Wednesday, August 19. 2009

SSD, ZFS, L2ARC and mysql

Charles Suresh published some interesting findings in "Improving MySQL/Innodb/ZFS read-only performance by 5x using SSDs for L2ARC". In this case he tried a workload with low locality, where just 5% of the blocks where reread again (thus showing cache-busting behaviour). Instead of a pretty minimal performance improvement (as suggested by theory) Charles got an performance improvement by factor 5.

At end this workload was one of this corner cases defying standard tuning knowledge. Normally you would match database block size and storage block size to get optimal performance. But this would hurt performance in this special case because the prefetching of ZFS wouldn't help as less data is cached. Cached? Yes, cached! ZFS doesn't cache just the mysql block you've used in this situation. That wouldn't be sensible. When you already have the data, you can cache them. Let's assume you have a 128 KByte blocksize and the 16 KByte blocksize. So you've read 8 blocks mysql-blocks with one ZFS block. Even in a cache defying workload there is a certain propability that even when you don't use block x again, you will use block x+1 to x+7 while it's in the cache. And this prefetch by mismatching blocksizes is largely responsible for the performance boost, where you didn't expected one. But: Without the mismatching blocksize you wouldn't have read this data into the cache, thus the system would have to go to the disk, thus resulting in lower performance.

Obviously the same effect is true for the normal ARC but as the ARC is normally much smaller than the L2ARC you have a high propability that the prefetched but unused mysql blocks are already evicted from the cache, when you need them. The larger cache provided the SSD reduces the propability that the data gets evicted from (L2)ARC before your workload uses the prefetched blocks.

Posted by Joerg Moellenkamp in English, Solaris at 06:38