DBMS Implementation : MySQL - Introduction

woonhak.kang

woonagi319@skku.edu

Contents

- Introduction
- Overview
- Run DBMS and Tools
- Misc.
- QnA
- Reference

Introduction

- `woonhak@vldb:~$ whoami`
  - `Ph.D Candidate (5th year)`
- Read open source for fun
  - PostgreSQL, MySQL/InnoDB, SQLite
  - Performance evaluation tools : fio, open source benchmark tools
    - benchmarkSQL, oltpbenchmark, osdl-dbt

Introduction

- How DBMS engines work ?
  - Query processing
  - Optimizer
  - **Storage engine**
  - Transaction manager
- How to ?
  - Manual
  - Source code
  - `gdb and fprintf(stderr, “INFO: XXX”)`
  - `Performance stats`

Introduction

- Plan
  - 1st week
    - Overview
    - Run DBMS and Tools
  - 2nd week
    - Code review for InnoDB initialize procedures (startup database)
  - 3rd week
    - Buffer manager
    - File Storage manager
  - 4th week
    - Transaction manager
    - Log manager
- Term Project
  - Implement something in the MySQL
  - flash optimized techniques, NVM

OVERVIEW

MySQL

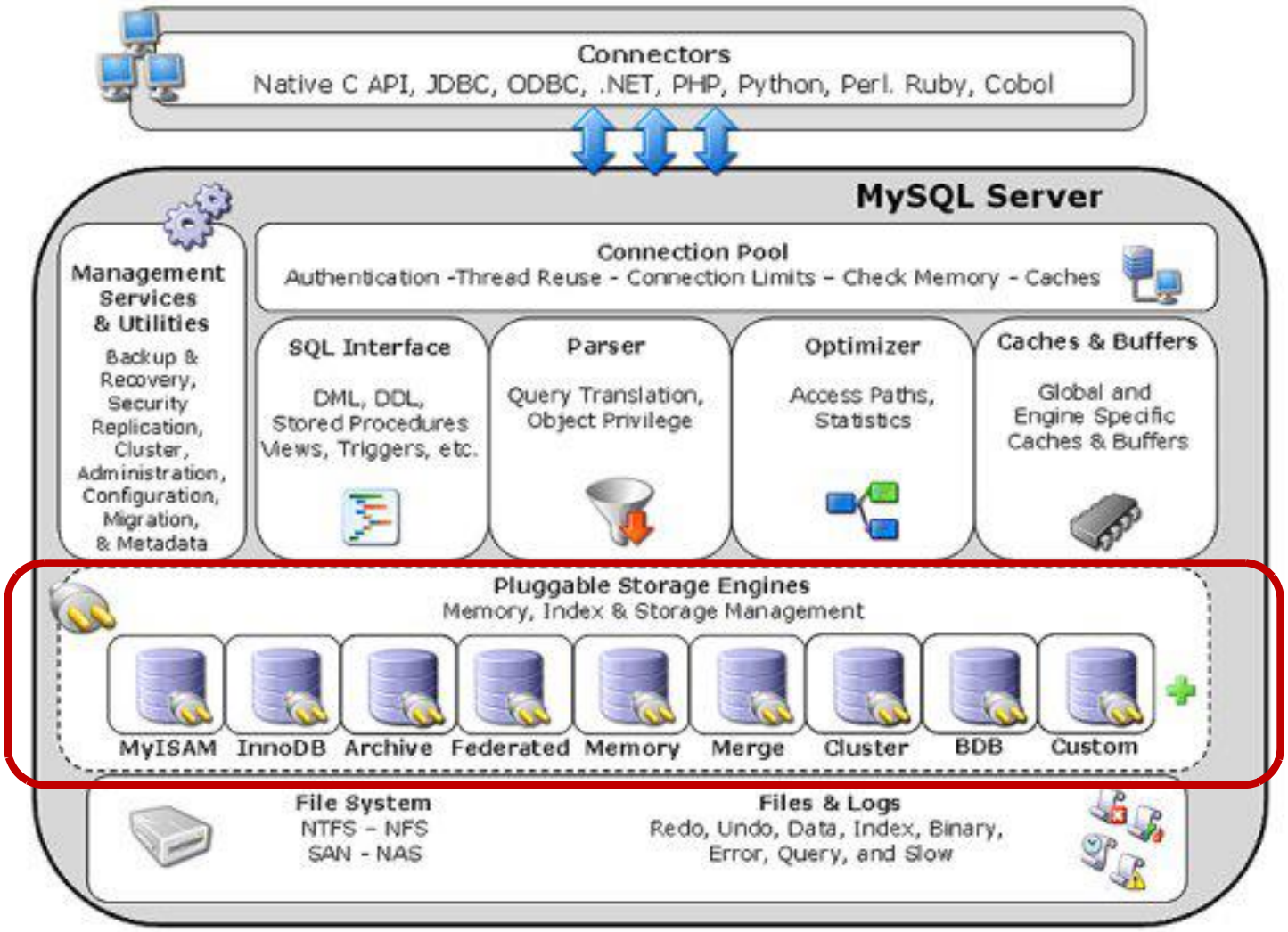
- Database server
- Used in many companies
  - Facebook, Twitter, Google, Pinterest and so on.

