Blog Export: c0t0d0s0.org, http://www.c0t0d0s0.org/

Monday, October 13. 2008

L2ARC on ramdisks?

I thought a little bit about the idea of transforming server into solid state disks. The idea in the mail of Chris Greer on zfs-discuss was to use mirrored iSCSI shared ramdisks as a storage for the seperated ZILs. But i think you could use the concept as well for L2ARC as well - e.g. for large databases. One of the sizing rules of databases: More main memory never hurts. Nothing helps the performance of a database more than even more memory. The rule of "main memory never hurts" is based on the fact, that a hard disk has only a few IOPS compared with the main memory and hard drive access massively hurts the performance of your database.

But obviously the size of memory is limited, albeit the this limit is quite high with systems with memory sizes in the range of 512 GB on 4 rack units. But how can you get more memory into your database system, when all DIMM slots are filled with the biggest available DIMMS.

I had an idea while cooking tea this evening while i thought about a discussion with a colleague: Let´s assume an architecture based on a X4600 as a head in front of four X4600 each fully maxed to 512GB. All the nodes are connected with Infiniband. The first X4600 is your normal database server (for example mysql or LarryBase). You put your data into an ZFS storage pool. This storage pool is augumented with L2ARC devices. But now comes the plot twist. Let´s use the 512GB X4600 as huge ramdisks (yes, i know, every engineers heart will crying now) speaking via iSER (no TCP/IP, just RDMA) at 20 GBit/s to the central database node. This would give you a cache in the size of almost 2 TB plus the cache on the database server itself.. By using L2ARC you could use the memory as database caches of other systems without using a database doing a combination of the memory resources by other means, for example the CacheFusion stuff of Oracle. You don´t have to fuse the caches of other databases servers. The other servers are caches. You don´t have to partition the databases.

It would be interesting how such an system would perform in comparision to a Oracle RAC or other memory implementations. Anybody out there willing to test this ... my Infiniband switches are in the laundry at the moment

Posted by Joerg Moellenkamp in English, Solaris at 18:47

...und die Opterons in den L2ARC-Servern werden eigentlich nur als "Speichererweiterung" benoetigt... Wie lange ist denn der Timeout, dass statt L2ARC vom zpool gelesen wird, falls ein L2ARC-Rechner mal weg ist?
Anonymous on Oct 15 2008, 09:46

Nun ... ich habe schon ueberlegt, ob man die Opterons in den Systemen zur Kompression nutzen kann, wenn man uebr die ramdisks noch ein ZFS legt und das dann via iSCSI geht.

Also ich habe das noch nicht genau nachgeschaut mit dem Timeout, aber beim ziehen des Netzwerkkabels bei eigenen Tests konnte ich keine Unterbrechung wahrnehmen (aber natuerlich in den Benchmarks nachweisen)
Anonymous on Oct 15 2008, 10:30

This is actually a very stupid solution.

1. 20Gbit/second RDMA is only 2.5GB/sec(theoretical peak). A SAS controller with dual 4x SAS ports can give you about 24Gbit/sec(300Mbit*8 ports). So hooking Intel or Samsung SLC SSDs on SAS expanders will be a lot cheaper. (You don't have to buy X4600s with 8 AMD opteron 8000 CPUs let alone 64 8GB dual rank DIMMs)

2. This is exactly the idea behind memcache. Instead of 512GB ram per node talking over 20Gbit infiniband, you have 64GB ram nodes talking over 4Gbit trunked GigE switches. Except a single X4600 with 512GB of ram can probably cost you more if you bought commodity 64GB servers.

3. Failure Mode. If one of the 512GB L2ARC server fails, then your disks will be going berserk trying to refill 512GB of random reads from your disks, assuming that you have one expensive standby. If a 64GB memcache server fails, the cost is 1/8 the X4600.
Anonymous on Dec 9 2008, 21:33

Don´t dismiss an idea as stupid as long you don´t understand all factors ...

When an application is capable to use memcache, okay .. that´s fine ... but this solution would increase the cache for an interface all applications already can use, the ARC of ZFS. Look at the usual suspects of closed-source software. I´m not aware of memcache support for Siebel, Oracle, Notes ...

Of course you could use SAS instead of Infiniband as well. Simply use the SAS Target in Opensolaris.

By the way ... it´s not about building a substitute for SSD. It´s more about of building something for even more extreme IOPS rates with more DRAM-like characteristics.

And you could build the same out of smaller servers as well based on this solution. I just took the X4600 because it´s the biggest x86 server at the moment memorywise ...
Anonymous on Dec 10 2008, 08:19