

---

Subject: [PATCH] memory.swappiness

Posted by [yamamoto](#) on Mon, 03 Dec 2007 02:54:16 GMT

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

---

here's a trivial patch to implement memory.swappiness,  
which controls swappiness for cgroup memory reclamation.

it's against 2.6.24-rc3-mm2.

YAMAMOTO Takashi

Signed-off-by: YAMAMOTO Takashi <yamamoto@valinux.co.jp>

---

--- linux-2.6.24-rc3-mm2-swappiness/include/linux/memcontrol.h.BACKUP 2007-12-03  
11:49:27.176669111 +0900

+++ linux-2.6.24-rc3-mm2-swappiness/include/linux/memcontrol.h 2007-12-03  
10:00:29.049448425 +0900

```
@@ -46,6 +46,7 @@ extern void mem_cgroup_out_of_memory(str
extern int mem_cgroup_cache_charge(struct page *page, struct mm_struct *mm,
    gfp_t gfp_mask);
int task_in_mem_cgroup(struct task_struct *task, const struct mem_cgroup *mem);
+extern int mem_cgroup_swappiness(struct mem_cgroup *mem);
```

```
static inline struct mem_cgroup *mm_cgroup(const struct mm_struct *mm)
```

```
{
```

--- linux-2.6.24-rc3-mm2-swappiness/mm/vmscan.c.BACKUP 2007-12-03 07:49:00.000000000  
+0900

+++ linux-2.6.24-rc3-mm2-swappiness/mm/vmscan.c 2007-12-03 10:01:57.559803379 +0900

```
@@ -1030,7 +1030,7 @@ static int calc_reclaim_mapped(struct sc
```

```
*
```

```
* Max temporary value is vm_total_pages*100.
```

```
*/
```

```
- imbalance *= (vm_swappiness + 1);
```

```
+ imbalance *= (sc->swappiness + 1);
```

```
imbalance /= 100;
```

```
/*
```

```
@@ -1445,7 +1445,7 @@ unsigned long try_to_free_mem_cgroup_pag
```

```
.may_writepage = !laptop_mode,
```

```
.may_swap = 1,
```

```
.swap_cluster_max = SWAP_CLUSTER_MAX,
```

```
- .swappiness = vm_swappiness,
```

```
+ .swappiness = mem_cgroup_swappiness(mem_cont),
```

```
.order = 0,
```

```
.mem_cgroup = mem_cont,
```

```
.isolate_pages = mem_cgroup_isolate_pages,
```

```

--- linux-2.6.24-rc3-mm2-swappiness/mm/memcontrol.c.BACKUP 2007-12-03
07:49:00.000000000 +0900
+++ linux-2.6.24-rc3-mm2-swappiness/mm/memcontrol.c 2007-12-03 11:22:40.157163781 +0900
@@ -133,6 +133,7 @@ struct mem_cgroup {

    unsigned long control_type; /* control RSS or RSS+Pagecache */
    int prev_priority; /* for recording reclaim priority */
+ unsigned int swappiness; /* swappiness */
    /*
     * statistics.
     */
@@ -1077,7 +1078,23 @@ static int mem_control_stat_open(struct
    return single_open(file, mem_control_stat_show, cont);
}

+static int mem_cgroup_swappiness_write(struct cgroup *cont, struct cftype *cft,
+    u64 val)
+{
+ struct mem_cgroup *mem = mem_cgroup_from_cont(cont);
+
+ if (val > 100)
+ return -EINVAL;
+ mem->swappiness = val;
+ return 0;
+}
+
+static u64 mem_cgroup_swappiness_read(struct cgroup *cont, struct cftype *cft)
+{
+ struct mem_cgroup *mem = mem_cgroup_from_cont(cont);
+
+ return mem->swappiness;
+}

static struct cftype mem_cgroup_files[] = {
{
@@ -1110,8 +1127,21 @@ static struct cftype mem_cgroup_files[]
    .name = "stat",
    .open = mem_control_stat_open,
},
+ {
+ .name = "swappiness",
+ .write_uint = mem_cgroup_swappiness_write,
+ .read_uint = mem_cgroup_swappiness_read,
+ },
};

+/* XXX probably it's better to move try_to_free_mem_cgroup_pages to
+ memcontrol.c and kill this */

```

```

+int mem_cgroup_swappiness(struct mem_cgroup *mem)
+{
+
+ return mem->swappiness;
+}
+
static int alloc_mem_cgroup_per_zone_info(struct mem_cgroup *mem, int node)
{
    struct mem_cgroup_per_node *pn;
@@ -1155,6 +1185,8 @@ mem_cgroup_create(struct cgroup_subsys *
    res_counter_init(&mem->res);

    mem->control_type = MEM_CGROUP_TYPE_ALL;
+ mem->swappiness = 60; /* XXX probably should inherit a value from
+ either parent cgroup or global vm_swappiness */
    memset(&mem->info, 0, sizeof(mem->info));

    for_each_node_state(node, N_POSSIBLE)

```

---

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

---



---

Subject: Re: [PATCH] memory.swappiness  
Posted by [Balbir Singh](#) on Mon, 03 Dec 2007 03:25:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

