The userspace solution for control groups

Linux Kongress 2010

Dhaval Giani
dhaval.giani@gmail.com

RETIS Lab, Scuola Superiore Sant'Anna

September 2010

Control Groups

What are cgroups?

Control Groups

What are cgroups?
Well, the last talk should have covered it

Control Groups

cgroups should die.

-Peter Zijlstra

Control Groups

Next time something is added to the kernel please mark it as "Hey, please don't use it, this is only here so that you don't use it. Thanks!" Maybe then dumb-ass folks like me will notice and refrain from using it.

-Lennart Poettering

libcgroup

Looks to provide a programming interface without the programmer having to care about how cgroups are setup.

libcgroup

Looks to provide a programming interface without the programmer having to care about how cgroups are setup.

- Tools

libcgroup

Looks to provide a programming interface without the programmer having to care about how cgroups are setup.

- Tools
- Library

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy **all** control group hierarchies

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy **all** control group hierarchies unless you use Fedora where it does not touch systemd hierarchies

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy **all** control group hierarchies unless you use Fedora where it does not touch systemd hierarchies
- cgexec - Used to start a process in a cgroup

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy **all** control group hierarchies unless you use Fedora where it does not touch systemd hierarchies
- cgexec - Used to start a process in a cgroup
- cgred - Automatic classification daemon originally based on user classfication.

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy **all** control group hierarchies unless you use Fedora where it does not touch systemd hierarchies
- cgexec - Used to start a process in a cgroup
- cgred - Automatic classification daemon originally based on user classfication. Now enhanced for process based classification as well.

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy **all** control group hierarchies unless you use Fedora where it does not touch systemd hierarchies
- cgexec - Used to start a process in a cgroup
- cgred - Automatic classification daemon originally based on user classfication. Now enhanced for process based classification as well.
- cgset/cgget - List cgroup values

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy **all** control group hierarchies unless you use Fedora where it does not touch systemd hierarchies
- cgexec - Used to start a process in a cgroup
- cgred - Automatic classification daemon originally based on user classfication. Now enhanced for process based classification as well.
- cgset/cgget - List cgroup values
- lscgroup - List all cgroups

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy **all** control group hierarchies unless you use Fedora where it does not touch systemd hierarchies
- cgexec - Used to start a process in a cgroup
- cgred - Automatic classification daemon originally based on user classfication. Now enhanced for process based classification as well.
- cgset/cgget - List cgroup values
- lscgroup - List all cgroups
- Some more

libcgroup - tools

- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy **all** control group hierarchies unless you use Fedora where it does not touch systemd hierarchies
- cgexec - Used to start a process in a cgroup
- cgred - Automatic classification daemon originally based on user classfication. Now enhanced for process based classification as well.
- cgset/cgget - List cgroup values
- lscgroup - List all cgroups
- Some more
- cgsnapshot - Under review right now, to generate configurations from current setup.

libcgroup - tools

Basically trying to cover a good set of requirements from the administrator's point of view.

libcgroup - library

Three main types of API

libcgroup - library

Three main types of API

- cgroup manipulation API

libcgroup - library

Three main types of API

- cgroup manipulation API
- data structure manipulation API

libcgroup - library

Three main types of API

- cgroup manipulation API
- data structure manipulation API
- configuration API

libcgroup API

The manipulation API

libcgroup API

The manipulation API

- cgroup_init

libcgroup API

The manipulation API

- cgroup_init
- cgroup_create_cgroup

libcgroup API

The manipulation API

- cgroup_init
- cgroup_create_cgroup
- cgroup_modify_cgroup

libcgroup API

The manipulation API

- cgroup_init
- cgroup_create_cgroup
- cgroup_modify_cgroup
- cgroup_delete_cgroup

libcgroup API

The manipulation API

- cgroup_init
- cgroup_create_cgroup
- cgroup_modify_cgroup
- cgroup_delete_cgroup
- cgroup_get_cgroup

libcgroup API

The manipulation API

- cgroup_init
- cgroup_create_cgroup
- cgroup_modify_cgroup
- cgroup_delete_cgroup
- cgroup_get_cgroup
- some more

libcgroup API

Used to modify the cgroup data structure

libcgroup API

Used to modify the cgroup data structure which is the main data structure describing the cgroup

libcgroup API

Used to modify the cgroup data structure which is the main data structure describing the cgroup

