 1024KB, and then take a drop for record length larger than 1024KB.
- The peaks are resulted from CPU cache effect.
- The performance of write is lower than the performance of re-write due to metadata overhead.
- The performance of read is lower than the performance of re-read, because the file system maintains a cache of the data in order to improve the throughput.

5.3.3 RACS

1. GET and PUT for fs repo and s3 repo, all iterations are all the same fsrepo: GET s3 repo: GET fsrepo: PUT s3repo: PUT 262646.25 300071.4286 116746.1111 123574.1177
2. for GET and pUT, fsrepo has worse performance than s3repo. For DELETE, however, fsrepo takes less time than s3 repo
3. In addition, performance operation delete object for fsrepo is pretty stable, but for s3 repo each iteration yields different results. fsrepo: DELETE s3repo: DELETE 0.0001 0.06822 0.00009 0.07597 0.00011 0.11095 0.0001 0.07941 0.0001 0.0608 0.00009 0.07994 0.00011 0.06937 0.00012 0.08113 0.0002 0.08288 0.00011 0.06328

6 REFERENCES

1. IOZone Filesystem Benchmark. Available at:<http://www.iozone.org/docs/IOzone_msword_98.pdf> [Accessed 21 December 2014]

2. D.Williams,H.Jamjoom, and H. Weatherspoon. "Plug into the Supercloud," Internet Computing, IEEE, vol.17, no.2, 2013, pp.28- 34. Available at <http://ieeexplore.ieee.org.proxy.library.cornell.edu/stamp/stamp.jsp?tp=&arnumber=6365162&isnumber=6488666> [Accessed 22 December 2014]

3. D.Williams,H.Jamjoom, and H. Weatherspoon. "The Xen-Blanket: Virtualize Once, Run Everywhere," Proc. ACM EuroSys, ACM, 2012. pp. 113âĂŞ126. Available at <http://www.cs.cornell.edu/courses/CS5412/2014sp/papers/xen_blanket_eurosys_2012.pdf> [Accessed 22 December 2014]

7 APPENDIX 1 SYSTEM CONFIGURATION

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                32
On-line CPU(s) list:   0-31
Thread(s) per core:    2
Core(s) per socket:    8
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 62
Stepping:              4
CPU MHz:               1200.000

BogoMIPS:              5199.94
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              20480K
NUMA node0 CPU(s):     0-7,16-23
NUMA node1 CPU(s):     8-15,24-31
Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
255 heads, 63 sectors/track, 121601 cylinders, total 1953525168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x90909090
```

8 APPENDIX 2 DETAILED RESULTS (DD COMMAND)

```
###level 0 :
```

```
##for 8M I reduced count to 250
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=8M count=250 oflag=direct
250+0 records in
250+0 records out
2097152000 bytes (2.1 GB) copied, 20.9896 s, 99.9 MB/s
root@pc26:/users/weijia#
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=8M count=250 oflag=direct
250+0 records in
250+0 records out
2097152000 bytes (2.1 GB) copied, 22.2949 s, 94.1 MB/s
root@pc26:/users/weijia#
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=8M count=250 oflag=direct
250+0 records in
250+0 records out
2097152000 bytes (2.1 GB) copied, 20.3902 s, 103 MB/s
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=8M count=250 oflag=direct
250+0 records in
250+0 records out
2097152000 bytes (2.1 GB) copied, 21.2067 s, 98.9 MB/s
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=8M count=250 oflag=direct
250+0 records in
250+0 records out
2097152000 bytes (2.1 GB) copied, 21.1149 s, 99.3 MB/s
root@pc26:/users/weijia#


level 0
##### for 4M count is still 500

root@pc26:/users/weijia#
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=4M count=500 oflag=direct
500+0 records in
500+0 records out
2097152000 bytes (2.1 GB) copied, 20.0025 s, 105 MB/s
root@pc26:/users/weijia#
```

