UNIVERSITY OF NOTRE DAME

OpenAFS
A HPC filesystem?

Rich Sudlow
Center for Research Computing
University of Notre Dame
rich@nd.edu
http://www.nd.edu/~rich

Notre Dame's Union Station Facility

- **Located approximately 3 ½ miles from campus in downtown South Bend, Indiana**
- **Site used due to inadequate space, cooling, UPS, etc necessary for compute intensive hardware.**
- **Facility leased based on footprint - ~ 1,000 square feet and a power utilization fee with uplift for AC, UPS, etc - approximately 2 x standard power costs**


TO SOUTH STREET
TELEGRAPH

University of Notre Dame – Center for Research Computing http://crc.nd.edu

- # 342 on Top 500 - 2.796 TFlops – June 2006
- # 480 on Top 500 – November 2006
- 2796/4618 = 60 % efficiency using mixed Xeon & Opteron architectures – network comprised of Extreme Networks Summit 400 Gb switches in a stacked configuration.
- Member of the North West Indiana Computational Grid with Purdue Calumet & West Lafayette. http://nwicgrid.org – Open Science Grid software stack


Sun
microsystems

Filesystems evaluated this past year

- IBRIX – used at Purdue – talked to vendor numerous times – on-site presentation – At Purdue IBRIX/NFS translators used to keep from having to purchase IBRIX clients.
- Lustre – evaluation with Data Direct hardware using 4 Sun x4100 Object Store Targets (OST) – Great for large file transfers – limited client support. A number of issues being addressed or soon – quotas, module load kernel.
- HP's Scalable FS – HP's implementation of Lustre – Lagging features – tied to a single vendor for hardware and software.
- NetApp GX Beta – Looks and feels a LOT like AFS but with NFS. Purchased a 4 server 3050 GX using SATA drives 9/2006. Currently used for distributed scratch space in CRC. Security an issue for clients outside CRC.
- Some of these filesystems are clear choices at "BIG" sites but what's the right fit at Notre Dame, with a small but growing research community and limited staff. Typical research being done on fast desktops and also "super" computers. What filesystem works best for that?

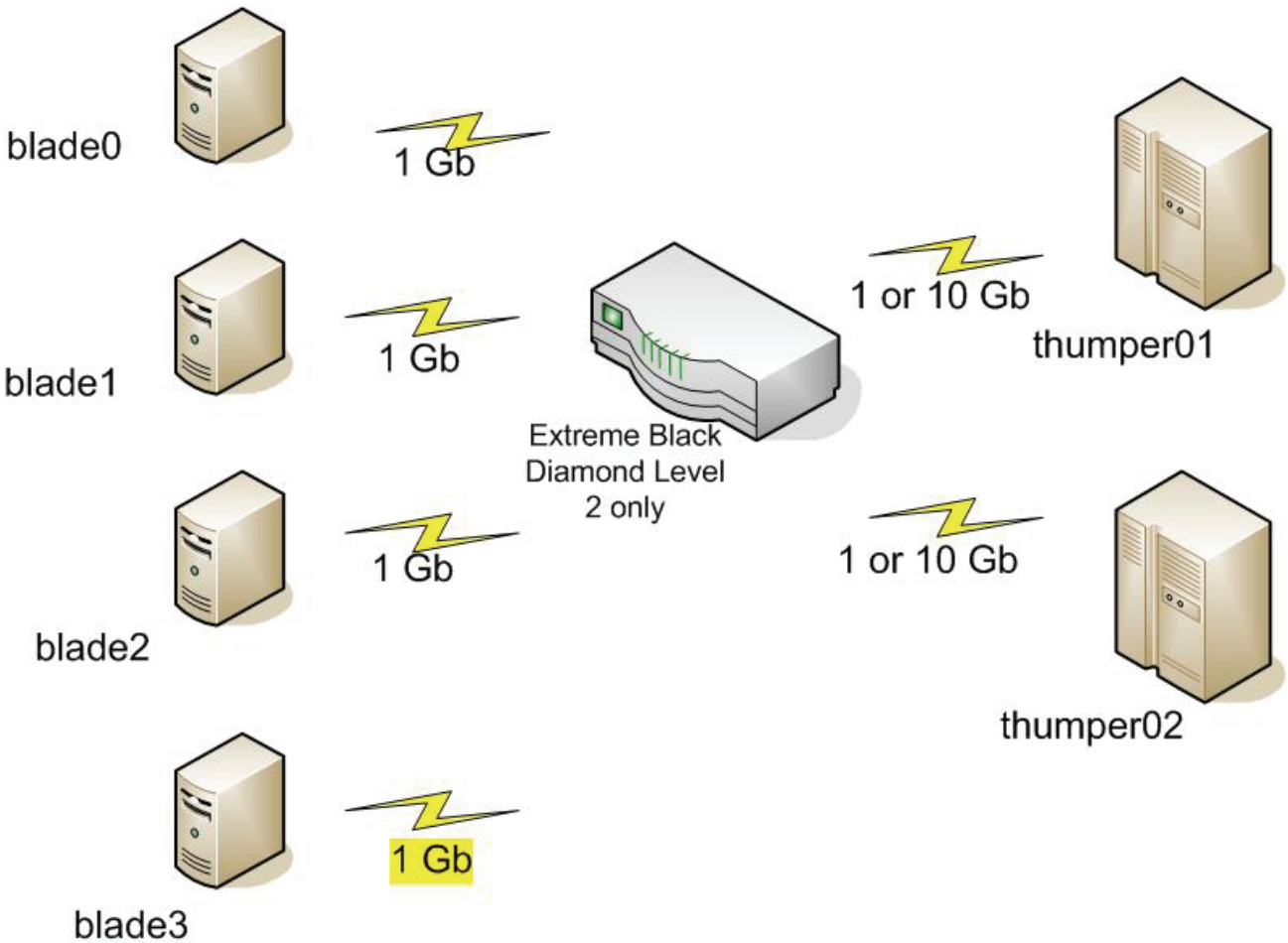
Net App GX

- 4 server NetApp GX 3050 servers 8 Gb interfaces
- 56 TB raw -14 TB per server.
- ~33 TB avail after formatting, WAFL, etc
- Currently used for distributed scratch space /dscratch on CRC resources.

Why not OpenAFS?

- We've been using AFS since 1990 – users familiar with it – prefer it over "new" NFS home directories. Used to distribute applications to Unix clients – which would continue to be necessary anyway. Currently use AFS for Sun Grid Engine (SGE) and for CRC users home space.
- Problems – not the best performance – server outages bring everything down – token lifetime – too secure? – less than liberal in giving out usable quotas to research users.
- We have a lot of nice hardware for a test cell – lets test the performance.

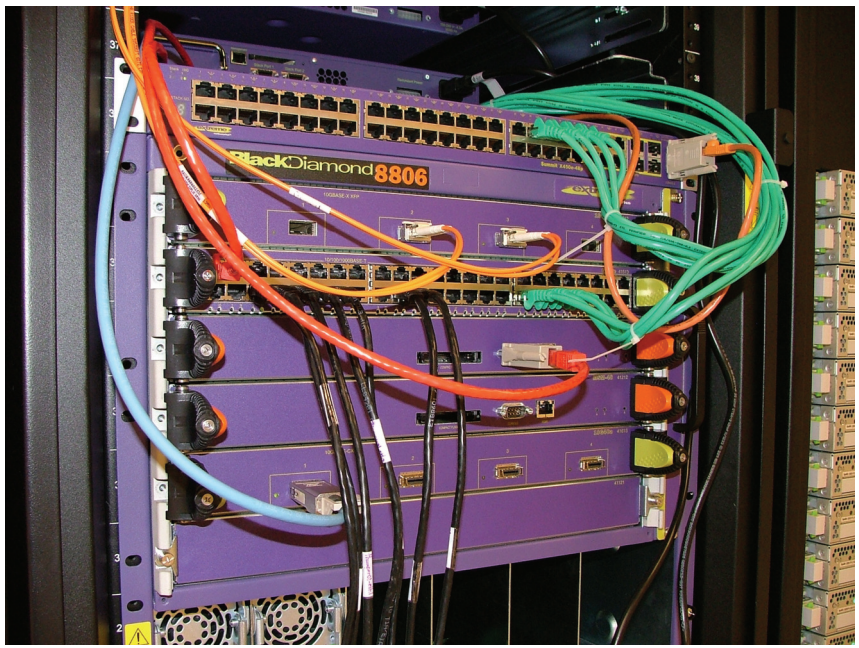
OpenAFS test environment



AFS clients

- Beta tester for Sun Blade 8000P
- 4 socket 8 core 2.6 GHz Opteron
- 2 x 73GB SAS disk
- 32 GB RAM
- Gb network
- Red Hat 4 U4 kernel 2.6.9-42.0.10.ELsmp

