MySQL

ORACLE

MySQL Performance : Profiling Memory & CPU Usage Directly from MySQL

Frederic Descamps (LeFred)
MySQL Community Manager @Oracle

Dimitri KRAVTCHUK
MySQL Performance Architect @Oracle

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

Agenda

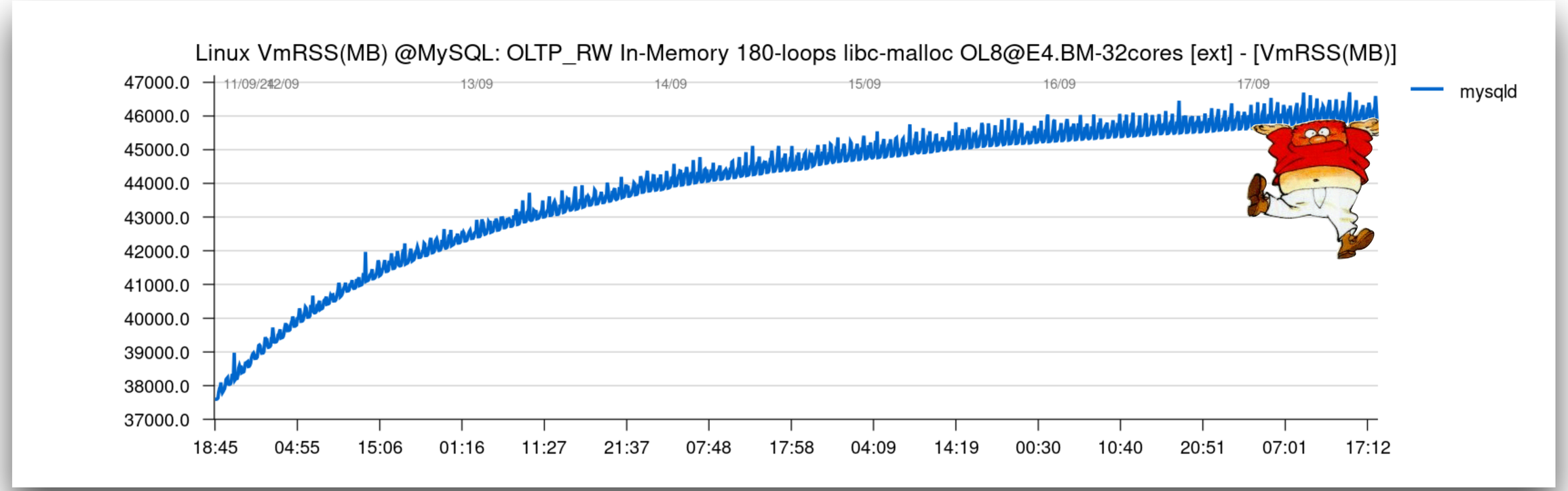
- Malloc Libs Overview
- Live Memory Profiling
- On-demand Memory Usage Profiling from MySQL
- Why not also CPU Profiler ?..
- Q & A

Malloc Libs Historically

- libc-malloc : not designed for MT-apps (not scaling)
- HOARD malloc
- mtmalloc
- umem
- jemalloc (generally preference on Linux)
- tcmalloc (my preferred ;-))
- etc..

