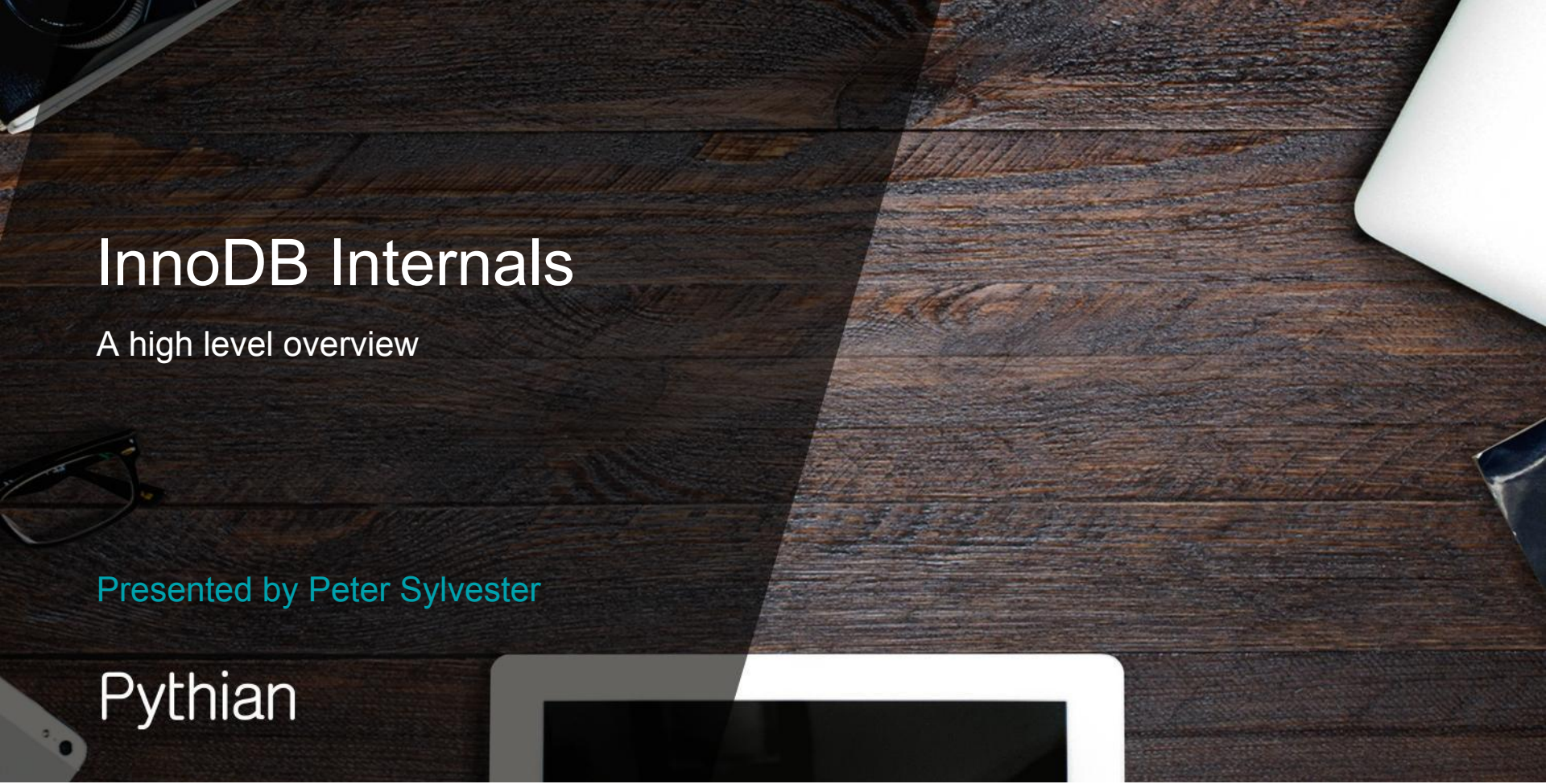

InnoDB Internals
A high level overview
Presented by Peter Sylvester
Pythian

About the presenter

- Joined Pythian team in January 2015
- Internal Principal Consultant at Pythian since June 2016
- Worked with MySQL since 2008 (5.0) in both DBA and DBD roles
- Originally worked as SQL Server DBA
- Is originally from Detroit, but currently lives in the Greater Toronto Area.
- Social media
  - Twitter = @PeterTheDBA
  - Linkedin = https://www.linkedin.com/in/petertsylvester

