Wednesday, September 28. 2016

user_reserve_hint_pct

One of the usual lines in customers /etc/system is the line to limit the the size of the ARC. For a long time you used the zfs_arc_limit parameter for doing so. However with Solaris 11.2 there is a new parameter. It's named user_reserve_hint_pct. It's currently the suggested way to limit the ARC. However it works different than the old parameter. I want to shed some light on this in this blog entry.
At first: Stricly speaking this parameter isn't a parameter of the ZFS. It's part of the VM system. With this parameter you don't set a value for the ARC size but the memory used by the kernel. As the ARC is the only component in the Kernel that can shrink on user demand, you essentially limit the ARC by proxy.

So with user_reserve_hint_pct you set a percentage, the system should reserve for application. So for example when you set 80, it reserves 80% of the physical memory for the applications. And 20% is for the rest. The problem is: As unimportant „the rest" sound, it is the room for very important components of the system.

When you execute a echo „::memstat" | mdb -k" you get an overview how the memory is used. The part highlighted in the output is the stuff you have to place into the 20% that are left over in my user_reserve_hint_pct=80 "calculation".

```
root@solaris:~# echo "::memstat" | mdb -k
Page Summary                Pages             Bytes  %Tot
----------------- ----------------  ----------------  ----
Kernel                     113747           444.3M   11%
ZFS Metadata                 7836            30.6M    1%
ZFS File Data               73142           285.7M    7%
Anon                        34035           132.9M    3%
Exec and libs                1728             6.7M    0%
Page cache                   6159            24.0M    1%
Free (cachelist)             1197             4.6M    0%
Free (freelist)            793293             3.0G   76%
Total                     1048463             3.9G
```

On systems capable of a deferred dump, you may see an addition area, that has to put into the sizing for memory needed by the kernel as well (this line is from a different system than the other example, so the numbers don't add up):

```
Defdump prealloc           147366             1.1G    4%
```

So add Kernel, Defdump prealloc, ZFS Metadata and the amount of ARC ZFS File Data, calculate how much percent the sum is from the physical memory available to the OS instance and substract this value from 100. This value is your value for user_reserve_hint_pct (honestly i would round it down to the next value divisible by 5)

This has some implications: If you size the percentage too low, your application may not start (as it encounters timeouts while waiting on memory) or take longer to start (except right after reboot or your application doesn't allocate a large amount of memory). If you size it to high, you reserve memory for application that may never come and that would be possible better used for caching. If you size it unreasonable high, you starve the kernel from memory, get to ridiculous low ARC sizes and the performance of the system will massively go down the drain. With a wrongly choosen parameter it's perfectly possible to have several hundred gigabyte free, and still have an kernel without enough memory.

But what it to high, what is the correct value? Well, that you really harm the performance of the system is more pronounced in small memory configurations, as 80% on a 1TB may leave ample space for this memory areas, but on a 8 GB system this is a completely different story. So be cautious when getting to a value for user_reserve_hint_pct. I would like to give you some hints.

Only limit ARC when you have applications needing large areas of memory or kernel zones. If you just do NFS, you don't need limit. I have seen ARC limiting to often as some kind of default /etc/system tuning. Think about it
Observe the kernel memory usage with echo „::memstat" | mdb -k" and check from time to time if your setting for user_reserve_hint_pct is still sensible

In the past many people used kstat -p | grep "c_max" in order to find out how large the ARC can grow. When using user_reserve_hint_pct this isn't valid any longer. When you didn't have set it in /etc/system it just show the default value.

The size of ARC is controlled by user_reserve_hint_pct and zfs_arc_max in parallel, the ARC is limited by what ever leads to the lowest value. As the value of user_reserve_hint_pct leads to a smaller size than the one of the zfs_arc_max default, the parameter user_reserve_hint_pct is the active limit. So remember to remove zfs_arc_max when you use user_reserve_hint_pct

There is a good MOS note about this. When tuning this parameter you should consult „Memory Management Between ZFS and Applications in Oracle Solaris 11.x (Doc ID 1663862.1)“ on support.oracle.com for some guidance.

Posted by Joerg Moellenkamp in English, Solaris at 19:18