ORACLE

The InnoDB Storage Engine for MySQL

Morgan Tocker, MySQL Community Manager
http://www.tocker.ca/

Safe Harbor Statement

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.

4 Years of MySQL Innovation

MySQL Cluster 7.3

MySQL Workbench 6.0

MySQL Migration Wizard

MySQL 5.6

MySQL 5.5

Windows installer & Tools

MySQL 5.7

MySQL Cluster Manager

MySQL Enterprise Monitor 2.3 & 3.0

MySQL Applier for Hadoop

MySQL Enterprise

Backup

Security

Scalability

HA

Audit

MySQL Utilities

MySQL Workbench 5.2 & 6.0

MySQL Cluster 7.2

MySQL Cluster 7.1

MySQL Enterprise Oracle Certifications

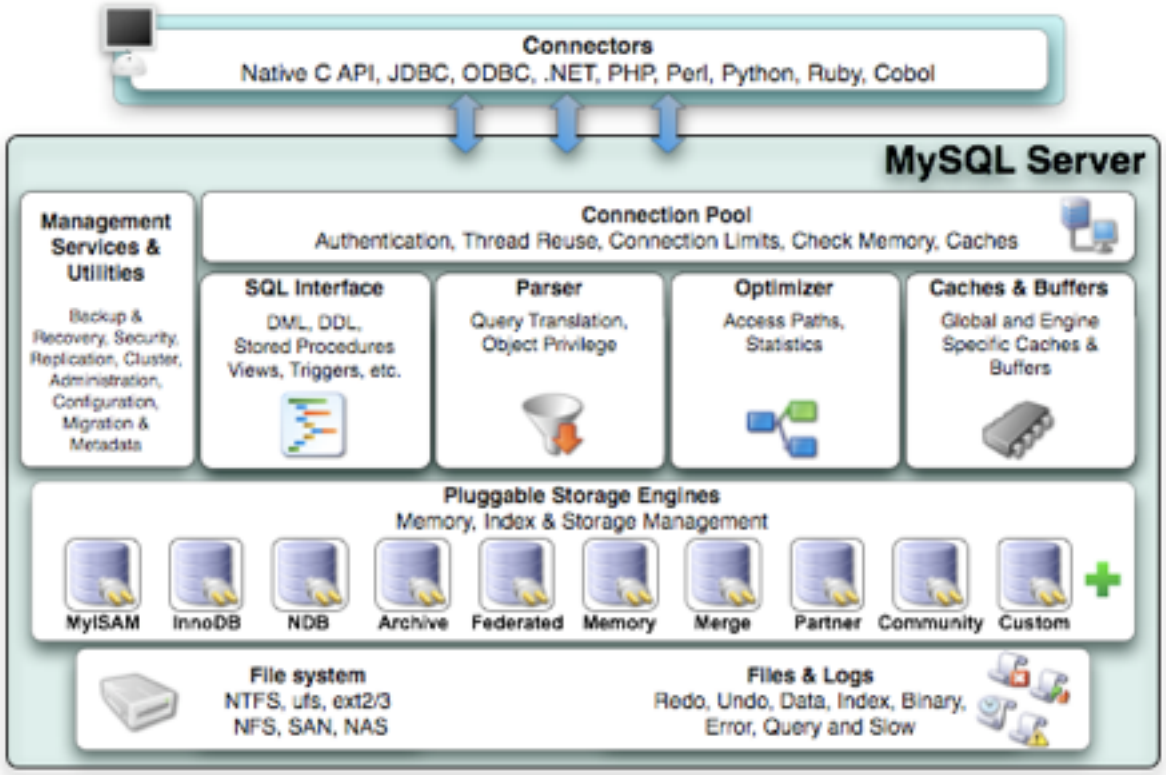
Hello and Welcome!

- I will be talking about InnoDB's internal behaviour.
- Not talking (much) about MySQL.
- Aim of this talk is to give you X-ray vision.
  - i.e. not so many direct takeaways, but one day it will help you debug a problem.

Prerequisites

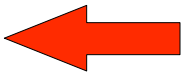
ORACLE

MySQL Architecture



IO Performance

L1 cache reference	0.5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	25 ns
Main memory reference	100 ns
Compress 1K bytes with Zippy	3,000 ns
Send 2K bytes over 1 Gbps network	20,000 ns
Read 1 MB sequentially from memory	250,000 ns
Round trip within same datacenter	500,000 ns
Disk seek	10,000,000 ns
Read 1 MB sequentially from disk	20,000,000 ns
Send packet CA->Netherlands->CA	150,000,000 ns



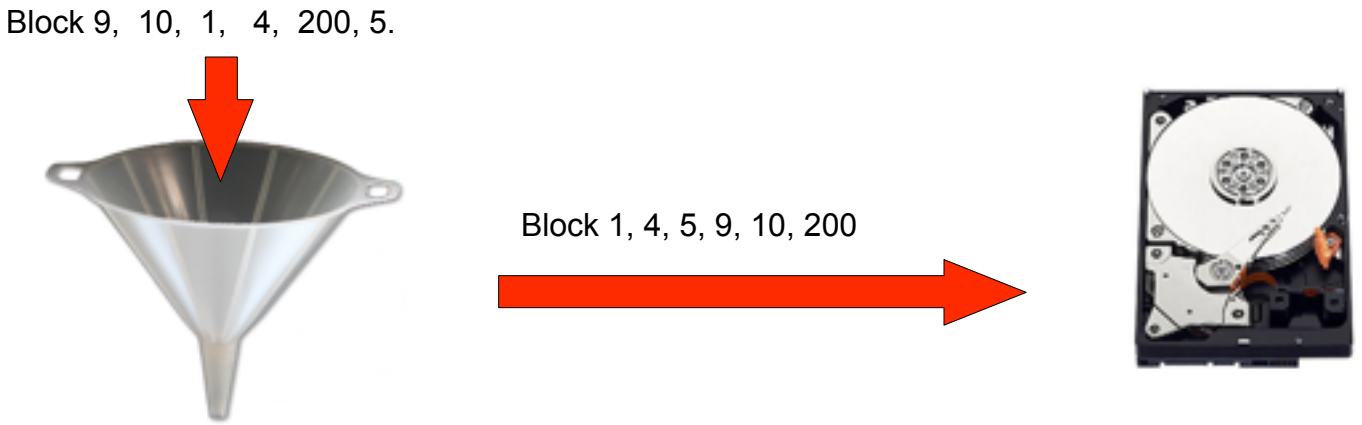
See: http://www.linux-mag.com/cache/7589/1.html and Google http://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009.pdf

IO Performance (cont.)

- 5-10ms per disk IO.
- Maybe 50us for a high end SSD.
  - Still not “memory speed”.

Buffered IO

- Operating Systems compensate well already.
- Reads are cached with free memory.
- Writes **don't happen instantly**.
  - A step is introduced to rewrite and merge.



fsync

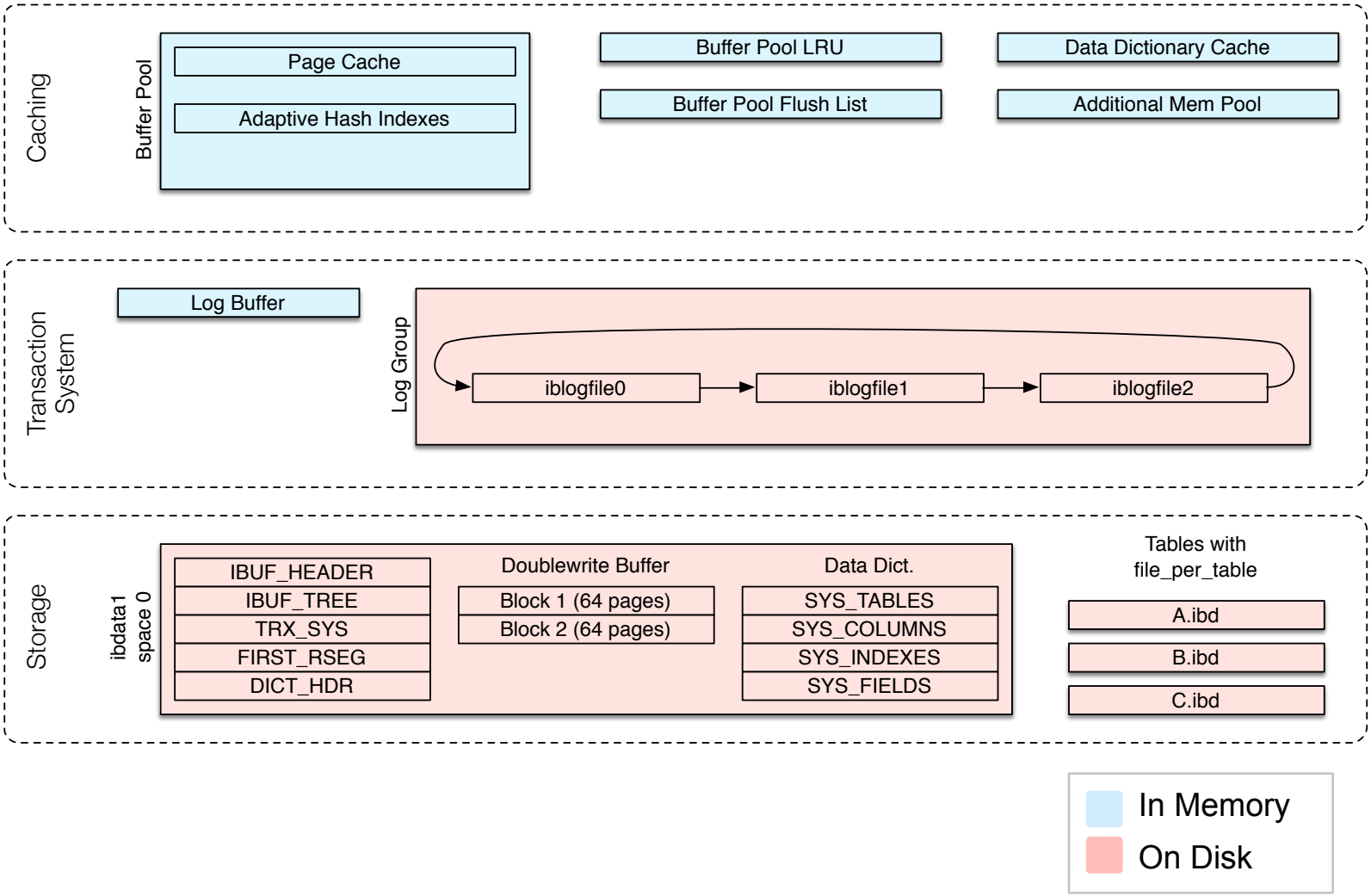
Synopsis
#include <unistd.h>
int fsync(int fd);
int fdatasync(int fd);

