Performance Testing the 7000 series, part 2 of 3

02 Apr 2009

I originally posted this at http://blogs.sun.com/brendan/entry/performance_testing_the_7000_series2.

With the release of the Sun Storage 7000 series there has been much interest in the products performance, which I've been demonstrating. In my previous post I listed 10 suggestions for performance testing – the big stuff you need to get right. The 10 suggestions are applicable to all perf testing on the 7000 series. Here I'll dive into the smaller details of max performance tuning, which may only be of interest if you'd like to see the system reach its limits.

The little stuff

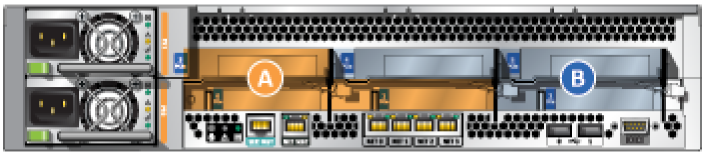
The following is a tuning checklist for achieving maximum performance on the Sun Storage 7410, particularly for finding performance limits. Please excuse the brevity of some descriptions, this was originally written as an internal Sun document and has been released here in the interests of transparency. This kind of tuning is used during product development, to drive systems as fast as possible to identify and solve bottlenecks.

These can all be useful points to consider, but they **do not apply to all workloads**. I've seen a number of systems that were misconfigured with random tuning suggestions found on the Internet. Please understand what you are doing before making changes.

Sun Storage 7410 Hardware

- **Use the max CPU and max DRAM** option (currently 4 sockets of quad core Opteron, and 128 Gbytes of DRAM.)
- **Use 10 GbE**. When using 2 ports, ideally use 2 x 10 GbE cards and one port on each, which balances across two PCI-E slots, two I/O controllers, and both CPU to I/O HyperTransports. Two ports on a single 10 GbE card will be limited by the PCI-E slot throughput.

  Using LACP to trunk over 1 GbE interfaces is a different way to increase network bandwidth, but you'll have port hashing to balance in small test environments (eg, fewer than 10 clients), which can be a headache based on the client attributes (my 20 client test farm all have *even-numbered* IP address and mac-addresses!)
- **Balance load across I/O controllers**. There are two I/O controllers in the 7410, the MCP55 and the IO55. The picture on the right shows which PCI-E slots are served by which controller, labeled A and B. When using two 10 GbE cards, they should (already) be installed in the top left and bottom right slots, so that their load is balanced across each controller. When three HBAs are used, they are put in the bottom left and both middle slots; if I'm running a test that only uses two of them, I'll use the middle ones.



- **Get as many JBODs** as possible with the largest disks possible. This will improve random I/O for cache busting workloads, although considering the amount of cache possible (DRAM plus Readzilla), to be cache busting it may become too artificial to be interesting.
- **Use 3 x HBAs**, configure dual paths to chains of 4 x JBODs (if you have the full 12.)
- **Consider Readzilla** for random I/O tests, but plan for a long (hours) warmup. Make sure the share database size is small (8 Kbytes) before file creation. Also use multiple readzillas for concurrent I/O (you get about 3100 x 8 Kbyte IOPS from each of the current STECs.) Note that Readzilla (read cache) can be enabled on a per filesystem basis.
- **Use Logzilla** for synchronous (O_DSYNC) write workloads, which may include database operations and small file workloads.

Sun Storage 7410 Settings

- **Configure mirroring** on the pool. If there is interest in a different profile, test it in addition to mirroring. Mirroring will especially help random IOPS read and write workloads that are cache busting.
- Create **multiple shares** for the clients to use, if this resembles the target environment. Depending on the workload, it may improve performance by a tiny amout for clients not to be hammering the same share.
- **Disable access time** updates on the shares.
- **128K recsize for streaming**. For streaming I/O tests: 128 Kbyte "database size" on the shares before file creation.
- **8K recsize for random**. Random I/O tests: set the share "database size" to match the application record size. I generally wouldn't go smaller than 4 Kbytes. If you make this extremely small then the ARC metadata to reference tiny amounts of data can begin to consume Gbytes of DRAM. I usually use 8 Kbytes.
- Consider testing with **LZJB compression** on the shares on to see if it improves performance (it may relieve back-end I/O throughput.) For compression, make sure your working set files actually contain data and aren't all zeros – ZFS has some tricks for compressing zero data that will artificially boost performance.
- Consider suspending DTrace based **Analytics** during tests for an extra percent or so of performance (especially the by-file, by-client, by-latency, by-size and by-offset ones.) The default Analytics (listed in the HELP wiki) are negligible, and I leave them on during my limit testing (without them I couldn't take screenshots.)
- **Don't benchmark a cold server** (after boot.) Run a throwaway benchmark first until it fills DRAM, then cancel it and run the real benchmark. This avoids a one-off kmem cache buffer creation penalty, that may cost 5% performance or so only in the minutes immediately after boot.

