

Brave new world of unified cgroups

Zbigniew Jędrzejewski-Szmek



zbyszek@in.waw.pl



Ustroń, 2019.11.30

Why this talk?

Why this talk?

Unified hierarchy (a.k.a.) cgroups v2 is the default default in systemd 243 (Sept. 2019)

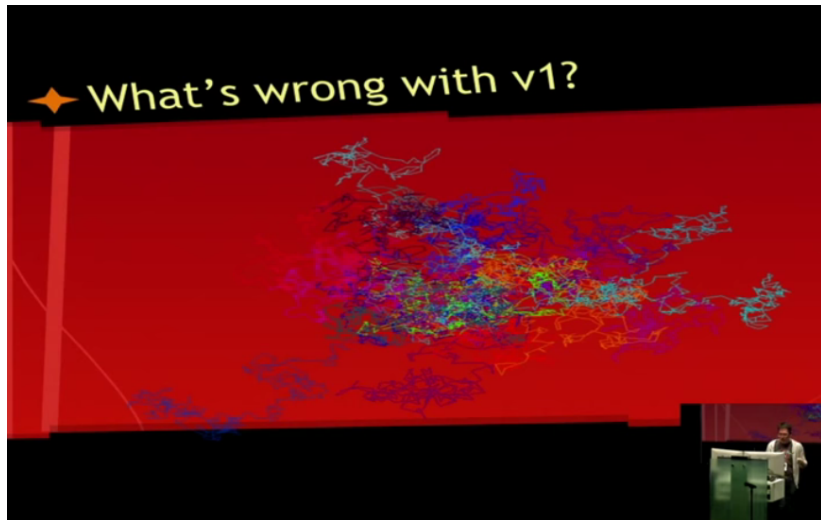
<https://fedoraproject.org/wiki/Changes/CGroupsV2> (for F31, now)

Brief history of cgroups in the linux kernel

- ▶ cgroups v1 — 2.6.24 (2008)
- ▶ v2 announced — 2012
- ▶ available — 2013
- ▶ stable — 4.5 (2016)
- ▶ threaded mode — 4.14
- ▶ CPU controller — 4.15 (2017)
- ▶ CPUSSET controller — 5.0 (2019)
- ▶ freezer — 5.1 (2019)

Why?

What's wrong with v1



What's wrong with v1

- ▶ design follows implementation

What's wrong with v1

- ▶ design follows implementation
- ▶ inconsistent interface

What's wrong with v1

- ▶ design follows implementation
- ▶ inconsistent interface
- ▶ infinite number of hierarchies

What's wrong with v1

- ▶ design follows implementation
- ▶ inconsistent interface
- ▶ infinite number of hierarchies
- ▶ no hierarchy

What's wrong with v1

- ▶ design follows implementation
- ▶ inconsistent interface
- ▶ infinite number of hierarchies
- ▶ no hierarchy
- ▶ unusable limits

What's wrong with v1

- ▶ design follows implementation
- ▶ inconsistent interface
- ▶ infinite number of hierarchies
- ▶ no hierarchy
- ▶ unusable limits
- ▶ threads not processes

What's wrong with v1

- ▶ design follows implementation
- ▶ inconsistent interface
- ▶ infinite number of hierarchies
- ▶ no hierarchy
- ▶ unusable limits
- ▶ threads not processes
- ▶ no secure delegation

```
$ ls /sys/fs/cgroup/memory/
```

```
cgroup.clone_children
```

```
cgroup.event_control
```

```
cgroup.procs
```

```
cgroup.sane_behavior
```

```
memory.failcnt
```

```
memory.force_empty
```

```
memory.limit_in_bytes
```

```
memory.max_usage_in_bytes
```

```
memory.soft_limit_in_bytes
```

```
memory.usage_in_bytes
```

```
memory.use_hierarchy
```

```
memory.memsw.failcnt
```

```
memory.memsw.limit_in_bytes
```

```
memory.memsw.max_usage_in_bytes
```

```
memory.memsw.usage_in_bytes
```

```
memory.move_charge_at_immigrate
```

```
tasks
```

```
notify_on_release
```

```
memory.kmem.failcnt
```

```
memory.kmem.limit_in_bytes
```

```
memory.kmem.max_usage_in_bytes
```

```
memory.kmem.slabinfo
```

```
memory.kmem.tcp.failcnt
```

```
memory.kmem.tcp.limit_in_bytes
```

```
memory.kmem.tcp.max_usage_in_bytes
```

```
memory.kmem.tcp.usage_in_bytes
```

```
memory.kmem.usage_in_bytes
```

```
memory.numa_stat
```

```
memory.oom_control
```

```
memory.pressure_level
```

```
memory.stat
```

```
memory.swappiness
```

```
release_agent
```

```
system.slice/
```

```
init.scope/
```

```
user.slice/
```

