Brave new world of unified cgroups

Michal Sekletár
Zbigniew Jędrzejewski-Szmek

Red Hat

msekleta@redhat.com
zbyszek@in.waw.pl

Brno, 2020.01.24

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+)

Version two is just nicer!

Control groups

Control groups (cgroups) is a Linux subsystem that has two main purposes

- Process tracking
- Resource distribution

Control groups

Cgroup – associate a set of tasks with a set of parameters for one or more controllers

Controller – schedules a resource or applie per-cgroup limits

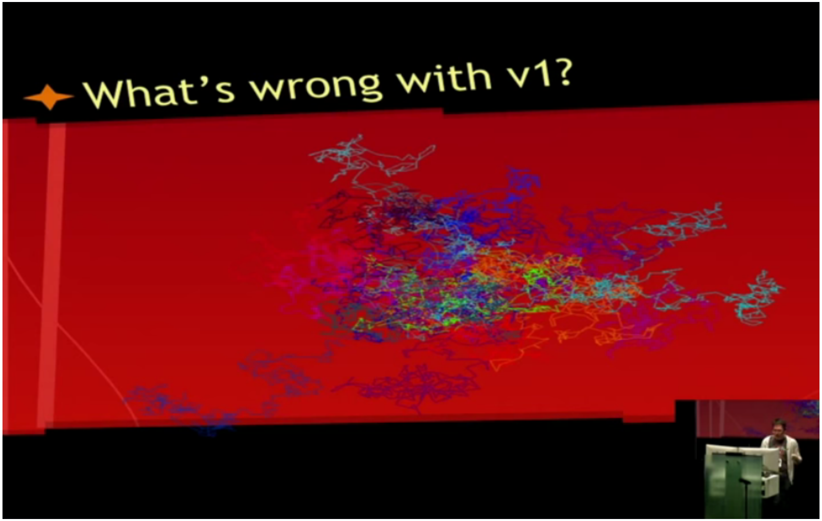
Hierarchy – set of cgroups arranged in a tree, every process is in exactly one cgroup

Brief history of cgroups in the linux kernel

	kernel	
2008	2.6.24	task control groups
2012		v2 rewrite announced
2013		v2 available
2015	4.3	pids controller
2016	4.5	v2 stable!
2017	4.14	threaded mode
	4.15	cpu controller
2019	5.0	cpuset controller
	5.1	freezer

Why?

What’s wrong with v1



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
- not hierarchical

What's wrong with v1

- design follows implementation
- inconsistent interface
- infinite number of hierarchies
- not hierarchical
- unusable limits

What's wrong with v1

- design follows implementation
- inconsistent interface
- infinite number of hierarchies
- not hierarchical
- unusable limits
- no cooperation between contollers

What’s wrong with v1

- design follows implementation
- inconsistent interface
- infinite number of hierarchies
- not hierarchical
- unusable limits
- no cooperation between contollers
- threads not processes

What’s wrong with v1

- design follows implementation
- inconsistent interface
- infinite number of hierarchies
- not hierarchical
- unusable limits
- no cooperation between contollers
- threads not processes
- no secure delegation

```
$ ls /sys/fs/cgroup/memory/
...
memory.limit_in_bytes       memory.kmem.tcp.limit_in_bytes
memory.usage_in_bytes       memory.kmem.tcp.usage_in_bytes
memory.max_usage_in_bytes   memory.kmem.tcp.max_usage_in_byte
memory.soft_limit_in_bytes

memory.kmem.limit_in_bytes     memory.memsw.limit_in_bytes
memory.kmem.usage_in_bytes     memory.memsw.usage_in_bytes
memory.kmem.max_usage_in_bytes memory.memsw.max_usage_in_byte

memory.kmem.slabinfo              memory.use_hierarchy
memory.move_charge_at_immigrate  cgroup.sane_behavior
...
```

inconsistent interface

v1	default	range
`cpu.shares`	1024	2-262144
`blkio.bfq.weight`	500	10–1000
v2		
`cpu.weight`	100	1–10000
`io.weight`[1]	100	1–10000

[1] https://github.com/systemd/systemd/pull/13335

v2

Design:

