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

_____
Containers mailing list