```
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=4M count=500 oflag=direct
500+0 records in
500+0 records out
2097152000 bytes (2.1 GB) copied, 21.1274 s, 99.3 MB/s
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=4M count=500 oflag=direct
500+0 records in
500+0 records out
2097152000 bytes (2.1 GB) copied, 20.5618 s, 102 MB/s
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=4M count=500 oflag=direct
500+0 records in
500+0 records out
2097152000 bytes (2.1 GB) copied, 20.9734 s, 100 MB/s
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=4M count=500 oflag=direct
500+0 records in
500+0 records out
2097152000 bytes (2.1 GB) copied, 20.1158 s, 104 MB/s
root@pc26:/users/weijia# dd if=/dev/zero of=out bs=4M count=500 oflag=direct
500+0 records in
500+0 records out
2097152000 bytes (2.1 GB) copied, 20.5397 s, 102 MB/s
root@pc26:/users/weijia#
```

9 APPENDIX 3 DETAILED RESULTS (IOZONE COMMAND)

Writer Report

	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	789967	488236	615952	492717	500060								
128	526961	566435	770474	630273	637004	653282							
256	1254656	654798	630572	733527	672010	682691	686182						
512	619084	665720	675563	736750	713970	706221	750919	741840					
1024	639150	830457	906148	1678504	778665	757382	1119120	856970	748538				
2048	644222	683541	734069	865239	1590196	795968	817870	774579	795304	803112			
4096	662782	737624	748388	766588	794268	1041153	867415	808357	833698	802354	1118786		
8192	1035249	849105	759217	845407	806305	1238375	933871	868991	1188802	902000	821005	1270850	
16384	899436	1085824	1106953	1197666	960603	1347260	854806	1409635	1047763	1255659	1592837	1014357	1874569
32768	0	0	0	0	1670121	1117328	1346268	1434872	1296706	1380577	1473658	1770007	1371210
65536	0	0	0	0	1603452	1829643	1559451	1694613	1743531	1861978	1813587	1579188	1597813
131072	0	0	0	0	1706151	1803508	1731809	1890775	1879270	1751384	1778087	1903370	1740638
262144	0	0	0	0	1898535	1859230	1915446	1794931	1957478	1933129	1967676	1950039	1839891
524288	0	0	0	0	1883937	1901144	1951551	1938919	1953114	1954738	1962229	1946154	1886050

Re-writer Report

	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	1232453	1460430	2467108	1304314	1638743								
128	3445772	1852520	3380677	3380677	1543594	1727352							
256	2588541	3667079	1842366	2081678	1924938	2015259	3654598						
512	2150051	1759070	2116152	2509312	5278892	2150051	4420457	3941742					
1024	1687075	2837198	3291652	4195100	3121783	4178773	2592318	2869422	2916180				
2048	3121562	4063729	3205431	4016229	4128177	3074635	2285891	4104506	2981782	3346546			
4096	2691164	4027344	3133979	3806915	3091124	3747948	3091124	2950952	5165626	2767462	2422154		
8192	2510611	4244664	3229054	2150735	3345053	3422010	3230268	3901055	2372450	3897515	2807212	2428626	
16384	2784441	3262443	2929368	2696064	3261050	3170916	3642541	3401987	3619137	3263218	3281292	2842019	2816048
32768	0	0	0	0	3093673	3077050	2968905	2960081	3177352	3133882	2768947	2628156	2979718
65536	0	0	0	0	2979163	3056990	2629962	3159269	3122413	3143660	3260331	3065308	2626494
131072	0	0	0	0	3048816	3085797	3172043	3191285	3157759	2981018	3087166	3082510	2919778
262144	0	0	0	0	3007615	3182518	3066329	3178157	3230002	2975321	3212074	3111055	2948884
524288	0	0	0	0	3161220	3209326	3179489	3203752	3164135	3201546	3227261	3175540	2950145

Reader Report

	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	########	5860307	8182586	5860307	6421025								
128	8036304	########	7917784	########	########	5545860							
256	4197489	7518900	########	6398720	########	5569035	7314033						
512	5886650	########	########	7022379	########	########	9859630	########					
1024	9142007	8061038	8391792	9569770	8000971	6208343	6364746	7160597	7319232				
2048	6359119	########	8029434	9189005	########	7470770	8820994	7962448	7451329	5800851			