Network

- **Use jumbo frames** on clients and server ports (not needed for management). The 7410 has plenty of horsepower to drive 1 GbE with or without jumbo frames, so their use on 1 GbE is more to help the clients keep up. On 10 GbE the 7410s peak throughput will improve by about 20% with jumbo frames. This ratio would be higher, but we are using LSO (large send offload) with this driver (nxge) which keeps the non-jumbo frame throughput pretty high to start with.
- **Check your 10 Gbit switches**. Don't assume a switch with multiple 10 GbE ports can drive them at the same time; some 10 Gbit switches we've tested cap at 11 Gbits/sec. The 7410 can have 4 x 10 GbE ports, so make sure the switches can handle the load, such as by using multiple switches. You don't want to test the *switches* by mistake.

Clients

- **Lots of clients**. At least 10. 20+ is better. Everyone who tries to test the 7410 (including me) gets client bound at some point.
- **A single client can't drive 10 GbE**, without a lot of custom tuning. And if the clients have 1 GbE interfaces, it should (obviously) take at least 10 clients to drive 10 GbE.
- Connect the clients to **dedicated switch(s)** connected to the 7410.
- You can **reduce client caching** by booting the clients with less DRAM: eg, eeprom physmem=786432 for 3 Gbytes on Solaris x86. Otherwise the test can hit from client cache and test the client instead of the target!

Client Workload

- **NFSv3** is generally the fastest protocol to test.
- **Consider umount/mount** cycles on the clients between runs to flush the client cache.
- **NFS client mount options**: Tune the "rsize" and "wsize" option to match the workload. Eg, rsize=8192 for random 8 Kbyte IOPS tests, to reduce unnecessary read-ahead; and rsize=131072 for streaming (and also try setting nfs3_bsize to 131072 in /etc/system on the clients for the streaming tests.)

  For read tests, try "forcedirectio" if your NFS client supports this option (Solaris does) – this especially helps clients apply a heavier workload by not using their own cache. Don't leave this option enabled for write tests.
- Whatever benchmark tool you use, you want to try **multiple threads/processes per client**. For max streaming tests I generally use 2 processes per client with 20 clients, so thats 40 client processes in total; for max random IOPS tests I use a lot more (100+ client processes in total.)
- **Don't trust benchmark tools** – most are misleading in some way. Roch explains Bonnie++ here, and we have more posts planned for other tools. Always double check what the benchmark tool is doing using Analytics on the 7410.
- **Benchmark working set**: very important! To properly test the 7410, you need to plan different total file sizes or working sets for the benchmarks. Ideally you'd run one benchmark that the server could entirely cache in DRAM, another that cached in Readzilla, and another that hit disk.

  If you are using a benchmark tool without tuning the file size, you are probably getting this wrong. For example, the defaults for iozone are very small (512 Mbytes and less) and can entirely cache on the client. If I want to test both DRAM and disk performance on a 7410 with 128 Gbytes of DRAM, I'll use a total file size of 100 Gbytes for the DRAM test, and at least 1 Terabyte (preferably 10 Terabytes) for the disk test.
- Keep a close eye on the clients for any issues. Eg, **DTrace** kernel activity.
- Getting the best numbers involves **tuning the clients** as much as possible. For example, use a large value for autoup (300); tune tcp_recv_hiwat (for read tests, 400K should be good in general, 1 MB or more for long latency links.) The aim is to eliminate any effects from the available clients, and have results which are bounded by the targets performance.
- **Aim for my limits**: That will help sanity check your numbers, to see if they are way off.

The Sun Storage 7410 doesn't need special tuning at all (no messing with /etc/system settings.) If it did, we'd consider that a bug we should fix. Indeed, this is part of what Fishworks is about – the expert tuning has already been done in our products. What there is left for the customer is simple and industry common: pick mirroring or double parity RAID, jumbo frames, no access time updates and tuning the filesystem record size. The *clients* require much, much more tuning and fussing when doing these performance tests.

In my next post, I'll show the simple tools I use to apply test workloads.

About this blog

Homepage
Blog
Full Site Map
Sys Perf book
Linux Perf
Perf Methods
USE Method
TSA Method
Off-CPU Analysis
Active Bench.
Flame Graphs
Heat Maps
Frequency Trails
Colony Graphs
perf Examples
eBPF Tools
DTrace Tools
DTraceToolkit
DtkshDemos
Guessing Game
Specials
Books
Other Sites