
Subject: OpenVZ Density

Posted by [hm2k](#) on Tue, 29 Jul 2008 15:59:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

I'm trying to work out how many users or "units" I could run per server depending on the ram.

I'm looking at getting a new server with either 4GB or 8GB of ram.

The only information I could find on this appeared to be unofficial, and without any substance...

<http://en.wikipedia.org/wiki/OpenVZ#Density>

http://en.opensuse.org/OpenVZ_virtualization#Density

http://wiki.vpslink.com/index.php?title=OpenVZ_docs#Density

None of them state where the information originally came from, so there is no way to know which is the original or in fact where the information came from.

The graph has no real data to support it.

But then this is all just insignificant compared to the real issue...

768mb of ram/120 containers=6.4mb of ram per container, maybe less if you consider overheads.

How does this work?

I don't know any linux distros that will run on 6.4mb of ram, certainly the pre-created templates don't anyway.

Does this information have any real grounds?

It says that if it was 2GB of ram it could run up to 320, that means on 8GB it could run 1280, right?

What if each container was using 128mb of ram, how many could you run then?

$128/6.4=20$, therefore... $1280/20=64$...

Does that mean I can run 64 x 128mb containers on 8GB of ram?

What about overheads?

Is there any real data on density and details on how it can be scaled?

How many containers can you reliably run on a hardware node? What kind of ratio are we talking?

Subject: Re: OpenVZ Density

Posted by [dowdle](#) on Wed, 30 Jul 2008 03:25:09 GMT

Well, if you are one who doesn't want to overcommit your RAM, just take the amount of RAM you have and divide it by the amount of RAM you want each container to have. Oh, btw... your RAM includes swap so don't forget to include that in your calculation.

You might want to run a few `vzsplit` commands and see how it parcels things out.

There isn't a real formula for container density because what services you run in a container and the resources it needs may vary greatly from container to container.

The scenarios you see in some of the examples are using the minimal config template I believe... and like you mentioned, they are pretty worthless for real world deployments... but you should be able to install a minimal OS Template in them... that has almost nothing running when it starts up. I believe I saw one OS Template, don't remember which, that had like 3 processes running at startup. You can probably fit that in ~6MB of RAM.

I have a two containers that basically run minimal services plus apache as a Linux distro repo server... and they seem to use about 48MB of RAM when I look at them... but I'm sure there are times when they use more. I have given them much more RAM than they use... because on the host nodes they are running on there is a surplus of resources.

Subject: Re: OpenVZ Density
Posted by [hm2k](#) on Wed, 30 Jul 2008 11:20:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

dowdle wrote on Tue, 29 July 2008 23:25 Well, if you are one who doesn't want to overcommit your RAM, just take the amount of RAM you have and divide it by the amount of RAM you want each container to have. Oh, btw... your RAM includes swap so don't forget to include that in your calculation.

8000MB RAM/128MB RAM = 62.5 containers...

How does that ensure I don't overcommit?

What do you mean RAM includes swap? How would I include that into the calculation?

dowdle wrote on Tue, 29 July 2008 23:25 You might want to run a few `vzsplit` commands and see how it parcels things out.

<http://man.cx/vzsplit>(

Not sure how this helps...?

dowdle wrote on Tue, 29 July 2008 23:25 There isn't a real formula for container density because what services you run in a container and the resources it needs may vary greatly from container to container.

Not if all the containers are identical.

dowdle wrote on Tue, 29 July 2008 23:25 The scenarios you see in some of the examples are using the minimal config template I believe... and like you mentioned, they are pretty worthless for real world deployments... but you should be able to install a minimal OS Template in them... that has almost nothing running when it starts up. I believe I saw one OS Template, don't remember which, that had like 3 processes running at startup. You can probably fit that in ~6MB of RAM.

The example given apparently runs init, syslogd, crond, sshd and Apache. I suspect it's more than 6MB of RAM, but I haven't tested it.

Perhaps I will...

dowdle wrote on Tue, 29 July 2008 23:25 I have a two containers that basically run minimal services plus apache as a Linux distro repo server... and they seem to use about 48MB of RAM when I look at them... but I'm sure there are times when they use more. I have given them much more RAM than they use... because on the host nodes they are running on there is a surplus of resources.

Even m0n0wall, one of the smallest OS's I know requires 64MB of ram to run... It is however, FreeBSD based, not linux...

