
Subject: [RFC][PATCH][0/4] Memory controller (RSS Control)
Posted by [Balbir Singh](#) on Mon, 19 Feb 2007 06:50:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

This patch applies on top of Paul Menage's container patches (V7) posted at

<http://lkml.org/lkml/2007/2/12/88>

It implements a controller within the containers framework for limiting memory usage (RSS usage).

The memory controller was discussed at length in the RFC posted to lkml
<http://lkml.org/lkml/2006/10/30/51>

Steps to use the controller

0. Download the patches, apply the patches
1. Turn on CONFIG_CONTAINER_MEMCTLR in kernel config, build the kernel and boot into the new kernel
2. mount -t container container -o memctlr /<mount point>
3. cd /<mount point>
 optionally do (mkdir <directory>; cd <directory>) under /<mount point>
4. echo \$\$ > tasks (attaches the current shell to the container)
5. echo -n (limit value) > memctlr_limit
6. cat memctlr_usage
7. Run tasks, check the usage of the controller, reclaim behaviour
8. Report bugs, get bug fixes and iterate (goto step 0).

Advantages of the patchset

1. Zero overhead in struct page (struct page is not expanded)
2. Minimal changes to the core-mm code
3. Shared pages are not reclaimed unless all mappings belong to overlimit containers.
4. It can be used to debug drivers/applications/kernel components in a constrained memory environment (similar to mem=XXX option), except that several containers can be created simultaneously without rebooting and the limits can be changed. NOTE: There is no support for limiting kernel memory allocations and page cache control (presently).

Testing

Ran kernbench and lmbench with containers enabled (container filesystem not mounted), they seemed to run fine
Created containers, attached tasks to containers with lower limits than

the memory the tasks require (memory hog tests) and ran some basic tests on them

TODO's and improvement areas

-
- 1. Come up with cool page replacement algorithms for containers
(if possible without any changes to struct page)
- 2. Add page cache control
- 3. Add kernel memory allocator control
- 4. Extract benchmark numbers and overhead data

Comments & criticism are welcome.

Series

memctlr-setup.patch
memctlr-acct.patch
memctlr-reclaim-on-limit.patch
memctlr-doc.patch

--
Warm Regards,
Balbir Singh
