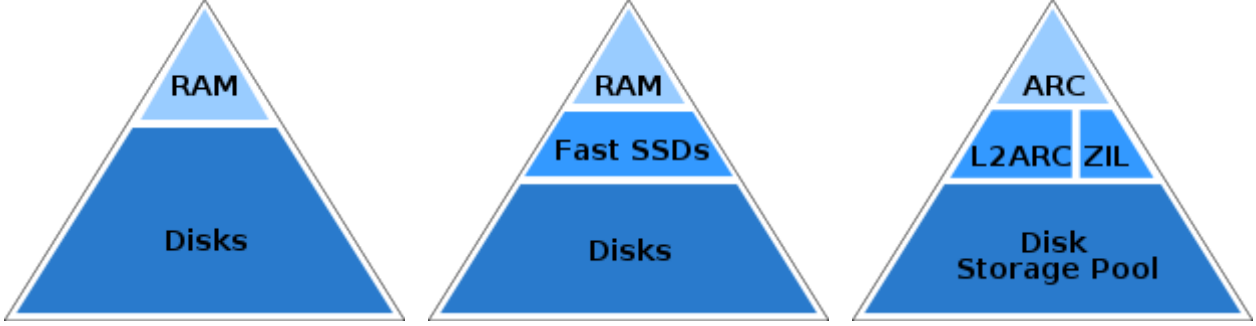
Hybrid Storage Pool: Top Speeds

08 Oct 2009

I originally posted this at http://blogs.sun.com/brendan/entry/hybrid_storage_pool_top_speeds.

I'd like to show the benefit of the Hybrid Storage Pool (HSP) in the simplest possible way, as an addition to my top speed results for the Sun Storage 7410. The HSP uses both read and write optimized flash based solid state disks (SSDs) to improve the performance of file I/O, with the ZFS L2ARC and SLOG (separate ZIL) technologies:



old model *new model* *with ZFS*

I've demonstrated both the HSP technologies previously in detail, and explained how they work:

- HSP's random read cache: L2ARC screenshots
- HSP's separate intent log: SLOG screenshots

Here I'll test how these HSP technologies perform with simple workloads, to measure best-case latency.

Disclaimer: Performance isn't simple. Measuring the speed of simple workloads can be easy; inferring from this how a system will behave in any given production environment can be quite hard. The best way to understand performance is to test the system with a workload that closely resembles the production environment.

Latency

The results I'll measure for the L2ARC and SLOG will be latency, not IOPS or throughput. It doesn't make sense to compare IOPS values alone while ignoring the latency of those IOPS ("10,000 IOPS" from flash memory does not equal "10,000 IOPS" from slow rotating disks!) Throughput isn't characteristic either – while the HSP delivers great throughput, it usually does so by minimizing use of the L2ARC and SLOG and using multiple disk spindles instead.

The latency presented here will be measured from the NFS layer using Analytics on the Sun Storage server. This latency will include the overheads from processing the NFS protocol, the ZFS filesystem, and waiting for the flash based device I/O itself. The reason is to best portray the delivered performance of the whole product, rather than showing component performance.

Testing Notes

A couple of notes about performance testing SSDs.

- **Age the SSDs**: Flash memory based SSDs can be considerably faster the first time you fill them. The second time they are populated the I/O latency can become much worse as the data becomes fragmented. (Fragmentation may sound odd in the context of flash memory, but this behavior does exist.) So, the SSDs I test are aged – they've been completely filled with data many times before.
- **Use Gbytes of data**: Like rotating disks, SSD devices can have on-disk DRAM to cache I/O. If you are interested in flash memory read latency but test this by rereading a small amount of data (Mbytes), you may hit from the on-disk cache instead. I use large working sets (in terms of Gbytes) to avoid this.

L2ARC

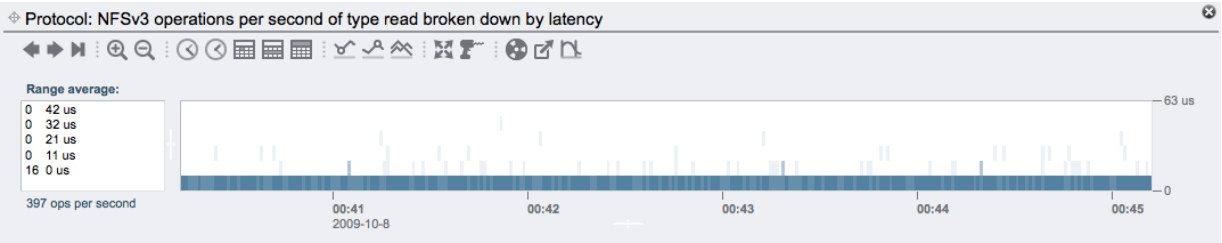
See my screenshots for the full introduction to the L2ARC. To characterize it, I'll perform random 512 byte reads over NFSv3 from a single thread on a single client. The target is a 7410 (Barcelona) with 6 L2ARC devices (Readzillas). Some of the workload will hit from DRAM, some from the L2ARC, and some from disk. The amount from each will depend on cache warmth and working set size; what we are interested in here is the latency.

