Subject: Re: Linux-VServer example results for sharing vs. separate mappings ...
Posted by Herbert Poetzl on Sat, 24 Mar 2007 18:38:06 GMT
View Forum Message <> Reply to Message

On Fri, Mar 23, 2007 at 09:42:35PM -0800, Andrew Morton wrote:
> On Fri, 23 Mar 2007 20:30:00 +0100 Herbert Poetzl <herbert@13thfloor.at> wrote:
>
> >
> > Hi Eric!
> > Hi Folks!
> >
> > here is a real world example result from one of my tests
> > regarding the benefit of sharing over separate memory
> >
> > the setup is quite simple, a typical machine used by
> > providers all over the world, a dual Pentium D 3.2GHz
> > with 4GB of memory and a single 160GB SATA disk running
> > a Linux-VServer kernel (2.6.19.7-vs2.2.0-rc18)
> >
> > the Guest systems used are Mandriva 2007 guests with
> > syslog, crond, sshd, apache, postfix and postgresql
> > installed and running (all in all 17 processes per guest)
> >
> > the disk space used by one guests is roughly 148MB
> >
> > in addition to that, a normal host system is running
> > with a few daemons (like sshd, httpd, postfix ...)
> >
> >
> > the first test setup is starting 200 of those guests
> > one after the other and measuring the memory usage
> > before and after the guest did start, as well as
> > recording the time used to start them ...
> >
> > this is done right after the machine was rebooted, in
> > one test with 200 separate guests (i.e. 200 x 148MB)
> > and in a second run with 200 unified guests (which
> > means roughly 138MB of shared files)
>
> Please define your terms.
> What is a "separated guest", what is a "unified guest"
> and how do they differ?

separated guests are complete Linux Distributions which
do not share (filesystem wise) anything with any other
guest ... i.e. all files and executables have to be
paged in and get separate mappings (and thus separate
memory)

unified guests use a mechanism we (Linux-VServer) call
'unification' which can be considered an advanced form
of hard linking (i.e. we add special flags to protect
those hard links from modification. such a file is
copied on demand (CoW Link Breaking) on the first attempt
to be modified (attributes or content)

so although all guests use a separate namespace (i.e.
will have separate dentries) they share most of the files
(those which are not modified) via inodes (and the inode
cache of course)

> If a "separated" guest is something in which separate
> guests will use distinct physical pages to cache the
> contents of /etc/passwd (ie: a separate filesystem
> per guest) then I don't think that's interesting
> information, frankly.

well, you didn't bother to answer my questions regarding
your suggested approach yet, and as I am concerned that
some of the suggested approaches sacrifice performance
and resource sharing/efficiency for simplicity or (as
we recently had) 'ability to explain it to the customer'
I thought I provide some data how much resource sharing
can help (the overall performance)

> Because nobody (afaik) is proposing that pagecache be
> duplicated across instances in this fashion.
>
> We obviously must share pagecache across instances -
> if we didn't want to do that then we could do something
> completely dumb such as use xen/kvm/vmware/etc ;)

exactly my words ...

> The issue with pagecache (afaik) is that if we use
> containers based on physical pages (an approach which
> is much preferred by myself) then we can get in a
> situation where a pagecache page is physically in
> container A, is not actually used by any process in
> container A, but is being releatedly referenced by
> processes which are in other containers and hence
> unjustly consumes resources in container A.

> How significant a problem this is likely to be I do
> not know.

well, with a little imagination, you can extrapolate
that from the data you removed from this email, as one
example case would be to start two unified guests one
after the other, then shutdown almost everything in
the first one, you will end up with the first one being
accounted all the 'shared' data used by the second one
while the second one will have roughly the resources
accounted the first one actually uses ...

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
```

> And there are perhaps things which we can do about it.

best,
Herbert

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers