UCRL-ID-141831

The Raw Disk I/O Performance of Compaq Storage Works RAID Arrays under Tru64 UNIX

A.C. Uselton

October 19, 2000

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

DISCLAIMER

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

The Raw Disk I/O Performance of Compaq StorageWorks RAID arrays under Tru64 UNIX

Andrew C. Uselton

October 19, 2000

Abstract

We report on the raw disk i/o performance of a set of Compaq StorageWorks RAID arrays connected to our cluster of Compaq ES40 computers via Fibre Channel. The best cumulative peak sustained data rate is 117MB/s per node for reads and 77MB/s per node for writes. This value occurs for a configuration in which a node has two Fibre Channel interfaces to a switch, which in turn has two connections to each of two Compaq StorageWorks RAID arrays. Each RAID array has two HS-G80 RAID controllers controlling (together) two 5+p RAID chains. A 10% more space efficient arrangement using a single 11+p RAID chain in place of the two 5+P chains is 25% slower for reads and 40% slower for writes.

1 Introduction

This report presents the data transfer performance of the mass storage subsystem for a high performance computing system. All of the tests reported use unbuffered character data tranfers, or so-called *raw i/o*. The tests include several configurations of the storage subsytem in order to do the following: determine the optimum configuration, establish a baseline for further testing in conjunction with various file systems, and provide independent confirmation of the vendor's performance claims.

"I/o" is the generic term for input data transfer operations, or reads, and output data transfer operations, or writes. This report details values for the rate, in Megabytes per second (MB/s), at which data may be read from and written to the storage subsytem, and refers to such a value as a *data rate*. There are other measures for performance, including *latency*, *reliablility* or *stability*[1], *cpu utilization*, *price* and *capacity*, but this report confines itself to data rate. Each i/o reads or writes a *block* of data whose size is measured in bytes. An i/o performance test consists of measuring the duration of one or more i/o operations of a given *block size*. The data rate varies depending on the configuration

[1]Both the "mean time between failures" and the ability to continue in the presence of errors.

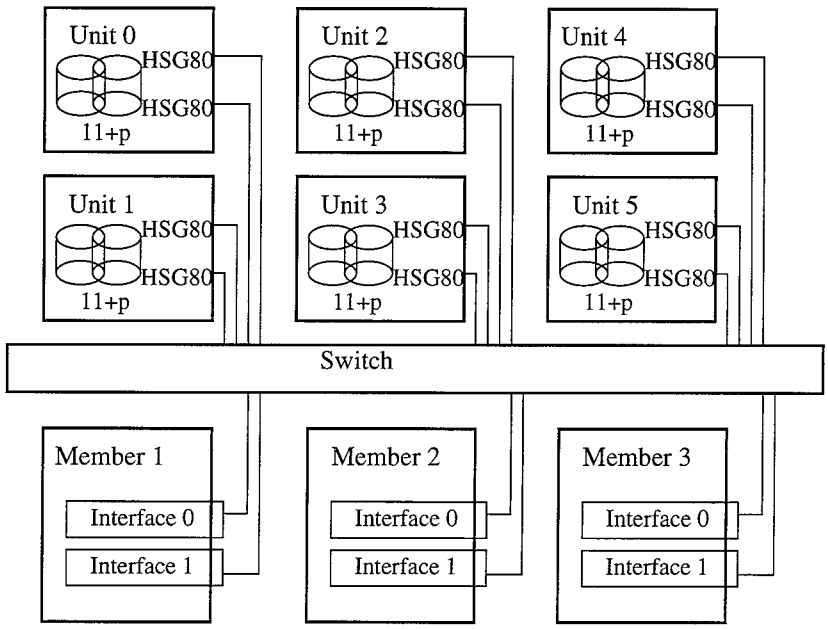

Figure 1: Test environment for clusters A and B

of the storage subsystem and depending on the details of the i/o. The data rate assymptotically descends to a stable value as the total amount read or written becomes very large. The first tests[2] varied the total amount of the i/o in order to find the asymptotic value of data rate (the *sustained data rate*). In each case testing with the total amount of i/o four times the size of memory gives a good approximation of the sustained data rate. The reported tests vary the block size and show that the data rate improves with block size to a maximum for block sizes of 512KB or larger.

The computer system being tested consists of 128 nodes connected via a high performance network interconnect. Each node is a Compaq ES40, which is an *Alpha*-based 4-way Symetric Multiprocessor (SMP) computer. The nodes are grouped into four clusters of 32 nodes each. The clusters are named *cluster A*, *cluster B*, *cluster C*, and *cluster D*. In each cluster each of the first three nodes, *member 1*, *member 2*, and *member 3*, is connected to a set of Compaq StorageWorks RAID arrays and is referred to as an *i/o node*. The clusters are running Compaq's Sierra cluster software based on the Tru64 UNIX operaing system. In each cluster members 1 and 2 each has 8 Gigabytes (GB) of memory while members 3 through 32 each has 2 GB of memory.

