CPU performance analysis for MySQL using Hot/Cold Flame Graphs

Vinicius M. Grippa
Senior Support Engineer for MySQL/MongoDB
vinicius.grippa@percona.com

PERCONA

About Percona

- Founded in 2006
- 190+ staff worldwide
- Percona blog has **100k+ readers** per month
- Percona tools have been used **over a billion times** in the past 3 years
- Percona conferences attract **thousands of attendees** from some of the world's largest companies (**#PerconaLive**)
- Percona software is installed on **millions of servers**

About me

- Support Engineer at Percona since 2017.
- Working with MySQL for over six years.
- Working with databases for over nine years.
- Speaker at global conferences and meetups about MySQL/MongoDB.
- Social media
  - https://www.linkedin.com/in/vinicius-grippa/
  - vinicius.grippa@percona.com / vgrippa@gmail.com

So Brendan

Or "What are FlameGraphs?"

What are FlameGraphs

Flame graphs are a visualization of profiled software, allowing the most frequent code-paths to be identified quickly and accurately

~ Brendan Gregg

http://www.brendangregg.com/FlameGraphs/cpuflamegraphs.html

What are FlameGraphs: code path

```
int main(){
    ...
    GetBookList(book);
    ...
}

void GetBookList(Book book){
    …
    while (book =
    GetBook(book);
    ...
}
```

```
void GetBook(Book book){
   a = GetAuthor(book);
   pub = GetPublisher(book);
   p =
GetPrints(book).toString();

   printf("%s published by
%s [prints: %s]", a,pub,p);
    ...
}
```

PERCONA

What are FlameGraphs: code path

Possible call chains

main() -> GetBooksList() -> GetBook() -> GetAuthor()...

main() -> GetBooksList() -> GetBook() -> GetEditorial()...

main() -> GetBooksList() -> GetBook() -> GetPrints()...

What are FlameGraphs: profiling

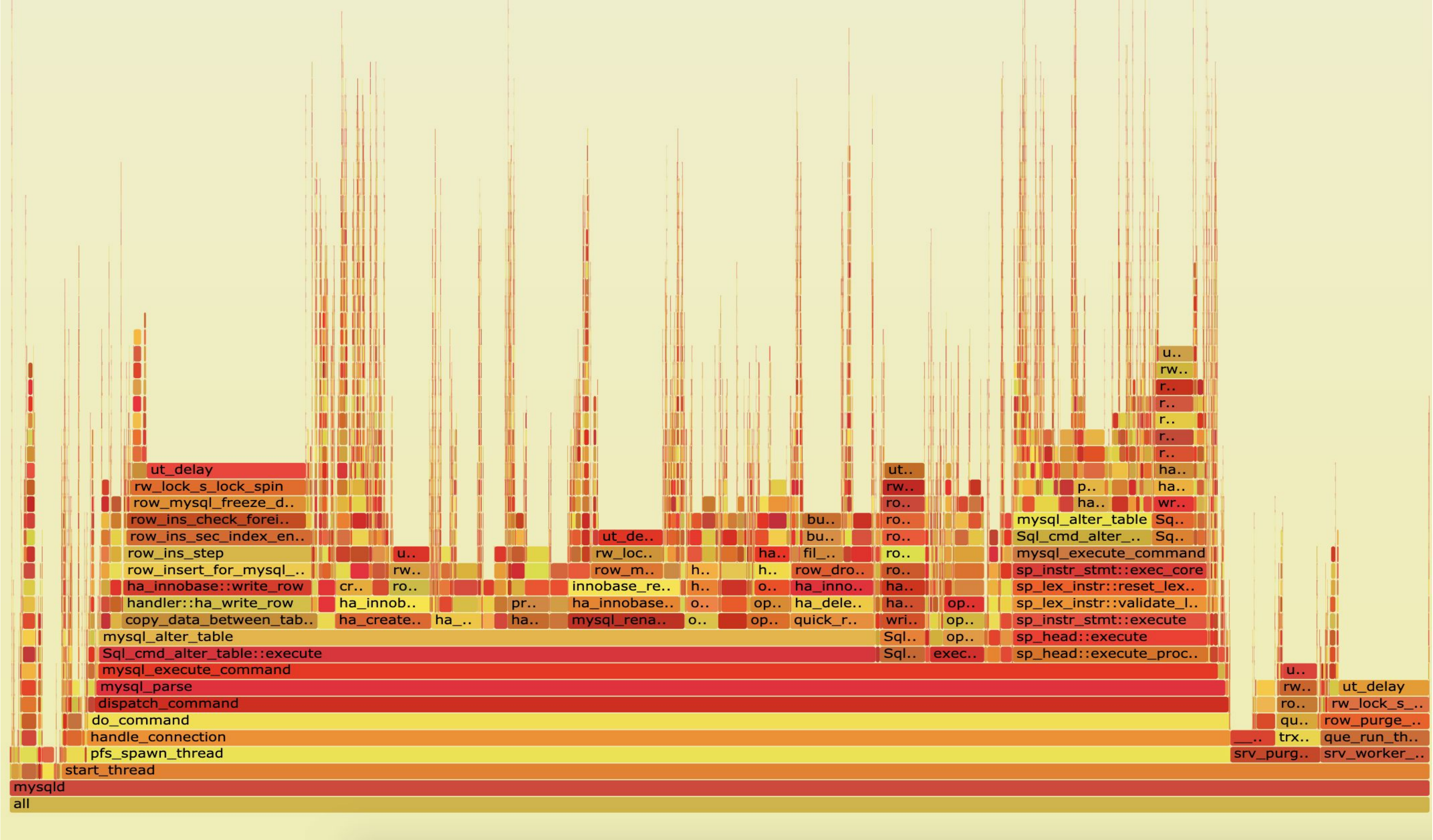
Captures of metrics of some type for later aggregation.

