

How To Configure Virtual Memory (Swap File) on a VPS

03/28/2014 09:35 PM - Daniel Curtis

Status:	Closed	Start date:	03/28/2014
Priority:	Normal	Due date:	
Assignee:	Daniel Curtis	% Done:	100%
Category:		Estimated time:	1.00 hour
Target version:		Spent time:	1.00 hour

Description

This article will cover the pros and cons of using virtual memory or a swap file (paging), determining if your VPS node already uses virtual memory or paging, the differences between a swap partition and a swap file, information on how to create a swap file, and how to configure the system's "swappiness" (how likely it is to use virtual memory as well as determining the appropriate size to use).

A quote from the Paging (swap file) article on Wikipedia:

In computer operating systems, paging is one of the memory-management schemes by which a computer can store and retrieve data from secondary storage for use in main memory. In the paging memory-management scheme, the operating system retrieves data from secondary storage in same-size blocks called pages. The main advantage of paging over memory segmentation is that it allows the physical address space of a process to be non-contiguous. Before paging came into use, systems had to fit whole programs into storage contiguously, which caused various storage and fragmentation problems.

Paging is an important part of virtual memory implementation in most contemporary general-purpose operating systems, allowing them to use disk storage for data that does not fit into physical random-access memory (RAM).

Introduction - Requirements and Why

What is it and why would I use it?

Whether you have a 512mb VPS node or a 8gb VPS node; Arch, Fedora, CentOS, Debian, or Ubuntu; applications or servers/daemons may need more memory (or sometimes more memory allocated) than you have physically. More specifically in our case, what has been allocated to the virtual server to get the job done.

If you're dealing with any production server, you need to know that if virtual memory is not enabled and your system has no more free memory...then if a program or service - perhaps your web server - needs to allocate more memory, it will fail! Depending on your platform and configuration, this can result in many undesirable or unstable conditions including other applications (i.e. other processes than the one asking for memory) being forced to close to free the needed memory, to it failing and crashing the program - or whole server - entirely.

Because of this, I personally recommend anyone, on nearly any system - be it a VPS node, dedicated server, your Windows PC or Mac or even your Android tablet or phone - should have at least a small amount of virtual memory enabled.

How it Works

Virtual memory allows your system (and thus your apps) additional virtual RAM beyond what your system physically has - or in the case of VPS nodes, what is allocated. It does this by using your disk for the extra, 'virtual' memory and swaps data in and out of system memory and virtual memory as it's needed.

You should be aware that reading/writing from disk (even DigitalOcean's lightning fast SSDs) is at least several times slower than reading/writing from real system RAM. While virtual memory will give you wiggle room, allow more applications/servers to run on one VPS node and guard against out-of-memory errors, it is not a replacement solution in cases where you actually need more memory/to upgrade your virtual server.

While the contents of this article may seem trivial to seasoned admins, I feel it is valuable information for anyone using DigitalOcean's hosting services, especially those new to DigitalOcean, VPS systems in general, or to managing their own servers.

Requirements

The requirements are pretty simple and this technique should work on all distributions and VPS node types - in fact, it will even work on your Android phone or tablet (if you have root & busybox installed).

- a VPS node or virtual server, powered on (or a dedicated server, a linux-based system, etc.)
- root terminal access (ssh, vnc, local)
- the commands free, swapon, swapoff, dd, mkswap will be used, and these all should be available on any platform you use with your VPS node.

Most platforms automatically use and manage virtual memory and will create either a special swap partition or a file on the system partition automatically during installation with the size usually being based on, or a multiple of, available system RAM, eg. 1024mb swap for 512mb RAM.

Pros & Cons

DigitalOcean's "droplets" use SSDs (solid state disks) which are several times faster than regular hard drives and do not suffer from low seek times (caused by a hard drive's head having to physically move across the disk to read data) and low IO requests per second. SSDs can read from multiple areas simultaneously while hard drives can generally only read from one area at a time.

While it is never a good idea - especially with web, mail, db servers - to rely heavily on virtual memory, DO's SSDs help using virtual memory be less painful and more logical.

