Subject: Re: [RFC][PATCH 5/7] Per-container OOM killer and page reclamation
Posted by xemul on Sun, 11 Mar 2007 08:39:18 GMT
View Forum Message <> Reply to Message

Balbir Singh wrote:
> Hi, Pavel,
>
> Please find my patch to add LRU behaviour to your latest RSS controller.

Thanks for participation and additional testing :)
I'll include this into next generation of patches.

> Balbir Singh
> Linux Technology Center
> IBM, ISTL
>
>
> -- ------------
>
> Add LRU behaviour to the RSS controller patches posted by Pavel Emelianov
>
> http://lkml.org/lkml/2007/3/6/198
>
> which was in turn similar to the RSS controller posted by me
>
> http://lkml.org/lkml/2007/2/26/8
>
> Pavel's patches have a per container list of pages, which helps reduce
> reclaim time of the RSS controller but the per container list of pages is
> in FIFO order. I've implemented active and inactive lists per container to
> help select the right set of pages to reclaim when the container is under
> memory pressure.
>
> I've tested these patches on a ppc64 machine and they work fine for
> the minimal testing I've done.
>
> Pavel would you please include these patches in your next iteration.
>
> Comments, suggestions and further improvements are as always welcome!
>
> Signed-off-by: <balbir@in.ibm.com>
> ---
>
> include/linux/rss_container.h | 1
> mm/rss_container.c | 47 +++++++++++++++++++++++++++++++++-----------
> mm/swap.c | 5 ++++
> mm/vmscan.c | 3 ++
> 4 files changed, 44 insertions(+), 12 deletions(-)

>
> diff -puN include/linux/rss_container.h~rss-container-lru2 include/linux/rss_container.h
> --- linux-2.6.20/include/linux/rss_container.h~rss-container-lru 2 2007-03-09
22:52:56.000000000 +0530
> +++ linux-2.6.20-balbir/include/linux/rss_container.h 2007-03-10 00:39:59.000000000 +0530
> @@ -19,6 +19,7 @@ int container_rss_prepare(struct page *,
> void container_rss_add(struct page_container *);
> void container_rss_del(struct page_container *);
> void container_rss_release(struct page_container *);
> +void container_rss_move_lists(struct page *pg, bool active);
>
> int mm_init_container(struct mm_struct *mm, struct task_struct *tsk);
> void mm_free_container(struct mm_struct *mm);
> diff -puN mm/rss_container.c~rss-container-lru2 mm/rss_container.c
> --- linux-2.6.20/mm/rss_container.c~rss-container-lru2 2007-03-09 22:52:56.000000000 +0530
> +++ linux-2.6.20-balbir/mm/rss_container.c 2007-03-10 02:42:54.000000000 +0530
> @@ -17,7 +17,8 @@ static struct container_subsys rss_subsy
>
> struct rss_container {
>   struct res_counter res;
> - struct list_head page_list;
> + struct list_head inactive_list;
> + struct list_head active_list;
>   struct container_subsys_state css;
> };
>
> @@ -96,6 +97,26 @@ void container_rss_release(struct page_c
>   kfree(pc);
> }
>
> +void container_rss_move_lists(struct page *pg, bool active)
> +{
> + struct rss_container *rss;
> + struct page_container *pc;
> +
> + if (!page_mapped(pg))
> + return;
> +
> + pc = page_container(pg);
> + BUG_ON(!pc);
> + rss = pc->cnt;
> +
> + spin_lock_irq(&rss->res.lock);
> + if (active)
> + list_move(&pc->list, &rss->active_list);
> + else
> + list_move(&pc->list, &rss->inactive_list);
> + spin_unlock_irq(&rss->res.lock);

```
> +}
> +
>  void container_rss_add(struct page_container *pc)
>  {
>   struct page *pg;
> @@ -105,7 +126,7 @@ void container_rss_add(struct page_conta
>   rss = pc->cnt;
>
>   spin_lock(&rss->res.lock);
> - list_add(&pc->list, &rss->page_list);
> + list_add(&pc->list, &rss->active_list);
>   spin_unlock(&rss->res.lock);
>
>   page_container(pg) = pc;
> @@ -141,7 +162,10 @@ unsigned long container_isolate_pages(un
>   struct zone *z;
>
>   spin_lock_irq(&rss->res.lock);
> - src = &rss->page_list;
> + if (active)
> +  src = &rss->active_list;
> + else
> +  src = &rss->inactive_list;
>
>   for (scan = 0; scan < nr_to_scan && !list_empty(src); scan++) {
>    pc = list_entry(src->prev, struct page_container, list);
> @@ -152,13 +176,10 @@ unsigned long container_isolate_pages(un
>
>    spin_lock(&z->lru_lock);
>    if (PageLRU(page)) {
> -   if ((active && PageActive(page)) ||
> -    (!active && !PageActive(page))) {
> -    if (likely(get_page_unless_zero(page))) {
> -     ClearPageLRU(page);
> -     nr_taken++;
> -     list_move(&page->lru, dst);
> -    }
> +   if (likely(get_page_unless_zero(page))) {
> +    ClearPageLRU(page);
> +    nr_taken++;
> +    list_move(&page->lru, dst);
>     }
>    }
>    spin_unlock(&z->lru_lock);
> @@ -212,7 +233,8 @@ static int rss_create(struct container_s
>    return -ENOMEM;
>
>   res_counter_init(&rss->res);
```

