



Percona

Unbiased Open Source Database Experts



The low hanging fruits of MySQL



Who Am I?

Kedar Vaijanapurkar

- Living in Vadodara, Gujarat, India
- Major components: SPOF (wife) and two HA (highly active) replicas.
- Educated Masters in Computer Applications (2006)
- Eventual DBA and loving it (VB/VC/C# >> Perl/PHP >> MySQL)
- Enjoys music, photography, chess, cricket, [blogging](#) and knowledge sharing
- Knows a bit about MySQL ecosystem, still learning.
- Proud Perconian



Agenda

- Low hanging fruit
- Before you pluck
- Identifying LHFM and quick tips-tricks
- Conclusions



Low hanging fruit



Low hanging fruit

- Easy to achieve
- Common knowledge
- Proven to be successful
- No brainer / Copy-paste solution / little to no sweat



Before plucking

- It depends, it always does.
- Nothing is low unless you know.
- Production is not for trial and error.

Identifying Low Hanging Fruits



Where are the fruits

- Configuration
 - Database
 - System
- Database Design
- Queries
- Replication
- Backups
- Tools
- Clouds

MySQL - Configuration



Configuration (DB)

- `Innodb_buffer_pool_size`: Memory space holding in-memory data structures of InnoDB, buffers, caches, indexes and even row-data.
 - One man army, carries the load single handedly.
 - 75-80% of total memory
 - Dedicated database server?
- `innodb_buffer_pool_instances`: Defines number of instances / chunks of the innodb bufferpool
 - Helps reducing contention for concurrent bufferpool access
 - One instance upto 1 to 2G
 - “SHOW ENGINE INNODB STATUS” and look for Mutex on buf0buf.c
 - [Refer](#)

Configuration (DB)

- `InnoDB_flush_method`: The way it open files / logs.
 - The default `f_sync()` is slow on some, use `O_DIRECT` instead.
 - Bypass page cache and perform any IO operations directly against storage.
 - [Refer](#)
- `InnoDB_log_file_size`: The redo log space
 - Helps reducing IO in write-intensive workloads
 - Flushes dirty pages in the buffer pool at least once per log file cycle.
 - Confused by documentation? [Larger log files also make crash recovery slower VS Make your redo log files big, even as big as the buffer pool.](#)
 - I'd not worry and raise to the largest 1 hour worth traffic.
 - [Refer](#)

Configuration (DB)

- `InnoDB_log_buffer_size`: Buffer that writes to log file
 - Make it large enough to your largest transaction.
- `innodb_read_io_threads` and `innodb_write_io_threads`: Number of BG threads to handle IO
 - Observe pending read requests in `SHOW ENGINE INNODB STATUS`
 - [Read](#)
- `query_cache_size`: TL;DR - Disable it.
 - It isn't there in MySQL 8.0
- `log_output`: FILE
 - Don't store logs in database.

Configuration (DB)

- `table_open_cache / table_definition_cache`:
 - Opens table for each concurrent access
 - Observe the `Opened_tables` status (`show global status like '%Opened%'`)
 - [Refer](#).

Configuration (System)

- Individual partitions for datadir and binlogs
- Swappiness: Kernel's love towards swapping out memory pages
 - Have swap but don't let it swap (Set it to 1)
 - [Refer](#)
- noatime, dirnoatime: Do we really need access times for MySQL data files?
 - Use them for your MySQL related mounts.
 - [Refer](#)
- IO Schedulers: Most works better with noop
 - benchmark

OOM

- Dear MySQL, I promise you 1000G of memory.
 - Know what you have (free -m)
 - Know what you need (max_connections, traffic pattern)
 - Memory Usage: global-buffers + active-connections (per-thread-buffers)

```
mysql> SELECT ( @@key_buffer_size + @@query_cache_size + @@innodb_buffer_pool_size + @@innodb_log_buffer_size +
1000 * ( @@read_buffer_size + @@read_rnd_buffer_size + @@sort_buffer_size + @@join_buffer_size +
@@binlog_cache_size + @@thread_stack + @@tmp_table_size ) ) / (1024 * 1024 * 1024) AS MAX_MEMORY_GB\G
***** 1. row *****
MAX_MEMORY_GB: 16.9036
```

```
mysql> SELECT ( @@key_buffer_size + @@query_cache_size + @@innodb_buffer_pool_size + @@innodb_log_buffer_size +
500 * ( @@read_buffer_size + @@read_rnd_buffer_size + @@sort_buffer_size + @@join_buffer_size +
@@binlog_cache_size + @@thread_stack + @@tmp_table_size ) ) / (1024 * 1024 * 1024) AS MAX_MEMORY_GB\G
***** 1. row *****
MAX_MEMORY_GB: 8.5265
```


OOM

- Reduce the likeness to get killed: OOMScoreAdj/OOMScore
 - (-1000) => disable OOM Kill
 - (1000) => process more likely to get killed
 - `echo '-800' > /proc/$(pidof mysqld)/oom_score_adj`
 - Add this to “Service”: `OOMScoreAdjust=-800`
- Refer [This](#) and [This](#)

MySQL - DB Design



DB Design

- Use InnoDB tables - it is default since many years!
 - Galera, Aurora - Won't support MyISAM
- Create Primary Keys for your InnoDB tables.
 - Clustered Index
 - If no PK / UK, InnoDB generates internal hidden "GEN_CLUST_INDEX" on "ROW ID" values.
 - Less scalable ([Read](#))
 - VARCHAR, Large PK, UUIDs ([Read](#)), Auto Increment
- Signed and Unsigned columns.
- TEXT is not to store "text" information. Use proper data types.

