Performance Testing the 7000 series, part 1 of 3

23 Mar 2009

I originally posted this at http://blogs.sun.com/brendan/entry/performance_testing_the_7000_series1.

With the introduction of the Sun Storage 7000 series there has been much interest in its performance, which I've been demonstrating in this blog. Along with Bryan and Roch, I've also been helping other teams properly size and evaluate their configurations through performance testing. The advent of technologies like the hybrid storage pool makes performance testing more complicated, but no less important.

Over the course of performance testing and analysis, we assembled best practices internally to help Sun staff avoid common testing mistakes, tune their systems for maximum performance, and properly test the hybrid storage pool. In the interest of transparency and helping others understand the issues surrounding performance testing, I'll be posting this information over two posts, and my load generation scripts in a third.

Performance Testing - Top 10 Suggestions:

1. **Sanity Test**

   Before accepting a test result, find ways to sanity test the numbers.

   When testing throughput over a gigabit network interface, the theoretical maximum is about 120 Mbytes/sec (converting 1 GbE to bytes.) I've been handed results of 300 Mbytes/sec and faster over a gigabit link, which is clearly wrong. Think of ways to sanity test results, such as checking against limits.

   IOPS can be checked in a similar way: 20,000 x 8 Kbyte read ops/sec would require about 156 Mbytes/sec of network throughput, plus protocol headers – too much for a 1 GbE link.

2. **Double Check**

   When collecting performance data, use an alternate method to confirm your results.

6. **Disks Matter**

   Don't ignore the impact of rotational storage.

   A full Sun Storage 7410 can have access to a ton of read cache: up to 128 Gbytes of DRAM and six 100 Gbyte SSDs. While these caches can greatly improve performance, disk performance can't be ignored as data must eventually be written to (and read from) disk. A *bare minimum* of two fully-populated JBODs are required to properly gauge 7410 performance.

7. **Check Your Storage Profile**

   Evaluate the desired redundancy profile against mirroring.

   The default RAID-Z2 storage profile on the 7000 series provides double-parity, but can also deliver lower performance than mirroring, particularly with random reads. Test your

If a test measures network throughput, validate results from different points in the data path: switches, routers, and of course the origin and destination. A result can appear sane but still be wrong. I've discovered misconfigurations and software bugs this way, by checking if the numbers add up end-to-end.

3. **Beware of Client Caching**

   File access protocols may cache data on the client, which is performance tested instead of the fileserver.

   This mistake should be caught by the above two steps, but it is so common it deserves a separate mention. If you test a fileserver with a file small enough to fit within the client's RAM, you may be testing client memory bandwidth, not fileserver performance. This is currently the most frequent mistake we see people make when testing NFS performance.

4. **Distribute Client Load**

   Use multiple clients, at least 10.

   The Sun Storage 7410 has no problem saturating 10 GbE interfaces, but it's difficult for a client to do the same. A fileserver's optimized kernel can respond to requests much quicker than client-side software can generate them. In general, it takes twice the CPU horsepower to drive load than it does to accept it.

   Network bandwidth can also be a bottleneck: it takes at least ten 1 Gbit clients to max out a 10 Gbit interface. The 7410 has been shown to serve NFS at 1.9 Gbytes/sec, so at least sixteen 1 Gbit clients would be required to test max performance.

5. **Drive CPUs to Saturation**

   workload with mirroring as well as RAID-Z2, then compare price/performance and price/Gbyte to best understand the tradeoff made.

8. **Use Readzillas for Random Reads**

   Use multiple SSDs, tune your record size, and allow for warmup.

   Readzillas (read-biased SSDs), can greatly improve random read performance, if configured properly and given time to warm up. Each Readzilla currently delivers around 3,100 x 8 Kbyte read ops/sec, and has 100 Gbytes of capacity. For best performance, use as many Readzillas as possible for concurrent I/O. Also consider that, due to the low-throughput nature of random-read workloads, it can take several hours to warm up 600 Gbytes of read cache.

   On the 7000 series, when using Readzillas on random read workloads, adjust the database record size from its 128 Kbyte default down to 8 Kbytes *before* creating files, or size it to match your application record size. Data is retrieved from Readzillas by their record size, and smaller record sizes improve the available IOPS from the read cache. This must be set before file creation, as ZFS doesn't currently rewrite files after this change.

9. **Use Logzillas for Synchronous Writes**

   Accelerate synchronous write workloads with SSD based intent logs.

   Some file system operations, like file and directory creation, and writes to database log files are considered "synchronous writes," requiring data be on disk before the client can

If the CPUs are idle, the system is not operating at peak performance.

The ultimate limiter in the 7000 series is measured as CPU utilization, and with the 7410's four quad-core Opterons, it takes a tremendous workload to reach its limits. To see the system at peak performance, add more clients, a faster network, or more drives to serve I/O. If the CPUs are not maxed out, they can handle more load.

This is a simplification, but a useful one. Some workloads are CPU heavy due to the cycles to process instructions, others with CPU wait cycles for various I/O bus transfers. Either way, it's measured as percent CPU utilization, and when that reaches 100% the system can generally go no faster (although it may go a little faster if polling threads and mutex contention backs off.)

continue. Flash-based intent log devices, or Logzillas, can only dramatically speed up workloads comprised of synchronous writes; otherwise, data is written directly to disk.

Logzillas can provide roughly 10,000 write ops/sec, depending on write size, or about 100 Mbytes/sec of write throughput, and scale linearly to meet demand.

10. **Document Your Test System**

    Either test a max config, or make it clear that you didn't.

    While it's not always possible or practicable to obtain a maximum configuration for testing purposes, the temptation to share and use results without a strong caveat to this effect should be resisted. Every performance result should be accompanied by details on the target system, and a comparison to a maximum configuration, to give an accurate representation of a product's true capabilities.

The main source of problems in performance testing that we've seen is the misuse of benchmark software and the misinterpretation of their results. The above suggestions should help the tester avoid the most common problems in this field. No matter how popular or widely-used benchmark software is, the tester is obliged to verify the results. And by paying sufficient attention to the test environment – i.e. system configuration, client load balance, network bandwidth – you can avoid common pitfalls (such as measuring 300 Mbytes/sec over a 1 Gbit/sec interface, which was courtesy of a popular benchmarking tool.)

In part two, I'll step through a more detailed checklist for max performance testing.