SR Performance Analysis

S. Hopkins, M. Ennis
Coraid, Inc.

Summary. This paper analyzes the performance of the SR appliances. The SR1521 is used to explain the local throughput rates for initialization and reconstruction. We then detail the AoE (ATA over Ethernet) throughput capability of each SR appliance for several RAID configurations. In this document the AoE throughput is determined as sustained sequential I/O throughput for the most optimal network configuration each appliance supports.

SR Local Rates

The following examples use the most common configuration for the SR1521 appliance, a RAID5 array over 14 disks leaving one disk available as a spare. Local rates for other configurations can be found in the full performance table listings.

The rate at which the SR is capable of rebuilding RAID5 parity is relevant for RAID5 initialization during first time creation as well as after any power failure. The `when` command reports the rate of all local reconstructions with an estimated time to completion. The rates stated below are the amount of total data being processed per second.

The SR1521 is capable of a rate of 488,112 KB/s:

```
SR shelf 21> list -l
 0 1070.527GB offline
  0.0   1070.527GB raid5 initing 425501
    0.0.0  normal       82.348GB 21.0
    0.0.1  normal       82.348GB 21.1
    0.0.2  normal       82.348GB 21.2
    0.0.3  normal       82.348GB 21.3
    0.0.4  normal       82.348GB 21.4
    0.0.5  normal       82.348GB 21.5
    0.0.6  normal       82.348GB 21.6
    0.0.7  normal       82.348GB 21.7
    0.0.8  normal       82.348GB 21.8
    0.0.9  normal       82.348GB 21.9
    0.0.10 normal       82.348GB 21.10
    0.0.11 normal       82.348GB 21.11
    0.0.12 normal       82.348GB 21.12
    0.0.13 normal       82.348GB 21.13
SR shelf 21> when
0.0 488112 KBps 0:25:59 left
SR shelf 21>
```

Extending the example to a 10.5TB RAID5 array over 14 750GB disks, the SR1521 will complete parity initialization in approximately 5 hours and 52 minutes.

Another important metric is the rate at which a disk can be reconstructed. During disk reconstruction the array is susceptible to a double failure. Faster disk reconstruction rates directly correlate to reduced exposure to this risk. The rates stated below are the amount of total data being processed per second.

The SR1521 is capable of a rate of 563,347 KB/s:

```
SR shelf 21> list -l
 0 1070.527GB offline
  0.0   1070.527GB raid5 needrecover recovering degraded 1316
    0.0.0  normal       82.348GB 21.0
    0.0.1  normal       82.348GB 21.1
    0.0.2  normal       82.348GB 21.2
    0.0.3  normal       82.348GB 21.3
    0.0.4  normal       82.348GB 21.4
    0.0.5  normal       82.348GB 21.5
    0.0.6  normal       82.348GB 21.6
    0.0.7  normal       82.348GB 21.7
    0.0.8  normal       82.348GB 21.8
    0.0.9  normal       82.348GB 21.9
    0.0.10 normal       82.348GB 21.10
    0.0.11 normal       82.348GB 21.11
    0.0.12 normal       82.348GB 21.12
    0.0.13 replaced     82.348GB 21.13
SR shelf 21> when
0.0 563347 KBps 0:33:50 left
SR shelf 21>
```

Extending the example as before, the SR1521 will complete recovery of a 750GB disk in a 10.5TB RAID5 array in approximately 5 hours and 10 minutes.

Additional local throughput rates are provided for each SR appliance in the performance tables that follow.

The following table displays disk rebuild and parity initialization rates for RAID5, RAID10, and RAID1. RAID10 and RAID1 both rebuild a disk mirror from a single disk; the rebuild effort is essentially the same, but both are presented for clarity.

The following numbers represent the total amount of processed data per second. These rates are sampled using the when command at the beginning of rebuild / initialization and are in units of KB/s. These rates are specifically relevant to the Samsung Spinpoint 500GB hard drives, model SAMSUNG HD512LJ; other disk models may exhibit higher or lower rates based on their capability. As the work proceeds further into the disk(s) the rate will decrease due to the slower disk zones internal to the disk(s). The amount of decrease varies with disk model.

Model	# Disks in RAID5	RAID5 REBUILD	RAID5 PARITY INIT	RAID10 REBUILD	RAID1 REBUILD
SR2461	23	961,479	688,557	174,867	171,293
SR1661	15	921,314	649,226	169,733	169,047
SR1521	14	561,487	487,202	168,764	168,585
SR1520	14	229,926	213,656	168,392	168,827
SR420	4	200,618	158,544	168,973	167,977

SR AoE Throughput Rates

The throughput statistics detailed in the following tables were achieved by averaging the results of three independent runs of ddt for the given SR configuration. `Ddt` is a simple tool that writes and reads sequentially to a file through a file system to determine the throughput capability of the file system and underlying storage. For a full description of `ddt`, please see *Appendix A*. Throughput for standard size Ethernet frames as well as jumbo Ethernet frames are presented.

