zalando

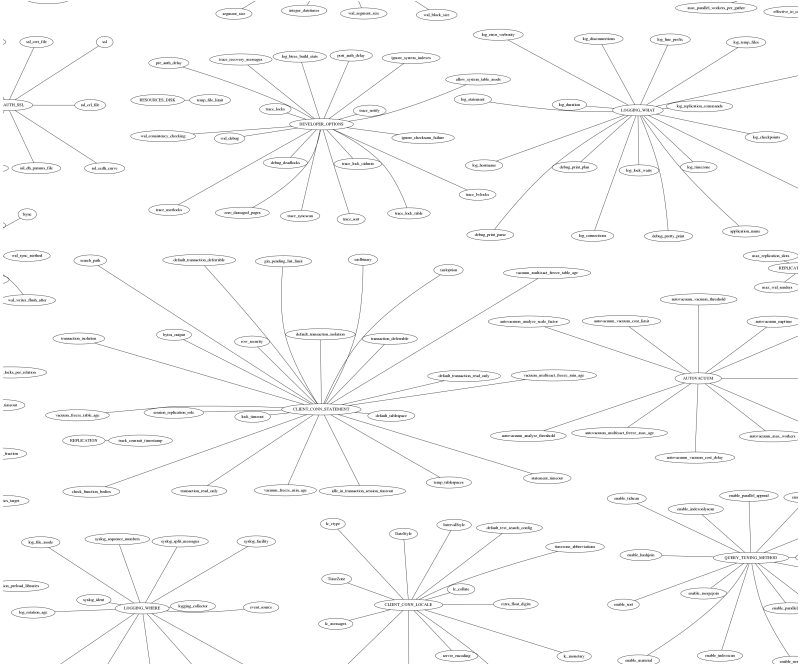
POSTGRESQL LINUX KERNEL

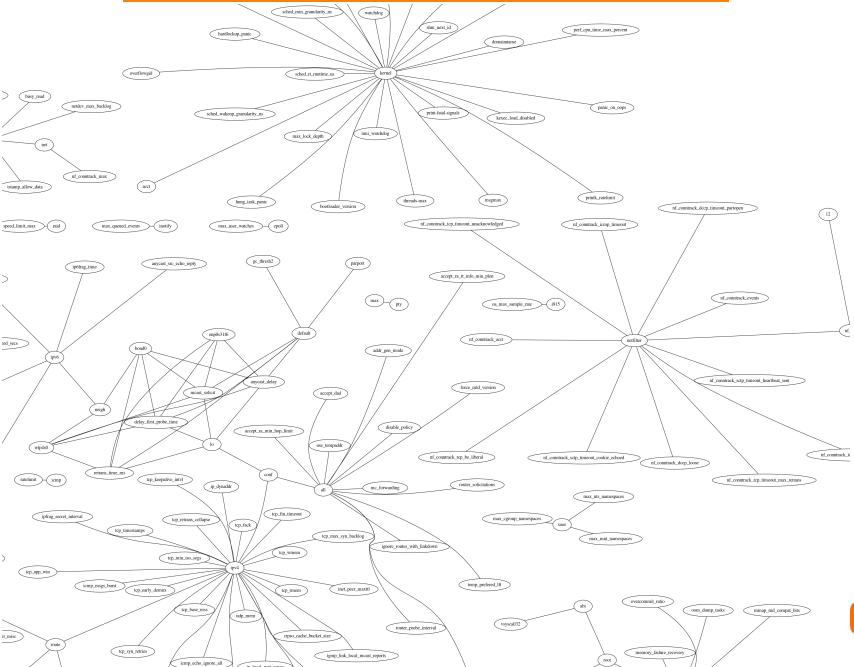
FRIENDSHIP

DMITRY DOLGOV

22-10-2018

zalando


wal_keep_segments
max_standby_streaming_delay
max_standby_archive_delay
max_replication_slots
wal_receiver_status_interval
hot_standby_feedback
wal_receiver_timeout
wal_receiver_timeout
max_wal_senders
max_replication_slots
wal_receiver_timeout
hot_standby
max_standby_streaming_delay
max_standby_archive_delay
hot_standby_feedback
max_wal_senders
max_replication_slots
wal_receiver_timeout
max_wal_senders
max_replication_slots


sched_cars_granularity_ns
watchdog
sbc_acct_id
perf_cpu_time_max_percent
hardlockup_panic
domainname
overflowgid
sched_rt_runtime_ns
kernel
busy_read
netdev_max_backlog
sched_wakeup_granularity_ns
panic_on_oops
panic_final_reports
kexec_load_disabled
max_lock_depth
nmi_watchdog
net
nf_conntrack_max
timerslack_allow_data
acct
hung_task_panic
threads-max
osrelease
panic_timeout
nf_conntrack_decp_timeout_partopen
12
speed_limit_max
null
max_queued_events
inotify
max_user_watches
epoll
nf_conntrack_tcp_timeout_unacknowledged
nf_conntrack_icmp_timeout
getting_time
ntp-cost_no_solve_mpls
jc_thresh2
purpot
accept_ra_n_info_min_plen
max
pty
ra_min_sample_rate
ipv15
nf_conntrack_events
default
netfilter
ipv6
icmp6
nf_conntrack_acct
nf_conntrack_tcp_timeout_heartbeat_sent
mpls/ilk
adk_pm_mode
mount_delay
mount_subcl
accept_chd
force_add_version
nf_conntrack_tcp_be_liberal
nf_conntrack_tcp_timeout_cookie_reload
nf_conntrack_decp_loose
nf_conntrack_tcp_timeout_max_retrans
mpls
delay_first_probe_time
accept_ra_min_hop_limit
double_policy
tc
mc_torqadd
sunrise_flow_ttc
sunrise
xrp
ip
ip_forwarding
router_solicitations
ip_forqadve_ttrt1
ip_dmadd
tcp_fin_timeout
tcp_fack
tcp_max_syn_backlog
tcp_timestamps
tcp_retrans_collapse
tcp_winscl
ignore_routes_with_linkdown
tcp_mtu_no_ssp
tcp
tcp_app_win
max_rt_namespaces
max_cgroup_namespaces
uts
max_mnt_namespaces
tcp_early_demux
tcp_timeo
tcp_max_orphans
tcp_base_rtos
udp_mem
tcp_rfc1337
tcp_echo_ignore_all
tcp_fastopen_cookie_size
tcp_fsk_local_mount_reports
tcp_probe_interval
sbi
rcvcomm_rate
core_dump_tasks
rmap_nul_coupel_bits
vpcsel32
memory_failure_recovery
vmx
zalonado
zalonado_ttl_ttl_ttl

