
Subject: [-mm PATCH 0/9] Memory controller introduction (v4)

Posted by [Balbir Singh](#) on Fri, 27 Jul 2007 20:09:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

Here's version 4 of the memory controller.

Changelog since version 3

1. Ported to v11 of the containers patchset (2.6.23-rc1-mm1). Paul Menage helped immensely with a detailed review of v3
2. Reclaim is retried to allow reclaim of pages coming in as a result of mapped pages reclaim (swap cache growing as a result of RSS reclaim)
3. `page_referenced()` is now container aware. During container reclaim, references from other containers do not prevent a page from being reclaimed from a non-referencing container
4. Fixed a possible race condition spotted by YAMAMOTO Takashi

Changelog since version 2

1. Improved error handling in `mm/memory.c` (spotted by YAMAMOTO Takashi)
2. Test results included
3. `try_to_free_mem_container_pages()` bug fix (`sc->may_writepage` is now set to `!laptop_mode`)

Changelog since version 1

1. Fixed some compile time errors (in `mm/migrate.c` from Vaidyanathan S)
2. Fixed a panic seen when `LIST_DEBUG` is enabled
3. Added a mechanism to control whether we track page cache or both page cache and mapped pages (as requested by Pavel)
4. Dave Hansen provided detail review comments on the code.

This patchset implements another version of the memory controller. These patches have been through a big churn, the first set of patches were posted last year and earlier this year at <http://lkml.org/lkml/2007/2/19/10>

This patchset draws from the patches listed above and from some of the contents of the patches posted by Vaidyanathan for page cache control. <http://lkml.org/lkml/2007/6/20/92>

At OLS, the resource management BOF, it was discussed that we need to manage RSS and unmapped page cache together. This patchset is a step towards that

TODO's

1. Add memory controller water mark support. Reclaim on high water mark
2. Add support for shrinking on limit change

3. Add per zone per container LRU lists (this is being actively worked on by Pavel Emelianov)
4. Figure out a better CLUI for the controller
5. Add better statistics
6. Explore using `read_unit64()` as recommended by Paul Menage (NOTE: `read_ulong()` would also be nice to have)

In case you have been using/testing the RSS controller, you'll find that this controller works slower than the RSS controller. The reason being that both swap cache and page cache is accounted for, so pages do go out to swap upon reclaim (they cannot live in the swap cache).

Any test output, feedback, comments, suggestions are welcome! I am committed to fixing any bugs and improving the performance of the memory controller. Do not hesitate to send any fixes, request for fixes that is required.

Using the patches

1. Enable Memory controller configuration
2. Compile and boot the new kernel
3. `mount -t container container -o mem_container /container`
will mount the memory controller to the `/container` mount point
4. `mkdir /container/a`
5. `echo $$ > /container/a/tasks` (add tasks to the new container)
6. `echo -n <num_pages> > /container/a/mem_limit`
example
`echo -n 204800 > /container/a/mem_limit`, sets the limit to 800 MB on a system with 4K page size
7. run tasks, see the memory controller work
8. Report results, provide feedback
9. Develop/use new patches and go to step 1

Test Results

Results for version 3 of the patch were posted at <http://lwn.net/Articles/242554/>

The code was also tested on a power box with regular machine usage scenarios, the config disabled and with a stress suite that touched all the memory in the system and was limited in a container.

series

```
res_counters_infra.patch
mem-control-setup.patch
mem-control-accounting-setup.patch
mem-control-accounting.patch
mem-control-task-migration.patch
```

mem-control-lru-and-reclaim.patch
mem-control-out-of-memory.patch
mem-control-choose-rss-vs-rss-and-pagecache.patch
mem-control-per-container-page-referenced.patch

--

Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
