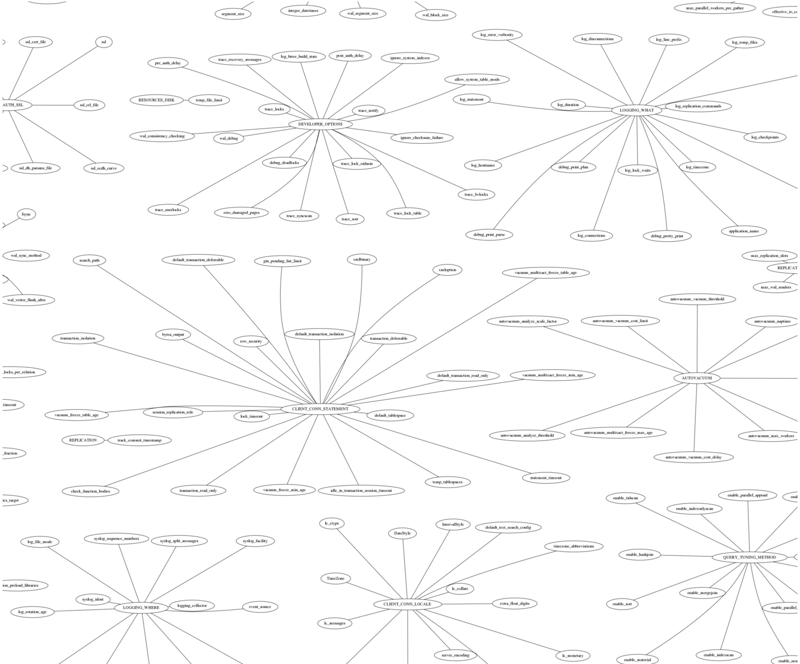
zalando

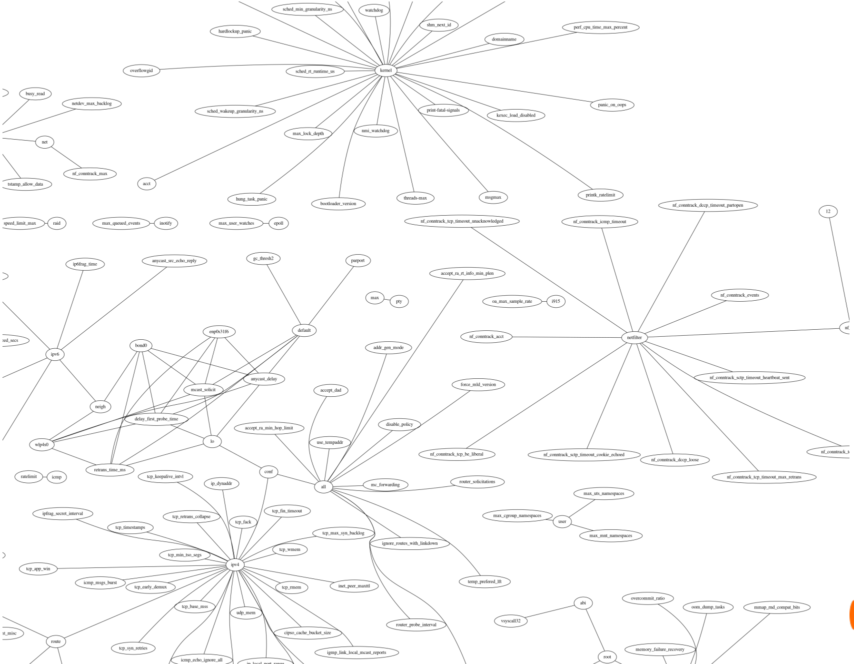
POSTGRESQL
LINUX KERNEL

FRIENDSHIP

DMITRY DOLGOV

22-10-2018


segment_size
integer_datetimes
wal_segment_size
wal_block_size
max_parallel_workers_per_gather
ssl_cert_file
ssl
AUTH_SSL
ssl_crl_file
ssl_dh_params_file
ssl_ecdh_curve
pre_auth_delay
trace_recovery_messages
log_btree_build_stats
post_auth_delay
ignore_system_indexes
allow_system_table_mods
RESOURCES_DISK
temp_file_limit
trace_locks
trace_notify
DEVELOPER_OPTIONS
wal_consistency_checking
wal_debug
ignore_checksum_failure
debug_deadlocks
trace_lock_oidmin
trace_lwlocks
trace_userlocks
zero_damaged_pages
trace_syncscan
trace_sort
trace_lock_table
log_error_verbosity
log_disconnections
log_line_prefix
log_temp_files
log_statement
log_duration
LOGGING_WHAT
log_replication_commands
log_checkpoints
log_hostname
debug_print_plan
log_lock_waits
log_timezone
debug_print_parse
log_connections
debug_pretty_print
application_name
fsync
wal_sync_method
wal_writer_flush_after
search_path
default_transaction_deferrable
gin_pending_list_limit
xmlbinary
xmloption
vacuum_multixact_freeze_table_age
max_replication_slots
max_wal_senders
autovacuum_vacuum_threshold
autovacuum_analyze_scale_factor
autovacuum_vacuum_cost_limit
autovacuum_naptime
transaction_isolation
bytea_output
row_security
default_transaction_isolation
transaction_deferrable
default_transaction_read_only
vacuum_multixact_freeze_min_age
AUTOVACUUM
vacuum_freeze_table_age
session_replication_role
lock_timeout
CLIENT_CONN_STATEMENT
default_tablespace
REPLICATION
track_commit_timestamp
autovacuum_analyze_threshold
autovacuum_multixact_freeze_max_age
autovacuum_max_workers
autovacuum_vacuum_cost_delay
check_function_bodies
transaction_read_only
vacuum_freeze_min_age
idle_in_transaction_session_timeout
temp_tablespaces
statement_timeout
enable_tidscan
enable_parallel_append
enable_indexonlyscan
lc_ctype
DateStyle
IntervalStyle
default_text_search_config
timezone_abbreviations
log_file_mode
syslog_sequence_numbers
