# Comparative I/O Performance Evaluation of the Silicon Graphics Iris 4D/240S and Challenge L using the Self Scaling I/O Benchmark

*Carlo Kopp†*
*Carlo.Kopp@pha.com.au*

*ABSTRACT*

This report details the findings of a series of I/O benchmark tests using the Self Scaling I/O benchmark by Chen and Patterson. The test systems were a Silicon Graphics Iris 4D/240S server, and a Challenge L server. Results include both comparative analysis, as well as parametrised testing of the Challenge L varying both disks and installed main memory.

## 1. Introduction

The purpose of the tests to quantify the performance improvement seen with the replacement of the 4D/240 with the Challenge, as well as to scrutinise the possible performance gains to be seen on the Challenge with the installation of an alternate disk drive type, and additional memory.

Comparative tests between the two systems used the Fujitsu M2654HA 2 GB drives which were subsequently transfered between operational systems. Testing the two systems with the same drives was intended to provide an unbiased comparison of I/O throughput.

## 2. The Self Scaling I/O Benchmark

Established performance benchmarks generally do a poor job of analysing disk I/O performance. The SPEC suite is oriented toward CPU performance, the TPC series are transaction oriented, the SPEC/LADDIS NFS benchmark is designed for networked environments, while only the Bonnie and IOstone benchmarks are specifically oriented toward I/O throughput.

Both the Bonnie and IOstone benchmarks have a number of important limitations, which limit their applicability as a general analytical tool. IOstone will access only 1 MB of data, and is thus unable to saturate a contemporary system with traffic. Bonnie touches a full 100 MB of data, but even that may not be adequate for a contemporary file server. In either instance, the central weakness lies in an inability to scale the applied workload to the system automatically.

In 1993, Peter Chen and David Patterson, then both at UCB EECS, identified these limitations in existing I/O benchmark suites. They subsequently proceeded to develop a benchmark which addresses the problems discussed above (Chen P.M., Patterson D.A., Storage Performance - Metrics and Benchmarks, UCB EECS Technical Report, and Chen P.M., Patterson D.A., A New Approach to I/O Performance Evaluation - Self Scaling I/O Benchmarks, ACM Sigmetrics 1993 Conference on Measurement and Modelling of Computer Systems).

This benchmark is the Self Scaling I/O Benchmark. The central idea behind the benchmark was to produce a tool which would adaptively explore the performance of the tested system, with varying workload parameters. In this fashion a single benchmark would be capable of providing useful comparisons between machines with widely ranging I/O performance, in every instance the benchmark would in effect "explore" the performance space of the tested system.

---

† Carlo Kopp is a senior systems programmer and performance engineer with Peter Harding & Associates, Pty Ltd. He has extensive industry experience including the design of communications equipment and workstation I/O and motherboards.

The benchmark produces a workload defined by five separate parameters. The benchmark when running will find a focal point for each parameter, and then graph performance for one parameter, while all others are fixed. The result is a set of plots, which provide a indication of performance with the variation of the swept parameter. The focal point is chosen to lie at the middle of the range of any given parameter. The five parameters are :

**uniqueBytes** - the number of unique bytes of data written or read in the workload, essentially a measure of the data size. It is used to isolate cache performance from hardware performance.

**sizeMean** - the average size of an individual I/O request.

**readFrac** - the fraction of reads, were 1 - readFrac is the fraction of writes. This parameter is also important when analysing cache performance.

**seqFrac** - the fraction of I/O requests that sequentially follow the preceding request.

**processNum** - the number of processes concurrently generating I/O request, ie a metric of concurrency.

Chen and Patterson's published test results indicate that the benchmark yields an accuracy between predicted and measured performance typically between 10% and 15%, with a repeatability between test runs of about 5%. Given the nature of the measurement, these are indeed very good figures.

## 3. Tested System Configurations - Silicon Graphics Iris 4D/240S

The **Iris 4D/240S** was tested in the following configuration:

4 25 MHZ IP7 Processors
FPU: MIPS R2010A/R3010 VLSI Floating Point Chip Revision: 2.0
CPU: MIPS R2000A/R3000 Processor Chip Revision: 2.0
On-board serial ports: 2 per CPU board
Data cache size: 64 Kbytes
Instruction cache size: 64 Kbytes
Secondary data cache size: 256 Kbytes
Main memory size: 96 Mbytes
I/O board, slot F: IO3
Interphase 4210 Cougar VME-SCSI controller 0: Firmware revision A05
Disk drive: unit 4 on VME-SCSI controller 0
Disk drive: unit 3 on VME-SCSI controller 0
Disk drive: unit 2 on VME-SCSI controller 0
Disk drive: unit 1 on VME-SCSI controller 0
Integral Ethernet: et0, IO3
ENP-10 Ethernet controller: enp0, firmware version 4 (SGI)
Integral SCSI controller 1: Version WD33C93A, revision 9
Integral SCSI controller 0: Version WD33C93A, revision 9
Tape drive: unit 7 on SCSI controller 0: QIC 150
Disk drive: unit 1 on SCSI controller 0
VME bus: adapter 0

The disk drive type tested was specified as follows:

**Fujitsu M2654HA** 5.25" 2.4 GB SCSI-2 Drive

Data Transfer Performance:

10 MHz Clock Differential SCSI-2 Interface
HDA Data Transfer Rate @ 4.758 MB/sec
256kByte Data Buffer, Read Lookahead and Write Caching

Access Time Performance:

5,400 RPM Spindle
Rotational Latency, Av @ 5.56 msec
Track-Track Seek @ 3.0 msec
Max Seek @ 22.0 msec
Average Seek @ 11 msec