What are FlameGraphs

```
# profile it
perf record -a -g -F99 -p $(pgrep -x mysqld) --
sleep 60

# make it machine-readable
perf script > s.out

# graph it!
stackcollapse-perf.pl s.out | flamegraph.pl > s.svg
```



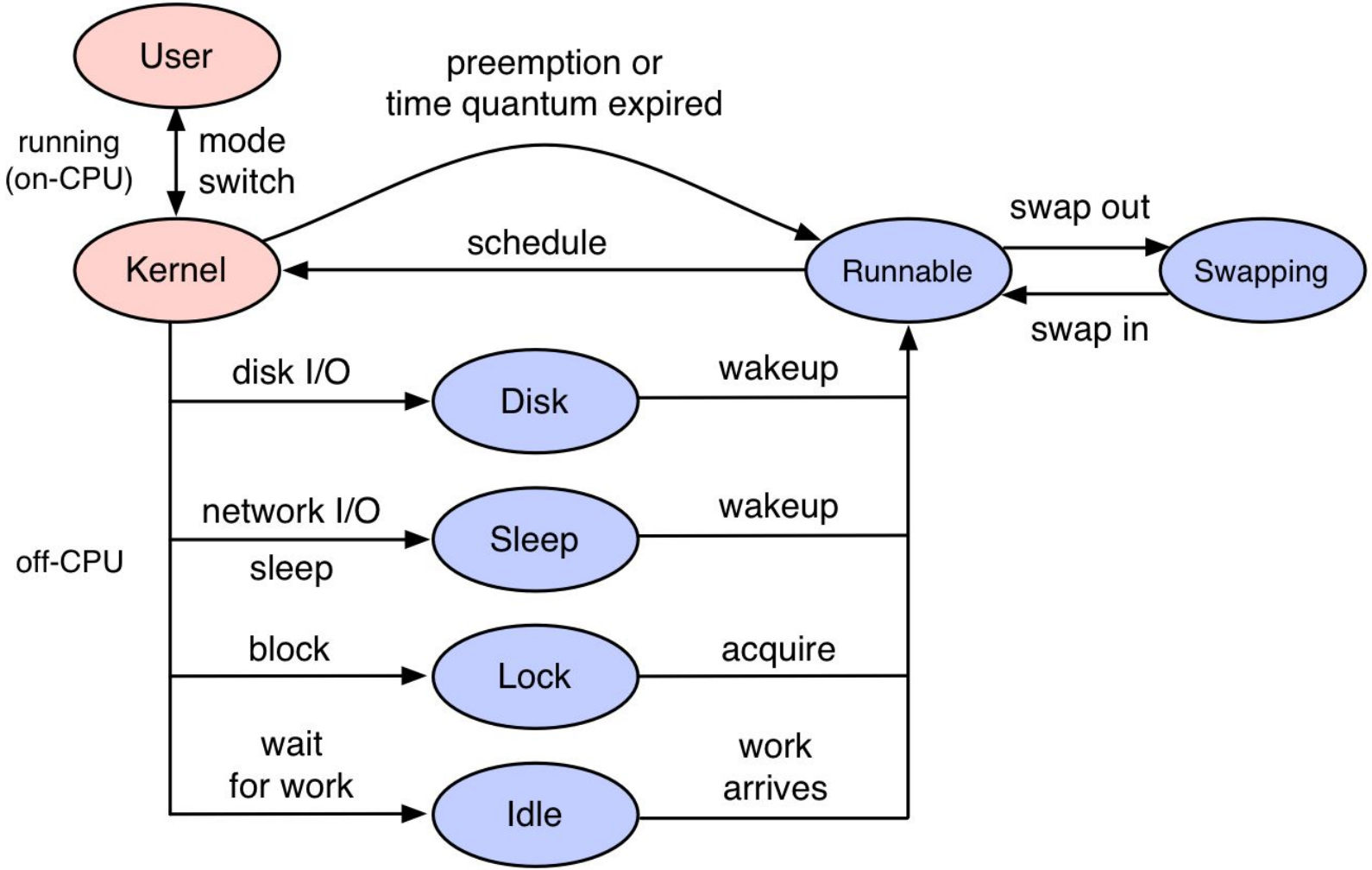
ut_delay
rw_lock_s_lock_spin
row_mysql_freeze_d..
row_ins_check_forei..
row_ins_sec_index_en..
row_ins_step
row_insert_for_mysql_..
ha_innobase::write_row
handler::ha_write_row
copy_data_between_tab..
mysql_alter_table
Sql_cmd_alter_table::execute
mysql_execute_command
mysql_parse
dispatch_command
do_command
handle_connection
pfs_spawn_thread
start_thread
mysqld
all
ha_innob..
ha_create..
innobase_re..
ha_innobase..
mysql_rena..
row_dro..
ha_inno..
ha_dele..
quick_r..
mysql_alter_table
Sql_cmd_alter_..
mysql_execute_command
sp_instr_stmt::exec_core
sp_lex_instr::reset_lex..
sp_lex_instr::validate_l..
sp_instr_stmt::execute
sp_head::execute
sp_head::execute_proc..
ut_delay
rw_lock_s_..
row_purge_..
que_run_th..
srv_worker_..
srv_purg..

