Subject: Re: Linux Memory Overcommit in OpenVZ
Posted by charlesl on Sat, 12 Jul 2008 20:27:14 GMT
View Forum Message <> Reply to Message

After further tests (using the previous programs I documented above under HN low memory availability), the answer to my earlier question is: yes. I can summarise my conclusions as follows ("memory" = RAM + Swap here):
A VE may allocate any amount it likes up to privvmpages_bar, since the kernel doesn't account this memory and therefore there is no effect on the HN.
If a VE tries to use more memory in an HN-low-memory situation, and is over its guarantee of vmguarpages_bar, its request for allocation is rejected.
If a VE tries to allocate + use memory in an HN-low-memory situation, but is below its vmguarpages_bar, the request will succeed and processes in other containers will be killed to compensate under OOM conditions.
If the previous happens, the HN is out of memory (OOM) and OpenVZ must trim back on containers using more than their guaranteed barriers. It will therefore reduce processes in containers down to their oomguarpages_bar until there is sufficiently free memory (starting with the container in largest excess). Quite how it decides which processes to kill I don't know, nor particularly care (though feel free to add this information!) However, it will kill all processes it sees fit which may include any which purely allocate memory and do not use it (as happened in my tests). Although killing processes made purely of allocated memory seems pointless (since it makes no difference to the free memory in the kernel), in reality all programs will use around 50% their total allocated memory - thus I guess it isn't necessary to discriminate which processes to kill based on the ratio of used/allocated memory on a process-by-process basis. Though this could be an enhancement to the OpenVZ OOM algorithm I suppose - attempt to kill the fewest processes possible, so start by killing those using (rather than allocating) the largest amount of memory.
After an OOM has occurred, VEs may continue to allocate any amount of memory up to vmguarpages_bar without problem - if another OOM occurs (as a result of using this memory) these processes may again be killed.
This is all especially important when using programs that allocate far more than they ever use (and there are a lot of those) - but as long as the server never hits OOM it's all safe and good.
That's all perhaps a bit detailed for most people, but I like to feel I know the system I'm using inside out