PROFESSIONAL
Postgres

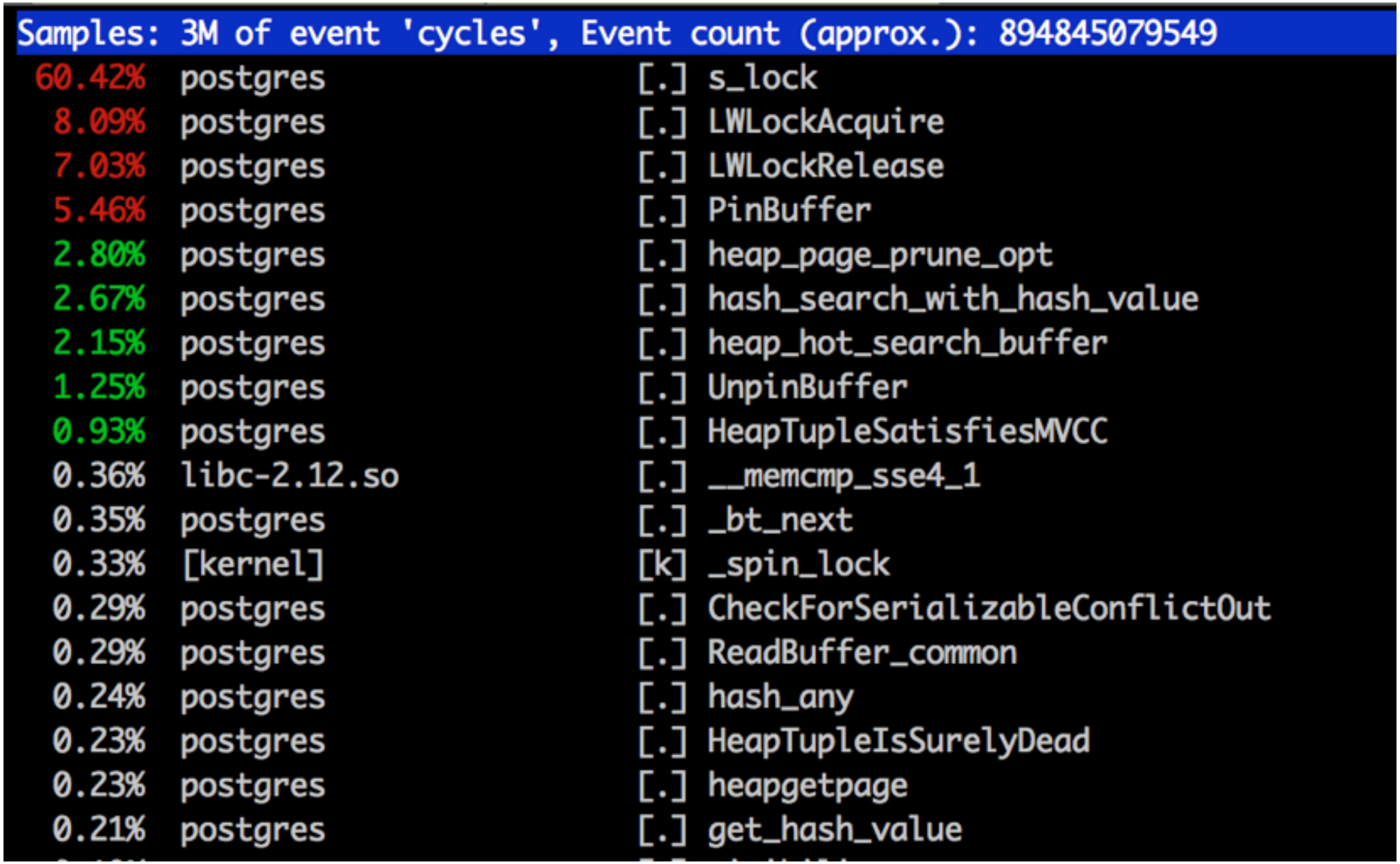
Waits monitoring in PostgreSQL

Ildus Kurbangaliev
i.kurbangaliev@gmail.com

Tools

- perf
- pg_stat_statements
- pg_stat_kcache
- SystemTap
- and others...

Perf



```
Samples: 3M of event 'cycles', Event count (approx.): 894845079549
60.42%  postgres              [.] s_lock
 8.09%  postgres              [.] LWLockAcquire
 7.03%  postgres              [.] LWLockRelease
 5.46%  postgres              [.] PinBuffer
 2.80%  postgres              [.] heap_page_prune_opt
 2.67%  postgres              [.] hash_search_with_hash_value
 2.15%  postgres              [.] heap_hot_search_buffer
 1.25%  postgres              [.] UnpinBuffer
 0.93%  postgres              [.] HeapTupleSatisfiesMVCC
 0.36%  libc-2.12.so          [.] __memcmp_sse4_1
 0.35%  postgres              [.] _bt_next
 0.33%  [kernel]              [k] _spin_lock
 0.29%  postgres              [.] CheckForSerializableConflictOut
 0.29%  postgres              [.] ReadBuffer_common
 0.24%  postgres              [.] hash_any
 0.23%  postgres              [.] HeapTupleIsSurelyDead
 0.23%  postgres              [.] heapgetpage
 0.21%  postgres              [.] get_hash_value
```

Monitored waits

- Locks (heavyweight)
- LWLocks (lightweight locks)
- Latch
- Network
- Storage (IO)

Subtypes of waits

- Locks (heavyweight)
  - 8 types (9 in 9.5+)
- LWLocks (lightweight locks)
- Latch
- Network
  - Reads and writes
- Storage (IO)
  - Reads and writes (storage manager, xlog, slru)

LWLocks

- Locks (lightweight)

Individual (41)

Groups of locks (buffer manager, processes, predicate and others)

User defined LWLocks (extensions)

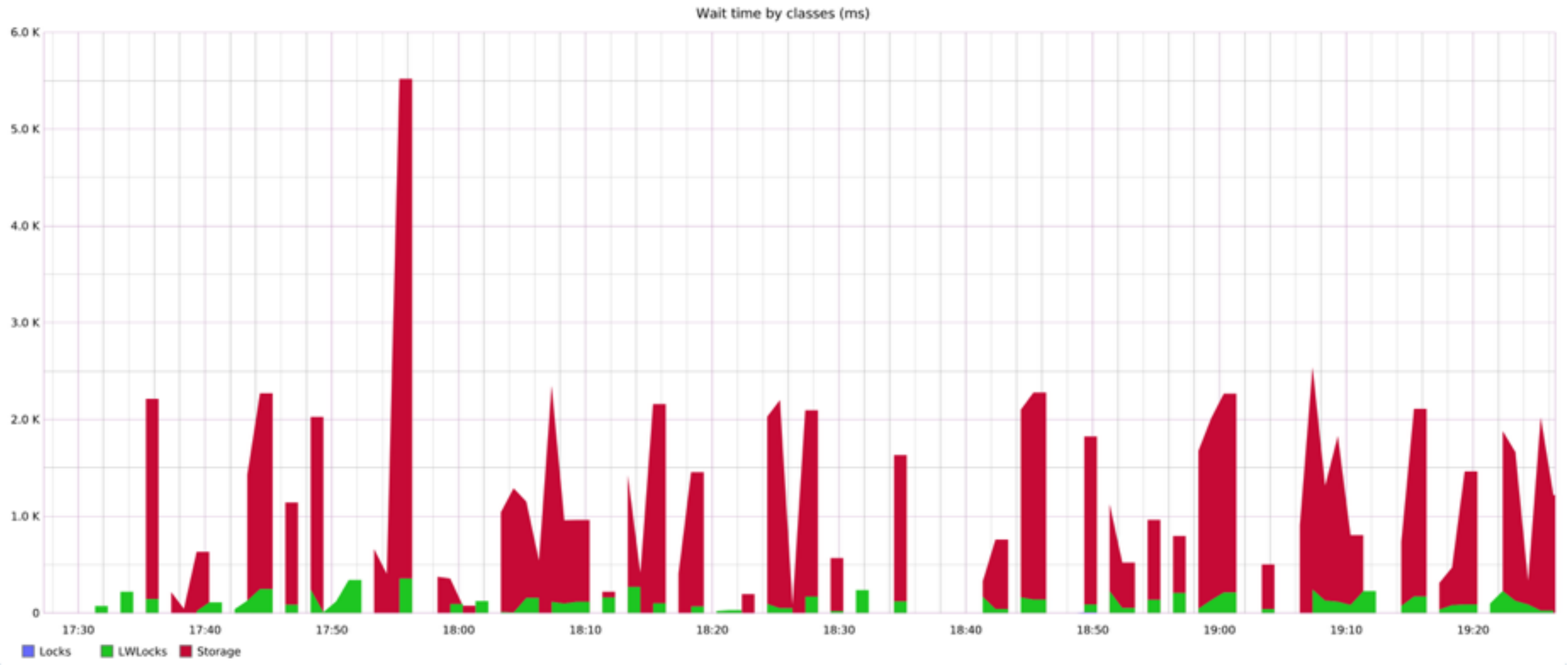
Profiling

```
b1=# select * from pg_stat_wait_profile
 where event_name = 'WALWriteLock' limit 1;

-[ RECORD 1 ]------------
pid         | 1804
class_id    | 1
class_name  | LWLocks
event_id    | 8
event_name  | WALWriteLock
wait_time   | 8719
wait_count  | 6
```

