Subject: [PATCH 0/3] Consolidate cgroup files creation for resource counters
Posted by Pavel Emelianov on Wed, 03 Oct 2007 10:53:27 GMT
View Forum Message <> Reply to Message

Right now we have only one controller in -mm tree - the memory
one - and it initializes all its files manually. After I developed
the kernel memory one it turned out, that the names of resource
counter specific files has changed. In the future, if anything in
the resource counters change, we'll have to rewrite all the
controllers we will have by that time.

To make all the containers working with the resource counters
look similar I propose to create the resource counters specific
files in one place.

Any comments/wishes before I send it to Andrew are welcome.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

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

Subject: [PATCH 1/3] Typedefs the read and write functions in cftype
Posted by Pavel Emelianov on Wed, 03 Oct 2007 10:55:09 GMT
View Forum Message <> Reply to Message

This is just to reduce the code amount in the future.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

```
diff --git a/include/linux/cgroup.h b/include/linux/cgroup.h
index 8747932..0635004 100644
--- a/include/linux/cgroup.h
+++ b/include/linux/cgroup.h
@@ -178,6 +178,15 @@ struct css_set {
  * - the 'cftype' of the file is file->f_dentry->d_fsdata
  */

+struct cftype;
+
+typedef ssize_t (*cft_read) (struct cgroup *cont, struct cftype *cft,
+    struct file *file,
+    char __user *buf, size_t nbytes, loff_t *ppos);
+typedef ssize_t (*cft_write) (struct cgroup *cont, struct cftype *cft,
```

```
+     struct file *file,
+     const char __user *buf, size_t nbytes, loff_t *ppos);
+
 #define MAX_CFTYPE_NAME 64
 struct cftype {
  /* By convention, the name should begin with the name of the
@@ -185,18 +194,14 @@ struct cftype {
  char name[MAX_CFTYPE_NAME];
  int private;
  int (*open) (struct inode *inode, struct file *file);
- ssize_t (*read) (struct cgroup *cont, struct cftype *cft,
-    struct file *file,
-    char __user *buf, size_t nbytes, loff_t *ppos);
+ cft_read read;
  /*
   * read_uint() is a shortcut for the common case of returning a
   * single integer. Use it in place of read()
   */
  u64 (*read_uint) (struct cgroup *cont, struct cftype *cft);
- ssize_t (*write) (struct cgroup *cont, struct cftype *cft,
-     struct file *file,
-     const char __user *buf, size_t nbytes, loff_t *ppos);

+ cft_write write;
  /*
   * write_uint() is a shortcut for the common case of accepting
   * a single integer (as parsed by simple_strtoull) from
```



---

## Subject: [PATCH 2/3] Introduce the res_counter_populate() function
Posted by Pavel Emelianov on Wed, 03 Oct 2007 10:57:54 GMT
View Forum Message <> Reply to Message

This one is responsible for initializing the RES_CFT_MAX files
properly and register them inside the container.

The caller must provide the cgroup, the cgroup_subsys, the
RES_CFT_MAX * sizeof(cftype) chunk of uninitialized memory
and the read and write callbacks.

