zalando

K8S RESOURCE SHARING

FOR POSTGRESQL EXPERTS

DMITRY DOLGOV

03-09-2019

patroni & postgres-operator

```yaml
resources:
  requests:
    cpu: "1000m"
    memory: "1GB"
  limits:
    cpu: "2000m"
    memory: "2GB"
```

?

Allocatable

```
Capacity:
 cpu:                            16
 hugepages-1Gi:                  0
 hugepages-2Mi:                  0
 memory:                         65947396Ki
Allocatable:
 cpu:                            15800m
 hugepages-1Gi:                  0
 hugepages-2Mi:                  0
 memory:                         65388292Ki
```

Allocatable

```
Capacity:
 cpu:                          16
 hugepages-1Gi:                0
 hugepages-2Mi:                0
 memory:                       65947396Ki
Allocatable:
 cpu:                          15800m
 hugepages-1Gi:                0
 hugepages-2Mi:                0
 memory:                       65388292Ki
```

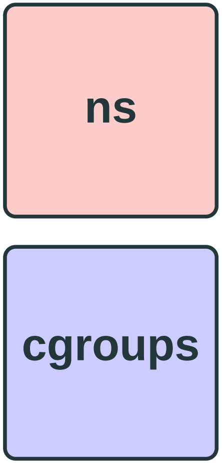
K8S appliance?

zalando

libcontainer/SPEC.md

cgroups

libcontainer/SPEC.md



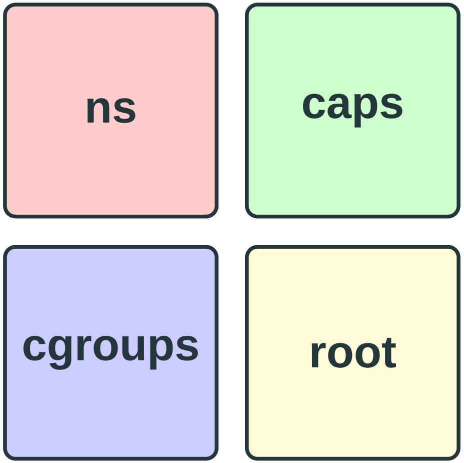
libcontainer/SPEC.md

ns

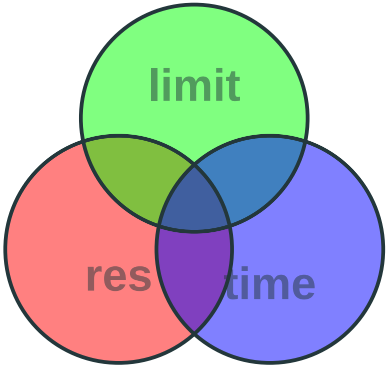
caps

cgroups

libcontainer/SPEC.md

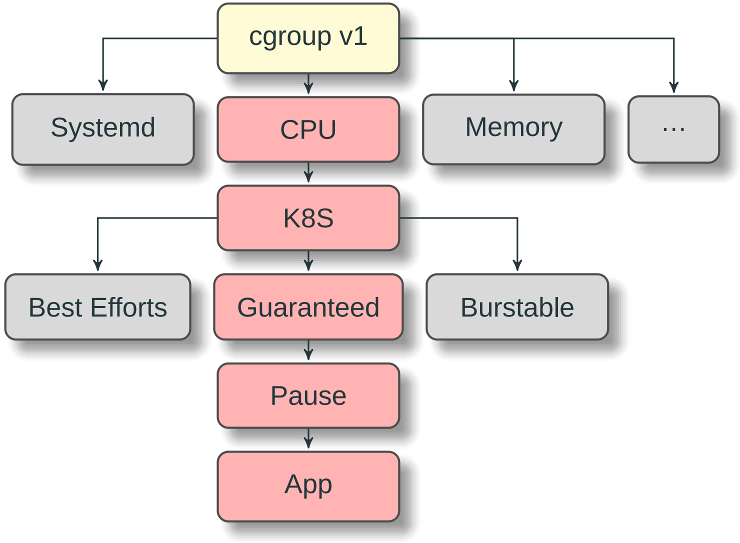


libcontainer/SPEC.md

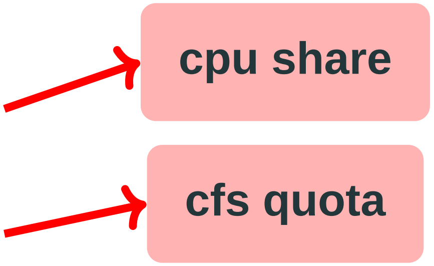

limit
res
time

- docker
- rkt
- LXC
- . . .

