Subject: Re: [ckrm-tech] [patch00/05]: Containers(V2)- Introduction
Posted by Chandra Seetharaman on Wed, 27 Sep 2006 19:50:44 GMT
View Forum Message <> Reply to Message

Rohit,

I finally looked into your memory controller patches. Here are some of the issues I see:

(All points below are in the context of page limit of containers being hit and the new code starts freeing up pages)

1. LRU is ignored totally thereby thrashing the working set (as pointed by Peter Zijlstra).
2. Frees up file pages first when hitting the page limit thereby making vm_swappiness ineffective.
3. Starts writing back pages when the # of file pages is close to the limit, thereby breaking the current writeback algorithm/logic.
4. MAPPED files are not counted against the page limit. why ?. This affects reclamation behavior and makes vm_swappiness ineffective.
5. Starts freeing up pages from the first task or the first file in the linked list. This logic unfairly penalizes the early members of the list.
6. Both active and inactive pages use physical pages. But, the controller only counts active pages and not inactive pages. why ?
7. Page limit is checked against the sum of (anon and file pages) in some places and against active pages at some other places. IMO, it should be always compared to the same value.

BTW, It will be easier to read/follow the patches if you separate them out as functionalities.

regards,

chandra

On Tue, 2006-09-19 at 19:16 -0700, Rohit Seth wrote:
> Containers:
>
> Commodity HW is becoming more powerful. This is giving opportunity to
> run different workloads on the same platform for better HW resource
> utilization. To run different workloads efficiently on the same
> platform, it is critical that we have a notion of limits for each
> workload in Linux kernel. Current cpuset feature in Linux kernel
> provides grouping of CPU and memory support to some extent (for NUMA
> machines).
>
> For example, a user can run a batch job like backup inside containers.

> This job if run unconstrained could step over most of the memory present
> in system thus impacting other workloads running on the system at that
> time. But when the same job is run inside containers then the backup
> job is run within container limits.
>
> We use the term container to indicate a structure against which we track
> and charge utilization of system resources like memory, tasks etc for a
> workload. Containers will allow system admins to customize the
> underlying platform for different applications based on their
> performance and HW resource utilization needs. Containers contain
> enough infrastructure to allow optimal resource utilization without
> bogging down rest of the kernel. A system admin should be able to
> create, manage and free containers easily.
>
> At the same time, changes in kernel are minimized so as this support can
> be easily integrated with mainline kernel.
>
> The user interface for containers is through configfs. Appropriate file
> system privileges are required to do operations on each container.
> Currently implemented container resources are automatically visible to
> user space through /configfs/container/<container_name> after a
> container is created.
>
> Signed-off-by: Rohit Seth <rohitseth@google.com>
>
> Diffstat for the patch set (against linux-2.6.18-rc6-mm2_:
>

```
>  Documentation/containers.txt |   65 ++++
>  fs/inode.c                   |    3
>  include/linux/container.h    |  167 ++++++++++
>  include/linux/fs.h           |    5
>  include/linux/mm_inline.h    |    4
>  include/linux/mm_types.h     |    4
>  include/linux/sched.h        |    6
>  init/Kconfig                 |    8
>  kernel/Makefile              |    1
>  kernel/container_configfs.c  |  440 ++++++++++++++++++++++++++++
>  kernel/exit.c                |    2
>  kernel/fork.c                |    9
>  mm/Makefile                  |    2
>  mm/container.c               |  658 +++++++++++++++++++++++++++++++++++++++++++
>  mm/container_mm.c            |  512 +++++++++++++++++++++++++++++++++
>  mm/filemap.c                 |    4
>  mm/page_alloc.c              |    3
>  mm/rmap.c                    |    8
>  mm/swap.c                    |    1
>  mm/vmscan.c                  |    1
>  20 files changed, 1902 insertions(+), 1 deletion(-)
```

>
> Changes from version 1:
> Fixed the Documentation error
> Fixed the corruption in container task list
> Added the support for showing all the tasks belonging to a container
> through showtask attribute
> moved the Kconfig changes to init directory (from mm)
> Fixed the bug of unregistering container subsystem if we are not able to
> create workqueue
> Better support for handling limits for file pages. This now includes
> support for flushing and invalidating page cache pages.
> Minor other changes.
>
> ** *****
> This patch set has basic container support that includes:
>
> - Create a container using mkdir command in configfs
>
> - Free a container using rmdir command
>
> - Dynamically adjust memory and task limits for container.
>
> - Add/Remove a task to container (given a pid)
>
> - Files are currently added as part of open from a task that already
> belongs to a container.
>
> - Keep track of active, anonymous, mapped and pagecache usage of
> container memory
>
> - Does not allow more than task_limit number of tasks to be created in
> the container.
>
> - Over the limit memory handler is called when number of pages (anon +
> pagecache) exceed the limit. It is also called when number of active
> pages exceed the page limit. Currently, this memory handler scans the
> mappings and tasks belonging to container (file and anonymous) and tries
> to deactivate pages. If the number of page cache pages is also high
> then it also invalidate mappings. The thought behind this scheme is, it
> is okay for containers to go over limit as long they run in degraded
> manner when they are over their limit. Also, if there is any memory
> pressure then pages belonging to over the limit container(s) become
> prime candidates for kernel reclaimer. Container mutex is also held
> during the time this handler is working its way through to prevent any
> further addition of resources (like tasks or mappings) to this
> container. Though it also blocks removal of same resources from the
> container for the same time. It is possible that over the limit page
> handler takes lot of time if memory pressure on a container is

> continuously very high. The limits, like how long a task should
> schedule out when it hits memory limit, is also on the lower side at
> present (particularly when it is memory hogger). But should be easy to
> change if need be.
>
> - Indicate the number of times the page limit and task limit is hit
>
> - Indicate the tasks (pids) belonging to container.
>
> Below is a one line description for patches that will follow:
>
> [patch01]: Documentation on how to use containers
> (Documentation/container.txt)
>
> [patch02]: Changes in the generic part of kernel code
>
> [patch03]: Container's interface with configfs
>
> [patch04]: Core container support
>
> [patch05]: Over the limit memory handler.
>
> TODO:
>
> - some code(like container_add_task) in mm/container.c should go
> elsewhere.
> - Support adding/removing a file name to container through configfs
> - /proc/pid/container to show the container id (or name)
> - More testing for memory controller. Currently it is possible that
> limits are exceeded. See if a call to reclaim can be easily integrated.
> - Kernel memory tracking (based on patches from BC)
> - Limit on user locked memory
> - Huge memory support
> - Stress testing with containers
> - One shot view of all containers
> - CKRM folks are interested in seeing all processes belonging to a
> container. Add the attribute show_tasks to container.
> - Add logic so that the sum of limits are not exceeding appropriate
> system requirements.
> - Extend it with other controllers (CPU and Disk I/O)
> - Add flags bits for supporting different actions (like in some cases
> provide a hard memory limit and in some cases it could be soft).
> - Capability to kill processes for the extreme cases.
> ...
>
> This is based on lot of discussions over last month or so. I hope this
> patch set is something that we can agree and more support can be added
> on top of this. Please provide feedback and add other extensions that

> are useful in the TODO list.
>
> Thanks,
> -rohit
>
>
>
>
>

--

```
 ------------------------------------------------------------ ----------
   Chandra Seetharaman               | Be careful what you choose....
             - sekharan@us.ibm.com   |      .......you may get it.
 ------------------------------------------------------------ ----------
```