Test Network



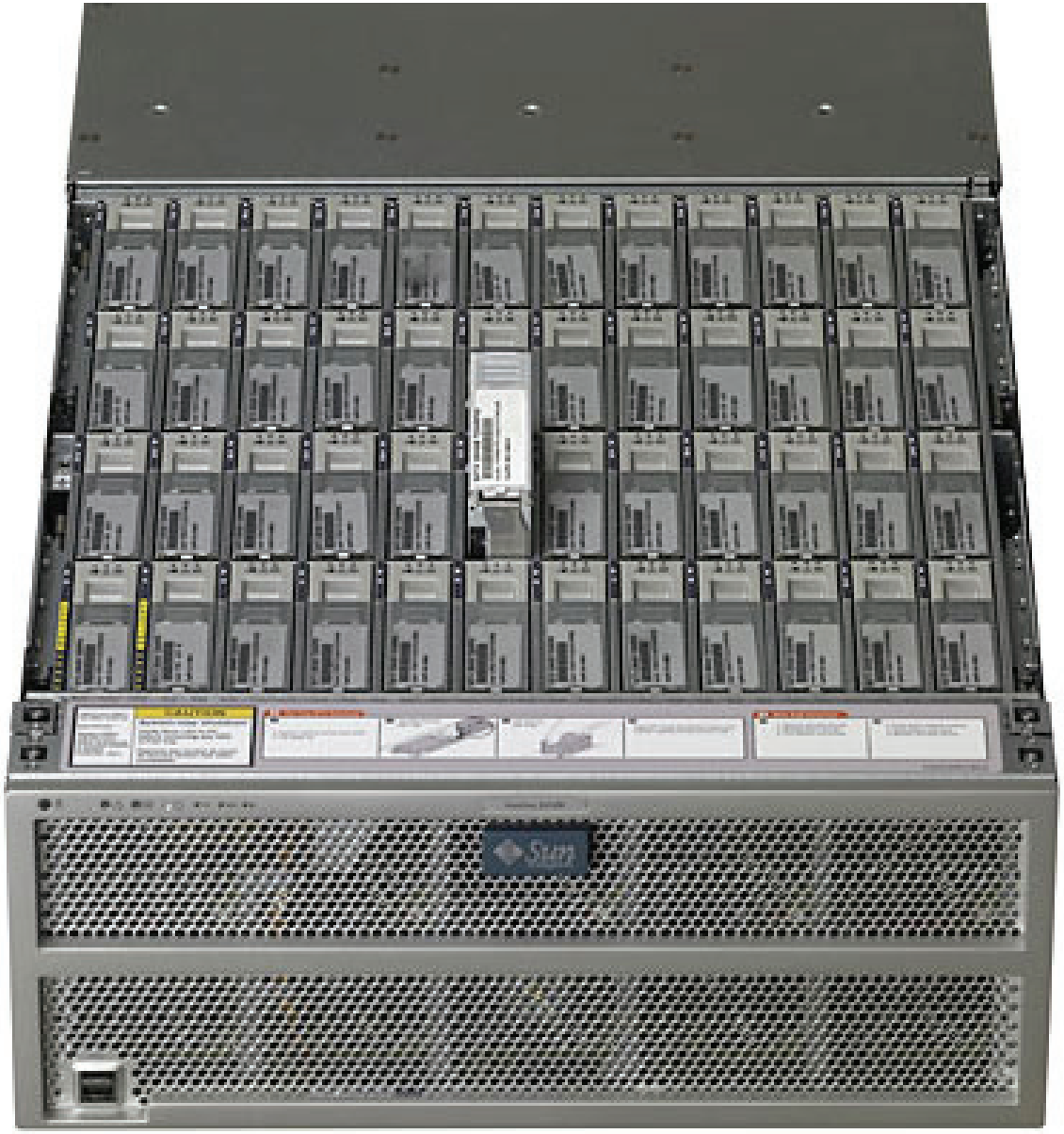
- Beta testing Extreme Networks Black Diamond 8806 with Cu 10 Gb interfaces.
- Summit 450 with Cu 10 Gb uplink
- All ethernet switches are not the same. Do side by side testing, especially 48 port switches for clusters.

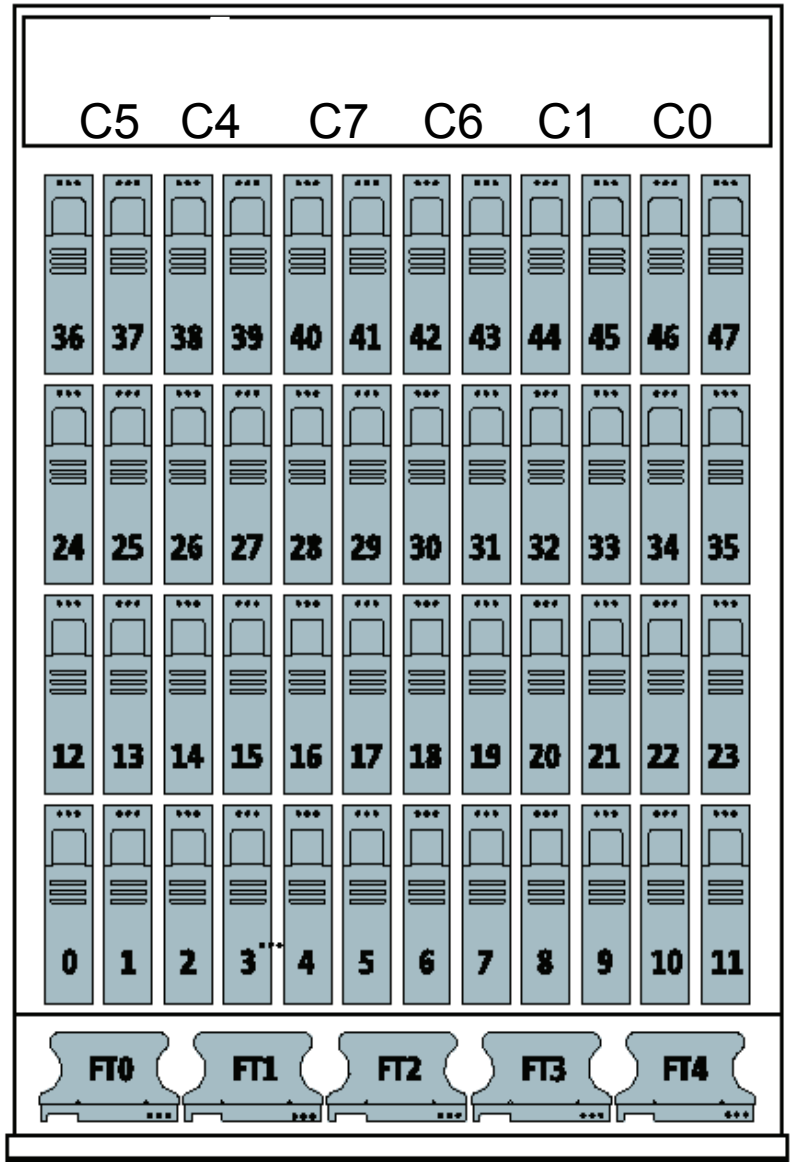
Sun's x4500 "Thumper" AFS server

- Inexpensive disk space – Could be used for NFS fileserver / OpenAFS servers / Lustre. ZFS – 10 Gb network interfaces available. Runs Solaris 10 or Red Hat 4U4. Purchased first one 11/2006 another in 1/2007.
- One Thumper configured to use Solaris 10 – Kernel 125101-04. The other configured with Red Hat 4 U4 kernel 2.6.9-42.0.10.ELsmp along with Blade clients – problems with 10 Gb card running under Red Hat – so switched back to Gb for tests – Red Hat tests not reported here – see web page for results. Later configured Red Hat Thumper as a 1 & 10 Gb Solaris client.

Testing done

- Frequently use diskrate for quick tests – decided to use iozone as it was more complete and produces nice graphs.
- Solaris local UFS filesystem using Solaris Volume Manager
- Red Hat local ext3 filesystem – Raid 10 (mdadm) used on mirrors
- Servers and clients running OpenAFS 1.4.4 --enable-large-fileserver --enable-namei-fileserver --enablesuper-groups --enable-fast-restart, --enable-debug-kernel –enable-debug –enable-debug-lwp
- Single RH OpenAFS client using diskcache and memory cache – write and read testing – 5 GB cache – minimal tuning -chunksize 19 –fakestat Note: Using client tuning from Mike Garrison's talk last year resulted in worse numbers but that may have been due to overall cachesize being smaller - 3 GB rather than 5.
- Single Solaris OpenAFS client using memory cache – 5 GB cache – Tested using 1 Gb (to compare with RH) & 10 Gb (for fastest single client performance).
- Multiple RH clients using memory cache – write and read.
- Extensive test results can be found on web page at http://www.nd.edu/~rich/afsbpw2007