Plan

Overview

zalando

Plan

Overview

Resources

- ➔ Execution scheduling
- ➔ Memory management
- ➔ Storage IO

zalando

Plan

Overview

Resources

- ➔ Execution scheduling
- ➔ Memory management
- ➔ Storage IO

Abstracted layers

- ➔ Virtualization
- ➔ Containerization

zalando

Overview

Bidirectional resource management

IPC mechanisms

New hardware support

…

zalando

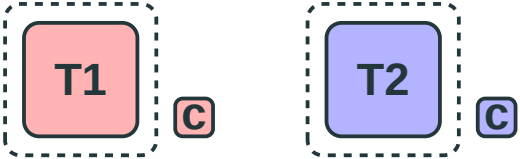
Execution scheduling

zalando

```
# Experiment 1
transaction type: pg_long.sql
latency average = 1312.903 ms

# Experiment 2
SQL script 1: pg_long.sql
 - weight: 1 (targets 50.0% of total)
 - latency average = 1426.928 ms

SQL script 2: pg_short.sql
 - weight: 1 (targets 50.0% of total)
 - latency average = 303.092 ms
```

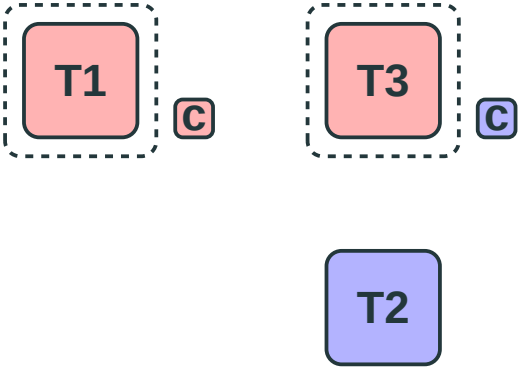
zalando

Scheduling



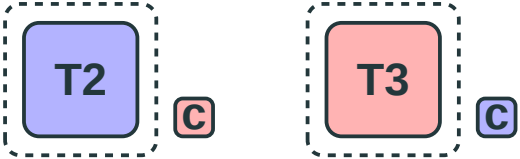
zalando

Scheduling



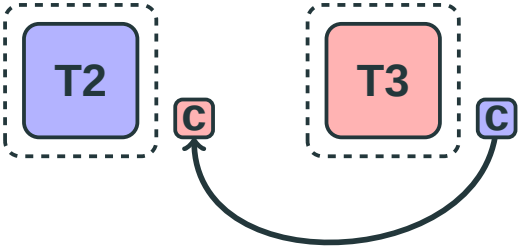
zalando

Scheduling



zalando

Scheduling



zalando

```
# Experiment 1
12,396,382,649      cache-misses # 28.562%
 2,750              cpu-migrations

# Experiment 2
20,665,817,234      cache-misses # 28.533%
10,460              cpu-migrations
```


zalando

```
          time    cpu  01234  task name                       wait time  sch delay   run time
                               [tid/pid]                          (msec)     (msec)     (msec)
-------------- ------  -----  ------------------------------  ---------  ---------  ---------
   4227.834476 [0003]     s   postgres[12935]                     0.000      0.000      0.000
   4227.834895 [0003]     s   postgres[12935]                     0.000      0.000      0.418
   4227.835478 [0003]     s   postgres[25080]                     0.000      0.040      0.583
   4227.836485 [0003]     s   postgres[25080]                     0.000      0.000      1.007
   4227.837482 [0003]     s   postgres[25080]                     0.000      0.000      0.996
   4227.837784 [0003]     s   postgres[25080]                     0.000      0.000      0.302
   4227.837989 [0003]      m    postgres[25080]                                                   migrated: :25077 cpu 1 => 3
   4227.837993 [0003]     s   postgres[25080]                     0.000      0.000      0.208
   4227.848487 [0003]     s   postgres[25080]                     0.000      0.000     10.493
   4227.848991 [0003]     s   postgres[25080]                     0.000      0.000      0.504
   4227.849487 [0003]     s   postgres[25080]                     0.000      0.000      0.495
   4227.849748 [0003]     s   postgres[25080]                     0.000      0.000      0.260
   4227.849912 [0003]     s   postgres[25080]                     0.000      0.000      0.164
   4227.851477 [0001]   s     postgres[25082]                     0.000      0.000      0.000
   4227.851481 [0002]    s    postgres[25080]                     0.000      0.000      0.000
   4227.851778 [0003]     s   postgres[12935]                    15.017      0.000      1.866
   4227.852259 [0003]      m    postgres[12935]                                                   migrated: postgres[25083] cpu 1 => 3
   4227.852263 [0003]     s   postgres[12935]                     0.000      0.000      0.484
   4227.852477 [0003]     s   postgres[25083]                     0.000      0.058      0.214
   4227.852478 [0001]   s     postgres[25082]                     0.000      0.000      1.001
   4227.852614 [0002]    s    postgres[12935]                     0.000      0.000      1.133
```

