Feed me more: MySQL Memory analysed

FOSDEM MySQL Devroom 2013

Raghavendra Prabhu
raghavendra.prabhu@percona.com

Percona

3rd Feb, 2013

Outline

Overview

- During capacity planning, a frequent question which pops up is "How much memory should I allocate for MySQL and for the system in general?".
- Memory is quite underestimated much like everything else but more so than others.
- Umpteen thumb rules about this on the Internet, often overruns the memory and/or frequently OOMs.

Bufferbloat

- Buffers
  - read_buffer, read_rnd_buffer, join_buffer_size, sort_buffer_size

```
cache->length=length+blobs*sizeof(char*);
cache->blobs=blobs;
*blob_ptr=0;
size=max(thd->variables.join_buff_size, cache->length);
```

  - MySQL variables - max_connections, table_cache and open_files_limit

```
wanted_files=10+max_connections+table_cache_size*2;
max_open_files=max(max(wanted_files, max_connections*5),open_files_limit)
```

  - tmp_table_size and max_heap_size on tmpfs
  - tmpdir and tmpfs

InnoDB

- Buffer pool
  - Bounded
  - Not initialized at startup unless done so (Twitter, XtraDB)
  - O_DIRECT, ALL_O_DIRECT
- Adaptive Hash Index
  - Role, Bounded, LRU/Reclaim and disable/enable
  - Multiple partitions - implications on locking
  - Hash tables & Internals: mem_heap_create_in_btr_search, MEM_HEAP_BTR_SEARCH, MEM_HEAP_BUFFER, MEM_HEAP_DYNAMIC
- Data dictionary
  - Unbounded
  - XtraDB variable - innodb_dict_size_limit
  - Other ways
- Change buffer
  - XtraDB variable to limit this / Why to limit

PERCONA
Performance Consulting Experts

OS

- Sysctls
  - vm.swappiness, vm.vfs_cache_pressure, vm.drop_caches
  - vm.overcommit_memory, vm.zone_reclaim_mode
- NUMA
  - NUMA policies: DEFAULT, BIND, INTERLEAVE, PREFERRED
  - Numastat: numa_hit, numa_miss, numa_foreign, interleave_hit, local_node, other_node; lp:1083488
  - numad and AutoNUMA

Instrumentation/Tools

- Instrumentation
  - getrusage - RUSAGE_THREAD, RUSAGE_SELF, RUSAGE_CHILDREN
  - Malloc
    - tcmalloc - also supports profiling
    - jemalloc and e/glibc malloc
    - additional_mem_size and malloc, use_sys_malloc
  - Perf
    - perf-top, perf-list
    - perf-record, perf-report
    - perf-timechart
- Tools
  - smem, atop, page-types - PSS, stall/pag, memory-map
  - Valgrind - massif, memcheck, callgrind - UNIV_DEBUG_VALGRIND
  - LLVM/Clang - ASAN, MSAN - CTI v/s DBI

Code

- Hinting

  Not with O_DIRECT handles, Users: Galera, Xtrabackup, MySQL
  - Madvise and Fadvise
    - MADV_DONTDUMP
    - MADV_WILLNEED, MADV_DONTNEED, MADV_SEQUENTIAL
    - Caveats
  - Fincore

```
sudo linux-fincore /data/mysql/galera.cache
filename
----
/data/mysql/galera.cache
--
total cached size: 15,944,663,040
```

  - Don't mix O_DIRECT and others

OOM and Cgroups

- OOM adjustments
  - vm.panic_on_oom, oom_kill_allocating_task, oom_score_adj
- Memory Cgroups
  - Multiple instances and/or shared
  - Per instance settings like swappiness
  - Overhead?
  - OOM: Proactive and Reactive measures
    - Proactive: Turn down the load
    - Reactive: Kill! or move

Virtualization

- Memory ballooning
  - You balloon the memory whenever required
  - Qemu monitor: info balloon
  - balloon virtio
- KSM with KVM/Xen

Misc

- Hugepages huge pages
  - Supported in mysql
  - Boot-time allocation better
  - Fragmentation and performance
- Transparent Huge pages
  - No need for setup
  - Possible bugs? - compaction_alloc and compact_zone
  - Disable the defrag

PERCONA
Performance Consulting Experts

Cases

- NUMA and swap
- mem_cgroup_del_lru_list and watermark

```
19.08% [kernel] [k] mem_cgroup_del_lru_list
14.52% [kernel] [k] intel_idle
7.98% [kernel] [k] __isolate_lru_page
6.14% [kernel] [k] shrink_inactive_list
3.83% [kernel] [k] mem_cgroup_add_lru_list
3.60% mysqld [.] 0x4c584e
3.14% [kernel] [k] page_waitqueue
3.09% [kernel] [k] isolate_pages_global
```

PERCONA
Performance Consulting Experts

Examples

- smem stack

```
smem -m -t -k -P aurora
Map                 PIDs   AVGPSS      PSS
[stack:34760]                    1     1.8M     1.8M
[stack:34668]                    1     2.0M     2.0M
[stack:34694]                    1     2.2M     2.2M
[heap]                           1     6.4M     6.4M
[stack:34746]                    1     9.2M     9.2M
[stack:35015]                    1    19.5M    19.5M
/usr/lib/aurora/libxul.so        1    24.4M    24.4M
<anonymous>                      2   337.8M   675.7M
```

PERCONA
Performance Consulting Experts

Examples (contd.)