For linux, Damn Small Linux requires 8MB RAM to run too...

But then neither of these are even supported by OpenVZ...

The smallest precreated template that openvz supplies is mandriva-2006-i386-minimal.tar.gz - it's system requirements are minimum of 256mb of ram... See:
http://www.mandriva.com/archives/en/linux/spring/system_requirements.html

Beyond that is centos-4-i386-minimal.tar.gz, but I was unable to find the system requirements for that or it's rhel counterpart.

In conclusion I have yet to make any progress on resolving this.

Subject: Re: OpenVZ Density
Posted by [JimL](#) on Wed, 30 Jul 2008 11:56:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

As a practical matter I think you can over commit ram a bunch. Remember that in most cases only a small fraction of the address space of any program needs to be in real memory at any time. The OS takes care of paging stuff in and out of memory for you. You may reach a point by overcommitting that you are thrashing the system, but you can surely overcommit at least 2 - 3 times that and probably a lot more, but that'll depend on your applications. The only real way to know is to try it. Just load a system up with a bunch of ves and put synthetic loads on them. Then bring them up one at a time until the system gets sluggish or worse.

There's no fixed formula that anyone will commit to since it all depends on what the ve's are doing.

Jim.

Subject: Re: OpenVZ Density

Posted by [dowdle](#) on Wed, 30 Jul 2008 12:56:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

With regards to OpenVZ and memory related system resource parameters, swap is included as if it were part of RAM. How would you include it in your calculations? Well, you'd add however big your host node swap is to your total physical RAM. You can add, right?

Why would you have to view a web page to read the vzsplint man page? How does it help? Well, you can see what values it gives for the various parameters for different numbers of containers. You can also give it an impossible value for the number of containers you want (say 99999999) and it'll tell you what it thinks is the maximum capacity of your machine. Of course the maximum capacity of your machine is just that, the max... and not a recommended number for real world deployment.

You say your containers will be all alike... software and process / configuration-wise that might be true... but that may not always be true load wise... unless you yourself happen to be the only consumer of your containers services.

A container is a group of child processes under a unique init process, and while you can run different Linux distributions in a container, you only need the resources necessary to run the processes... and don't need to follow a particular distribution's system requirements for memory. That is to say that if a distribution says it requires a minimum of 256MB of RAM to run, that does not hold true with a container running that distribution... especially when the OpenVZ flavor of that distribution as provided by a pre-created OS Template has been trimmed down considerably from a standard distribution install. For example, most distributions these days have a GUI installer and install X11, and set the default runlevel to 5 for a graphical login. None of the pre-created OS templates provided by the OpenVZ project include X11, they do not use the distribution's native installer, and there isn't a kernel running inside of the container so all of the resources needed by a kernel and all of the threads are not there. Since OpenVZ (by default anyway) isn't trying to provide you with a desktop OS a lot of the standard services of many distros are not required... like an X font server, hal, sound server, etc, etc, etc. A container really needs to run only those processes you want. Given all of these factors, a container can use substantially less memory / resources than a full distribution running on a physical machine or under machine / hardware virtualization.

Your comment about the smallest pre-created OS Template (Mandriva) has no bearing. The size of the OS Template represents files on the disk and not necessarily running processes that are going to be started up. I haven't installed a wide variety of different distros lately... but from my past experiences many of the "minimal" OS Templates do not include apache... or much beyond the bare minimum for a running system. Like I said, I've seen a few containers created from

"minimal" pre-created OS Templates that only have 2 or 3 processes running by default when they start up. Since I've not installed the Mandriva OS Template you mention, I looked and it is 28MB in size. That is pretty darn minimal I must say if you compare that to the stock Mandriva install media size. I don't know if that includes Apache or not... but I would be surprised if it did. Once you get a little more experience playing around with different OS Templates you'll start to realize that I'm not making this up... and some of them do use a tiny amount of RAM. I'm not saying that a container made from a minimal OS Template is that useful for anything but they can be handy for testing... and answering questions like... what's the maximum number of containers that is doable?

How do you ensure you don't overcommit? You don't configure however many containers you create to have access to more memory (again, physical and swap together) than you have. There are three camps... those who think overcommitment of resources should become the 8th deadly sin, those to don't care, and those who routinely overcommit resources. I fall into the later and I've been running OpenVZ for about 3 years now and haven't run into any significant problems as a result of resource overcommitment. Of course that's with the containers I run and the loads I have. YMMV.