syslog_split_messages
syslog_facility
enable_hashjoin
QUERY_TUNING_METHOD
TimeZone
lc_collate
syslog_ident
LOGGING_WHERE
logging_collector
event_source
log_rotation_age
CLIENT_CONN_LOCALE
extra_float_digits
enable_mergejoin
enable_sort
lc_messages
server_encoding
lc_monetary
enable_material
enable_indexscan


sched_min_granularity_ns
watchdog
hardlockup_panic
shm_next_id
perf_cpu_time_max_percent
domainname
overflowgid
sched_rt_runtime_us
kernel
busy_read
netdev_max_backlog
sched_wakeup_granularity_ns
panic_on_oops
print-fatal-signals
kexec_load_disabled
max_lock_depth
nmi_watchdog
net
nf_conntrack_max
tstamp_allow_data
acct
hung_task_panic
bootloader_version
threads-max
msgmax
printk_ratelimit
nf_conntrack_dccp_timeout_partopen
12
speed_limit_max
raid
max_queued_events
inotify
max_user_watches
epoll
nf_conntrack_tcp_timeout_unacknowledged
nf_conntrack_icmp_timeout
gc_thresh2
parport
ip6frag_time
anycast_src_echo_reply
accept_ra_rt_info_min_plen
max
pty
oa_max_sample_rate
i915
nf_conntrack_events
enp0s31f6
default
bond0
nf_conntrack_acct
netfilter
ipv6
addr_gen_mode
anycast_delay
mcast_solicit
nf_conntrack_sctp_timeout_heartbeat_sent
accept_dad
force_mld_version
neigh
delay_first_probe_time
disable_policy
accept_ra_min_hop_limit
lo
use_tempaddr
wlp4s0
nf_conntrack_tcp_be_liberal
nf_conntrack_sctp_timeout_cookie_echoed
nf_conntrack_dccp_loose
retrans_time_ms
conf
ratelimit
icmp
tcp_keepalive_intvl
ip_dynaddr
all
mc_forwarding
router_solicitations
nf_conntrack_tcp_timeout_max_retrans
max_uts_namespaces
ipfrag_secret_interval
tcp_retrans_collapse
tcp_fin_timeout
max_cgroup_namespaces
user
tcp_timestamps
tcp_fack
tcp_max_syn_backlog
max_mnt_namespaces
tcp_wmem
ignore_routes_with_linkdown
tcp_min_tso_segs
tcp_app_win
ipv4
icmp_msgs_burst
tcp_early_demux
tcp_rmem
inet_peer_maxttl
temp_prefered_lft
overcommit_ratio
abi
oom_dump_tasks
mmap_rnd_compat_bits
tcp_base_mss
udp_mem
vsyscall32
router_probe_interval
cipso_cache_bucket_size
route
tcp_syn_retries
igmp_link_local_mcast_reports
memory_failure_recovery
icmp_echo_ignore_all
root

