

SR Performance Analysis

S. Hopkins, M. Ennis, J. Boothe
CORAID, Inc.

Summary: This paper analyzes the performance of the SR appliances. First there will be a brief discussion of local parity initialization and LUN rebuild rates, followed by details on the AoE (ATA over Ethernet) throughput capability of each SR appliance for several RAID configurations.

SR Local Rates

The rate at which the SR is capable of rebuilding RAID5 parity is relevant for RAID5 initialization following LUN creation as well as after an unclean shutdown or power failure. The `when` command reports the rate of all local reconstructions with current rate in KB/s, plus an estimated time to completion.

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 following tables display rebuild and parity initialization rates for RAID5, and rebuild rates for 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 numbers in the following tables represent the total amount of data processed per second. The rates are sampled using the `when` command one minute after the beginning of rebuild/initialization and are presented in units of MB/s.

These rates are specifically relevant to the Western Digital RE3 1 TB hard drive, model WDI002FBYS-01A6B0; other disk models may exhibit higher or lower rates depending on their capability. As work proceeds further into the disk(s) the rate will decrease due to the slower disk zones, and the amount of decrease varies with disk model.

Model	# Disks in RAID5	RAID5 Rebuild	RAID5 Parity Init
SR2461	23	1510.79	1213.35
SR2421	23	1330.67	1156.03
SR1661	15	1288.73	651.72
SR1521	14	827.04	660.14
SR431	4	433.11	122.45

Model	# Disks in RAID10	RAID10 Rebuild	RAID1 (2-disk) Rebuild
SR2461	22	223.62	223.57
SR2421	22	223.54	223.21
SR1661	14	224.14	223.33
SR1521	14	223.20	223.10
SR431	4	222.92	221.83

SR AoE Throughput Rates

The throughput statistics detailed in the following tables were achieved by averaging the results of five independent runs of `ddt` for the given SR configuration. `Ddt` is a simple tool that writes and reads sequentially to a file through a filesystem to determine the throughput capability for the filesystem and underlying storage. For a full description of `ddt` please see Appendix A.

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 2 GB RAM. The Linux kernel was 2.6.22.x, and the AoE driver `aoe6-64`. For the 1GbE tests the client used the Intel 82546GB controller; for the 10GbE tests the Myricom 10GbE was used. SR firmware used is SR20081204.

For each configuration an XFS filesystem was placed on the AoE device being tested. The filesystem was mounted, then `ddt` was run against this mount point. Throughput for standard size ethernet frames as well as jumbo ethernet frames are presented.

Each table contains a header describing the SR appliance tested and the physical network connection(s) used to obtain the reported throughput. Throughput on the 10GbE optical fiber options available for the SR2461 and the SR1661 are equivalent to the CX4 performance as the fiber medium is 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 the addition of 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 disks in the appliance for LUN elements, the remaining disks can be assigned as hot spares for failure allocation. The SR431 is an exception to this as with only four disks customers typically sacrifice automatic failure allocation for additional storage space.

SR2461-C2, one 10 GbE CX4 link

MTU	MB/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	453.15	164.69	276.06	76.52	391.99	104.70	42.26	42.08
	Read	563.05	383.40	544.76	192.50	544.49	287.14	104.41	106.21
1500	Write	166.83	139.91	100.25	72.27	108.41	90.65	42.53	41.64
	Read	179.71	180.24	149.29	147.29	149.05	146.99	103.76	106.09

SR2461-G, four 1GbE links

MTU	MB/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	379.28	163.36	244.63	73.48	340.48	105.01	42.08	38.27
	Read	474.54	382.22	464.81	190.64	467.29	282.38	108.55	107.21
1500	Write	176.67	138.35	108.33	66.91	121.21	94.83	37.52	41.09
	Read	168.38	168.68	147.15	144.96	147.00	144.13	107.21	108.71

SR2421, two 1GbE links

MTU	MB/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	240.03	158.60	211.84	75.83	239.39	105.83	41.38	41.36
	Read	241.97	242.17	242.11	191.95	242.01	241.61	103.22	106.10
1500	Write	120.12	108.70	68.11	59.41	72.31	67.10	40.69	41.19
	Read	112.94	112.50	99.09	98.35	99.04	98.07	83.94	106.92

SR1661-C2, one 10 GbE CX4 link

MTU	MB/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	428.18	160.42	215.25	75.37	344.95	105.85	41.99	42.31
	Read	559.32	343.91	522.82	191.02	537.47	264.75	104.29	106.32
1500	Write	172.04	138.93	99.84	69.89	109.57	88.41	42.46	41.81
	Read	189.90	187.66	155.80	155.14	156.26	151.10	103.41	107.50

SR1521, two 1GbE links

MTU	MB/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	239.90	163.46	178.41	77.34	230.25	109.75	41.79	44.25
	Read	242.07	241.57	242.01	189.75	241.84	241.59	107.37	107.89
1500	Write	114.74	106.67	65.88	58.97	70.60	66.07	39.63	43.21
	Read	109.68	110.06	97.19	96.51	96.62	96.06	82.29	106.55

SR431, two 1GbE links

MTU	MB/s	RAID0 4- disk	RAID10 4- disk	RAID5 4- disk	RAID1 2- disk	JBOD 1 disk
9000	Write	161.79	74.64	106.37	40.73	42.89
	Read	239.83	172.83	232.70	97.69	97.67
1500	Write	106.60	58.54	66.08	39.00	42.13
	Read	111.22	96.80	97.16	82.41	98.91

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 does not report CPU utilization properly in the face of multiple CPUs, inflates write throughput by omitting a data sync operation as part of the write test, and does not give the SR time between 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++` cannot be expected to account. We have written a 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
usage: ddt [-?] [-c count] [-b bs] dir
[root@stuart ~]#
```

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, then time reading `count` blocks of size `bs` from the file. 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: 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:

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

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 system-wide.

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 percentage during each test.

```
[root@stuart ~]# ddt /mnt/8.1
Writing to /mnt/8.1/ddt.9343 ... syncing ... done.
sleeping 10 seconds ... done.
Reading from /mnt/8.1/ddt.9343 ... done.
4096 MiB  KiB/s    CPU%
Write   378889    37
Read    492000    62
[root@stuart ~]#
```

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

The source for the `ddt` program, plus many other useful resources for the SR series of appliances, are available from the SR support page at <http://support.coraid.com/support/sr/>.

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