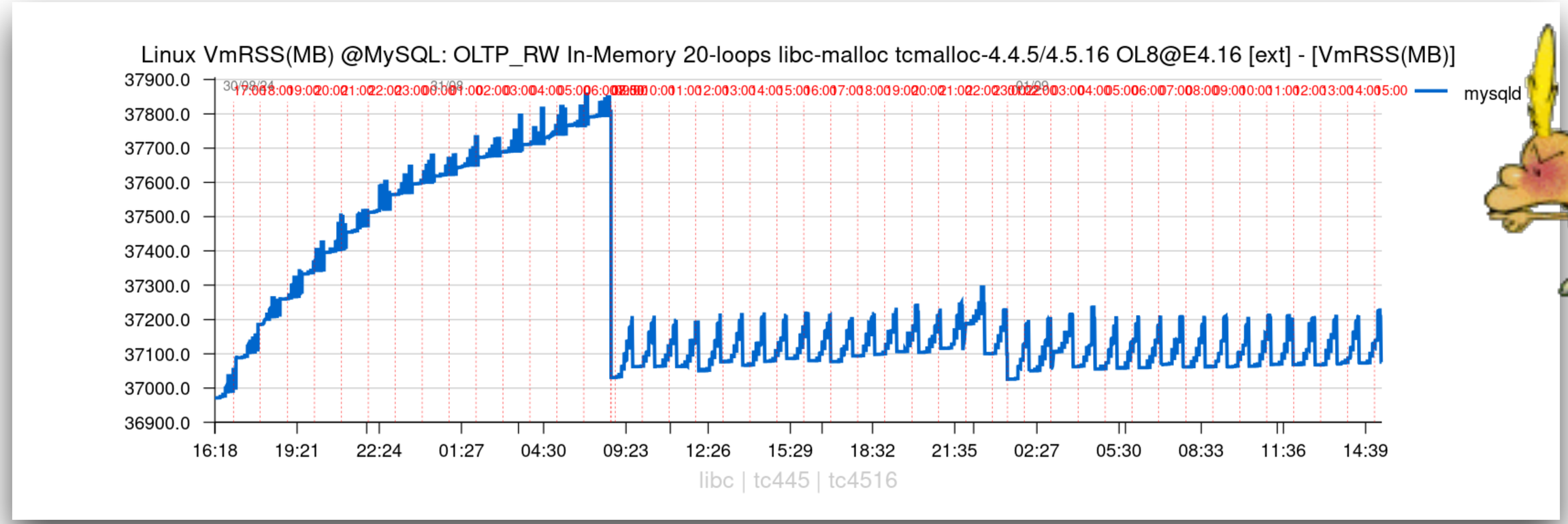
Malloc Libs Today

- glibc-malloc :
  - the latest glibc-malloc can be used with MT-apps today in most cases
  - but memory fragmentation !
- glibc-malloc : +10GB RSS usage over 1 week..



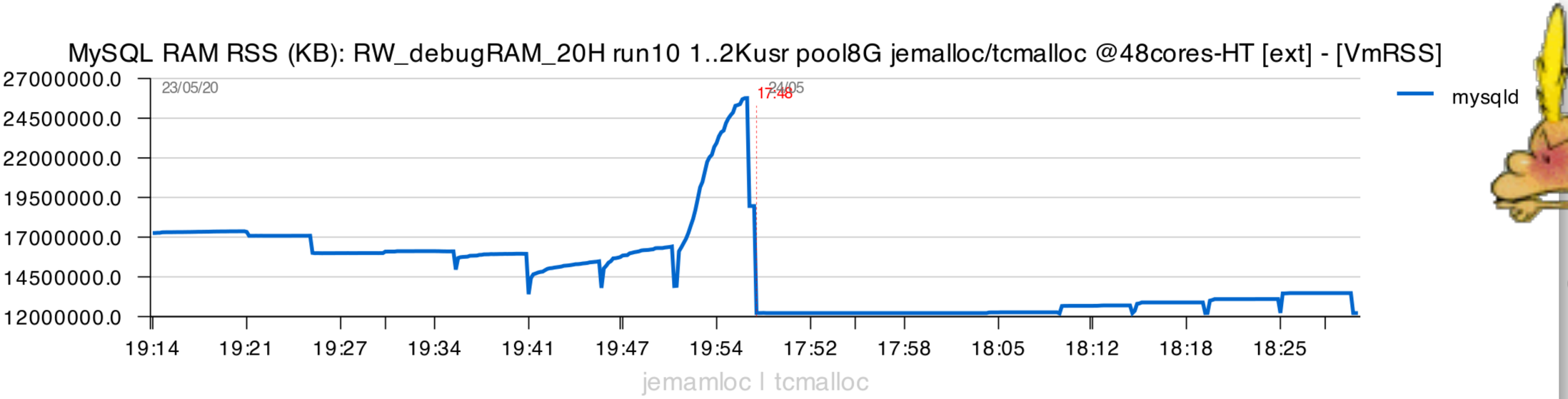
Malloc Libs Today

- glibc-malloc -vs- tcmalloc :



Malloc Libs Today

- jemalloc -vs- tcmalloc :



Memory Leaks in MySQL

- very rare today, but still possible
- cannot be ignored (often ends by OOM-killer)
- can be totally unexpected / difficult to reproduce
- MySQL PFS Memory Instrumentation : the main tool !
- tcmalloc Memory Profiler : great additional help !

tcmalloc Memory Profiler : General Use Case

- set env. variable HEAPPROFILE=/path/to/heap-dumps
- start MySQL Server with tcmalloc in LD_PRELOAD
- follow / analyze dumped heap files
- use “pprof” to print reports / call-stacks in PDF
- NOTE : limited usage possibilities
  - e.g. you need to restart MySQL
  - e.g. you need to have a reproducible test case
  - e.g. isolated debug only

tcmalloc Memory Profiler : Advanced Solution

- start MySQL Server with tcmalloc
  - use LD_PRELOAD / or compile with tcmalloc
- call tcmalloc API to provide /path/to/heap-dumps
- call tcmalloc API to start / stop profiling
- call tcmalloc API to dump collected data
- NOTE : wide flexibility in usage !
  - e.g. can be involved **on-demand** !
  - e.g. can be **auto-triggered** when a problem is detected !
  - e.g. zero overhead while disabled ! => can be deployed in Production !

- **Q** : which way to implement ?
- **A** : as MySQL Component !

tcmalloc Memory Profiler : MySQL Component

- no changes to MySQL code / no need to compile with tcmalloc
- install gperftools pkg on MySQL host system
- setup env. variables on MySQL start :
  - LD_PRELOAD=/path/to/tcmalloc.so
  - HEAP_PROFILE_ALLOCATION_INTERVAL=0 (def: 1GB)
  - HEAP_PROFILE_INUSE_INTERVAL=0 (def: 100MB)

- install / uninstall component on-demand
- start / stop / dump profiling via MySQL connection
- install "pprof" on the MySQL host to allow direct reports
- NOTE : also available for jemalloc as another Component

MySQL Profiler Component : Usage

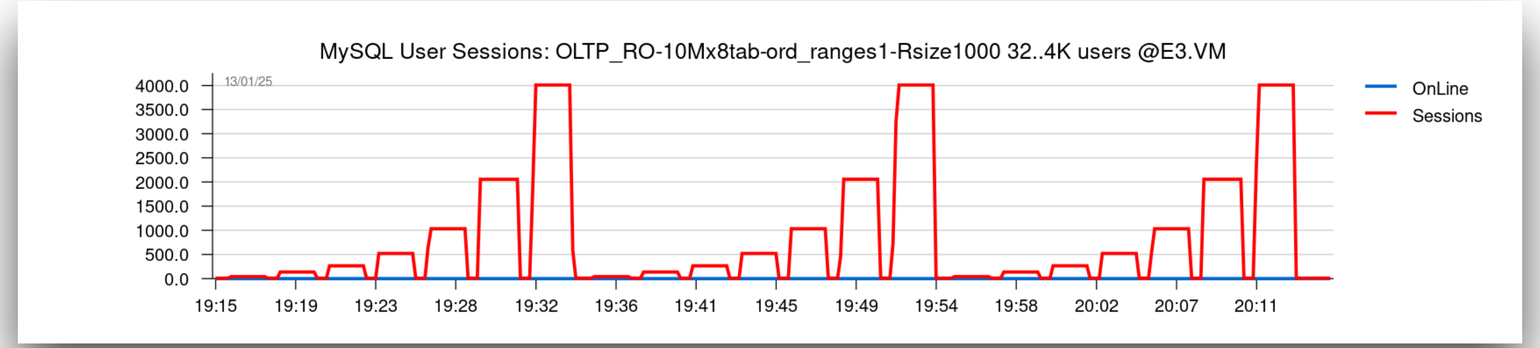
```
mysql> install component 'file://component_profiler';
mysql> install component ‘file://component_profiler_memory';

mysql> select * from mysql.component;
mysql> show global variables like 'profiler%';
mysql> show global status like 'profiler%';

mysql> select profiler_cleanup();
mysql> select memprof_start();                  – we also auto-dump on start
mysql> select memprof_dump();                   -- on-demand dump
mysql> select memprof_stop();                   -- we also auto-dump on stop

mysql> select memprof_report()\G ;              -- report from the last dump
mysql> select memprof_report( 'text', 20, 'dump-name' )\G ;
mysql> select memprof_report( 'dot' )\G ;       -- to build callgraph PDF
```