Plan

Plan

Overview

Plan

Overview

Resources

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

zalando

Plan

Overview

Resources

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

Abstracted layers

- ➔ Virtualization
- ➔ Containerization

Overview

Bidirectional resource management

IPC mechanisms

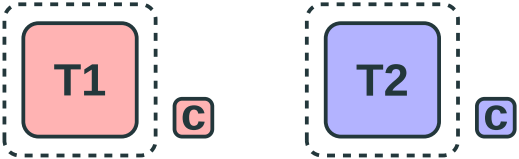
New hardware support

...

Execution scheduling

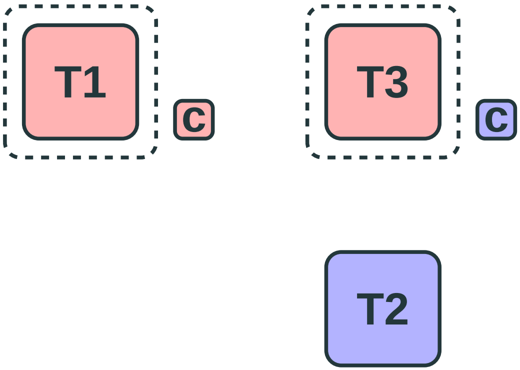
```
# Experiment 1
transaction type: pg_long.sql
latency average = 1312.903 ms

# 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
```

Scheduling

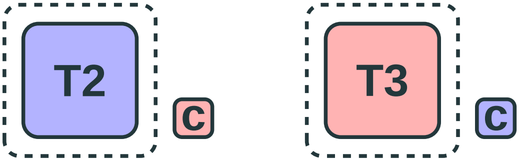


zalando

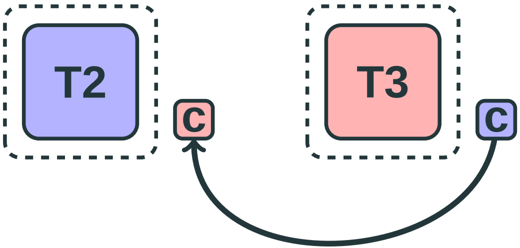
Scheduling



Scheduling



Scheduling



```
# 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
```

```
          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
```

Tunables

Tunables

➔ /proc/sys/kernel/sched_migration_cost_ns

Tunables

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

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

Tunables

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

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

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

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

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        |                                        |
```

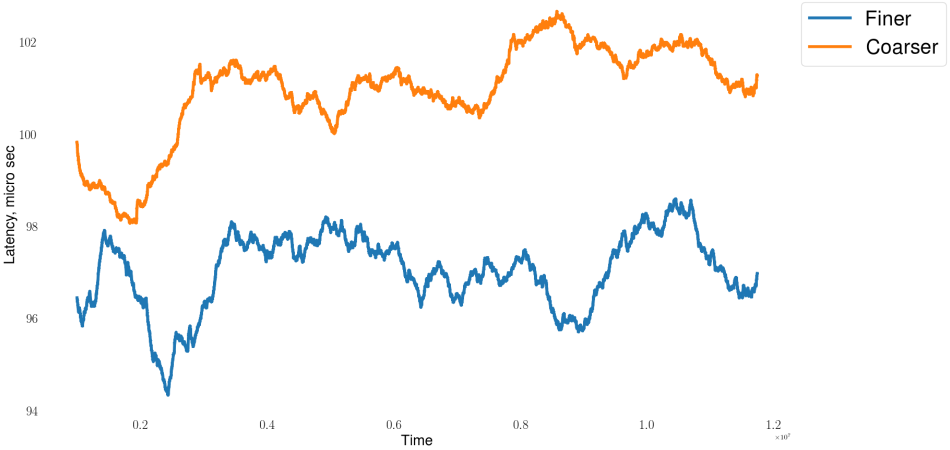

zalando

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       |*                                       |
```