You haven't made any progress solving this? Well then.. STFU and do some testing. You'll learn more by doing than bitching. Am I being harsh? Perhaps... but it seems to me that I've spent quite a bit of time attempting to answer your questions and I've only gotten whining thus far.

I do recommend that you play around with OpenVZ for a little while, do some testing, and gain a little confidence with it before trying to create a production environment. Read the Users Guide (which is admittedly a little outdated), the man pages for the various tools, and any wiki articles you find related to questions that come up while on you journey. I wish you luck on that journey and will be happy to answer any additional questions you might come up with in the future.

Subject: Re: OpenVZ Density
Posted by [dowdle](#) on Wed, 30 Jul 2008 13:03:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

gozargah,

Why are you asking about a quota error in the middle of this topic (which is completely unrelated)? I'm not going to even attempt to provide an answer here. If you do a few searches on the forum you'll probably find reference to the problem and how to fix it. I have encountered that once or twice in the last 3 years but I don't recall the specifics. After doing a few searches if you are still unable to find a fix for your issue, create a new forum topic.

Subject: Re: OpenVZ Density
Posted by [hm2k](#) on Wed, 30 Jul 2008 13:11:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

All you had to say is that you don't know.

Subject: Re: OpenVZ Density
Posted by [dowdle](#) on Wed, 30 Jul 2008 13:33:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

The answer isn't that I don't know... the answer has been clearly given by two people... and to review:

Q: What is the formula for computing maximum container density on a given piece of hardware?

A: There isn't one.

An obvious basic formula would be:

(physical RAM + swap) divided by (amount of ram [or average] you want each container to have)
= number of containers

Of course that's just for memory... and given the fact that OpenVZ has about 20-ish resource parameters, 4 or so of which are memory related, that formula would only answer one piece. I'll admit that the memory resource parameters are quite complex and somewhat confusing. There is a wiki page on resource management where they discuss the various memory parameters and give formulas showing what each represents... and they are quite complicated... not by mathematical standards... but by wrapping your head around it standards.

I don't usually come off sounding like a smart ass... but for whatever reason... I have in this thread. Sorry. I hope you can overlook that and make some progress.

Subject: Re: OpenVZ Density
Posted by [hm2k](#) on Wed, 30 Jul 2008 13:44:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

I get the feeling of blind leading the blind.

It's clear you have a lack of understanding of the original points I made.

Therefore on that note, I don't think you can really assist me further.

Hopefully someone more credible to the project may be able to.

Subject: Re: OpenVZ Density
Posted by [dowdle](#) on Wed, 30 Jul 2008 14:27:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'll ignore your an eye for an eye... and offer a clarification:

I have seen the same density formula question asked several times over the last couple of years... and the answer has always been that there isn't one. That doesn't mean you or someone else

can't come up with one... but due to the complexities previously mentioned, I think it has discouraged people from trying.

With regards to me ignoring your points or me having a lack understanding of them, I disagree.

I have about 20 or so containers on 2 physical hosts... and their resource uses vary greatly even though they are all the same Linux distribution (CentOS) running on the same host node distribution (RHEL and CentOS). They vary from a simple webserver with no additional system accounts, to a system that runs a telnet bbs that sometimes has a number of online users, to Zimbra Collaboration Suite which is an enormous system which includes a fantastic but very bloated Java server app and says that it should always be run on a dedicated server.

Anyhoo, I must admit that I've had a lot different situation than you where you want a lot of all-the-same containers. I could be mistaken, don't think I am... but feel free to come back and correct me if you find what you are looking for. If there is a useful formula for container density calculation, I'd like to know what it is too.

Subject: A better answer

Posted by [dowdle](#) on Wed, 30 Jul 2008 14:54:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Ok, the formula you gave (slightly modified to take into account swap) is a good formula if you do not want to overcommit resources. How does that ensure you don't overallocate resources? Well, if you don't give your containers access to more RAM than you have, whenever they try to use more than they have been given the OpenVZ kernel will cause a resource allocation failure and the User Beancounter (UBC) failcnt for the appropriate resource will be incremented.

You can get higher density by overcommitting resources... which works well for many people... but you really have to watch your UBCs and tweak your container's configs to avoid UBC failcnts.

