Subject: [RFC][-mm PATCH 2/8] Memory controller containers setup (v3)
Posted by Balbir Singh on Fri, 20 Jul 2007 08:24:16 GMT
View Forum Message <> Reply to Message

Setup the memory container and add basic hooks and controls to integrate
and work with the container.

Signed-off-by: <balbir@linux.vnet.ibm.com>
---

include/linux/container_subsys.h | 6 +
include/linux/memcontrol.h | 21 +++++
init/Kconfig | 8 ++
mm/Makefile | 1
mm/memcontrol.c | 143 ++
5 files changed, 179 insertions(+)

diff -puN include/linux/container_subsys.h~mem-control-setup include/linux/container_subsys.h
--- linux-2.6.22-rc6/include/linux/container_subsys.h~mem-control-setup 2007-07-20
13:13:24.000000000 +0530
+++ linux-2.6.22-rc6-balbir/include/linux/container_subsys.h 2007-07-20 13:13:24.000000000
+0530
@@ -30,3 +30,9 @@ SUBSYS(ns)
#endif

/* */
+
+#ifdef CONFIG_CONTAINER_MEM_CONT
+SUBSYS(mem_container)
+#endif
+
+/* */
diff -puN init/Kconfig~mem-control-setup init/Kconfig
--- linux-2.6.22-rc6/init/Kconfig~mem-control-setup 2007-07-20 13:13:24.000000000 +0530
+++ linux-2.6.22-rc6-balbir/init/Kconfig 2007-07-20 13:13:24.000000000 +0530
@@ -360,6 +360,14 @@ config CONTAINER_NS
for instance virtual servers and checkpoint/restart
jobs.