Description
fsync() transfers ("flushes") all modified in-core data of (i.e., modified buffer cache pages for) the file referred to by the file descriptor fd to the disk device (or other permanent storage device) where that file resides. The call blocks until the device reports that the transfer has completed. It also flushes metadata information associated with the file (see stat(2)).

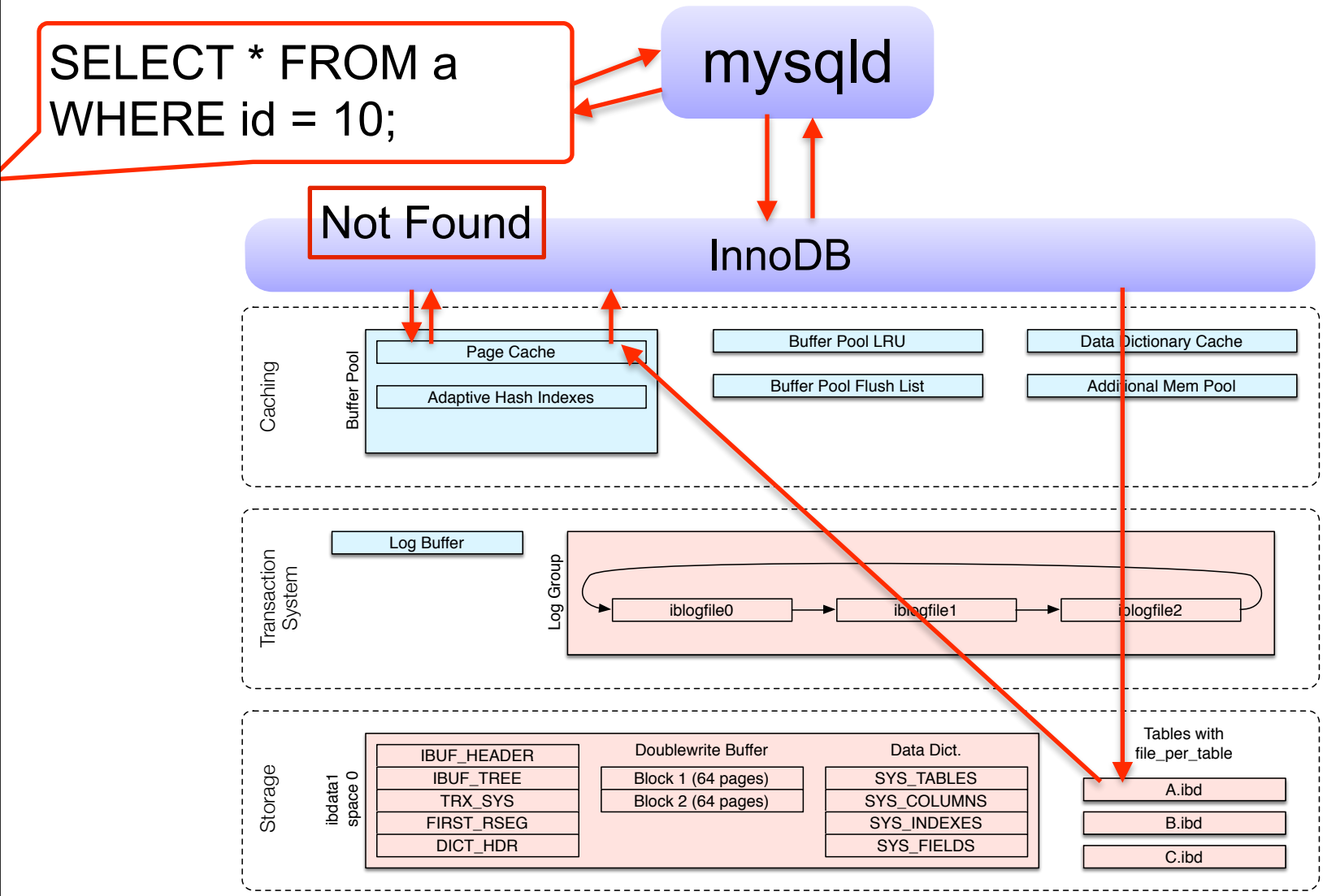
Basic Operation

ORACLE

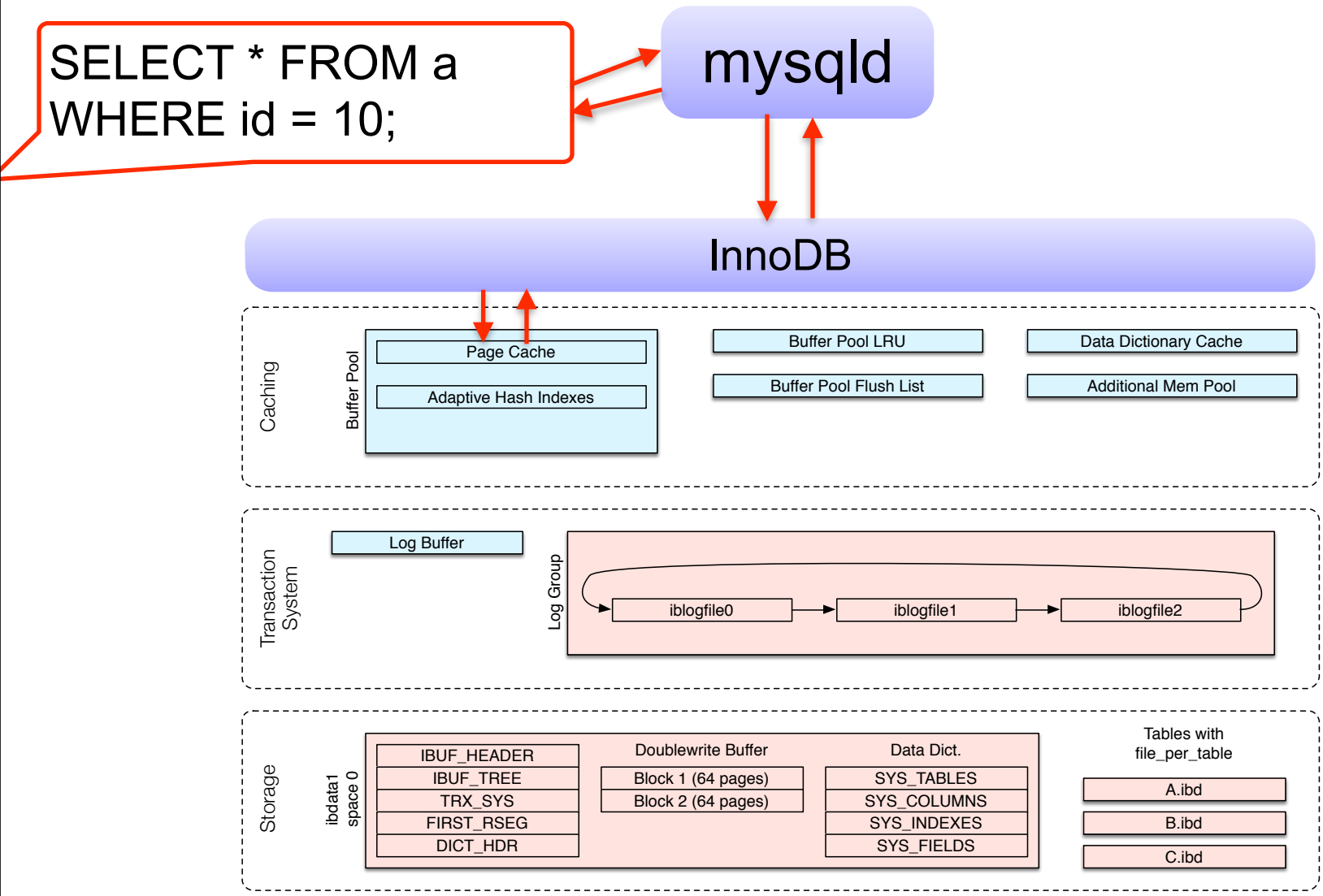
InnoDB High Level Overview



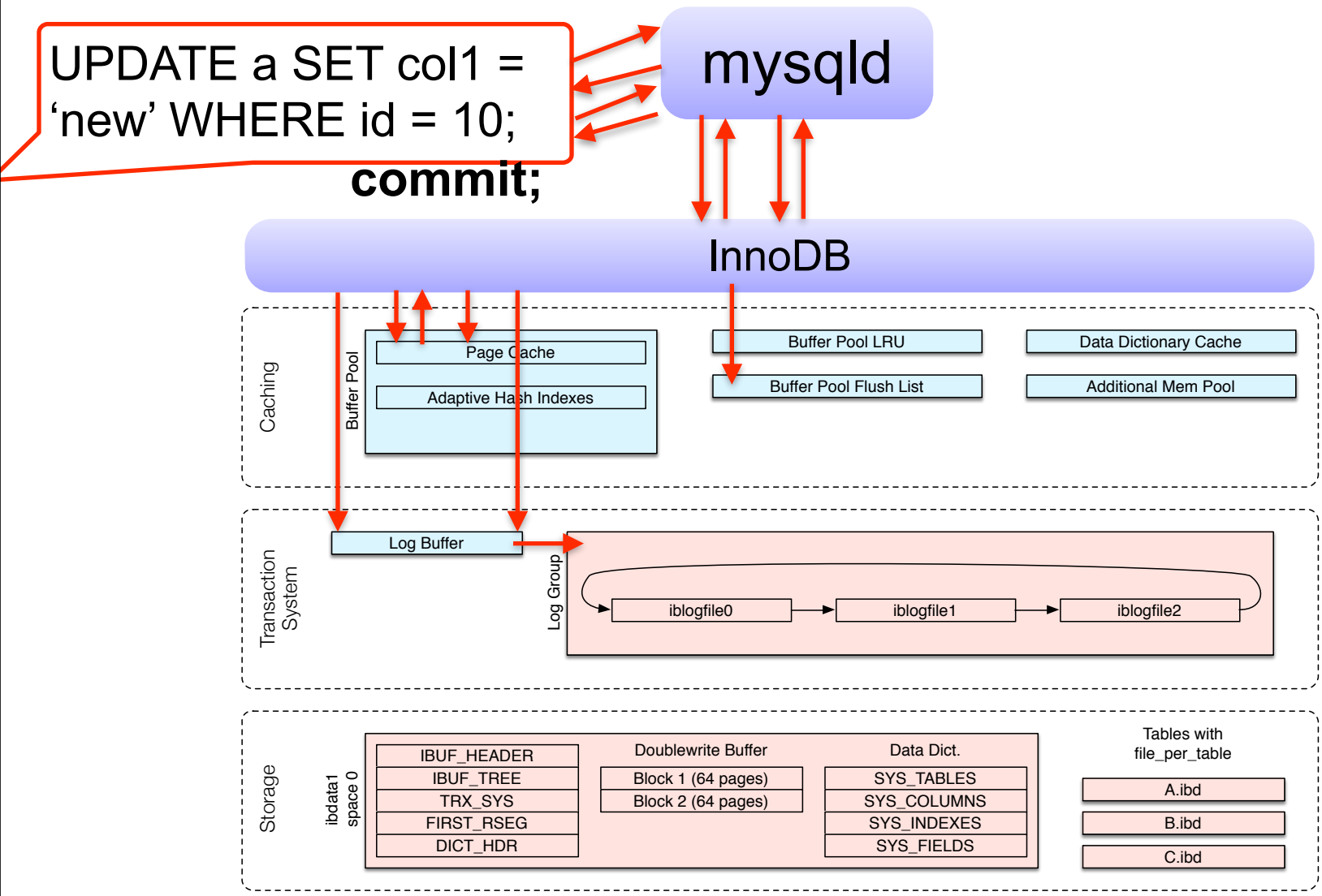