259 systems in ranking, April 2015

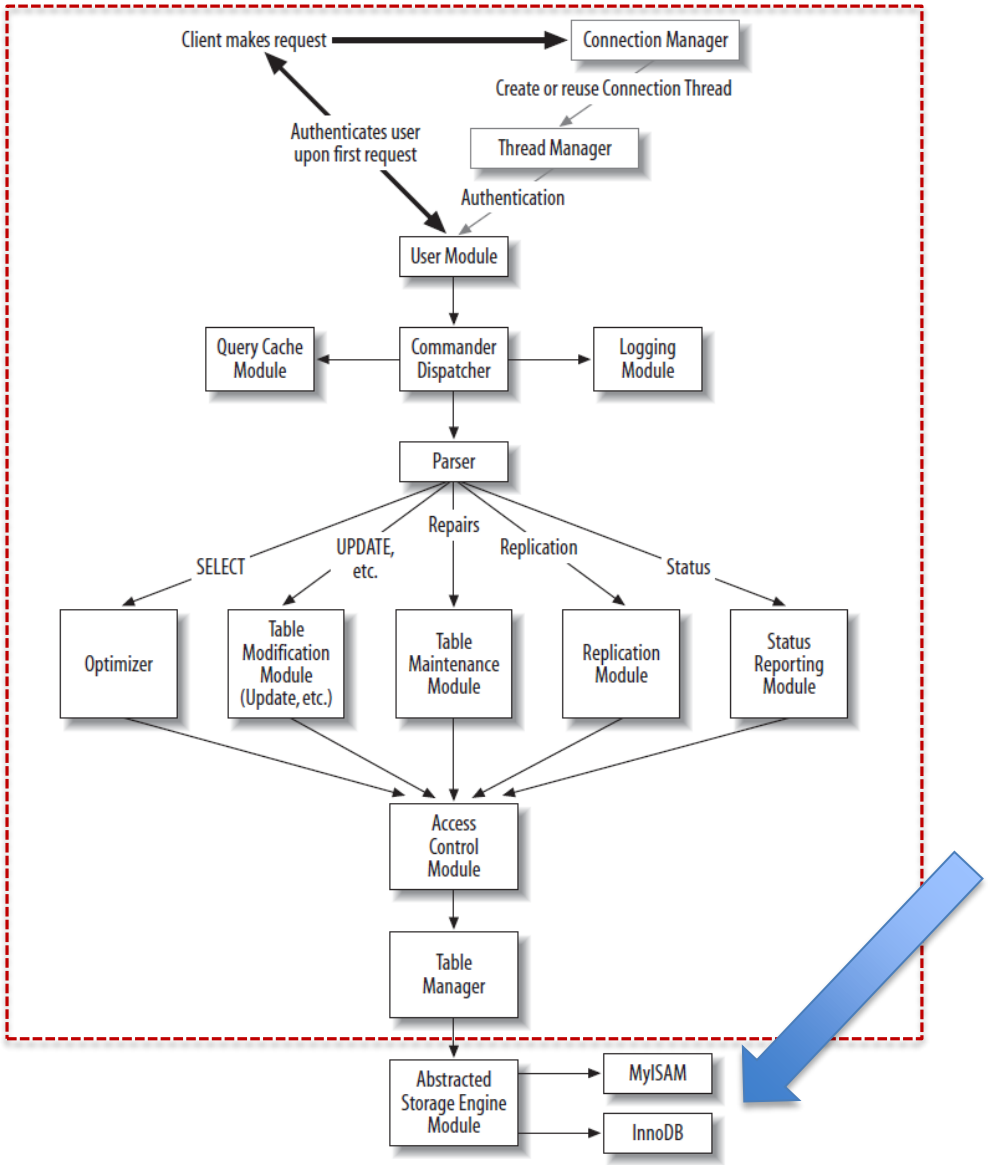
Rank			DBMS	Database Model	Score		
Apr 2015	Mar 2015	Apr 2014			Apr 2015	Mar 2015	Apr 2014
1.	1.	1.	Oracle	Relational DBMS	1446.13	-22.96	-67.95
2.	2.	2.	MySQL	Relational DBMS	1284.58	+23.49	-8.09
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1149.11	-15.68	-61.31
4.	4.	↑ 5.	MongoDB ⊞	Document store	278.59	+3.58	+64.25
5.	5.	↓ 4.	PostgreSQL	Relational DBMS	268.31	+3.88	+38.08
6.	6.	6.	DB2	Relational DBMS	197.65	-1.20	+13.06
7.	7.	7.	Microsoft Access	Relational DBMS	142.19	+0.50	-0.57
8.	8.	↑ 9.	Cassandra ⊞	Wide column store	104.89	-2.42	+26.17
9.	9.	↓ 8.	SQLite	Relational DBMS	102.30	+0.59	+12.13
10.	10.	↑ 13.	Redis	Key-value store	94.55	-2.49	+36.09

source - http://db-engines.com/en/ranking

Big Picture


Connectors
Native C API, JDBC, ODBC, .NET, PHP, Python, Perl. Ruby, Cobol
MySQL Server
Management Services & Utilities
Backup & Recovery, Security Replication, Cluster, Administration, Configuration, Migration, & Metadata
Connection Pool
Authentication -Thread Reuse - Connection Limits - Check Memory - Caches
SQL Interface
DML, DDL, Stored Procedures Views, Triggers, etc.
Parser
Query Translation, Object Privilege
Optimizer
Access Paths, Statistics
Caches & Buffers
Global and Engine Specific Caches & Buffers
Pluggable Storage Engines
Memory, Index & Storage Management
MyISAM
InnoDB
Archive
Federated
Memory
Merge
Cluster
BDB
Custom
File System
NTFS - NFS
SAN - NAS
Files & Logs
Redo, Undo, Data, Index, Binary, Error, Query, and Slow

Query Processing



Pluggable Storage Engines

Table 10-1. MySQL storage engine comparison

	MyISAM	InnoDB	Memory	Merge	NDB	Archive	Federated
Transactions	No	Yes	No	No	Yes	No	No
Indexing	B-tree, R-tree, full text	B-tree	Hash, B-tree	B-tree, R-tree	Hash, B-tree	None	Depends on the remote table engine
Storage	Local disk	Local disk	RAM	Local disk	Remote and local cluster nodes	Local disk	Remote MySQLserver instance
Caching	Key cache	Key and data cache	N/A	Same as MyISAM	Key and data cache	None	Depends on the remote table engine
Locking	Table	Row	Table	Table	Row	Row	Relies on the remote table engine
Foreign keys	No	Yes	No	No	No	No	Depends on the remote table engine

Storage Engine Interface

- MySQL supports several different storage engine
  - to use **the same API : storage engine layer**
- With adding storage engine interface
  - InnoDB could be easily integrated with MySQL
  - InnoDB supports : Transaction, multi-versioning, row-level locking

Storage Engine Interface

- `Handler`
  - **interface between the storage engine and the MySQL Optimizer**
  - abstract class
  - each storage engines implement a subclass of handler
- `Handlerton`
  - After version 5.0
  - added to allow storage engines to provide their own hooks such as **initialization, transaction commit, save-point, rollback** -> not involve one-table
- In this slides
  - Source files : MySQL-5.6.XX