- Valgrind

```
./mysql-test-run.pl -valgrind
-valgrind-option="-suppressions=$PWD/valgrind.supp"
-valgrind-option='-show-reachable=yes'
-valgrind-option='-gen-suppressions=all'
-vardir=$HOME/mysql t/fake.test

=13145== 16,384 bytes in 1 blocks are still reachable in loss record 738 o
=13145==  at 0x4C2C1DE: realloc (in /usr/lib/valgrind/vgpreload_memcheck-a
=13145==  by 0x5D78B60: CRYPTO_realloc (in /usr/lib/libcrypto.so.1.0.0)
=13145==  by 0x5E20E31: lh_insert (in /usr/lib/libcrypto.so.1.0.0)
=13145==  by 0x5E23E0D: int_err_set_item (in /usr/lib/libcrypto.so.1.0.0)
=13145==  by 0x5E24458: ERR_load_strings (in /usr/lib/libcrypto.so.1.0.0)
=13145==  by 0x5AF99BD: ERR_load_SSL_strings (in /usr/lib/libssl.so.1.0.0)
=13145==  by 0xA3C315: new_VioSSLFd (viosslfactories.c:159)
=13145==  by 0xA3C91E: new_VioSSLAcceptorFd (viosslfactories.c:288)
=13145==  by 0x51D509: mysqld_main(int, char**) (mysqld.cc:3735)
=13145==  by 0x513974: main (main.cc:25)
```

Examples (contd.)

- Perf-report

```
#   7.15%  mysqld  mysqld                [.] 0x11b0fe
   - 0x959389
      - 91.87% yaSSL::SSL::makeMasterSecret()
         - 79.56% yaSSL::CertManager::sendVerify() const
              page_cur_parse_insert_rec.clone.0
              row_search_for_mysql
              row_search_for_mysql
            - row_search_for_mysql
               - 61.21% row_vers_impl_x_locked_off_kernel
                    Create_func_from_unixtime::create_native(THD*, st_mysq
                    Create_func_is_used_lock::create(THD*, Item*)
                    Item_func_set_user_var::fix_length_and_dec()
                    Item_func::fix_fields(THD*, Item**)
                    handler::delete_table(char const*)
                    Item_copy_string::val_int()
                    Item_field::val_bool_result()
                    Item_cache_datetime::val_str(String*)
                    Item_type_holder::make_field_by_type(TABLE*)
                    Item_param::set_param_type_and_swap_value(Item_param*)
```

MySQL 5.6

- Multi thread purge
- Data dictionary LRU
- Malloc
  - Grouping allocations
  - Stack instead of heap
  - Removing allocations
- Page size - XtraDB 5.5 and MySQL 5.6

Bugs/Features

- Bugs reported / in-review
  - Per thread variables: http://dev.mysql.com/worklog/task/?id=681
  - Fadvise relay logs: https://bugs.launchpad.net/percona-server/+bug/1073170
  - LRU and AHI: https://bugs.launchpad.net/percona-server/+bug/1083536
  - Data Dictionary and Buffer pool: https://bugs.launchpad.net/percona-server/+bug/1083514
  - MADV_DONTDUMP: https://bugs.launchpad.net/percona-server/+bug/1092645
  - SSL leak: https://bugs.launchpad.net/percona-server/+bug/1049076
  - Leak on a filtered slave: https://bugs.launchpad.net/percona-server/+bug/1042946
- Features planned/interest

Appendix I

- Insert call-chain

```
btr_search_build_page_hash_index
btr_search_update_hash_on_insert
btr_search_update_hash_ref
ha_insert_for_fold
mem_heap_alloc(hash_get_heap(table, fold), sizeof(ha_node_t))
mem_heap_add_block
mem_heap_create_block
buf_block_alloc
```

Appendix II

- Hash table

```
struct hash_table_struct {
        ulint           n_cells;
        hash_cell_t*    array;
        ulint           n_mutexes;
        mutex_t*        mutexes;
        mem_heap_t**    heaps;
        mem_heap_t*     heap;
};
```

Appendix III

- Heap structure

```
struct mem_block_info_struct {
        ulint   magic_n;
        char    file_name[8];
        ulint   line;
        UT_LIST_BASE_NODE_T(mem_block_t) base;
        UT_LIST_NODE_T(mem_block_t) list;
        ulint   len;
        ulint   total_size;
        ulint   type;
        ulint   free;
        ulint   start;
        void*   free_block;
        void*   buf_block;
};
```