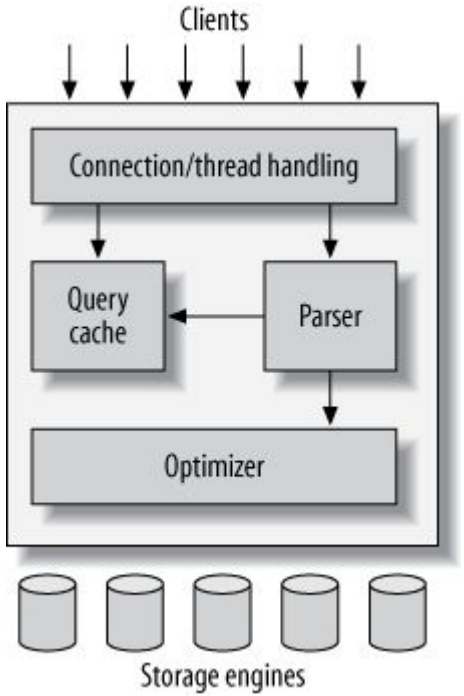
Basis of presentation

- Re-read the MySQL 5.6 reference guide on InnoDB and found it to be very vague.
- Reached out to other SMEs and performed my own lab work to help fill in the gaps.
- Turned my notes into a 5-part blog post series covering InnoDB Mechanics (MySQL 5.6) related to…
  - Memory
  - I/O (file structure and logs)
  - I/O (table data)
  - Concurrency
  - Consistency / Statistics Handlings
  - 55 Variables covered

Presentation outline

- What is InnoDB?
- Importance of InnoDB
- InnoDB: A brief history
- InnoDB High Level Mechanics
  - Memory
  - IO

What is InnoDB?



Why is it important to understand InnoDB mechanics?

- InnoDB is the ACID compliant solution for MySQL, making it the most attractive and most commonly used storage engine for modern installations of MySQL today.
- MySQL 8.0 is starting to phase out MYISAM engine
  - No transportable MYISAM tablespaces
  - mysql schema is now INNODB
- AWS RDS platforms either discourage use of MYISAM (MySQL / RDS backup consistency issues) or just don’t allow it at all (Aurora)
- InnoDB = MySQL (OLTP) : MySQL (OLTP) = InnoDB

Having a high level understanding of InnoDB mechanics allows for ease of configuration and ease of troubleshooting

InnoDB: A brief History

- 1995: Originally developed by Innobase (Heikki Turri. Finland)
- 2001: MySQL 3.23, InnoDB support is added
- 2003: MySQL 4.0, InnoDB is enabled by default
- 2010: MySQL 5.1, InnoDB plugin 1.0.7 is released as a plugin, allowed for Barracuda file format (eventually rolled into later minor versions of MySQL 5.5)
- 2010: MySQL 5.5, InnoDB is default storage engine, fast index creation added
- 2013: MySQL 5.6, InnoDB memcached plugin, fulltext index, and Online DDL are added
- 2015: MySQL 5.7, InnoDB specific table partitioning is added
- ???: MySQL 8.0, the beginning of MYISAM phase out

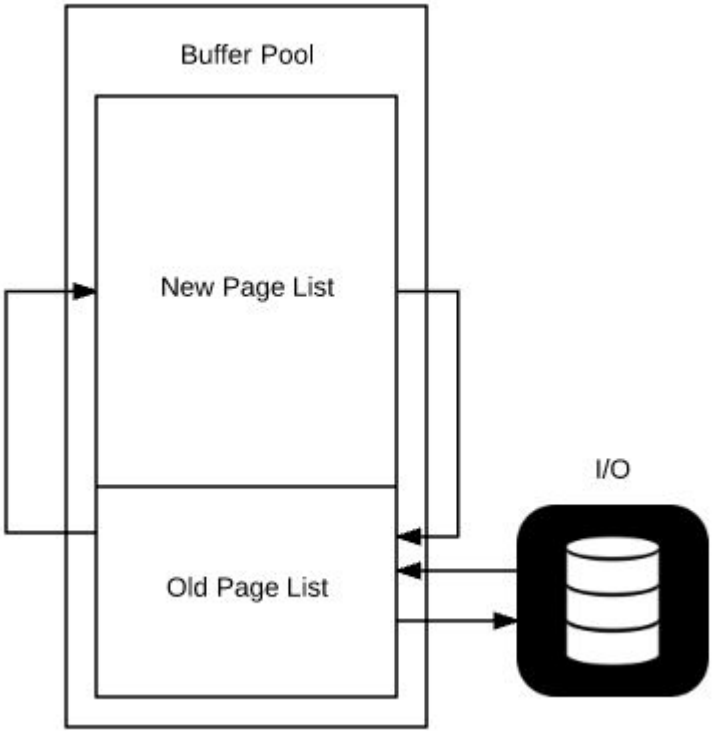
InnoDB Mechanics

- Memory
- I/O
- Other considerations

InnoDB Mechanics: Memory

- InnoDB Buffer Pool
- Old blocks percentage / Old blocks time
- Buffer pool instances
- Adaptive Hash indexing

InnoDB Mechanics: Memory: Buffer Pool



InnoDB Mechanics: Memory: Buffer Pool

innodb_buffer_pool_size

- Default: 128M
- Should be approximately 75% of system memory
- Note that global memory is also shared with MYISAM key cache

Metric:

- SHOW ENGINE INNODB STATUS
  - BUFFER POOL AND MEMORY
    - Free Buffers
    - Buffer pool hit rate

InnoDB Mechanics: Memory: Buffer Pool

innodb_old_blocks_pct

- Default: 37%
- Consider increasing under high eviction (not made young) scenarios where a page is being evicted before it is considered ‘hot’
- Consider decreasing if you have largely stagnant data

Metric

- SHOW ENGINE INNODB STATUS
  - BUFFER POOL AND MEMORY
    - Pages not made young

InnoDB Mechanics: Memory: Buffer Pool

innodb_old_blocks_time

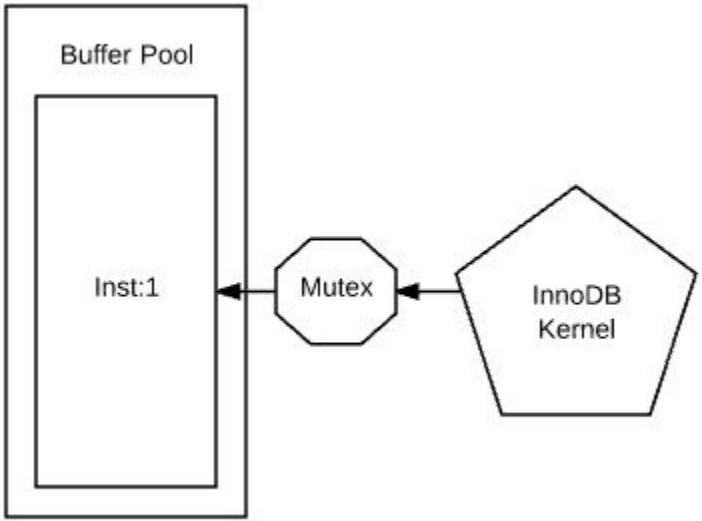
- Default: 1 second
- Prevent hot pages from being removed from the buffer pool by full table scans, commonly occurring with logical backups like mysqldump

Metric

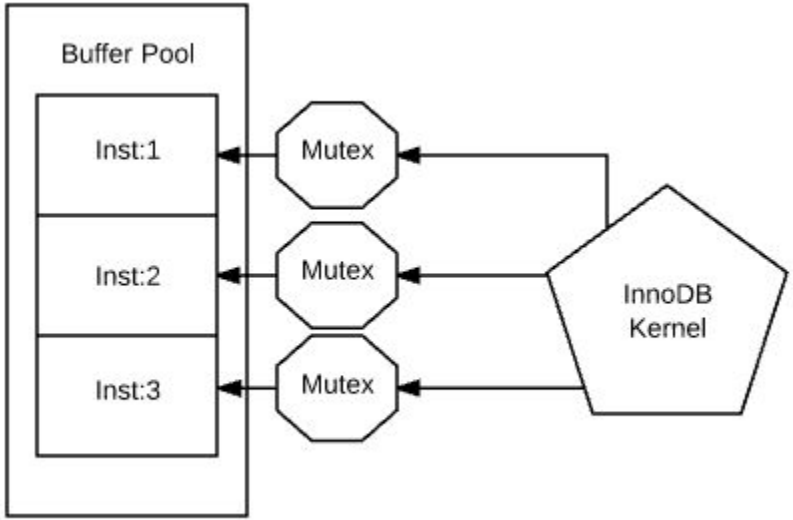
- SHOW ENGINE INNODB STATUS
  - BUFFER POOL AND MEMORY
    - Pages not made young

Experimentation with old blocks pct and old blocks time may be required to find the best configuration for your use case.