YAMAMOTO Takashi wrote:

```

> here's a trivial patch to implement memory.swappiness,
> which controls swappiness for cgroup memory reclamation.
>
> it's against 2.6.24-rc3-mm2.
>
> YAMAMOTO Takashi
>
>
> Signed-off-by: YAMAMOTO Takashi <yamamoto@valinux.co.jp>
> ---
>
> --- linux-2.6.24-rc3-mm2-swappiness/include/linux/memcontrol.h.BACKUP 2007-12-03
11:49:27.176669111 +0900
> +++ linux-2.6.24-rc3-mm2-swappiness/include/linux/memcontrol.h 2007-12-03
10:00:29.049448425 +0900
> @@ -46,6 +46,7 @@ extern void mem_cgroup_out_of_memory(str
> extern int mem_cgroup_cache_charge(struct page *page, struct mm_struct *mm,
>     gfp_t gfp_mask);
> int task_in_mem_cgroup(struct task_struct *task, const struct mem_cgroup *mem);

```

```

> +extern int mem_cgroup_swappiness(struct mem_cgroup *mem);
>
> static inline struct mem_cgroup *mm_cgroup(const struct mm_struct *mm)
> {
> --- linux-2.6.24-rc3-mm2-swappiness/mm/vmscan.c.BACKUP 2007-12-03 07:49:00.000000000
+0900
> +++ linux-2.6.24-rc3-mm2-swappiness/mm/vmscan.c 2007-12-03 10:01:57.559803379 +0900
> @@ -1030,7 +1030,7 @@ static int calc_reclaim_mapped(struct sc
> *
> * Max temporary value is vm_total_pages*100.
> */
> - imbalance *= (vm_swappiness + 1);
> + imbalance *= (sc->swappiness + 1);
> imbalance /= 100;
>
> /*
> @@ -1445,7 +1445,7 @@ unsigned long try_to_free_mem_cgroup_pag
> .may_writepage = !laptop_mode,
> .may_swap = 1,
> .swap_cluster_max = SWAP_CLUSTER_MAX,
> - .swappiness = vm_swappiness,
> + .swappiness = mem_cgroup_swappiness(mem_cont),
> .order = 0,
> .mem_cgroup = mem_cont,
> .isolate_pages = mem_cgroup_isolate_pages,
> --- linux-2.6.24-rc3-mm2-swappiness/mm/memcontrol.c.BACKUP 2007-12-03
07:49:00.000000000 +0900
> +++ linux-2.6.24-rc3-mm2-swappiness/mm/memcontrol.c 2007-12-03 11:22:40.157163781
+0900
> @@ -133,6 +133,7 @@ struct mem_cgroup {
>
> unsigned long control_type; /* control RSS or RSS+Pagecache */
> int prev_priority; /* for recording reclaim priority */
> + unsigned int swappiness; /* swappiness */
> /*
> * statistics.
> */
> @@ -1077,7 +1078,23 @@ static int mem_control_stat_open(struct
> return single_open(file, mem_control_stat_show, cont);
> }
>
> +static int mem_cgroup_swappiness_write(struct cgroup *cont, struct cftype *cft,
> + u64 val)
> +{
> + struct mem_cgroup *mem = mem_cgroup_from_cont(cont);
> +
> + if (val > 100)
> + return -EINVAL;

```

```

> + mem->swappiness = val;
> + return 0;
> +}
> +
> +static u64 mem_cgroup_swappiness_read(struct cgroup *cont, struct cftype *cft)
> +{
> + struct mem_cgroup *mem = mem_cgroup_from_cont(cont);
> +
> + return mem->swappiness;
> +}
> +
> static struct cftype mem_cgroup_files[] = {
> {
> @@ -1110,8 +1127,21 @@ static struct cftype mem_cgroup_files[]
> .name = "stat",
> .open = mem_control_stat_open,
> },
> +{
> + .name = "swappiness",
> + .write_uint = mem_cgroup_swappiness_write,
> + .read_uint = mem_cgroup_swappiness_read,
> + },
> };
> +
> +/* XXX probably it's better to move try_to_free_mem_cgroup_pages to
> + memcontrol.c and kill this */
> +int mem_cgroup_swappiness(struct mem_cgroup *mem)
> +{
> +
> + return mem->swappiness;
> +}
> +
> static int alloc_mem_cgroup_per_zone_info(struct mem_cgroup *mem, int node)
> {
> struct mem_cgroup_per_node *pn;
> @@ -1155,6 +1185,8 @@ mem_cgroup_create(struct cgroup_subsys *
> res_counter_init(&mem->res);
> +
> mem->control_type = MEM_CGROUP_TYPE_ALL;
> + mem->swappiness = 60; /* XXX probably should inherit a value from
> + either parent cgroup or global vm_swappiness */

```

I prefer inheriting from the parent, but since our hierarchy support needs work (changes to res\_counter to support actual hierarchies), I would prefer using global vm\_swappiness.

```

> memset(&mem->info, 0, sizeof(mem->info));
>

```

> for\_each\_node\_state(node, N\_POSSIBLE)

I like this patch very much

Acked-by: Balbir Singh <balbir@linux.vnet.ibm.com>

--

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>

---