wait_time and **wait_count** fields show time spent on waits

Profiling



Red - storage waits

Waits history

```
b1=# select * from pg_stat_wait_history limit 1;
-[ RECORD 1 ]-----------------------------
pid        | 1809
sample_ts  | 2015-10-29 04:58:53.85285-04
class_id   | 3
class_name | Storage
event_id   | 0
event_name | SMGR_READ
wait_time  | 10299
p1         | 1663
p2         | 16384
p3         | 12214
p4         | 0
p5         | 1
```

Tracing

It has a big **overhead** and can be used only within separate sessions

```
terminal 1: $ psql b1

terminal 2:
$ ps ax | grep postgres
<...>
11085 ?        Ss     0:00 postgres: postgres b1 [local]
idle

$ psql b1 -c "select pg_start_trace(11085, '/tmp/
f.trace')"

terminal 1:
b1=# CREATE TABLE t1 AS SELECT i, i*10 AS i1 FROM
generate_series(1,10) i;
SELECT 10
```

Tracing

```
terminal 2:
$ tail -f /tmp/f.trace
stop 2015-07-10 10:03:35.603458-04 Network
start 2015-07-10 10:03:35.603464-04 Network READ 0 0 0 0 0
stop 2015-07-10 10:03:44.099587-04 Network
start 2015-07-10 10:03:44.100401-04 Storage READ 1663 16384 1259 2 0
stop 2015-07-10 10:03:44.100424-04 Storage
start 2015-07-10 10:03:44.102549-04 Network WRITE 0 0 0 0 0
stop 2015-07-10 10:03:44.102573-04 Network
start 2015-07-10 10:03:44.102582-04 Network READ 0 0 0 0 0
stop 2015-07-10 10:05:33.029975-04 Network
start 2015-07-10 10:05:33.030205-04 Storage READ 1663 16384 2691 0 28
stop 2015-07-10 10:05:33.030233-04 Storage
start 2015-07-10 10:05:33.030246-04 Storage READ 1663 16384 1255 0 50
stop 2015-07-10 10:05:33.03026-04 Storage
```

Overhead

Server configuration:

- Intel(R) Xeon(R) CPU X5675@3.07GHz, 24 cores
- RAM 24 GB
- pgbench -S 500 ~ 1.6 Gb

Reduce impact of disk I/O:

- fsync off
- tmpfs

Monitoring off

```
$ pgbench -S b1 -c 96 -j 4 -T 300
starting vacuum...end.
transaction type: SELECT only
scaling factor: 500
query mode: simple
number of clients: 96
number of threads: 4
duration: 300 s
number of transactions actually processed: 39349816
latency average: 0.732 ms
tps = 131130.859559 (including connections establishing)
tps = 131153.752204 (excluding connections establishing)
```

Monitoring on

```
$ pgbench -S b1 -c 96 -j 4 -T 300
starting vacuum...end.
transaction type: SELECT only
scaling factor: 500
query mode: simple
number of clients: 96
number of threads: 4
duration: 300 s
number of transactions actually processed: 39172607
latency average: 0.735 ms
tps = 130574.626755 (including connections establishing)
tps = 130600.767440 (excluding connections establishing)
```

Thank you

Ildus Kurbangaliev
i.kurbangaliev@gmail.com