CAUTION
Sun



vicepb – single disk

vicepc – 2 disk stripe

vicepd – 3 disk stripe

vicepe – 4 disk stripe

vicepf – 6 disk stripe

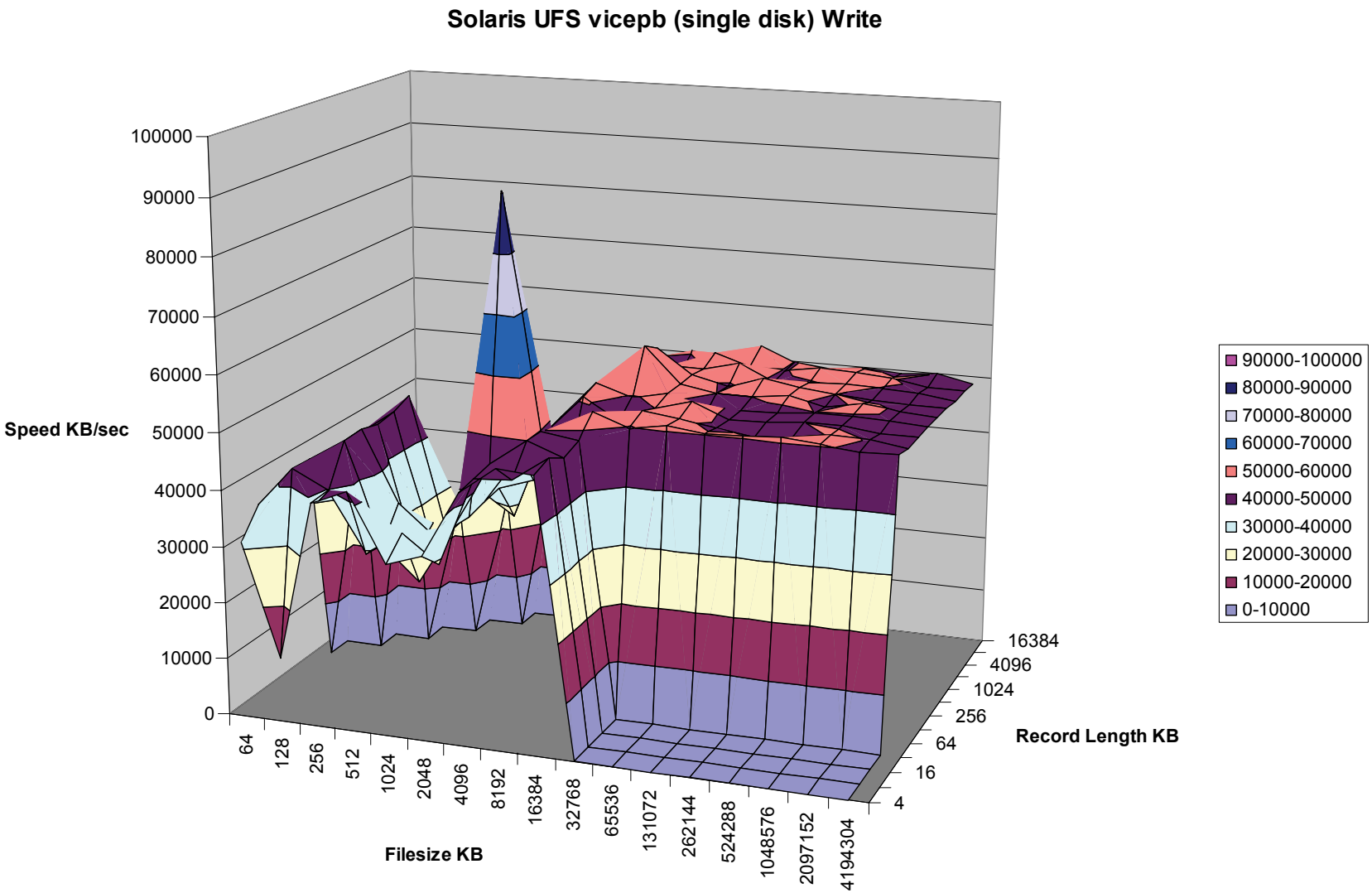
vicepg – single disk mirror

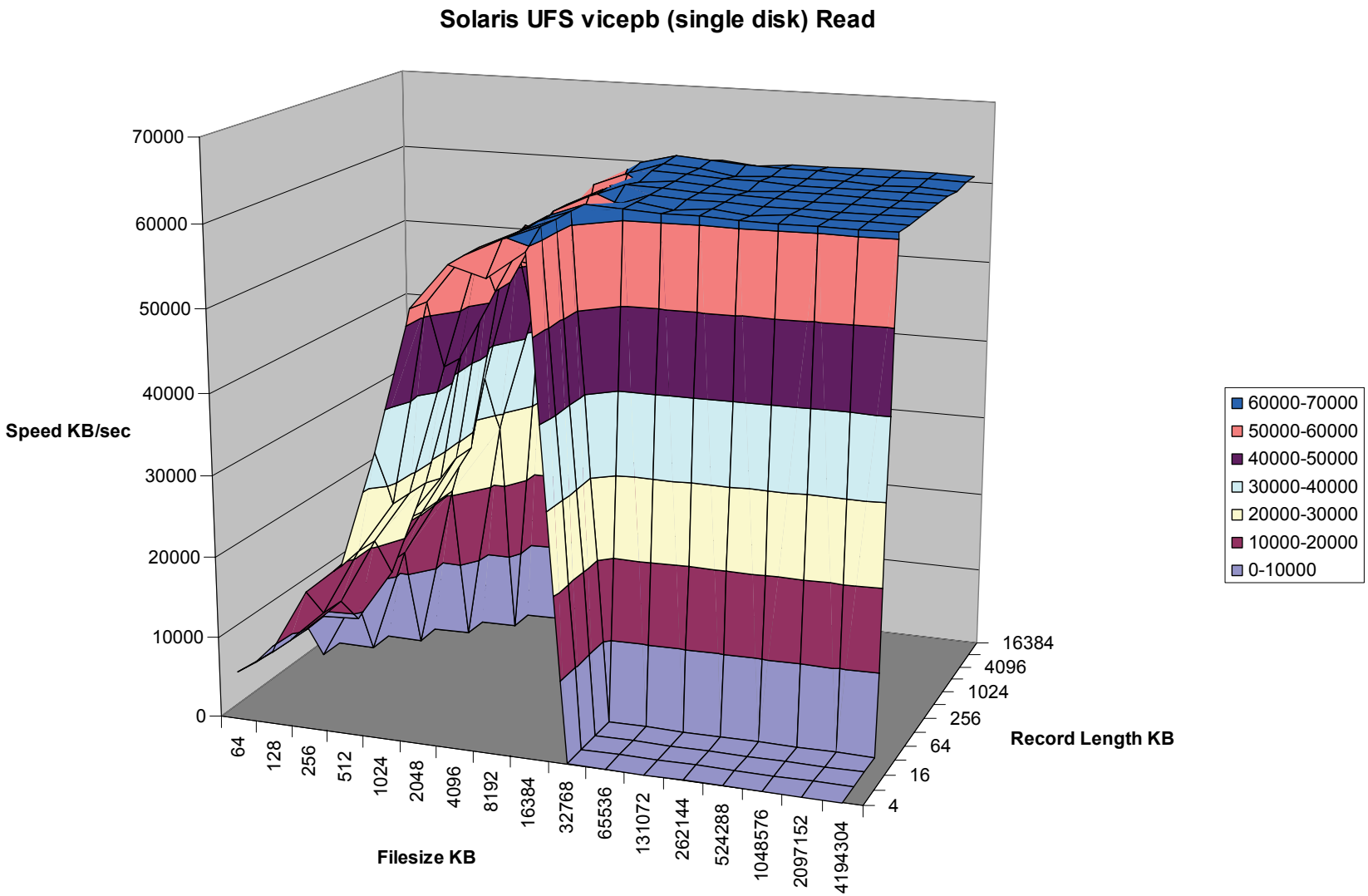
viceph – 2 disk stripe mirror

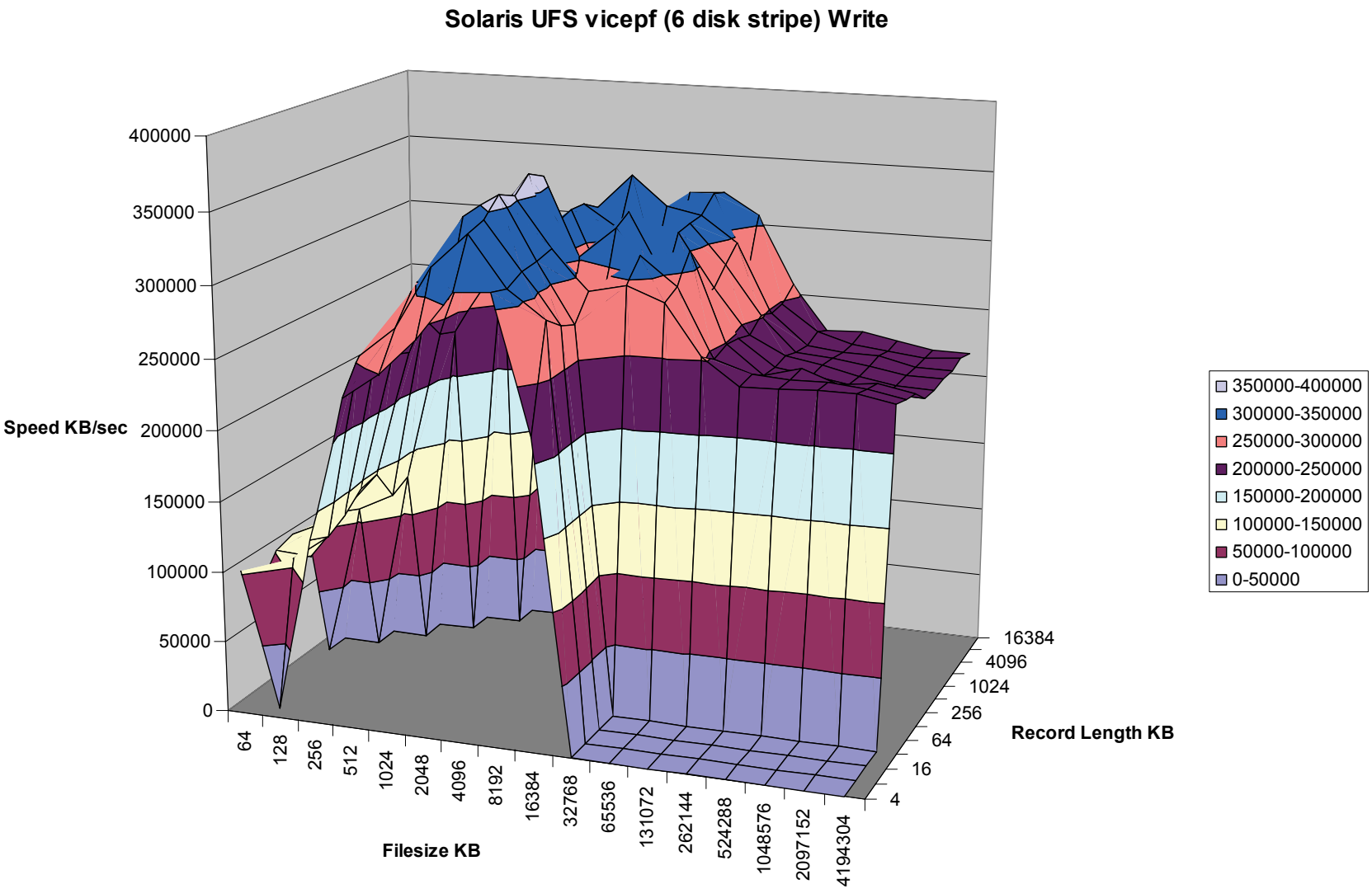
vicepi – 3 disk stripe mirror

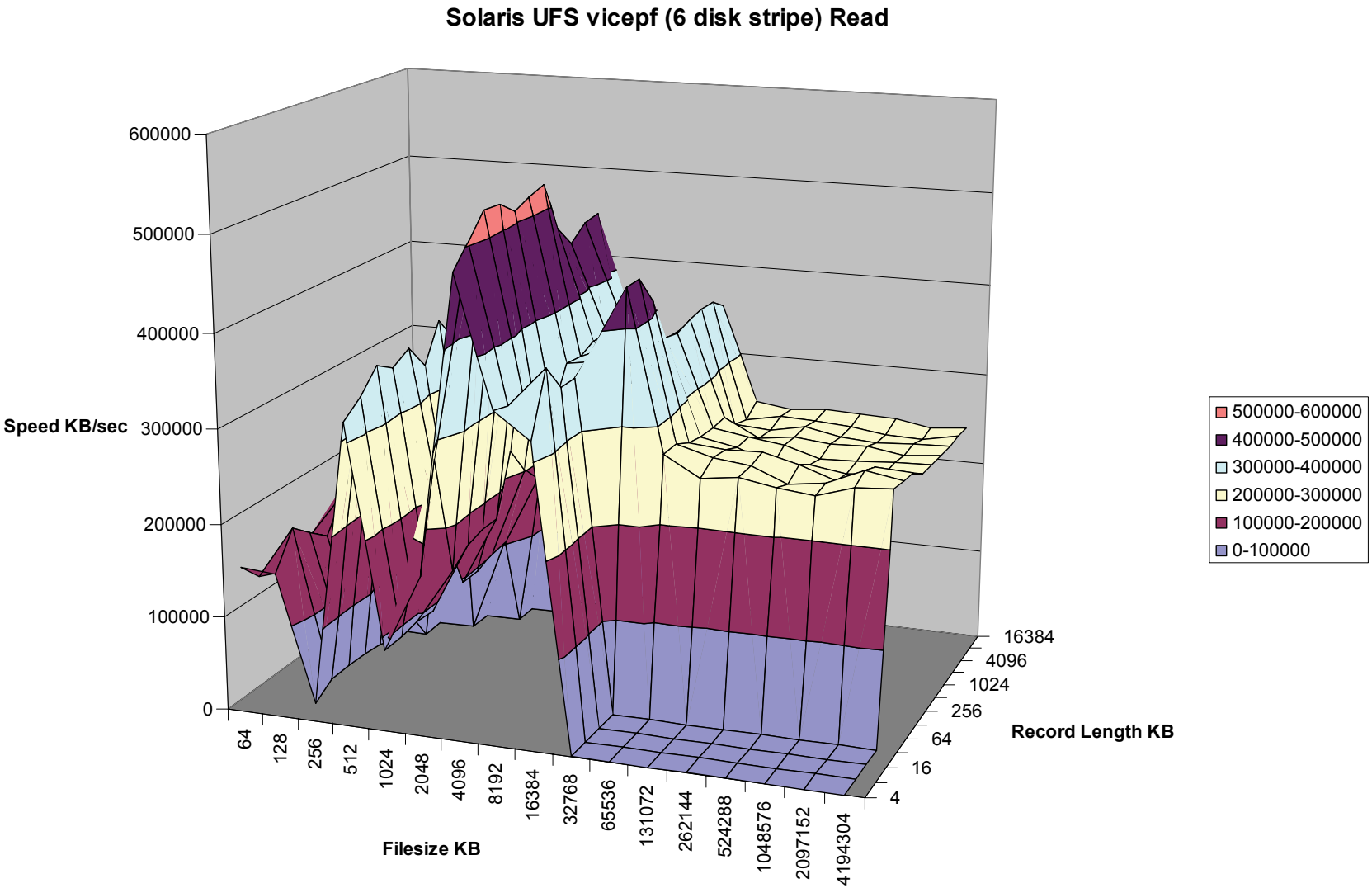
vicep{ j, k, l } – 3 disk stripe – only used for read problem encountered in multiclient test.

Solaris UFS Local Filesystems


Solaris UFS vicepb (single disk) Write
Speed KB/sec
100000
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
Filesize KB
16384
4096
1024
256
64
16
4
Record Length KB
90000-100000
80000-90000
70000-80000
60000-70000
50000-60000