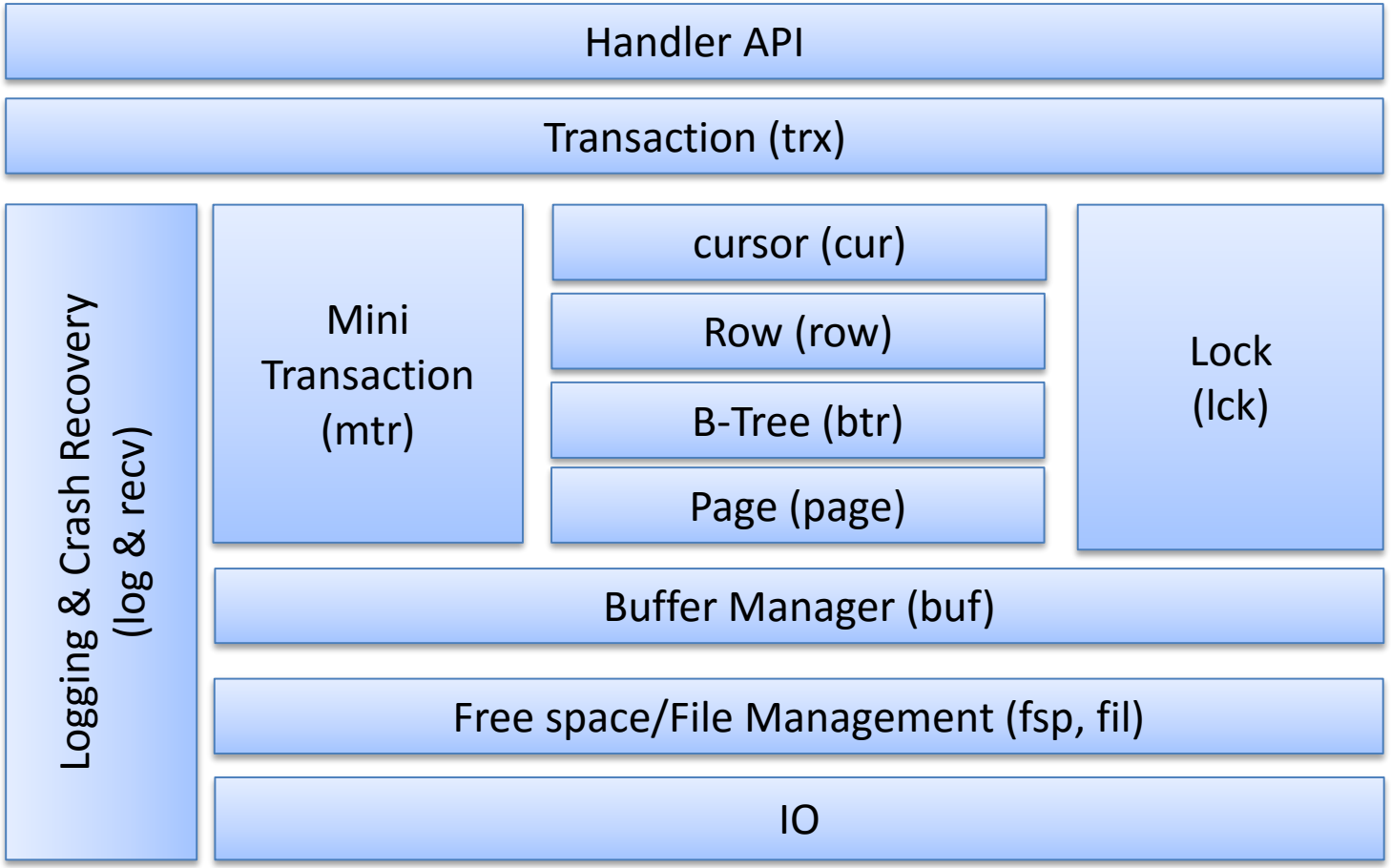
InnoDB

- Until MySQL 5.1, the InnoDB version has been tied closely to the MySQL release
- MySQL 5.1 pluggable storage engine API
  - Developers have increased freedom to make improvements independent of MySQL
- MySQL 5.5, InnoDB engine became default in MySQL

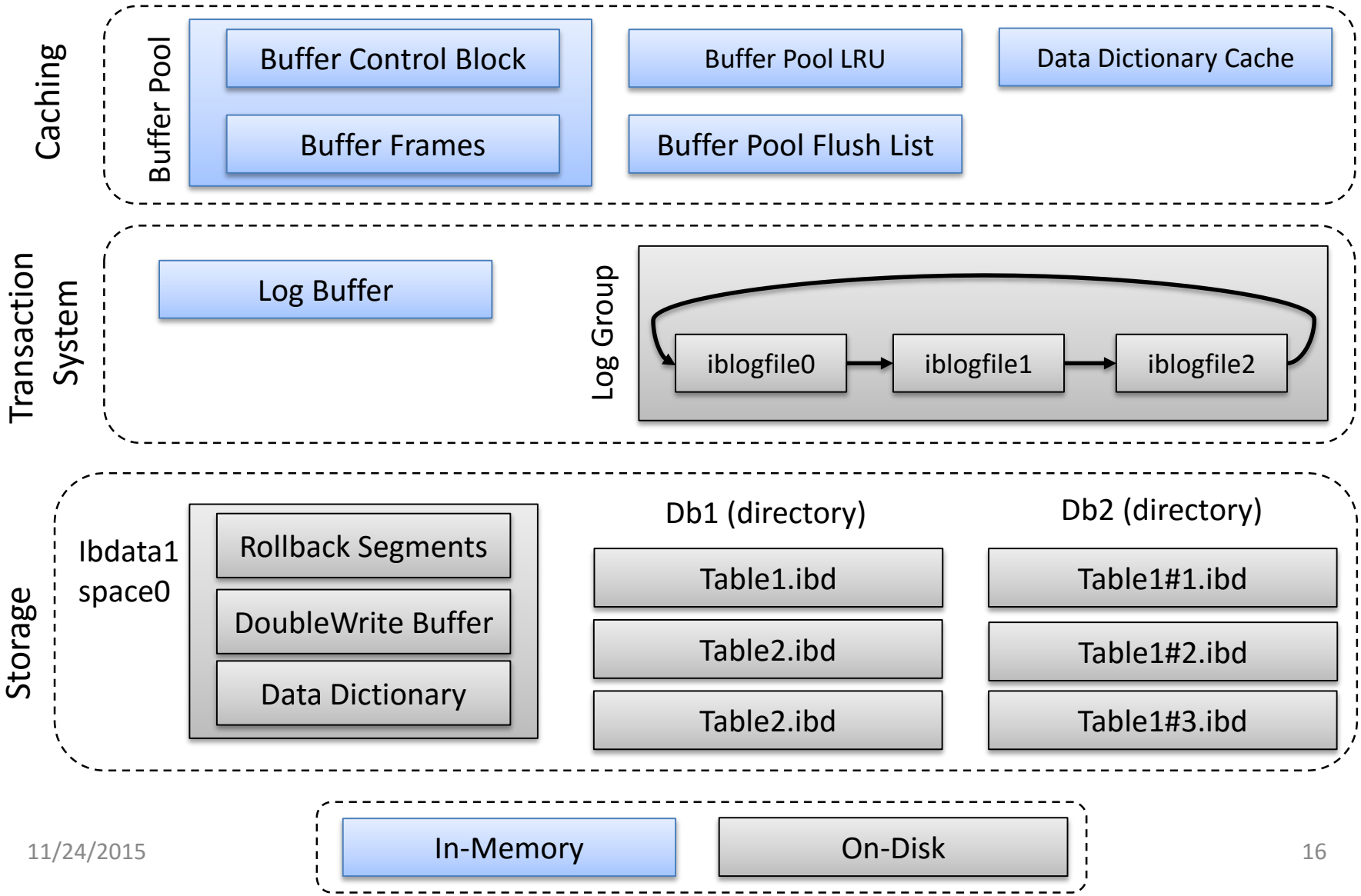
InnoDB Features

- Design
  - Modeled on “Transaction Processing : Concepts & Techniques”
- Transaction support : ACID compliance
- Row-level locking
- MVCC – readers don’t block writers
- Crash Recovery
- Index only scans
- Insert buffering (change buffering)

