Subject: Re: [RFC][PATCH 2/7] RSS controller core
Posted by Herbert Poetzl on Mon, 12 Mar 2007 22:41:29 GMT
View Forum Message <> Reply to Message

On Mon, Mar 12, 2007 at 11:42:59AM -0700, Dave Hansen wrote:
> How about we drill down on these a bit more.
>
> On Mon, 2007-03-12 at 02:00 +0100, Herbert Poetzl wrote:
> > - shared mappings of 'shared' files (binaries
> > and libraries) to allow for reduced memory
> > footprint when N identical guests are running
>
> So, it sounds like this can be phrased as a requirement like:
>
> "Guests must be able to share pages."
>
> Can you give us an idea why this is so?

sure, one reason for this is that guests tend to
be similar (or almost identical) which results
in quite a lot of 'shared' libraries and executables
which would otherwise get cached for each guest and
would also be mapped for each guest separately

> On a typical vserver system,

there is nothing like a typical Linux-VServer system :)

> how much memory would be lost if guests were not permitted
> to share pages like this?

let me give a real world example here:

- typical guest with 600MB disk space
- about 100MB guest specific data (not shared)
- assumed that 80% of the libs/tools are used

gives 400MB of shared read only data

assumed you are running 100 guests on a host,
that makes ~39GB of virtual memory which will
get paged in and out over and over again ...

.. compared to 400MB shared pages in memory :)

> How much does this decrease the density of vservers?

well, let's look at the overall memory resource

function with the above assumptions:

 with sharing: f(N) = N*80M + 400M
 without sharing: g(N) = N*480M

so the decrease N->inf: g/f -> 6 (factor)

which is quite realistic, if you consider that
there are only so many distributions, OTOH, the
factor might become less important when the
guest specific data grows ...

> > - virtual 'physical' limit should not cause
> >   swap out when there are still pages left on
> >   the host system (but pages of over limit guests
> >   can be preferred for swapping)
>
> Is this a really hard requirement?

no, not hard, but a reasonable optimization ...

let me note once again, that for full isolation
you better go with Xen or some other Hypervisor
because if you make it work like Xen, it will
become as slow and resource hungry as any other
paravirtualization solution ...

> It seems a bit fluffy to me.

most optimizations might look strange at first
glance, but when you check what the limitting
factors for OS-Level virtualizations are, you
will find that it looks like this:

(in order of decreasing relevance)

 - I/O subsystem
 - available memory
 - network performance
 - CPU performance

note: this is for 'typical' guests, not for
number crunching or special database, or pure
network bound applications/guests ...

> An added bonus if we can do it, but certainly not the
> most important requirement in the bunch.

nope, not the _most_ important one, but it
all summs up :)

> What are the consequences if this isn't done? Doesn't
> a loaded system eventually have all of its pages used
> anyway, so won't this always be a temporary situation?

let's consider a quite limited guest (or several
of them) which have a 'RAM' limit of 64MB and
additional 64MB of 'virtual swap' assigned ...

if they use roughly 96MB (memory footprint) then
having this 'fluffy' optimization will keep them
running without any effect on the host side, but
without, they will continously swap in and out
which will affect not only the host, but also the
other guests ...

> This also seems potentially harmful if we aren't able
> to get pages *back* that we've given to a guest.

no, the idea is not to keep them unconditionally,
the concept is to allow them to stay, even if the
guest has reached the RSS limit and a 'real' system
would have to swap pages out (or simply drop them)
to get other pages mapped ...

> Tasks can pin pages in lots of creative ways.

sure, this is why we should have proper limits
for that too :)

> > - accounting and limits have to be consistent
> >   and should roughly represent the actual used
> >   memory/swap (modulo optimizations, I can go
> >   into detail here, if necessary)
>
> So, consistency is important, but is precision?

IMHO precision is not that important, of course,
the values should be in the same ballpark ...

> If we, for instance, used one of the hashing schemes,
> we could have some imprecise decisions made but the
> system would stay consistent overall.

it is also important that the lack of precision
cannot be exploited to allocate unreasonable

ammounts of resources ...

at least Linux-VServer could live with +/- 10%
(or probably more) as I said, it is mainly used
for preventing DoS or DoR attacks ...

> This requirement also doesn't seem to push us in the
> direction of having distinct page owners, or some
> sharing mechanism, because both would be consistent.

> > - OOM handling on a per guest basis, i.e. some
> >   out of memory condition in guest A must not
> >   affect guest B
>
> I'll agree that this one is important and well stated
> as-is. Any disagreement on this one?

nope ...

best,
Herbert

> -- Dave

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers