Subject: [RFC][PATCH 4/4] memcg shrinking usage
Posted by KAMEZAWA Hiroyuki on Fri, 18 Jul 2008 10:37:54 GMT
View Forum Message <> Reply to Message

Andrew Morton suggested me "If you want to add background reclaim
per memcg, do that in user-land. Don't add kernel thread at el more."

In general, when we try to implement some automatic rich control to memcg,
there are two choices.
1. do in the kernel.
2. do in the user.

In these days, memory subsystem is getting larger and lareger and
"Do something compliated" in the kernel is not good manner.
And Linux has some amount of helper programs in userland for years.

This patch adds a core interface to implement a user daemon program
which controles memcg.

- memory.shrink_in_bytes.

When a user want to shrink memory.usage below 400M, he can write

echo 400M > memory.shrink_in_bytes

TODO:
- add a notifier interface to trigger daemon's action.
- add asynchronous mode....is difficult..maybe the daemon has to use
  thread or fork to do control memcg in asynchronous manner.

Signed-off-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>

Documentation/controllers/memory.txt | 20 ++++++++++++++++++---
mm/memcontrol.c | 20 ++++++++++++++++++++
2 files changed, 37 insertions(+), 3 deletions(-)

Index: mmtom-stamp-2008-07-15-15-39/mm/memcontrol.c
===
--- mmtom-stamp-2008-07-15-15-39.orig/mm/memcontrol.c
+++ mmtom-stamp-2008-07-15-15-39/mm/memcontrol.c
@@ -40,6 +40,16 @@ struct cgroup_subsys mem_cgroup_subsys _
static struct kmem_cache *page_cgroup_cache __read_mostly;
#define MEM_CGROUP_RECLAIM_RETRIES 5

+
+#define MEMCG_FILETAG_BASE 0xab00
+
+enum {

```
+ PRIVATE_FILETAG_SHRINK,
+ PRIVATE_FILETAG_MAX,
+};
+
+#define MEMCG_FILETAG_SHRINK (MEMCG_FILETAG_BASE +
PRIVATE_FILETAG_SHRINK)
+
 /*
  * Statistics for memory cgroup.
  */
@@ -950,6 +960,11 @@ static int mem_cgroup_write(struct cgrou
   if (!ret)
    ret = mem_cgroup_resize_limit(memcg, val);
   break;
+ case MEMCG_FILETAG_SHRINK:
+  ret = res_counter_memparse_write_strategy(buffer, &val):
+  if (!ret)
+   ret= mem_cgroup_shrink_usage_to(memcg, val);
+  break;
  default:
   ret = -EINVAL; /* should be BUG() ? */
   break;
@@ -1061,6 +1076,11 @@ static struct cftype mem_cgroup_files[]
   .name = "stat",
   .read_map = mem_control_stat_show,
  },
+ {
+  .name = "shrink_in_bytes",
+  .private = MEMCG_FILETAG_SHRINK,
+  .write_string = mem_cgroup_write,
+ },
 };

 static int alloc_mem_cgroup_per_zone_info(struct mem_cgroup *mem, int node)
Index: mmtom-stamp-2008-07-15-15-39/Documentation/controllers/memory.txt
===================================================================
--- mmtom-stamp-2008-07-15-15-39.orig/Documentation/controllers/memory.txt
+++ mmtom-stamp-2008-07-15-15-39/Documentation/controllers/memory.txt
@@ -152,17 +152,17 @@ The memory controller uses the following

 3. User Interface

-0. Configuration
+3.0 Configuration

 a. Enable CONFIG_CGROUPS
 b. Enable CONFIG_RESOURCE_COUNTERS
 c. Enable CONFIG_CGROUP_MEM_RES_CTLR
```

-1. Prepare the cgroups
+3.1 Prepare the cgroups
 # mkdir -p /cgroups
 # mount -t cgroup none /cgroups -o memory

-2. Make the new group and move bash into it
+3.2 Make the new group and move bash into it
 # mkdir /cgroups/0
 # echo $$ > /cgroups/0/tasks

@@ -196,6 +196,7 @@ this file after a write to guarantee the
 The memory.failcnt field gives the number of times that the cgroup limit was
 exceeded.

+3.4 force_empty
 The memory.stat file gives accounting information. Now, the number of
 caches, RSS and Active pages/Inactive pages are shown.

@@ -205,6 +206,19 @@ The memory.force_empty gives an interfac

 will drop all charges in cgroup. Currently, this is maintained for test.

+3.5 shrink_in_bytes
+A user can try to reclaim memory under a group. If you want to shrink the
+memory usage to below 400M,
+
+# echo 400M > memory.shrink_in_bytes
+
+The kernel tries to shrink memory usage to be under 400M.
+
+This is a kind of hard-to-use operation by hand but this is for group management
+softwares. They enables background page reclaim, and well-scheduled load
+balancing between groups.
+If it is hard to shrink usage, the kernel returns -EBUSY.
+
 4. Testing

 Balbir posted lmbench, AIM9, LTP and vmmstress results [10] and [11].