Two Linux clients were used for these tests, one supporting 10GbE and one supporting multiple 1GbE. Both systems used a single dual core 3.0GHz Woodcrest CPU with 2GB RAM. The Linux kernel was 2.6.22.x and the aoe driver was aoe6-52. For the 1GbE tests, the client used the Intel 82546GB controller. For the 10GbE tests, the client used the Myricom 10GbE controller.

For each configuration, an XFS file system was placed on the resulting AoE device. The file system was mounted, and `ddt` was run against this mount point.

Each table contains a header describing the SR Appliance tested and the physical network connection(s) used to obtain the reported throughput. Throughput for the 10GbE Fiber options on the SR are equivalent to the CX4 statistics as the fiber mediums are only a change at the physical layer.

Configurations using a small and large number of disks are presented to show the range of capability of the appliance. Generally speaking, adding disks to a raid level will increase the throughput of reads or writes, or both. For the RAID5 and RAID10 examples containing a large number of disks we elected to show common configurations used by customers; by not using all the disks in the appliance the remaining disks can be declared as hot spares for failure allocation. The SR420 is an exception to this as with only 4 disks customers typically sacrifice automatic failure allocation for additional storage space.

SR2461-C, one 10GbE CX4 link

MTU	KiB / s	RAID0 24 DISK	RAID0 4 DISK	RAID10 22 DISK	RAID10 4 DISK	RAID5 23 DISK	RAID5 4 DISK	RAID1 2 DISK	JBOD 1 DISK
9000	WRITE:	412,585.67	130,821.67	263,015.33	74,658.33	360,791.67	91,006.00	37,648.00	41,875.33
	READ:	542,028.33	275,790.67	474,937.67	162,056.67	491,914.00	200,351.00	81,611.33	85,720.00
1500	WRITE:	165,102.67	127,512.67	99,379.33	68,077.67	107,693.00	79,663.67	36,653.00	41,669.33
	READ:	182,877.67	175,938.67	150,829.67	146,714.00	151,930.67	132,851.67	81,192.67	85,638.67

SR2461-G, four 1GbE links

MTU	KiB / s	RAID0 24 DISK	RAID0 4 DISK	RAID10 22 DISK	RAID10 4 DISK	RAID5 23 DISK	RAID5 4 DISK	RAID1 2 DISK	JBOD 1 DISK
9000	WRITE:	384,200.67	130,037.33	250,124.67	72,710.00	334,118.67	93,055.00	34,557.00	38,713.33
	READ:	464,154.67	281,230.33	420,450.67	156,122.67	425,319.00	206,599.67	79,220.00	80,636.33
1500	WRITE:	180,080.33	117,813.00	107,836.00	57,859.67	119,381.67	74,700.33	29,978.67	38,068.67
	READ:	169,159.67	163,106.67	147,327.33	137,397.67	147,686.00	130,251.33	78,951.33	80,574.00

SR1661-C, one 10GbE CX4 link

MTU	KiB / s	RAID0 16 DISK	RAID0 4 DISK	RAID10 14 DISK	RAID10 4 DISK	RAID5 15 DISK	RAID5 4 DISK	RAID1 2 DISK	JBOD 1 DISK
9000	WRITE:	449,692.33	131,149.67	213,507.00	74,082.00	313,376.33	90,986.00	36,597.67	39,343.67
	READ:	540,503.67	274,961.67	473,776.33	155,097.67	472,426.67	199,274.67	77,847.67	80,164.00
1500	WRITE:	165,079.00	125,623.00	89,319.33	64,785.00	106,712.00	78,956.33	35,410.33	39,551.33
	READ:	183,143.67	175,018.33	144,565.33	147,530.67	151,430.33	133,675.67	78,232.33	80,838.33

SR1661-G, four 1GbE links

MTU	KiB / s	RAID0 16 DISK	RAID0 4 DISK	RAID10 14 DISK	RAID10 4 DISK	RAID5 15 DISK	RAID5 4 DISK	RAID1 2 DISK	JBOD 1 DISK
9000	WRITE:	379,220.67	135,310.33	207,164.67	73,052.00	301,720.67	95,559.00	35,236.33	42,817.33
	READ:	440,663.33	283,546.67	432,722.33	157,675.33	406,900.00	209,191.33	79,075.00	88,506.67
1500	WRITE:	176,378.33	118,719.67	94,815.33	57,754.67	116,935.67	74,427.67	29,298.67	37,294.33
	READ:	166,992.67	159,025.67	139,341.00	135,803.67	144,523.00	128,442.67	78,118.67	89,044.33

SR1521, two 1GbE links

MTU	KiB / s	RAID0 15 DISK	RAID0 4 DISK	RAID10 14 DISK	RAID10 4 DISK	RAID5 14 DISK	RAID5 4 DISK	RAID1 2 DISK	JBOD 1 DISK
9000	WRITE:	239,333.33	106,840.33	162,068.67	71,196.67	216,853.67	92,672.33	36,525.00	38,057.67
	READ:	240,682.33	240,891.33	235,480.33	161,025.00	236,099.00	212,946.67	80,943.00	84,789.67
1500	WRITE:	114,288.67	97,436.00	65,229.67	54,341.00	70,224.33	59,805.33	35,432.00	36,611.00
	READ:	112,122.00	110,449.33	95,800.67	92,239.67	98,338.00	91,226.67	80,979.33	84,619.33