InnoDB Mechanics: Memory: Buffer Pool



InnoDB Mechanics: Memory: Buffer Pool



InnoDB Mechanics: Memory: Buffer Pool

innodb_buffer_pool_instances

- Default: 8, unless you have a buffer pool less than 1G, then default will be 1. In earlier versions of 5.6 the default was 1 regardless.

Metric

- Performance Schema
  - Instrument: wait/synch/mutex/innodb/buf_pool_mutex

InnoDB Mechanics: Memory: Adaptive Hash Index

- Adds dynamic hash indexing to innodb data pages referencing a prefix of frequently accessed b-tree index pages / leaf nodes
- Is enabled by default
- No control of the algorithm

Metric:

- SHOW ENGINE INNODB STATUS
  - INSERT BUFFER AND ADAPTIVE HASH INDEX
    - Adaptive hash searches vs non-hash searches

InnoDB Mechanics: Memory: Adaptive Hash Index

- Single resource
- Partitioning supported in Percona 5.6 and added in Oracle MySQL 5.7 and MariaDB 10.2
- If you see semaphore issues that relate to source file btr0sea.c, you need to add partitions or disable adaptive hash indexing entirely. You may also see improvement by adjusting innodb_thread_concurrency.
- Mutex latches are directly tied to individual indexes, so contention issues may not be resolved by adding partitioning if you have one hot index

InnoDB Mechanics: Memory: Considerations

- Ensure that your memory configuration doesn’t put you in a position where OOM is a possibility
- Remember that there are other global memory caches like the MYISAM key buffer as well as session memory caches like the sort, join, read, and read_rnd buffer
- Mysqlcalculator.com is a good baseline

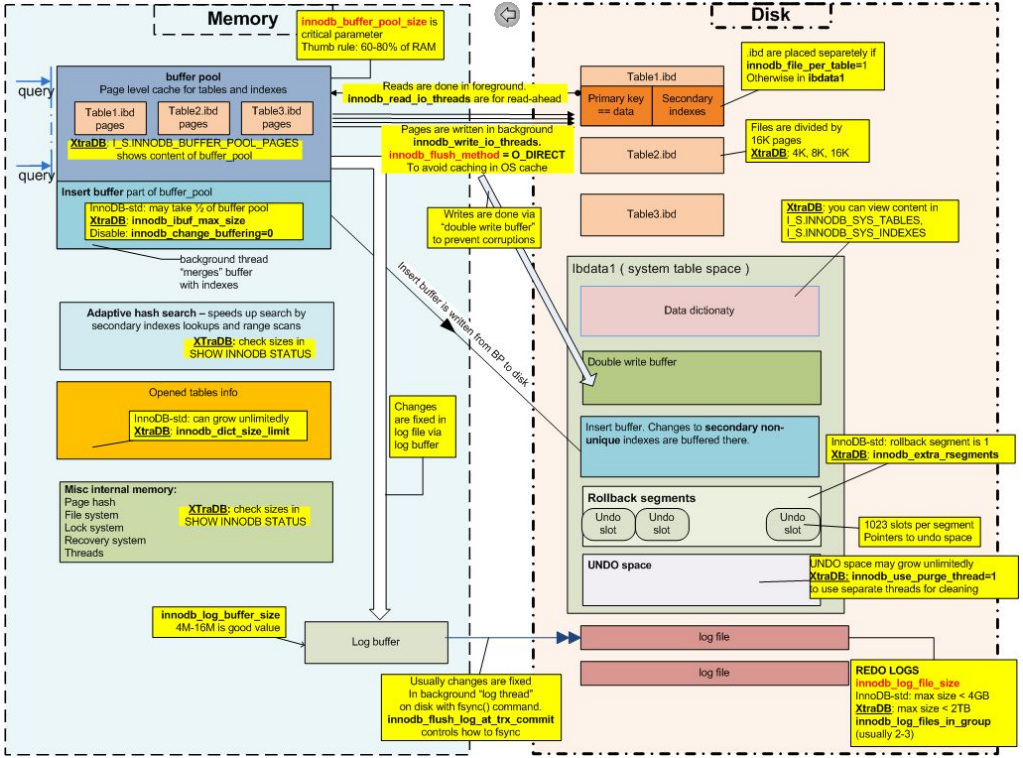
InnoDB Mechanics: Memory

Q&A

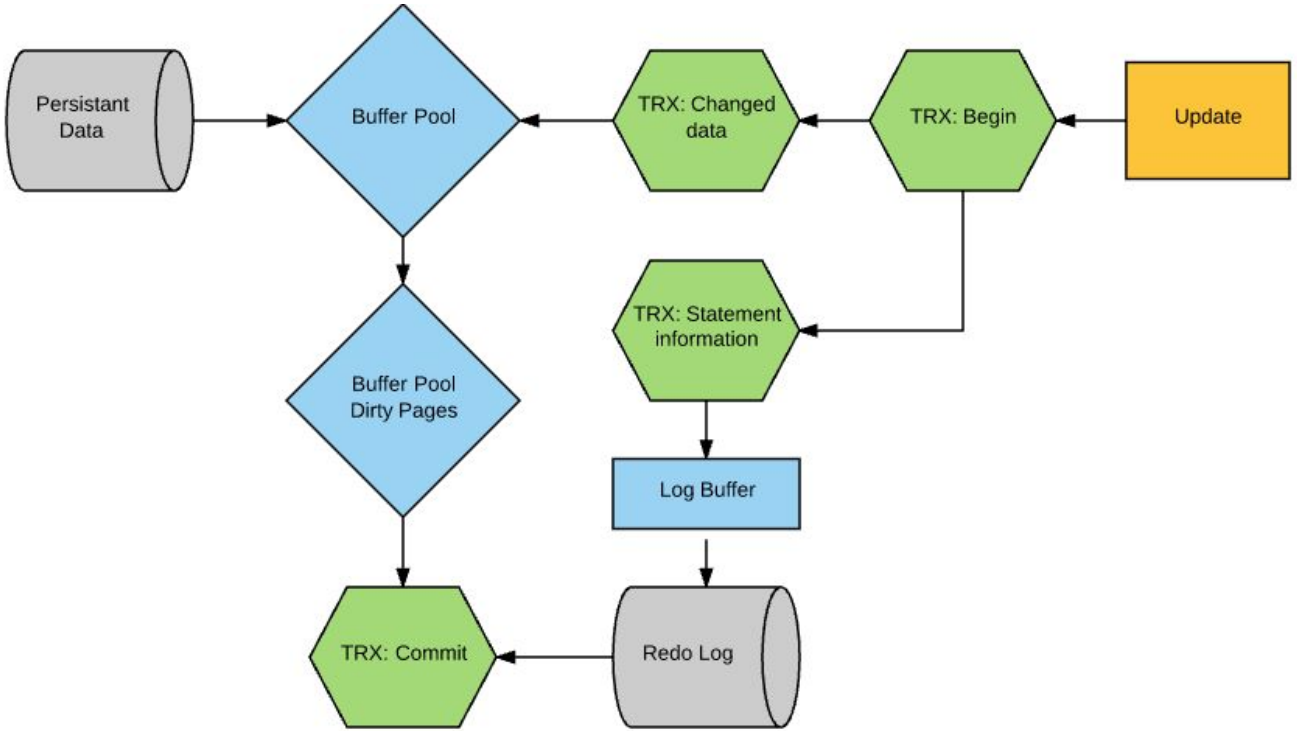
InnoDB Mechanics: I/O

- Transaction Processes / Redo logs
- Flush processes
- Other considerations

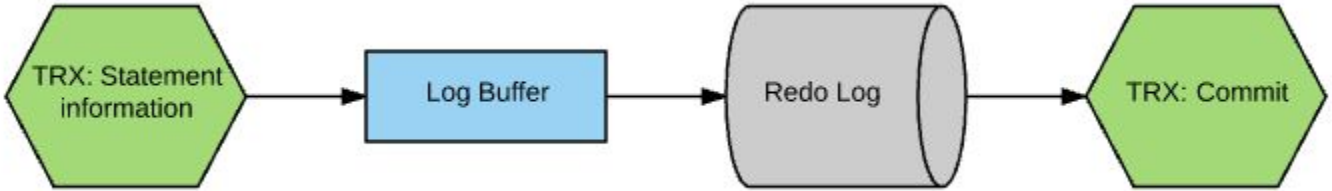
InnoDB Mechanics: I/O: Complicated, eh?