40000-50000
30000-40000
20000-30000
10000-20000
0-10000

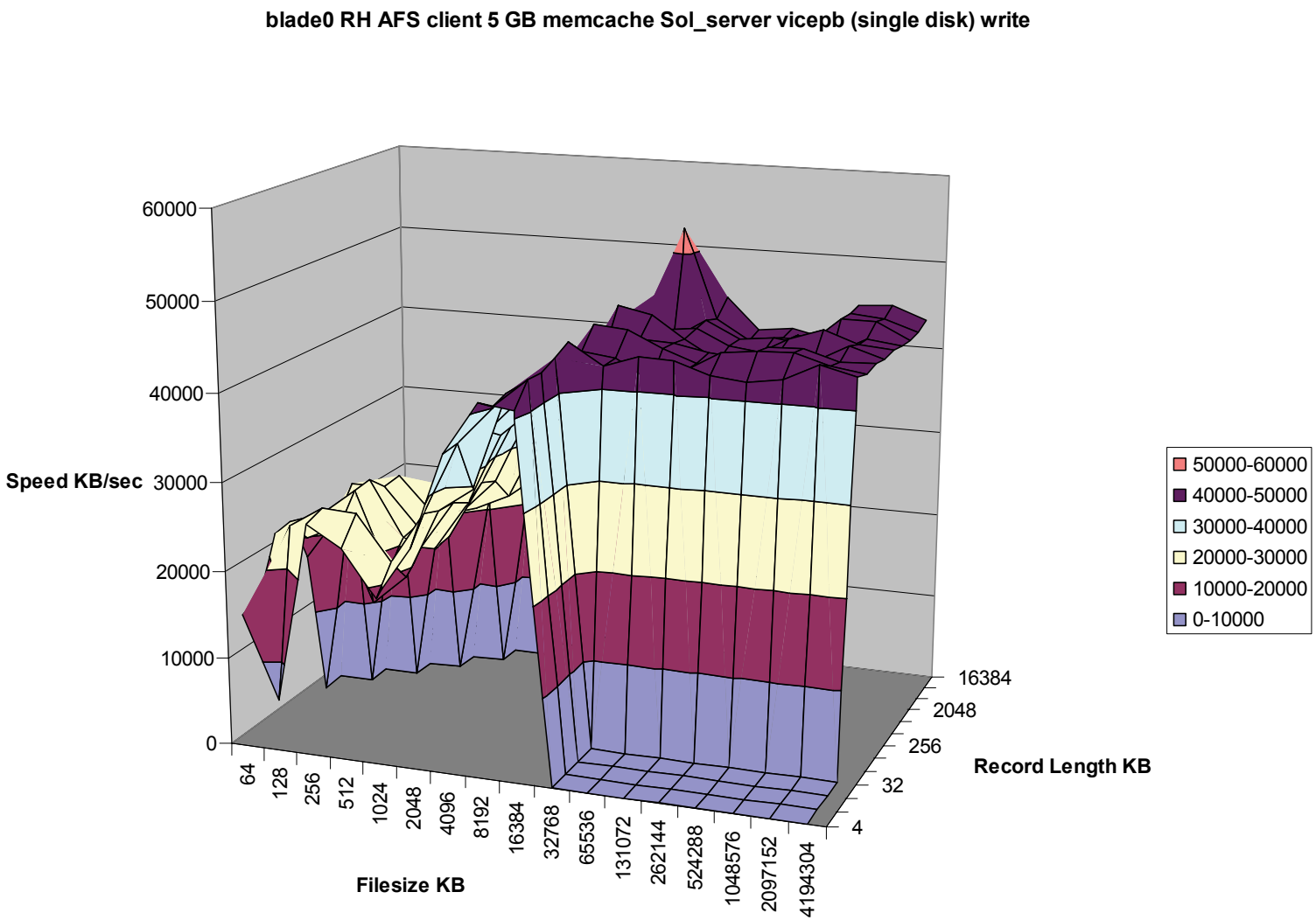

Solaris UFS vicepb (single disk) Read
Speed KB/sec
70000
60000
50000
40000
30000
20000
10000
0
Filesize KB
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
Record Length KB
16384
4096
1024
256
64
16
4
60000-70000
50000-60000
40000-50000
30000-40000
20000-30000
10000-20000
0-10000

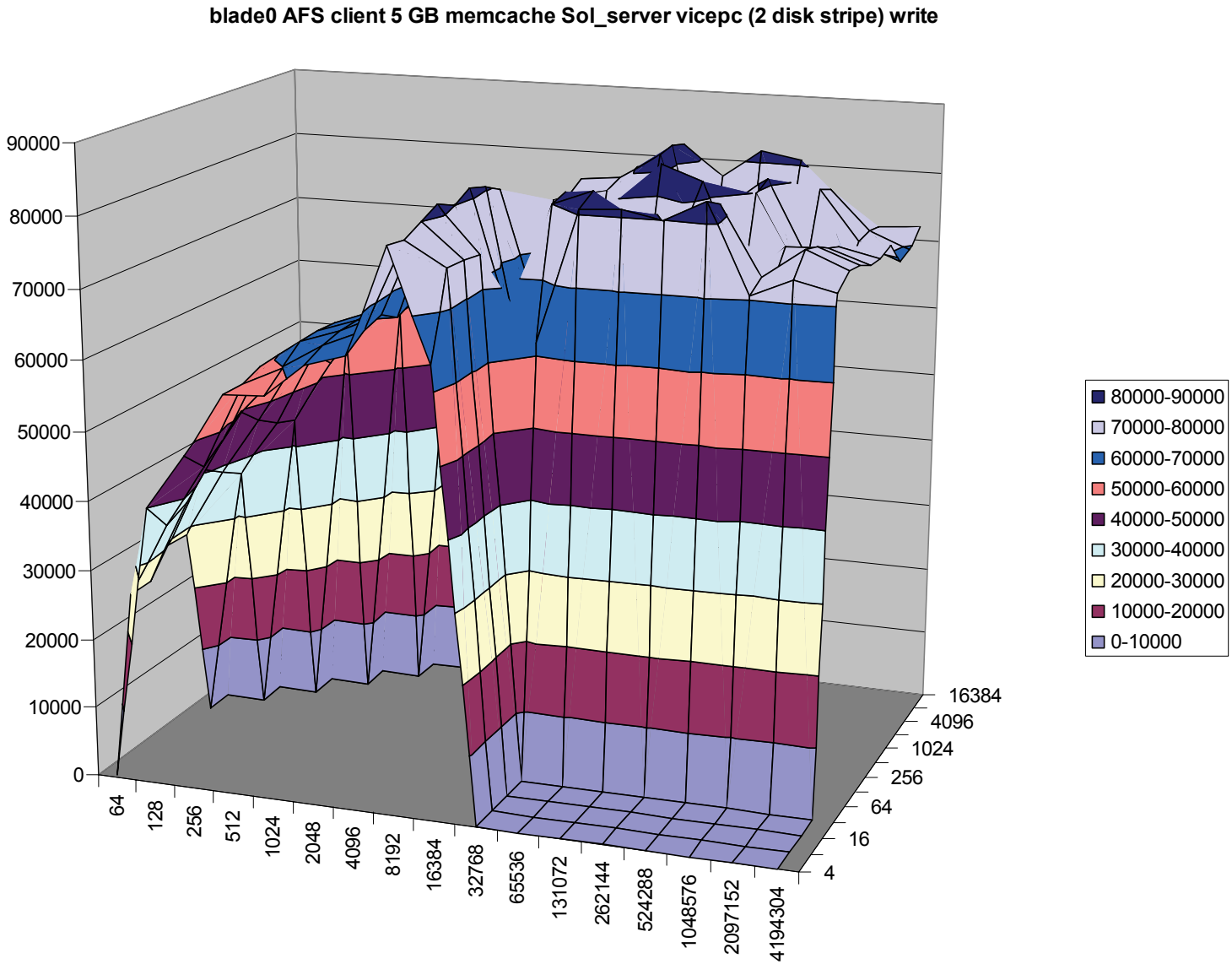

Solaris UFS vicepf (6 disk stripe) Write
Speed KB/sec
400000
350000
300000
250000
200000
150000
100000
50000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
Filesize KB
16384
4096
1024
256
64
16
4
Record Length KB
350000-400000
300000-350000
250000-300000
200000-250000
150000-200000
100000-150000
50000-100000
0-50000