Cold Graphs

What are ColdGraphs / Off-CPU analysis

Off-CPU analysis is a performance methodology where off-CPU time is measured and studied, along with context such as stack traces.

http://www.brendangregg.com/offcpuanalysis.html


User
running
(on-CPU)
mode
switch
preemption or
time quantum expired
Kernel
schedule
Runnable
swap out
Swapping
swap in
disk I/O
Disk
wakeup
network I/O
Sleep
wakeup
off-CPU
sleep
block
Lock
acquire
wait
for work
Idle
work
arrives

First things First

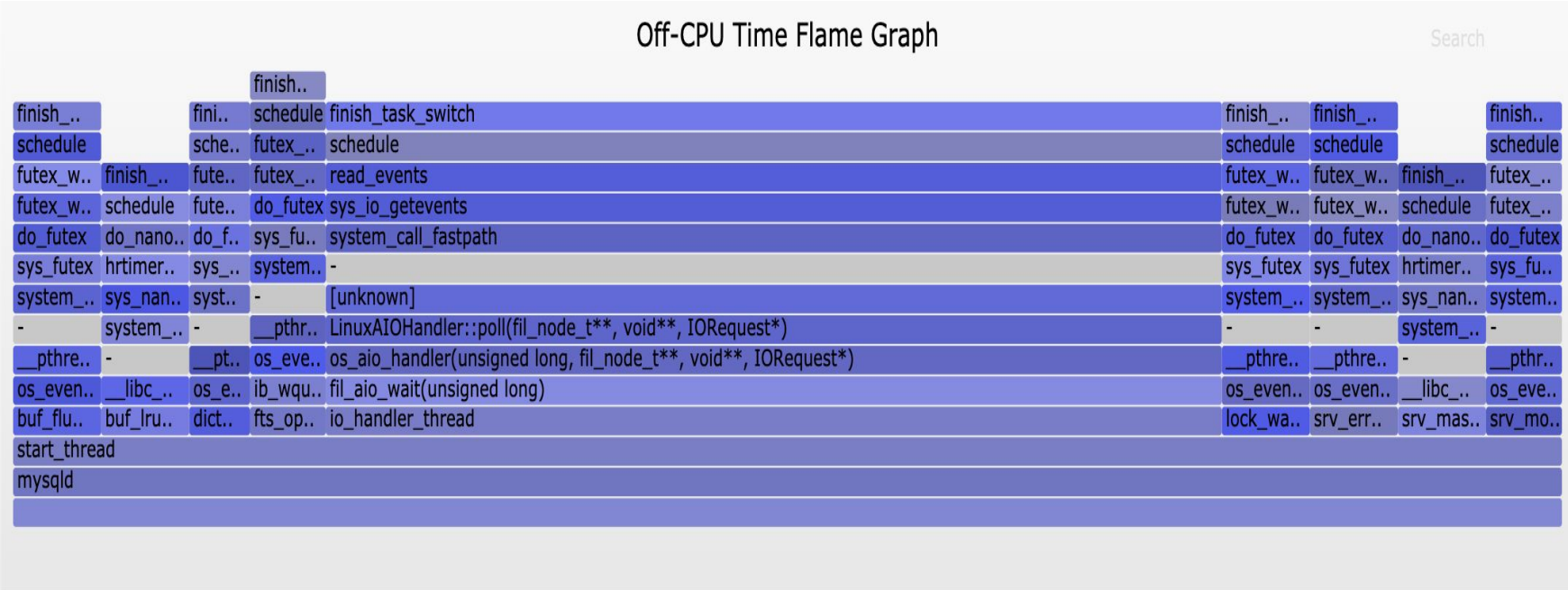
```
# RHEL
yum install bcc-tools

# Ubuntu
sudo apt-get install bpfcc-tools linux-headers-$(uname -r)

# Validating
ls -larth /usr/share/bcc/tools/
```

Profiling

```
# /usr/share/bcc/tools/offcputime -df -p `pgrep -nx mysqld` 30 > out.stacks
# git clone https://github.com/brendangregg/FlameGraph
# cd FlameGraph
# ./flamegraph.pl --color=io --title="Off-CPU Time Flame Graph" --countname=us < out.stacks > out.svg
```


Off-CPU Time Flame Graph
Search
finish..
finish_.. fini.. schedule finish_task_switch finish_.. finish_.. finish..
schedule sche.. futex_.. schedule schedule schedule schedule
futex_w.. finish_.. fute.. futex_.. read_events futex_w.. futex_w.. finish_.. futex_..