Query (pages not in buffer pool)



Query (pages in buffer pool)



Update Query in a Transaction (simplified)



Log files

- **Provide recovery.**
  - Only written to in regular operation.
  - Read only required if there is a crash.
- Are rewritten over-and-over again.
  - Think of it like a tank tread.

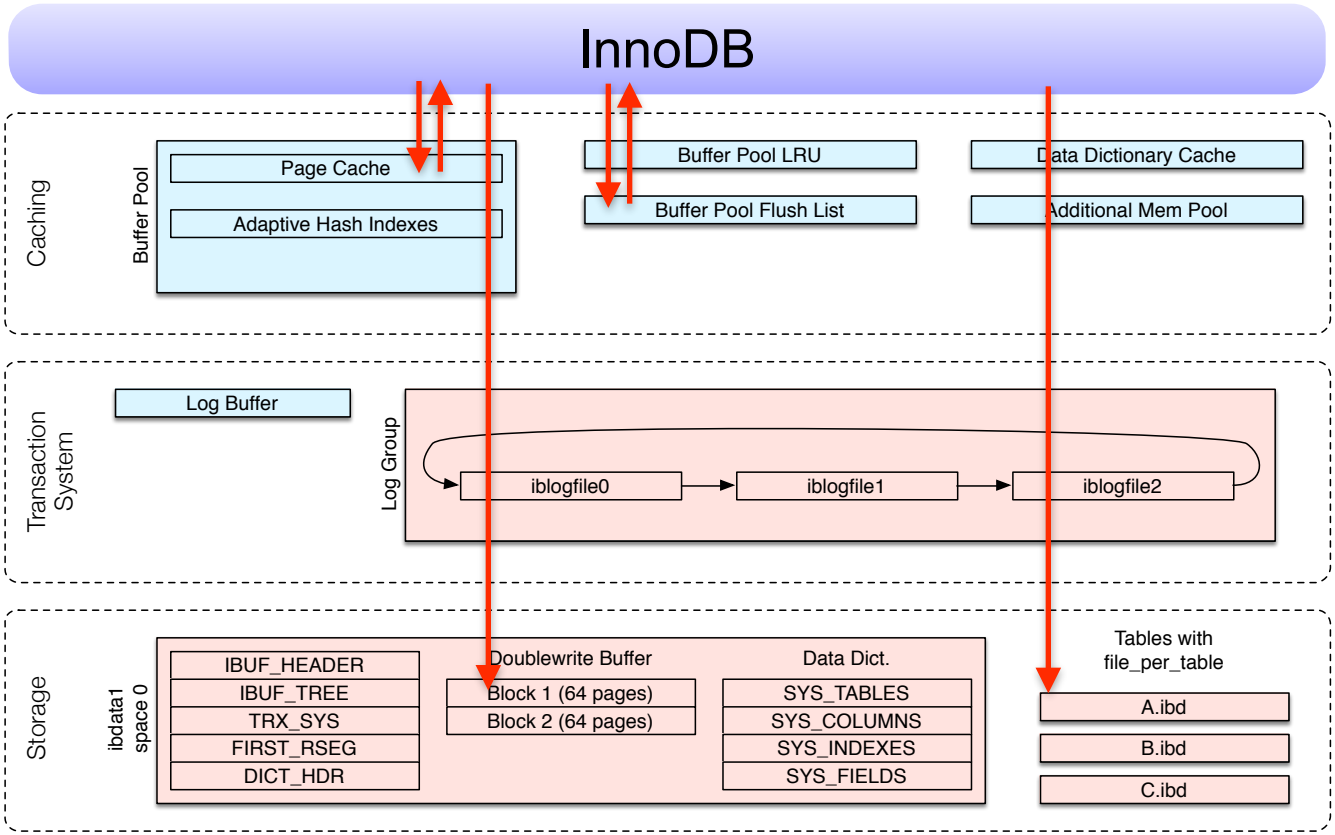
Log files (cont.)

- Are an optimization!
  - 512B aligned sequential writes.
  - Tablespace writes are 16KiB random writes.
- Tablespace writes to same pages in close time window can be merged.
  - Just need a large enough log file.