The **Irix 4.3** kernel configuration used was as follows (SGI defaults):

```
 group: cachefs (dynamically changeable)
        cachefs_max_threads = 5 (0x5)
        cachefs_readahead = 1 (0x1)
        cachefs_max_lru = 1000 (0x3e8)

 group: uds (statically changeable)
        unpdg_recvspace = 16384 (0x4000)
```

unpdg_sendspace = 8192 (0x2000)
unpst_recvspace = 16384 (0x4000)
unpst_sendspace = 16384 (0x4000)

group: snfs (statically changeable)
svc_maxdupreqs = 682 (0x2aa)
nfs_portmon = 0 (0x0)

group: shm (statically changeable)
shmall = 512 (0x200)
sshmseg = 100 (0x64)
shmmni = 100 (0x64)
shmmin = 1 (0x1)
shmmax = 536870912 (0x20000000)

group: sem (statically changeable)
semaem = 16384 (0x4000)
semvmx = 32767 (0x7fff)
semume = 10 (0xa)
semopm = 10 (0xa)
semmsl = 25 (0x19)
semmnu = 60 (0x3c)
semmns = 2048 (0x800)
semmni = 200 (0xc8)

group: pipefs (dynamically changeable)
svr3pipe = 1 (0x1)

group: lockmgr (dynamically changeable)
working_timeout = 5 (0x5)
normal_timeout = 5 (0x5)
first_timeout = 1 (0x1)

group: msg (statically changeable)
msgseg = 1536 (0x600)
msgtql = 40 (0x28)
msgssz = 8 (0x8)
msgmni = 50 (0x32)
msgmnb = 32768 (0x8000)
msgmax = 16384 (0x4000)

group: internal (statically changeable)
mrmeterhash = 31 (0x1f)
smeterhash = 255 (0xff)
dumplo = 0 (0x0)
putbufsz = 1024 (0x400)
conbufsz = 1024 (0x400)
histmax = 0 (0x0)

group: splock_pool (statically changeable)
file_pool_size = 1024 (0x400)
vnode_pool_size = 1024 (0x400)
sema_pool_size = 8192 (0x2000)

```
group: mload (statically changeable)
        vfssw_extra = 5 (0x5)
        fmodsw_extra = 20 (0x14)
        cdevsw_extra = 23 (0x17)
        bdevsw_extra = 21 (0x15)

group: bufcache (statically changeable)
        nbuf = 714 (0x2ca)

group: paging (dynamically changeable)
        autoup = 10 (0xa)
        dwcluster = 64 (0x40)
        rsshogslop = 20 (0x14)
        rsshogfrac = 75 (0x4b)
        tlbdrop = 100 (0x64)
        maxlkmem = 2000 (0x7d0)
        minasmem = 50 (0x32)
        minarmem = 50 (0x32)
        vfs_syncr = 30 (0x1e)
        bdflushr = 5 (0x5)
        maxdc = 100 (0x64)
        maxfc = 100 (0x64)
        maxsc = 100 (0x64)
        gpgsmsk = 2 (0x2)
        gpgshi = 1980 (0x7bc)
        gpgslo = 990 (0x3de)

group: memsize (statically changeable)
        maxphyscolors = 4096 (0x1000)
        maxpglst = 100 (0x64)
        maxdmasz = 1024 (0x400)
        syssegsz = 12287 (0x2fff)
        maxpmem = 0 (0x0)

group: timer (statically changeable)
        timetrim = 1089270 (0x109ef6)
        itimer_on_clkcpu = 0 (0x0)
        fasthz = 1200 (0x4b0)

group: signals (dynamically changeable)
        maxsigq = 96 (0x60)

group: actions (statically changeable)
        nactions = 68 (0x44)

group: streams (statically changeable)
        strctlsz = 1024 (0x400)
        strmsgsz = 32768 (0x8000)
        nstrpush = 9 (0x9)

group: numproc (statically changeable)
        ngroups_max = 16 (0x10)
        ncsize = 1078 (0x436)
        ncallout = 219 (0xdb)
```

callout_himark = 471 (0x1d7)
ndquot = 1078 (0x436)
nproc = 439 (0x1b7)

group: resource (statically changeable)
rlimit_rss_max = 536870912 (0x20000000)
rlimit_rss_cur = 92557312 (0x5845000)
rlimit_vmem_max = 536870912 (0x20000000)
rlimit_vmem_cur = 536870912 (0x20000000)
rlimit_nofile_max = 2500 (0x9c4)
rlimit_nofile_cur = 200 (0xc8)
rlimit_core_max = 2147483647 (0x7fffffff)
rlimit_core_cur = 2147483647 (0x7fffffff)
rlimit_stack_max = 536870912 (0x20000000)
rlimit_stack_cur = 67108864 (0x4000000)
rlimit_data_max = 536870912 (0x20000000)
rlimit_data_cur = 536870912 (0x20000000)
rlimit_fsize_max = 255 (0xff)
rlimit_fsize_cur = 0 (0x0)
rlimit_cpu_max = 2147483647 (0x7fffffff)
rlimit_cpu_cur = 2147483647 (0x7fffffff)

group: limits (dynamically changeable)
vnode_free_ratio = 2 (0x2)
maxup = 150 (0x96)
reserve_ncallout = 8 (0x8)
maxsymlinks = 30 (0x1e)
nprofile = 100 (0x64)
maxwatchpoints = 100 (0x64)
shlbmax = 8 (0x8)
ncargs = 20480 (0x5000)

group: misc (dynamically changeable)
module_unld_delay = 5 (0x5)
ecc_recover_enable = 60 (0x3c)
panic_on_sbe = 0 (0x0)
corepluspid = 0 (0x0)
r4k_div_patch = 0 (0x0)

group: switch (statically changeable)
reboot_on_panic = -1 (0xffffffff)
use_old_serialnum = 0 (0x0)
posix_tty_default = 0 (0x0)
restricted_chown = 0 (0x0)
nosuidshells = 1 (0x1)

group: efs (dynamically changeable)
efs_inline = 0 (0x0)

group: disp (dynamically changeable)
runq_dl_maxuse = 700 (0x2bc)
runq_dl_nonpriv = 200 (0xc8)
runq_dl_refframe = 1000 (0x3e8)
slice_size = 3 (0x3)

```
ndpri_lolim = 39 (0x27)
ndpri_hilim = 128 (0x80)
```

## 4. Tested System Configurations - Silicon Graphics Challenge L

The **Challenge L** was tested in the following configurations:

2 150 MHZ IP19 Processors
CPU: MIPS R4400 Processor Chip Revision: 5.0
FPU: MIPS R4010 Floating Point Chip Revision: 0.0
Data cache size: 16 Kbytes
Instruction cache size: 16 Kbytes
Secondary unified instruction/data cache size: 1 Mbyte
Main memory size: 128 Mbytes, 2-way interleaved
I/O board, Ebus slot 5: IO4 revision 1
Integral EPC serial ports: 4
Integral Ethernet controller: et0, Ebus slot 5
EPC external interrupts
Integral SCSI controller 4: Version WD33C95A, differential, revision 0
Integral SCSI controller 3: Version WD33C95A, differential, revision 0
Disk drive: unit 5 on SCSI controller 3
Disk drive: unit 4 on SCSI controller 3
Disk drive: unit 2 on SCSI controller 3
Disk drive: unit 1 on SCSI controller 3
Integral SCSI controller 2: Version WD33C95A, single ended, revision 0
Tape drive: unit 7 on SCSI controller 2: 8mm(8500) cartridge
Disk drive: unit 4 on SCSI controller 2
Disk drive: unit 3 on SCSI controller 2
Disk drive: unit 2 on SCSI controller 2
Tape drive: unit 1 on SCSI controller 2: 8mm(8500) cartridge
Integral SCSI controller 1: Version WD33C95A, differential, revision 0
Disk drive: unit 1 on SCSI controller 1
Integral SCSI controller 0: Version WD33C95A, single ended, revision 0
CDROM: unit 3 on SCSI controller 0
CC synchronization join counter
Integral EPC parallel port: Ebus slot 5
VME bus: adapter 0 mapped to adapter 21
VME bus: adapter 21