Figure 1 shows the layout of the i/o nodes and the RAID arrays for clusters A and B. The connection between i/o nodes and RAID arrays is via *Fibre Channel*, which can transfer $100MB$ per second. The figure shows each of the i/o nodes with two Fibre Channel interfaces connecting to a Fibre Channel switch. The switch is connected via Fibre Channel to each of two HSG80 RAID controllers on each RAID array. There are six RAID arrays in each cluster, and they are

[2]Only reported in the appendix.

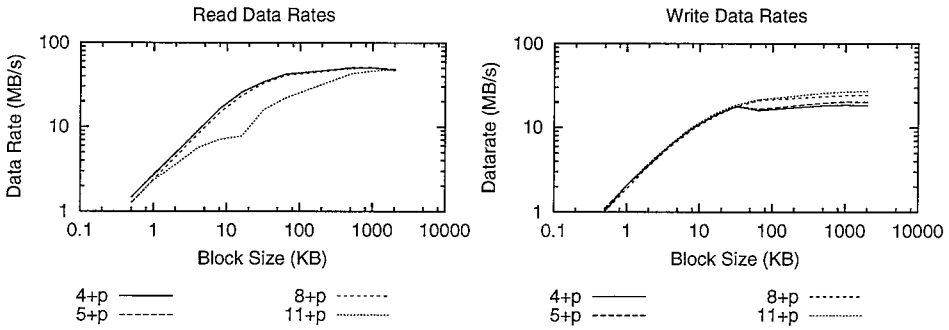

Figure 2: Read and Write data rates for single unit i/o

labelled *Unit 0* through *Unit 5*. In Figure 1 each RAID array has a set of twelve SCSI disks in a RAID-5 configuration with one disk acting as a *parity disk*. We refer to this as an 11+p RAID chain. The only change in clusters C and D is that each RAID array is configured to have two 5+p RAID chains in place of one 11+p RAID chain.

The remainder of this report consists of four sections each of which details the results of a particular group of tests. In Section 2 the data rate varies with the size of the RAID chain. In Section 3 the data rate of two 5+p RAID chains is 60% better for reads over a single 11+p RAID chain and 40% better for writes at the expense of 10% of the available storage space. In Section 4 the data rate when using two RAID arrays via a single (member) interface is about double that of one, but using three RAID arrays increases the data rate very little. Similarly, in Section 5 using two interfaces (to access two or more RAID arrays) about doubles the data rate. This report concludes with a summary of the results, a reccomendation for the storage subsystem's configuration, and some suggestions for further research. The author reccommends that each RAID array be configured with two equal sized RAID chains, each as large as the RAID array can contain. In such a configuration each node should be able to sustain at least a 117MB/s read data rate and a 77MB/s write data rate. An appendix discusses methodology.

2 The size of the RAID chain

Figure 2 graphs the read and write data rates against block size for four configurations of RAID chain: 4+p, 5+p, 8+p, and 11+p. Each test was conducted

on member 2 of one of the clusters, and the i/o was directed through a single Fibre Channel interface to a single RAID chain in a single RAID array. In each case the node has 2GB of memory and the sustained data rate was measured at a value for total i/o of 8GB. Each curve represents a set of tests each conducted with block size set to one of $512B$, $1024B$ $(= 1KB)$, $2KB$, $4KB$, $8KB$, $16KB$, $32KB$, $64KB$, $512KB$, $1024KB$ $(= 1MB)$, and $2MB$.

The read data rate curves for 4+p, 5+p, and 8+p, are nearly identical and show two prominent features. First, when the block size is $512KB$ or larger the read data rate is in the vicinity of $50MB/s$ which appears to be the peak sustained read data rate a single RAID chain can produce regardles of the number of disks in the chain. Second, for block sizes $16KB$ and smaller (a log plot of) the data rate is linear with block size. The later point deserves some elaboration.

If data rate is linear in block size then the time for an i/o operation is constant, independent of block size. The explanation is as follows: Let r represent the data rate, b the block size, n the number of blocks read, t the total amount read, and s the time it took to read t. Then

$$r = t/s = b * n/s \tag{1}$$

If r is a linear function of b then n/s is constant, which is to say that the number of blocks read per second does not change as b changes. If the data rate were linear in block size then the slope of the curve in the log plot would be exactly one:

$$r = mb \rightarrow \log(r) = \log(mb) = \log(m) + \log(b) \tag{2}$$

In fact the slope is approximately 1.1:

$$1.1 \log(b) = log(b^{1.1}) \tag{3}$$

Which leaves us to wonder why $n/s \sim b^{0.1}$. In any case the RAID chain can produce at most $50MB/s$ and no more than 3000blocks/s at best.