Checkpoint (Background Activity)

(nothing)

mysqld



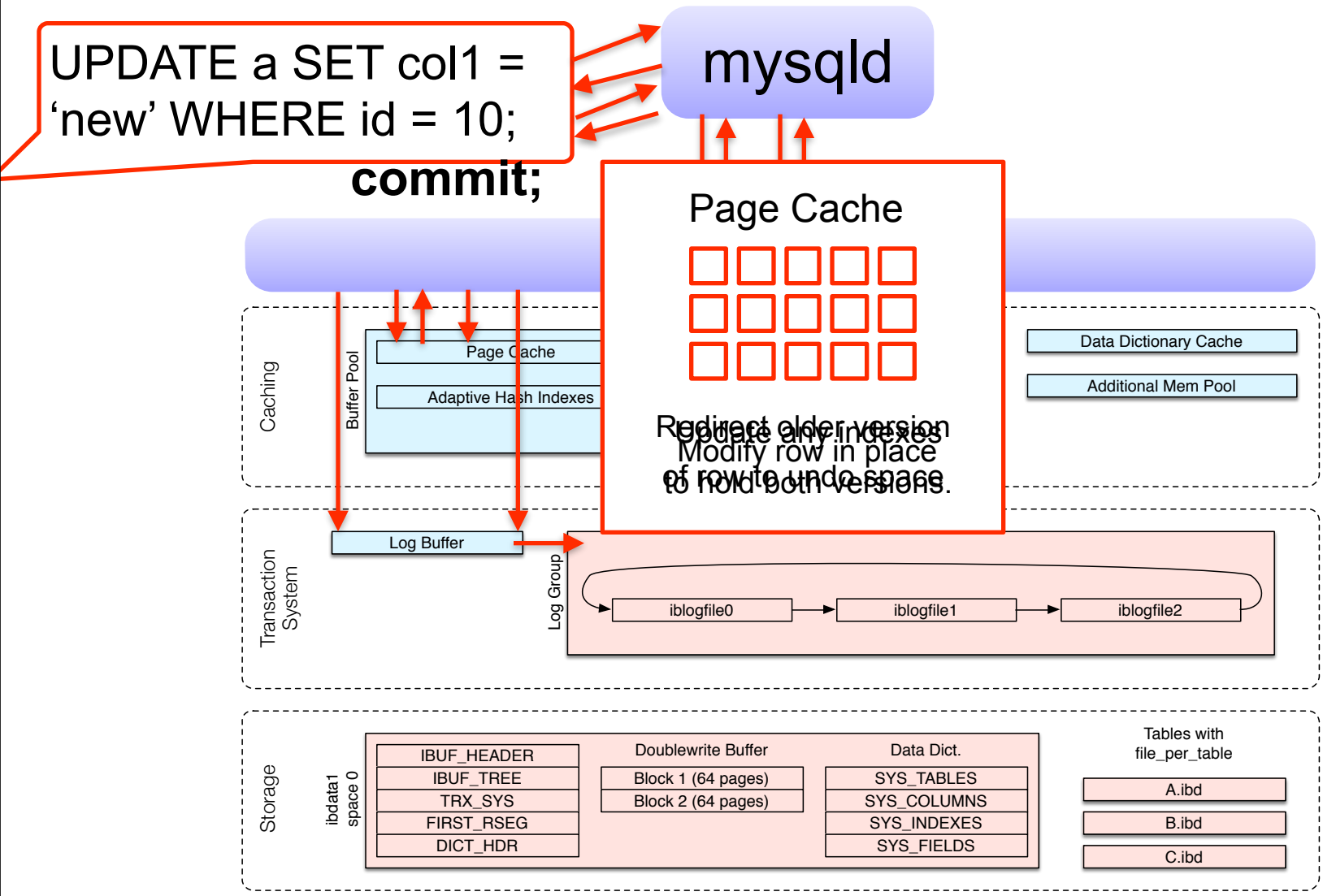
FAQ

- **Q:** What do we write to the log - is it committed data only, or can we write uncommitted data as well?
- **A:** Both.

FAQ

- **Q:** How do you unapply transactions?
- **A:** UNDO space.
  Think of it like a hidden table internally stored in ibdata1.

Update Query (More Accurate*)



Update Query (cont.)

- Background purge process is able to clean old rows from UNDO as soon as oldest transaction advances forward.

Summarized Performance Characteristics

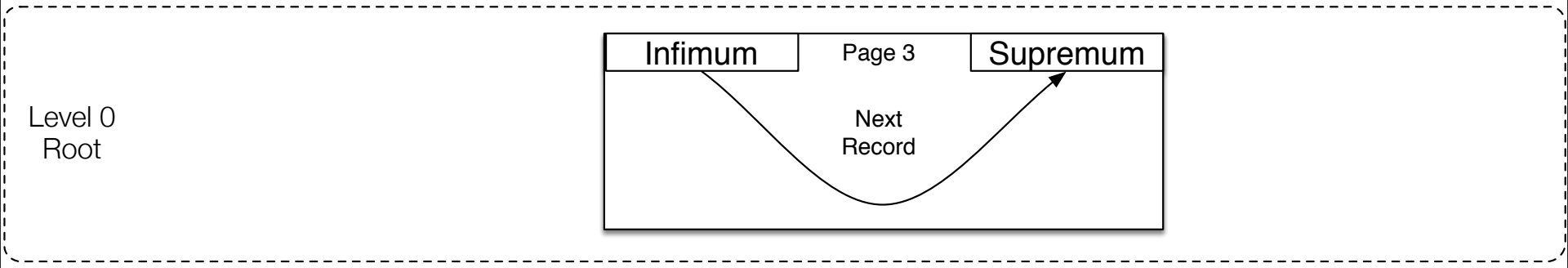
- **Log Files:**
  - Are short sequential writes.
  - They permit InnoDB to delay tablespace writes - enabling more merging/optimization.
- **Buffer Pool:**
  - “In memory version of the tablespace”.
  - Loading/unloading via modified LRU algorithm.

Index Structure

- Indexes and “data” in InnoDB are B+Trees.
  - Clustered Index design means that data itself is stored in an index.

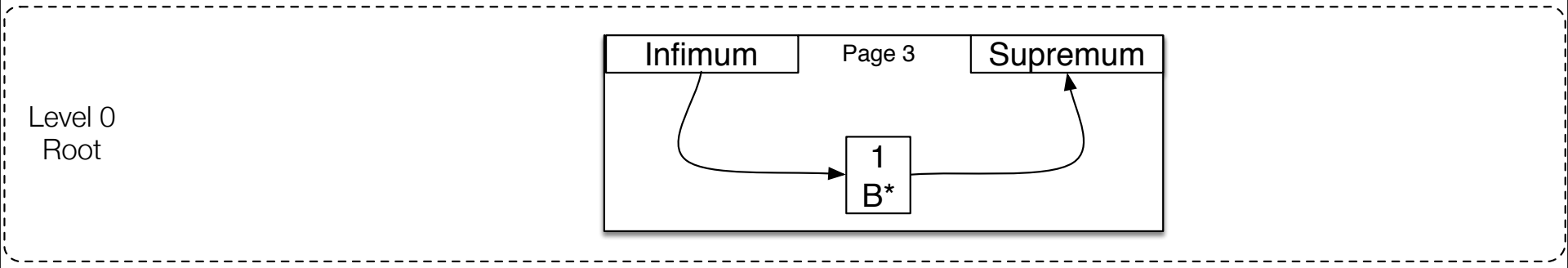
Index Structure (cont.)

Empty root



Index Structure (cont.)

Insert: 1

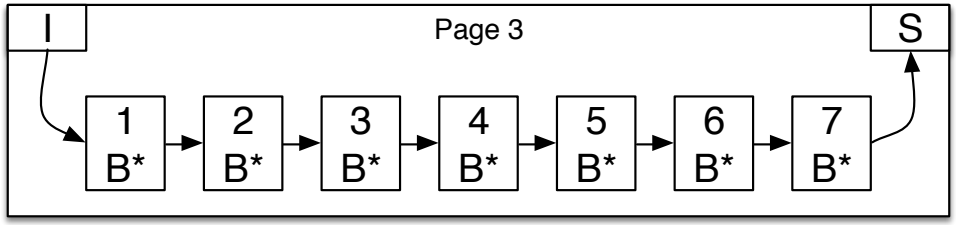


Page size is 16KB
B* is 2KB

Index Structure (cont.)