zalando

Wakeup granularity, microsec



zalando

CPU hotplug and HyperThreading

Intel® 64 and IA-32 Architectures Optimization Reference Manual

CPU hotplug and HyperThreading

➔ Share execution state and cache

Intel® 64 and IA-32 Architectures Optimization Reference Manual

CPU hotplug and HyperThreading

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

Intel® 64 and IA-32 Architectures Optimization Reference Manual

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

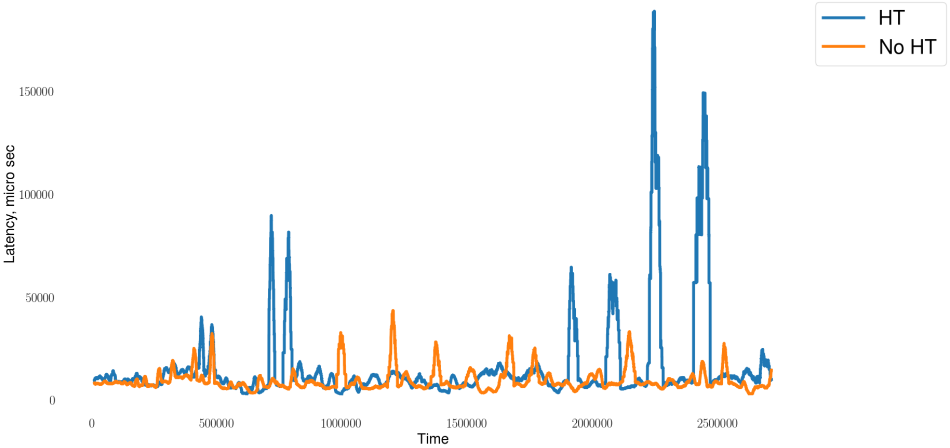
CPU hotplug and HyperThreading

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

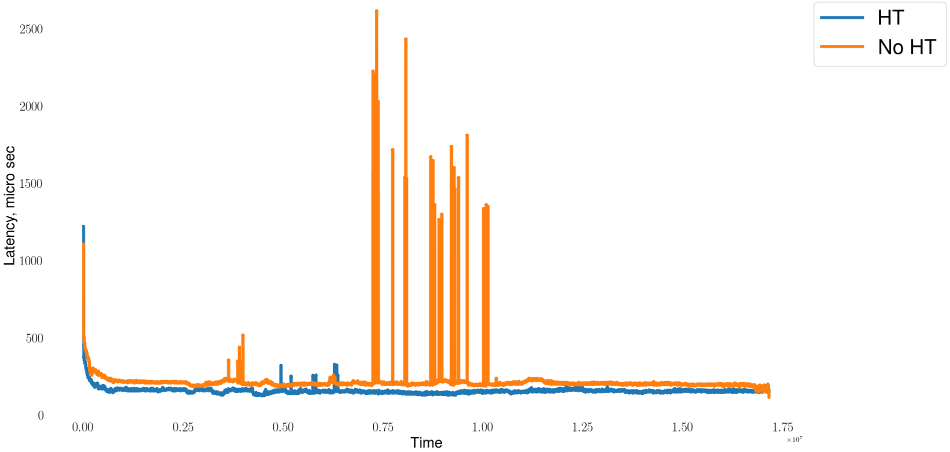
Intel® 64 and IA-32 Architectures Optimization Reference Manual