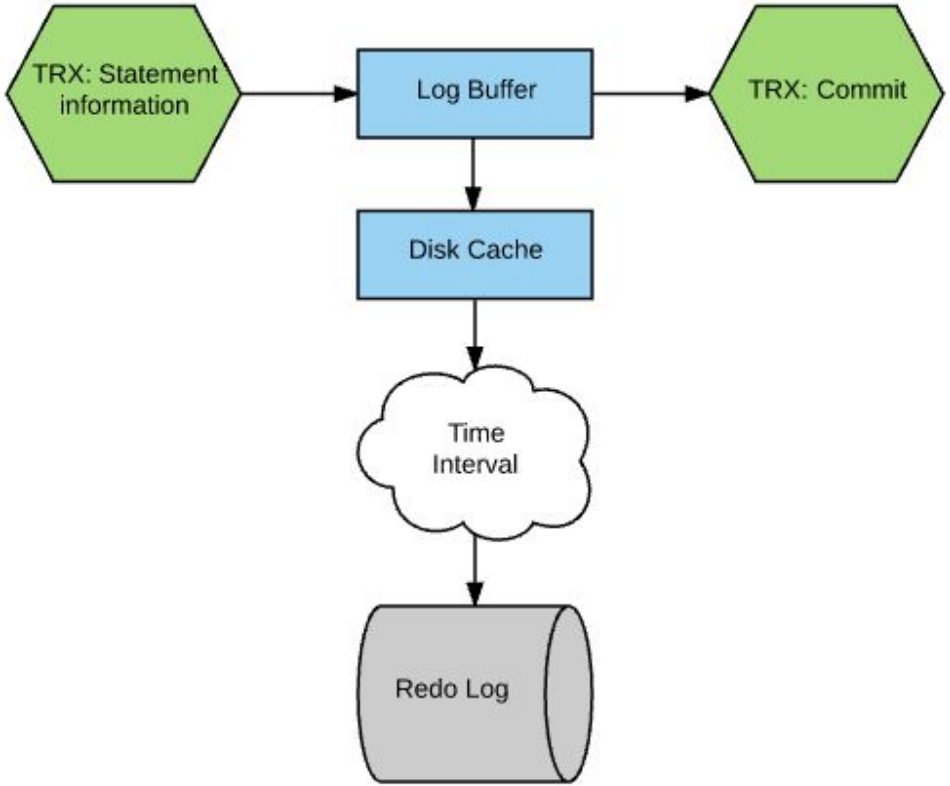
InnoDB Mechanics: I/O: Redo Logs



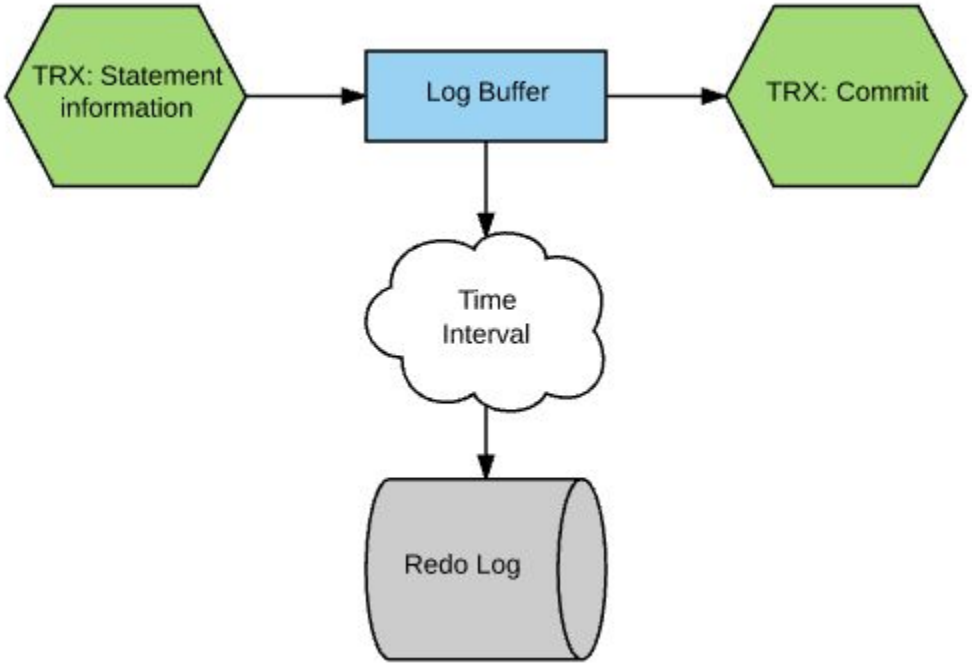
InnoDB Mechanics: I/O: Redo Logs: Flush log at trx commit 1



InnoDB Mechanics: I/O: Redo Logs: Flush log at trx commit 2



InnoDB Mechanics: I/O: Redo Logs: Flush log at trx commit 0



InnoDB Mechanics: Redo Logs

innodb_log_file_size / innodb_log_files_in_group

- Size of redo log = Innodb_log_file_size * innodb_log_files_in_group
- Default = 48M * 2 = 96M
- Should be able to support 1 hour DML traffic