In the future, if we add more res_counter files, change their
names or anything else, no caller will be affected.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

```
diff --git a/include/linux/res_counter.h b/include/linux/res_counter.h
index 61363ce..8ee8316 100644
--- a/include/linux/res_counter.h
+++ b/include/linux/res_counter.h
@@ -58,6 +58,9 @@ ssize_t res_counter_write(struct res_cou
   const char __user *buf, size_t nbytes, loff_t *pos,
   int (*write_strategy)(char *buf, unsigned long long *val));

+int res_counter_populate(struct cgroup_subsys *ss, struct cgroup *cont,
+  struct cftype files[], cft_read read_fn, cft_write write_fn);
+
 /*
  * the field descriptors. one for each member of res_counter
  */
@@ -66,6 +69,8 @@ enum {
  RES_USAGE,
  RES_LIMIT,
  RES_FAILCNT,
+
+ RES_CFT_MAX,
 };

 /*
diff --git a/kernel/res_counter.c b/kernel/res_counter.c
index d7f43cd..018eb2b 100644
--- a/kernel/res_counter.c
+++ b/kernel/res_counter.c
@@ -10,9 +10,34 @@
 #include <linux/types.h>
 #include <linux/parser.h>
 #include <linux/fs.h>
+#include <linux/cgroup.h>
 #include <linux/res_counter.h>
 #include <linux/uaccess.h>

+int res_counter_populate(struct cgroup_subsys *ss, struct cgroup *cont,
+  struct cftype files[], cft_read read_fn, cft_write write_fn)
+{
+ /*
+  * to be on the safe side
+  */
+ memset(files, 0, RES_CFT_MAX * sizeof(struct cftype));
+
+ strcpy(files[RES_USAGE].name, "usage_in_bytes");
```

```
+ files[RES_USAGE].private = RES_USAGE;
+ files[RES_USAGE].read = read_fn;
+
+ strcpy(files[RES_LIMIT].name, "limit_in_bytes");
+ files[RES_LIMIT].private = RES_LIMIT;
+ files[RES_LIMIT].read = read_fn;
+ files[RES_LIMIT].write = write_fn;
+
+ strcpy(files[RES_FAILCNT].name, "failcnt");
+ files[RES_FAILCNT].private = RES_FAILCNT;
+ files[RES_FAILCNT].read = read_fn;
+
+ return cgroup_add_files(cont, ss, files, RES_CFT_MAX);
+}
+
 void res_counter_init(struct res_counter *counter)
 {
  spin_lock_init(&counter->lock);
```



---

Subject: [PATCH 3/3] Use the res_counter_populate in memory controller
Posted by Pavel Emelianov on Wed, 03 Oct 2007 10:59:40 GMT
View Forum Message <> Reply to Message

Note, that the controller code dealing with the cftype files
for resource counters becomes much shorter and won't have to
be changed in the future.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

```
diff --git a/mm/memcontrol.c b/mm/memcontrol.c
index 1b8bf24..d1b43bc 100644
--- a/mm/memcontrol.c
+++ b/mm/memcontrol.c
@@ -519,28 +519,14 @@ static ssize_t mem_control_type_read(str

 static struct cftype mem_cgroup_files[] = {
  {
-  .name = "usage_in_bytes",
-  .private = RES_USAGE,
-  .read = mem_cgroup_read,
```

```
- },
- {
-  .name = "limit_in_bytes",
-  .private = RES_LIMIT,
-  .write = mem_cgroup_write,
-  .read = mem_cgroup_read,
- },
- {
-  .name = "failcnt",
-  .private = RES_FAILCNT,
-  .read = mem_cgroup_read,
- },
- {
   .name = "control_type",
   .write = mem_control_type_write,
   .read = mem_control_type_read,
  },
 };

+static struct cftype mem_res_counter_files[RES_CFT_MAX];
+
 static struct mem_cgroup init_mem_cgroup;

 static struct cgroup_subsys_state *
@@ -574,6 +560,13 @@ static void mem_cgroup_destroy(struct cg
 static int mem_cgroup_populate(struct cgroup_subsys *ss,
     struct cgroup *cont)
 {
+ int ret;
+
+ ret = res_counter_populate(ss, cont, mem_res_counter_files,
+    mem_cgroup_read, mem_cgroup_write);
+ if (ret)
+  return ret;
+
  return cgroup_add_files(cont, ss, mem_cgroup_files,
      ARRAY_SIZE(mem_cgroup_files));
 }
```



Subject: Re: [PATCH 2/3] Introduce the res_counter_populate() function
Posted by Paul Menage on Wed, 03 Oct 2007 16:02:42 GMT
View Forum Message <> Reply to Message

On 10/3/07, Pavel Emelyanov <xemul@openvz.org> wrote:
> +
> +	strcpy(files[RES_USAGE].name, "usage_in_bytes");

This prevents the resource counters from being used for anything that
doesn't measure its limit/usage in bytes. E.g. number of tasks.

I like the idea of having a unified interface, but can't they just be
called "usage" and "limit"?