Alternate tests saw the use of 256 MB Main memory size

The disk types tested were the Fujitsu M2654HA (above) and the Seagate Elite 9:

**ST-410800N**
**Elite 9**

UNFORMATTED CAPACITY ______________________10,800
FORMATTED CAPACITY (512 byte blk) ________9,090
AVERAGE SECTORS PER TRACK _________________133 (rounded down)
ACTUATOR TYPE _____________________________ROTARY VOICE COIL
TRACKS _____________________________________132,975
CYLINDERS __________________________________4925
HEADS ______________________________________27
DISCS (5.25 in) ____________________________14
MEDIA TYPE _________________________________THIN FILM
HEAD TYPE __________________________________THIN FILM
RECORDING METHOD ___________________________ZBR RLL (1,7)

INTERNAL TRANSFER RATE (mbits/sec) _______45-66
INTERNAL TRANSFER RATE AVG (mbyte/sec) ___7.2
EXTERNAL TRANSFER RATE (mbyte/sec) _______10 (burst)
SPINDLE SPEED (RPM) ______________________5,400
AVERAGE LATENCY (mSEC) ___________________5.55
COMMAND OVERHEAD (mSEC) __________________<0.5
BUFFER ___________________________________1024 Kbyte
Read Look-Ahead, Adaptive, Multi-Segmented
Cache
INTERFACE ________________________________SCSI-2 FAST
BYTES PER TRACK __________________________63,000-91,000
SECTORS PER DRIVE ________________________17,845,731
BYTES PER CYLINDER _______________________1,058,400 to 1,587,600
TPI (TRACKS PER INCH) ____________________3,979
BPI (BITS PER INCH) ______________________
AVERAGE ACCESS (ms) read/write ___________11/12
SINGLE TRACK SEEK (ms) read/write ________0.9/1.7
MAX FULL SEEK (ms) read/write ____________23/24
MTBF (power-on hours) Class A Room _______NA
MTBF (power-on hours) Office _____________500,000
POWER DISSIPATION (watts/BTUs) Active ____25/85
Idle ______21/72
POWER REQUIREMENTS: Single-ended/Differential
+12V START-UP (amps) _3.98/4.60
+12V TYPICAL (amps) __1.90/1.88
+5V START-UP (amps) __2.37/3.48
+5V TYPICAL (amps) ___1.96/2.77
TYPICAL ( watts seek/read)_32.60/36.41
IDLE (watts) ______________28.34/31.81
USER MANUAL PART NUMBER __________________

** Already low-level formatted at the factory with nine spare sectors per cylinder, one spare cylinders/unit, one system cylinder/unit, and one diagnostic cylinder/unit.

Physical:

Height (inches/mm): 3.25/82.6
Width (inches/mm): 5.75/146.1
Depth (inches/mm): 8.0/203
Weight (lbs/kg): 7.8/3.6

The **Irix 5.3** kernel configuration used was as follows (SGI defaults):

```
group: cachefs (dynamically changeable)
        cachefs_max_threads = 5 (0x5)
        cachefs_readahead = 1 (0x1)
        cachefs_max_lru = 1000 (0x3e8)

group: uds (statically changeable)
        unpdg_recvspace = 16384 (0x4000)
        unpdg_sendspace = 8192 (0x2000)
        unpst_recvspace = 16384 (0x4000)
        unpst_sendspace = 16384 (0x4000)
```

```
group: snfs (statically changeable)
        svc_maxdupreqs = 955 (0x3bb)
        nfs_portmon = 0 (0x0)

group: shm (statically changeable)
        shmall = 512 (0x200)
        sshmseg = 100 (0x64)
        shmmni = 100 (0x64)
        shmmin = 1 (0x1)
        shmmax = 536870912 (0x20000000)

group: sem (statically changeable)
        semaem = 16384 (0x4000)
        semvmx = 32767 (0x7fff)
        semume = 10 (0xa)
        semopm = 10 (0xa)
        semmsl = 25 (0x19)
        semmnu = 60 (0x3c)
        semmns = 2048 (0x800)
        semmni = 200 (0xc8)

group: pipefs (dynamically changeable)
        svr3pipe = 1 (0x1)

group: lockmgr (dynamically changeable)
        working_timeout = 5 (0x5)
        normal_timeout = 5 (0x5)
        first_timeout = 1 (0x1)

group: msg (statically changeable)
        msgseg = 1536 (0x600)
        msgtql = 40 (0x28)
        msgssz = 8 (0x8)
        msgmni = 50 (0x32)
        msgmnb = 32768 (0x8000)
        msgmax = 16384 (0x4000)

group: internal (statically changeable)
        mrmeterhash = 31 (0x1f)
        smeterhash = 255 (0xff)
        dumplo = 0 (0x0)
        putbufsz = 1024 (0x400)
        conbufsz = 1024 (0x400)
        histmax = 0 (0x0)

group: splock_pool (statically changeable)
        file_pool_size = 1024 (0x400)
        vnode_pool_size = 1024 (0x400)
        sema_pool_size = 8192 (0x2000)

group: mload (statically changeable)
        vfssw_extra = 5 (0x5)
        fmodsw_extra = 20 (0x14)
        cdevsw_extra = 23 (0x17)
```

```
        bdevsw_extra = 21 (0x15)

group: bufcache (statically changeable)
        nbuf = 919 (0x397)

group: paging (dynamically changeable)
        autoup = 10 (0xa)
        dwcluster = 64 (0x40)
        rsshogslop = 20 (0x14)
        rsshogfrac = 75 (0x4b)
        tlbdrop = 100 (0x64)
        maxlkmem = 2000 (0x7d0)
        minasmem = 50 (0x32)
        minarmem = 50 (0x32)
        vfs_syncr = 30 (0x1e)
        bdflushr = 5 (0x5)
        maxdc = 100 (0x64)
        maxfc = 100 (0x64)
        maxsc = 100 (0x64)
        gpgsmsk = 1 (0x1)
        gpgshi = 2644 (0xa54)
        gpgslo = 1322 (0x52a)

group: memsize (statically changeable)
        maxphyscolors = 4096 (0x1000)
        maxpglst = 100 (0x64)
        maxdmasz = 1024 (0x400)
        syssegsz = 20991 (0x51ff)
        maxpmem = 0 (0x0)

group: timer (statically changeable)
        timetrim = 31325 (0x7a5d)
        itimer_on_clkcpu = 0 (0x0)
        fasthz = 1000 (0x3e8)

group: signals (dynamically changeable)
        maxsigq = 96 (0x60)

group: actions (statically changeable)
        nactions = 62 (0x3e)

group: streams (statically changeable)
        strctlsz = 1024 (0x400)
        strmsgsz = 32768 (0x8000)
        nstrpush = 9 (0x9)

group: numproc (statically changeable)
        ngroups_max = 16 (0x10)
        ncsize = 1352 (0x548)
        ncallout = 288 (0x120)
        callout_himark = 608 (0x260)
        ndquot = 1352 (0x548)
        nproc = 576 (0x240)
```