> - INIT_LIST_HEAD(&rss->page_list);
> + INIT_LIST_HEAD(&rss->inactive_list);
> + INIT_LIST_HEAD(&rss->active_list);
> cont->subsys[rss_subsys.subsys_id] = &rss->css;
> return 0;
> }
> @@ -284,7 +306,8 @@ static __init int rss_create_early(struc
>
> rss = &init_rss_container;
> res_counter_init(&rss->res);
> - INIT_LIST_HEAD(&rss->page_list);
> + INIT_LIST_HEAD(&rss->inactive_list);
> + INIT_LIST_HEAD(&rss->active_list);
> cont->subsys[rss_subsys.subsys_id] = &rss->css;
> ss->create = rss_create;
> return 0;
> diff -puN mm/vmscan.c~rss-container-lru2 mm/vmscan.c
> --- linux-2.6.20/mm/vmscan.c~rss-container-lru2 2007-03-09 22:52:56.000000000 +0530
> +++ linux-2.6.20-balbir/mm/vmscan.c 2007-03-10 00:42:35.000000000 +0530
> @@ -1142,6 +1142,7 @@ static unsigned long container_shrink_pa
> else
> add_page_to_inactive_list(z, page);
> spin_unlock_irq(&z->lru_lock);
> + container_rss_move_lists(page, false);
>
> put_page(page);
> }
> @@ -1191,6 +1192,7 @@ static void container_shrink_pages_activ
> list_move(&page->lru, &z->inactive_list);
> z->nr_inactive++;
> spin_unlock_irq(&z->lru_lock);
> + container_rss_move_lists(page, false);
>
> put_page(page);
> }
> @@ -1206,6 +1208,7 @@ static void container_shrink_pages_activ
> list_move(&page->lru, &z->active_list);
> z->nr_active++;
> spin_unlock_irq(&z->lru_lock);
> + container_rss_move_lists(page, true);
>
> put_page(page);
> }
> diff -puN mm/swap.c~rss-container-lru2 mm/swap.c
> --- linux-2.6.20/mm/swap.c~rss-container-lru2 2007-03-10 00:42:38.000000000 +0530
> +++ linux-2.6.20-balbir/mm/swap.c 2007-03-10 01:20:39.000000000 +0530
> @@ -30,6 +30,7 @@
> #include <linux/cpu.h>

```
>  #include <linux/notifier.h>
>  #include <linux/init.h>
> +#include <linux/rss_container.h>
>
>  /* How many pages do we try to swap or page in/out together? */
>  int page_cluster;
> @@ -140,6 +141,7 @@ int rotate_reclaimable_page(struct page
>  void fastcall activate_page(struct page *page)
>  {
>   struct zone *zone = page_zone(page);
> + bool moved = false;
>
>   spin_lock_irq(&zone->lru_lock);
>   if (PageLRU(page) && !PageActive(page)) {
> @@ -147,8 +149,11 @@ void fastcall activate_page(struct page
>    SetPageActive(page);
>    add_page_to_active_list(zone, page);
>    __count_vm_event(PGACTIVATE);
> +  moved = true;
>   }
>   spin_unlock_irq(&zone->lru_lock);
> + if (moved)
> +  container_rss_move_lists(page, true);
>  }
>
>  /*
> _
```