The 11+p configuration of RAID chain has generally similar limitting behaviors, but shows a remarkable drop in data rate for medium sized blocks. It looks like the sum of two distinct asymptotic behaviors. One may wonder what is happenning that does not happen for the other RAID chains.

The write data rate curves are similar for all four configurations of RAID chain. For block sizes of $32KB$ and less the behaviors are identical, with an assymptotic increase of data rate to a maximum near $20MB/s$. Above $32KB$ the 11+p curve continues smoothly to the highest asymptotic value of $27MB/s$. For large block sizes the smaller RAID chains show smaller peak values. On may speculate that the 11+p write data rate curve is a limitting case for StorageWorks RAID chains.

3 The number of chains in a RAID array

Figure 3 again shows the read and write data rates for the 11+p RAID chain as well as an arrangement in which the same twelve disks are configured in

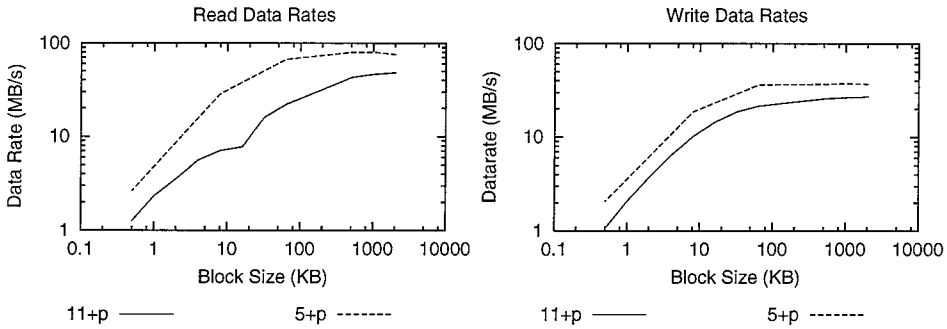

Figure 3: Twelve disks configured as one 11+p or two 5+p chains

the RAID array as two separate RAID chains. The benefit of doing so is that both HSG80s can be active at once servicing i/o operations. The result is a 60% increase in read data rate and a 36% increase in write data rate. In fact the read data rate is near the theoretical maximum for SCSI transfers. The write rate for the two 5+p chains is nearly double the write rate for one, so we may speculate that two larger RAID chains might perform even better. The advantage of the 11+p RAID chain is that it has 10% more space than the two 5+p chains.

4 The number of RAID arrays

In Figures 4 and 5 the graphs show the data rate for a single node using a single Fibre Channel interface reading from and writing to one, two, and three 11+p chains and (respectively) two, four, and six 5+p chains in one, two, and three units.

The read and write data rates for the 11+p chains improve when writing to two units as does the write data rate for the 5+p chains. Once the read data rate from a node reaches around $80MB/s$ in cannot be improved upon by adding more units to the node's interface. Similarly, a node seems to be unable to write more than $50MB/s$ through a single interface no matter how many units it can write to. It is clear that if performance is an issue then a single interface should not be given more than two units for its i/o resposibilities.