4096	6714018	########	########	9287391	7301864	9394037	6451792	7249484	9943171	7135072	########		
8192	7758240	9547539	9558162	7556896	########	9775714	########	########	########	9547539	5681356	########	
16384	9581893	5361251	8485137	8102942	9230526	########	9143328	########	7802110	9164056	8596590	8479901	5423449
32768	0	0	0	0	7742688	7403597	7818446	6898574	7257768	7735715	8319143	7941782	4742740
65536	0	0	0	0	6384520	8954317	4971583	7578071	8498100	8447695	8532127	7771321	4659515
131072	0	0	0	0	7540321	8783752	7805172	7414023	8593183	7016325	8330036	7806946	4522661
262144	0	0	0	0	8197125	8175609	8351219	7281977	8513393	8460332	8579156	7862313	4457621
524288	0	0	0	0	9587239	9178858	7726938	8620618	8626130	8707802	8472517	8065849	4430166

Re-reader Report

	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	7100397	6271021	########	5860307	7940539								
128	6727225	8036304	7476717	8548124	########	4934216							
256	########	7735574	7571923	6554972	8815202	5810112	########						
512	9145790	8394979	9859630	7091952	########	8265729	8394979	########					
1024	7369466	7699755	8137402	8822754	8061038	6059442	6441107	7208671	7369466				
2048	6690992	9349021	8198022	7159438	9567698	7556202	8940345	7782107	7496851	########			
4096	6532752	########	9846293	9394037	7715028	9707206	6555186	7087972	########	9685316	########		
8192	9352628	########	9625100	6272536	########	9916786	9684785	########	########	9728660	########	########	
16384	8555918	########	########	8728714	########	9834199	########	9905073	########	7366374	########	9026826	5426876
32768	0	0	0	0	9380982	9547815	8039337	9191516	8730825	8870574	9309181	6430251	4992187
65536	0	0	0	0	9623631	9089335	8710298	8866798	8804036	6825228	9117372	8159324	4766988
131072	0	0	0	0	9905723	7756279	8884396	8859769	9021292	6528793	9059498	8151679	4468113
262144	0	0	0	0	9923317	8829375	8201772	9133612	8392077	9270614	9174155	8331286	4534801
524288	0	0	0	0	9960616	9610787	9148462	9057723	9224059	8900845	9014909	8327967	4477041

Random Read Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	5735102	4988978	7940539	4897948	5860307								
128	5545860	6406138	6812590	7476717	9977956	4267461							
256	9868433	6719046	7073132	6249745	8271916	5455847	9434868						
512	7513788	7309196	9468384	6736026	########	7900804	8265729	9681823					
1024	6128613	6787181	7472033	8391792	########	5885083	6364746	7257394	7257394				
2048	5535444	8261095	7616502	7698414	7911115	6900622	8857376	7726111	7470770	########			
4096	5520868	9368423	9163546	9000323	7542289	########	6512939	7286380	########	9846293	########		
8192	9845745	########	8790341	4974045	########	9614327	9728660	########	########	9492149	########	########	
16384	9219381	########	########	9092516	########	########	########	########	########	7680895	########	8291637	5481418
32768	0	0	0	0	9939317	9984083	9520039	9186601	9665998	9160883	9112293	7884378	4967827
65536	0	0	0	0	9749585	9496622	9113443	9037040	9191763	9153197	8948196	8139030	4620743
131072	0	0	0	0	9658257	9300428	8681935	8879517	8965972	8896472	8964656	8027412	4593708
262144	0	0	0	0	9397550	7476832	8824344	9027944	9096209	9123456	9000305	8241980	4506734
524288	0	0	0	0	8882186	9278939	8927549	8977517	9073719	9173268	8572427	8253265	4447414
Random Write Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	2203800	1722886	2561267	1722886	2052169								
128	2326073	2714132	2660335	5122535	5325799	1778862							
256	4496299	2872459	3197509	2325094	2392442	2441400	4332998						
512	4204110	2678309	5822804	3145087	6018636	2625909	3849877	4571003					
1024	3085895	3085895	3631168	4014717	4943530	2844715	3130886	3461434	3483896				
2048	2828608	3857514	3736694	3730203	4451191	3618634	4376355	3743207	3407614	6363830			