InnoDB Architecture

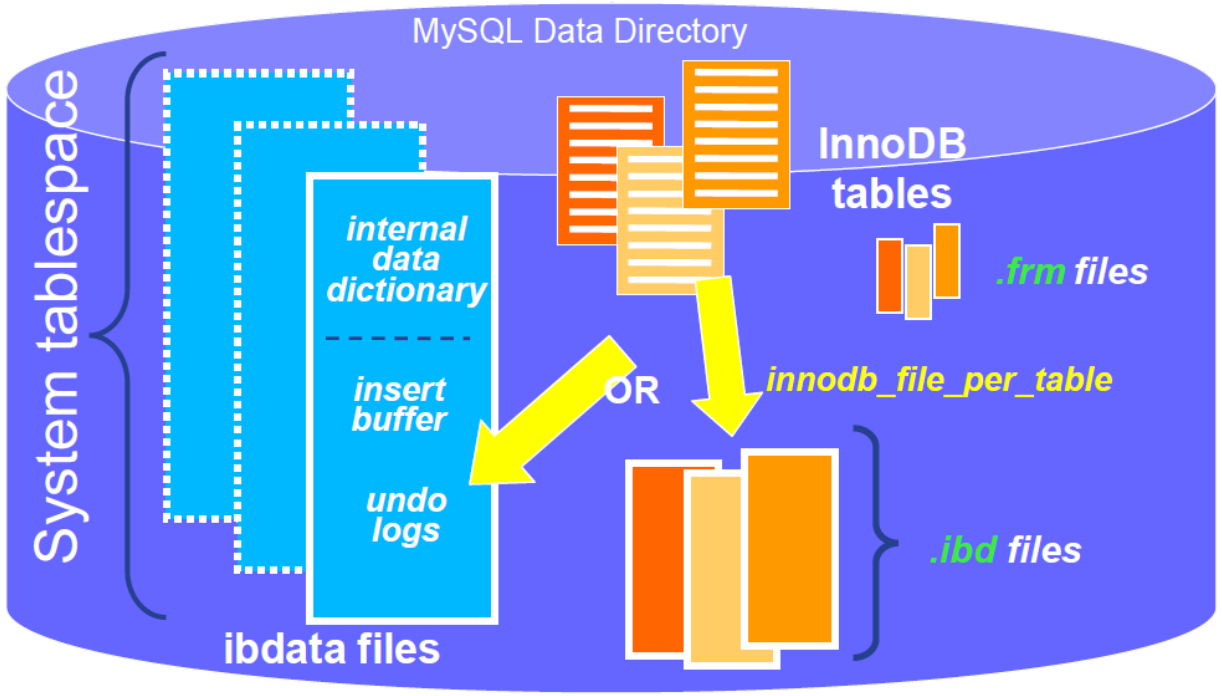


InnoDB High Level Overview



InnoDB Database Files

- Table files : .frm
- Data files : .ibd
  - System tablespace
  - Data tablespace
- Log files : ibX_logfile



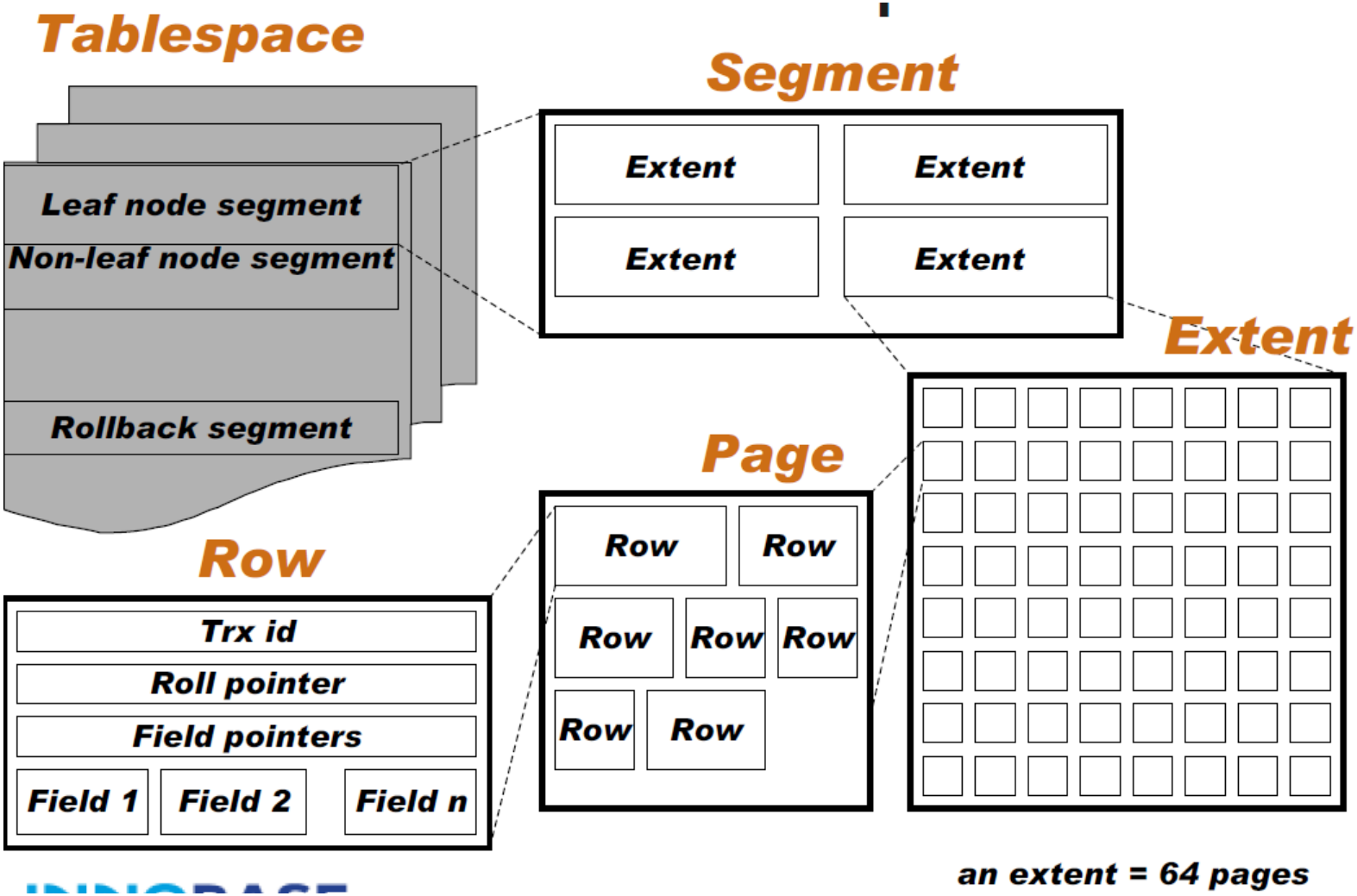
InnoDB Tablespace

- A tablespace consists of multiple files and/or raw disk partitions.
- A file/partition is a collection of segments.
- A segment consists of fixed-length pages.
- Default page size is 16KB in uncompressed tablespaces, and 1KB-16KB in compressed tablespaces (for both data and index)

InnoDB Tablespace

- System Tablespace
  - Internal Data Dictionary (Catalog)
  - Undo
  - Change Buffering
  - Doublewrite Buffer
  - MySQL Replication info