zalando

Tunables

zalando

Tunables

- /proc/sys/kernel/sched_migration_cost_ns

zalando

Tunables

- ➔ /proc/sys/kernel/sched_migration_cost_ns
- ➔ 500000

zalando

Tunables

- /proc/sys/kernel/sched_migration_cost_ns
- 500000
- /proc/sys/kernel/sched_wakeup_granularity_ns

zalando

Tunables

- /proc/sys/kernel/sched_migration_cost_ns
- 500000
- /proc/sys/kernel/sched_wakeup_granularity_ns
- 2000000

zalando

Tunables

- /proc/sys/kernel/sched_migration_cost_ns
- 500000
- /proc/sys/kernel/sched_wakeup_granularity_ns
- 2000000
- /proc/sys/kernel/sched_min_granularity_ns

zalando

Tunables

- /proc/sys/kernel/sched_migration_cost_ns
- 500000
- /proc/sys/kernel/sched_wakeup_granularity_ns
- 2000000
- /proc/sys/kernel/sched_min_granularity_ns
- 1500000

zalando

Tunables

- /proc/sys/kernel/sched_migration_cost_ns
- 500000
- /proc/sys/kernel/sched_wakeup_granularity_ns
- 2000000
- /proc/sys/kernel/sched_min_granularity_ns
- 1500000
- /proc/sys/kernel/sched_latency_ns

zalando

Tunables

- /proc/sys/kernel/sched_migration_cost_ns
- 500000
- /proc/sys/kernel/sched_wakeup_granularity_ns
- 2000000
- /proc/sys/kernel/sched_min_granularity_ns
- 1500000
- /proc/sys/kernel/sched_latency_ns
- 12000000

zalando

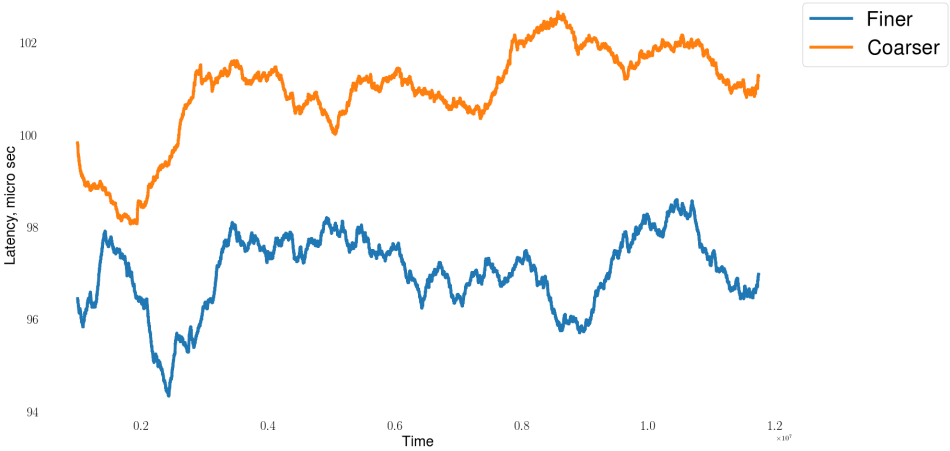
pgbench and pg_dump

```
real    1m38.990s
user    1m9.127s
sys     0m2.066s

     usecs               : count     distribution
         0 -> 1          : 16       |                                        |
         2 -> 3          : 4604     |**                                      |
         4 -> 7          : 6812     |****                                    |
         8 -> 15         : 14888    |*********                               |
        16 -> 31         : 19267    |***********                             |
        32 -> 63         : 65795    |****************************************|
        64 -> 127        : 50454    |******************************          |
       128 -> 255        : 16393    |*********                               |
       256 -> 511        : 5981     |***                                     |
       512 -> 1023       : 12300    |*******                                 |
      1024 -> 2047       : 48       |                                        |
      2048 -> 4095       : 0        |                                        |
```

pgbench and pg_dump

```
real    1m32.030s
user    1m8.559s
sys     0m1.641s

     usecs               : count     distribution
         0 -> 1          : 1        |                                        |
         2 -> 3          : 8        |                                        |
         4 -> 7          : 25       |                                        |
         8 -> 15         : 46       |*                                       |
        16 -> 31         : 189      |*******                                 |
        32 -> 63         : 119      |****                                    |
        64 -> 127        : 96       |***                                     |
       128 -> 255        : 93       |***                                     |
       256 -> 511        : 238      |*********                               |
       512 -> 1023       : 323      |************                            |
      1024 -> 2047       : 1012     |****************************************|
      2048 -> 4095       : 47       |*                                       |
```

Wakeup granularity, microsec



CPU hotplug and HyperThreading

Intel® 64 and IA-32 Architectures Optimization Reference Manual

zalando

CPU hotplug and HyperThreading

➔ Share execution state and cache

Intel® 64 and IA-32 Architectures Optimization Reference Manual

zalando

CPU hotplug and HyperThreading

- ➔ Share execution state and cache
- ➔ Spin locks have significant impact

Intel® 64 and IA-32 Architectures Optimization Reference Manual

zalando

CPU hotplug and HyperThreading

- ➔ Share execution state and cache
- ➔ Spin locks have significant impact
- ➔ PAUSE instruction (skylake latency 140 cycles)