Pros

- Protection against OOM (out of memory) errors, crashes, memory-related system unpredictability/instability.
- Increases available memory to the system and allows more programs to be run concurrently & more safely
- DigitalOcean's SSD storage reduces VM-associated lag and thrashing while increases paging response times

Cons

- Thrashing is a possibility: when the system is busy and actively using on average more memory than is available physically, the VM system is forced to continually 'swap' program data to and from disk and in and out of RAM as needed. We have probably all seen the result of this in a Windows PC slightly overloaded with the hard drive seemingly grinding away endlessly. It's unbearably slow and not fun. It can, however, be avoided with correct configuration and this is one con that is not as bad here - again due to the speed of DigitalOcean's SSDs.
- Uses disk space, generally depending on system memory. If your VPS node has 512mb, I recommend using 512mb-1.5gb for swap; however that is 512mb-1.5gb less space you will have available on your VPS node's disk.
- It is generally recommended and preferred to use a dedicated disk partition for swap; however this is not possible in a VPS node and we must use a swap file/disk image instead.

Check if Enabled on your VPS

It is entirely possible your configuration already makes use of virtual memory. The commands below will show you how to determine whether it is enabled or not, and if it is, it's size and configuration.

Open a terminal or SSH/VNC to your server - these commands are all performed in a terminal or shell.

Don't forget, to make changes, you need to be root. You can check what user you are logged in with the command whoami. If it does not respond with root or 0, you can type su to start a root shell.

You can check if your VPS node already has virtual memory enabled by typing the "free" command at the prompt in a terminal:

```
free
```

The "free" command shows your system's available physical and virtual memory.

If you have virtual memory enabled already, you can skip ahead to "A Note About Swap Partitions" and then the configuration section. When enabled, the output will look like this:

```
free
```

	total	used	free	shared	buffers	cached
Mem:	361996	360392	1604	0	1988	54376
-/+ buffers/cache:		304028	57968			
Swap:	249896	0	249896			

If it is not enabled, the output will look like this:

```
free
```

	total	used	free	shared	buffers	cached
Mem:	361996	360392	1604	0	2320	54444
-/+ buffers/cache:		303628	58368			
Swap:	0	0	0			

You can also narrow down the output with `free | grep Swap`. This will only show the Swap: line, total, used and free VM. (Remember, by default, `grep` is case sensitive!)

```
free | grep Swap
```

```
Swap:      249896          0      249896
```

Swap Partitions, Swap Files, & Disk Images

A Note About Swap Partitions

Generally, for Linux-based systems, it is preferred to have a dedicated swap partition on your hard disk. Most systems will automatically do this during a normal installation, and if your swap is already configured, it may very well be possible it is setup with a partition. Unfortunately, other than configuring the below `sysctl` settings, resizing swap partitions is outside the scope of this article.

You can however still use the method presented here to add a swap file and increase available virtual memory above and beyond the partitioned swap area. But if a swap is already configured and managed by the system, additional virtual memory may be unnecessary and it may be better to leave the default settings unless you know what you are doing.

What is a Swap File?

Instead of using a dedicated partition, some systems (especially Windows) will store their virtual memory in a special file. This is also an option when partitioning is not possible or feasible. On Linux, this file is actually a disk image.

What is a Disk Image?

A disk image normally includes files and information about those files, as well as the filesystem that contains them.

Examples include your VPS node's snapshots, a function which creates a disk image of your entire VPS node for backup, duplication, migration, etc.

Another prime example is most Linux distributions (or distros) come as disk images, usually in `.iso` format, which allows you to either mount them or burn them to a disk.

Whatever way you use an image, the important thing to remember is that they store the filesystem information, and as such they can be used just like the filesystem they were imaged from - or recreate it entirely.

Creating the Swap File

In the case of a VPS node, we cannot partition our storage, so we need to use a disk image. *NIX operating systems use a dedicated filesystem for swap, this is why as I said above that using a partition is preferred. What we can do is create an empty disk image sized according to required virtual memory, initialize it with a swap filesystem, then turn it on.

First, you need to decide where to put this file. You need to have enough space free on the partition you put it on for whatever amount of MB you use for swap, eg. 512mb of swap creates a ~512mb swap file.

You can use the command `df -h` to view your mounted partitions and filesystems, as well as their sizes & free space.

I recommend placing this file in `/var` and calling it `"swap.img"`. We will `cd` to `/var` and create this file, then set its permissions to 600.

NOTE: It is important for security to set the file permissions to 600 so no other users can read the file directly, otherwise system memory could be read or worse.

```
cd /var
touch swap.img
chmod 600 swap.img
```

Sizing

Now we will size the file. Sizing is important, and the best size will vary depending on your system and use case.

In general, I recommend 1-2x the available system RAM. So, if you have a 512mb VPS node, use 512mb-1gb swap. If you have a 1gb VPS node use 1gb-2gb swap, etc.

This is not a hard and fast rule, for example if you have a 4gb VPS node it may be best to use little (512mb) or no swap at all.

It entirely depends on your use, but these instructions are tailored for a 512mb VPS node. We will use the command `dd` to fill our swap file with zeroes or nothingness to stretch it to the size we need. In this case we are using 1gb or 1024mb. This may take a minute.

```
dd if=/dev/zero of=/var/swap.img bs=1024k count=1000
```

```
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 4.0868896 s, 253 MB/s
```

Preparing the Disk Image

And here, we will initialize the swap filesystem.

```
mkswap /var/swap.img
```

```
Setting up swapspace version 1, size = 1020 GiB
no label, UUID=72761533-8xbe-436l-b07e-c0sabe9cedf3
```

Once that's done, it's ready for use!

Enabling and Disabling Swap

Enable your Swap File

We will use `swapon` to enable it. Upon success there will be no output, but you can check with `free`.

```
swapon /var/swap.img
free
```

	total	used	free	shared	buffers	cached
Mem:	503596	478928	24668	0	38832	102384
-/+ buffers/cache:		337712	165884			
Swap:	1048572	1780	1046792			

You can use `swapoff /var/swap.img` to turn it off.

Enable your Swap File During Boot

Note that `swapon` only enables the file for your current boot; if you reboot it will not come back online unless you either script your `swapon` to run at boot, or change your `/etc/fstab` which in most cases is much easier and is the method we will use here.

All you have to do is add a line to your `/etc/fstab` file to make it ready at boot. Be careful! This file can break your system if it isn't formatted correctly or if it's overwritten. If you put your **swap.img** in **/var**, you can copy/paste the below command with no issue.

```
echo "/var/swap.img    none    swap    sw    0    0" >> /etc/fstab
```

Configuration, Priority and sysctl settings

Once your swap space is online, there is not much that needs to be configured. You could even stop here, if you aren't interested in

the nitty gritty details - once it's enabled generally it will work just fine for most setups.

Priorities

If you are going to be using swap files or partitions spanning multiple devices and device types (unlikely on a droplet), you may want to order the priority for each of these swap areas, either using the faster or more idle storage over slower and/or busier storage. In most cases, you can specify the priority as a parameter to swapon or in your /etc/fstab.

The system will use swap areas of higher priority before using swap areas of lower priority. swapon -p Example:

```
swapon -p 100 /var/swap.img
swapon -p 10 /mnt/SecondDrive/swap.img
```

And, for /etc/fstab priorities can be set using the pri= parameter, like this:

```
/var/swap.img none swap defaults,pri=100 0 0
/mnt/SecondDrive/swap.img none swap defaults,pri=10 0 0
```

sysctl settings (and sysfs)

We will use the command sysctl to change settings dedicated to the Linux virtual memory manager.

There is only one setting I would recommend changing: vm.swappiness. This setting tells the Linux kernel/VM handler how likely it should be to use VM. It is a percent value, between 0 & 100.

If you set this value to 0, the VM handler will be least likely to use any available swap space, and should use all available system memory first. If you set it to 100, the VM handler will be most likely to use available swap space and will try to leave a greater portion of system memory free for use.

I personally recommend using a value of 30%, this should be a happy medium between swapping and system memory. Please note that this value is more of a target goal than a hard rule.

Below is an example on how to modify sysctl settings. For more info, you can type sysctl --help or man sysctl.

```
sysctl -w vm.swappiness=30

vm.swappiness = 30
```

You can also run sysctl -a to list ALL sysctl options (not just VM) or sysctl -a | grep vm..

Or, to view a single setting (you can change the key name after the grep command for each setting):

```
sysctl -a | grep vm.swappiness

vm.swappiness = 30
```

There are many other settings for the Linux VM (vm.*) - though they are outside of the general scope of this article and I generally recommend not changing their settings. I have, however, included a small list of options at the end of this article (in the APPENDIX) that may be worth tweaking or learning about and I added my recommended values (in brackets) for those options.

The Linux kernel website has a full list of options and their use, though not every kernel may implement all options. At the time of writing, this document is already slightly out of date but by no means useless or deprecated. Here's the link:

<https://www.kernel.org/doc/Documentation/sysctl/vm.txt>.

Resources

- <https://www.digitalocean.com/community/articles/how-to-configure-virtual-memory-swap-file-on-a-vps>

History

#1 - 04/05/2014 07:30 AM - Daniel Curtis

- Status changed from In Progress to Closed