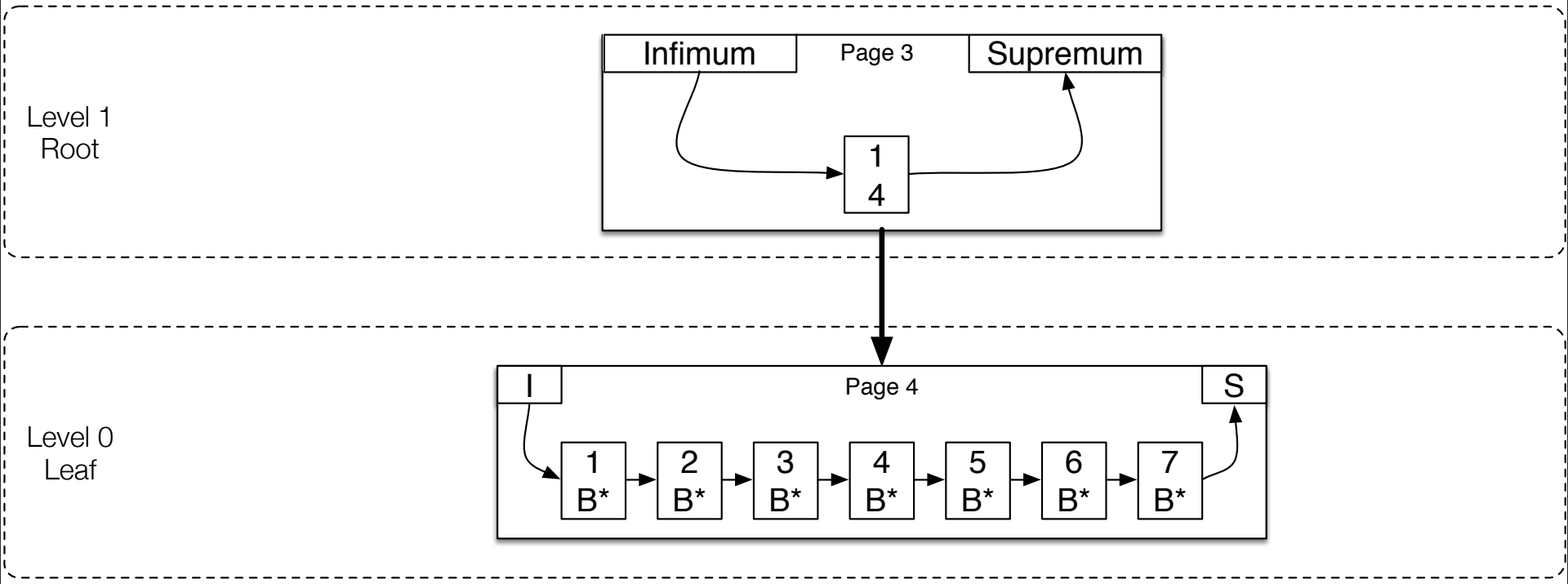
Insert: 1 to 7

Level 0
Root



Index Structure (cont.)

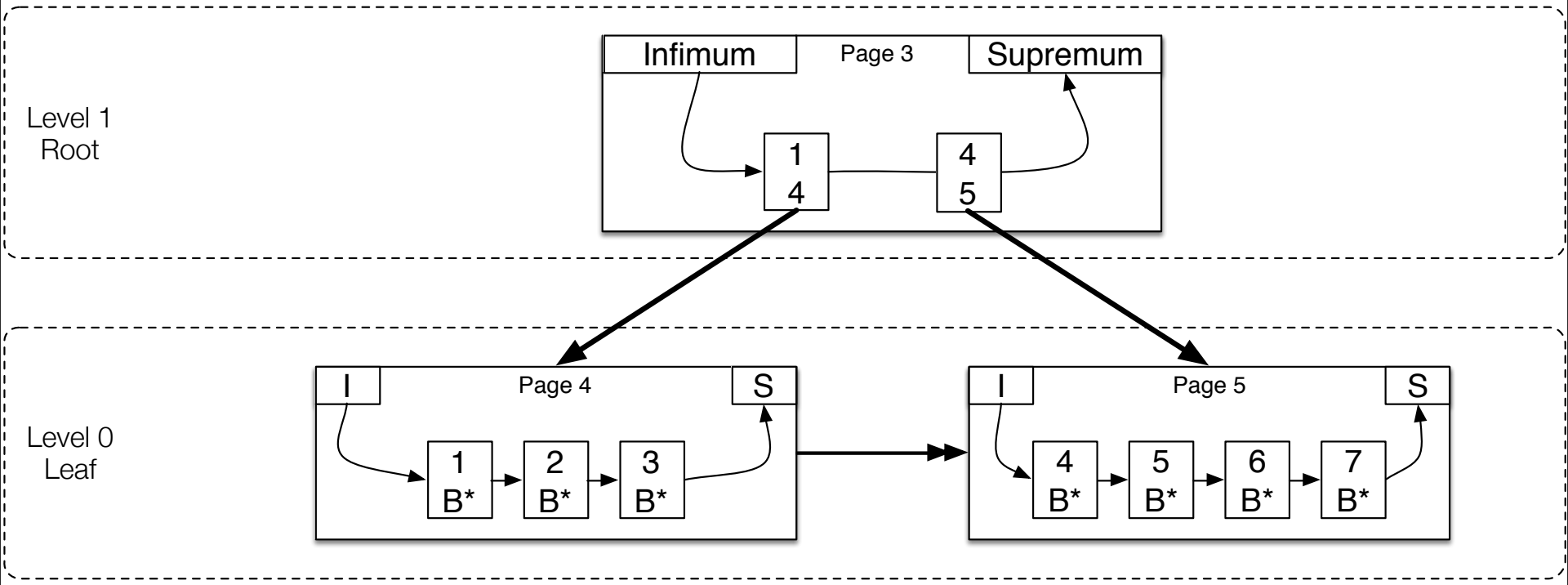
Insert: 8



Allocate new page and link in root
Move records to new page
Split new page

Index Structure (cont.)

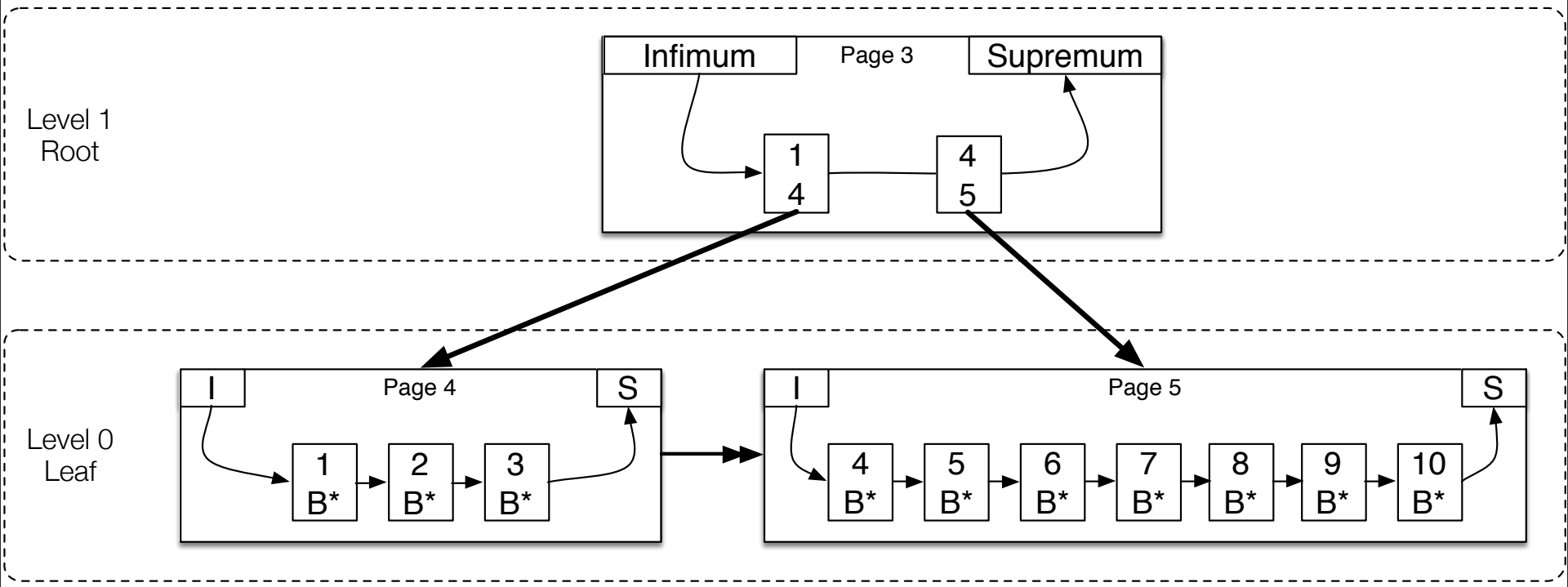
Insert: 8 (Cont.)



Split at the middle of original page

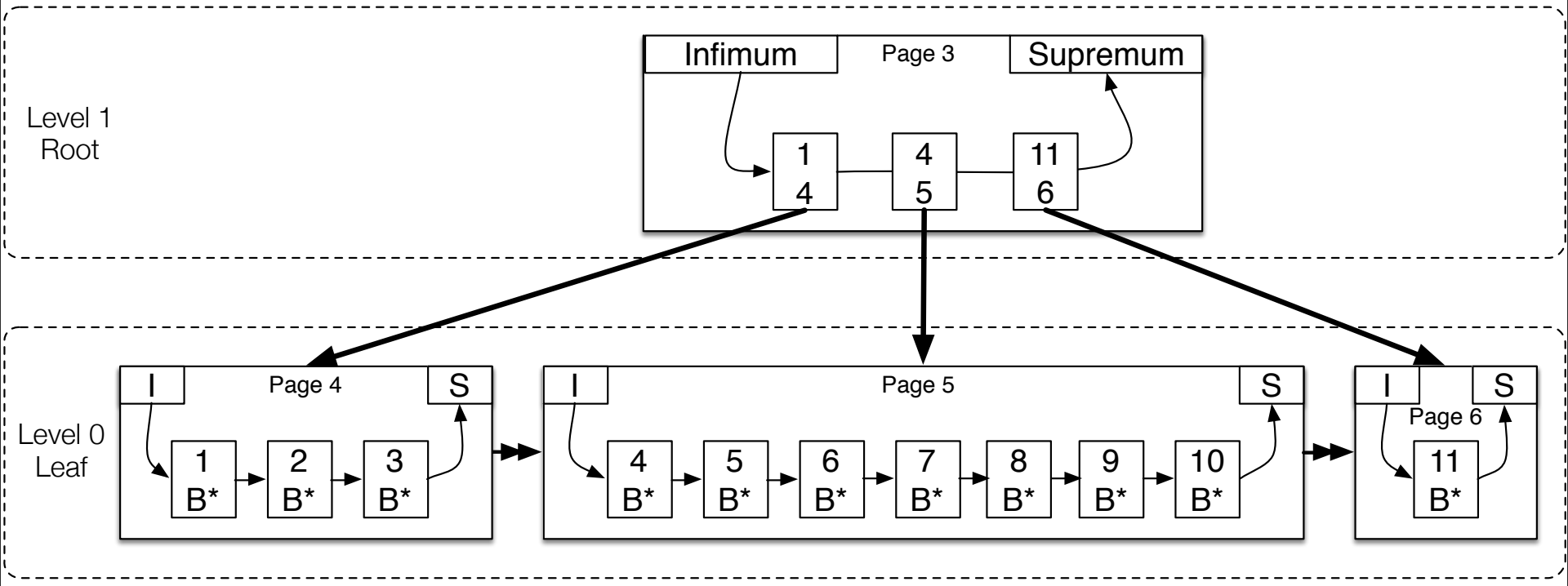
Index Structure (cont.)