Intel® 64 and IA-32 Architectures Optimization Reference Manual

zalando

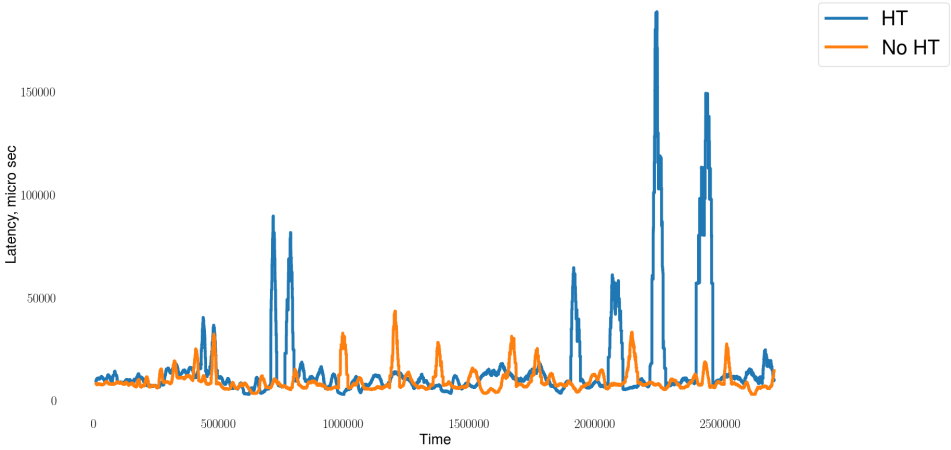
CPU hotplug and HyperThreading

- ➔ Share execution state and cache
- ➔ Spin locks have significant impact
- ➔ PAUSE instruction (skylake latency 140 cycles)
- ➔ More deviation for latency

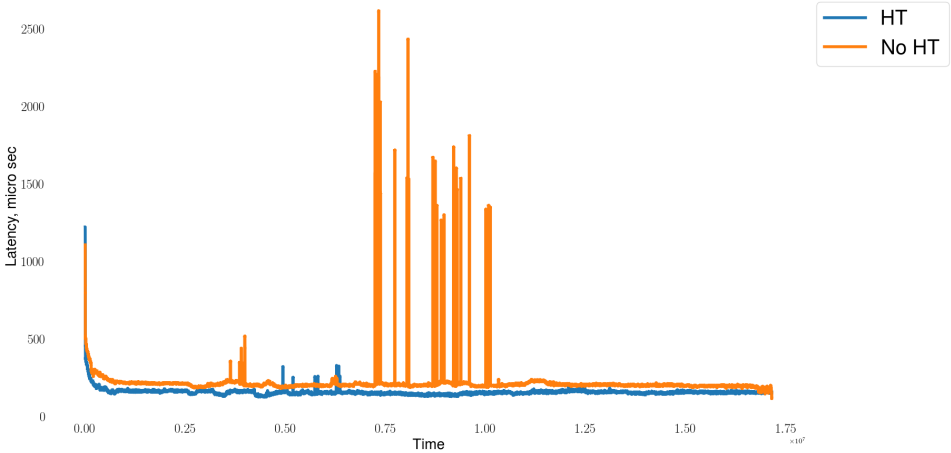
Intel® 64 and IA-32 Architectures Optimization Reference Manual

zalando

Latency rolling standard deviation, r/w



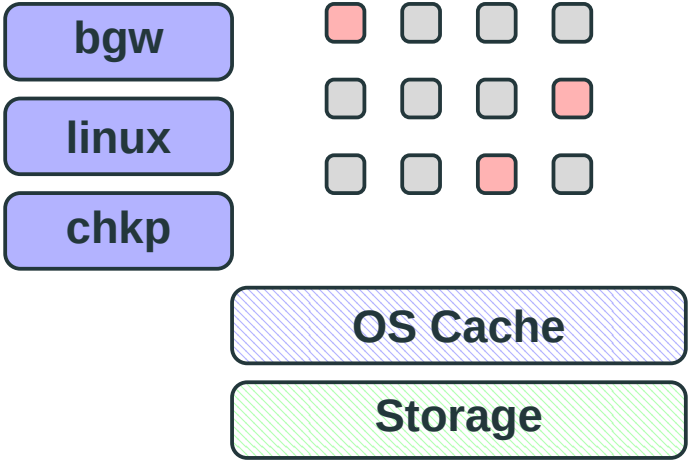
Latency rolling standard deviation, readonly



