
Subject: [RFC][PATCH][0/4] Memory controller (RSS Control) (
Posted by [Balbir Singh](#) on Sat, 24 Feb 2007 14:45:03 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>

This is version 2 of the patch, version 1 was posted at
<http://lkml.org/lkml/2007/2/19/10>

I have tried to incorporate all comments, more details can be found in the changelog's of individual patches. Any remaining mistakes are all my fault.

The next question could be why release version 2?

1. It serves a decision point to decide if we should move to a per-container LRU list. Walking through the global LRU is slow, in this patchset I've tried to address the LRU churning issue. The patch `memcontrol-reclaim-on-limit` has more details
2. I've included fixes for several of the comments/issues raised in version 1

Steps to use the controller

-
0. Download the patches, apply the patches
 1. Turn on `CONFIG_CONTAINER_MEMCONTROL` in kernel config, build the kernel and boot into the new kernel
 2. `mount -t container container -o memcontrol /<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) > memcontrol_limit`
 6. `cat memcontrol_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

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.

Tested the patches on UML and PowerPC. On UML tried the patches with the config enabled and disabled (sanity check) and with containers enabled but the memory controller disabled.

TODO's and improvement areas

1. Come up with cool page replacement algorithms for containers - still holds good (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

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

--

Warm Regards,
Balbir Singh