group: resource (statically changeable)
rlimit_rss_max = 536870912 (0x20000000)
rlimit_rss_cur = 123834368 (0x7619000)
rlimit_vmem_max = 536870912 (0x20000000)
rlimit_vmem_cur = 536870912 (0x20000000)
rlimit_nofile_max = 2500 (0x9c4)
rlimit_nofile_cur = 200 (0xc8)
rlimit_core_max = 2147483647 (0x7fffffff)
rlimit_core_cur = 2147483647 (0x7fffffff)
rlimit_stack_max = 536870912 (0x20000000)
rlimit_stack_cur = 67108864 (0x4000000)
rlimit_data_max = 536870912 (0x20000000)
rlimit_data_cur = 536870912 (0x20000000)
rlimit_fsize_max = 255 (0xff)
rlimit_fsize_cur = 0 (0x0)
rlimit_cpu_max = 2147483647 (0x7fffffff)
rlimit_cpu_cur = 2147483647 (0x7fffffff)

group: limits (dynamically changeable)
vnode_free_ratio = 2 (0x2)
maxup = 150 (0x96)
reserve_ncallout = 5 (0x5)
maxsymlinks = 30 (0x1e)
nprofile = 100 (0x64)
maxwatchpoints = 100 (0x64)
shlbmax = 8 (0x8)
ncargs = 20480 (0x5000)

group: misc (dynamically changeable)
module_unld_delay = 5 (0x5)
ecc_recover_enable = 60 (0x3c)
panic_on_sbe = 0 (0x0)
corepluspid = 0 (0x0)
r4k_div_patch = 0 (0x0)

group: switch (statically changeable)
reboot_on_panic = -1 (0xffffffff)
use_old_serialnum = 0 (0x0)
posix_tty_default = 0 (0x0)
restricted_chown = 0 (0x0)
nosuidshells = 1 (0x1)

group: efs (dynamically changeable)
efs_inline = 0 (0x0)

group: disp (dynamically changeable)
runq_dl_maxuse = 700 (0x2bc)
runq_dl_nonpriv = 200 (0xc8)
runq_dl_refframe = 1000 (0x3e8)
slice_size = 3 (0x3)
ndpri_lolim = 39 (0x27)
ndpri_hilim = 128 (0x80)

## 5. Test Results - Iris 4D/240S

The Iris was tested only with Fujitsu M2654HA drive array, and the measurement was used as a baseline for comparing the performance of the Challenge L.

Performance for the Iris is depicted in Fig.1, which contains plots of throughput in Mbytes/s for range of values of uniqueBytes, sizeMean, readFrac, seqFrac and processNum. Interpretation of these plots yields some interesting insights into system performance.