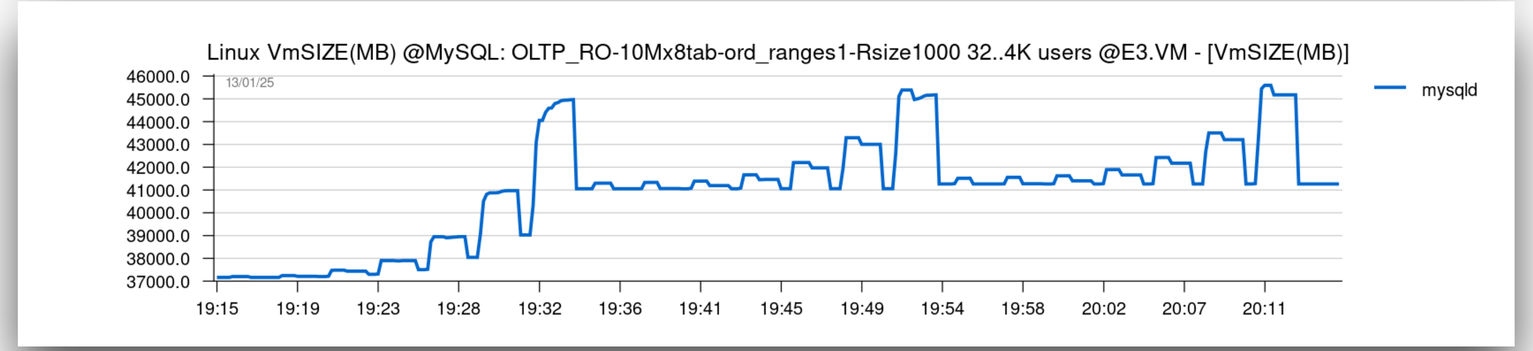
Full details ref. : https://github.com/lefred/mysql-component-profiler

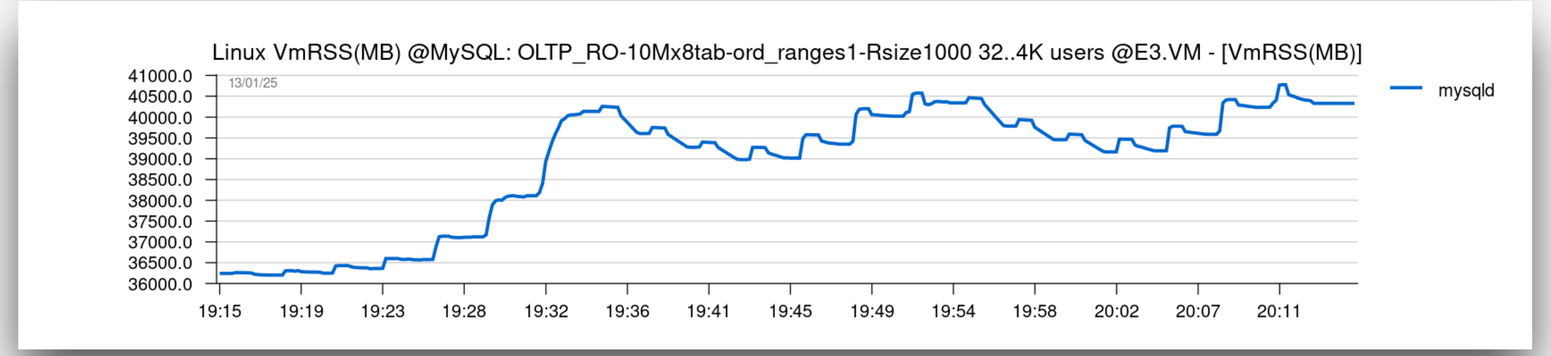
MySQL Profiler Component : Test Case (run x3 times)

```
$ export LD_PRELOAD="/apps/lib/libtcmalloc_and_profiler.so.4.6.11"
$ export HEAP_PROFILE_ALLOCATION_INTERVAL=0
$ export HEAP_PROFILE_INUSE_INTERVAL=0

$ mysql.server start                                        # BP size 32G
$ bash sb11-OLTP_RO_50M_8tab-uniform-notrx.sh 512 300       # prewarm..

for usr in 32 128 256 512 1024 2048 4000
do
  mysql> select profiler_cleanup();
  mysql> select memprof_start( 60 );
  sleep 5

  $ bash sb11-OLTP_RO_10M_8tab-uniform-ord_ranges1-Rsize1000-notrx.sh $usr 120

  mysql> select memprof_report( 'text', 100 )\G ;
  sleep 20
done
```