Memory management

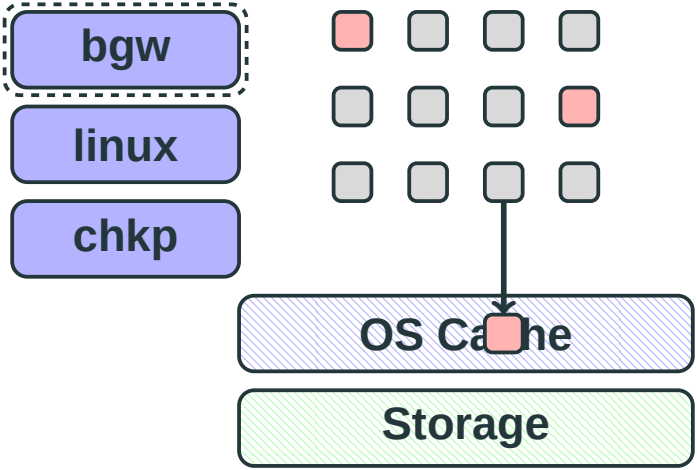
zalando

Dirty pages



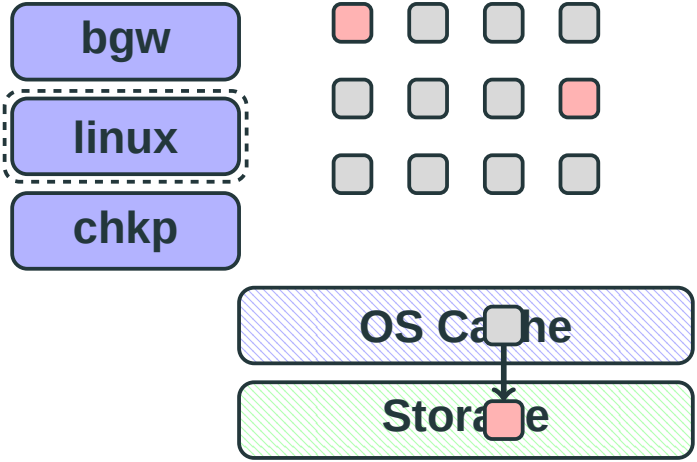
zalando

Dirty pages



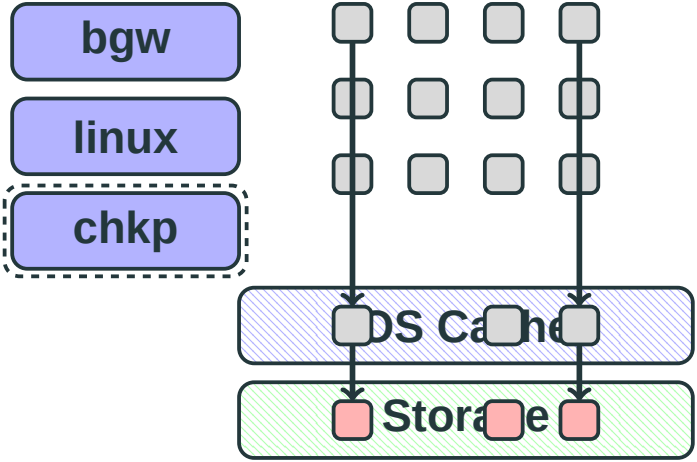
zalando

Dirty pages



zalando

Dirty pages



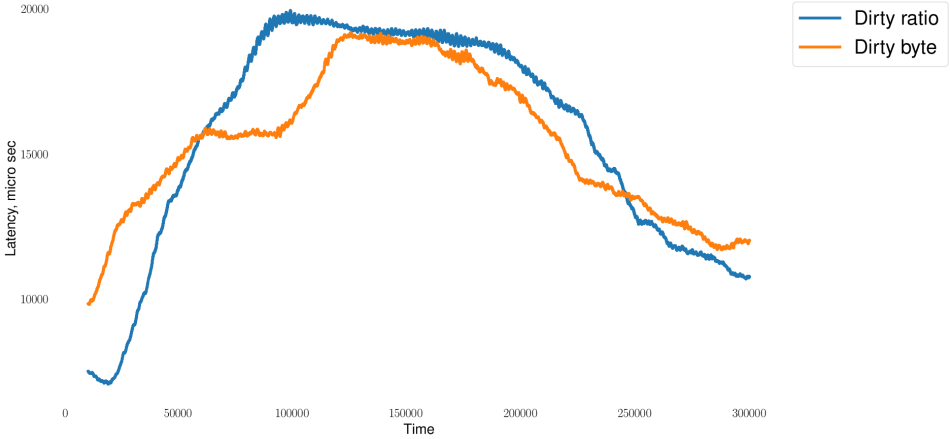
zalando

Dirty pages, r/w

- ➔ vm.dirty_ratio 20
- ➔ vm.dirty_background_ratio 10
- ➔ vm.dirty_bytes 0
- ➔ vm.dirty_backround_bytes 0

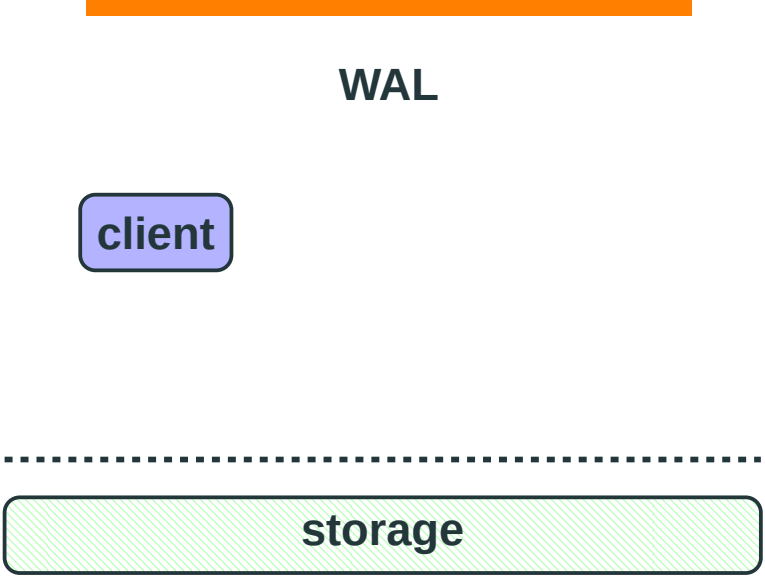
zalando

Dirty pages


Dirty ratio
Dirty byte
Latency, micro sec
20000
15000
10000
0
50000
100000
150000
200000
250000
300000
Time