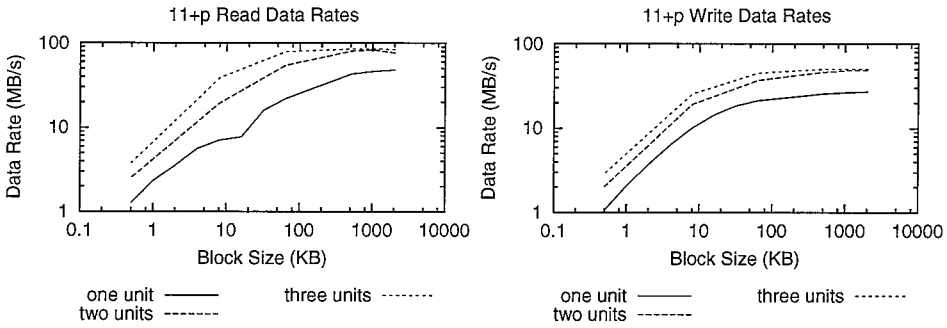

Figure 4: I/o to multiple 11+p units via a single interface

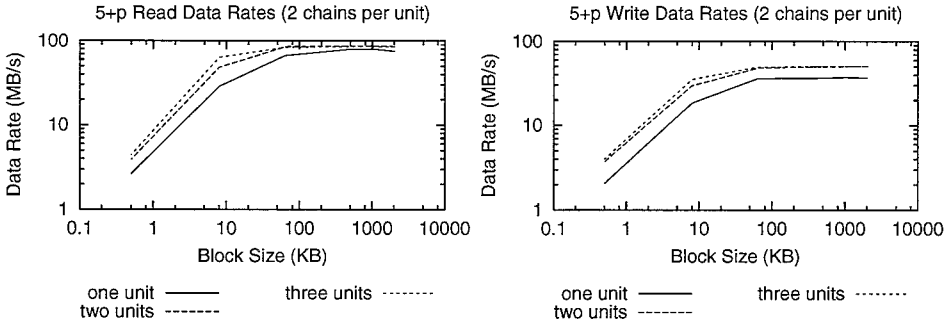

Figure 5: I/o to multiple 5+p units (two chains each) via a single inteface

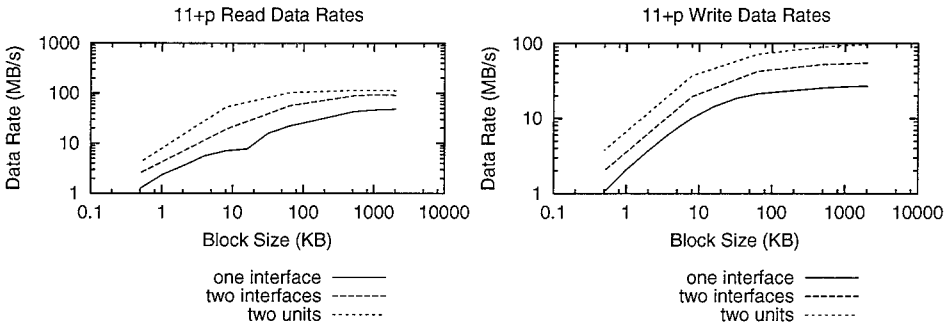

Figure 6: I/o to two 11+p units via two interfaces

5 The number of interfaces

Figures 6 and 7 show the preformance gain from using two interfaces on a single node. In each graph there is the curve already presented in which a single interface communicates with a single unit (either configured with an 11+p chian as in Figure 6 or two 5+p chains as in Figure 7). Next, each graph shows a curve for the read and write data rates when employing two interfaces. And finally, there is a curve for the performance of i/o to two units on each of two interfaces.

The read and write data rates for using two 11+p RAID chains on two interfaces double those for a single unit, as does the write data rate for the 5+p chains double. The read data rate increases by about 50% to $115MB/s$. For both the 11+p and the 5+p configurations the peak read data rate is $114MB/s$ and the peak write data rate near $100MB/s$. Tests with three units on each interface show no improvement. We may speculate that a third interface would not increase the read data rate, since it is not increasing with the addition of a second or third unit to the interface whereas a second unit did improve the performance when only one interface was employed. On the other hand the write data rate for two interfaces always doubled the data rate for the same configuration with one interface, so a third interface might still improve the write performance.

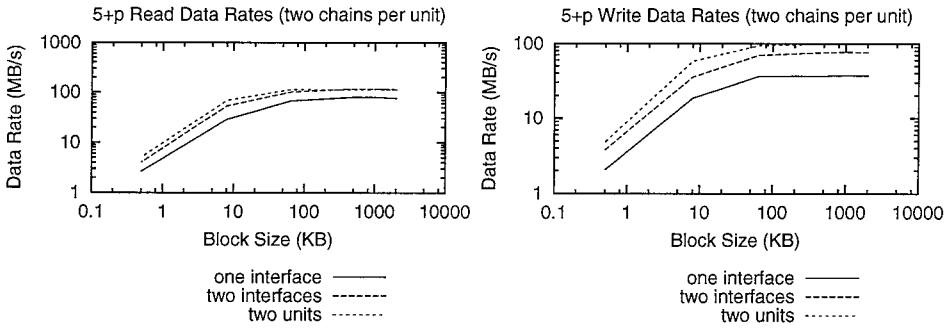

Figure 7: I/o to two 5+p units (two chains per unit) via two intefaces

6 Conclusion

Current plans for this computer system are for it to have three i/o nodes communicating with six StorageWorks units, and for the entire storage subsystem to be gathered into a single large parallel file system. If each of the i/o nodes is configured with one unit on each interface and if each unit is configured with two 5+p RAID chains then the cumulative peak sustained data rate for the storage subsystem is (in theory) $351MB/s$ for reads and $231MB/s$ for writes. This is the configuration that the author recommends. A configuration using 11+p RAID chains could achieve $282MB/s$ for reads and $165MB/s$ for writes, and would have 10% more file space. An altered arrangement (with more units) in which two units were on each interface and only two i/o nodes were used could achieve $228MB/s$ for reads and $200MB/s$ for writes.