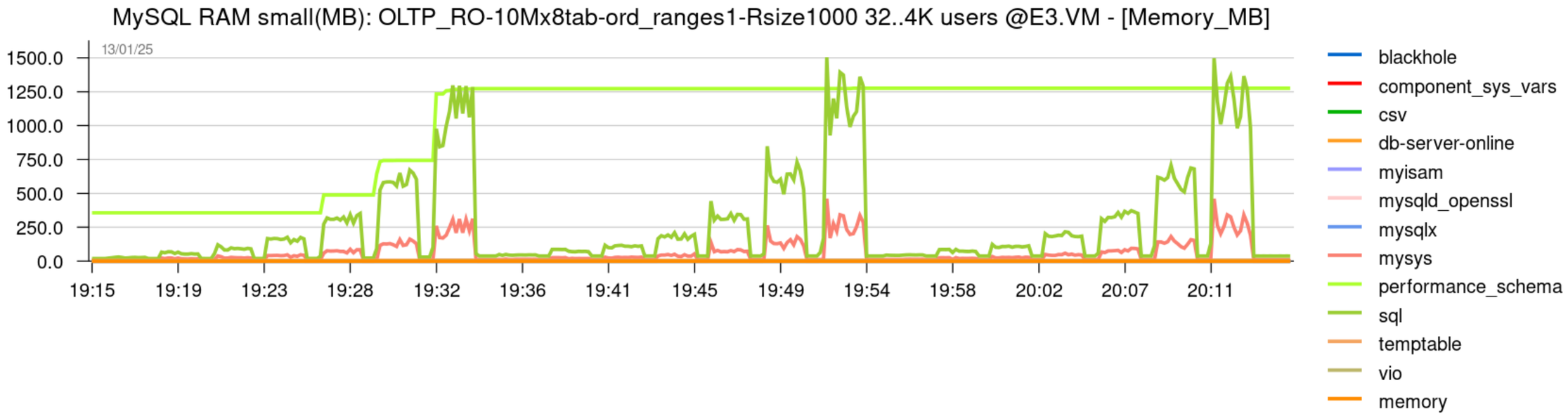
MySQL Profiler Component : Test Case, VmSIZE / RSS