inconsistent interface

	v1	v2
CPUShares	1024	100
BlockIOWeight	500	100

no hierarchy

every controller implements custom rules
nested cgroups with flat semantics

unusable limits

`memory.limit_in_bytes` — when exceeded, SIGKILL or
userspace oom handling

unusable limits

`memory.limit_in_bytes` — when exceeded, SIGKILL or userspace oom handling

`memory.max_usage_in_bytes`,
`memory.memsw.max_usage_in_bytes`
 $\text{swap} + \text{memory} \geq \text{memory}$

unusable limits

`memory.limit_in_bytes` — when exceeded, SIGKILL or userspace oom handling

`memory.max_usage_in_bytes`,
`memory.memsw.max_usage_in_bytes`
 $\text{swap} + \text{memory} \geq \text{memory}$

freezer v1 implementation: repurposed suspend&resume code
SIGKILL doesn't work
from userspace *or* kernelspace
→ deadlocks

threads not processes

Unnecessary complexity

Doesn't make sense for most controllers

no secure delegation

not hierarchical

no secure delegation

not hierarchical
release_agent

no secure delegation

not hierarchical

release_agent

rumors of security issues

Design:

- ▶ single hierarchy
- ▶ controllers are fully hierarchical
- ▶ consistent interface
- ▶ small number of controllers: memory, io, pids, cpu, cpuset
- ▶ (controllers can be turned midway through the tree)
- ▶ high-level knobs
- ▶ soft limits

Design, ctd.:

- ▶ processes not threads
- ▶ no processes in inner nodes
- ▶ ~~single writer~~

Status quo

v1 only: k8s, CRI-O, Docker, Containerd, runc (in progress),
OCI runtime spec

v2 too: Buildah+Podman+skopeo, crun, libvirt, JVM, snapd,
systemd

Old vs. New

v1 controller	v2 solution
memory	memory
cpu+cpuacct	cpu
cpuset	cpuset
blkio	io
pids	pids

Old vs. New

v1 controller	v2 solution
memory	memory
cpu+cpuacct	cpu
cpuset	cpuset
blkio	io
pids	pids
freezer	replaced by cgroup.freeze

Old vs. New

v1 controller	v2 solution
memory	memory
cpu+cpuacct	cpu
cpuset	cpuset
blkio	io
pids	pids
freezer	replaced by cgroup.freeze
devices	eBPF filters

Old vs. New

v1 controller	v2 solution
memory	memory
cpu+cpuacct	cpu
cpuset	cpuset
blkio	io
pids	pids
freezer	replaced by cgroup.freeze
devices	eBPF filters
net_cls,net_prio	
perf_event	matching by cgroup
systemd	

Old vs. New

v1 controller	v2 solution
memory	memory
cpu+cpuacct	cpu
cpuset	cpuset
blkio	io
pids	pids
freezer	replaced by cgroup.freeze
devices	eBPF filters
net_cls,net_prio	matching by cgroup
perf_event	
systemd	
hugetlb	—

What is happening now?

CPUSET in kernel 5.0, freezer in kernel 5.1

What is happening now?

CPUSET in kernel 5.0, freezer in kernel 5.1
(systemd supports both and translates v2 settings for v1)

What is happening now?

CPUSET in kernel 5.0, freezer in kernel 5.1
(systemd supports both and translates v2 settings for v1)
<https://fedoraproject.org/wiki/Changes/CGroupsV2>

What is happening now?

CPUSET in kernel 5.0, freezer in kernel 5.1
(systemd supports both and translates v2 settings for v1)
<https://fedoraproject.org/wiki/Changes/CGroupsV2>
JVM, libvirt, snapd patched

What is happening now?