This report leaves a few questions unanswered. Some questions will be addressed in a second report in which file systems are used for the i/o rather than raw i/o. It would be interesting to determine what causes the dip in read data rate for 11+p RAID chains. The StorageWorks RAID arrays have room for a total of 24 SCSI disks, so there might be an improvement in data rate for a configuration with two RAID chains of 11+p each in a single unit. All three of the i/o nodes can see all six of the StorageWorks units, so a test with all nodes writing to one unit would better determine the maximum performance of a single unit.

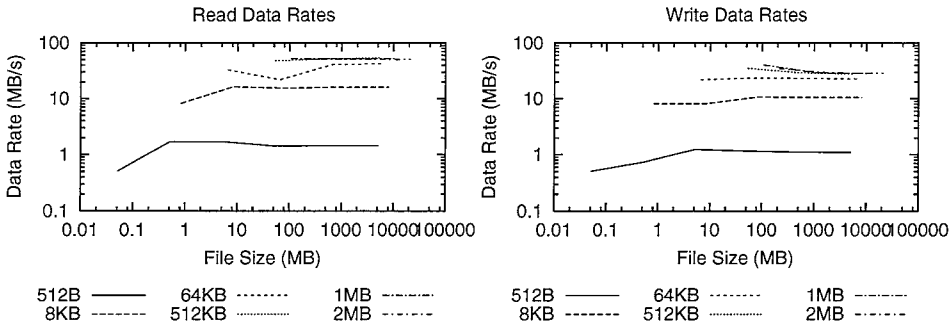

Figure 8: Data rate asymptotes for large files

A Methodology

All of the tests in this report followed the same basic scheme. In a UNIX shell the *dd* (disk duplicate) command copies from one location and to another. The *dd* command is invoked within a *time* command. The result is an estimate, acurate to about 1/10 second of the time necesary to perform the copy.

The *dd* command requires both in input file and an output file. When measuring read data rate one wants to be certain that the time spent writing is negligable. Conversely, when measuring write data rate one wants the time spent reading to be negligable. The "devices" */dev/zero* and */dev/null* perform this function. A degenerate case in which *dd* both reads from */dev/zero* and writes to */dev/null* has a data rate one to two orders of magnitude higher than when accessing an actual device. Thus a command like

```
time dd if=/dev/rdisk/dsk10c of=/dev/null
```

is a fair test of the read performance of *dsk10c*.

While all the tests in this report use raw disks with no buffering, there is always the possibility that some buffering is taking place in the RAID array itself. Figure 8 shows the data rate for performing the i/o tests with successively larger files and with various block sizes. The asymptotic behavior above one or two GB written converges to a value characteristic for the block size.

For the smallest files represented in Figure 8 the total amount of time spent reading or writing was comparable with the 1/10 second resolution of the *time* command. All of the tests shown in the main body of this report ran for more than 60 seconds, so the resolution of the *time* command was not a problem.

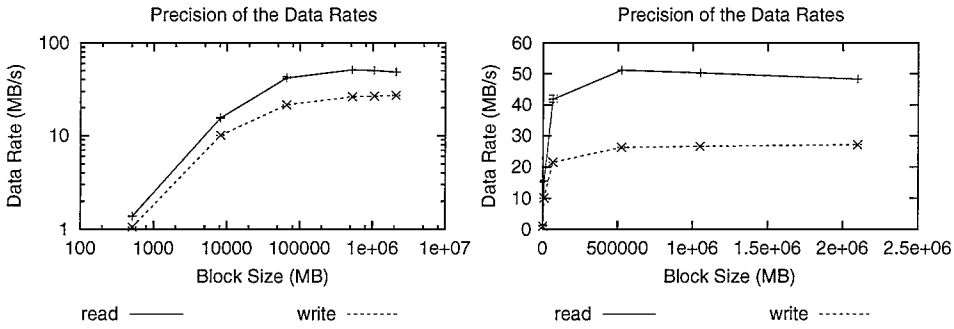

Figure 9: The measuments are precise to within 5%.

Error could enter into the observations from other sources as well, so one may wonder what the precision of the measurments is. Figure 9 shows the results of running tests for each data point five times and putting error bars around the extreme values. In every case the difference *delta* between the highest observed value and the lowest is less than 5% of the magnitude of the measurement. The log plot makes the error bars difficult to see so the same graph is shown without log scales on the right. Even here the error bars are so small they are difficult to see on the graph.