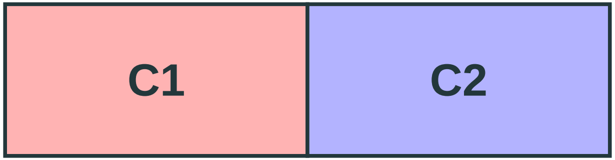
- podman
- cgroup v1
- cgroup v2
- . . .


cgroup v1
Systemd
CPU
Memory
...
K8S
Best Efforts
Guaranteed
Burstable
Pause
App

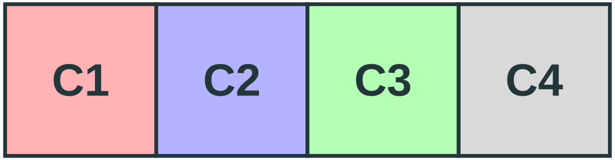
```yaml
resources:
  requests:
    cpu: "1000m"
    memory: "1GB"
  limits:
    cpu: "2000m"
    memory: "2GB"
```



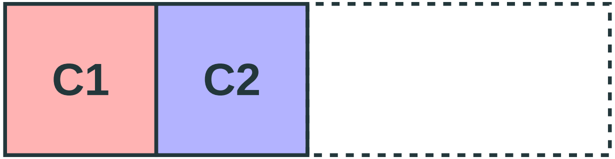
Share



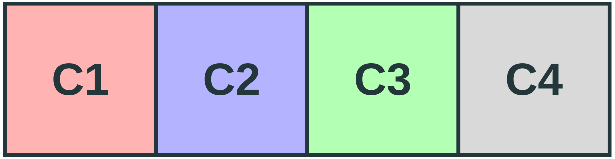
Share



Quota



Quota



Bandwidth accounting

```
# from /proc/sys/kernel/
sched_cfs_bandwidth_slice_us
# default=5ms
```

Exclusive CPU

- ➔ cpu management policy
- ➔ kube-reserved
- ➔ guaranteed (integer quantity)
- ➔ cpuset.cpus
- ➔ cpuset.cpuset.cpu_exclusive

```yaml
resources:
  requests:
    cpu: "1000m"
    memory: "1GB"
  limits:
    cpu: "2000m"
    memory: "2GB"
```

soft mem limit

hard mem limit

```yaml
resources:
  requests:
    cpu: "1000m"
    memory: "1GB"
  limits:
    cpu: "2000m"
    memory: "2GB"
```

~~soft mem limit~~

hard mem limit

Memory reclaim

```
# only under the memory pressure
=> page_reclaim.py --container 89c33bb3133f

[7382] postgres: 928K
[7138] postgres: 152K
[7136] postgres: 180K
[7468] postgres: 72M
[7464] postgres: 57M
[5451] postgres: 1M
```

Memory

- ➔ best efforts
- ➔ requests for QoS and oom_adj
- ➔ memory.kmem.limit_in_bytes
- ➔ MEMCG_CHARGE_BATCH (32 pages)
- ➔ shared memory CoW

```yaml
resources:
  requests:
    cpu: "1000m"
    memory: "1GB"
    hugepages-1Gi: "1GB"
    hugepages-2Mi: "1GB"
  limits:
    cpu: "2000m"
    memory: "2GB"
    hugepages-1Gi: "2GB"
    hugepages-2Mi: "2GB"
```

```yaml
resources:
  requests:
    cpu: "1000m"
    memory: "1GB"
    hugepages-1Gi: "1GB"
    hugepages-2Mi: "1GB"
  limits:
    cpu: "2000m"
    memory: "2GB"
    hugepages-1Gi: "2GB"
    hugepages-2Mi: "2GB"
```

➔ classic

```
resources:
  requests:
    cpu: "1000m"
    memory: "1GB"
    hugepages-1Gi: "1GB"
    hugepages-2Mi: "1GB"
  limits:
    cpu: "2000m"
    memory: "2GB"
    hugepages-1Gi: "2GB"
    hugepages-2Mi: "2GB"
```

➔ classic

➔ transparent

zalando

```yaml
resources:
  requests:
    cpu: "1000m"
    memory: "1GB"
    hugepages-1Gi: "1GB"
    hugepages-2Mi: "1GB"
  limits:
    cpu: "2000m"
    memory: "2GB"
    hugepages-1Gi: "2GB"
    hugepages-2Mi: "2GB"
```

→ classic

→ transparent

→ isolation only per pod

zalando

```yaml
resources:
  requests:
    cpu: "1000m"
    memory: "1GB"
    hugepages-1Gi: "1GB"
    hugepages-2Mi: "1GB"
  limits:
    cpu: "2000m"
    memory: "2GB"
    hugepages-1Gi: "2GB"
    hugepages-2Mi: "2GB"
```

