

# Top 10 MySQL 5.7 variables

Shiv Iyer



**MinervaDB**

# Safe Harbour Statement

This presentation is intended to outline our technology focus and research interests of MinervaDB. 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 tools used, best practices, data ops. checklists and all research guidance directions remains at the sole discretion of MinervaDB .



**MinervaDB**

# Configure MySQL for performance, scalability and reliability

Best practices for configuring my.CNF :

- Change only one system variable at a time in my.cnf, This helps in estimating and quantifying the benefits of the change .
- Never accept duplicate settings in my.cnf, We recommend using version control to track the system variable changes in my.cnf and always retain the copy / backup of my.cnf for restore / recovery .
- Most of the system variables can be changed at runtime with SET GLOBAL, It really good because we can rollback if the change is causing the new problem but never forget to note down the changes which you need to add in the my.cnf before next MySQL restart (*we often see this mistake happening*) .
- Confirm MySQL system variables are added after [mysqld] in my.cnf .
- Never copy my.cf of another server presuming optimal performance, If you have simple queries in MySQL then there is no need to increase sort\_buffer\_size even if you have 512 GB RAM .
- **We seriously care about your MySQL reliability above performance so we always recommend setting sync\_binlog=1 (presuming binary logging enabled) and innodb\_flush\_log\_at\_trx\_commit=1**



**MinervaDB**

MySQL 5.7 System Variables we consider changing immediately after successful installation ...



**MinervaDB**

The following below are 4 system variables / my.cnf values I consider changing immediately after successful MySQL installation for performance and reliability :

- **innodb\_buffer\_pool\_size** - InnoDB buffer pool caches both data and indexes so always consider setting this value between 50%-75% of available RAM, Does not need to be larger than the database .
- **innodb\_flush\_log\_at\_trx\_commit** - set this value to "1" (*default*) for maximum reliability and "0" or "2" for less reliability and more performance (**we don't recommend this setting**)
- **innodb\_log\_file\_size** - larger the value less checkpoint flush activity required in buffer pool (this also makes crash recovery slower) so there will be less disk I/O and benefit in the performance . If your application is write intensive you can set this value to "1 GB" (I don't prefer setting this value too high !!)
- **innodb\_flush\_method** - You can set this value to "O\_DIRECT" to avoid double buffering .



**MinervaDB**

# Advanced my.cnf configuration in MySQL 5.7

- **innodb\_io\_capacity** and **innodb\_io\_capacity\_max** - consider setting these two system variables only if you have a lots of writes all the time (it does not apply for reads / SELECTS) and use solid-state disks (SSD), We recommend you to benchmark disk throughput (I use sysbench) before setting this value . If you have a server with 1 SSD we recommend innodb\_io\_capacity "4000" and innodb\_io\_capacity\_max "8000" .
- **InnoDB\_autoinc\_lock\_mode** - Setting innodb\_autoinc\_lock\_mode "2" (interleaved mode) improve the performance of MySQL by not depending on table-level AUTO-INC lock, This makes multi-row insert statements with auto-increment primary key more optimal, This requires binlog\_format=ROW or MIXED (and ROW is the default in MySQL 5.7). There may be gaps in the auto-increment values generated by "bulk inserts" , but only if there are concurrently executing "INSERT - like" statements



**MinervaDB**

# MySQL 5.7 has much better default values

- **innodb\_buffer\_pool\_dump\_at\_shutdown** and **innodb\_buffer\_pool\_load\_startup** are enabled by default - This records the pages cached in the InnoDB buffer pool when MySQL server is shutdown and shorten the warm-up process at the next restart .
- **Innodb\_file\_per\_table** = ON (DEFAULT)
- **Innodb\_stats\_on\_metadata** = OFF (DEFAULT) , so statistics for queries like SHOW TABLE STATUS OR INFORMATION\_SCHEMA.TABLES are not updated,This improves performance and stability of execution plans for queries that involve InnoDB tables .
- **QUERY\_CACHE\_TYPE** = 0 (DEFAULT) , so it does not acquire the query cache mutex at all and there is reduced overhead in query execution .



**MinervaDB**

# We use `slow_query_log` and `pt-query-digest` for tracing MySQL

We can dynamically enable `slow_query_log` and `long_query_time` (I usually log queries taking more than 1 sec. to execute) for tracing MySQL performance. The `pt-query-digest` (Percona Toolkit) from Percona makes aggregation of `slow_query_log` better

`pt-query-digest` usage :

`pt-query-digest slow.log`



**MinervaDB**



# Contact Shiv Iyer

Email - [shiv@minervadb.com](mailto:shiv@minervadb.com)

Skype - [shiv@minervadb.com](https://www.skype.com/people/shiv@minervadb.com)

Google Hangouts - [shiv@minervadb.com](https://www.google.com/hangouts/people/shiv@minervadb.com)

Twitter - <https://twitter.com/thewebscaledba>

Facebook - <https://www.facebook.com/TheWebScaleDBA>

LinkedIn - <https://www.linkedin.com/in/thewebscaledba/>



**MinervaDB**

Thank You



**MinervaDB**