CPUSET in kernel 5.0, freezer in kernel 5.1

(systemd supports both and translates v2 settings for v1)

<https://fedoraproject.org/wiki/Changes/CGroupsV2>

JVM, libvirt, snapd patched

crun works

docker/mobi and most container runtimes — nope

What is happening now?

CPUSET in kernel 5.0, freezer in kernel 5.1

(systemd supports both and translates v2 settings for v1)

<https://fedoraproject.org/wiki/Changes/CGroupsV2>

JVM, libvirt, snapd patched

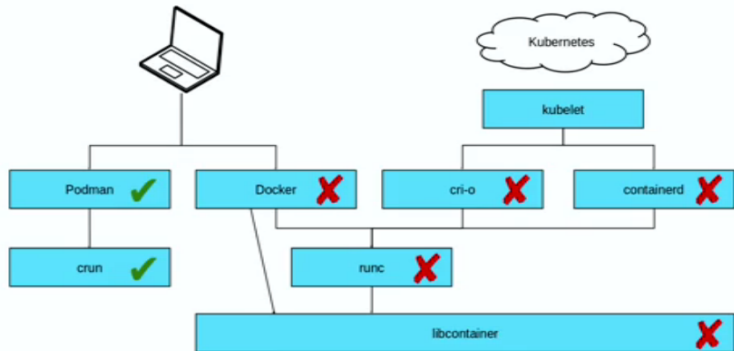
crun works

docker/mobi and most container runtimes — nope

“Red Hat suite”: podman, buildah, skopeo — works

Status of container runtimes

Which components support unified hierarchy?



Delegation

- ▶ less-privileged process own a part of the cgroup tree
- ▶ implemented through file system permissions
- ▶ Delegate=yes in systemd units

```
$ sudo systemd-cgls
```

```
Control group /:  
- .slice  
├─ user.slice  
│   ├─ user-6.slice  
│   │   └─ user@6.service ...  
│   │       └─ init.scope  
│   │           ├─ 1963 /usr/lib/systemd/systemd --user  
│   │           └─ 2001 (sd-pam)  
│   └─ user-1000.slice  
│       └─ user@1000.service ...  
│           ├─ gsd-xsettings.service  
│           │   └─ 412129 /usr/libexec/gsd-xsettings  
│           ├─ gvfs-goa-volume-monitor.service  
│           │   └─ 412027 /usr/libexec/gvfs-goa-volume-monitor  
│           ├─ gsd-power.service  
│           │   └─ 412104 /usr/libexec/gsd-power  
│           ├─ dbus\x2d:1.1\x2dorg.gnome.Epiphany.SearchProvider.slice  
│           │   └─ dbus-:1.1-org.gnome.Epiphany.SearchProvider@0.service  
│           │       └─ 415659 /usr/libexec/epiphany-search-provider  
│           ├─ xdg-permission-store.service  
│           │   └─ 411997 /usr/libexec/xdg-permission-store  
│           ├─ dbus-broker.service  
│           │   └─ 411532 /usr/bin/dbus-broker-launch --scope user  
│           │       └─ 411533 dbus-broker --log 4 --controller 11 --machine-id 08a5690a2eed47cf92ac0a  
│           ├─ xdg-document-portal.service  
│           │   └─ 412300 /usr/libexec/xdg-document-portal  
│           ├─ dbus\x2d:1.1\x2dorg.gnome.OnlineAccounts.slice  
│           │   └─ dbus-:1.1-org.gnome.OnlineAccounts@0.service  
│           │       └─ 412024 /usr/libexec/goa-daemon  
│           └─ tracker-store.service  
│               └─ 468488 /usr/libexec/tracker-store
```



```
$ ls -l .../user.slice/user-1000.slice/user@1000.service
-r--r--r--. root      root      cgroup.controllers
-r--r--r--. root      root      cgroup.events
-rw-r--r--. root      root      cgroup.freeze
-rw-r--r--. root      root      cgroup.max.depth
-rw-r--r--. root      root      cgroup.max.descendants
-r--r--r--. root      root      cgroup.stat
-rw-r--r--. zbyszek  zbyszek  cgroup.procs
-rw-r--r--. zbyszek  zbyszek  cgroup.threads
-rw-r--r--. zbyszek  zbyszek  cgroup.subtree_control
-r--r--r--. root      root      pids.current
-r--r--r--. root      root      pids.events
-rw-r--r--. root      root      pids.max
...
drwxr-xr-x. zbyszek  zbyszek  pipewire.service/
drwxr-xr-x. zbyszek  zbyszek  pulseaudio.service/
drwxr-xr-x. zbyszek  zbyszek  xdg-desktop-portal-gtk.service/
drwxr-xr-x. zbyszek  zbyszek  xdg-desktop-portal.service/
drwxr-xr-x. zbyszek  zbyszek  xdg-document-portal.service/
```