4096	2848213	4405314	4441761	4371684	3664018	5988509	4294102	3589754	4922872	4768469	6410863		
8192	5050818	5970616	6196743	2580178	5347148	4893290	4964702	6501608	6516405	4708228	6384425	5542066	
16384	4192306	5506895	6056817	6274716	6447807	6582447	6097662	5962227	5700587	5233056	4768276	3675073	3205079
32768	0	0	0	0	3490441	3511398	3429388	3481158	3521564	3499328	3446329	3266382	3242417
65536	0	0	0	0	3354632	3429423	3461467	3443860	3423145	3510223	3518896	3417060	3189549
131072	0	0	0	0	3358337	3414848	3424399	3483439	3512955	3490694	3502258	3454698	3189397
262144	0	0	0	0	3310108	3245074	3444641	3491723	3522497	3538037	3515031	3450218	3189451
524288	0	0	0	0	3355551	3405042	3470019	3490691	3504292	3517999	3533363	3456427	3164673
Backward Read Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	3057153	4564786	2467108	1385075	2222043								
128	2673584	3277486	3053774	3657016	3277486	7082197							
256	3316006	3756894	3935921	3465855	5347168	4998665	4070199						
512	5278892	5822804	5744919	4916336	4973263	4532414	4690818	7871843					
1024	4854136	4789183	4789183	5390231	5853003	5303701	5096034	8970167	5917516				
2048	4633675	5376082	5403134	5372719	5986827	7036283	6608629	5489457	6340344	6543188			
4096	4644707	5411313	6872481	5988509	5843879	5468151	7460407	7099689	5625724	8028705	5231694		
8192	8257953	5688881	6296676	########	7038325	7502444	6506533	6114043	6814966	7600357	8685901	8837822	
16384	5274831	5795783	7735364	7126502	7074413	7245981	6968951	5612591	7478608	7376654	7102196	5849060	4576476
32768	0	0	0	0	8421088	7091132	8436596	7262370	8086639	7928496	7786995	6166553	4590662
65536	0	0	0	0	8039056	8593746	7718296	8619887	8119796	7940820	6458778	7449103	4369952
131072	0	0	0	0	7717841	8915806	7892575	7267210	7762631	8615808	6879704	7610256	4086558
262144	0	0	0	0	9230765	6611106	7938952	8740447	8784651	8912540	8718684	7889506	4445833
524288	0	0	0	0	9449677	9131253	8799725	8851439	8926933	8707940	8769091	7937628	4491223
Record Rewrite Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	2662899	1460430	1363961	1226821	1828508								
128	1603529	2673584	1911894	2453642	2367096	2286447							
256	2463808	2325094	2413957	2533571	2880164	4197489	2486631						
512	6086873	3795443	3580298	2891043	2910635	2845081	2782414	3877684					
1024	2192645	2934110	2688041	2950233	3160839	3291652	2985091	3821805	2908281				
2048	5389574	3304067	3277593	3093459	3525079	7609754	3786104	2929912	3367537	4886737			
4096	4740836	7149920	7161842	6049663	8205092	5902100	4311344	4576653	6978564	4125010	5980170		
8192	6214676	7706040	4350542	5681356	8183217	4454894	7346838	3361415	3777121	5404339	6821731	5457560	
16384	6109046	7695517	7599353	4481274	7805655	7816309	7204956	6751231	7001613	7074413	6978151	5799696	3188275
32768	0	0	0	0	8227017	7839406	7270438	5049782	6929180	6939326	6442912	5417208	3313953
65536	0	0	0	0	8132048	7952307	7356993	7087988	6992781	7071941	6946131	5470881	3295234
131072	0	0	0	0	8290966	8158453	7310212	7056039	7044285	7031042	6932450	5278155	3247328
262144	0	0	0	0	8278158	7810761	7291103	7075600	7125998	7107434	6469359	5629750	3217676
524288	0	0	0	0	7728540	7306336	7005828	6972000	7124423	6927327	7006096	5659536	3255536

Stride Read Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	4564786	7100397	1679761	2662899	2801873								
128	2784517	4557257	3867787	2843510	3536566	3759450							
256	3557725	4054829	3935921	5428265	5810112	3654598	4070199						
512	3064306	4237291	4827914	4701087	3905895	5278892	4660280	7022379					
1024	4114719	4854136	9142007	4741598	6327241	6244448	5507740	5336651	9655828				
2048	4853603	5044577	5095447	5447681	6097316	9140117	5970183	5640860	5222461	4960102			
4096	7364466	5469892	5475122	6007355	6522830	########	6480999	7277121	6114255	6930704	6769576		
8192	8974009	5059743	########	5936574	5776876	6469778	5565406	9893942	8952964	9342456	6455193	6908138	
