Subject: Re: [RFC][patch 3/11][CFQ-cgroup] Introduce cgroup subsystem
Posted by Paul Menage on Wed, 02 Apr 2008 22:41:58 GMT
View Forum Message <> Reply to Message

On Tue, Apr 1, 2008 at 2:32 AM, Satoshi UCHIDA <s-uchida@ap.jp.nec.com> wrote:
> This patch introduces a simple cgroup subsystem.
> New cgroup subsystem is called cfq_cgroup.
>
>   Signed-off-by: Satoshi UCHIDA <uchida@ap.jp.nec.com>
>
> diff --git a/block/Makefile b/block/Makefile
> index 5a43c7d..ea07b46 100644
> --- a/block/Makefile
> +++ b/block/Makefile
> @@ -11,6 +11,7 @@ obj-$(CONFIG_IOSCHED_NOOP)   += noop-iosched.o
>   obj-$(CONFIG_IOSCHED_AS)   += as-iosched.o
>   obj-$(CONFIG_IOSCHED_DEADLINE) += deadline-iosched.o
>   obj-$(CONFIG_IOSCHED_CFQ)   += cfq-iosched.o
> +obj-$(CONFIG_CGROUP_CFQ)   += cfq-cgroup.o
>
>   obj-$(CONFIG_BLK_DEV_IO_TRACE) += blktrace.o
>   obj-$(CONFIG_BLOCK_COMPAT)   += compat_ioctl.o
> diff --git a/block/cfq-cgroup.c b/block/cfq-cgroup.c
> new file mode 100644
> index 0000000..cea2b92
> --- /dev/null
> +++ b/block/cfq-cgroup.c
> @@ -0,0 +1,57 @@
> +/*
> + * CFQ CGROUP disk scheduler.
> + *
> + *   This program is a wrapper program that is
> + *   extend CFQ disk scheduler for handling
> + *   cgroup subsystem.
> + *
> + *   This program is based on original CFQ code.
> + *
> + * Copyright (C) 2008 Satoshi UCHIDA <s-uchida@ap.jp.nec.com>
> + *   and NEC Corp.
> + */
> +
> +#include <linux/blkdev.h>
> +#include <linux/cgroup.h>
> +#include <linux/cfq-iosched.h>
> +
> +struct cfq_cgroup {
> +   struct cgroup_subsys_state css;
> +};

```
> +
> +
>  +static inline struct cfq_cgroup *cgroup_to_cfq_cgroup(struct cgroup *cont)
>  +{
>  +	return container_of(cgroup_subsys_state(cont, cfq_cgroup_subsys_id),
>  +			    struct cfq_cgroup, css);
>  +}
>  +
>  +static struct cgroup_subsys_state *
>  +cfq_cgroup_create(struct cgroup_subsys *ss, struct cgroup *cont)
>  +{
>  +	struct cfq_cgroup *cfqc;
>  +
>  +	if (!capable(CAP_SYS_ADMIN))
>  +		return ERR_PTR(-EPERM);
>  +
>  +	if (!cgroup_is_descendant(cont))
>  +		return ERR_PTR(-EPERM);
```

What are these checks for? Cgroups already provides filesystem permissions to control directory creation, and the "descendant" check looks like it may have been cut/pasted from the nsproxy subsystem.

```
>  +
>  +	cfqc = kzalloc(sizeof(struct cfq_cgroup), GFP_KERNEL);
>  +	if (unlikely(!cfqc))
>  +		return ERR_PTR(-ENOMEM);
>  +
>  +	return &cfqc->css;
>  +}
>  +
>  +static void cfq_cgroup_destroy(struct cgroup_subsys *ss, struct cgroup *cont)
>  +{
>  +	kfree(cgroup_to_cfq_cgroup(cont));
>  +}
>  +
>  +struct cgroup_subsys cfq_cgroup_subsys = {
>  +	.name = "cfq_cgroup",
>  +	.create = cfq_cgroup_create,
>  +	.destroy = cfq_cgroup_destroy,
>  +	.subsys_id = cfq_cgroup_subsys_id,
>  +};
>  diff --git a/include/linux/cgroup_subsys.h b/include/linux/cgroup_subsys.h
>  index 1ddebfc..5d2e991 100644
>  --- a/include/linux/cgroup_subsys.h
>  +++ b/include/linux/cgroup_subsys.h
>  @@ -42,3 +42,9 @@ SUBSYS(mem_cgroup)
```

> #endif
>
> /* */
> +
> +#ifdef CONFIG_CGROUP_CFQ
> +SUBSYS(cfq_cgroup)
> +#endif
> +
> +/* */

To fit with the convention for other subsystems, simply "cfq" would be a better name than "cfq_cgroup". (Clearly it's a cgroup subsystem from context).

Is this subsystem meant to allow you to control any device that uses CFQ, or is it specific to disks? It would be nice to be able to allow different groups have different guarantees on different disks.

Paul

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