

---

Subject: [RFC][PATCH 0/3] Containers: Pagecache accounting and control subsystem (v3)

Posted by [Vaidyanathan Srinivas](#) on Wed, 23 May 2007 14:48:57 GMT

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

---

Containers: Pagecache accounting and control subsystem (v3)

-----

This patch extends the RSS controller to account and reclaim pagecache and swapcache pages. This is a prototype to demonstrate that the existing container infrastructure is useful to build different VM controller.

This patch is based on RSS Controller V2 by Pavel and Balbir. This patch depends on

1. Paul Menage's Containers (V8): Generic Process Containers

<http://lkml.org/lkml/2007/4/6/297>

2. Pavel Emelianov's RSS controller based on process containers (v2)

<http://lkml.org/lkml/2007/4/9/78>

3. Balbir's fixes for RSS controller as mentioned in

<http://lkml.org/lkml/2007/5/17/232>

This is very much work-in-progress, you can certainly expect hangs/crash if both pagecache and rss limits are set and the container is stressed.

Comments, suggestions and criticisms are welcome.

Thanks,  
Vaidy

Features:

-----

- \* No new subsystem is added. The RSS controller subsystem is extended since most of the code can be shared between pagecache control and RSS control.
- \* The accounting number include pages in swap cache and filesystem buffer pages apart from pagecache, basically everything under NR\_FILE\_PAGES is counted as pagecache.
- \* Limits on pagecache can be set by `echo -n 100000 > pagecache_limit` on the /container file system. The unit is in pages or 4 kilobytes
- \* If the pagecache utilisation limit is exceeded, the container reclaim code is invoked to recover pages from the container.

Advantages:

-----

- \* Minimal code changes to RSS controller to include pagecache pages

Limitations:

-----

- \* All limitation of RSS controller v2 applies to this code as well
- \* Page reclaim needs to be reworked to select correct pages when the respective limits are exceeded
- \* Concurrent and recursive triggering of reclaimer code is a mess leading to deadlocks. Reclaimer needs to be serialised and reworked to do the right job and also improve performance

#### Usage:

-----

- \* Add all dependent patches before including this patch
- \* No new config settings apart from enabling CONFIG\_RSS\_CONTAINER
- \* Boot new kernel
- \* Mount container filesystem  
mount -t container none /container  
cd /container
- \* Create new container  
mkdir mybox  
cd /container/mybox
- \* Add current shell to container  
echo \$\$ > tasks
- \* There are two files pagecache\_usage and pagecache\_limit
- \* In order to set limit, echo value in pages (4KB) to pagecache\_limit  
echo -n 100000 > pagecache\_limit  
#This would set 409MB limit on pagecache usage
- \* Trash the system from current shell using scp/cp/dd/tar etc
- \* Watch pagecache\_usage and /proc/meminfo to verify behavior

#### Tests:

-----

- \* Simple dd/cat/cp test on pagecache limit
- \* rss\_limit was tested with simple test application that would malloc predefined size of memory and touch them to allocate pages.

#### ToDo:

----

- \* Optimise the reclaim. Currently isolate\_container\_pages does not distinguish between whether pagecache limit is hit or rss limit is hit
- \* Prevent concurrent reclaim and recursive reclaim when both limits are set.

#### Patch Series:

-----

pagecache-controller-v3-setup.patch  
pagecache-controller-v3-acct.patch  
pagecache-controller-v3-acct-hooks.patch