Insert: 9 and 10



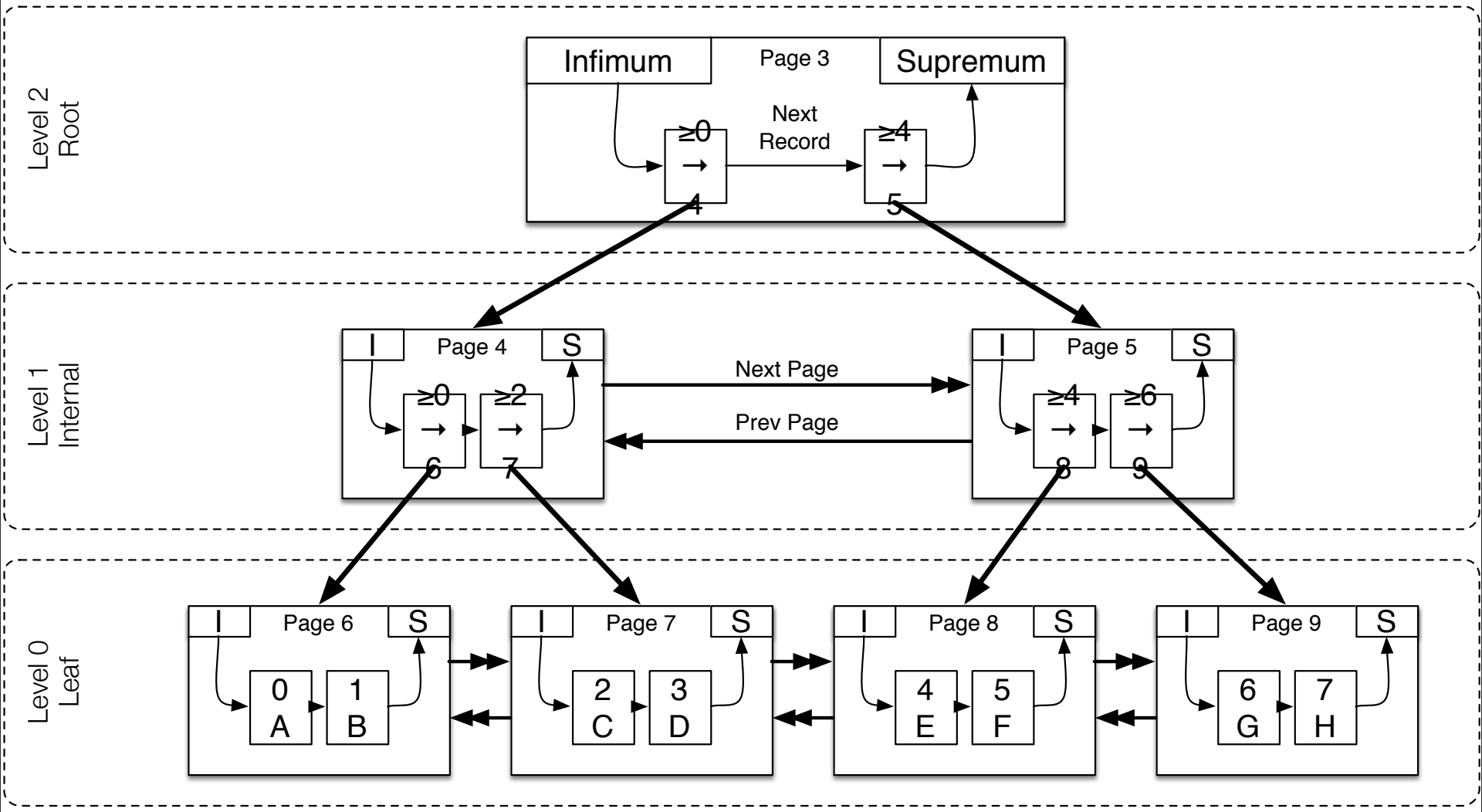
Index Structure (cont.)

