
Subject: [PATCH 23/33] memory controller containers setup v7
Posted by [Paul Menage](#) on Mon, 17 Sep 2007 21:03:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Balbir Singh <balbir@linux.vnet.ibm.com>
(container->cgroup renaming by Paul Menage <menage@google.com>)

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

Signed-off-by: Balbir Singh <balbir@linux.vnet.ibm.com>
Signed-off-by: Paul Menage <menage@google.com>

```
include/linux/cgroup_subsys.h | 6 +
include/linux/memcontrol.h    | 21 +++++
init/Kconfig                  | 7 +
mm/Makefile                   | 1
mm/memcontrol.c               | 127 ++++++
5 files changed, 162 insertions(+)
```

diff -puN include/linux/cgroup_subsys.h~memory-controller-cgroups-setup-v7

include/linux/cgroup_subsys.h

--- a/include/linux/cgroup_subsys.h~memory-controller-cgroups-setup-v7

+++ a/include/linux/cgroup_subsys.h

@@ -30,3 +30,9 @@ SUBSYS(ns)

#endif

/* */

+

+#ifdef CONFIG_CGROUP_MEM_CONT

+SUBSYS(mem_cgroup)

+#endif

+

/* */

diff -puN /dev/null include/linux/memcontrol.h

--- /dev/null

+++ a/include/linux/memcontrol.h

@@ -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 */
+
diff -puN init/Kconfig~memory-controller-cgroups-setup-v7 init/Kconfig
--- a/init/Kconfig~memory-controller-cgroups-setup-v7
+++ a/init/Kconfig
@@ -346,6 +346,13 @@ config SYSFS_DEPRECATED
    If you are using a distro that was released in 2006 or later,
    it should be safe to say N here.

+config CGROUP_MEM_CONT
+ bool "Memory controller for cgroups"
+ depends on CGROUPS && 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~memory-controller-cgroups-setup-v7 mm/Makefile
--- a/mm/Makefile~memory-controller-cgroups-setup-v7
+++ a/mm/Makefile
@@ -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_CGROUP_MEM_CONT) += memcontrol.o

diff -puN /dev/null mm/memcontrol.c
--- /dev/null
+++ a/mm/memcontrol.c
@@ -0,0 +1,127 @@
+/* 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/cgroup.h>
+
+struct cgroup_subsys mem_cgroup_subsys;
+
+/*
+ * The memory controller data structure. The memory controller controls both
+ * page cache and RSS per cgroup. 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_cgroup {
+ struct cgroup_subsys_state css;
+ /*
+  * the counter to account for memory usage
+  */
+ struct res_counter res;
+};
+
+/*
+ * A page_cgroup page is associated with every page descriptor. The
+ * page_cgroup helps us identify information about the cgroup
+ */
+struct page_cgroup {
+ struct list_head lru; /* per cgroup LRU list */
+ struct page *page;
+ struct mem_cgroup *mem_cgroup;
+};
+
+static inline
+struct mem_cgroup *mem_cgroup_from_cont(struct cgroup *cont)
+{

```

```

+ return container_of(cgroup_subsys_state(cont,
+ mem_cgroup_subsys_id), struct mem_cgroup,
+ css);
+}
+
+static ssize_t mem_cgroup_read(struct cgroup *cont, struct cftype *cft,
+ struct file *file, char __user *userbuf, size_t nbytes,
+ loff_t *ppos)
+{
+ return res_counter_read(&mem_cgroup_from_cont(cont)->res,
+ cft->private, userbuf, nbytes, ppos);
+}
+
+static ssize_t mem_cgroup_write(struct cgroup *cont, struct cftype *cft,
+ struct file *file, const char __user *userbuf,
+ size_t nbytes, loff_t *ppos)
+{
+ return res_counter_write(&mem_cgroup_from_cont(cont)->res,
+ cft->private, userbuf, nbytes, ppos);
+}
+
+static struct cftype mem_cgroup_files[] = {
+ {
+ .name = "usage",
+ .private = RES_USAGE,
+ .read = mem_cgroup_read,
+ },
+ {
+ .name = "limit",
+ .private = RES_LIMIT,
+ .write = mem_cgroup_write,
+ .read = mem_cgroup_read,
+ },
+ {
+ .name = "failcnt",
+ .private = RES_FAILCNT,
+ .read = mem_cgroup_read,
+ },
+ };
+
+static struct cgroup_subsys_state *
+mem_cgroup_create(struct cgroup_subsys *ss, struct cgroup *cont)
+{
+ struct mem_cgroup *mem;
+
+ mem = kzalloc(sizeof(struct mem_cgroup), GFP_KERNEL);
+ if (!mem)
+ return -ENOMEM;

```

```
+
+ res_counter_init(&mem->res);
+ return &mem->css;
+}
+
+static void mem_cgroup_destroy(struct cgroup_subsys *ss,
+    struct cgroup *cont)
+{
+    kfree(mem_cgroup_from_cont(cont));
+}
+
+static int mem_cgroup_populate(struct cgroup_subsys *ss,
+    struct cgroup *cont)
+{
+    return cgroup_add_files(cont, ss, mem_cgroup_files,
+        ARRAY_SIZE(mem_cgroup_files));
+}
+
+struct cgroup_subsys mem_cgroup_subsys = {
+    .name = "memory",
+    .subsys_id = mem_cgroup_subsys_id,
+    .create = mem_cgroup_create,
+    .destroy = mem_cgroup_destroy,
+    .populate = mem_cgroup_populate,
+    .early_init = 0,
+};
```

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