Metric

- SHOW ENGINE INNODB STATUS
  - LOG
    - Log sequence number
- ((LSN position) - (LSN position 1 hour ago)) / 1024^2 = Desired log size in M

InnoDB Mechanics: Redo Logs

innodb_log_buffer_size

- Default = 8M

Metric

- SHOW ENGINE INNODB STATUS
  - LOG
    - Log Sequence Number - Log flushed up to = Used log buffer (bytes)
- Aim to go no higher than 30% consumption with normal traffic. Even when you are flushing as part of the transaction commit process.

InnoDB Mechanics: Redo Logs

innodb_flush_log_at_trx_commit

- Default = 1
- Setting to anything other than 1 forces sacrifices ACID compliance for performance gain

Metric

- Performance schema
  - Instrument: wait/io/file/innodb/innodb_log_file

InnoDB Mechanics: Sync Binlog

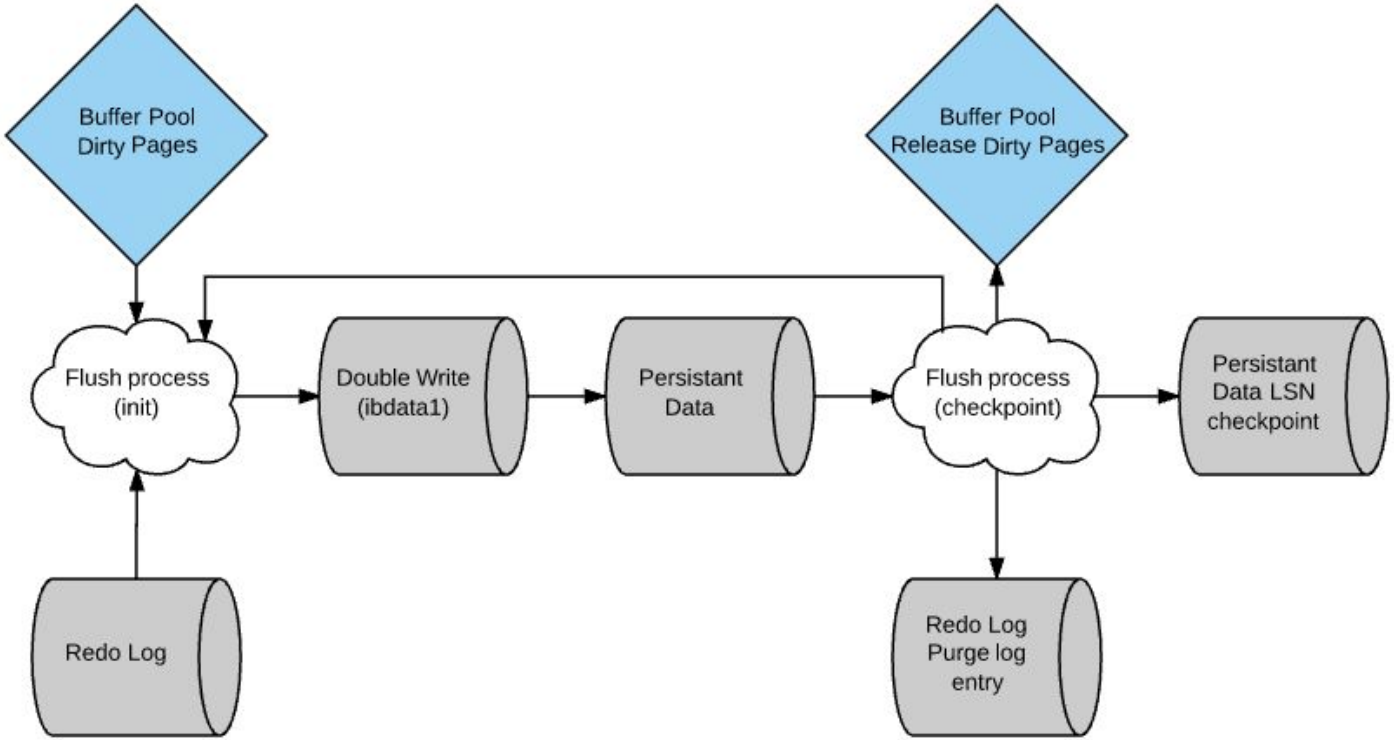
sync_binlog

- Unlike redo logs, binary logs are meant to drive PITR and replication and is not a part of InnoDB data consistency
- Synchronizes the flushing of commit groups to binary log at transaction commit
- Default = 0
- Set to 1 when you want the binary log to stay in sync with InnoDB. This will have a performance impact. Setting to 0 is still ACID compliant, but the binary logs are not considered to be consistent to the rest of the RDBMS.