Insert: 11

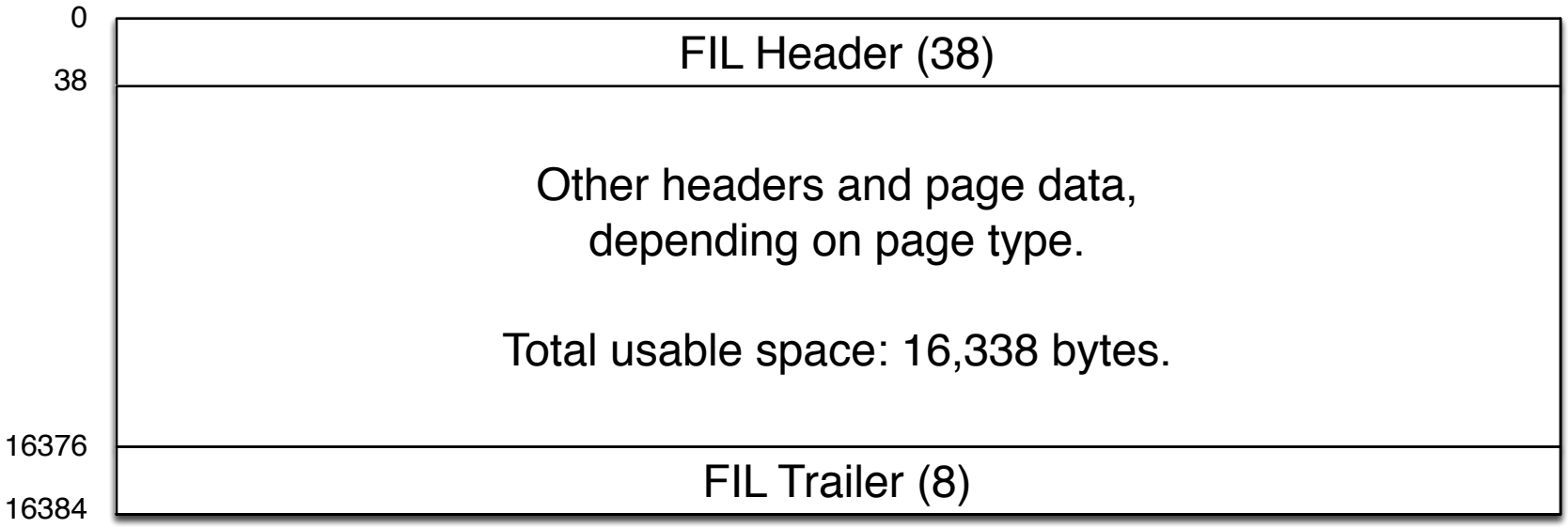


Insert leads to a split at the insertion point

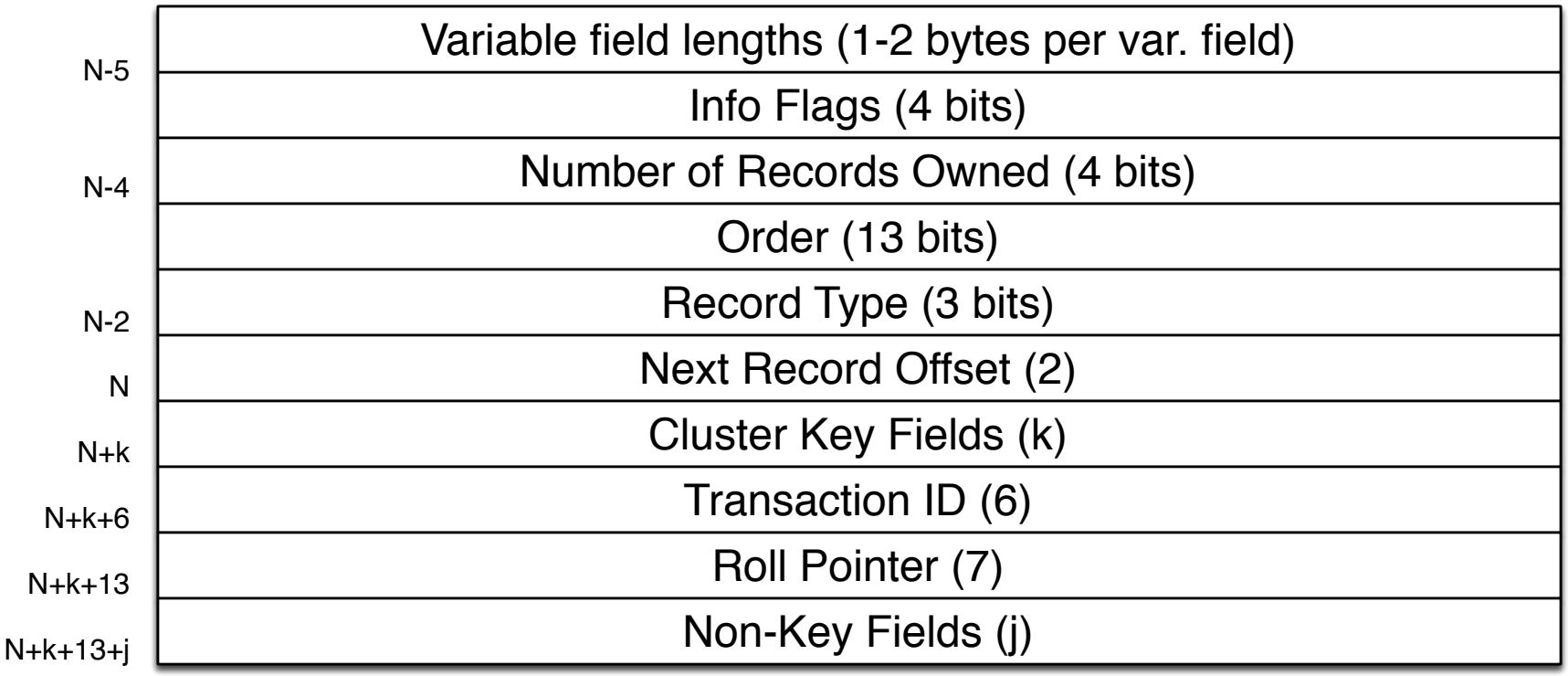
Index Structure



Page Format


0
38
FIL Header (38)
Other headers and page data,
depending on page type.
Total usable space: 16,338 bytes.
16376
FIL Trailer (8)
16384

Row Format


N-5
N-4
N-2
N
N+k
N+k+6
N+k+13
N+k+13+j
Variable field lengths (1-2 bytes per var. field)
Info Flags (4 bits)
Number of Records Owned (4 bits)
Order (13 bits)
Record Type (3 bits)
Next Record Offset (2)
Cluster Key Fields (k)
Transaction ID (6)
Roll Pointer (7)
Non-Key Fields (j)

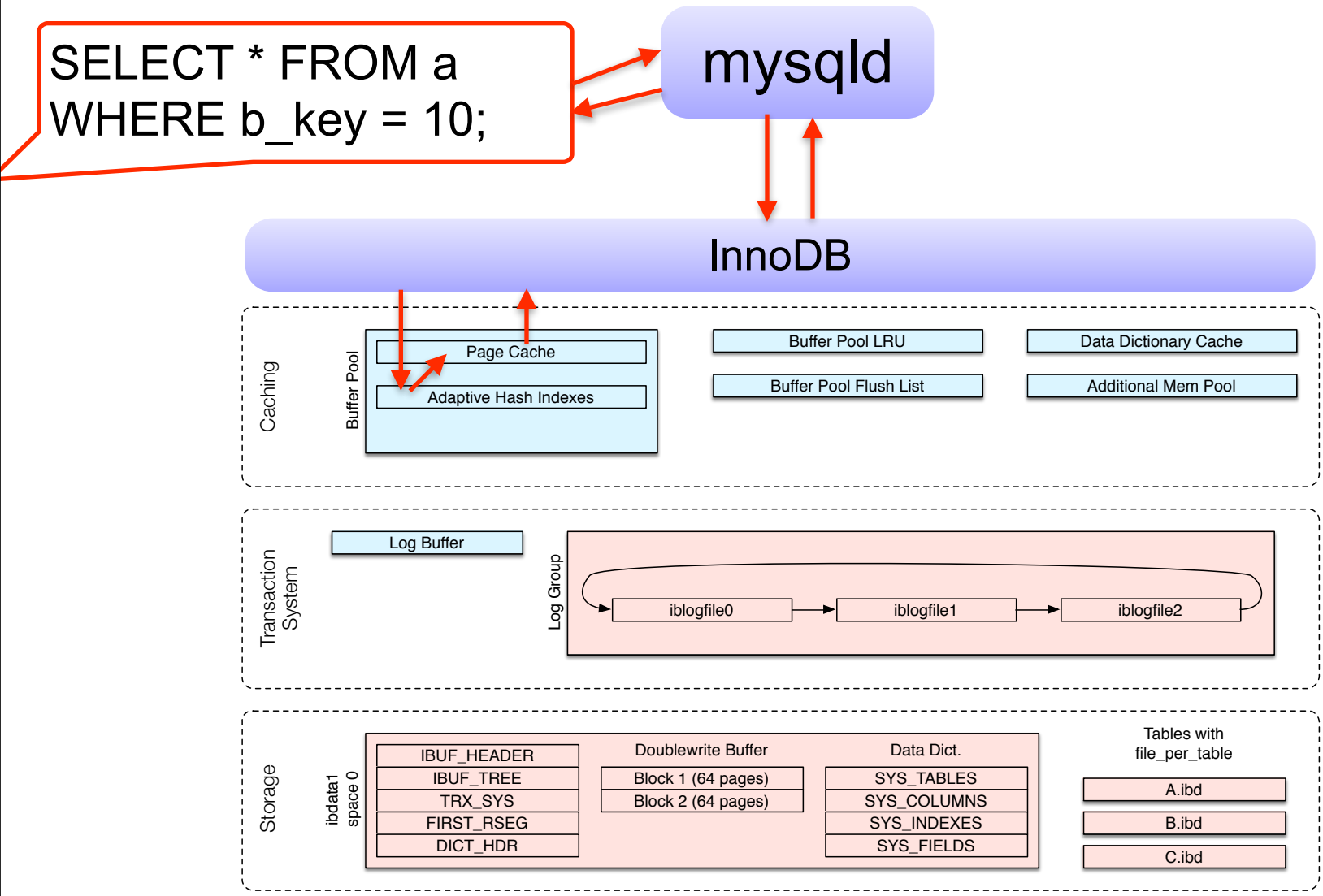
Conclusion

- Page is basic unit of storage.
- Default is 16KiB
- Rows of variable length.