- cgroup_new_cgroup

libcgroup API

Used to modify the cgroup data structure which is the main data structure describing the cgroup

- cgroup_new_cgroup
- cgroup_add_controller

libcgroup API

Used to modify the cgroup data structure which is the main data structure describing the cgroup

- cgroup_new_cgroup
- cgroup_add_controller
- cgroup_add_value family

libcgroup API

Used to modify the cgroup data structure which is the main data structure describing the cgroup

- cgroup_new_cgroup
- cgroup_add_controller
- cgroup_add_value family
- cgroup_[sg]et_uid_gid

libcgroup API

Used by cgconfigparser and cgclear.

libcgroup API

Used by cgconfigparser and cgclear.
Used to load the configuration file, both for the superuser and the regular user (the so called cascaded configurations).

libcgroup API

Used by cgconfigparser and cgclear.
Used to load the configuration file, both for the superuser and the regular user (the so called cascaded configurations).
Not yet implemented for cgclear, but its on its way.

Issues

Written by kernel developers!

Issues

Written by kernel developers!
So thread safe

Issues

Written by kernel developers!
So thread safe as opposed to thread aware!

Issues

Written by kernel developers!
So thread safe as opposed to thread aware!
The locking model is being modified at
`git://libcg.git.sf.net/gitroot/libcg/libcgroup-context.git`

Issues

Key Goal - Subsystem Independent Programming.

Issues

Key Goal - Subsystem Independent Programming.

Or

Programmer should not care how subsystems are mounted.

Subsystem Independent Programming

- Programmer does not need to bother where cgroups are mounted

Subsystem Independent Programming

- Programmer does not need to bother where cgroups are mounted
- **But** the how still has to be bothered about

Subsystem Independent Programming

- Programmer does not need to bother where cgroups are mounted
- **But** the how still has to be bothered about

Blame the cgroup developers for that

Subsystem Independent Programming

Consider the following two scenarios

Subsystem Independent Programming

Consider the following two scenarios

- mount -t cgroup -o memory,cpuset cgroup /cgroup

Subsystem Independent Programming

Consider the following two scenarios

- mount -t cgroup -o memory,cpuset cgroup /cgroup

and

Subsystem Independent Programming

Consider the following two scenarios

- mount -t cgroup -o memory,cpuset cgroup /cgroup

and

- mount -t cgroup -o cpuset cpuset /cgroup/cpuset
- mount -t cgroup -o memory memory /cgroup/memory

Subsystem Independent Programming

Consider the following two scenarios

- mount -t cgroup -o memory,cpuset cgroup /cgroup

and

- mount -t cgroup -o cpuset cpuset /cgroup/cpuset
- mount -t cgroup -o memory memory /cgroup/memory

What is the main difference between the two from a programmer's point of view?

Subsystem Independent Programming

Simple solution

Subsystem Independent Programming

Simple solution
Copy the values from the parent

Subsystem Independent Programming

Consider another scenario

Subsystem Independent Programming

Consider another scenario

- mount -t cgroup -o cpu,cpuset cgroup /cgroup

Subsystem Independent Programming

Consider another scenario

- mount -t cgroup -o cpu,cpuset cgroup /cgroup

Well, the simple solution does fail quite spectacularly

Subsystem Independent Programming

We do have another problem

Subsystem Independent Programming

We do have another problem
Programmer still needs to know the internals of the subsystem he wants to manipulate

Subsystem Independent Programming

Layer 2

Subsystem Independent Programming

Layer 2

- Subsystem specific programming

Subsystem Independent Programming

Layer 2

- Subsystem specific programming
- Possibility of callbacks

Subsystem Independent Programming

Layer 2

- Subsystem specific programming
- Possibility of callbacks
- Hiding some of the subsystem complexity

Subsystem Independent Programming

But is that the complete solution?

Subsystem Independent Programming

But is that the complete solution?

Not really

Subsystem Independent Programming

But is that the complete solution?

Not really

We need to modify both the kernel and the userspace!

libcgroup

Not a bad start, but still a long way to go!

Questions?

Thank you!