
Subject: [RFC][PATCH 2/9] cgroups: block: cfq: I/O bandwidth controlling subsystem for CGroups based on CFQ

Posted by [Vasily Tarasov](#) on Fri, 15 Feb 2008 06:59:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

From: Vasily Tarasov <vtaras@openvz.org>

Registers the cfqio_subsys subsystem in the CGroups framework.

Signed-off-by: Vasily Tarasov <vtaras@openvz.org>

--- /dev/null 2008-02-14 08:57:49.255923728 -0500

+++ linux-2.6.25-rc5-mm1/include/linux/cfqio-cgroup.h 2008-02-15 01:06:40.000000000 -0500

@@ -0,0 +1,26 @@

+/*

+ * include/linux/cfqio-cgroup.h

+ *

+ * cfqio_subsys: CGroup subsystem that allows CFQ recognize

+ * cgroups and perform scheduling according to this.

+ *

+ * Copyright (C) 2008 OpenVZ <http://openvz.org>

+ *

+ * Author: Vasily Tarasov <vtaras@openvz.org>

+ *

+ */

+

+#ifndef _LINUX_CFQIO_CGROUP_H

+#define _LINUX_CFQIO_CGROUP_H

+

+#define CFQIO_SS_IOPRIO_DEF 4

+#define CFQIO_SS_IOPRIO_MAX 7

+#define CFQIO_SS_IOPRIO_MIN 0

+

+/* cfqio_subsys's per cgroup state holder */

+struct cfqio_ss_css {

+ struct cgroup_subsys_state css;

+ unsigned int ioprio;

+};

+

+#endif /* _LINUX_CFQIO_CGROUP_H */

--- linux-2.6.25-rc5-mm1/include/linux/cgroup_subsys.h.subsys 2008-02-15 00:49:56.000000000 -0500

+++ linux-2.6.25-rc5-mm1/include/linux/cgroup_subsys.h 2008-02-15 01:06:40.000000000 -0500

@@ -42,3 +42,9 @@ SUBSYS(mem_cgroup)

#endif

```

/* */
+
+ #ifdef CONFIG_CGROUP_CFQIO
+ SUBSYS(cfqio)
+ #endif
+
+ /* */
--- linux-2.6.25-rc5-mm1/block/Makefile.subsys 2008-02-15 00:49:11.000000000 -0500
+++ linux-2.6.25-rc5-mm1/block/Makefile 2008-02-15 01:06:40.000000000 -0500
@@ -11,6 +11,7 @@ obj-$(CONFIG_IOSCHED_NOOP) += noop-iosch
obj-$(CONFIG_IOSCHED_AS) += as-iosched.o
obj-$(CONFIG_IOSCHED_DEADLINE) += deadline-iosched.o
obj-$(CONFIG_IOSCHED_CFQ) += cfq-iosched.o
+obj-$(CONFIG_CGROUP_CFQIO) += cfqio-cgroup.o

obj-$(CONFIG_BLK_DEV_IO_TRACE) += blktrace.o
obj-$(CONFIG_BLOCK_COMPAT) += compat_ioctl.o
--- /dev/null 2008-02-14 08:57:49.255923728 -0500
+++ linux-2.6.25-rc5-mm1/block/cfqio-cgroup.c 2008-02-15 01:06:40.000000000 -0500
@@ -0,0 +1,97 @@
+ /*
+  * block/cfqio-cgroup.c
+  *
+  * cfqio_cgroup: CGroup subsystem that allows CFQ recognize
+  * cgroups and perform scheduling according to this.
+  *
+  * Copyright (C) 2008 OpenVZ http://openvz.org
+  *
+  * Author: Vasily Tarasov <vtaras@openvz.org>
+  *
+  */
+
+ #include <linux/cgroup.h>
+ #include <linux/cfqio-cgroup.h>
+ #include <linux/err.h>
+
+ static void cfqio_ss_fini(struct cfqio_ss_css *cfqio_css)
+ {
+ }
+
+ static void cfqio_ss_init(struct cfqio_ss_css *cfqio_css)
+ {
+   cfqio_css->ioprio = CFQIO_SS_IOPRIO_DEF;
+ }
+
+ static struct cgroup_subsys_state *
+ cfqio_ss_cgrp_create(struct cgroup_subsys *subsys, struct cgroup *cgrp)
+ {

```

```

+ struct cfqio_ss_css *cfqio_css;
+
+ cfqio_css = kmalloc(sizeof(*cfqio_css), GFP_KERNEL);
+ if (!cfqio_css)
+ return ERR_PTR(-ENOMEM);
+
+ cfqio_ss_init(cfqio_css);
+
+ return &cfqio_css->css;
+}
+
+static inline struct cfqio_ss_css *
+cfqio_ss_cgrp_css(struct cgroup *cgrp)
+{
+ return container_of(cgrp->subsys[cfqio_subsys_id],
+ struct cfqio_ss_css, css);
+}
+
+static void
+cfqio_ss_cgrp_destroy(struct cgroup_subsys *subsys, struct cgroup *cgrp)
+{
+ struct cfqio_ss_css *cfqio_css;
+
+ cfqio_css = cfqio_ss_cgrp_css(cgrp);
+ cfqio_ss_fini(cfqio_css);
+ kfree(cfqio_css);
+ return;
+}
+
+static u64
+cfqio_ss_ioprio_read(struct cgroup *cgrp, struct cftype *cft)
+{
+ return (u64)cfqio_ss_cgrp_css(cgrp)->ioprio;
+}
+
+static int
+cfqio_ss_ioprio_write(struct cgroup *cgrp, struct cftype *cft, u64 val)
+{
+ if (val > CFQIO_SS_IOPRIO_MAX || val < CFQIO_SS_IOPRIO_MIN)
+ return -EINVAL;
+
+ cfqio_ss_cgrp_css(cgrp)->ioprio = val;
+
+ return 0;
+}
+
+/* array since more than one file are expected in the future */
+static struct cftype cfqio_ss_files[] = {

```

```
+ {
+ .name = "ioprio",
+ .read_uint = cfqio_ss_ioprio_read,
+ .write_uint = cfqio_ss_ioprio_write,
+ },
+};
+
+static int
+cfqio_ss_populate_dir(struct cgroup_subsys *ss, struct cgroup *cgrp)
+{
+ return cgroup_add_files(cgrp, ss, cfqio_ss_files,
+   ARRAY_SIZE(cfqio_ss_files));
+}
+
+struct cgroup_subsys cfqio_subsys = {
+ .name = "cfqio_subsys",
+ .subsys_id = cfqio_subsys_id,
+ .create = cfqio_ss_cgrp_create,
+ .destroy = cfqio_ss_cgrp_destroy,
+ .populate = cfqio_ss_populate_dir,
+};
```