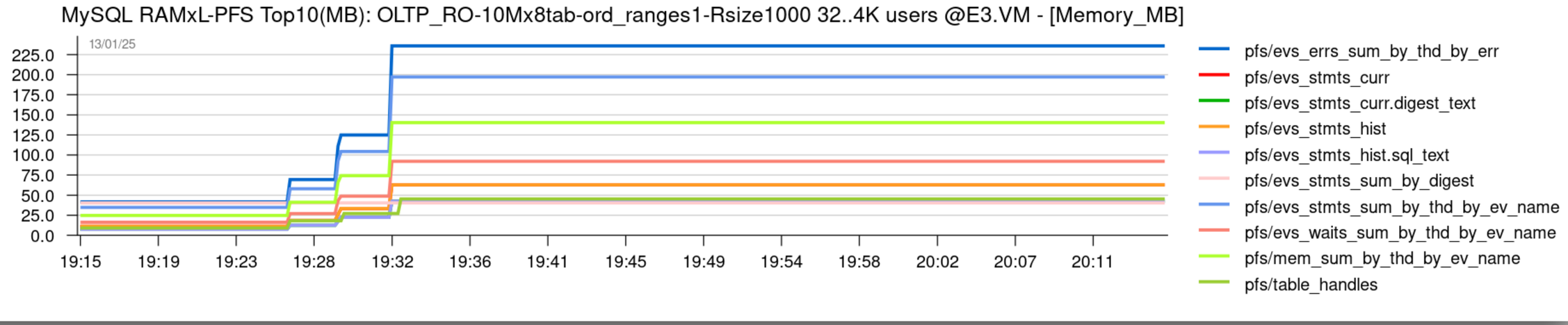


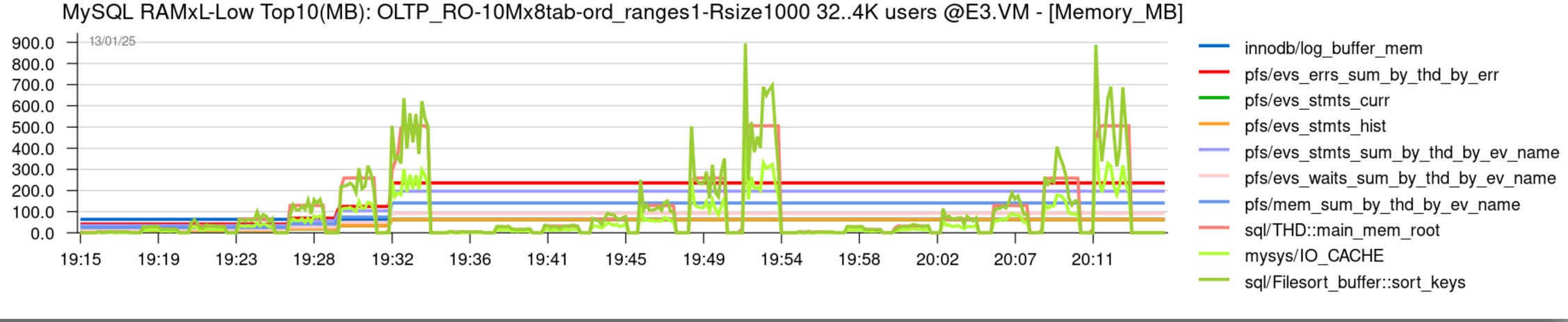


MySQL Profiler Component : Test Case, PFS Report



MySQL Profiler Component : Test Case, PFS Report (2)