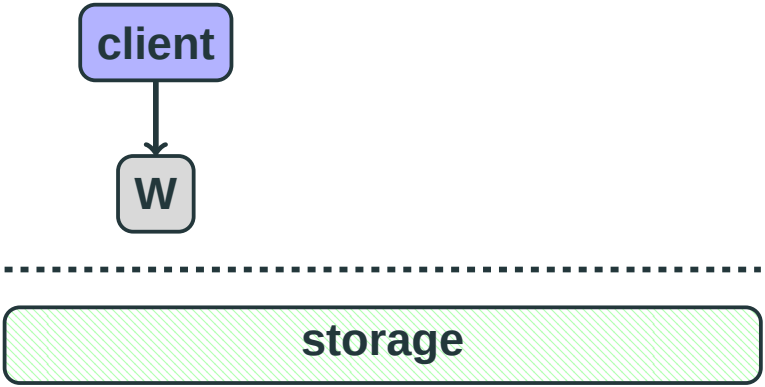
Storage IO

zalando



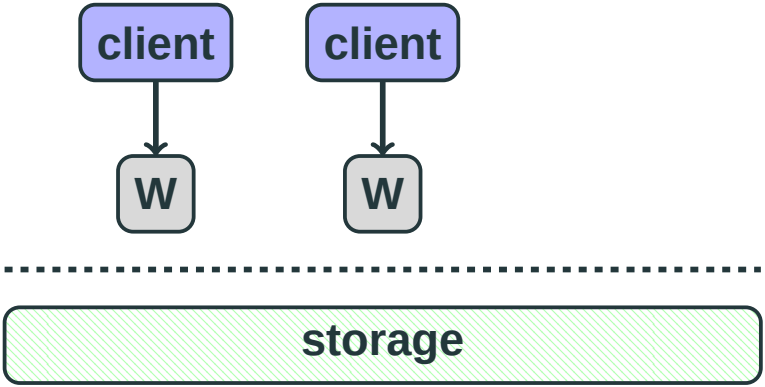
zalando

WAL



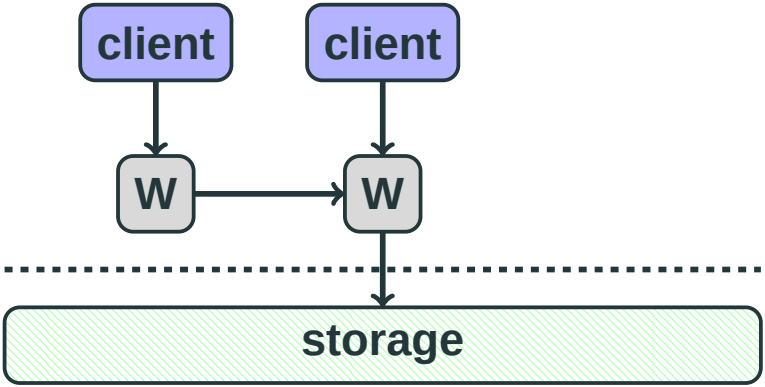
zalando

WAL



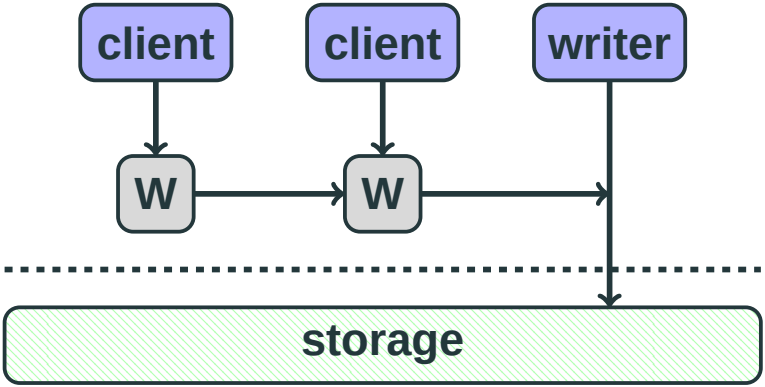
zalando

WAL



zalando

WAL



zalando

WAL

- ➔ Bufferer IO
- ➔ fdatasync
- ➔ Writeback error propagation

zalando

NVMe

- better for resourse sharing (PCI express) under the virtualization
- /sys/block/sda/queue/scheduler [noop|none]
- DSM operations

zalando

NVMe DSM

- ➔ Expected lifetime
- ➔ Prepare for some workload (read/write)
- ➔ Access frequency

zalando

DSM support

- Command DWORD 11 in ioctl
- fcntl SET_FILE_RW_HINT
- nvme-cli (ioctl)
- Specify a start block and a range length

NVM Express Revision 1.3c May 24, 2018

zalando

```
# get a start block
hdparm --fibmap data_file
data_file:
 filesystem blocksize 4096, begins at LBA 0;
 assuming 512 byte sectors.
 byte_offset  begin_LBA    end_LBA    sectors
           0   55041560   55041567          8

# set dsm for sequential read optimized
nvme dsm /dev/nvme1n01 --slbs=55041560 --blocks=1 --idr
```


zalando

Virtualization

zalando

Timekeeping

Timekeeping in VMware Virtual: Information Guide

zalando

Timekeeping

- ➔ Statistical sampling
  (occasional incorrect charging)

Timekeeping in VMware Virtual: Information Guide

zalando

Timekeeping

- ➔ Statistical sampling
  (occasional incorrect charging)
- ➔ Exact measurement (TSC time drift)