- → classic
- → transparent
- → isolation only per pod
- → no soft limits or reclaim (SIGBUS)

Huge pages

```
# perf record -e dTLB-loads,dTLB-stores -p PID
# huge_pages on
Samples: 832K of event 'dTLB-load-misses'
Event count (approx.): 640614445 : ~19% less
Samples: 736K of event 'dTLB-store-misses'
Event count (approx.): 72447300 : ~29% less

# huge_pages off
Samples: 894K of event 'dTLB-load-misses'
Event count (approx.): 784439650
Samples: 822K of event 'dTLB-store-misses'
Event count (approx.): 101471557
```

Huge pages

```
# perf record -e dTLB-loads,dTLB-stores -p PID
# huge_pages on
Samples: 832K of event 'dTLB-load-misses'
Event count (approx.): 640614445 : ~19% less
Samples: 736K of event 'dTLB-store-misses'
Event count (approx.): 72447300 : ~29% less

# huge_pages off
Samples: 894K of event 'dTLB-load-misses'
Event count (approx.): 784439650
Samples: 822K of event 'dTLB-store-misses'
Event count (approx.): 101471557
```

LLCache

```
=> pqos -s
L3CA COS definitions for Socket 0:
        L3CA COS0 => MASK 0xffff
        L3CA COS1 => MASK 0xf
        L3CA COS2 => MASK 0xff0
        L3CA COS3 => MASK 0xfff
Core information for socket 0:
        Core 0, L2ID 0, L3ID 0 => COS0
        Core 1, L2ID 1, L3ID 0 => COS0
        Core 2, L2ID 0, L3ID 0 => COS0
        Core 3, L2ID 1, L3ID 0 => COS0
```

zalando

LLCache

```
# COS1 4 cache ways, COS 2 next 8 cache ways
=> pqos -e "llc:1=0x000f;llc:2=0x0ff0;"
SOCKET 0 L3CA COS1 => MASK 0xf
SOCKET 0 L3CA COS2 => MASK 0xff0
Allocation configuration altered.
```

Block I/O

The creation of the io.latency block I/O controller

Block I/O

➔ sharing a disk between users in Linux is awful

The creation of the io.latency block I/O controller

Block I/O

- ➔ sharing a disk between users in Linux is awful
- ➔ throttling puts pressure somewhere else

The creation of the io.latency block I/O controller

Block I/O

- ➔ sharing a disk between users in Linux is awful
- ➔ throttling puts pressure somewhere else
- ➔ lack of trivially observable cost metric

The creation of the io.latency block I/O controller

Writeback monitoring

```
=> perf record -e writeback:writeback_written

kworker/u8:1 reason=periodic   nr_pages=101429
kworker/u8:1 reason=background nr_pages=MAX_ULONG
kworker/u8:3 reason=periodic   nr_pages=101457
```

Writeback monitoring

```
# pgbench insert workload
=> io_timeouts.py bin/postgres

[18335] END: MAX_SCHEDULE_TIMEOUT
[18333] END: MAX_SCHEDULE_TIMEOUT
[18331] END: MAX_SCHEDULE_TIMEOUT
[18318] truncate pgbench_history: MAX_SCHEDULE_TIMEOUT
```

