Subject: Re: [RFC] mm-controller Posted by Vaidyanathan Srinivas on Mon, 25 Jun 2007 18:22:03 GMT View Forum Message <> Reply to Message

Paul Menage wrote:

- > On 6/25/07, Paul Menage <menage@google.com> wrote:
- >> On 6/22/07, Vaidyanathan Srinivasan <svaidy@linux.vnet.ibm.com> wrote:
- >>> Merging both limits will eliminate the issue, however we would need
- >>> individual limits for pagecache and RSS for better control. There are
- >>> use cases for pagecache limit alone without RSS limit like the case of
- >>> database application using direct IO, backup applications and
- >>> streaming applications that does not make good use of pagecache.

>>>

- >> If streaming applications would otherwise litter the pagecache with
- >> unwanted data, then limiting their total memory footprint (with a
- >> single limit) and forcing them to drop old data sooner sounds like a
- >> great idea.

>

- > Actually, reading what you wrote more carefully, that's sort of what
- > you were already saying. But it's not clear why you wouldn't also want
- > to limit the anon pages for a job, if you're already concerned that
- > it's not playing nicely with the rest of the system.

Hi Paul,

Limiting memory footprint (RSS and pagecache) for multi media applications would work. However, generally streaming applications have a fairly constant RSS size (mapped pagecache pages + ANON) while the unmapped pagecache pages is what we want to control better.

If we have a combined limit for unmapped pagecache pages and RSS, then we will have to bring in vm_swappiness kind of knobs for each container to influence the per container reclaim process so as to not hurt the application performance badly.

RSS controller should be able to take care of the mapped memory footprint if needed. In case of database server, moving out any of it RSS pages will hurt it performance, while we are free to shrink the unmapped pagecache pages to any smaller limit since the database is using direct IO and does not benefit from pagecache.

With pagecache controller, we are able to split application's memory pages into mapped and unmapped pages. Ability to account and control unmapped pages in memory provides more possibilities for fine grain resource management.

--Vaidy