To find the top speed, the ZFS record size is set to 512 bytes to match the I/O size: but before you try this yourself, understand that ZFS record sizes smaller than 4 Kbytes do begin to cost noticeable amounts of DRAM (bigger metadata/data ratio) and significantly reduce streaming I/O performance. Before this sounds too

unlikely to be interesting, note that a workload involving thousands of tiny files may behave in a similar way; the ZFS record size only takes effect if the file becomes bigger than that size.

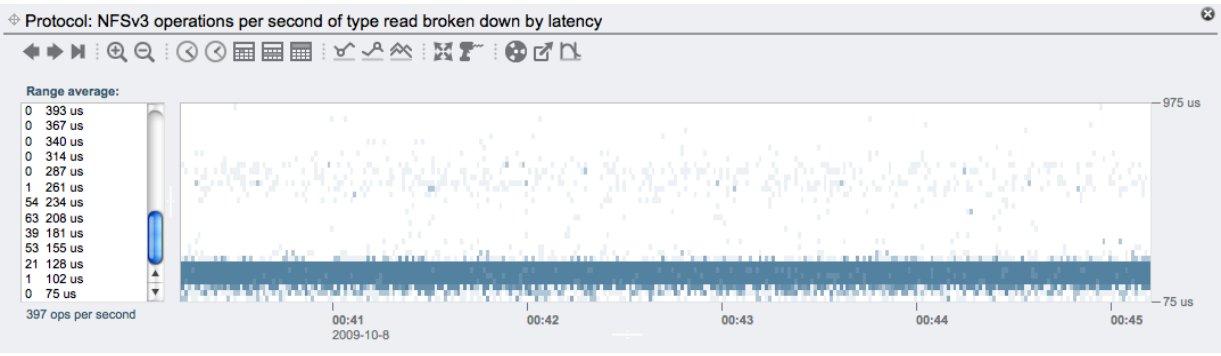
The following screenshots show NFS read latency heat maps at different vertical ranges (click for larger versions), each corresponds to the HSP technology returning that read. I've also listed the latency range which covers most of the I/O:

DRAM latency: 0 - 10 us



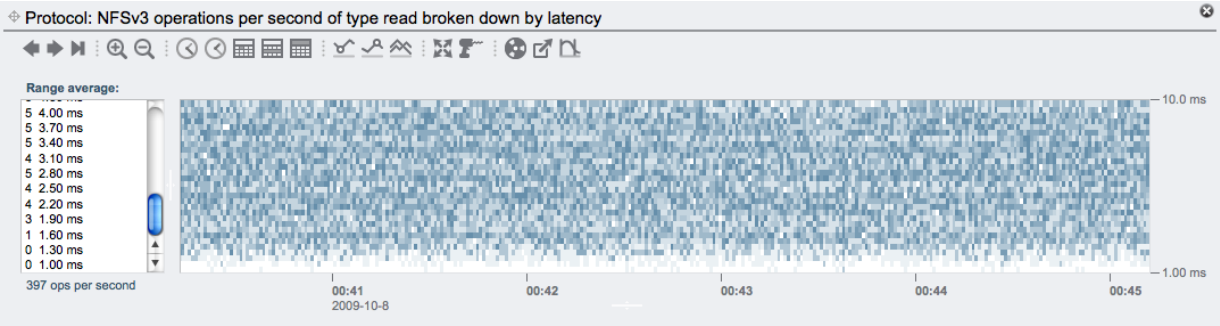
This shows hits from the ZFS DRAM cache (the ARC: Adaptive Replacement Cache), which are fast as expected.

L2ARC latency: 128 - 260 us



The L2ARC is returning these 512 byte reads with consistently low latency. The range may sound high when compared to advertised SSD 512 byte read latency, but remember that these are aged SSDs, and this time includes the ZFS and NFS overheads.

Disk latency: 1900 - 9700 us

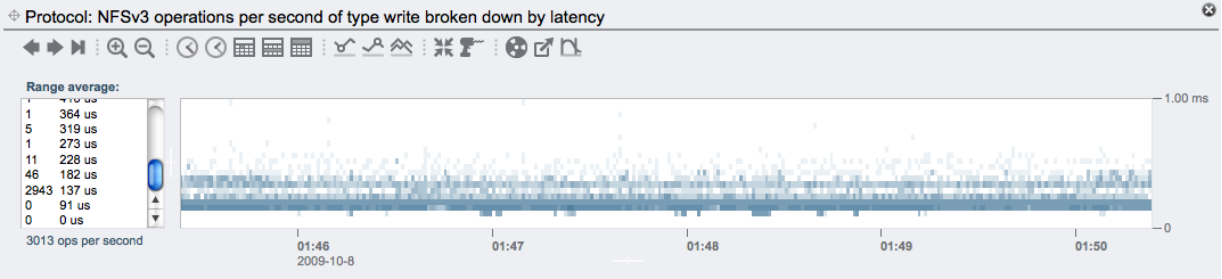


These 7200 RPM disks are returning the reads with latency from 2 to 10 ms, as expected for random disk I/O (platter rotation time + head seek time.)

SLOG

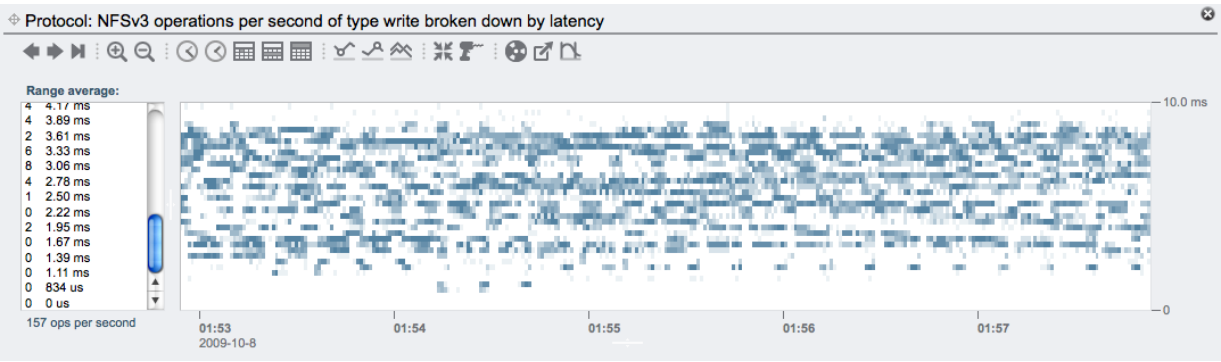
See my screenshots for the full introduction to the SLOG. To characterize it, I'll perform 512 byte O_DSYNC writes over NFSv3. Since these will all be handled by the SSD SLOG device (Logzilla), I'll also show what happens without them – the synchronous write latency to disk. As before, I'll show screenshots and latency ranges.

SLOG latency: 137 - 181 us



Most of the I/O is in a tight band in the 100 to 200 us range (the screenshot doesn't look so tight due to the false color palette; see the range average values on the left).

Disk latency: 1950 - 9170 us



Without the SSD based SLOG devices, synchronous write I/O is served here from 7200 RPM disks, with a latency between 2 and 9 ms.

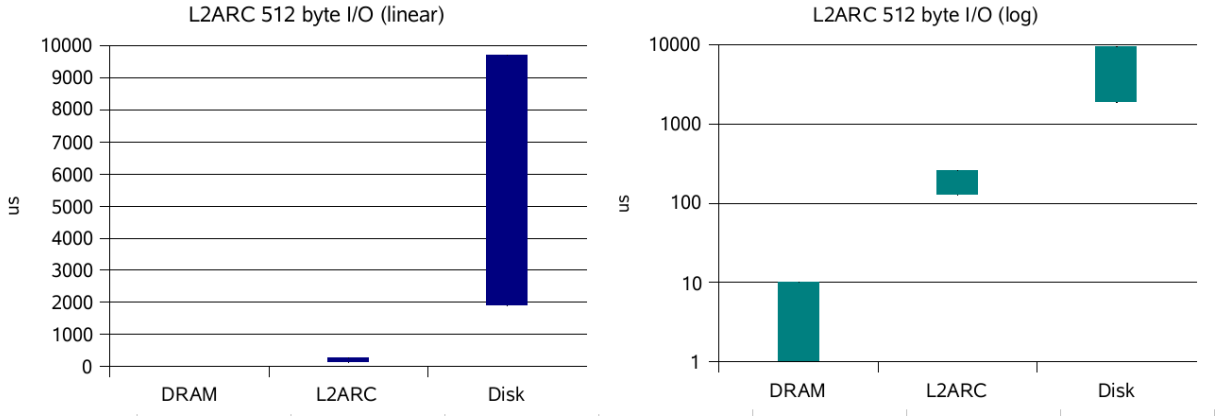
Reality Check

While I've been showing top speeds of the L2ARC and SLOG devices, I can help set expectations by describing cases where you will be slower (or faster!) than these top speeds.