MySQL Profiler Component : Test Case, PPROF Report

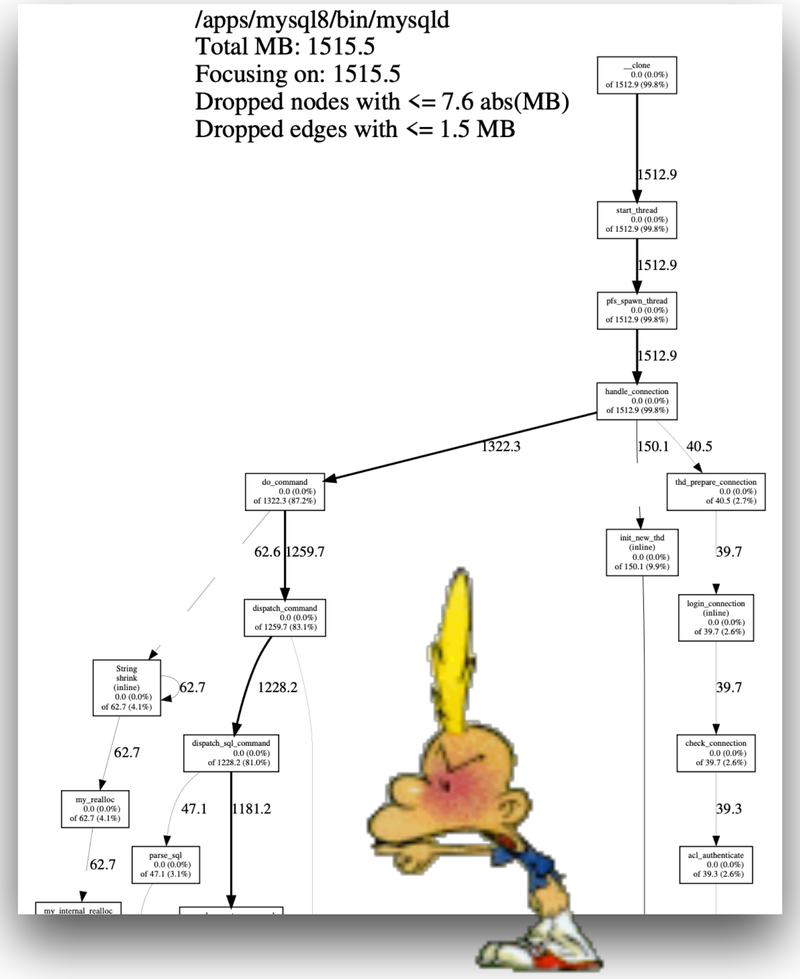
```
memprof_report( 'text', 100 ): Total: 1977.2 MB
  1336.0  67.6%  67.6%   1336.0  67.6% std_allocator (inline)
   515.9  26.1%  93.7%    515.9  26.1% pfs_malloc
    51.7   2.6%  96.3%    152.0   7.7% Channel_info::create_thd
    31.5   1.6%  97.9%     31.5   1.6% dd::cache::Element_map::Element_map (inline)
...

memprof_report( 'text', 100 ): Total: 1508.6 MB
  1391.5  92.2%  92.2%   1391.5  92.2% std_allocator (inline)
    51.7   3.4%  95.7%    150.1  10.0% Channel_info::create_thd
    30.9   2.1%  97.7%     30.9   2.1% dd::cache::Element_map::Element_map (inline)
...

memprof_report( 'text', 100 ): Total: 1515.5 MB
  1398.7  92.3%  92.3%   1398.7  92.3% std_allocator (inline)
    51.7   3.4%  95.7%    150.1   9.9% Channel_info::create_thd
    30.9   2.0%  97.7%     30.9   2.0% dd::cache::Element_map::Element_map (inline)
    12.5   0.8%  98.6%     31.6   2.1% THD::THD
...
```

MySQL Profiler Component : Test Case, PPROF Report (2)

```
  memprof_report( 'text', 100 ): Total: 1515.5 MB
    1398.7  92.3%  92.3%   1398.7  92.3% std_allocator (inline)
      51.7   3.4%  95.7%    150.1   9.9% Channel_info::create_thd
      30.9   2.0%  97.7%     30.9   2.0% dd::cache::Element_map::Element_map (inline)
      12.5   0.8%  98.6%     31.6   2.1% THD::THD
  ...

       0.0   0.0% 100.0%    481.6  31.8% Filesort_buffer::allocate_sized_block
       0.0   0.0% 100.0%      8.3   0.6% Filesort_buffer::commit_used_memory (inline)
       0.0   0.0% 100.0%    219.0  14.5% Filesort_buffer::get_contiguous_buffer
       0.0   0.0% 100.0%    262.6  17.3% Filesort_buffer::get_next_record_pointer
(inline)
  ...

       0.0   0.0% 100.0%    508.0  33.5% MEM_ROOT::Alloc (inline)
       0.0   0.0% 100.0%    508.0  33.5% MEM_ROOT::AllocBlock
       0.0   0.0% 100.0%    508.0  33.5% MEM_ROOT::AllocSlow
  ...
```

MySQL Profiler Component : Recap..

- (+) access directly from MySQL connection
- (+) on-demand install / uninstall
- (+) on-demand start / stop profiling & reports
- (+) small footprint in dumps
- (+) zero impact while profiling=OFF
- (+) cool to understand your RAM allocations
- (-) **huge overhead** when profiling=ON !!
  - rather consider to see a partial stall..
  - avoid long use in Production
  - memprof_start(Nsec) <= most safe use



- NOTE : be thankful to MySQL PFS memory instrumentation ! ;-))
  - enabled by default in MySQL Server + reasonably low overhead

MySQL Profiler Component : What about CPU ?

