Subject: Re: Setting RAM
Posted by kir on Thu, 06 Oct 2005 08:28:32 GMT
View Forum Message <> Reply to Message

All of the UBC parameters are explained and discussed in great details in yet-to-be-released OpenVZ SLM Guide. Below is an except describing the meaning of privvmpages, physpages, vmguarpages and oomguarpages.

Finally, make sure you are checking each VPS configuration you create using vzcgfvalidate(8) utility.

Quote:
vmguarpages: memory allocation guarantee.

This parameter controls how much memory is available to the Virtual Private Server (i.e. how much memory its applications can allocate by malloc(3) or other standard Linux memory allocation mechanisms). The more clients are served or the more "heavy" the application is, the more memory it needs.

The amount of memory that Virtual Private Server's applications are guaranteed to be able to allocate is specified as the barrier of vmguarpages parameter. The current amount of allocated memory space is accounted into privvmpages parameter, and vmguarpages parameter does not have its own accounting. The barrier and the limit of privvmpages parameter impose an upper limit on the memory allocations. The meaning of the limit for the vmguarpages parameter is unspecified in the current version and should be set to the maximal allowed value (2147483647).

If the current amount of allocated memory space does not exceed the guaranteed amount (the barrier of vmguarpages), memory allocations of Virtual Private Server's applications always succeed. If the current amount of allocated memory space exceeds the guarantee but below the barrier of privvmpages, allocations may or may not succeed, depending on the total amount of available memory in the system.

Starting from the barrier of privvmpages, normal priority allocations and, starting from the limit of privvmpages, all memory allocations made by the applications fail.

The memory allocation guarantee (vmguarpages) is a primary tool for controlling the memory available to Virtual Private Servers, because it allows administrators to provide Service Level Agreements -- agreements guaranteeing certain quality of service, certain amount of resources and general availability of the service. The unit of measurement of vmguarpages values is memory pages (4KB on 32-bit Intel-family processes).
The total memory allocation guarantees given to Virtual Private Servers are limited by the physical resources of the computer -- the size of RAM and the swap space.

privvmpages: memory allocation limit.

Privvmpages parameter allows controlling the amount of memory allocated by applications.

The barrier and the limit of privvmpages parameter control the upper boundary of the total size

of allocated memory. Note that this upper boundary doesn't guarantee that the Virtual Private Server will be able to allocate that much memory, neither does it guarantee that other Virtual Private Servers will be able to allocate their fair share of memory. The primary mechanism to control memory allocation is the vmguarpages guarantee.

Privvmpages parameter accounts allocated (but, possibly, not used yet) memory. The accounted value is an estimation how much memory will be really consumed when the Virtual Private Server's applications start to use the allocated memory. Consumed memory is accounted into oomguarpages parameter.

Since the memory accounted into privvmpages may not be actually used, the sum of current privvmpages values for all Virtual Private Servers may exceed the RAM and swap size of the computer.

There should be a safety gap between the barrier and the limit for privvmpages parameter to reduce the number of memory allocation failures that the application is unable to handle. This gap will be used for "high-priority" memory allocations, such as process stack expansion. Normal priority allocations will fail when the barrier if privvmpages is reached.

Total privvmpages should match the physical resources of the computer. Also, it is important not to allow any Virtual Private Server to allocate a significant portion of all system RAM to avoid serious service level degradation for other VPSs.

physpages: total number of RAM pages used by processes in this Virtual Private Server.

For memory pages used by several different Virtual Private Servers (mappings of shared libraries, for example), only a fraction of a page is charged to each Virtual Private Server. The sum of the physpages usage for all Virtual Private Servers corresponds to the total number of pages used in the system by all Virtual Private Servers.

Physpages is an accounting-only parameter currently. In future OpenVZ releases, this parameter will allow to provide guaranteed amount of application memory, residing in RAM and not swappable. For compatibility with future versions, the barrier of this parameter should be set to 0 and the limit to the maximal allowed value (2147483647).

oomguarpages: the guaranteed amount of memory for the case the memory is "over-booked" (out-of-memory kill guarantee).

Oomguarpages parameter is related to vmguarpages. If applications start to consume more memory than the computer has, the system faces an out-of-memory condition. In this case the operating system will start to kill Virtual Private Server's processes to free some memory and prevent the total death of the system. Although it happens very rarely in typical system loads, killing processes in out-of-memory situations is a normal reaction of the system, and it is built into every Linux kernel.

Oomguarpages parameter accounts the total amount of memory and swap space used by the processes of a particular Virtual Private Server. The barrier of the oomguarpages parameter is the out-of-memory guarantee.

If the current usage of memory and swap space (the value of oomguarpages) plus the amount of used kernel memory (kmemsize) and socket buffers is below the barrier, processes in this Virtual Private Server are guaranteed not to be killed in out-of-memory situations. If the system is in out-of-memory situation and there are several Virtual Private Servers with oomguarpages excess, applications in the Virtual Private Server with the biggest excess will be killed first. The failcnt counter of oomguarpages parameter increases when a process in this Virtual Private Server is killed because of out-of-memory situation.

If the administrator needs to make sure that some application won't be forcedly killed regardless of the application's behavior, setting the privvmpages limit to a value not greater than the oomguarpages guarantee significantly reduce the likelihood of the application being killed, and setting it to a half of the oomguarpages guarantee completely pre vents it. Such configurations are not popular because they significantly reduce the utilization of the hardware.

The meaning of the limit for the oomguarpages parameter is unspecified in the current version.

The total out-of-memory guarantees given to the Virtual Private Servers should not exceed the physical capacity of the computer. If guarantees are given for more than the system has, in out-of-memory situations applications in Virtual Private Servers with guaranteed level of service and system daemons may be killed.