Subject: Re: [PATCH 0/8] RSS controller based on process containers (v3.1)
Posted by Vaidyanathan Srinivas on Fri, 08 Jun 2007 17:44:36 GMT
View Forum Message <> Reply to Message

Herbert Poetzl wrote:
> On Fri, Jun 08, 2007 at 04:39:28PM +0400, Pavel Emelianov wrote:
>> Herbert Poetzl wrote:
>>> On Mon, Jun 04, 2007 at 05:25:25PM +0400, Pavel Emelianov wrote:
[snip]
>>>> When this usage exceeds the limit set some pages are reclaimed
>>>> from the owning container. In case no reclamation possible the OOM
>>>> killer starts thinning out the container.
>>> so the system (physical machine) starts reclaiming
>>> and probably swapping even when there is no need
>>> to do so?
>> Good catch! The system will start reclaiming right when the
>> container hits the limit to expend its IO bandwidth. Not some
>> other's one that hit the global limit due to some bad container
>> was allowed to go above it.
>
> well, from the system PoV, a constantly swapping
> guest (on an otherwise unused host) is definitely
> something you do not really want, not to talk
> about a tightly packed host system, where guests
> start hogging the I/O with _unnecessary_ swapping
>
>>> e.g. a system with a single guest, limited to 10k
>>> pages, with a working set of 15k pages in different
>>> apps would continuously swap (trash?) on an otherwise
>>> unused (100k+ pages) system?
>>>

Hi Herbert,

When the reclaim process started, swappable pages are unmapped and
moved to swapcache. The RSS accounting treats the page as dropped,
but however the page will remain in memory until there is enough
global pressure to push it out to disk. When a page is faulted-in,
the page is just remapped from the swapcache.

In effect container 'trashing' is not as bad in an otherwise unused
system. Well, certainly we pay a penalty to move around the pages
instead of keeping them mapped in RSS as long as free memory was
available.

The pagecache controller is suppose to track the pagecache and
swapcache pages and push out pages to disk. As Balbir and Pavel have
mentioned, we will be building the features in stages.

--Vaidy

[snip]