The typical user that has a lot of all-the-same containers are hosting systems... and given the competitive nature of them... I haven't see much in the way of papers written that show best practices for density.

There are two theories on config tweaking. 1) Start low and tweak up. That certainly gives you the highest density... but will usually lead to any load spikes causing failcnts. 1) Starting high and tweaking down or just giving more resources than a container typically uses to better allow for load spikes not causing failcnts.

I'd have to say that the "make one config for a lot of containers all alike for maximum density approach" has potentially lead to the perception that VPSes, containers, or OS Virtualization doesn't work well. But what else are hosting providers going to do when the have to make configurations that match up to a small number service plans / products? As long as you give a container access to a reasonable amount of resources, you really shouldn't have a problem... so use the basic formula.

Subject: Re: OpenVZ Density
Posted by [dowdle](#) on Wed, 30 Jul 2008 16:25:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Just for fun, I downloaded the Mandriva OS Template and created a container from it. Here's the processes it runs by default and the amount of RAM it uses:

```
[root@openvz2 cache]# vzctl enter devel2  
entered into VE 199055
```

```
[root@devel2 /]# pstree -nup
```

```
[root@devel2 /]# top -b -n 1 | head -5 | tail -2  
Mem: 9982512k total, 4544k used, 9977968k free, 0k buffers  
Swap: 0k total, 0k used, 0k free, 0k cached
```

So, without my vzctl, bash, pstree processes there are only 4 processes. The top output says that it is using about 4.5MB of RAM... and again that includes the additional processes I'm using to measure with. I believe the same would apply for most of the OS Templates marked as "minimal".

Why have such a small system? Some people have unique needs and only want to do one service (for isolation) so they don't want much. Less stuff installed by default means less stuff to update. The minimal ones don't even have man installed.

I've seen some people complain and call even the minimal OS Templates "bloated" but those are the folks creating a lot of special purpose containers.

Subject: Re: OpenVZ Density
Posted by [hm2k](#) on Wed, 30 Jul 2008 16:34:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

You're asking me like I wrote the original research.

The reason why I am here is because I am questioning it.

Subject: Re: OpenVZ Density
Posted by [dowdle](#) on Wed, 30 Jul 2008 19:41:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

hm2k wrote on Tue, 29 July 2008 09:59[[urlhttp://en.wikipedia.org/wiki/OpenVZ#Density](http://en.wikipedia.org/wiki/OpenVZ#Density)]

I believe I've seen the graph and info that is there came from a presentation I've seen given by Kir (the OpenVZ Project Manager)... and I've also seen it in various other presentations given by various Parallels folks in the last couple of years. So that's the source of it. I did a search and I can't seem to find any copies of those presentations online... although I did find a few broken links.

In an effort to duplicate the findings... I just created a CentOS 5 i386 default container... using an updated OS Template I made for contrib (uptodate as of last Sunday). I turned off sendmail, xinetd, edited ksyslogd out of the syslogd startup script (because it isn't needed), and reconfigured apache to only start up one server (all stock modules loaded) and here's what I got:

```
[root@openvz2 ~]# vzlist
  VEID   NPROC STATUS IP_ADDR   HOSTNAME
  199055     6 running 153.90.199.55  devel2.cs.montana.edu
[root@openvz2 ~]# vzctl enter devel2
entered into VE 199055
[root@devel2 /]# pstree -nup
init(1)-+-syslogd(28571)
  |-sshd(28596)
  |-crond(28619)
  |-httpd(29747)---httpd(29748,apache)
  `--vzctl(29757)---bash(29758)---pstree(29772)
[root@devel2 /]# top -b -n 1 | head -5 | tail -2
Mem: 9982512k total,  9856k used, 9972656k free,    0k buffers
Swap:   0k total,    0k used,    0k free,    0k cached
```

That's the closest I could get... which is a little over 9MB of RAM being used.

Since the material is a little older maybe it was taken from a distro running an older apache (like the 1.3 series) that uses less RAM. Anyway, my results with the latest stuff weren't that far off.

If I leave everything turned on and just go with the default (sendmail, xinetd, default number of apaches, etc), the container takes up about 19MB of RAM. I have yum installed in my containers (many of the pre-created OS Templates do not include it and that is bad) and if I were to run that to do updates, of course the container would use a lot more RAM.

So, does that make you feel any better?
