Subject: [1/1] [PATCH -mm] Update getdelays to become containerstats aware
Posted by Balbir Singh on Fri, 08 Jun 2007 18:14:00 GMT
View Forum Message <> Reply to Message

Update the getdelays utility to become containerstats aware. A new -C option
has been added. It takes in a container path and prints out a summary of
task states in the container.

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

 Documentation/accounting/getdelays.c | 43 ++-
 1 file changed, 42 insertions(+), 1 deletion(-)

diff -puN Documentation/accounting/getdelays.c~containerstats-update-getdelays
Documentation/accounting/getdelays.c
---
linux-2.6.22-rc24-mm2/Documentation/accounting/getdelays.c~containerstats-update-getdelays 2
007-06-08 23:11:56.000000000 +0530
+++ linux-2.6.22-rc24-mm2-balbir/Documentation/accounting/getdelays.c 2007-06-08
23:12:47.000000000 +0530
@@ -26,6 +26,7 @@

 #include <linux/genetlink.h>
 #include <linux/taskstats.h>
+#include <linux/containerstats.h>

 /*
  * Generic macros for dealing with netlink sockets. Might be duplicated
@@ -79,6 +80,7 @@ static void usage(void)
  fprintf(stderr, " -i: print IO accounting (works only with -p)\n");
  fprintf(stderr, " -l: listen forever\n");
  fprintf(stderr, " -v: debug on\n");
+ fprintf(stderr, " -C: container path\n");
 }

 /*
@@ -213,6 +215,14 @@ void task_context_switch_counts(struct t
        t->nvcsw, t->nivcsw);
 }

+void print_containerstats(struct containerstats *c)
+{
+ printf("sleeping %llu, blocked %llu, running %llu, stopped %llu, "
+ "uninterruptible %llu\n", c->nr_sleeping, c->nr_io_wait,
+ c->nr_running, c->nr_stopped, c->nr_uninterruptible);
+}

```
+
+
 void print_ioacct(struct taskstats *t)
 {
  printf("%s: read=%llu, write=%llu, cancelled_write=%llu\n",
@@ -240,11 +250,14 @@ int main(int argc, char *argv[])
  int maskset = 0;
  char *logfile = NULL;
  int loop = 0;
+ int containerset = 0;
+ char containerpath[1024];
+ int cfd = 0;

  struct msgtemplate msg;

  while (1) {
-  c = getopt(argc, argv, "qdiw:r:m:t:p:vl");
+  c = getopt(argc, argv, "qdiw:r:m:t:p:vlC:");
   if (c < 0)
    break;

@@ -261,6 +274,10 @@ int main(int argc, char *argv[])
    printf("printing task/process context switch rates\n");
    print_task_context_switch_counts = 1;
    break;
+  case 'C':
+   containerset = 1;
+   strncpy(containerpath, optarg, strlen(optarg) + 1);
+   break;
   case 'w':
    logfile = strdup(optarg);
    printf("write to file %s\n", logfile);
@@ -335,6 +352,11 @@ int main(int argc, char *argv[])
   }
  }

+ if (tid && containerset) {
+  fprintf(stderr, "Select either -t or -C, not both\n");
+  goto err;
+ }
+
  if (tid) {
   rc = send_cmd(nl_sd, id, mypid, TASKSTATS_CMD_GET,
         cmd_type, &tid, sizeof(__u32));
@@ -345,6 +367,20 @@ int main(int argc, char *argv[])
   }
  }
```

```
+ if (containerset) {
+  cfd = open(containerpath, O_RDONLY);
+  if (cfd < 0) {
+   perror("error opening container file");
+   goto err;
+  }
+  rc = send_cmd(nl_sd, id, mypid, CONTAINERSTATS_CMD_GET,
+        CONTAINERSTATS_CMD_ATTR_FD, &cfd, sizeof(__u32));
+  if (rc < 0) {
+   perror("error sending containerstats command");
+   goto err;
+  }
+ }
+
  do {
   int i;

@@ -423,6 +459,9 @@ int main(int argc, char *argv[])
     }
     break;

+   case CONTAINERSTATS_TYPE_CONTAINER_STATS:
+    print_containerstats(NLA_DATA(na));
+    break;
    default:
     fprintf(stderr, "Unknown nla_type %d\n",
      na->nla_type);
@@ -444,5 +483,7 @@ err:
  close(nl_sd);
  if (fd)
   close(fd);
+ if (cfd)
+  close(cfd);
  return 0;
 }
```

_

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

_____