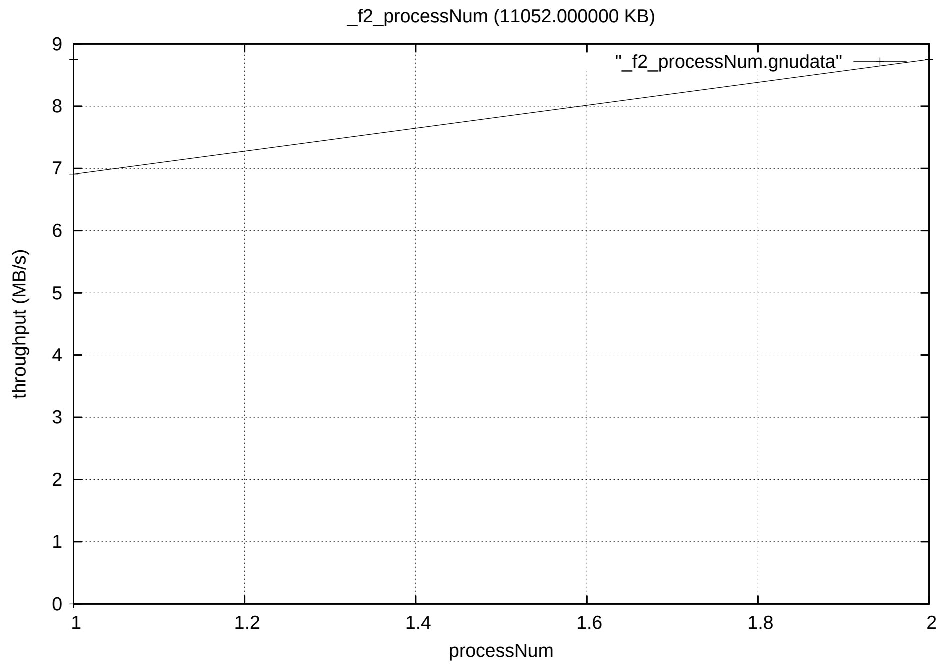


**Figure 1.1**

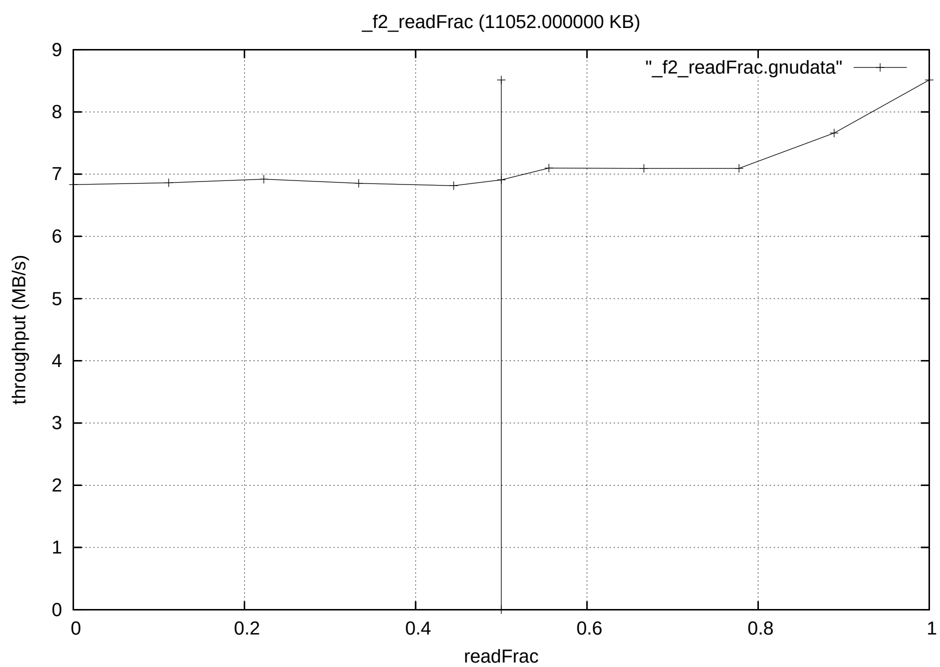


**Figure 1.2**

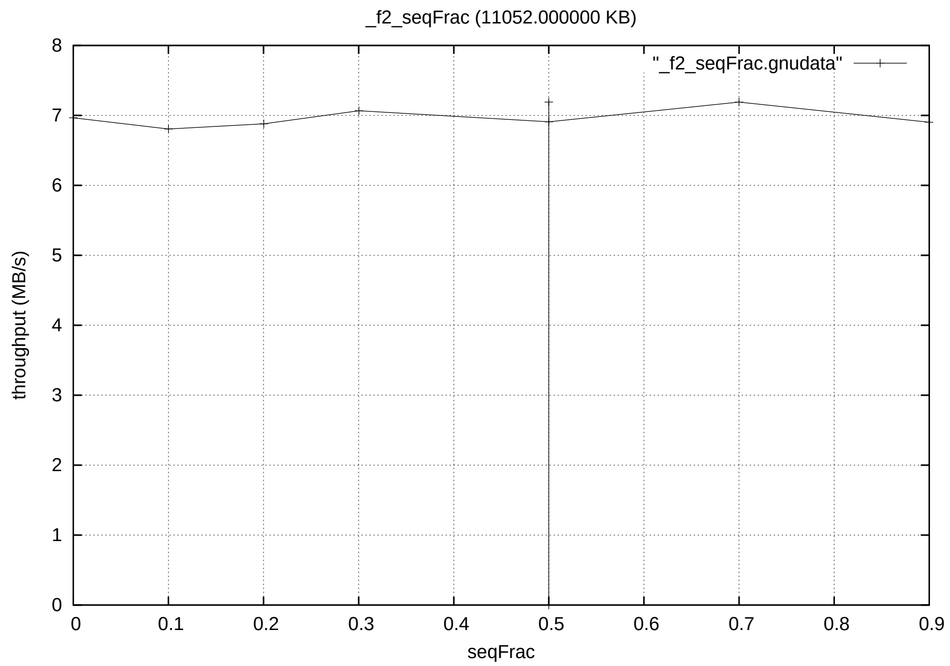


**Figure 1.3**

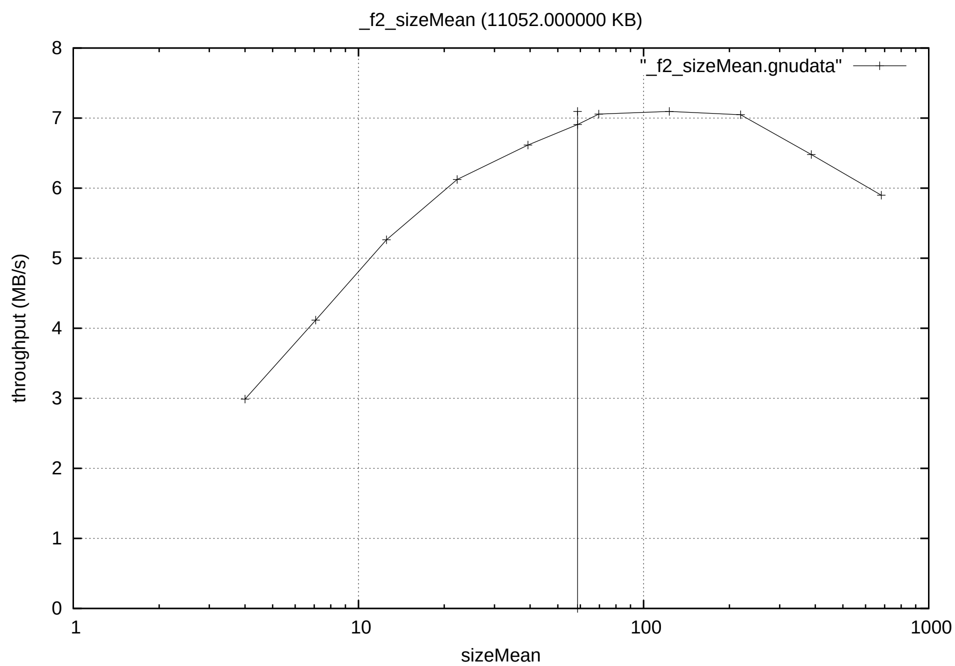


**Figure 1.4**

The plot of processNum indicates that only 7 to 9 processes were actively generating workload. The readFrac plot tells us that throughput performance for large reads is slightly better than that for writes, this is consistent with disk drives using read lookahead caching, but no write caching, on a System V host which employs a writeback cache strategy (periodic fsyncs). The seqFrac plot indicates that sequential operations offer similar throughput to random accesses. The sizeMean plot clearly shows that increasing transfer size improves measured throughput to a point, beyond which overheads diminish performance.