The foregoing characterizes the *precision* of the measurment process. That is, repeated tests are close together - within 5%. An alternative means of measuring the same values is available in the *Ronnie* benchmark. Figure 10 shows the results of running *Ronnie* on a $4+p$ configured RAID array and writing to a single chain with 1MB blocks. The data rate for running one *Ronnie* job is $15MB/s$, around 20% less than the $18MB/s$ computed using *time* and *dd*. The accuracy of the methods presented should be taken as no better than 20%. Note that adding a second parallel *Ronnie* job doubles the data rate, but additional jobs reduce the data rate back to arond $24MB/s$.

Also shown in Figure 10 are the values for CPU utilization while running *Ronnie*. A single *Ronnie* job consumes about 0.2% of the capacity of a node. This is in keeping with the values *top* reported when running a test with *dd*. That is, none of the tests reported here put much load on the nodes' CPUs.

B Sample Scripts

Here is a typical script for invoking *time* and *dd*.

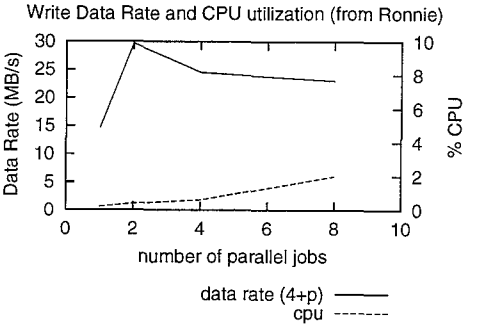

Figure 10: The measurements are accurate within about 20%.

```
#!/bin/sh
# iotest script for testing raw device data rates

node='uname -n'
echo $0 $* on $node >iotest${node}-${1}-${2}-${3}.log

if [  "X$3" = "X" ] ; then
        echo "usage: $0 <i> <u> <c>"
echo "       <i> number of interfaces, 1 or 2"
echo "       <u> number of units, 1, 2, or 3"
echo "       <i> number of chains, 1 or 2"
        exit 1
fi

nintf=$1
nunit=$2
nchain=$3

case $nintf in
1 )
interfaces="int0"
;;
2 )
interfaces="int0 int1"
;;
* )
echo "The first argument should be 1 or 2, not "$nintf"."
exit 0
;;
esac

case $nunit in
1 )
units="unit0"
;;
2 )
units="unit0 unit1"
;;
3 )
units="unit0 unit1 unit2"
;;
* )
echo "The second argument should be 1, 2,  or 3, not "$nunit"."
exit 0
;;
esac

case $nchain in
1 )
chains="chain0"
```

```
;;
2 )
chains="chain0 chain1"
;;
* )
echo "The third argument should be 1 or 2, not "$nchain"."
exit 0
;;
esac

disks=`for intf in $interfaces
do
for unit in $units
do
for chain in $chains
do
grep $node cluster.conf | grep $intf | \
          grep $unit | grep $chain | awk '{print $5}'
done
done
done`


echo From $node test $disks >>iotest${node}-${1}-${2}-${3}.log

for disk in $disks
do
    if [ -e /dev/rdisk/${disk}c ] && [ ! -f /dev/rdisk/${disk}c ]
    then
echo "/dev/rdisk/${disk}c does appear to be a character device"
/sbin/disklabel -z /dev/rdisk/${disk}c 2>/dev/null
    else
echo "/dev/rdisk/${disk}c doesn't appear to be a character device"
exit 0
    fi
done

while read reps blocksize
do
  echo "read:" $blocksize $reps $disks `date` \
                  >>iotest${node}-${1}-${2}-${3}.log
  for disk in $disks
  do
    rawdisk="/dev/rdisk/${disk}c"
    time dd if=$rawdisk of=/dev/null bs=$blocksize count=$reps \
                              >read${node}${disk}.tmp 2>&1 &
  done
  wait
  echo "write:" $blocksize $reps `date` \
                      >>iotest${node}-${1}-${2}-${3}.log
  for disk in $disks
  do
    rawdisk="/dev/rdisk/${disk}c"
    time dd of=$rawdisk if=/dev/zero bs=$blocksize count=$reps
                      >write${node}${disk}.tmp 2>&1 &
  done
  wait
  echo "end:" $blocksize $reps `date` \
                      >>iotest${node}-${1}-${2}-${3}.log
  echo -n $reps $blocksize
  for disk in $disks
  do
    timeval=`egrep "^real" read${node}${disk}.tmp | \
                                        awk '{print $2}'`
    echo -n " $disk:$timeval "
    echo read${node}${disk}.tmp  \
                  >>iotest${node}-${1}-${2}-${3}.log
    cat read${node}${disk}.tmp \
                  >>iotest${node}-${1}-${2}-${3}.log
  done
  for disk in $disks
  do
    timeval=`egrep "^real" write${node}${disk}.tmp \
                                       | awk '{print $2}'`
    echo -n " $disk:$timeval "
    echo write${node}${disk}.tmp  \
                >>iotest${node}-${1}-${2}-${3}.log
    cat write${node}${disk}.tmp \
                >>iotest${node}-${1}-${2}-${3}.log
  done
  echo

done<iotest.conf
# End of iotest script
```

The script uses *sh* which relies on */usr/bin/time* rather than a built in function. On the other hand, *wait* is a built in and is used here, since */usr/bin/wait*

does not behave as one might expect[3].