zalando

Latency rolling standard deviation, r/w

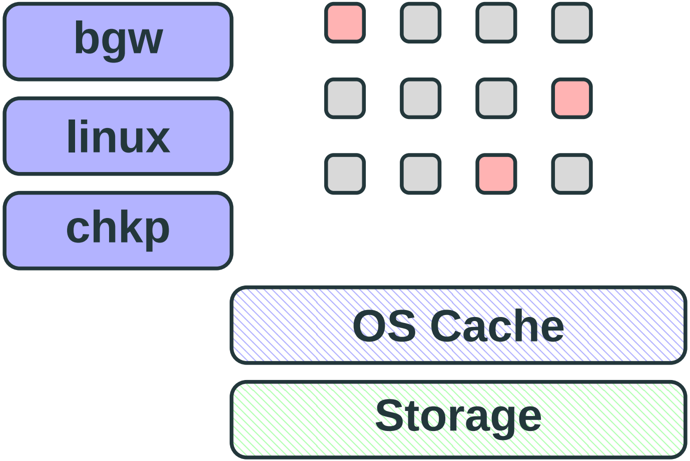


Latency rolling standard deviation, readonly

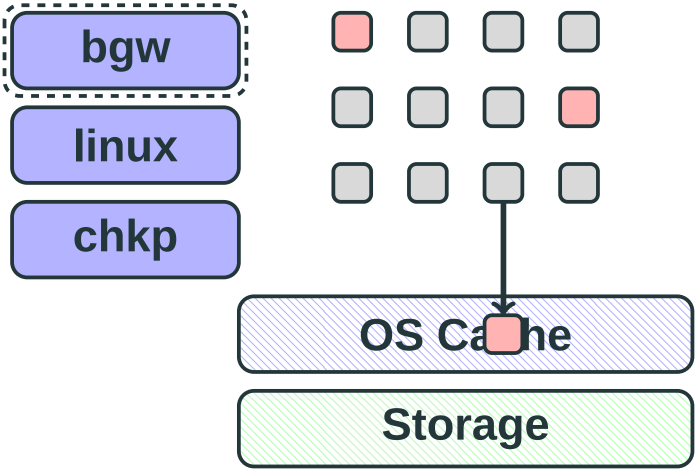


Memory management

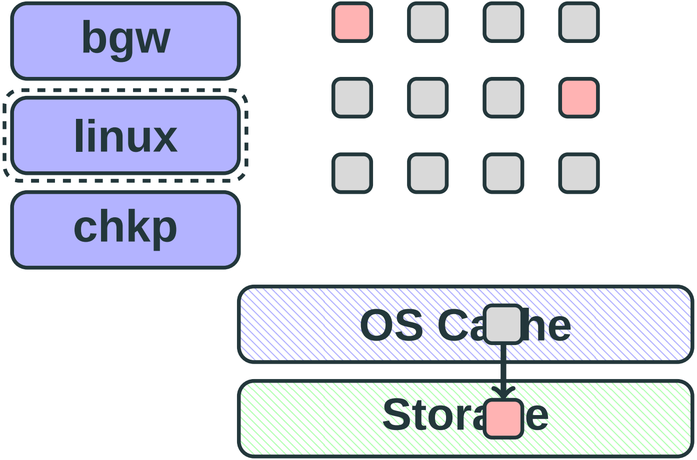
Dirty pages



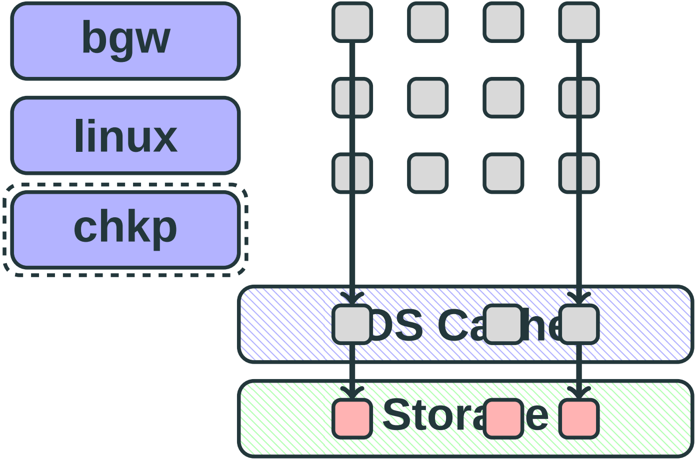
Dirty pages



Dirty pages



Dirty pages