SR1520, one 1GbE link[1]

MTU	KiB / s	RAID0 15 DISK	RAID0 4 DISK	RAID10 14 DISK	RAID10 4 DISK	RAID5 14 DISK	RAID5 4 DISK	RAID1 2 DISK	JBOD 1 DISK
4200	WRITE:	104,253.00	100,832.00	52,251.67	49,750.33	96,317.00	72,052.00	34,966.00	36,765.67
	READ:	119,956.33	120,094.00	119,336.33	117,316.67	119,962.33	117,551.33	77,844.00	82,833.67
1500	WRITE:	71,905.67	67,677.00	41,572.00	38,183.67	42,842.33	40,682.33	33,684.00	36,431.00
	READ:	70,285.33	70,654.00	62,926.67	60,193.33	62,777.33	59,120.33	52,552.00	69,435.00

SR420, one 1GbE link[1]

MTU	KiB / s	RAID0 4 DISK	RAID10 4 DISK	RAID5 4 DISK	RAID1 2 DISK	JBOD 1 DISK
4200	WRITE:	101,397.33	49,857.33	72,223.33	36,278.33	38,432.00
	READ:	119,911.00	117,312.00	117,916.33	80,640.67	83,902.67
1500	WRITE:	67,447.00	38,406.33	40,499.33	33,870.67	37,483.33
	READ:	68,630.33	59,344.33	57,831.00	52,251.33	68,615.00

1 *For an explanation of why the SR1520 and SR420 perform best with one interface and an MTU of 4200, please see Application Note ANSR001 available at the SR support page (URL available in Appendix B).*

Appendix A - Performance Analysis with ddt

Performance analysis of the SR series of appliances has in the past been performed using bonnie++ as the benchmark. When used to analyze SR throughput and client system resource usage, bonnie++ has limitations: it only reports the CPU utilization for the bonnie++ user process, it does not report utilization properly in the face of multiple CPUs, it inflates write throughput by omitting a data sync operation as part of the write test, and it does not give the SR time between the write and read stages to flush its dirty buffers to avoid having previous writes affect the reads. The last item is something for which bonnie++ can not be expected to account. We have written a new program, ddt, to overcome these limitations.

Essentially, ddt is dd with timing information. No attempt has been made to make ddt accept the same options or command line syntax as dd. To obtain accurate CPU utilization, ddt uses a 2.6 Linux kernel proc file. As a result, ddt may not run correctly on 2.4 Linux kernels.

```
[root@stuart ddt-6]# ./ddt
usage: ./ddt [-?] [-c count] [-b bs] dir
```

The ddt program only requires one argument, the directory to be used for performance testing. It will create a file in this directory and time the task of writing count blocks of size bs to the file. It will then time reading count blocks of size bs from the file. It then reports the results. By default count is 16Ki (2^{14}) and bs is 256Ki (2^{18}); the default settings will write and read a file of size 4GiB. The source of the writes is random data returned from a malloc(bs).

In its output ddt accounts for CPU utilization in read and write tests by using the counters in /proc/stat. The /proc/stat file accounts for time spent in the following areas:

- user: normal processes executing in user mode
- nice: niced processes executing in user mode
- system: processes executing in kernel mode
- idle: twiddling thumbs
- iowait: waiting for I/O to complete
- irq: servicing interrupts
- softirq: servicing softirqs

CPU utilization is calculated as follows. Idle and iowait are summed to calculate the time spent not performing I/O (m). The sum of all counters is calculated and stored (n). The usual percentage calculation then follows:

$$\%\text{CPU utilization} = (n - m) * 100 / n$$

For this calculation to be most accurate, the client machine must not be otherwise in use as the /proc/stat counters are for all processes systemwide.

For more information on /proc/stat in Linux, see Documentation/filesystems/proc.txt in your favorite 2.6 Linux kernel source tree.

The following is an example run of ddt. Three columns of data are output. The first column states the amount of data written / read and displays the row labels for the subsequent statistics. The second column lists the respective throughput rate in KiB/s. The third column presents the total CPU% utilization during each test.

```
[root@stuart ddt-6]# ./ddt /mnt/e21.0/
Writing to /mnt/e21.0/ddt.17999 ... syncing ... done.
sleeping 10 seconds ... done.
Reading from /mnt/e21.0/ddt.17999 ... done.
4096 MiB  KiB/s   CPU%
Write    147718      8
Read     213178     15
[root@stuart ddt-6]#
```

Note the raw counter numbers are reported for validation. Due to the way bonnie++ calculates process utilization, only the user and system counters above would have been reported.

The source for the ddt program is available from the SR support page at coraid.com[1].

1 Please see Appendix B for URL

Appendix B - References

The SR support page includes the SR firmware, user manual, and related docs:

`http://www.coraid.com/support/sr/`

Please e-mail `support@coraid.com` with any questions or comments.