Subject: [patch -rss] Make RSS accounting display more user friendly
Posted by Balbir Singh on Wed, 20 Jun 2007 15:46:55 GMT
View Forum Message <> Reply to Message

Display the current usage and limit in a more user friendly manner. Number
of pages can be confusing if the page size is different. Some systems
can choose a page size of 64KB.

NOTE: Values shown in MB and GB could be rounded off.

Sample display

:~ # cat /container/rss_limit
17179869183 (GB), 9223372036854775807 pages
:~ # cat /container/rss_usage
36 (MB), 9220 pages
:~ #

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

 include/linux/res_counter.h | 6 ++++++
 kernel/res_counter.c | 27 ++++++++++++++++++++++++++----
 2 files changed, 29 insertions(+), 4 deletions(-)

diff -puN kernel/res_counter.c~rss-ui-display-kdb kernel/res_counter.c
--- linux-2.6.22-rc24-mm2/kernel/res_counter.c~rss-ui-display-kdb 2007-06-20
20:15:46.000000000 +0530
+++ linux-2.6.22-rc24-mm2-balbir/kernel/res_counter.c 2007-06-20 21:12:29.000000000 +0530
@@ -61,7 +61,8 @@ void res_counter_uncharge(struct res_cou
 }

-static inline unsigned long *res_counter_member(struct res_counter *cnt, int member)
+static inline unsigned long *res_counter_member(struct res_counter *cnt,
+   int member, int *format_mem)
 {
  switch (member) {
  case RES_USAGE:
@@ -69,6 +70,7 @@ static inline unsigned long *res_counter
  case RES_LIMIT:
   return &cnt->limit;
  case RES_FAILCNT:
+ *format_mem = 0;
   return &cnt->failcnt;
  };

@@ -81,10 +83,26 @@ ssize_t res_counter_read(struct res_coun

```
 {
  unsigned long *val;
  char buf[64], *s;
+ int format_mem = 1;
+ unsigned long val_kb = 0, val_mb = 0, val_gb = 0;

  s = buf;
- val = res_counter_member(cnt, member);
- s += sprintf(s, "%lu\n", *val);
+ val = res_counter_member(cnt, member, &format_mem);
+ /*
+  * NOTE: GB and MB values might be rounded off
+  */
+ if (format_mem) {
+  val_gb = (*val << PAGE_SHIFT) >> GB_SHIFT;
+  val_mb = (*val << PAGE_SHIFT) >> MB_SHIFT;
+  val_kb = (*val << PAGE_SHIFT) >> KB_SHIFT;
+  if (val_gb)
+   s += sprintf(s, "%lu (GB), %lu pages\n", val_gb, *val);
+  else if (val_mb)
+   s += sprintf(s, "%lu (MB), %lu pages\n", val_mb, *val);
+  else
+   s += sprintf(s, "%lu (KB), %lu pages\n", val_kb, *val);
+ } else
+  s += sprintf(s, "%lu\n", *val);
  return simple_read_from_buffer((void __user *)userbuf, nbytes,
    pos, buf, s - buf);
 }
@@ -95,6 +113,7 @@ ssize_t res_counter_write(struct res_cou
  int ret;
  char *buf, *end;
  unsigned long tmp, *val;
+ int nop;

  buf = kmalloc(nbytes + 1, GFP_KERNEL);
  ret = -ENOMEM;
@@ -111,7 +130,7 @@ ssize_t res_counter_write(struct res_cou
  if (*end != '\0')
   goto out_free;

- val = res_counter_member(cnt, member);
+ val = res_counter_member(cnt, member, &nop);
  *val = tmp;
  ret = nbytes;
 out_free:
diff -puN include/linux/res_counter.h~rss-ui-display-kdb include/linux/res_counter.h
--- linux-2.6.22-rc24-mm2/include/linux/res_counter.h~rss-ui-display-kdb 2007-06-20
20:15:46.000000000 +0530
```

```
+++ linux-2.6.22-rc24-mm2-balbir/include/linux/res_counter.h 2007-06-20 21:01:57.000000000
+0530
@@ -13,6 +13,12 @@

 #include <linux/container.h>

+#ifndef KB_SHIFT
+#define KB_SHIFT 10
+#define MB_SHIFT 20
+#define GB_SHIFT 30
+#endif
+
 /*
  * the core object. the container that wishes to account for some
  * resource may include this counter into its structures and use
_

--
 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

---