The userspace solution for control groups

Linux Kongress 2010

Dhaval Giani
dhaval.giani@gmail.com

RETIS Lab, Scuola Superiore Sant’Anna

September 2010
What are cgroups?
What are cgroups?
Well, the last talk should have covered it
cgroups should die.

-Peter Zijlstra
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
Looks to provide a programming interface without the programmer having to care about how cgroups are setup.
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
- **cgconfigparser** - Used for parsing a configuration file and maintaining persistence across reboots.
- cgconfigparser - Used for parsing a configuration file and maintaining persistence across reboots.
- cgclear - Used to destroy all control group hierarchies
- 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
- **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 classification.
- **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 classification. 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 classification. Now enhanced for process based classification as well.
- cgset/cgget - List cgroup values.
- `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 classification. Now enhanced for process based classification as well.
- `cgset/cgget` - List cgroup values.
- `lscgroup` - List all cgroups.

Dhaval Giani

The userspace solution for control groups.
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 classification. Now enhanced for process based classification as well.
- cgset/cgget - List cgroup values
- lscgroup - List all cgroups
- Some more
- `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 classification. Now enhanced for process-based classification as well.
- `cgset/cgget` - List cgroup values.
- `lscggroup` - List all cgroups.
- Some more
- `cgsnapshot` - Under review right now, to generate configurations from current setup.

---

Dhaval Giani

The userspace solution for control groups
Basically trying to cover a good set of requirements from the administrator’s point of view.
Three main types of API
Three main types of API

- cgroup manipulation API
libcgroup - library

Three main types of API

- cggroup manipulation API
- data structure manipulation API
Three main types of API

- cgroup manipulation API
- data structure manipulation API
- configuration API
The manipulation API
libcggroup API

The manipulation API

- cgroup_init
The manipulation API

- cgroup_init
- cgroup_create_cgroup
The manipulation API

- cgroup_init
- cgroup_create_cgroup
- cgroup_modify_cgroup
libcgroupl API

The manipulation API

- cgroup_init
- cgroup_create_cgroup
- cgroup_modify_cgroup
- cgroup_delete_cgroup
The manipulation API

- cgroup_init
- cgroup_create_cgroup
- cgroup_modify_cgroup
- cgroup_delete_cgroup
- cgroup_get_cgroup
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
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
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).
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!

Dhaval Giani
The userspace solution for control groups
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
Key Goal - Subsystem Independent Programming.
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

Blame the cgroup developers for that
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
Consider the following two scenarios

mount -t cgroup -o memory,cpuset cgroup /cgroup
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?
Consider the following two scenarios

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

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

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

and

- `mount -t cgroup -o cpuset cpuset /cgroup/cpuset`

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

Dhaval Giani
The userspace solution for control groups
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?
Simple solution
Simple solution
Copy the values from the parent
Consider another scenario
Consider another scenario

- mount -t cgroup -o cpu,cpuset cgroup /cgroup
Consider another scenario

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

Well, the simple solution does fail quite spectacularly
We do have another problem
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
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
But is that the complete solution?
But is that the complete solution?

Not really
But is that the complete solution?

Not really

We need to modify both the kernel and the userspace!
Not a bad start, but still a long way to go!
Questions?
Thank you!