Metric

- Performance schema
  - Instrument: wait/io/file/sql/binlog

InnoDB Mechanics: I/O: Flush / Checkpoint



InnoDB Mechanics: I/O: Adaptive Flushing

- innodb_adaptive_flushing
- innodb_adaptive_flushing_lwm
- innodb_max_dirty_pages_pct
- innodb_max_dirty_pages_pct_lwm
- innodb_flushing_avg_loops
- innodb_lru_scan_depth
- innodb_flush_neighbors
- innodb_flush_method
- innodb_write_io_threads
- innodb_io_capacity
- innodb_io_capacity_max

InnoDB Mechanics: I/O: Flush / Checkpoint

innodb_checksum_algorithm

- Write a checksum to the suffix of a flushed persistent data page
- Used to detect bit rot or partially written pages
- Default = INNODB (proprietary)
- Preferred = crc32, scans 32 bits at a time (vs 8), much faster
- Beware strict mode
- No downgrade option apart from logical restore

InnoDB Mechanics: I/O: Flush / Checkpoint

innodb_doublewrite

- Protects against partial page writes
- Recovery process scans pages in tables, if inconsistent with checksum, pull from double write. If Double write is inconsistent with checksum, discard and refer back to redo log.
- Default: 1 (on)
- Only disable if you have an atomic writing filesystem like ZFS or FusionIO, journalling is not enough.

Metric

- Performance schema
  - Instrument: wait/synch/mutex/InnoDB/buf_dblwr_mutex

InnoDB Mechanics: I/O: Other considerations

innodb_file_per_table

- Moves data into one ‘table space’ per table
- Speeds up table truncation
- Allows ease of data management, reclaiming space on disk, transportable tablespaces

Metric

- Performance schema
  - Instrument: wait/io/file/innodb/innodb_data_file
  - Table: file_summary_by_event_name
  - Table: file_summary_by_instance

InnoDB Mechanics: I/O: Other considerations

innodb_file_format

- Barracuda added as part of the InnoDB Plugin
- Adds dynamic and compressed row formats
- Supports all row formats in the previous version (Antelope)
- It was anticipated that there would be several formats developed, but Barracuda simply expanded on Antelope. Option to pick a file format is expected to be removed in 8.0
- Only disable if you believe you may have to downgrade to an older version of MySQL

InnoDB Mechanics: I/O

Q&A

InnoDB variables to consider for a new environment

- Innodb_buffer_pool_size
  - Approx 75% of system memory
- Innodb_file_per_table
  - On
- Innodb_file_format
  - Barracuda
- Innodb_checksum_algorithm
  - crc32 (strict if this is brand new with no data)
- Innodb_log_file_size / files in group
  - Start with at least 512MB, adjust to make sure you have 1 hour of DML covered

InnoDB variables to consider for a new environment

- sync_binlog
  - 1, keep it that way unless you need to move past an issue
- Innodb_flush_method
  - Direct_io is best for most modern use cases. You should still consider fsync if you are not using innodb file per table
- Innodb_io_capacity
  - Set to 60% of IOPS capacity, especially if you're using EC2 or RDS
  - Default is 200, most disk systems are considerably faster
- Innodb_io_capacity_max
  - Set to 99% of IOPS capacity

InnoDB monitoring

- Regardless of issue, always start by looking at the processlist
  - Command: Show processlist
- InnoDB engine status
  - Command: SHOW ENGINE INNODB STATUS
    - Semaphores / Mutexes (contention saturation)
    - Transaction info (can also get from information_schema.innodb_trx)
    - Background I/O thread status
    - Change buffer status
    - Adaptive hash index status
    - Redo log status
    - Buffer pool status
    - Row operations

InnoDB monitoring

- Performance_schema
  - Key tables
    - Events_waits_summary_global_by_event_name
    - Events_statements_summary_by_digest
  - 5.5: Consider enabling when you're having issues
  - 5.6: Consider enabling unless you have a highly concurrent workload
  - 5.7: Enable it
- Historical graphing
  - Percona Monitoring and Management Platform
  - Vivid Cortex

Q&A

THANK YOU

Peter Sylvester
linkedin.com/in/petertsylvester
@PeterTheDBA