Timekeeping in VMware Virtual: Information Guide

zalando

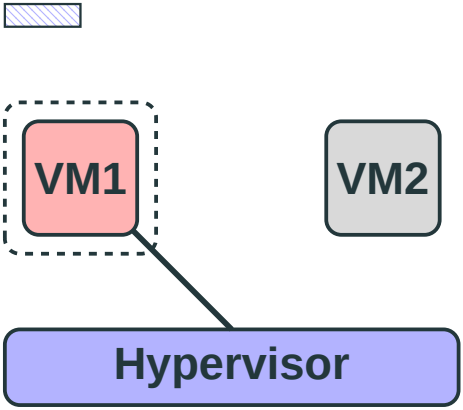
Timekeeping

- ➔ Statistical sampling
  (occasional incorrect charging)
- ➔ Exact measurement (TSC time drift)
- ➔ /sys/devices/system/clocksource/clocksource0/

Timekeeping in VMware Virtual: Information Guide

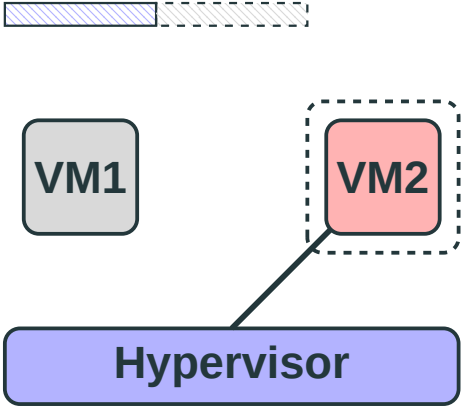
zalando

Scheduling



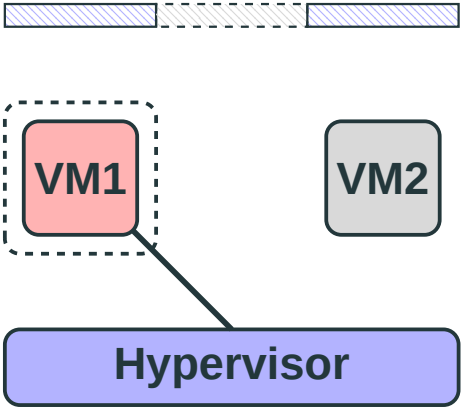
zalando

Scheduling



zalando

Scheduling



zalando

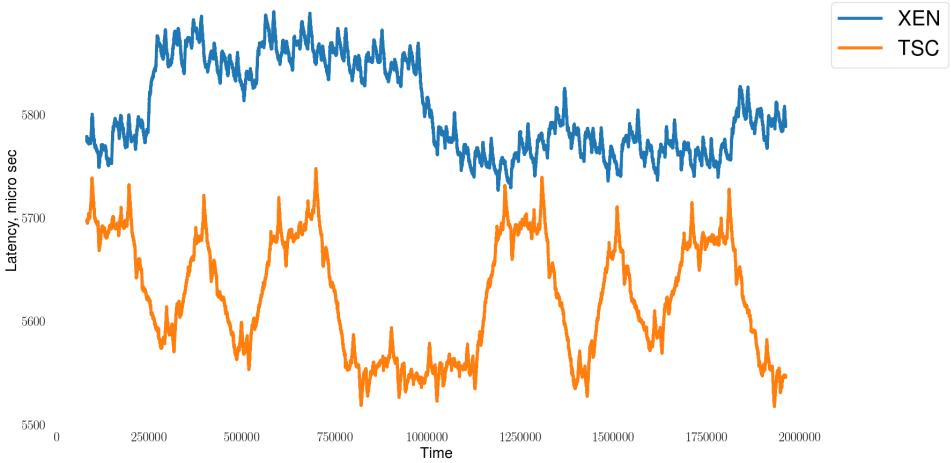
vDSO

- ➔ gettimeofday
- ➔ clock_gettime
- ➔ XEN doesn’t support vDSO for them
- ➔ unnecessary context switches to a kernel

Two frequently used system calls are 77% slower on AWS EC2

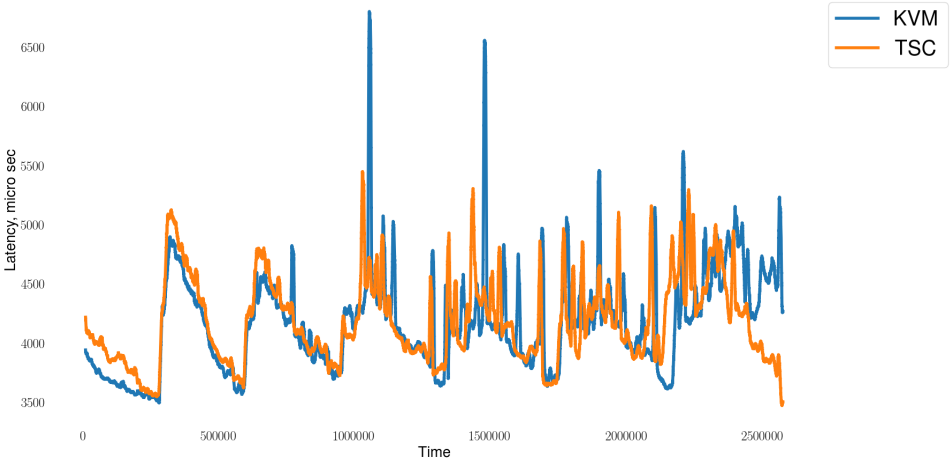
zalando

Latency m4.xlarge XEN/TSC, r/w



zalando

Latency m5.xlarge KVM/TSC, r/w



Locks

Intel® 64 and IA-32 Architectures Software Developer’s Manual, Vol. 3

zalando

Locks

➔ Lock holder preemption problem

Intel® 64 and IA-32 Architectures Software Developer’s Manual, Vol. 3

zalando

Locks

- ➔ Lock holder preemption problem
- ➔ Lock waiter preemption problem

Intel® 64 and IA-32 Architectures Software Developer’s Manual, Vol. 3

zalando

Locks

- ➔ Lock holder preemption problem
- ➔ Lock waiter preemption problem
- ➔ Intel PLE (pause loop exiting)