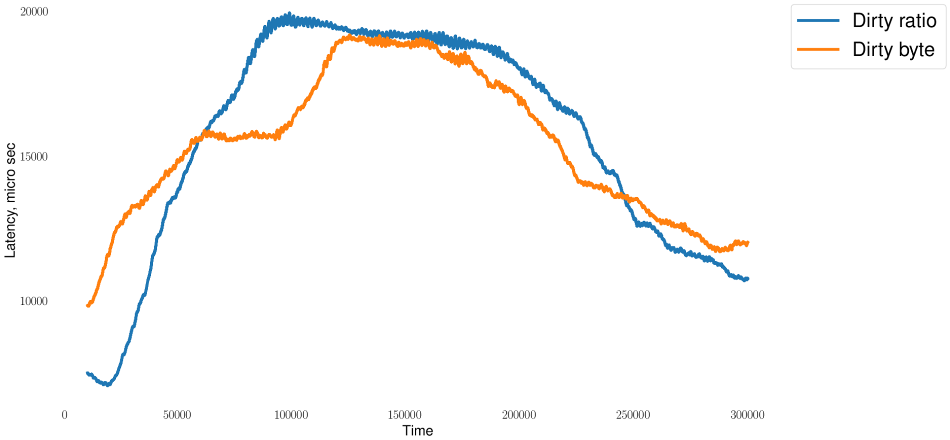


Dirty pages, r/w

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

zalando

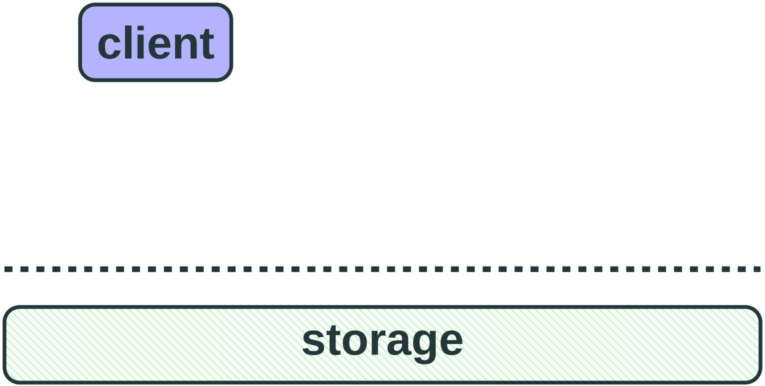
Dirty pages



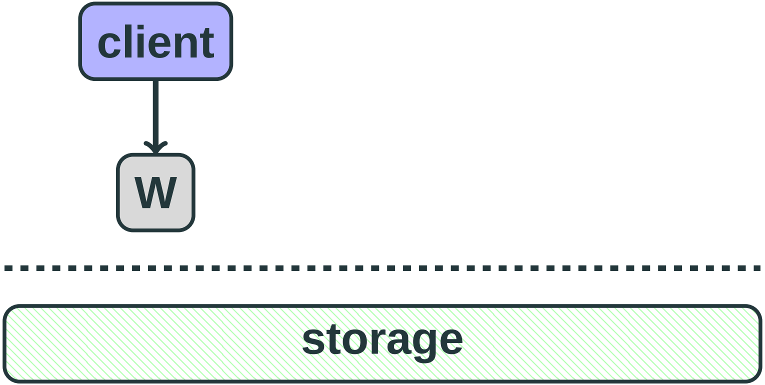
Dirty ratio
Dirty byte
20000
15000
10000
Latency, micro sec
0
50000
100000
150000
200000
250000
300000
Time