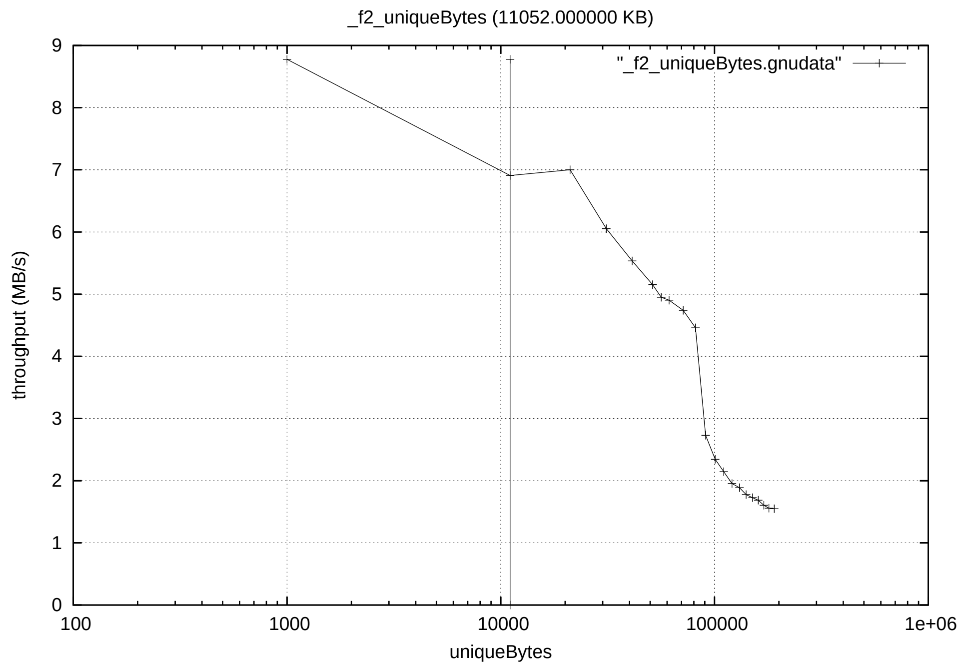


**Figure 1.5**

The most revealing plot however is that of uniqueBytes, a measure of the data size transfered. This plot allows us to determine both raw filesystem/hardware performance, as well as cache performance. For small values of uniqueBytes, throughput performance is dominated by the cache in main memory. With increasing uniqueBytes, the cache hit ratio declines and hardware performance comes into play. This is particularly evident for values between 10,000 and 20,000, where indeed the characteristic is almost flat at about 7 MBytes/s. Beyond this point, the performance of the hardware dominates. At values of between 60,000 and 80,000 , the measured performance corresponds to the raw bandwidth of the disk heads. Beyond this point, performance declines due the effects of filesystem behaviour. Performance in this area is primarily limited by the need to look up indirect block pointers from the inode, thus causing frequent multiple block accesses per I/O.

## 6. Test Results - Challenge L

The superior compute performance and memory bus bandwidth of the Challenge L is clearly evident from the plots, with better than 42 MBytes/s achieved for small I/O operations hitting cache (uniqueBytes).

