Subject: Re: [RFC][PATCH 2/7] RSS controller core
Posted by Herbert Poetzl on Mon, 12 Mar 2007 01:00:39 GMT
View Forum Message <> Reply to Message

On Sun, Mar 11, 2007 at 04:51:11AM -0800, Andrew Morton wrote:
> > On Sun, 11 Mar 2007 15:26:41 +0300 Kirill Korotaev <dev@sw.ru> wrote:
> > Andrew Morton wrote:
> > > On Tue, 06 Mar 2007 17:55:29 +0300
> > > Pavel Emelianov <xemul@sw.ru> wrote:
> > >
> > >
> > > >+struct rss_container {
> > > >+ struct res_counter res;
> > > >+ struct list_head page_list;
> > > >+ struct container_subsys_state css;
> > > >+};
> > > >+
> > > >+struct page_container {
> > > >+ struct page *page;
> > > >+ struct rss_container *cnt;
> > > >+ struct list_head list;
> > > >+};
> > >
> > >
> > > ah. This looks good. I'll find a hunk of time to go through
> > > this work and through Paul's patches. It'd be good to get both
> > > patchsets lined up in -mm within a couple of weeks. But..
> > >
> > > We need to decide whether we want to do per-container memory
> > > limitation via these data structures, or whether we do it via
> > > a physical scan of some software zone, possibly based on Mel's
> > > patches.
> > i.e. a separate memzone for each container?
>
> Yep. Straightforward machine partitioning. An attractive thing is that
> it 100% reuses existing page reclaim, unaltered.
>
> > imho memzone approach is inconvinient for pages sharing and shares
> > accounting. it also makes memory management more strict, forbids
> > overcommiting per-container etc.
>
> umm, who said they were requirements?

well, I guess all existing OS-Level virtualizations
(Linux-VServer, OpenVZ, and FreeVPS) have stated more
than one time that _sharing_ of resources is a central
element, and one especially important resource to share
is memory (RAM) ...

if your aim is full partitioning, we do not need to
bother with OS-Level isolation, we can simply use
Paravirtualization and be done ...

> > Maybe you have some ideas how we can decide on this?
>
> We need to work out what the requirements are before we can
> settle on an implementation.

Linux-VServer (and probably OpenVZ):

- shared mappings of 'shared' files (binaries
  and libraries) to allow for reduced memory
  footprint when N identical guests are running

- 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)

- 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)

- OOM handling on a per guest basis, i.e. some
  out of memory condition in guest A must not
  affect guest B

HTC,
Herbert

> Sigh. Who is running this show? Anyone?
>
> You can actually do a form of overcommittment by allowing multiple
> containers to share one or more of the zones. Whether that is
> sufficient or suitable I don't know. That depends on the requirements,
> and we haven't even discussed those, let alone agreed to them.
>
> _____
> Containers mailing list
> Containers@lists.osdl.org
> https://lists.osdl.org/mailman/listinfo/containers

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