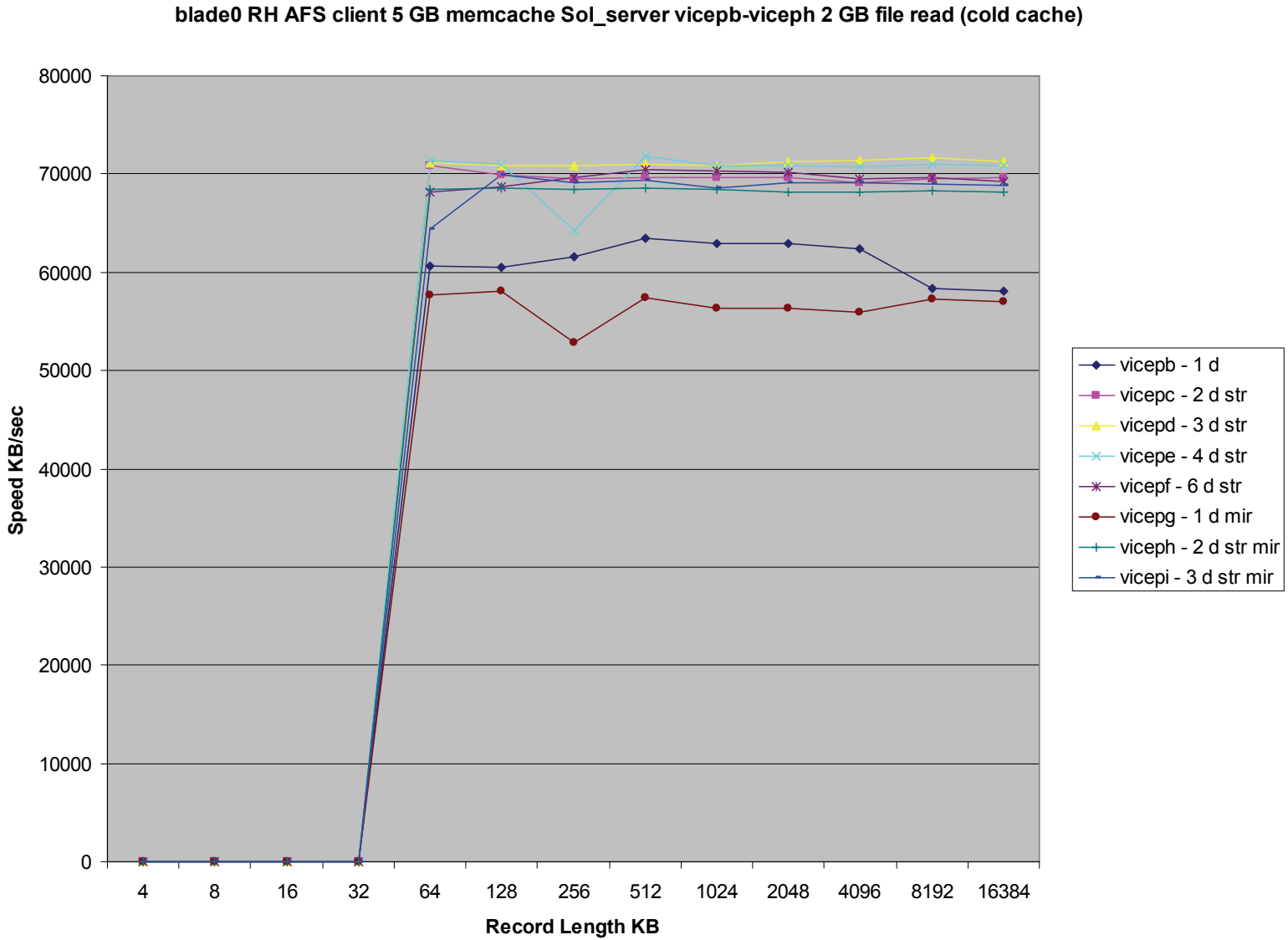

Solaris UFS vicepf (6 disk stripe) Read
Speed KB/sec
600000
500000
400000
300000
200000
100000
0
Filesize KB
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
Record Length KB
16384
4096
1024
256
64
16
4
500000-600000
400000-500000
300000-400000
200000-300000
100000-200000
0-100000

AFS single client tests

Single Sun Blade 8000 Red Hat AS 4 AFS memcache client with 1 Gb network, Solaris AFS server using 1 Gb network (to compare with RH AFS 1 Gb server).

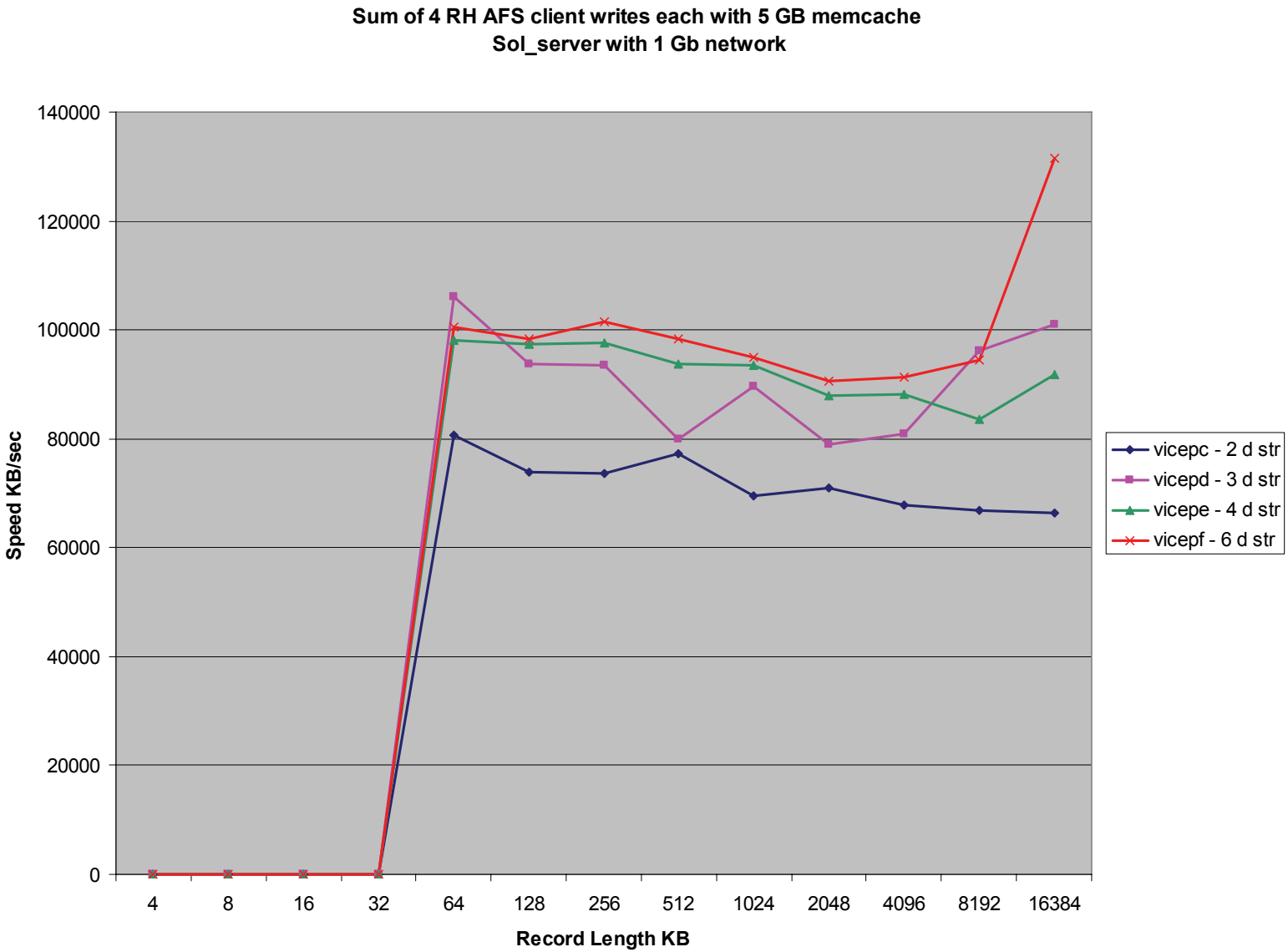

blade0 RH AFS client 5 GB memcache Sol_server vicepb (single disk) write
Speed KB/sec
60000
50000
40000
30000
20000
10000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
Filesize KB
16384
2048
256
32
4
Record Length KB
50000-60000
40000-50000
30000-40000
20000-30000
10000-20000
0-10000

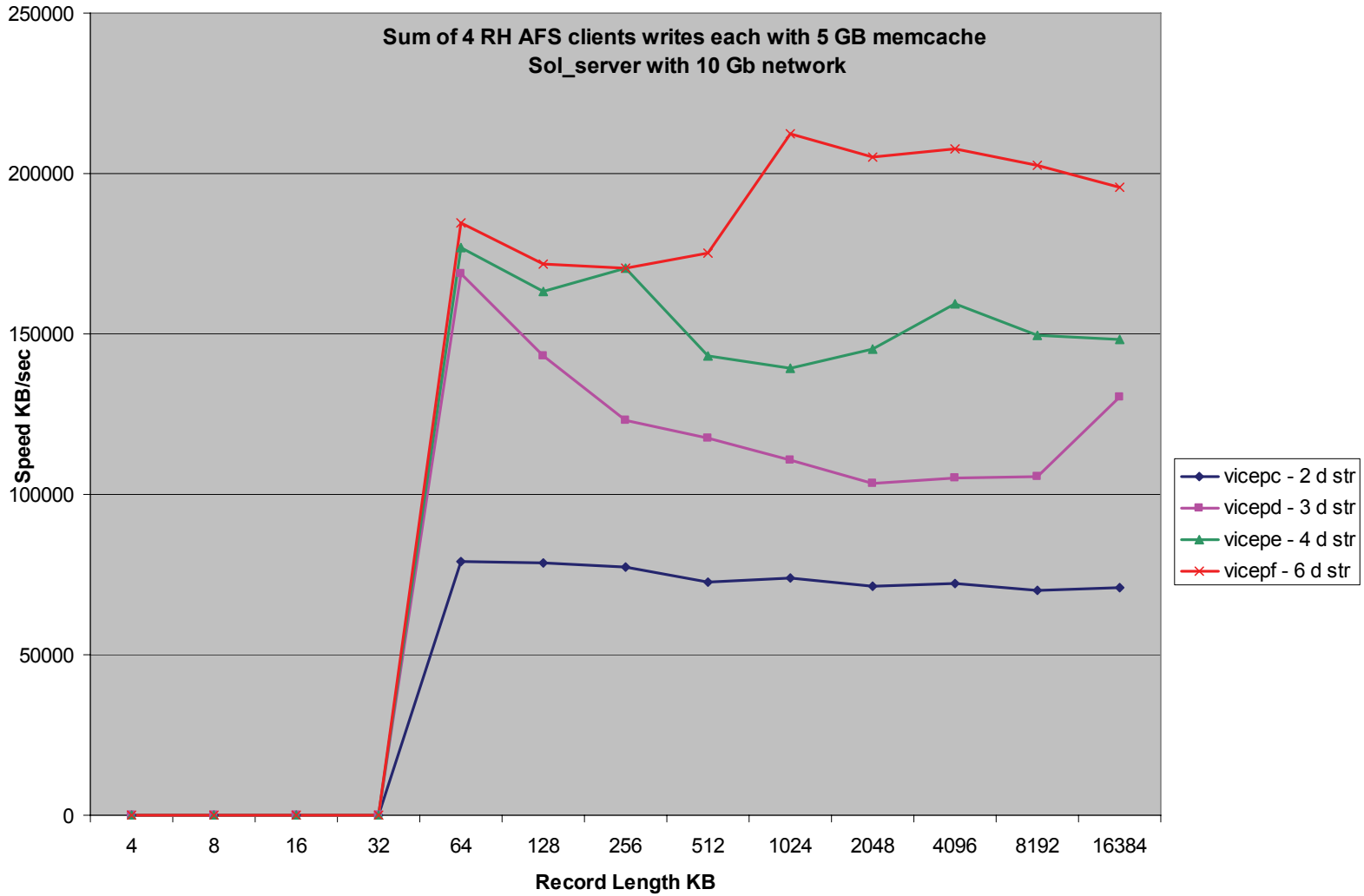

blade0 AFS client 5 GB memcache Sol_server vicepc (2 disk stripe) write
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
131072
262144
524288
1048576
2097152
4194304
16384
4096
1024
256
64
16
4
80000-90000
70000-80000
60000-70000
50000-60000
40000-50000
30000-40000
20000-30000
10000-20000
0-10000


blade0 RH AFS client 5 GB memcache Sol_server vicepb-viceph 2 GB file read (cold cache)
Speed KB/sec
80000
70000
60000
50000
40000
30000
20000
10000
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
Record Length KB
vicepb - 1 d
vicepc - 2 d str
vicepd - 3 d str
vicepe - 4 d str
vicepf - 6 d str
vicepg - 1 d mir
viceph - 2 d str mir
vicepi - 3 d str mir

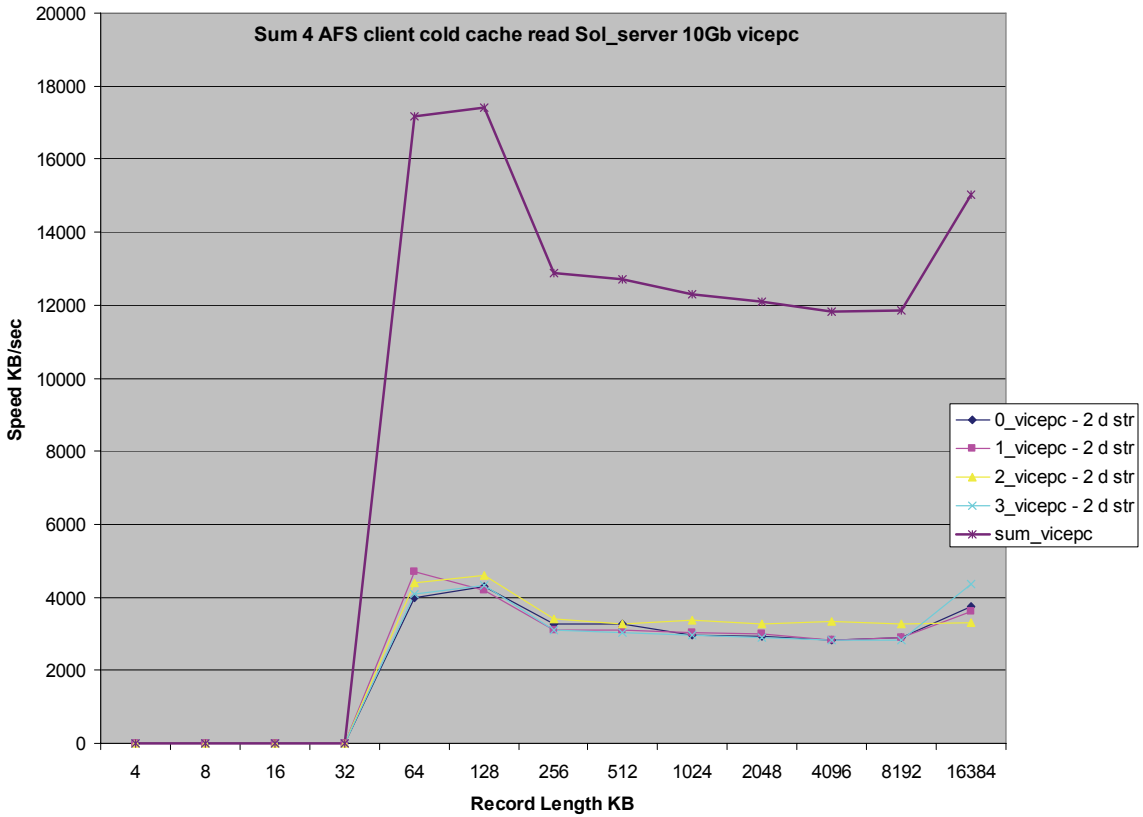
AFS multiple client tests

4 x Sun Blade 8000 Red Hat AS 4 AFS memcache client with 1 Gb network, Solaris AFS server using 1 & 10 Gb network

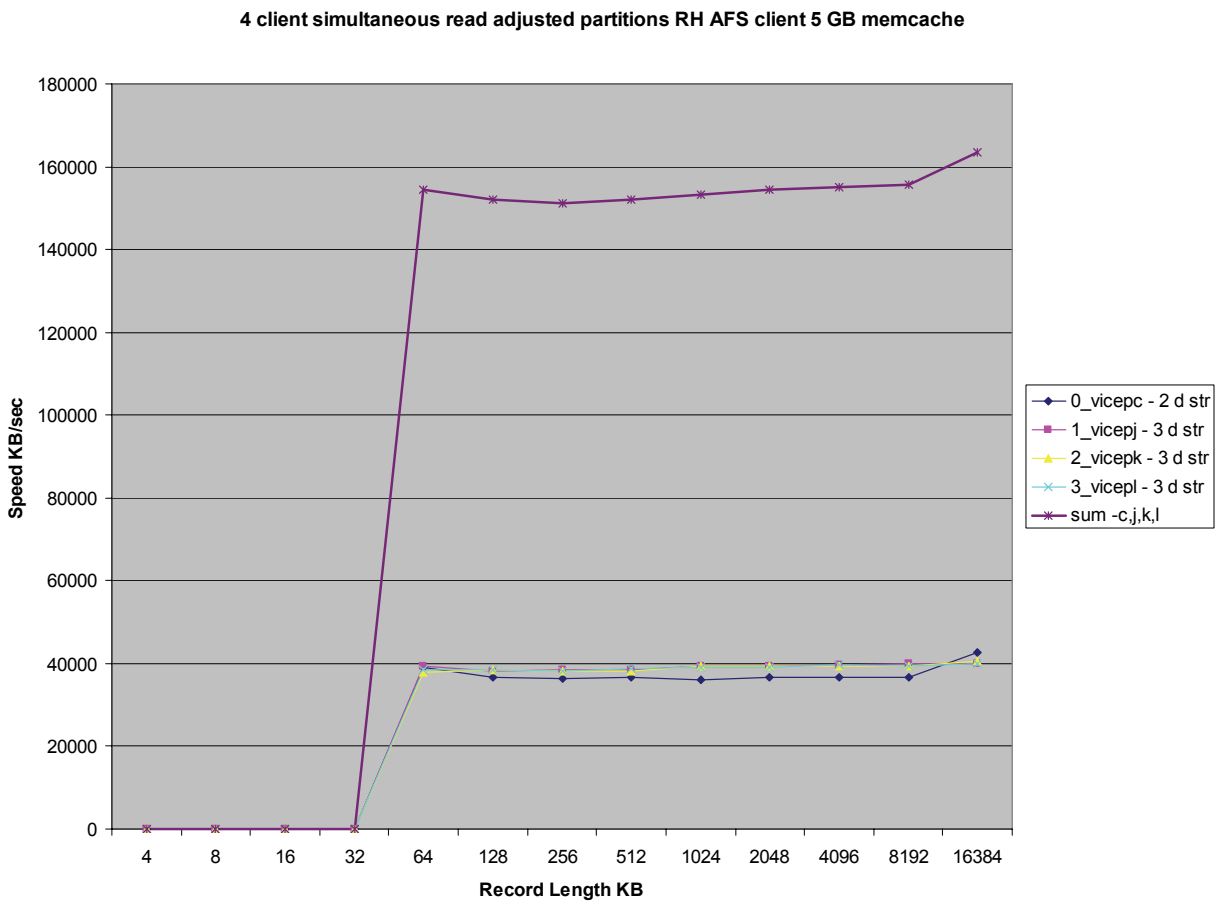

Sum of 4 RH AFS client writes each with 5 GB memcache
Sol_server with 1 Gb network
Speed KB/sec
140000
120000
100000
80000
60000
40000
20000
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
Record Length KB
vicepc - 2 d str
vicepd - 3 d str
vicepe - 4 d str
vicepf - 6 d str