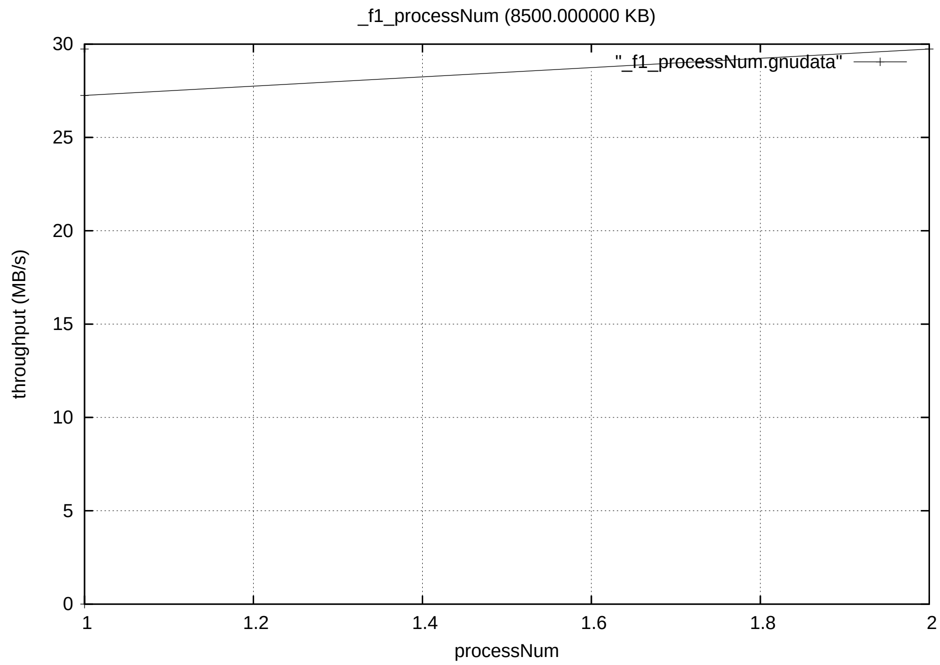


**Figure 2.1**

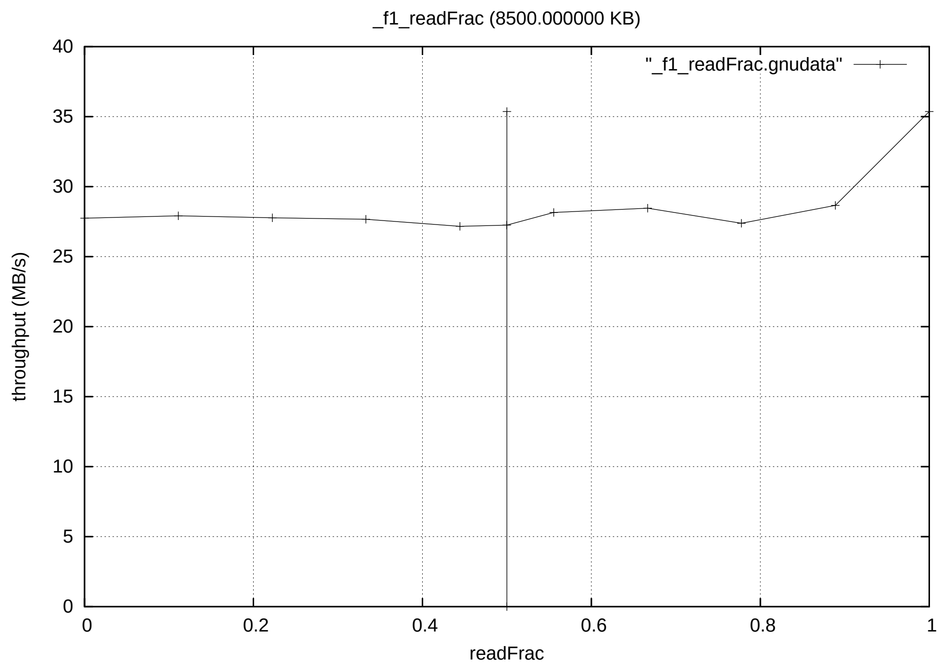


**Figure 2.2**

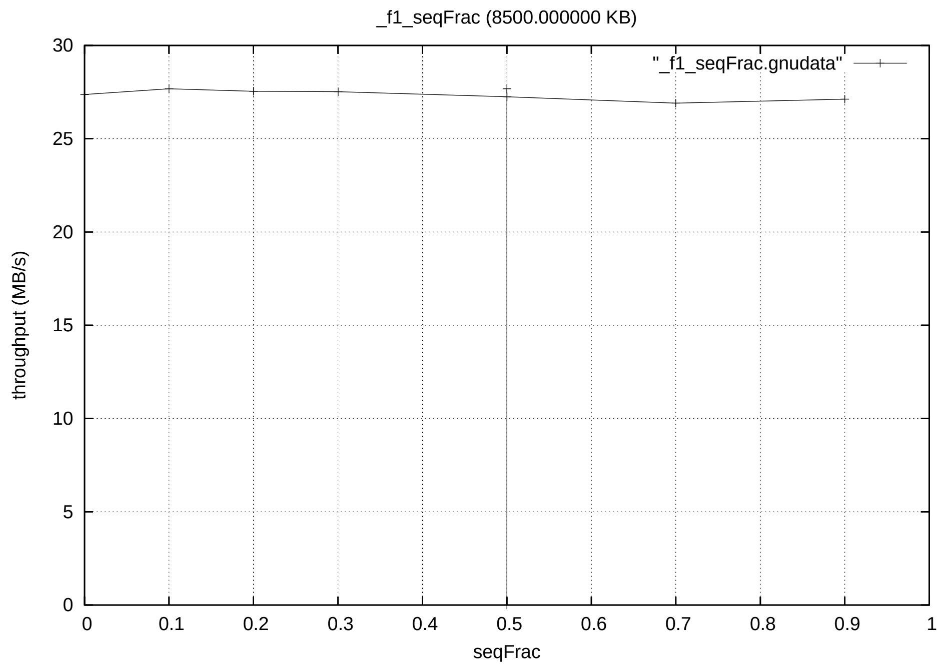


**Figure 2.3**

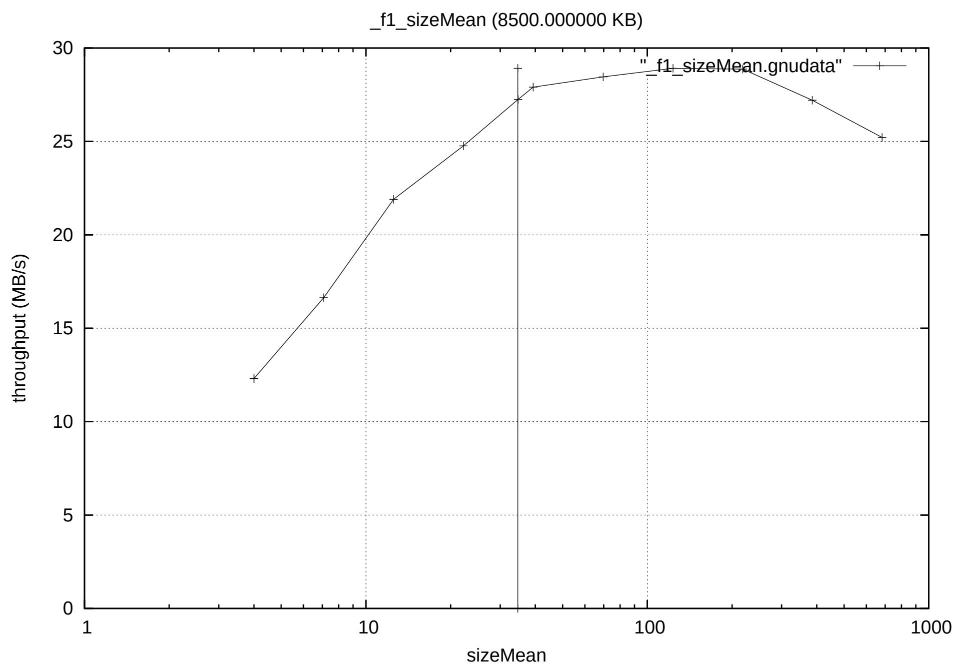


**Figure 2.4**

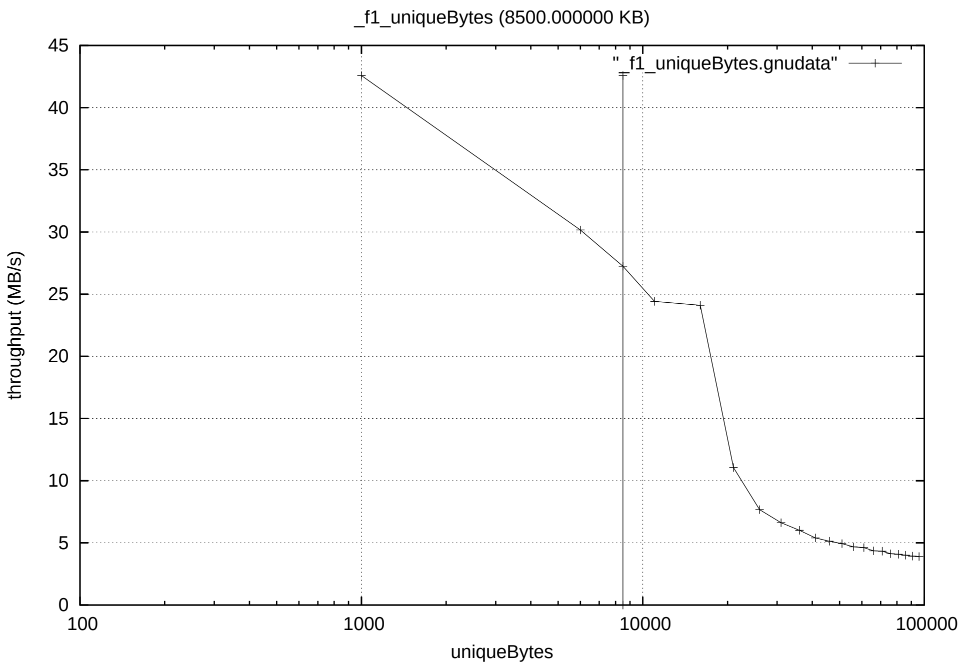


**Figure 2.5**

The plot of processNum indicates that between 27 to 29 processes were actively generating workload, a significantly higher workload than carried by the Iris. The readFrac plot tells us that throughput performance for large reads is slightly better than that for writes, this is again consistent with disk drives using read lookahead caching, but no write caching, on a System V host which employs a writeback cache strategy (periodic fsyncs). The seqFrac plot indicates that sequential operations offer similar throughput to random accesses, no significant differences were observed. The sizeMean plot again clearly shows that increasing transfer size improves measured throughput to a point, beyond which overheads diminish performance.