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

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

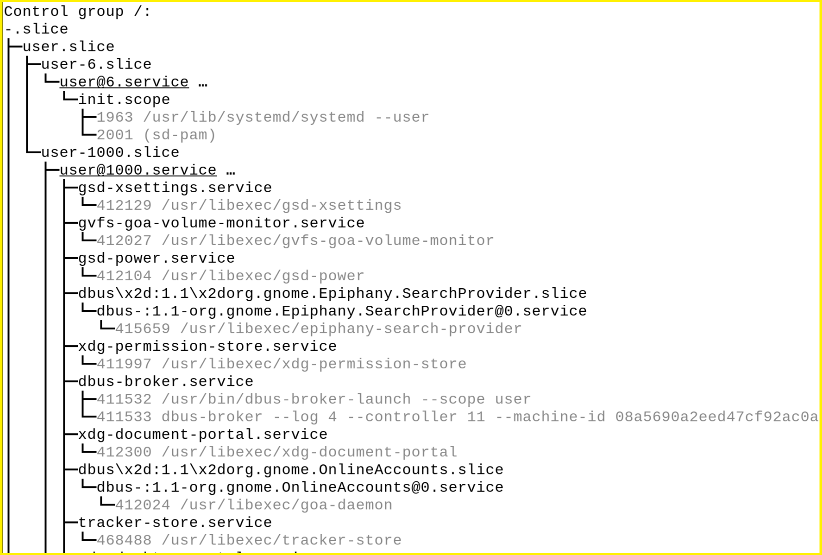
Old vs. New

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	matching by cgroup, eBPF
perf_event	eBPF
hugetlb	hugetlb (kernel 5.6)

Delegation

- less-privileged process owns a part of the cgroup tree
- implemented through file system permissions
- Delegate=yes in systemd units
- cutoff <u>not</u> at the directory level

$ 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

Resource management – Resource distribution models

- **Weights**
  - Resource is distributed by adding up the weights of all sub-cgroups and giving each the fraction matching its ratio against the sum.
  - Usually used to distribute stateless resources (CPU time)
  - Example: cpu.weight ([1-10000], default 100)

Resource management – Resource distribution models

- **Weights**
  - Resource is distributed by adding up the weights of all sub-cgroups and giving each the fraction matching its ratio against the sum.
  - Usually used to distribute stateless resources (CPU time)
  - Example: cpu.weight ([1-10000], default 100)
- **Limits**
  - Cgroup can consume up to configured amount of the resource
  - Overcommit is allowed (i.e. sum of sub-cgroup limits can exceed limit of the parent cgroup)
  - Example: memory.max

Resource management – Resource distribution models

- **Weights**
  - Resource is distributed by adding up the weights of all sub-cgroups and giving each the fraction matching its ratio against the sum.
  - Usually used to distribute stateless resources (CPU time)
  - Example: cpu.weight ([1-10000], default 100)
- **Limits**
  - Cgroup can consume up to configured amount of the resource
  - Overcommit is allowed (i.e. sum of sub-cgroup limits can exceed limit of the parent cgroup)
  - Example: memory.max
- **Protections**
  - Cgroup is protected (but not guaranteed) upto configured amount of the resource
  - Overcommit is also allowed
  - Example: memory.low

Resource management – Resource distribution models

- **Weights**
  - Resource is distributed by adding up the weights of all sub-cgroups and giving each the fraction matching its ratio against the sum.
  - Usually used to distribute stateless resources (CPU time)
  - Example: cpu.weight ([1-10000], default 100)
- **Limits**
  - Cgroup can consume up to configured amount of the resource
  - Overcommit is allowed (i.e. sum of sub-cgroup limits can exceed limit of the parent cgroup)
  - Example: memory.max
- **Protections**
  - Cgroup is protected (but not guaranteed) upto configured amount of the resource
  - Overcommit is also allowed
  - Example: memory.low
- **Allocations**
  - Exclusive allocations of the absolute amount of a finite resource (e.g. real-time budget)

