Subject: Re: Linux-VServer example results for sharing vs. separate mappings ...
Posted by Herbert Poetzl on Sun, 25 Mar 2007 14:40:12 GMT
View Forum Message <> Reply to Message

On Sat, Mar 24, 2007 at 08:29:51PM -0800, Andrew Morton wrote:
> On Sun, 25 Mar 2007 04:21:56 +0200 Herbert Poetzl <herbert@13thfloor.at> wrote:
>
> > > a) slice the machine into 128 fake NUMA nodes, use each node as the
> > >   basic block of memory allocation, manage the binding between these
> > >   memory hunks and process groups with cpusets.
> >
> > 128 sounds a little small to me, considering that we
> > already see 300+ Guests on older machines ....
> > (or am I missing something here?)
>
> Yes, you're missing something very significant. I'm talking about
> resource management (ie: partitioning) and you're talking about
> virtual servers. They're different applications, with quite a lot
> in common.

okay, but please explain how that what you call
'resource management' is different from the resource
accounting and resource limits used in Linux-VServer
(and please ignore the implementation details :)

> For resource management, a few fives or tens of containers is probably
> an upper bound.
>
> An impementation needs to address both requirements.
>
> > >   This is what google are testing, and it works.
> > >
> > > b) Create a new memory abstraction, call it the "software zone",
> > >   which is mostly decoupled from the present "hardware zones".
> > >   Most of the MM is reworked to use "software zones". The
> > >   "software zones" are runtime-resizeable, and obtain their pages
> > >   via some means from the hardware zones. A container uses a
> > >   software zone.
> > >
> > > c) Something else, similar to the above. Various schemes can be
> > >   envisaged, it isn't terribly important for this discussion.
> >
> > for me, the most natural approach is the one with
> > the least impact and smallest number of changes
> > in the (granted quite complex) system: leave
> > everything as is, from the 'entire system' point
> > of view, and do adjustments and decisions with the
> > additional Guest/Context information in mind ...

> >
> > e.g. if we decide to reclaim pages, and the 'normal'
> > mechanism would end up with 100 'equal' candidates,
> > the Guest badness can be a good additional criterion
> > to decide which pages get thrown out ...
> >
> > OTOH, the Guest status should never control the
> > entire system behaviour in a way which harms the
> > overall performance or resource efficiency
>
> On the contrary - if one container exceeds its allotted resource, we
> want the processes in that container to bear the majority of the cost
> of that. Ideally, all of the cost.

well, I totally agree here, the _container_ should bear
the cost, but not the entire system!

I'm fine (and I mentioned that several times) with
penalizing Guests which are over limit either in general
or on the actual resource access, but IMHO it is a very
bad idea to hurt the entire system like a partitioning
system would do ...

> > > All doable, if we indeed have a demonstrable problem
> > > which needs to be addressed.
> >
> > all in all I seem to be missing the 'original problem'
> > which basically forces us to do all those things you
> > describe instead of letting the Linux Memory System
> > work as it works right now and just get the accounting
> > right ...
>
> The VM presently cannot satisfy resource management requirements,
> because piggy activity from one job will impact the performance
> of all other jobs.

that is correct (at least to some extend)

> > > > note that the 'frowned upon' accounting Linux-VServer
> > > > does seems to work for those cases quite fine .. here
> > > > the relevant accounting/limits for three guests, the
> > > > first two unified and started in strict sequence, the
> > > > third one completely separate
> > > >
> > > > Limit current   min/max   soft/hard hits
> > > > VM: 41739   0/ 64023   -1/   -1   0
> > > > RSS: 8073   0/   9222   -1/   -1   0
> > > > ANON: 3110   0/   3405   -1/   -1   0

```
> > > > RMAP:   4960    0/   5889    -1/    -1    0
> > > > SHM:    7138    0/   7138    -1/    -1    0
> > > >
> > > > Limit  current    min/max     soft/hard  hits
> > > > VM:    41738    0/  64163    -1/    -1    0
> > > > RSS:    8058    0/   9383    -1/    -1    0
> > > > ANON:   3108    0/   3505    -1/    -1    0
> > > > RMAP:   4950    0/   5912    -1/    -1    0
> > > > SHM:    7138    0/   7138    -1/    -1    0
> > > >
> > > > Limit  current    min/max     soft/hard  hits
> > > > VM:    41738    0/  63912    -1/    -1    0
> > > > RSS:    8050    0/   9211    -1/    -1    0
> > > > ANON:   3104    0/   3399    -1/    -1    0
> > > > RMAP:   4946    0/   5885    -1/    -1    0
> > > > SHM:    7138    0/   7138    -1/    -1    0
```
> > >
> > > Sorry, I tend to go to sleep when presented with rows and rows of
> > > numbers. Sure, it's good to show the data but I much prefer it if the
> > > sender can tell us what the data means: the executive summary.
> >
> > sorry, I'm more the technical person and I hate
> > 'executive summaries' and similar stuff, but the
> > message is simple and clear: accouting works even
> > for shared/unified guests, all three guests show
> > reasonably similar values ...
>
> I don't see "accounting" as being useful for resource managment.
> I mean, so we have a bunch of numbers - so what?

IMHO it is the basis for resource management, if you cannot
account the used resources, you will not be able to limit
them in a proper way, no?

> The problem is: what do we do when the jobs in a container exceed
> their allotment?

there are two different kinds of resource, the ones you
basically 'give' on request and the ones which are 'taken'
when there is need. the former ones can easily be checked
with some limit and simply be denied when crossing over.
the latter ones should IMHO penalize the Guest in such way
that the 'good' Guests are not affected or even benefit
from the 'bad' ones ...

This, IMHO is the real challenge in OS-level isolation
and virtualization ... YMMV

> With zone-based physical containers we already have pretty much all
> the accounting we need, in the existing per-zone accounting.

maybe, but for example, saying that we can only have
128 limited Guests, because all you ever will have is like
a dozen 'partitions' makes the entire approach quite useless
for the typical scenarios Linux-VServer and OpenVZ is used
for ... note: we have users with 300+ limited Guests on
larger machines ....

HTC,
Herbert