+config CONTAINER_MEM_CONT
+ bool "Memory controller for containers"
+ select CONTAINERS
+ select RESOURCE_COUNTERS
+ help
+ Provides a memory controller that manages both page cache and
+ RSS memory.
+
config PROC_PID_CPUSET

```
  bool "Include legacy /proc/<pid>/cpuset file"
  depends on CPUSETS
diff -puN mm/Makefile~mem-control-setup mm/Makefile
--- linux-2.6.22-rc6/mm/Makefile~mem-control-setup 2007-07-20 13:13:24.000000000 +0530
+++ linux-2.6.22-rc6-balbir/mm/Makefile 2007-07-20 13:13:24.000000000 +0530
@@ -30,4 +30,5 @@ obj-$(CONFIG_FS_XIP) += filemap_xip.o
 obj-$(CONFIG_MIGRATION) += migrate.o
 obj-$(CONFIG_SMP) += allocpercpu.o
 obj-$(CONFIG_QUICKLIST) += quicklist.o
+obj-$(CONFIG_CONTAINER_MEM_CONT) += memcontrol.o

diff -puN /dev/null mm/memcontrol.c
--- /dev/null 2007-06-01 20:42:04.000000000 +0530
+++ linux-2.6.22-rc6-balbir/mm/memcontrol.c 2007-07-20 13:13:24.000000000 +0530
@@ -0,0 +1,143 @@
+/* memcontrol.c - Memory Controller
+ *
+ * Copyright IBM Corporation, 2007
+ * Author Balbir Singh <balbir@linux.vnet.ibm.com>
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
+ * GNU General Public License for more details.
+ */
+
+#include <linux/res_counter.h>
+#include <linux/memcontrol.h>
+#include <linux/container.h>
+
+struct container_subsys mem_container_subsys;
+
+/*
+ * The memory controller data structure. The memory controller controls both
+ * page cache and RSS per container. We would eventually like to provide
+ * statistics based on the statistics developed by Rik Van Riel for clock-pro,
+ * to help the administrator determine what knobs to tune.
+ *
+ * TODO: Add a water mark for the memory controller. Reclaim will begin when
+ * we hit the water mark.
+ */
+struct mem_container {
+ struct container_subsys_state css;
```

```
+ /*
+  * the counter to account for memory usage
+  */
+ struct res_counter res;
+};
+
+/*
+ * A meta page is associated with every page descriptor. The meta page
+ * helps us identify information about the container
+ */
+struct meta_page {
+ struct list_head lru;  /* per container LRU list */
+ struct page *page;
+ struct mem_container *mem_container;
+};
+
+
+static inline
+struct mem_container *mem_container_from_cont(struct container *cont)
+{
+ return container_of(container_subsys_state(cont,
+    mem_container_subsys_id), struct mem_container,
+    css);
+}
+
+static ssize_t mem_container_read(struct container *cont, struct cftype *cft,
+   struct file *file, char __user *userbuf, size_t nbytes,
+   loff_t *ppos)
+{
+ return res_counter_read(&mem_container_from_cont(cont)->res,
+    cft->private, userbuf, nbytes, ppos);
+}
+
+static ssize_t mem_container_write(struct container *cont, struct cftype *cft,
+    struct file *file, const char __user *userbuf,
+    size_t nbytes, loff_t *ppos)
+{
+ return res_counter_write(&mem_container_from_cont(cont)->res,
+    cft->private, userbuf, nbytes, ppos);
+}
+
+static struct cftype mem_container_usage = {
+ .name = "mem_usage",
+ .private = RES_USAGE,
+ .read = mem_container_read,
+};
+
+static struct cftype mem_container_limit = {
```

```
+ .name = "mem_limit",
+ .private = RES_LIMIT,
+ .write = mem_container_write,
+ .read = mem_container_read,
+};
+
+static struct cftype mem_container_failcnt = {
+ .name = "mem_failcnt",
+ .private = RES_FAILCNT,
+ .read = mem_container_read,
+};
+
+static int mem_container_create(struct container_subsys *ss,
+    struct container *cont)
+{
+ struct mem_container *mem;
+
+ mem = kzalloc(sizeof(struct mem_container), GFP_KERNEL);
+ if (!mem)
+  return -ENOMEM;
+
+ res_counter_init(&mem->res);
+ cont->subsys[mem_container_subsys_id] = &mem->css;
+ mem->css.container = cont;
+ return 0;
+}
+
+static void mem_container_destroy(struct container_subsys *ss,
+    struct container *cont)
+{
+ kfree(mem_container_from_cont(cont));
+}
+
+static int mem_container_populate(struct container_subsys *ss,
+    struct container *cont)
+{
+ int rc = 0;
+
+ rc = container_add_file(cont, &mem_container_usage);
+ if (rc < 0)
+  goto err;
+
+ rc = container_add_file(cont, &mem_container_limit);
+ if (rc < 0)
+  goto err;
+
+ rc = container_add_file(cont, &mem_container_failcnt);
+ if (rc < 0)
```

```
+  goto err;
+
+err:
+ return rc;
+}
+
+struct container_subsys mem_container_subsys = {
+ .name = "mem_container",
+ .subsys_id = mem_container_subsys_id,
+ .create = mem_container_create,
+ .destroy = mem_container_destroy,
+ .populate = mem_container_populate,
+ .early_init = 0,
+};
diff -puN /dev/null include/linux/memcontrol.h
--- /dev/null 2007-06-01 20:42:04.000000000 +0530
+++ linux-2.6.22-rc6-balbir/include/linux/memcontrol.h 2007-07-20 13:13:24.000000000 +0530
@@ -0,0 +1,21 @@
+/* memcontrol.h - Memory Controller
+ *
+ * Copyright IBM Corporation, 2007
+ * Author Balbir Singh <balbir@linux.vnet.ibm.com>
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
+ * GNU General Public License for more details.
+ */
+
+#ifndef _LINUX_MEMCONTROL_H
+#define _LINUX_MEMCONTROL_H
+
+#endif /* _LINUX_MEMCONTROL_H */
+

_

--
 Warm Regards,
 Balbir Singh
 Linux Technology Center
 IBM, ISTL
```

_____

Containers mailing list