futex_w.. schedule fute.. do_futex sys_io_getevents futex_w.. futex_w.. schedule futex_..
do_futex do_nano.. do_f.. sys_fu.. system_call_fastpath do_futex do_futex do_nano.. do_futex
sys_futex hrtimer.. sys_.. system.. - sys_futex sys_futex hrtimer.. sys_fu..
system_.. sys_nan.. syst.. - [unknown] system_.. system_.. sys_nan.. system..
- system_.. - __pthr.. LinuxAIOHandler::poll(fil_node_t**, void**, IORequest*) - - system_.. -
__pthre.. - __pt.. os_eve.. os_aio_handler(unsigned long, fil_node_t**, void**, IORequest*) __pthre.. __pthre.. - __pthr..
os_even.. __libc_.. os_e.. ib_wqu.. fil_aio_wait(unsigned long) os_even.. os_even.. __libc_.. os_eve..
buf_flu.. buf_lru.. dict.. fts_op.. io_handler_thread lock_wa.. srv_err.. srv_mas.. srv_mo..
start_thread
mysqld

PERCONA

PERCONA LIVE 2020

MAY 18 20
AUSTIN, TEXAS

Percona Live is the one and only event where all of the open source database solution companies come together with the community

MySQL, Mongo, Postgres, Elastic, Redis and more
Percona Live brings them to you.

- 3 Days
- Hands-on tutorials,
- Breakout sessions,
- Keynote addresses,
- Expo Hall
- Networking
- Lots of Fun!

Use **PRESENTER** for 20% off! Register now at perconalive.com

谢谢

Thank you

Grazie

Obrigado

Gracias

Open Source Database Experts