- **I/O size.** To find the top speeds above I used a application I/O size of 512 bytes. Bigger I/O takes longer, although most of the time for the 512 byte I/O is processing the I/O, not data transfer, so this won't scale as bad as it may sound. As an example, see this L2ARC 4 Kbyte I/O screenshot – it's a little slower than the 512 byte I/O, but not 8 times slower.
- **FS record size.** For the L2ARC test I set the filesystem record size to 512 bytes before creating the files, so that the L2ARC devices would be accessed in 512 byte blocks. To keep metadata/data ratios in check and to keep up streaming performance, I don't usually set the record size to below 4 Kbytes (although a smaller record size may be used inadvertently due to thousands of small files). The previous screenshot was also with a 4 Kbyte FS record size, which (along with the I/O size) contributed to the slightly higher latency.
- **Threads.** For these top speed tests, I only used one client process (and one thread) to perform I/O. If multiple threads access the L2ARC and SLOG devices simultaneously, then contention can increase the latency. The more simultaneous threads, the worse it gets. This is the same for disk I/O, but with the HSP we usually have more disk spindles than SSD devices, so the pool of disks can in some cases handle more concurrent I/O than the SSDs can (depends on the number of disks and pool profile.)
- **SSD technology.** The latencies shown above are for the current SSD devices we are shipping with the Sun Storage 7000 series. SSD technology has been improving quickly, so I'd expect these latencies to get better over time.
- **Workload.** The L2ARC is currently tuned for random reads, and the SSD SLOG devices are used for synchronous writes. If you test streaming reads or asynchronous writes, the HSP will deliver great performance, but it may not use the SSD devices very much. In these cases the latency will often be better, but that isn't showing SSD latency. The HSP uses the best tool for the job, which may or may not be the L2ARC and SLOG devices.

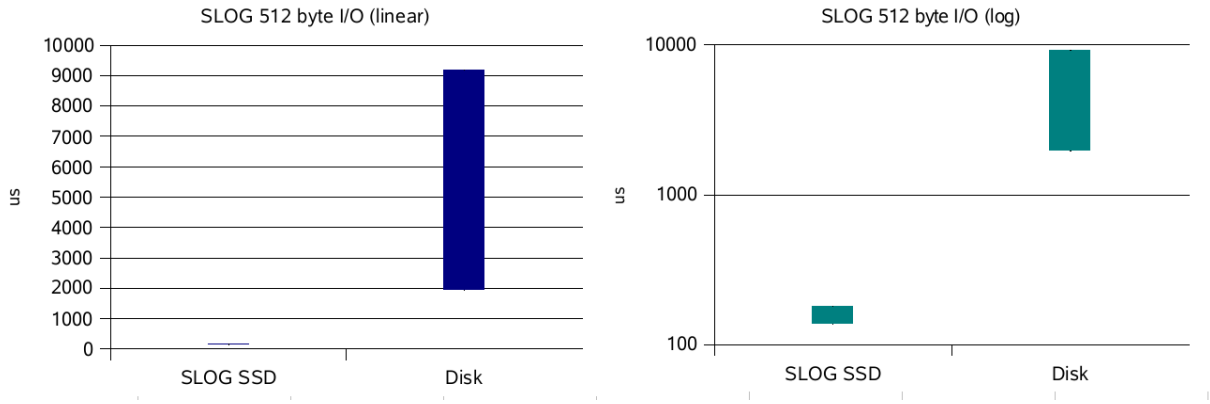
Summary

NFS latency ranges for the **L2ARC**:



Here DRAM latency doesn't quite register a pixel in height on the linear plot. The logarithmic plot shows the role of the SSD based L2ARC very well, in between DRAM and disk.

NFS latency ranges for the **SLOG**:



Conclusion

When the HSP uses read and write optimized flash memory SSDs the NFS I/O latency can reach as low as 0.1 ms, compared to 2 to 10 ms without these SSDs (disk + DRAM only.) These results aren't really surprising considering the technology.

I've updated my top speeds with these latency results. And as top speeds go, this is as good as it gets. Your latency will be higher with larger I/O sizes.