16384	3799635	9870927	7975978	9876601	8663787	5527714	7327884	4356280	########	6901759	8261731	8118258	4585637
32768	0	0	0	0	9330669	6022264	8167372	7785230	8854002	########	########	8268593	4614553
65536	0	0	0	0	9176114	8911643	8438877	8509939	8562426	8703403	9878879	8950818	4550144
131072	0	0	0	0	9346917	8810357	8479854	8060958	7551092	8828751	7774597	9701891	4521434
262144	0	0	0	0	9301514	9121564	8649956	8834128	8723596	8861326	8686860	7787085	4354694
524288	0	0	0	0	9315021	9173727	8763848	8737870	9080463	8773219	8908128	8148852	4480334
Fwrite Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	1421755	2662899	1357066	1279447	1599680								
128	1332828	2004703	1802755	2132084	1598754	2132084							
256	1897721	1985448	2170027	2458168	2812272	1956506	2285501						
512	1585086	2265742	2327124	4123387	2337255	2180616	1961520	3459188					
1024	2090206	2375790	3369115	2326879	2381058	2586074	2646630	2159571	4741598				
2048	2534049	2252916	2298122	2378293	2582813	3240499	4937295	2712491	4319142	3419824			
4096	3641495	4262142	2509913	3072878	4452120	4214051	2538094	3138560	2541097	2449785	3433360		
8192	4705649	4646469	5104089	2553525	5028642	4770989	2351344	3278036	3637950	3632950	3660039	4255705	
16384	3093687	5121895	3722049	3696621	4053575	4110310	3159980	3644086	3220701	3052732	3293875	2289595	3276130
32768	0	0	0	0	3260803	3291256	3143847	3179190	2952323	2885990	2764770	2788554	2357249
65536	0	0	0	0	3159123	3105550	2678996	3100996	2908459	2719683	2648128	2487879	2545170
131072	0	0	0	0	3204848	3162555	3107882	3116745	2879692	2709320	2552269	2351616	2297332
262144	0	0	0	0	3160498	3091904	3105308	3134495	2914205	2724316	2559451	2276860	2207195
524288	0	0	0	0	3227223	3127149	3119072	3120798	2914668	2725456	2561349	2261209	2169976
Fwrite Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	1421755	2662899	1357066	1279447	1599680								
128	1332828	2004703	1802755	2132084	1598754	2132084							
256	1897721	1985448	2170027	2458168	2812272	1956506	2285501						
512	1585086	2265742	2327124	4123387	2337255	2180616	1961520	3459188					
1024	2090206	2375790	3369115	2326879	2381058	2586074	2646630	2159571	4741598				
2048	2534049	2252916	2298122	2378293	2582813	3240499	4937295	2712491	4319142	3419824			
4096	3641495	4262142	2509913	3072878	4452120	4214051	2538094	3138560	2541097	2449785	3433360		
8192	4705649	4646469	5104089	2553525	5028642	4770989	2351344	3278036	3637950	3632950	3660039	4255705	
16384	3093687	5121895	3722049	3696621	4053575	4110310	3159980	3644086	3220701	3052732	3293875	2289595	3276130
32768	0	0	0	0	3260803	3291256	3143847	3179190	2952323	2885990	2764770	2788554	2357249
65536	0	0	0	0	3159123	3105550	2678996	3100996	2908459	2719683	2648128	2487879	2545170
131072	0	0	0	0	3204848	3162555	3107882	3116745	2879692	2709320	2552269	2351616	2297332
262144	0	0	0	0	3160498	3091904	3105308	3134495	2914205	2724316	2559451	2276860	2207195
524288	0	0	0	0	3227223	3127149	3119072	3120798	2914668	2725456	2561349	2261209	2169976
Re-fwrite Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	1484662	1255511	1734015	1947927	1599680								
128	1598754	2066432	1911894	2248149	2621367	2166499							
256	1924938	2015259	1970871	2533571	3043436	3410808	2285501						
512	3459188	2150051	2370799	3822466	2416145	2651850	3346002	3481620					
1024	2023250	2386350	3401130	5219904	2409105	5018624	3532609	3709575	4126579				
2048	3357009	2233004	2298737	4056054	4481379	3292669	4186524	3812993	5769681	4864597			
4096	2846797	4055868	4893426	4491697	5157871	4076076	3374022	3263773	3235498	6659365	4003879		
8192	3663551	3450533	4852518	3698649	5583494	3842165	3330140	3226628	3773388	3534295	4618365	3044239	
16384	3006784	2884972	3859602	3610390	3778534	3123782	3107115	3118961	2715239	2796907	3139336	2846374	3229631