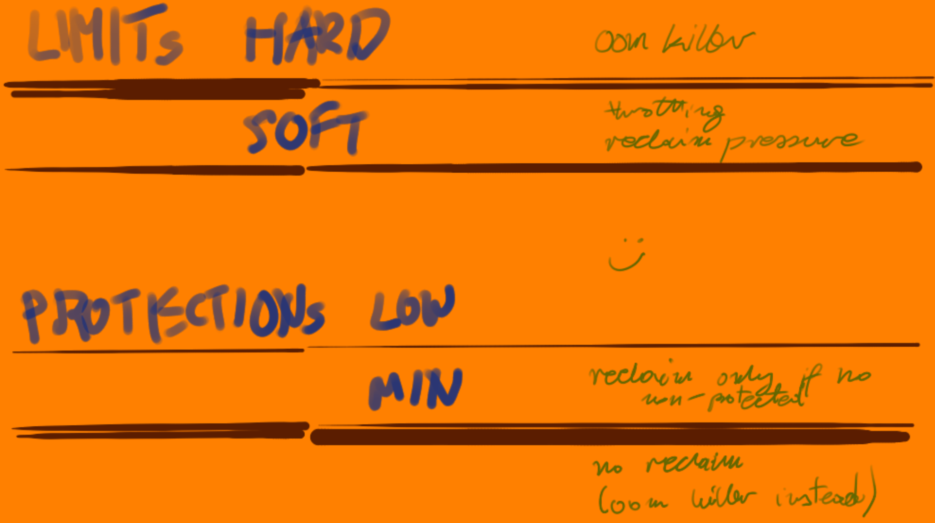
Resource management – Memory

Partitioning available memory with systemd and cgroup v2 memory controller is rather complicated. Multiple options are available,

- **memory.min** – Hard memory protection. If memory usage is below the limit the cg memory won't be reclaimed.
- **memory.low** – Soft memory protection. If memory usage is below the limit the cg memory can be reclaimed only if there is no memory to be reclaimed from unprotected cgroups.
- **memory.high** – Memory throttle limit. If memory usage goes above the limit the processes in the cgroup are throttled and put under heavy reclaim pressure.
- **memory.max** – Hard limit for memory usage. You can use K, M, G, T suffixes (e.g. MemoryMax=1G).

After you exhaust your memory limit then service is very likely to get killed by OOM killer. To prevent that you need to adjust `OOMScoreAdjust` value as well.

Memory protections and limits



Resource management – Block I/O

Block I/O controller in cgroup v2 allows for quite fine grained tuning. systemd provides following options for configuring this subsystem,

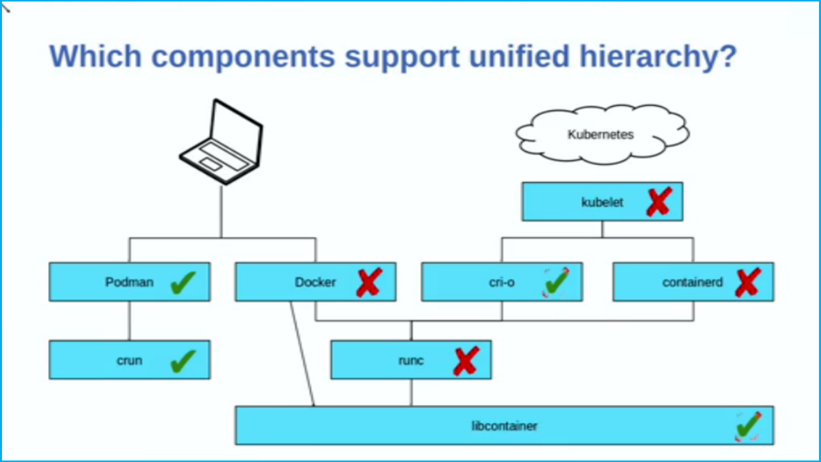
- **io.weight** – Set the default IO weight
- **io.max** – Absolute per device bandwidth. e.g. `8:16 rbps=2097152 wiops=120`
- **io.latency** – Define the per device I/O latency target (e.g. `8:16 target=10000`)

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

Status of container runtimes



Summary

control groups v2:

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

Links

this talk: `https://github.com/keszybz/cgroupsv2/raw/master/cgroupsv2.pdf`

docs:
`https://www.kernel.org/doc/html/latest/admin-guide/cgroup-v1/`
`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`

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

codesearch.debian.net/search?q=cgroup.type

`https://www.kernel.org/doc/html/latest/accounting/psi.html`

Links

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)
Linux 4.15
v4.14-rc2-7-g0d5936344f
https://www.youtube.com/watch?v=PzpG40WiEfM

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