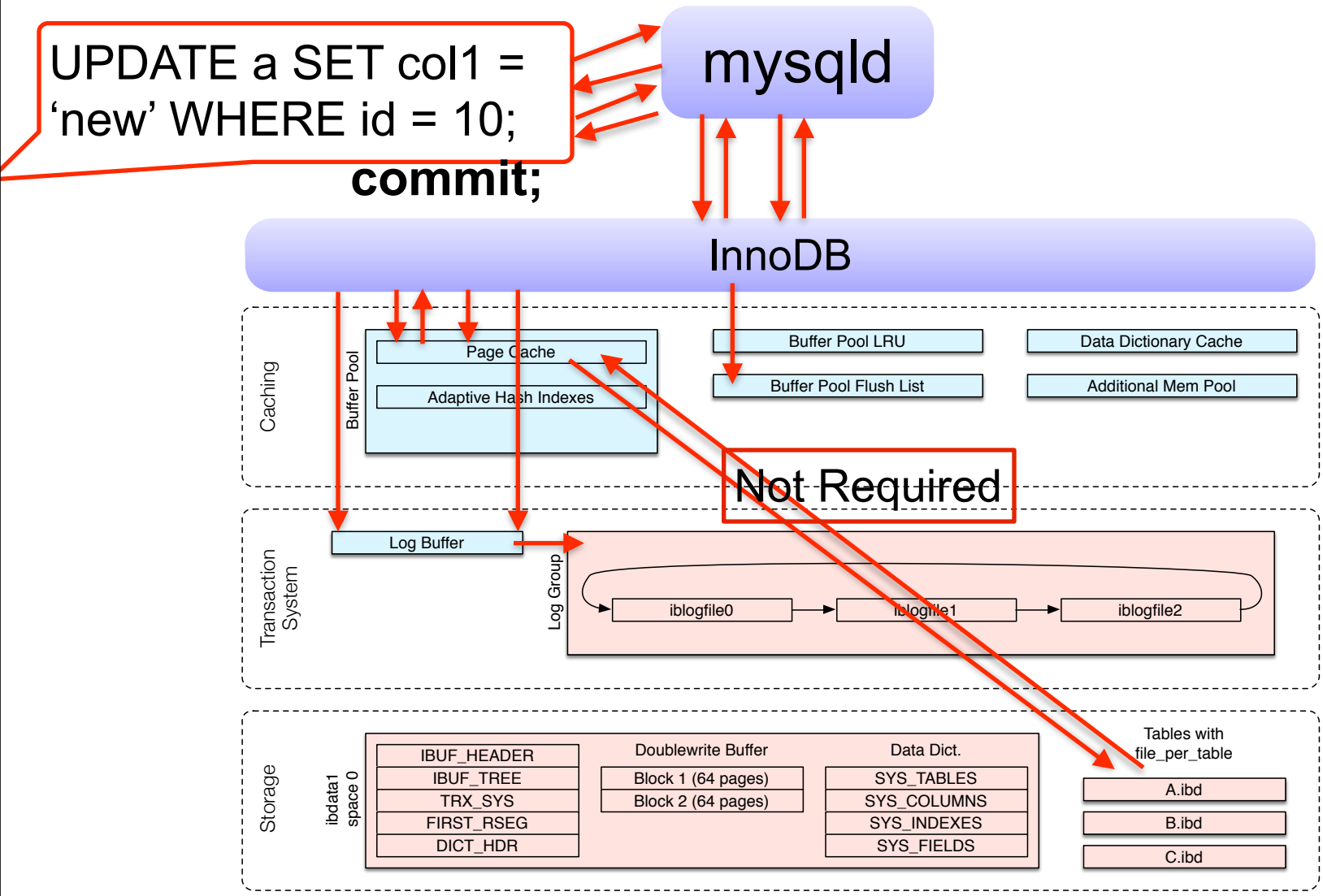
Two more useful features

- **Adaptive hash** - Partial hash index to accelerate secondary key lookups.
- **Change buffering** - when non-unique indexes are not in memory, changes can be temporarily buffered until they are.

Query (by secondary key)



Update Query (large table)



New Features

ORACLE

MySQL 5.5+

- IO Scalability
- Async IO
- Multiple Buffer Pools
- Adaptive Flushing
- Scan Resistant LRU
- Compressed Pages
- CPU Scalability
- Improved Atomics
- Spin Loops with PAUSE

MySQL 5.6+

- LRU Dump and Restore
- Improved Group Commit
- Fulltext Search
- Fast Read-only Transactions
- Memcached Interface
- Information Schema metadata tables
- Persistent Statistics
- Variable Page Size
- Online DDL
- Transportable Tablespace
- Transactional Replication Using InnoDB

MySQL 5.7+

- Faster Temporary Tables
- Index Lock Contention Reduction
- More Online DDL
  - Extend VARCHAR
  - Rename Index
- Improved Read-Only Transactions
- Improved CPU Scalability

Configuration

ORACLE

The Top 3

1. innodb-buffer-pool-size
2. innodb-log-file-size
3. innodb_flush_log_at_trx_commit

innodb-buffer-pool-size

- Really only one major buffer/cache settings to set.
- Responsible for all pages types (data, indexes, undo, insert buffer..)

innodb-buffer-pool-size (cont.)

- Recommendation is 50-80% of RAM.
- Default is 128M of RAM.
- Please allow 5-10% on top for other meta data to grow.

innodb-log-file-size

- Log files are on disk, but this contributes to how many unflushed (dirty) pages you can hold in memory.
- In theory larger log files = longer crash recovery.
  - In MySQL 5.5 -2G max.
  - In MySQL 5.6 - 4G is usually safe.
  - Early versions should be much smaller.
- Default is 48M Log Files.

innodb_flush_log_at_trx_commit

- Default is full ACID Compliance (=1)
- Can be set to 0/2 if you do not mind some data loss.

innodb_flush_log_at_trx_commit

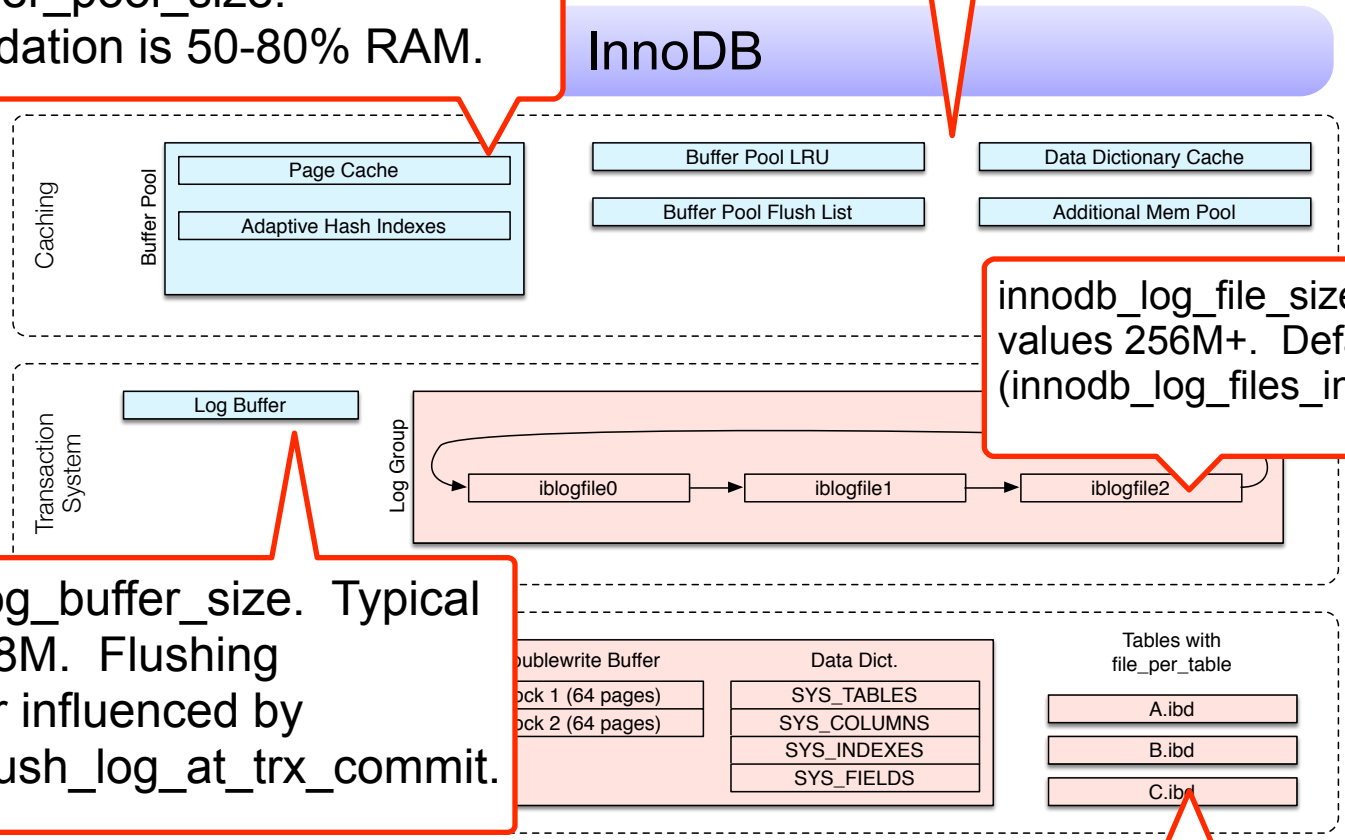
- **0 =** Log buffer written + synced once per second. Nothing done at commit.
- **1 =** Log buffer written + synced once per second + written and synced on commit.
- **2 =** Log buffer written + synced once per second + written (not synced) on commit.

2 is a slightly safer version of **0**.

Basic Configuratio

Requires about 5-10% of buffer pool size as overhead (not directly configurable).

Innodb_buffer_pool_size. Recommendation is 50-80% RAM.



innodb_log_file_size. Typical values 256M+. Default of 2 files (innodb_log_files_in_group).

innodb_log_buffer_size. Typical values 1-8M. Flushing behaviour influenced by innodb_flush_log_at_trx_commit.

innodb_file_per_table (Default: ON in 5.6+)

The ~Top 10

1. innodb-buffer-pool-size
2. innodb-log-file-size
3. innodb-log-buffer-size
4. innodb_flush_log_at_trx_commit
5. innodb_flush_method
6. innodb_flush_neighbors
7. innodb_io_capacity, innodb_io_capacity_max, innodb_lru_scan_depth
8. innodb-buffer-pool-instances
9. innodb_read_io_threads and innodb_write_io_threads

Less-likely to need configuration

- innodb_thread_concurrency
- innodb_concurrency_tickets
- innodb_max_pct_dirty_pages
- innodb_use_native_aio (always on)
- innodb_old_blocks_time (5.6 default: 1000)

Deprecated Settings

- Typically “remove on sight” from config files:
  - innodb_additional_mempool_size
  - innodb_use_sys_malloc

Credits

- InnoDB Architecture Diagrams via https://github.com/jeremycole/innodb_diagrams
- Available under (3-clause) BSD license
  Copyright (c) 2013, Twitter, Inc.
  Copyright (c) 2013, Jeremy Cole <jeremy@jcole.us>
  Copyright (c) 2013, Davi Arnaut <darnaut@gmail.com>