32768	0	0	0	0	3227114	2699602	2702044	3201257	2831877	2734787	2486781	2709929	3257556
65536	0	0	0	0	3193476	3064727	2558317	3101241	2641309	2735217	2578814	2382766	2582303
131072	0	0	0	0	3120460	3133657	3024128	3068367	2918476	2715838	2582240	2339896	2333242
262144	0	0	0	0	3159726	3125940	3006135	3049313	2927379	2630197	2574861	2298782	2161623
524288	0	0	0	0	3190477	3136838	3112754	3141687	2904401	2641317	2570466	2250414	2188253

Fread Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	4897948	4897948	2772930	5283570	2801873								
128	2326073	4135958	3560017	4012317	7082197	3867787							
256	7073132	4404088	4734192	5117791	5569035	4734192	4264168						
512	6472112	5331314	9145790	9344779	9510316	9145790	4660280	6414119					
1024	6650556	8610501	7869040	8122014	8457895	9655828	7066349	########	8326715				
2048	7036283	########	########	9268323	8678404	6845629	7189398	8866519	9140117	8643474			
4096	6379914	########	9937420	########	########	8734909	9801354	6944712	9891647	########	7877762		
8192	9194949	8723390	8732257	9717654	########	9032989	8480105	7530397	7891886	8077411	8224350	6809564	
16384	7638208	6772523	9097331	9045838	9266623	7717122	7817198	8004779	7252098	5704846	8471538	6628160	5099092
32768	0	0	0	0	6839526	8125364	7139016	6458049	8231451	6980563	8402040	7068157	4727080
65536	0	0	0	0	8855087	6567415	7937152	7785188	7894751	8218606	8355000	7543962	4564121
131072	0	0	0	0	8924925	8508333	8271133	8309388	8237300	8291592	8565602	7538046	4455367
262144	0	0	0	0	9327316	9092448	8611543	8573537	8566189	8668779	8545285	7907094	4527444
524288	0	0	0	0	9538749	9228124	8510619	8846952	8955143	8969827	8447553	7297607	4472342
Re-fread Report													
	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	3406295	4564786	4018152	6271021	1879725								
128	3053774	3445772	4267461	3759450	5122535	4407601							
256	4572895	3605511	3717869	4009406	3810220	3705040	3410808						
512	3678421	4532414	5227492	4131319	4340054	3905895	3877684	4447925					
1024	3849207	3808251	4589593	4178773	4249053	5363307	3778101	3655895	3414650				
2048	3892474	4046500	4674016	5785224	4394265	4783331	7368238	4601407	4686767	4613765			
4096	4063542	4581535	4184286	4982841	5192166	7615846	4526013	4581535	3860813	4703198	4576653		
8192	4053382	5053790	4710810	5101058	5207752	4802330	4643957	4670469	4692154	4654021	4528283	4670469	
16384	4178541	9106976	4765631	4814713	5025629	4966426	4851766	4378485	4423296	4221407	3996060	3801527	2944177
32768	0	0	0	0	4558081	4217237	4024000	3896787	3909312	4162066	3817875	3782986	2727948
65536	0	0	0	0	4355964	4222407	3822005	5042724	4246215	4253376	4668537	3648605	2933977
131072	0	0	0	0	4671306	4835704	5943447	4789572	4704646	4532244	4950979	4139203	3595059
262144	0	0	0	0	6461223	7683046	6547213	7193459	7096470	7689118	8052799	5713446	4110372
524288	0	0	0	0	9016906	8603350	7913520	8546407	7408538	7668707	7422918	7085833	4175692

10 APPENDIX 4 HOW TO SETUP STACK FOR TESTING SUPERCLOUD STORAGE

1. Obtain account on cloudlab, get it linked to an existing account such as the super
2. Choose Open Stack profile and start experiment.
3. Login to the servers through browser interface, setup ssh keys on the server
cp your_public_key to ~/.ssh/authorized_keys
4. SSH into servers - compute and controller nodes.
5. Install VNC on the servers.
6. Login through VNC
7. Install KVM.
8. Verify installation and check if libvirtd daemon is running.
8. Launch a virtual machine, using weijia's script or through VNC.
9. login to guest VM, check network connectivity. if network is not up, start DHCP cl
10. Install KVM on guest VM
11. Install Xen-Blanket on guest VM, manually or through script provided by Zhiming S
12. launch VM on the guest VM, check if it can connect to internet.

13. Supercloud stack ready!