InnoDB Tablespace



InnoDB Tablespace

- Single Table Space/Raw Disk

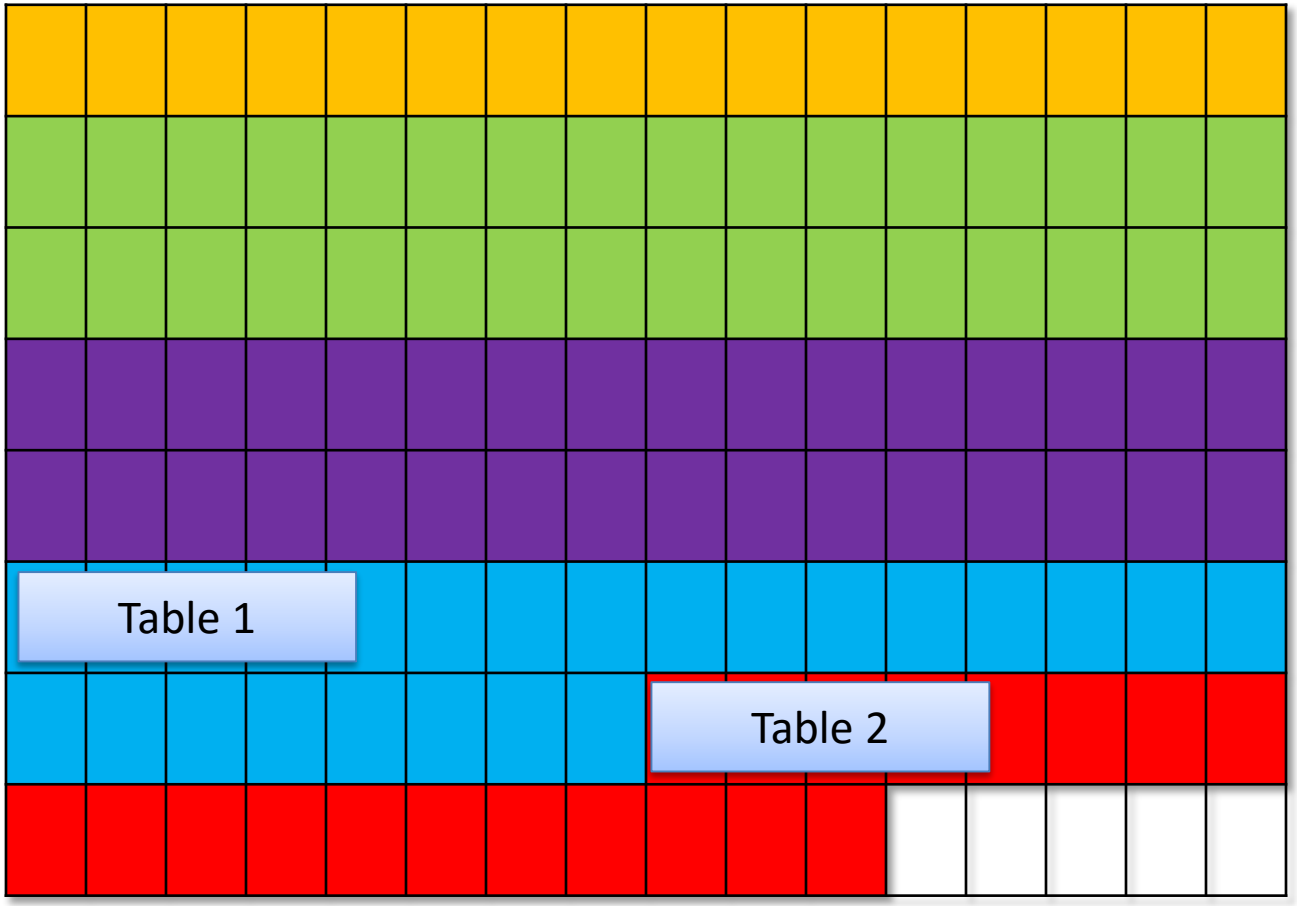


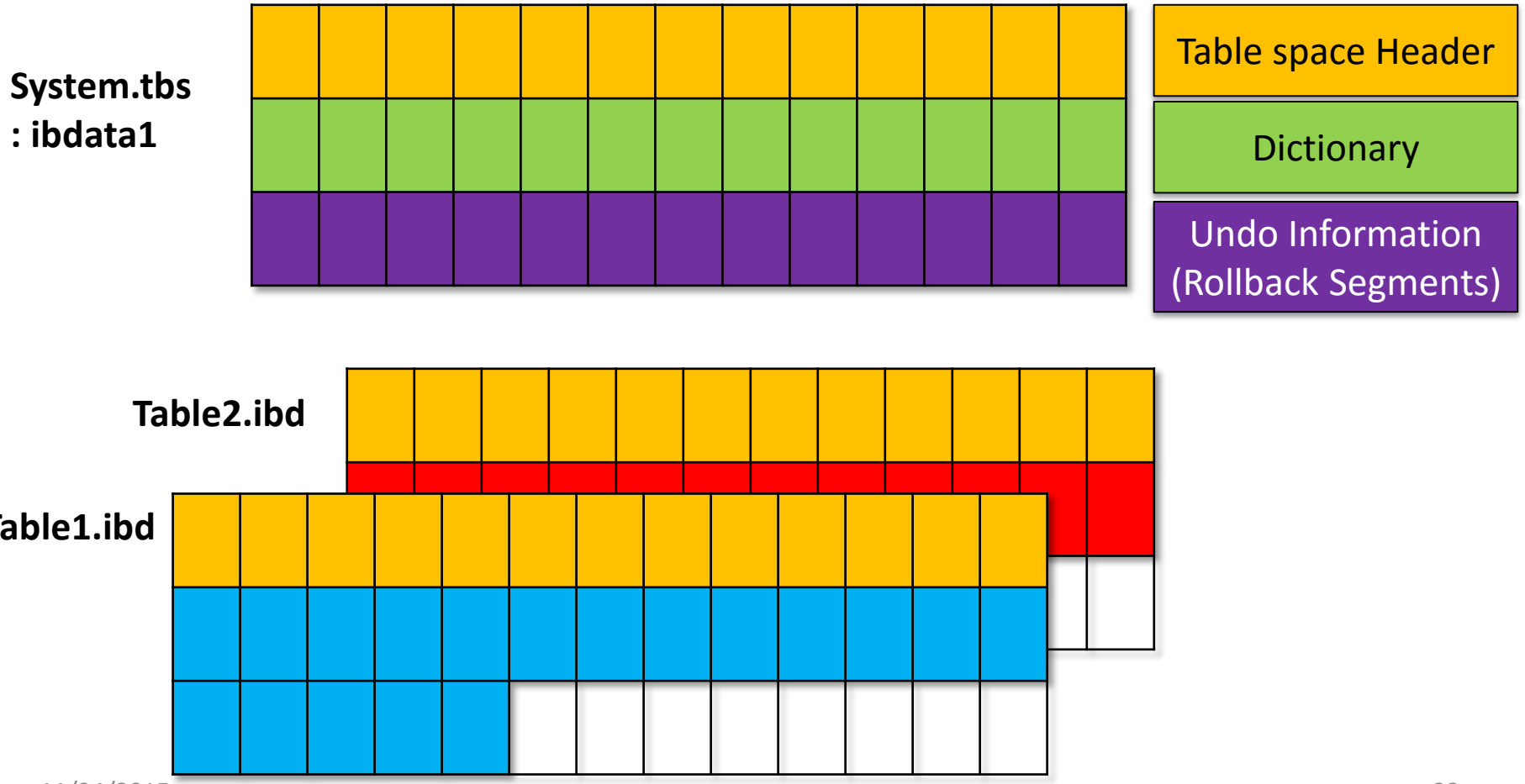
Table space Header

Dictionary

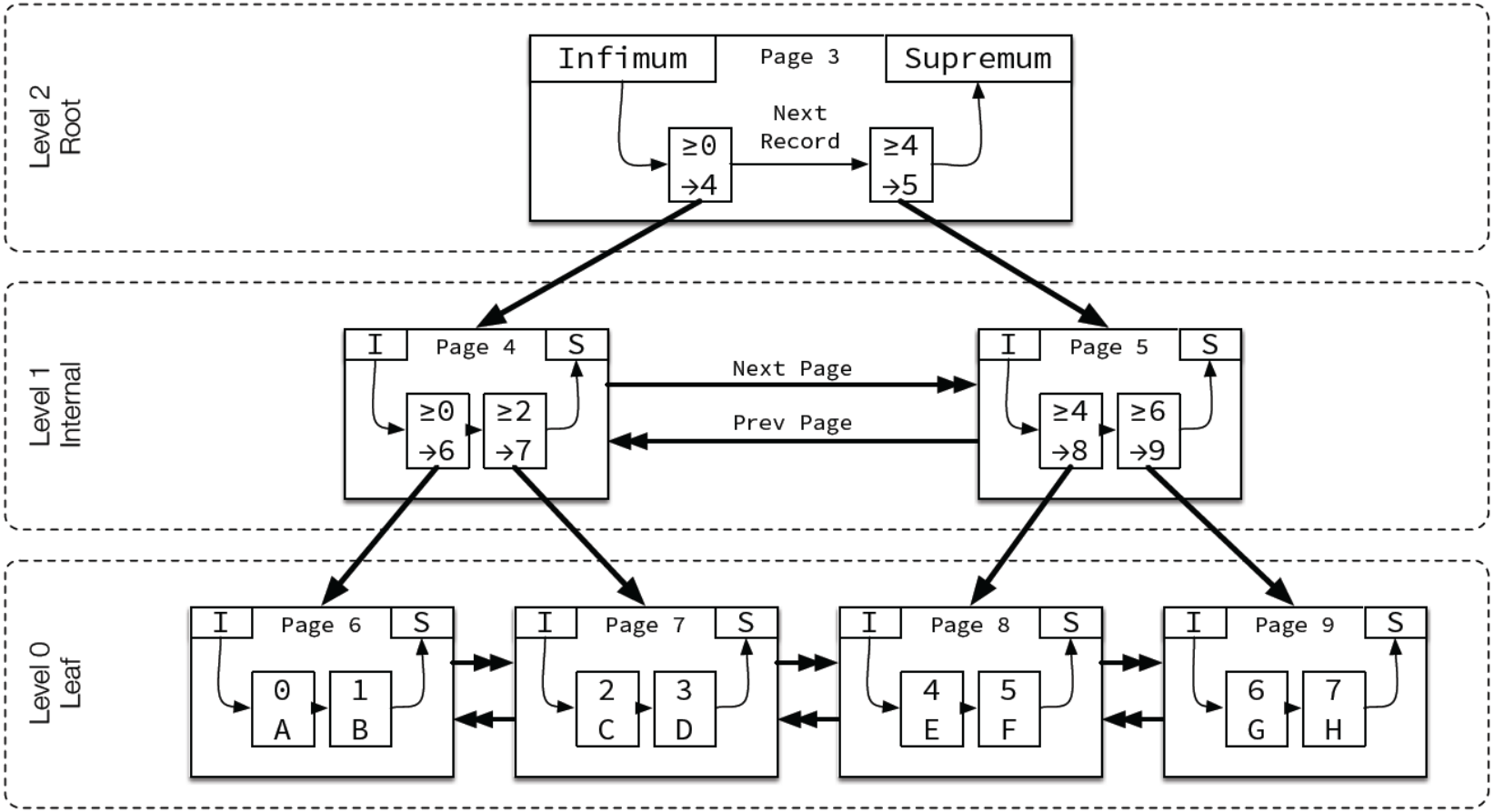
Undo Information (Rollback Segments)

InnoDB Tablespace

- Table Spaces in file_per_table



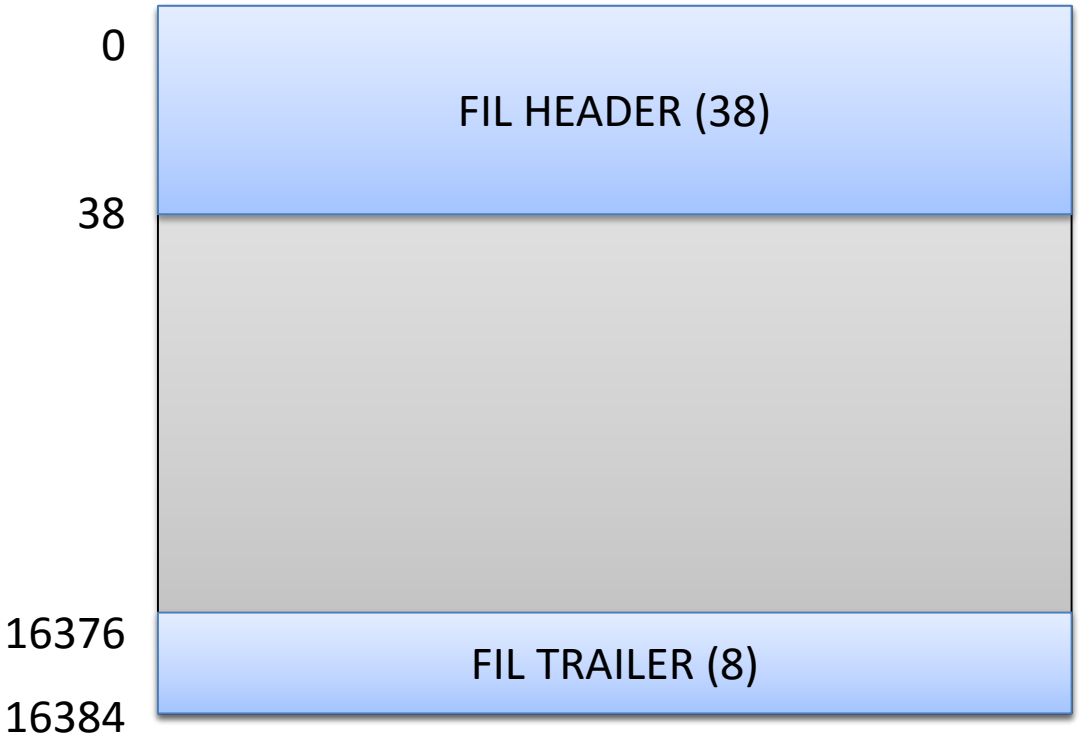
InnoDB Index Structure



Levels are numbered starting from 0 at the leaf pages, incrementing up the tree.
Pages on each level are doubly-linked with previous and next pointers in ascending order by key.
Records within a page are singly-linked with a next pointer in ascending order by key.
Infimum represents a value lower than any key on the page, and is always the first record in the singly-linked list of records.
Supremum represents a value higher than any key on the page, and is always the last record in the singly-linked list of records.
Non-leaf pages contain the minimum key of the child page and the child page number, called a "node pointer".