The script produces a series of times, which a separate filter then turns into data rates in files with the ".data" extension. The following typical *gnuplot* script generates a graph to be included in this paper (".eps") or on a web page (".gif"). This one generates the graph on the left in Figure 2.

```
set terminal postscript eps
set size 0.5,0.5
set key below
set logscale
set xlabel "Block Size (KB)"
set ylabel "Data Rate (MB/s)"
set title "Read Data Rates"
set output "eps/read-1-1-1.gnuplot.eps"
plot 'raw.data/iotest4+p-1-1-1.data' \
                        using 2:4 title "4+p" with lines, \
     'raw.data/iotest5+p-1-1-1.data' \
                        using 2:4 title "5+p" with lines, \
     'raw.data/iotest8+p-1-1-1.data' \
                        using 2:4 title "8+p" with lines, \
     'raw.data/iotest11+p-1-1-1.data' \
                        using 2:4 title "11+p" with lines
set terminal gif size 640,480
set size 1,1
set output "gif/read-1-1-1.gnuplot.gif"
plot 'raw.data/iotest4+p-1-1-1.data' \
                        using 2:4 title "4+p" with lines, \
     'raw.data/iotest5+p-1-1-1.data' \
                        using 2:4 title "5+p" with lines, \
     'raw.data/iotest8+p-1-1-1.data' \
                        using 2:4 title "8+p" with lines, \
     'raw.data/iotest11+p-1-1-1.data' \
                         using 2:4 title "11+p" with lines
```

C Data

The tables in Figures 11 through 27 show the contents of the".data" files from which all of the graphs were constructed.

[3]It creates a subprocess that waits only for itself, thus it return immediately

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
4096	2048	8192	47.1	18.2
8192	1024	8192	49.9	18.6
16384	512	8192	49.9	18.3
131072	64	8192	42.4	15.9
262144	32	8192	34.3	17.8
524288	16	8192	25.9	13.8
1048576	8	8192	16.4	9.8
2097152	4	8192	9.0	6.2
4194304	2	8192	4.9	3.6
8388608	1	8192	2.7	2.1
16777216	0.5	8192	1.5	1.0

Figure 11: A single 4+p RAID chain

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
4096	2048	8192	48.1	20.0
8192	1024	8192	50.2	20.2
16384	512	8192	50.4	19.5
131072	64	8192	42.2	16.4
262144	32	8192	34.3	17.9
524288	16	8192	25.5	13.9
1048576	8	8192	16.3	9.8
2097152	4	8192	8.9	6.2
4194304	2	8192	4.9	3.7
8388608	1	8192	2.7	2.1
16777216	1/2	8192	1.5	1.1

Figure 12: A single 5+p RAID chain

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
4096	2048	8192	47.0	24.4
8192	1024	8192	50.6	24.1
16384	512	8192	49.6	23.4
131072	64	8192	41.0	21.0
262144	32	8192	33.1	17.6
524288	16	8192	23.6	13.4
1048576	8	8192	14.7	9.4
2097152	4	8192	8.2	6.0
4194304	2	8192	4.4	3.5
8388608	1	8192	2.4	1.9
16777216	1/2	8192	1.3	1.0

Figure 13: A single 8+p RAID chain

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
4096	2048	8192	48.2	27.0
8192	1024	8192	46.3	26.5
16384	512	8192	43.0	25.7
131072	64	8192	22.0	21.4
262144	32	8192	15.9	18.6
524288	16	8192	7.8	14.4
1048576	8	8192	7.1	10.1
2097152	4	8192	5.6	6.4
4194304	2	8192	3.6	3.7
8388608	1	8192	2.3	2.1
16777216	1/2	8192	1.3	1.1

Figure 14: A single 11+p RAID chain

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	75.3	36.8
10000	1024	10000	79.9	37.4
20000	512	10000	79.8	36.7
150000	64	9375	67.0	36.2
600000	8	4688	28.7	18.5
1000000	1/2	488	2.7	2.1

