Subject: recommended swap space
Posted by Dietmar Maurer on Thu, 06 Dec 2007 17:10:53 GMT
View Forum Message <> Reply to Message

Hi all,

documentation says swap should be pyhsical RAM*2.

I wonder if that is still recommended for newer machines which has
really much RAM - lets say 16GB.

Should I really use 32GB swap space for such machine?

- Dietmar

Subject: Re: recommended swap space
Posted by Dariush Pietrzak on Thu, 06 Dec 2007 18:36:29 GMT
View Forum Message <> Reply to Message

> documentation says swap should be pyhsical RAM*2.
 This rule was created when HDD were many times faster compared to RAM then
they are today(and when programs needed way more virtual space in relation
to what could be available).
 Imagine how long it would take read/write 32G from HDD..., also, most
really large requirements for ram come from various layers of essentially
caching. In the 90s it was quite typical to run servers with half of
virtual space permanently swapped out (64M ram machine, with 128M swap,
and never less then 64M swap used, 512M machine with 1G swap and never less
then 512M of swap used etc..).
 It was possible to do that, because of large amounts of inactive code/very
rarely called code in programs, thus you could safely swap out half of the
code and safely assume that it won't ever be needed.
 These days, most of ram goes to data, not to code, and alot of stuff works
like hash tables - every single particular page of data is accessed relatively
infrequently (thus, it would be swapped out) but there are a lot of such
accesses and if you wouldn't want to make them wait for HDD.

 As a rule of a thumb, I assume that 10000rpm HDD can't handle swap larger
then ~512M-1G and 1500rpm HDD shouldn't be burdened with more then 1-2G of
swap.

> Should I really use 32GB swap space for such machine?
 If you know that your machine will still run with ~30G swapped out...

--
Key fingerprint = 40D0 9FFB 9939 7320 8294 05E0 BCC7 02C4 75CC 50D9
 Total Existance Failure

Subject: Re: recommended swap space
Posted by dev on Fri, 07 Dec 2007 09:13:24 GMT
View Forum Message <> Reply to Message

I wouldn't agree with you that easily.
Usually applications active data set (i.e. data which are frequently accessed)
is ~10-50% of the whole RSS.
Thus it allows to swap out most of apps memory w/o much performance penalty.

I've just checked a couple of production nodes I have access to and see for
example following numbers:
8GB RAM, 2GM swap total, 33 big VEs => 1.2GB swap used
8GB RAM, 16GB swap total, 125 VEs => ~8GB swap used
both machines work fine and statistics show normal latencies.

I would express it as:
1. swap can hardly be used for memory hungry applications like
   math calculations and similar stuff doing aggressive memory access
2. for common kind of applications swap allows to increase effective memory
   size quite significantly (compares to RAM size) and it also allows
   to provide overcommits for memory, i.e. have a summary of VE limits
   bigger then total RAM. if some of your VEs will have a bust in memory consumtion
   you always know that SWAP will help to handle it.

So depending on this one can select swap size:
a) If you don't know your workload and plan to overcommit your machine,
   swap size 1-2RAM is a good option.
b) If on the other hand you know that your apps will never be able
   to consume whole RAM, then swap can be minimal.

Thanks,
Kirill

Dariush Pietrzak wrote:
>>documentation says swap should be pyhsical RAM*2.
>
> This rule was created when HDD were many times faster compared to RAM then
> they are today(and when programs needed way more virtual space in relation
> to what could be available).
> Imagine how long it would take read/write 32G from HDD..., also, most
> really large requirements for ram come from various layers of essentially
> caching. In the 90s it was quite typical to run servers with half of
> virtual space permanently swapped out (64M ram machine, with 128M swap,
> and never less then 64M swap used, 512M machine with 1G swap and never less
> then 512M of swap used etc..).
> It was possible to do that, because of large amounts of inactive code/very
> rarely called code in programs, thus you could safely swap out half of the
> code and safely assume that it won't ever be needed.

> These days, most of ram goes to data, not to code, and alot of stuff works
> like hash tables - every single particular page of data is accessed relatively
> infrequently (thus, it would be swapped out) but there are a lot of such
> accesses and if you wouldn't want to make them wait for HDD.
>
> As a rule of a thumb, I assume that 10000rpm HDD can't handle swap larger
> then ~512M-1G and 1500rpm HDD shouldn't be burdened with more then 1-2G of
> swap.
>
>
>>Should I really use 32GB swap space for such machine?
>
> If you know that your machine will still run with ~30G swapped out...
>

Subject: Re: recommended swap space
Posted by dev on Fri, 07 Dec 2007 09:19:50 GMT
View Forum Message <> Reply to Message

Dariush Pietrzak wrote:
>>documentation says swap should be pyhsical RAM*2.
>
> This rule was created when HDD were many times faster compared to RAM then
> they are today(and when programs needed way more virtual space in relation
> to what could be available).
> Imagine how long it would take read/write 32G from HDD..., also, most
> really large requirements for ram come from various layers of essentially
> caching. In the 90s it was quite typical to run servers with half of
> virtual space permanently swapped out (64M ram machine, with 128M swap,
> and never less then 64M swap used, 512M machine with 1G swap and never less
> then 512M of swap used etc..).
> It was possible to do that, because of large amounts of inactive code/very
> rarely called code in programs, thus you could safely swap out half of the
> code and safely assume that it won't ever be needed.

BTW, code is mapped directly from binaries of the filesystem and never swaps out.
Instead, when you need some memory in Linux, kernel simple drops/frees such code pages
because it knows that it always can be loaded from the filesystem executable/library again.

Effectively this works as a swapping out, except for the fact that "write" (swap out)
operation is ommited and only "read" (swap in) is left.
So in some sense, swap out of code always works, even when you have no swap at all :)
And the fact that you don't notice this is another proof that swapout is not that
bad/slow thing for active memory working sets.

> These days, most of ram goes to data, not to code, and alot of stuff works
> like hash tables - every single particular page of data is accessed relatively

> infrequently (thus, it would be swapped out) but there are a lot of such
> accesses and if you wouldn't want to make them wait for HDD.

When we were investigating apache live migration, it turned out,
that it's active memory set is ~20% of the whole RSS.
i.e. 80% of apache memory can be safely swaped out cause too rarely accessed.

Thanks,
Kirill