Delegation

- ▶ Delegate=io pids memory ...
- ▶ delegation may be nested
- ▶ resources are divided hierarchically

Threaded mode

Threaded mode

v1 operated on threads, v2 operates on processes

Threaded mode

v1 operated on threads, v2 operates on processes

only supported by selected controllers (cpu, cpuset)

Threaded mode

v1 operated on threads, v2 operates on processes

only supported by selected controllers (cpu, cpuset)

a leaf cgroup may be switched to threaded mode and subdivided

```
$ echo threaded > $CGROUP/cgroup.type
```

Threaded mode

v1 operated on threads, v2 operates on processes

only supported by selected controllers (cpu, cpuset)

a leaf cgroup may be switched to threaded mode and subdivided

```
$ echo threaded > $CGROUP/cgroup.type
```

cgroup.threads

Threaded mode

v1 operated on threads, v2 operates on processes

only supported by selected controllers (cpu, cpuset)

a leaf cgroup may be switched to threaded mode and subdivided

```
$ echo threaded > $CGROUP/cgroup.type
```

cgroup.threads

used by libvirt

Summary

- ▶ fully hierarchical with safe delegation
- ▶ consistent interface
- ▶ efficient and scalable notifications
- ▶ fewer controllers, high-level knobs
- ▶ soft limits
- ▶ eBPF!
- ▶ better monitoring: PSI!

Links

<https://www.kernel.org/doc/html/latest/admin-guide/cgroup-v2.html>
<https://facebookmicrosites.github.io/cgroup2/docs/overview>
[systemd.resource-control\(5\)](https://systemd.io/CGROUP_DELEGATION.html)
https://systemd.io/CGROUP_DELEGATION.html
recent changes:
<https://www.redhat.com/sysadmin/fedora-31-control-group-v2>
<https://fedoraproject.org/wiki/Changes/CGroupsV2>
<https://www.youtube.com/watch?v=GznkuTXq8AQ&t=1s>
<https://medium.com/nttlabs/cgroup-v2-596d035be4d7>
<https://www.youtube.com/watch?v=yZpNsDe4Qzg> (Michael Kerrisk)
freezer for cgroup v2 v5.1-rc3-45-g76f969e894
<https://lwn.net/Articles/772377/>
https://bugzilla.redhat.com/show_bug.cgi?id=1727149 libvirt support in 5.5.0
https://bugzilla.redhat.com/show_bug.cgi?id=1438079 snapd support in snapd-2.41-1.fc31
<https://github.com/opencontainers/runc/pull/2113> for libcontainer
<https://github.com/opencontainers/runc/issues/654> for runc
<https://github.com/kubernetes/enhancements/pull/1370/files> for k8s
<https://github.com/systemd/systemd/issues/12460> systemd dbus
codesearch.debian.net/search?q=cgroup.type
<https://www.kernel.org/doc/html/latest/accounting/psi.html>

history:

https://kernelnewbies.org/Linux_2_6_24#Task_Control_Groups

https://kernelnewbies.org/Linux_3.16#Unified_Control_Group_hierarchy State of CPU controller in
cgroup v2 (2016) LWN: A milestone for control groups (2017) https://kernelnewbies.org/Linux_4.15#Better_CPU_usage_restrictions_with_the_CPU_resource_controller_for_cgroupv2

[https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=](https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=0d5936344f30aba0f6ddb92b030cb6a05168efe6)

[0d5936344f30aba0f6ddb92b030cb6a05168efe6](https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=0d5936344f30aba0f6ddb92b030cb6a05168efe6) <https://www.youtube.com/watch?v=PzpG40WiEfM>

https://www.youtube.com/watch?v=ikZ8_mRotT4