Subject: [RFC][PATCH 0/5] Container Freezer: Reuse Suspend Freezer
Posted by Matt Helsley on Thu, 24 Apr 2008 06:47:56 GMT
View Forum Message <> Reply to Message

This patchset reuses the container infrastructure and the swsusp freezer to freeze a group of tasks. I've merely taken Cedric's patches, forward-ported them to 2.6.25-mm1 and tested the expected common cases.

Changes since v1:
v2 (roughly patches 3 and 5):
 Moved the "kill" file into a separate cgroup subsystem (signal) and
  it's own patch.
 Changed the name of the file from freezer.freeze to freezer.state.
 Switched from taking 1 and 0 as input to the strings "FROZEN" and
  "RUNNING", respectively. This helps keep the interface
  human-usable if/when we need to more states.
 Checked that stopped or interrupted is "frozen enough"
  Since try_to_freeze() is called upon wakeup of these tasks
  this should be fine. This idea comes from recent changes to
  the freezer.
 Checked that if (task == current) whilst freezing cgroup we're ok
 Fixed bug where -EBUSY would always be returned when freezing
 Added code to handle userspace retries for any remaining -EBUSY

The freezer subsystem in the container filesystem defines a file named freezer.state. Writing "FROZEN" to the state file will freeze all tasks in the cgroup. Subsequently writing "RUNNING" will unfreeze the tasks in the cgroup. Reading will return the current state.

* Examples of usage :

```
   # mkdir /containers/freezer
   # mount -t cgroup -ofreezer,signal freezer  /containers/freezer
   # mkdir /containers/freezer/0
   # echo $some_pid > /containers/freezer/0/tasks
```

to get status of the freezer subsystem :

```
   # cat /containers/freezer/0/freezer.state
   RUNNING
```

to freeze all tasks in the container :

```
   # echo FROZEN > /containers/freezer/0/freezer.state
   # cat /containers/freezer/0/freezer.state
   FREEZING
   # cat /containers/freezer/0/freezer.state
   FROZEN
```

to unfreeze all tasks in the container :

```
   # echo RUNNING > /containers/freezer/0/freezer.state
   # cat /containers/freezer/0/freezer.state
   RUNNING
```

to kill all tasks in the container :

```
   # echo 9 > /containers/freezer/0/signal.kill
```

* Caveats:

  - The cgroup moves into the FROZEN state once all tasks in the cgroup are frozen. This is calculated and changed when the container file "freezer.state" is read or written.
  - Frozen containers will be unfrozen when a system is resumed after a suspend. This is addressed by a subsequent patch.

* Series

Applies to 2.6.25-mm1

The first patches make the freezer available to all architectures before implementing the freezer cgroup subsystem.

[RFC PATCH 1/5] Add TIF_FREEZE flag to all architectures
[RFC PATCH 2/5] Make refrigerator always available
[RFC PATCH 3/5] Implement freezer cgroup subsystem
[RFC PATCH 4/5] Skip frozen cgroups during power management resume
[RFC PATCH 5/5] Implement signal cgroup subsytem

Comments are welcome. I'm planning to finish up testing with ptrace'd and vforking processes and then, if it still seems appropriate, resubmit as a non-RFC series next.

Cheers,
 -Matt Helsley

--

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