## 7. Comparative Analysis - Iris 4D/240S vs Challenge L

Comparative analysis between the Iris and Challenge is concentrated on throughput vs uniqueBytes, as this provides a good indication of relative cache performance. The general form of the characteristics are very similar, and the test compares performance with the 2 GB Fujitsu M2654HA disks.

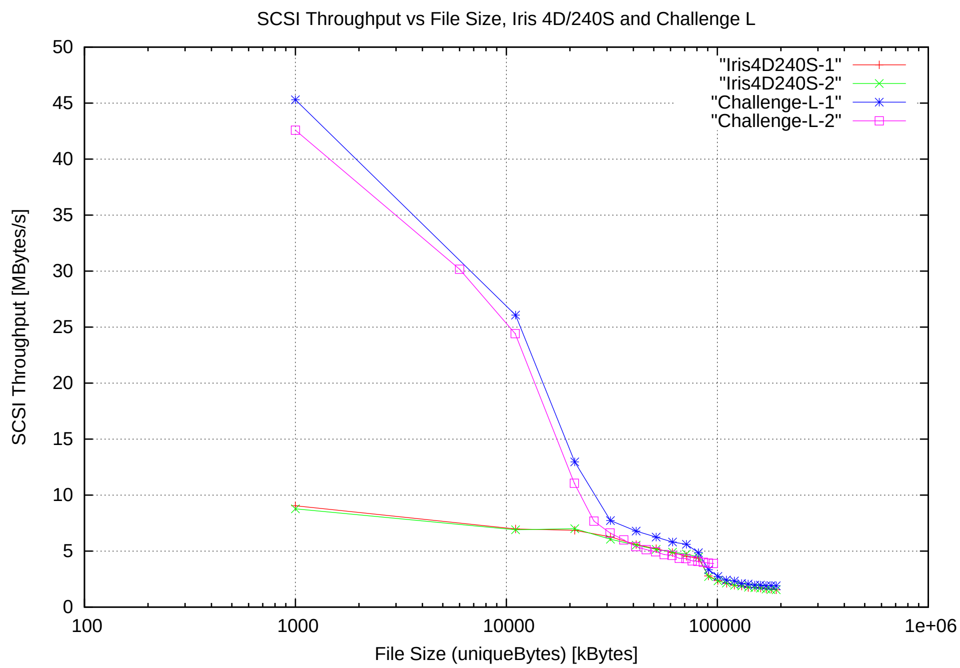


**Figure 3**

Figure 3 compares these characteristics between the two systems. Significantly, for small I/O operations the Challenge provides about five times the I/O throughput performance, which is also sustained over a larger operating envelope of I/O operation sizes.

## 8. Comparative Analysis - Challenge L with Fujitsu and Seagate Drives

Figure 4 compares the relative performance of a number of configurations of the Challenge L. The system was tested with 128 MB of memory, using the older Fujitsu M2654HA drive, as well as a state of the art Seagate Elite 9 GB drive. The test run for the Elite was subsequently repeated with 256 MB of main memory. Comparing the performance of the 128 MB system with the two drive types, it is quite evident that the newer drive delivers slightly better throughput performance, although not dramatically better.

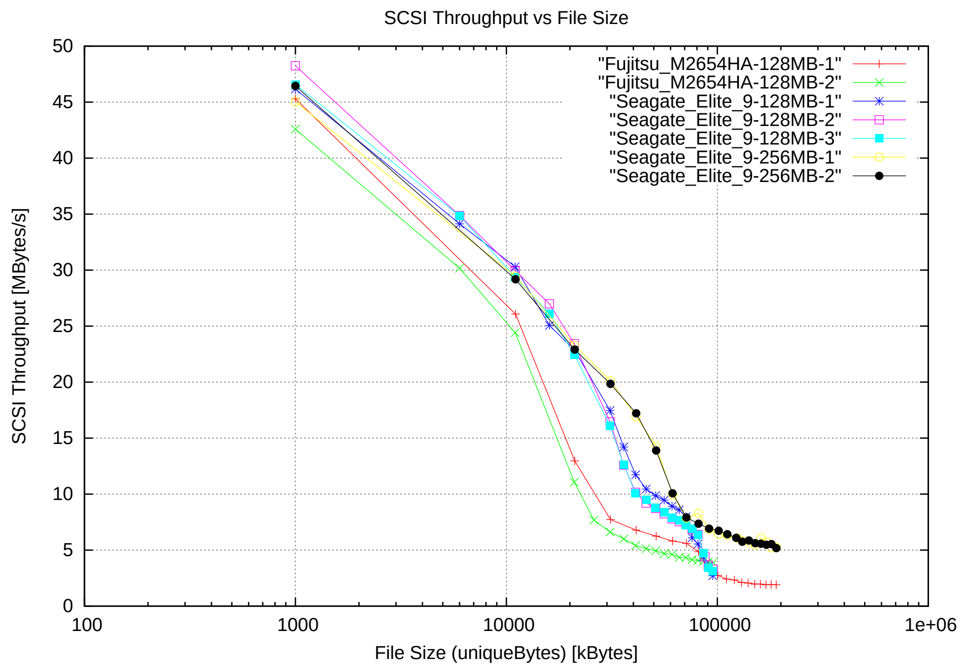


**Figure 4**

The outcome of the test was to retain the four 2 GB M2654HA Fujitsu drives rather than use the single replacement 9 GB Seagate drive, as the gain in performance did not justify the replacement cost of the drives. Comparing the 128 MB and 256 MB configurations yielded a similar outcome. The effect of the added memory was to delay the onset of the point at which the cache became ineffective, ie the performance gain did not justify the additional memory, which was subsequently not acquired.

## 9. Conclusions

The use of the Self Scaling I/O Benchmark provided an effective means of validating the customer's decision to acquire the Challenge L as a replacement server for the Iris 4D/240S.

Test results demonstrated that the Challenge will provide between 3.5 and 5 times the I/O performance of the Iris 4D/240S. Significantly, the achieved peak throughput for small I/O operations is 5 times better in the Challenge L, largely as a result of superior caching performance.