Sum of 4 RH AFS clients writes each with 5 GB memcache
Sol_server with 10 Gb network
Speed KB/sec
250000
200000
150000
100000
50000
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
Record Length KB
vicepc - 2 d str
vicepd - 3 d str
vicepe - 4 d str
vicepf - 6 d str

Multiclient read performance is poor. Running on RH Linux fileserver also has problem when running with iozone – some "dropouts" with RH single client testing too.



Lets test by moving each clients data to separate filesystem and rerun.



Results

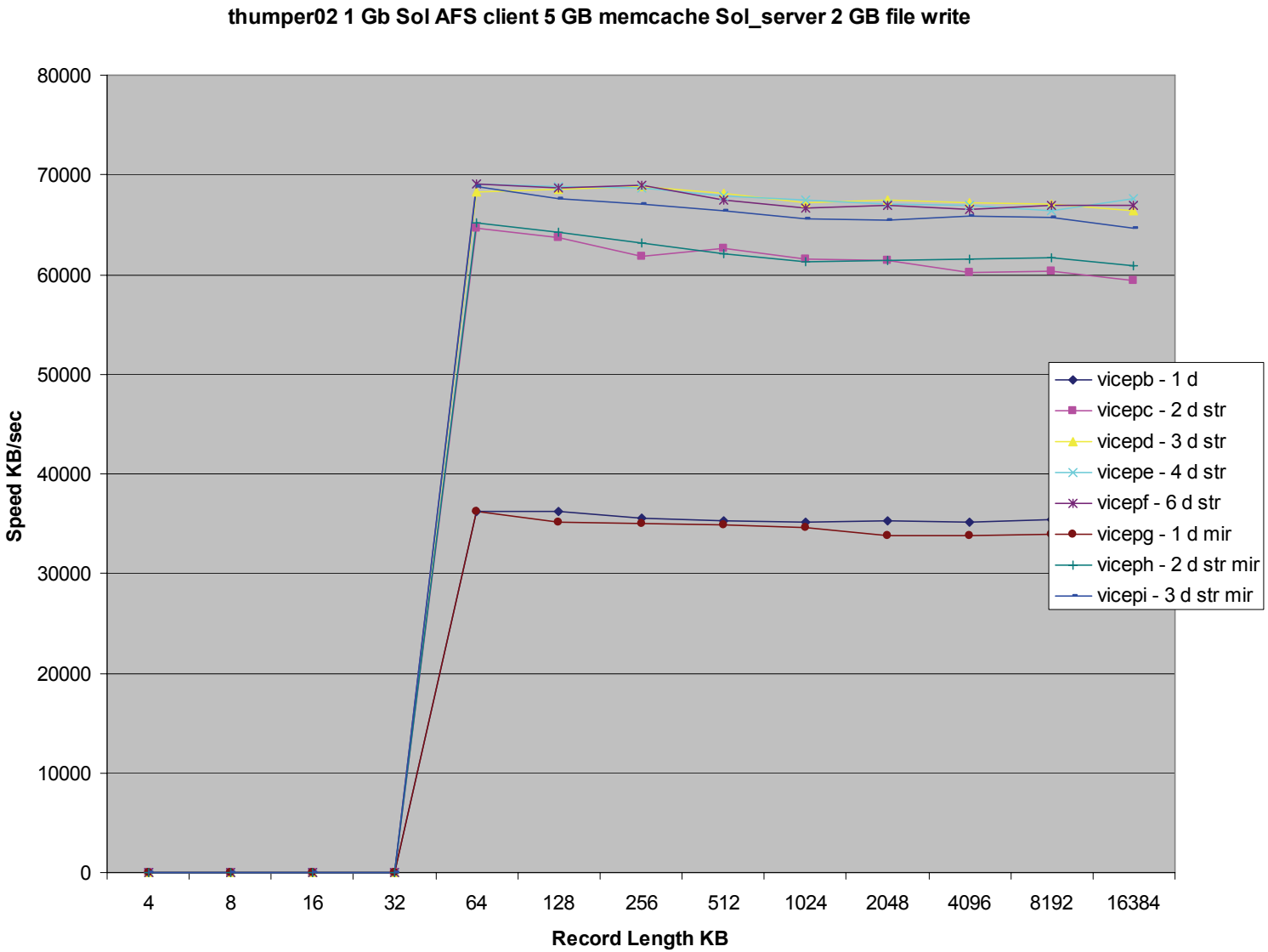
- Doing valid OpenAFS read tests are time consuming - caching effects both by processor (iozone) and Red Hat OS (readahead, readahead_early) – why turn off what we always use just for valid tests? Iozone typically just does unmount/remount of filesystem. In the real world do users flush data from their cache? Problems getting small read numbers that made sense with OpenAFS (512K) some obvious caching issues or problem with iozone – timing accuracy?
- Multi-client write tests good but problems with multi-client reads (OpenAFS bug? 60795) manually separating filesystems for each client fixed the problem – but this is critical issue in real operation.
- Performance results better than what I thought possible – Large memory cache provides great performance - 10 Gb network made more of a difference than what I thought – stability was were more of an issue than I thought.
- For fast clients you need to make sure that underlying filesystems are fast enough for good performance. For “thumper” 3 disk striped mirror (6 disk raid 10) across controllers makes most sense.
- Interesting to compare AFS results from 10 years ago with today’s http://www.nd.edu/~rich/Decorum97 - Performance still limited by cache – faster clients stress servers more – some past practices may not be valid currently e.g. single disk fileservers.

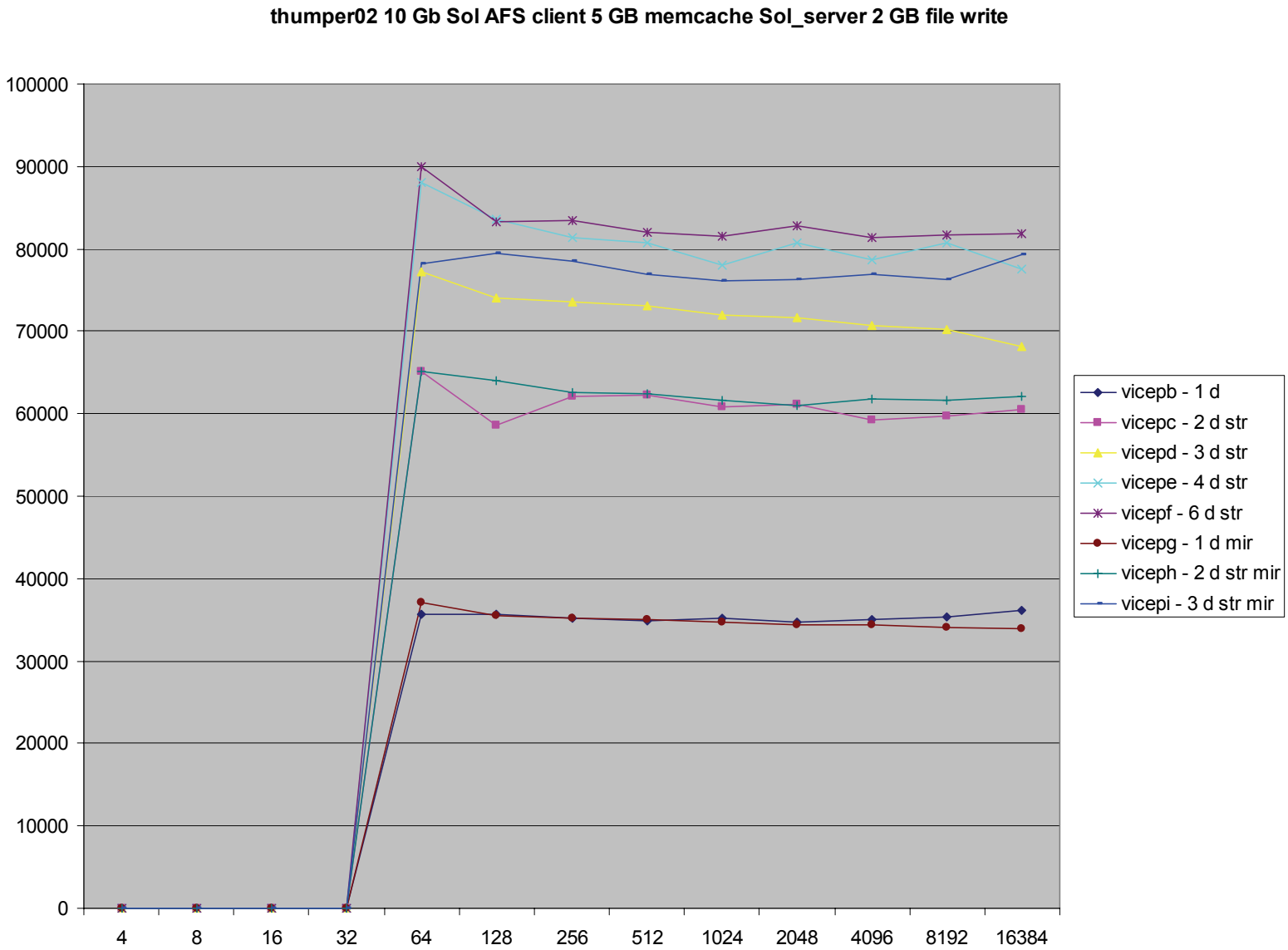
OpenAFS as a HPC filesystem?