Storage IO

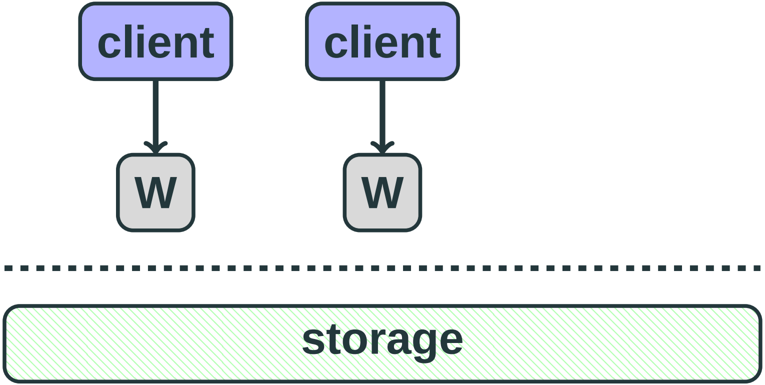
WAL



WAL

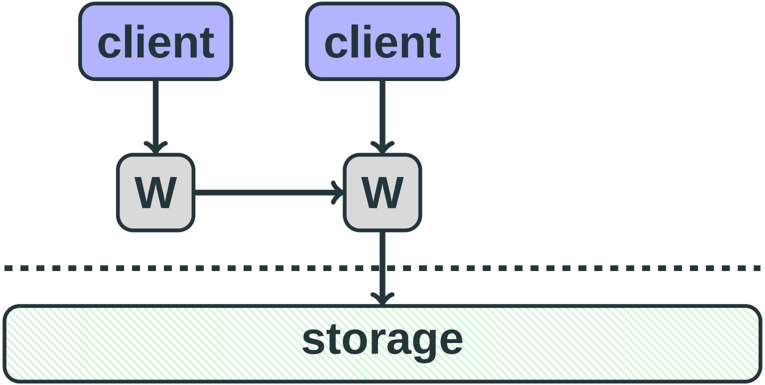


WAL

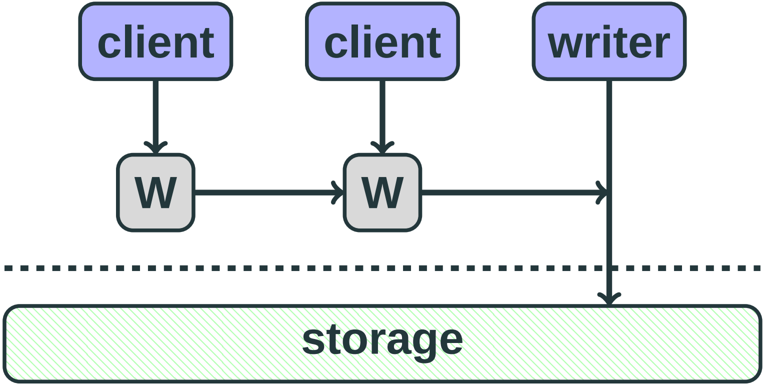


zalando

WAL



WAL



zalando

WAL

- Bufferer IO
- fdatasync
- Writeback error propagation

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

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

```
# 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

Timekeeping

Timekeeping in VMware Virtual: Information Guide

Timekeeping

→ Statistical sampling
(occasional incorrect charging)

Timekeeping in VMware Virtual: Information Guide

Timekeeping

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

Timekeeping in VMware Virtual: Information Guide

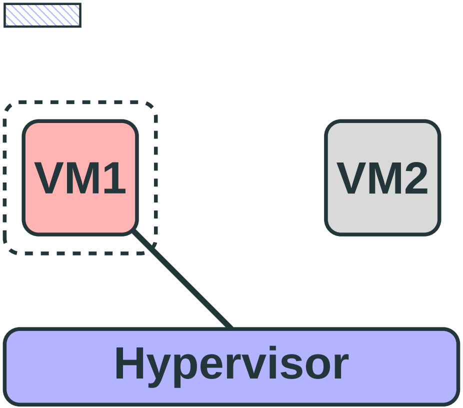
Timekeeping

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

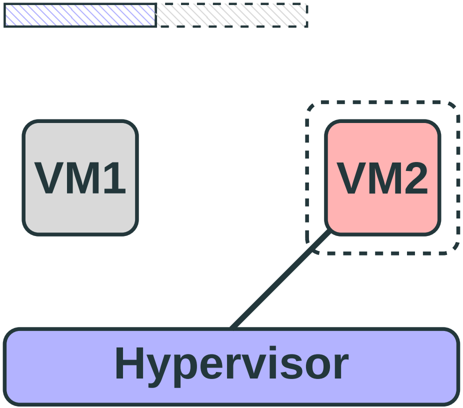
Timekeeping in VMware Virtual: Information Guide

zalando

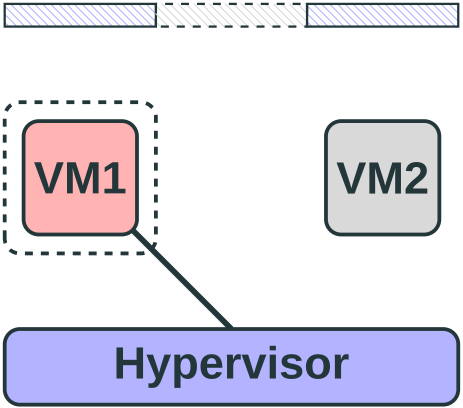
Scheduling



Scheduling



Scheduling



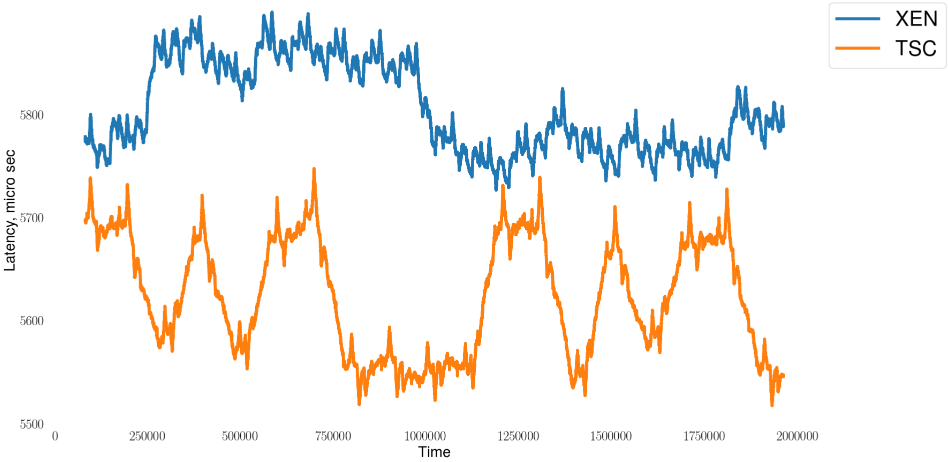
vDSO

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

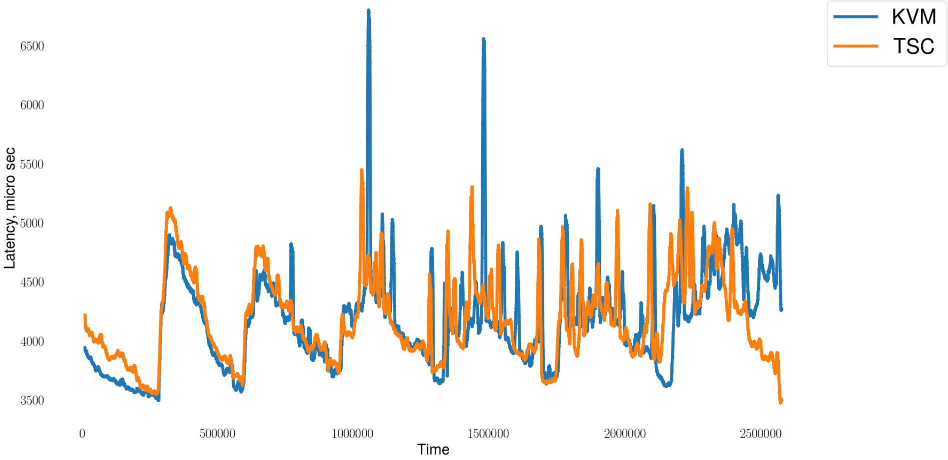
Two frequently used system calls are 77% slower on AWS EC2