- gperf-tools provides tcmalloc & CPU profilers
- you can use “cpuprof” component the same way as “memprof”
- (+) similar advantages as with “memprof”
- (+) **mostly zero overhead** !!
  - comparing to “perf” overhead which can be pretty important !

MySQL Profiler Component : “perf” -vs- “cpuprof”

- probably makes sense to consider both ;-))

```
vm32_e3
Samples: 192K of event 'cycles', 99 Hz, Event count (approx.): 383999112410 lost: 0/0 drop: 0/21649 [z]
Overhead  Shared Object                      Symbol
   7.33%  libc-2.17.so                       [.] __memcpy_ssse3
   3.73%  mysqld-8403-ssl111L                [.] my_strnxfrm_uca_900_tmpl<Mb_wc_utf8mb4, 1>
   2.92%  mysqld-8403-ssl111L                [.] row_search_mvcc
   1.89%  mysqld-8403-ssl111L                [.] my_lengthsp_8bit
   1.83%  mysqld-8403-ssl111L                [.] Sort_param::make_sortkey
   1.80%  mysqld-8403-ssl111L                [.] ha_innobase::general_fetch
   1.38%  libc-2.17.so                       [.] __memcmp_sse4_1
   1.29%  mysqld-8403-ssl111L                [.] std::__introsort_loop<__gnu_cxx::__normal_iterator<u
   1.27%  mysqld-8403-ssl111L                [.] rec_init_offsets_comp_ordinary
   1.22%  mysqld-8403-ssl111L                [.] my_sql_parser_parse
   1.20%  mysqld-8403-ssl111L                [.] cmp_dtuple_rec_with_match_low
   1.12%  mysqld-8403-ssl111L                [.] page_cur_search_with_match
   1.11%  mysqld-8403-ssl111L                [.] buf_page_hash_get_low
   1.10%  [kernel]                           [k] apic_timer_interrupt
   1.09%  libc-2.17.so                       [.] __GI_memset
   0.94%  mysqld-8403-ssl111L                [.] Item_func_between::val_int
```

```
vm32_e3
cpuprof_report(): Total: 57351 samples
    4066   7.1%   7.1%     4189   7.3% __libc_send
    3684   6.4%  13.5%     3684   6.4% __memcpy_ssse3
    2371   4.1%  17.6%     2390   4.2% __libc_recv
    1331   2.3%  20.0%     2670   4.7% my_sql_parser_parse
    1278   2.2%  22.2%     2093   3.6% for_each_weight (inline)
    1178   2.1%  24.3%     1178   2.1% my_lengthsp_8bit (inline)
     892   1.6%  25.8%     8354  14.6% ha_innobase::general_fetch
     880   1.5%  27.3%      880   1.5% __memcmp_sse4_1
     878   1.5%  28.9%     8203  14.3% row_search_mvcc
     697   1.2%  30.1%      697   1.2% __memset_sse2
     667   1.2%  31.2%     1418   2.5% cmp_varlen_keys (inline)
     635   1.1%  32.4%      873   1.5% rec_init_offsets_comp_ordinary
     573   1.0%  33.4%     4299   7.5% memcpy (inline)
     521   0.9%  34.3%     5764  10.1% Sort_param::make_sortkey
     378   0.7%  34.9%      531   0.9% cmp_dtuple_rec_with_match_low
     371   0.6%  35.6%      714   1.2% Item_func_between::val_int
     370   0.6%  36.2%      954   1.7% row_sel_field_store_in_mysql_format_func
```

One more thing.. ;-))

- LeFred
  - https://github.com/lefred/mysql-component-profiler - MySQL Profiler Component repo
  - https://lefred.be - LeFred's Blog Site

- DimitriK
  - http://dimitrik.free.fr - dim_STAT (live monitoring tool, as ex.: graphs for this talk, etc.)
  - http://dimitrik.free.fr/blog - Articles about MySQL Performance, etc.
  - http://dimitrik.free.fr/blog/posts/mysql-perf-bmk-kit.html - Benchmark Kit (aka BMK-kit)

Thank You !!!

ORACLE