DB Design

- Secondary indexes
 - Cardinality / Selectivity
 - Don't include PK in secondary index
 - Composite indexes
 - Covering indexes
 - Indexing dates (Range search of an Index won't use the next columns in the multi column index)
 - [Refer](#)

MySQL - Querying



Querying

- You don't always need SELECT *
- SQL is Query Language and not Programming Language (Keep business logics on application)
- SELECT * FROM large_table, another_large_table where true /* proudly written by kedar */;
- Know about [Explain](#), [Explain Extended](#) and [Explain Analyze](#) and use it for your query.
- Running analytical queries? We have MAX_EXECUTION_TIME.
 - `SELECT /*+ MAX_EXECUTION_TIME(300000) */ * FROM t1 INNER JOIN t2 ...`
- Avoid having string comparison that starts with wildcards (%)
 - `SELECT ... WHERE name like '%edar%';`
- Avoid wrapping columns with functions
 - `SELECT ... WHERE ... LOWER(name) = "kedar"`

MySQL - Replication



Replication

- Replication is not backup.
- Don't write on both the nodes in multi master replication.
- `slave_rows_search_algorithms = INDEX_SCAN,HASH_SCAN` (Not to worry >8.0)
- Keep your replica equally provisioned as primary.
- Parallel replication
 - `Binlog_transaction_dependency_tracking: COMMIT_ORDER vs WRITESET`
 - [Read](#)
- Delayed replicas
- Enable binary logs and `log-slave-updates` on replica.

MySQL - Backups



Backups

- Avoid backups on Primary.
 - Performance impact
- Monitor backups for success / failure
 - Failed backups should be identified immediately
- Restore your backups and validate.
 - If you can't restore as needed, the backup is good for nothing.
- Ensure remote retention.
 - Amazon buckets or Google cloud storage

Backups

- Verify your restores against RTO and RPO.
 - How fast you need to recover from disaster
 - How much of the data loss is acceptable
- [Watch](#)

Backups

- Mysqldump
 - Slow to backup and restore
 - Single table restore is somewhat easier
 - Version independent (mostly)
 - --single-transaction / --master-data
- Xtrabackup
 - Fast, Physical
 - Version dependent
 - Corruption is also backed-up.
- Mydumper/myloader
 - Logical backup
 - Multi threaded

MySQL - Tools



Tools

- Percona tools make life easy
 - pt-online-schema-change: Online alter operation
 - pt-kill: Kill the threads matching criteria
 - pt-table-checksum/pt-table-sync: Identify discrepancies and sync
 - pt-summary/pt-mysql-summary: Collect system and database information
 - pt-query-digest: Generate slow query digest for further analysis
 - pt-duplicate-key-checker: Identify duplicate indexes
 - `wget percona.com/get/TOOL`
 - [Read](#)

Tools

- PMM - Percona Monitoring and Management
 - WYSITYF - When you see it then you fix
 - Monitoring, Trend analysis and debugging
 - Not only MySQL but also MongoDB, ProxySQL, Percona Xtradb Cluster, AWS (RDS/Aurora), Postgres
 - [Explore](#)
 - [Get PMM](#)
 - [Watch](#)

MySQL - Cloud



Cloud (AWS)

- Let's talk about AWS today
 - Don't leave it to the default parameter group
 - Multi-AZ
 - Auto failover / HA
 - Backups / Performance
 - Don't store logs in table
 - No MyISAM in Aurora
 - Migrate fast using xtrabackup restores
 - Enable Deletion Protection and Encryption
 - Enable performance insights
 - You can [backtrack](#) in Aurora (undo deletes?)

Conclusion



Conclusion

Low hanging fruit

- Easy to achieve
- Common knowledge
- Proven to be successful
- No brainer / Copy-paste solution / little to no sweat



Knowledge is knowing that a tomato is a fruit. Wisdom is knowing not to put it in a fruit salad.

- Keep in mind the points we covered while reviewing mysql systems
- Test before implementing on Production
- For any questions reach out to
 - [me](#)
 - [Percona Forum](#)
 - [Percona Blog](#)



Are you **passionate**
about Open Source?!

We're looking for you!
Join us!



#RemoteWork

APPLY NOW: percona.com/careers

Thank you!