zalando

Latency m4.xlarge XEN/TSC, r/w



Latency m5.xlarge KVM/TSC, r/w



Locks

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

Locks

➔ Lock holder preemption problem

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

Locks

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

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

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

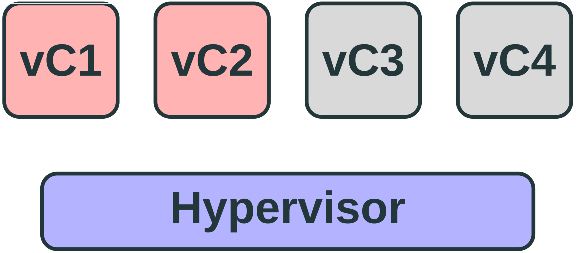
Locks

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

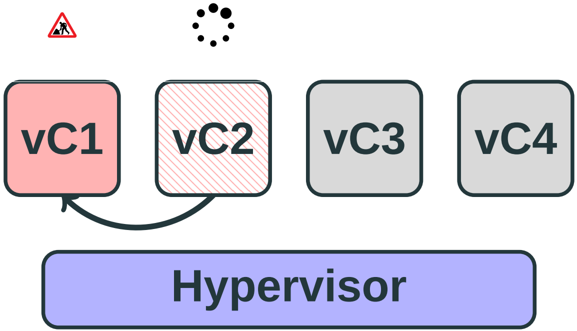
Intel® 64 and IA-32 Architectures Software Developer’s Manual, Vol. 3

zalando

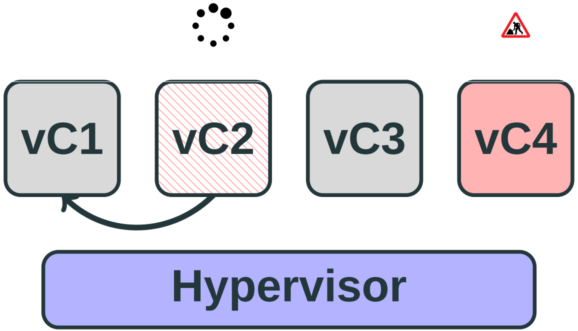
vCPU



vCPU



vCPU



Containerization

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]
```

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

Bad neighbour

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

```
# 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
```

Bad neighbour

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

Bad neighbour

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

Understanding Manycore Scalability of File Systems

Questions?

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