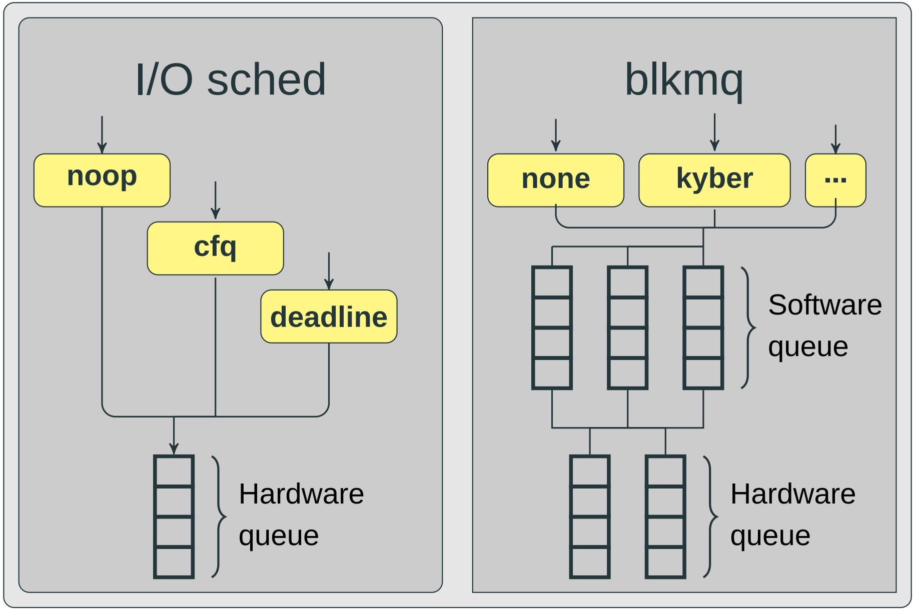
blkio cgroup v1

- ➔ Direct IO oriented
- ➔ no relationships between controllers
- ➔ writeback = memory + blkio controllers
- ➔ writeback accounted to the root cgroup

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git

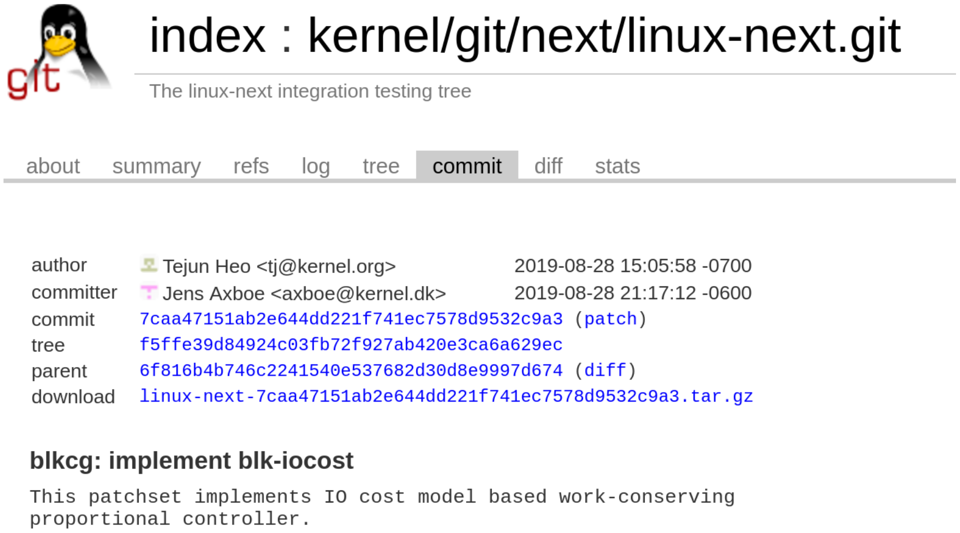
IO scheduler

```
=> cat /sys/block/xvdcj/queue/scheduler
[mq-deadline] kyber bfq none
```


I/O sched
noop
cfq
deadline
Hardware queue
blkmq
none
kyber
...
Software queue
Hardware queue

```
=> blk_mq.py --container 89c33bb3133f

 latency (us)          : count     distribution
    16 -> 31           : 0        |                                        |
    32 -> 63           : 19       |***                                     |
    64 -> 127          : 27       |****                                    |
   128 -> 255          : 6        |*                                       |
   256 -> 511          : 8        |*                                       |
   512 -> 1023         : 17       |***                                     |
  1024 -> 2047         : 40       |*******                                 |
  2048 -> 4095         : 126      |**********************                  |
  4096 -> 8191         : 144      |*************************               |
  8192 -> 16383        : 222      |****************************************|
 16384 -> 32767        : 120      |*********************                   |
 32768 -> 65535        : 44       |*******                                 |
```



index : kernel/git/next/linux-next.git

The linux-next integration testing tree

about summary refs log tree **commit** diff stats

author	Tejun Heo <tj@kernel.org>	2019-08-28 15:05:58 -0700
committer	Jens Axboe <axboe@kernel.dk>	2019-08-28 21:17:12 -0600
commit	7caa47151ab2e644dd221f741ec7578d9532c9a3 (patch)	
tree	f5ffe39d84924c03fb72f927ab420e3ca6a629ec	
parent	6f816b4b746c2241540e537682d30d8e9997d674 (diff)	
download	linux-next-7caa47151ab2e644dd221f741ec7578d9532c9a3.tar.gz	

blkcg: implement blk-iocost

```
This patchset implements IO cost model based work-conserving
proportional controller.
```

Network

- ➔ vlanx
- ➔ network class
- ➔ traffic control

Noise

- ➔ Futex
- ➔ Compaction
- ➔ Readahead
- ➔ Filesystem
- ➔ ...

Questions?

github.com/erthalion

@erthalion

dmitrii.dolgov at zalando dot de

9erthalion6 at gmail dot com