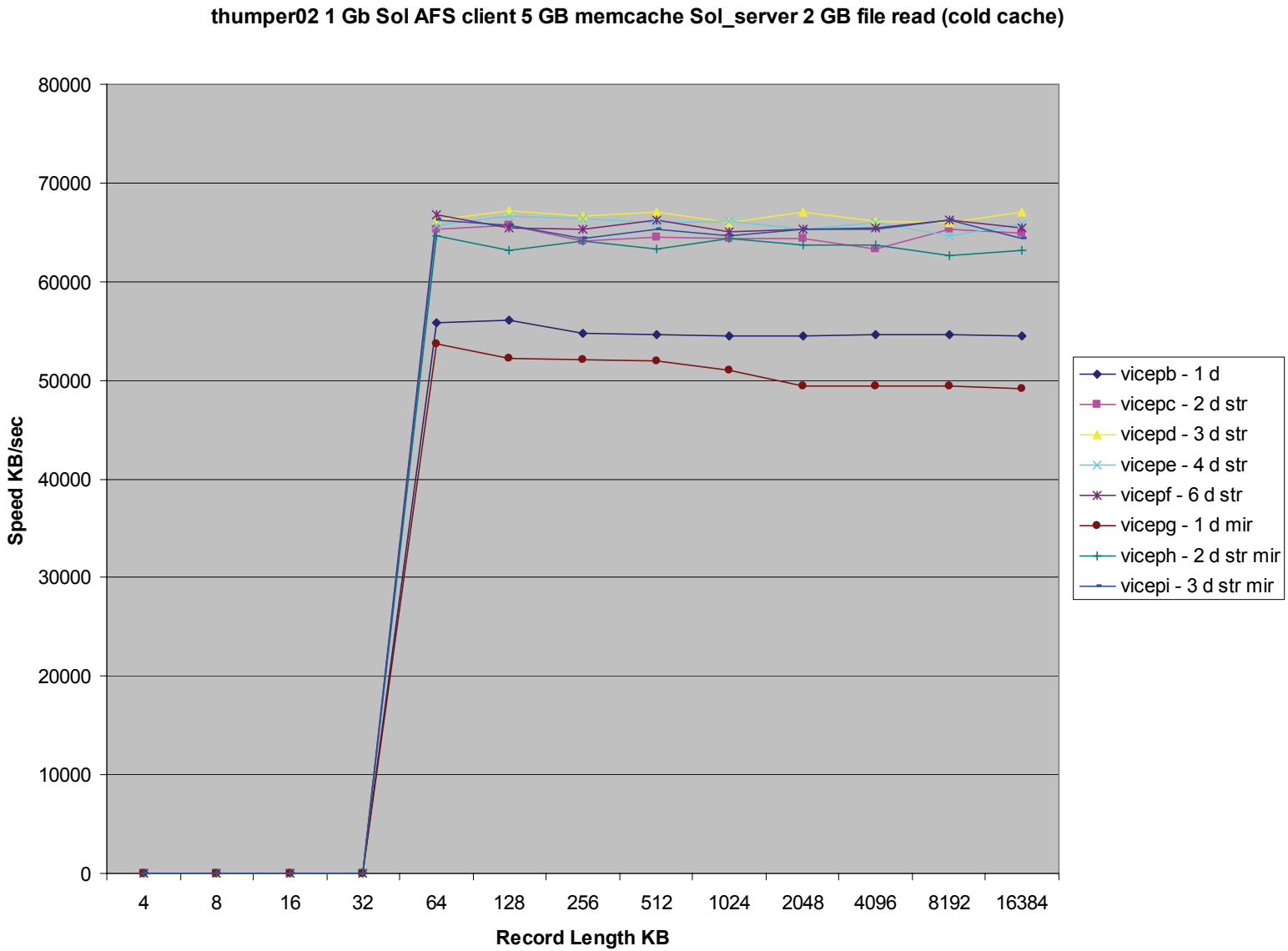
- Scalability is key – can the number of servers be scaled easily and cost effectively? AFS vendor neutral - heterogeneous OS support – allows for competition among vendors. Both key aspects when comparing HPC filesystems.
- Stability was an issue in testing. QA/Testing key - I believe this may be a goal of OpenAFS elders if not a work in progress.

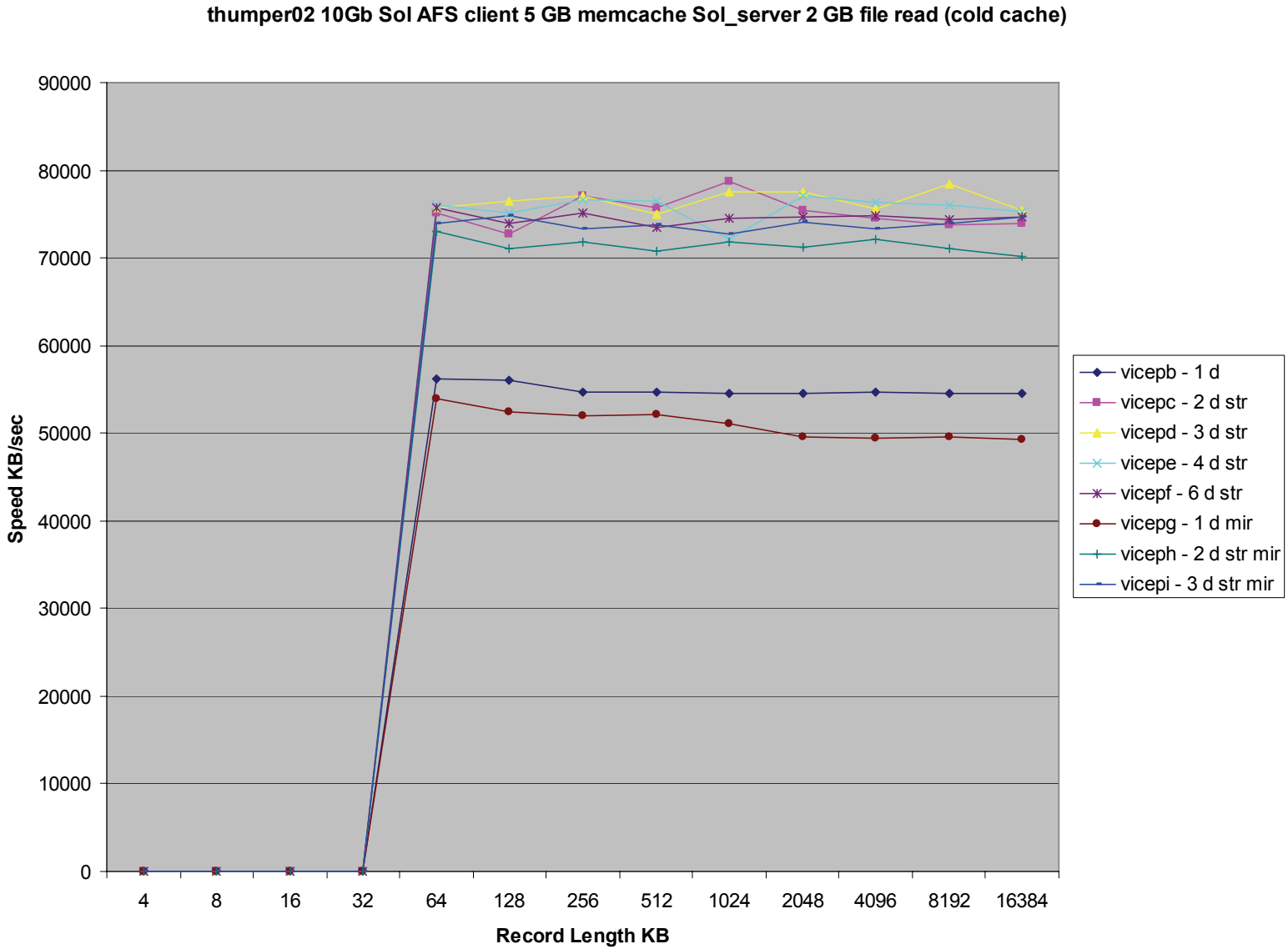
AFS single client tests

Single Solaris client using memcache 10 Gb network, Solaris server using 1 & 10 Gb server


thumper02 1 Gb Sol AFS client 5 GB memcache Sol_server 2 GB file write
Speed KB/sec
80000
70000
60000
50000
40000
30000
20000
10000
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
Record Length KB
vicepb - 1 d
vicepc - 2 d str
vicepd - 3 d str
vicepe - 4 d str
vicepf - 6 d str
vicepg - 1 d mir
viceph - 2 d str mir
vicepi - 3 d str mir


thumper02 10 Gb Sol AFS client 5 GB memcache Sol_server 2 GB file write
100000
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
vicepb - 1 d
vicepc - 2 d str
vicepd - 3 d str
vicepe - 4 d str
vicepf - 6 d str
vicepg - 1 d mir
viceph - 2 d str mir
vicepi - 3 d str mir


thumper02 1 Gb Sol AFS client 5 GB memcache Sol_server 2 GB file read (cold cache)
Speed KB/sec
80000
70000
60000
50000
40000
30000
20000
10000
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
Record Length KB
vicepb - 1 d
vicepc - 2 d str
vicepd - 3 d str
vicepe - 4 d str
vicepf - 6 d str
vicepg - 1 d mir
viceph - 2 d str mir
vicepi - 3 d str mir


thumper02 10Gb Sol AFS client 5 GB memcache Sol_server 2 GB file read (cold cache)
Speed KB/sec
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
Record Length KB
vicepb - 1 d
vicepc - 2 d str
vicepd - 3 d str
vicepe - 4 d str
vicepf - 6 d str
vicepg - 1 d mir
viceph - 2 d str mir
vicepi - 3 d str mir