Paul



---

Subject: Re: [PATCH 0/3] Consolidate cgroup files creation for resource counters
Posted by Balbir Singh on Wed, 03 Oct 2007 17:59:02 GMT
View Forum Message <> Reply to Message

Pavel Emelyanov wrote:
> Right now we have only one controller in -mm tree - the memory
> one - and it initializes all its files manually. After I developed
> the kernel memory one it turned out, that the names of resource
> counter specific files has changed. In the future, if anything in
> the resource counters change, we'll have to rewrite all the
> controllers we will have by that time.
>
> To make all the containers working with the resource counters
> look similar I propose to create the resource counters specific
> files in one place.
>
> Any comments/wishes before I send it to Andrew are welcome.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

For memory your changes make sense, but the file names cannot
be limit_in_bytes and usage_in_bytes for other controllers
(if and when they use res_counters)

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

Subject: Re: [PATCH 2/3] Introduce the res_counter_populate() function
Posted by Balbir Singh on Wed, 03 Oct 2007 18:02:29 GMT
View Forum Message <> Reply to Message

Pavel Emelyanov wrote:
>
> diff --git a/include/linux/res_counter.h b/include/linux/res_counter.h
> index 61363ce..8ee8316 100644
> --- a/include/linux/res_counter.h
> +++ b/include/linux/res_counter.h
> @@ -58,6 +58,9 @@ ssize_t res_counter_write(struct res_cou
>  const char __user *buf, size_t nbytes, loff_t *pos,
>  int (*write_strategy)(char *buf, unsigned long long *val));
>
> +int res_counter_populate(struct cgroup_subsys *ss, struct cgroup *cont,
> + struct cftype files[], cft_read read_fn, cft_write write_fn);
> +
>  /*
>  * the field descriptors. one for each member of res_counter
>  */
> @@ -66,6 +69,8 @@ enum {
>  RES_USAGE,
>  RES_LIMIT,
>  RES_FAILCNT,
> +
> + RES_CFT_MAX,
>  };
>

This is good

>  /*
> diff --git a/kernel/res_counter.c b/kernel/res_counter.c
> index d7f43cd..018eb2b 100644
> --- a/kernel/res_counter.c
> +++ b/kernel/res_counter.c
> @@ -10,9 +10,34 @@
>  #include <linux/types.h>
>  #include <linux/parser.h>
>  #include <linux/fs.h>
> +#include <linux/cgroup.h>
>  #include <linux/res_counter.h>
>  #include <linux/uaccess.h>
>

> +int res_counter_populate(struct cgroup_subsys *ss, struct cgroup *cont,
> +  struct cftype files[], cft_read read_fn, cft_write write_fn)
> +{
> + /*
> +  * to be on the safe side
> +  */
> + memset(files, 0, RES_CFT_MAX * sizeof(struct cftype));
> +
> + strcpy(files[RES_USAGE].name, "usage_in_bytes");

This needs to be controller pluggable

> + files[RES_USAGE].private = RES_USAGE;
> + files[RES_USAGE].read = read_fn;
> +
> + strcpy(files[RES_LIMIT].name, "limit_in_bytes");

This needs to be controller pluggable

> + files[RES_LIMIT].private = RES_LIMIT;
> + files[RES_LIMIT].read = read_fn;
> + files[RES_LIMIT].write = write_fn;
> +
> + strcpy(files[RES_FAILCNT].name, "failcnt");
> + files[RES_FAILCNT].private = RES_FAILCNT;
> + files[RES_FAILCNT].read = read_fn;
> +
> + return cgroup_add_files(cont, ss, files, RES_CFT_MAX);
> +}
> +

I've been thinking of writing a tool, that will generate all
the code necessary to write a controller under cgroups.
Another TODO (low priority).

>  void res_counter_init(struct res_counter *counter)
>  {
>   spin_lock_init(&counter->lock);
>
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers

--
Warm Regards,
Balbir Singh

Linux Technology Center
IBM, ISTL



Subject: Re: [PATCH 2/3] Introduce the res_counter_populate() function
Posted by Pavel Emelianov on Thu, 04 Oct 2007 06:40:23 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 10/3/07, Pavel Emelyanov <xemul@openvz.org> wrote:
>> +
>> + strcpy(files[RES_USAGE].name, "usage_in_bytes");
>
> This prevents the resource counters from being used for anything that
> doesn't measure its limit/usage in bytes. E.g. number of tasks.
>
> I like the idea of having a unified interface, but can't they just be
> called "usage" and "limit"?

I see. OK, I will rework the patches to make them more generic.
Moreover, I've just faced one more race in this set :(

> Paul
>

Thanks,
Pavel