Intel® 64 and IA-32 Architectures Software Developer's Manual, Vol. 3

zalando

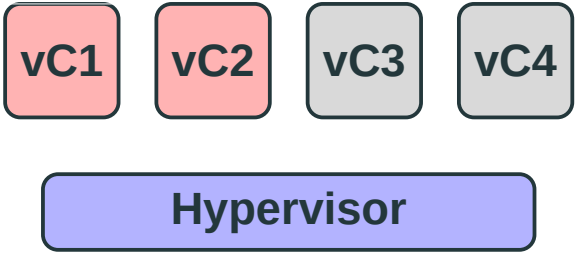
Locks

- ➔ Lock holder preemption problem
- ➔ Lock waiter preemption problem
- ➔ Intel PLE (pause loop exiting)
- ➔ PLE_Gap, PLE_Window

Intel® 64 and IA-32 Architectures Software Developer’s Manual, Vol. 3

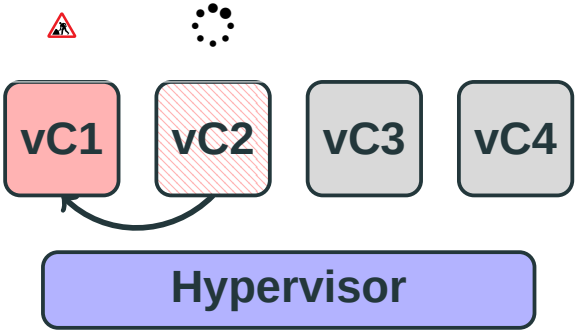
zalando

vCPU



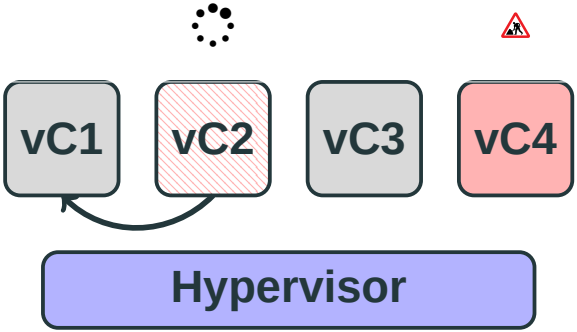
zalando

vCPU



zalando

vCPU



zalando

Containerization

zalando

cgroups controllers

- cpu,cpuacct
- cpuset
- memory
- devices
- freezer
- net_cls
- rdma
- blkio
- perf_event
- net_prio
- hugetlb
- pids
- rdma

zalando

```
8388   8388   postgres     blk_throtl_bio
        blk_throtl_bio+0x1          [kernel]
        dm_make_request+0x80        [kernel]
        generic_make_request+0xf6 [kernel]
        submit_bio+0x7d             [kernel]
        blkdev_issue_flush+0x68     [kernel]
        ext4_sync_file+0x310        [kernel]
        vfs_fsync_range+0x4b        [kernel]
        do_fsync+0x3d               [kernel]
        sys_fdatasync+0x13          [kernel]
        fdatasync+0x10              [libc-2.24.so]
        XLogBackgroundFlush+0x17e [postgres]
        WalWriterMain+0x1cb         [postgres]
        PostmasterMain+0xfea        [postgres]
```

zalando

bklio controller

- ➔ CFQ & throttling policy (generic block layer)
- ➔ No weight related options will work without CFQ
- ➔ Advisable io scheduler for SSD is noop/none
- ➔ Block layer do sampling to enforce throttling

zalando

throttle_sample_time

This is the time window that blk-throttle samples data, in millisecond. blk-throttle makes decision based on the samplings. Lower time means cgroups have more smooth throughput, but higher CPU overhead. This exists only when CONFIG_BLK_DEV_THROTTLING_LOW is enabled.

zalando

blkio

On traditional cgroup hierarchies, relationships between different controllers cannot be established making it impossible for writeback to operate accounting for cgroup resource restrictions and all writeback IOs are attributed to the root cgroup.

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

zalando

Bad neighbour

- ➔ memory fragmentation
- ➔ buddy allocator can fail
  to find a page of proper size
- ➔ kernel will start a compaction process

zalando

```
# Host, normal
 Zone: Normal
 Free KiB in zone: 807232.00
    Fragment size          Free fragments
    4096                   29612
    8192                   23308
    16384                  13495

# Host with a container
 Zone: Normal
 Free KiB in zone: 109700.00
    Fragment size          Free fragments
    4096                   3405
    8192                   7082
    16384                  1954
```


zalando

Bad neighbour

- ➔ PGSemaphore* funcions make use of futex
- ➔ Per-cpu hash table for futex with hash buckets

zalando

Bad neighbour

- ➔ WAL segment/heap file creation
- ➔ inode lock contention

Understanding Manycore Scalability of File Systems

zalando

Questions?

- github.com/erthalion
- github.com/erthalion/ansible-ycsb
- @erthalion
- dmitrii.dolgov at zalando dot de
- 9erthalion6 at gmail dot com

zalando