Figure 15: Two 5+p RAID chains in one RAID array

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	76.2	49.1
10000	1024	10000	83.7	48.2
20000	512	10000	80.7	46.2
150000	64	9375	54.3	37.0
600000	8	4688	19.5	19.3
1000000	1/2	488	2.5	2.0

Figure 16: Two 11+p RAID chains in two RAID arrays via one interface

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	84.4	50.5
10000	1024	10000	85.7	50.3
20000	512	10000	85.4	50.1
150000	64	9375	78.8	45.1
600000	8	4688	38.6	25.6
1000000	1/2	488	3.8	2.9

Figure 17: Three 11+p RAID chains in three RAID arrays via one interface

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	85.6	50.5
10000	1024	10000	86.8	50.3
20000	512	10000	86.9	50.3
150000	64	9375	85.0	48.2
600000	8	4688	48.3	29.8
1000000	1/2	488	3.9	3.7

Figure 18: Two 5+p RAID chains in each of two RAID arrays via one interface

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	85.6	50.6
10000	1024	10000	85.6	50.5
20000	512	10000	85.4	50.4
150000	64	9375	83.7	49.6
600000	8	4688	63.6	35.2
1000000	1/2	488	4.4	4.0

Figure 19: Two 5+p RAID chains in each of three RAID arrays via one interface

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	91.4	55.1
10000	1024	10000	94.0	53.8
20000	512	10000	89.4	52.6
150000	64	9375	56.5	42.5
600000	8	4688	18.9	19.6
1000000	1/2	488	2.6	2.0

Figure 20: One 11+p RAID chain in each of two RAID arrays via two interfaces

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	112.7	95.6
10000	1024	10000	113.8	93.9
20000	512	10000	114.0	90.8
150000	64	9375	103.5	71.9
600000	8	4688	52.9	37.1
1000000	1/2	488	4.2	3.8

Figure 21: One 11+p RAID chain in each of four RAID arrays via two interfaces

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	118.5	89.2
10000	1024	10000	118.6	88.5
20000	512	10000	118.5	87.6
150000	64	9375	112.0	80.3
600000	8	4688	68.7	47.3
1000000	1/2	488	4.4	4.0

Figure 22: One 11+p RAID chain in each of six RAID arrays via two interfaces

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	114.2	76.0
10000	1024	10000	116.7	76.7
20000	512	10000	117.6	75.9
150000	64	9375	101.2	69.4
600000	8	4688	53.1	35.6
1000000	0	488	4.0	3.8

Figure 23: Two 5+p RAID chains in each of two RAID arrays via two interfaces

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	114.4	99.8
10000	1024	10000	114.5	99.3
20000	512	10000	114.4	99.1
150000	64	9375	113.6	94.2
600000	8	4688	68.5	57.8
1000000	1/2	488	5.1	4.8

Figure 24: Two 5+p RAID chains in each of four RAID arrays via two interfaces

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
5000	2048	10000	114.7	99.8
10000	1024	10000	114.8	99.5
20000	512	10000	114.9	99.4
150000	64	9375	115.2	96.4
600000	8	4688	77.0	68.1
1000000	1/2	488	5.6	5.3

Figure 25: Two 5+p RAID chains in each of six RAID arrays via two interfaces

count	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)
100	512	0.05	0.5	0.5
1000	512	0.5	1.7	0.7
10000	512	5	1.7	1.2
100000	512	51	1.4	1.2
1000000	512	512	1.4	1.1
10000000	512	5120	1.4	1.1
100	8192	0.8	8.2	8.1
1000	8192	8	16.3	8.1
10000	8192	82	15.4	10.7
100000	8192	819	16.1	10.5
1000000	8192	8192	16.1	10.6
100	65536	7	32.7	21.8
1000	65536	66	21.8	23.4
10000	65536	655	40.7	23.0
100000	65536	6553	42.6	22.5
100	524288	52	47.6	34.9
1000	524288	524	51.9	29.2
10000	524288	5242	53.7	27.7
100	1048576	105	52.4	40.3
1000	1048576	1048	51.6	29.2
10000	1048576	10485	52.7	28.2
100	2097152	210	51.1	34.3
1000	2097152	2097	50.7	29.0
10000	2097152	20971	50.6	28.7

Figure 26: Varying the total amount of i/o to one 11+p RAID chain

number of jobs	block size (KB)	amount written (MB)	read rate (MB/s)	write rate (MB/s)	CPU utilization
1	1048576	52600	15	14.6	0.2
2	1048576	106721	(no data)	29.7	0.4
4	1048576	88290	(no data)	24.5	0.6
8	1048576	82924	23	23.0	2.0

Figure 27: Similar tests run using the *Ronnie* benchmark on one 4+p RAID chain