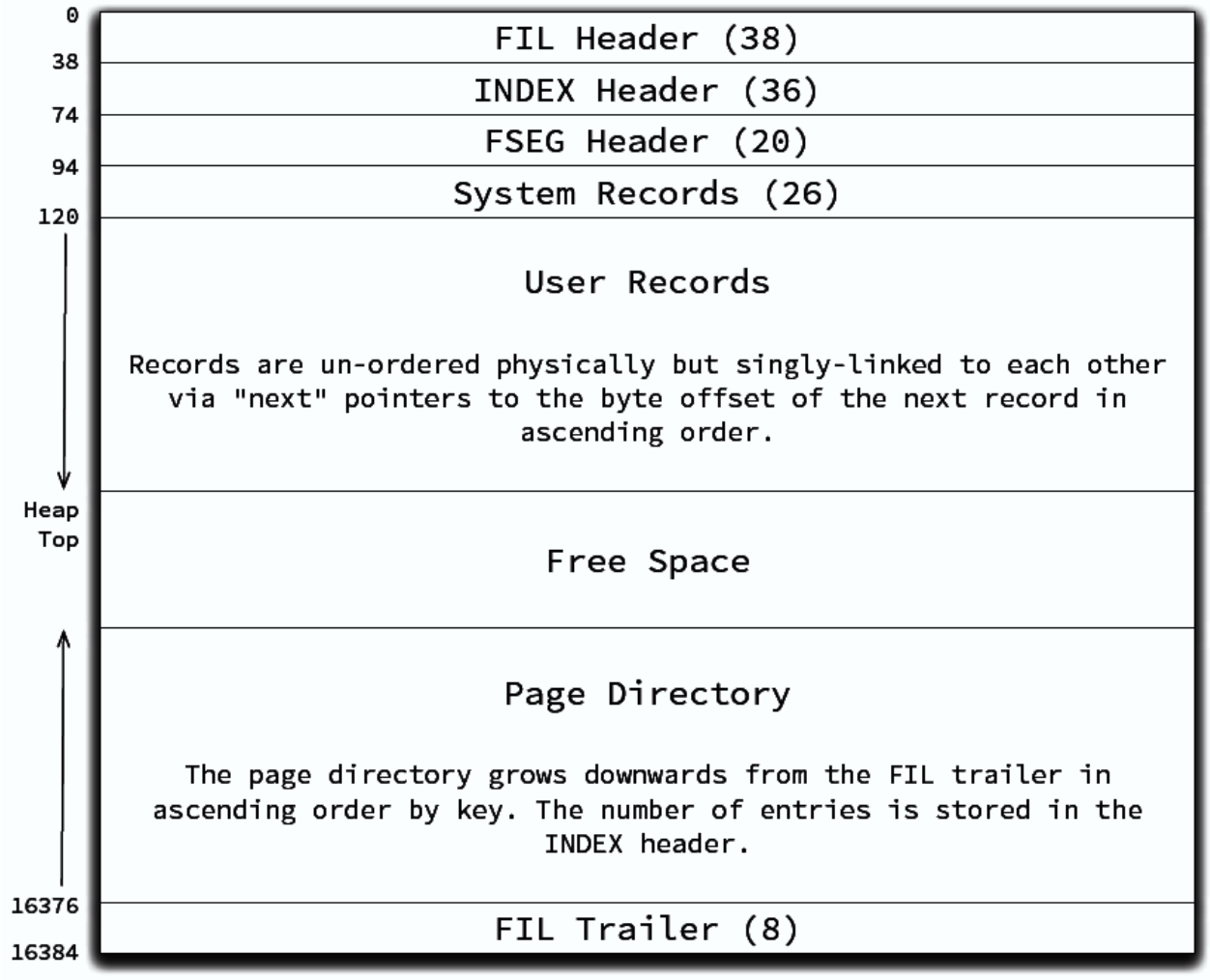
InnoDB Basic Page

- Basic page overview
  - All pages follow this page layout



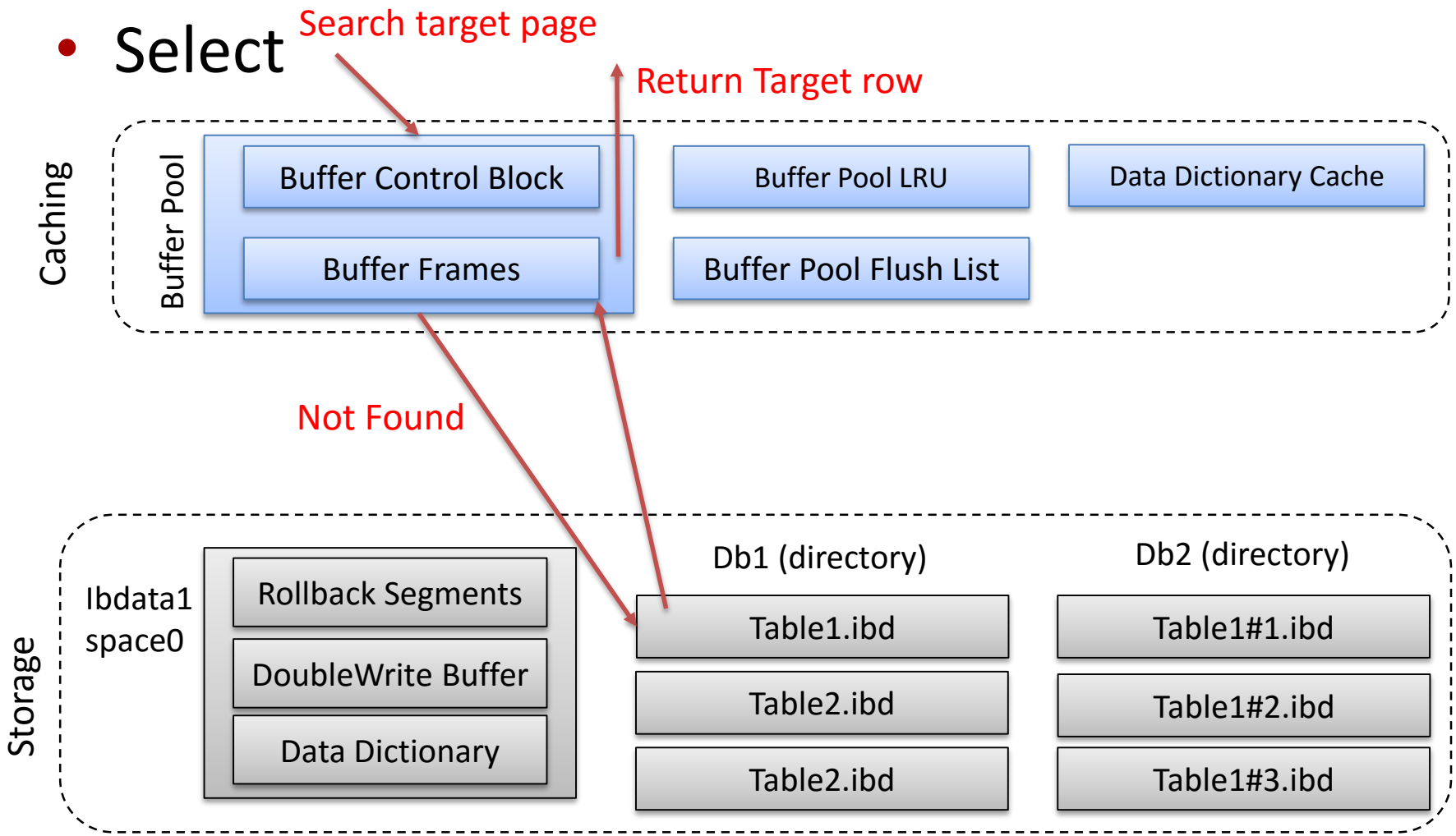
InnoDB Index Page

- Everything is Index in InnoDB



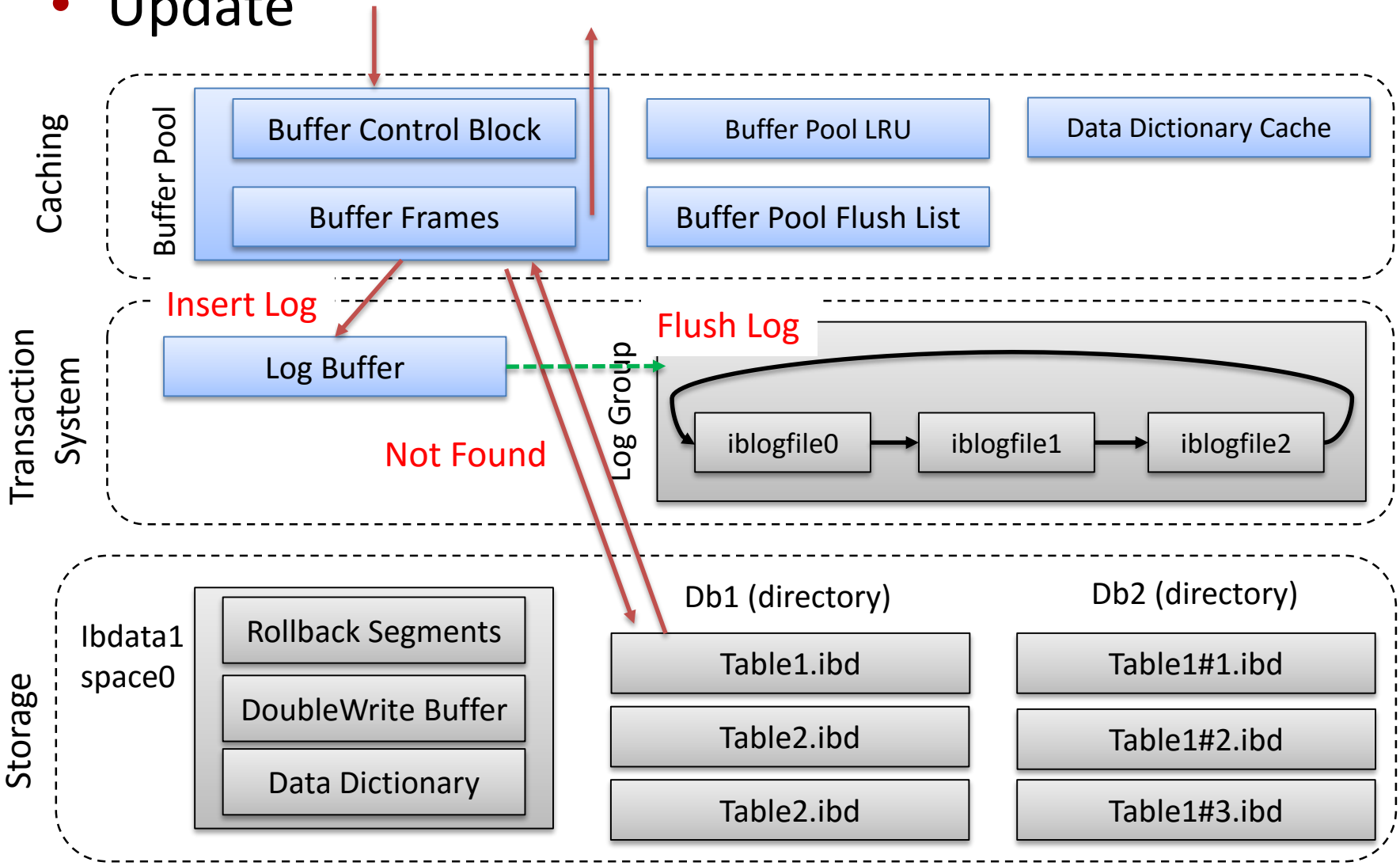
Basic Operations

- Select



Basic Operations

- Update



Memory Management

- Buffer Pool instances
  - Partitioned buffer pools
  - # of buffer pools – for the better concurrency
- Buffer Pool
  - Data pages, index pages, undo records, adaptive hash index
- Log buffer : redo records
- Buffer Replacement
  - Variant of LRU replacement (use mid-point insertion)

Threads

- User threads (MySQL server threads)
  - Process user request
- Master threads
- IO Threads
  - Read IO, Write IO
  - Insert Buffering
  - Log
- Purge threads : garbage collection
- Page cleaner (flusher) thread – background dirty page flushing

Background flusher

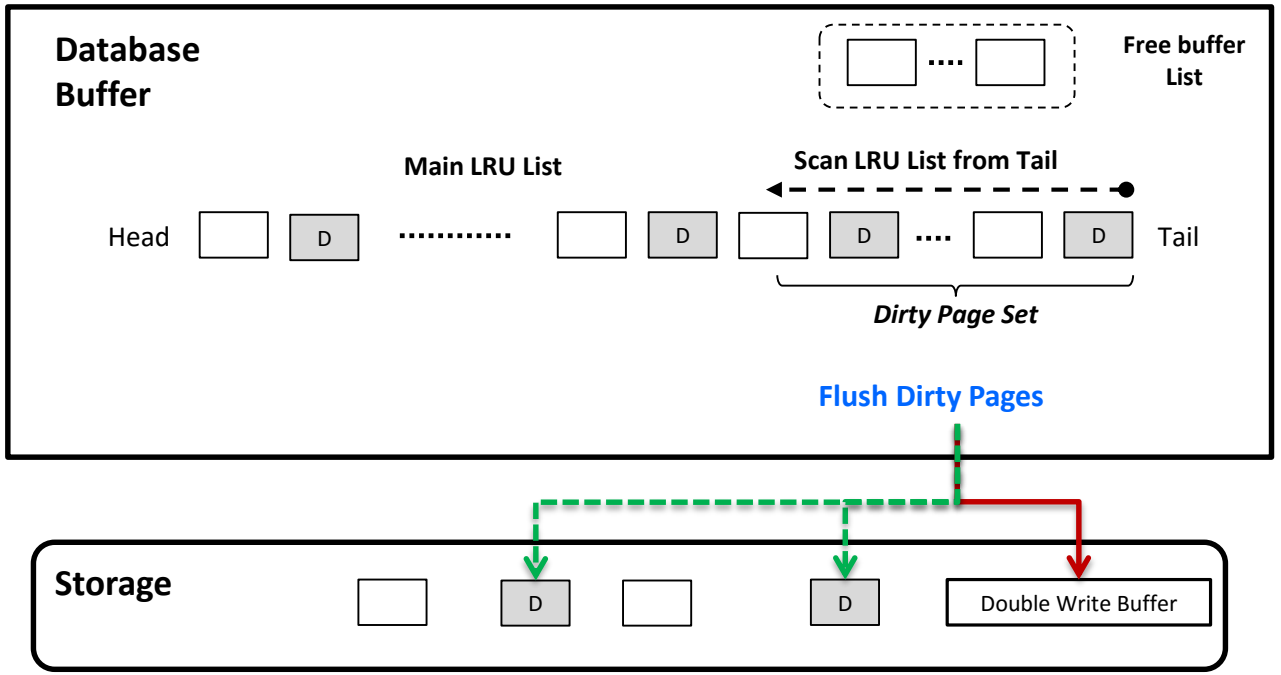
- Activity of writing dirty pages & logs to the disk
- 2 types of flushing
  - LRU flushing : based on LRU_list
  - Adaptive flushing (checkpoint)
    - based on flush_list
    - Strictly ordered on oldest_modification LSN
- Note: On-demand flush to make free buffer
  - Single page flushing

Flushing

- Flushing a batch typically involves:
  - Scanning the tail of the relevant list to find victims
  - Select neighbors as candidates for flushing as well : not good for SSD
  - Copy dirty pages to the doublewrite buffer
  - Writing double write buffer to disk
  - Sync double write buffer
  - Write to data files (use aio)
  - Sync all data files (before write doublewrite buffer again)

Double Write Buffer

- To avoid torn page written problem
- Write dirty pages special storage area in system table space priori to write database file



async IO

- Support Direct IO
  - *innodb_flush_method*
  - Avoid memory copy_from/to_user() overhead
- Two types of async IO
  - Simulated AIO : use multiple IO threads + sync io
  - Linux Native AIO : use multiple IO threads + kernel AIO (libaio + io_submit)
- Read/Write Operations
  - Read op is done by user thread : use pread()
  - Write op is done by AIO
- Log Operations
  - buffered IO + fsync()
  - 512 byte sync IO
- Impose durability
  - Call fdatasync()

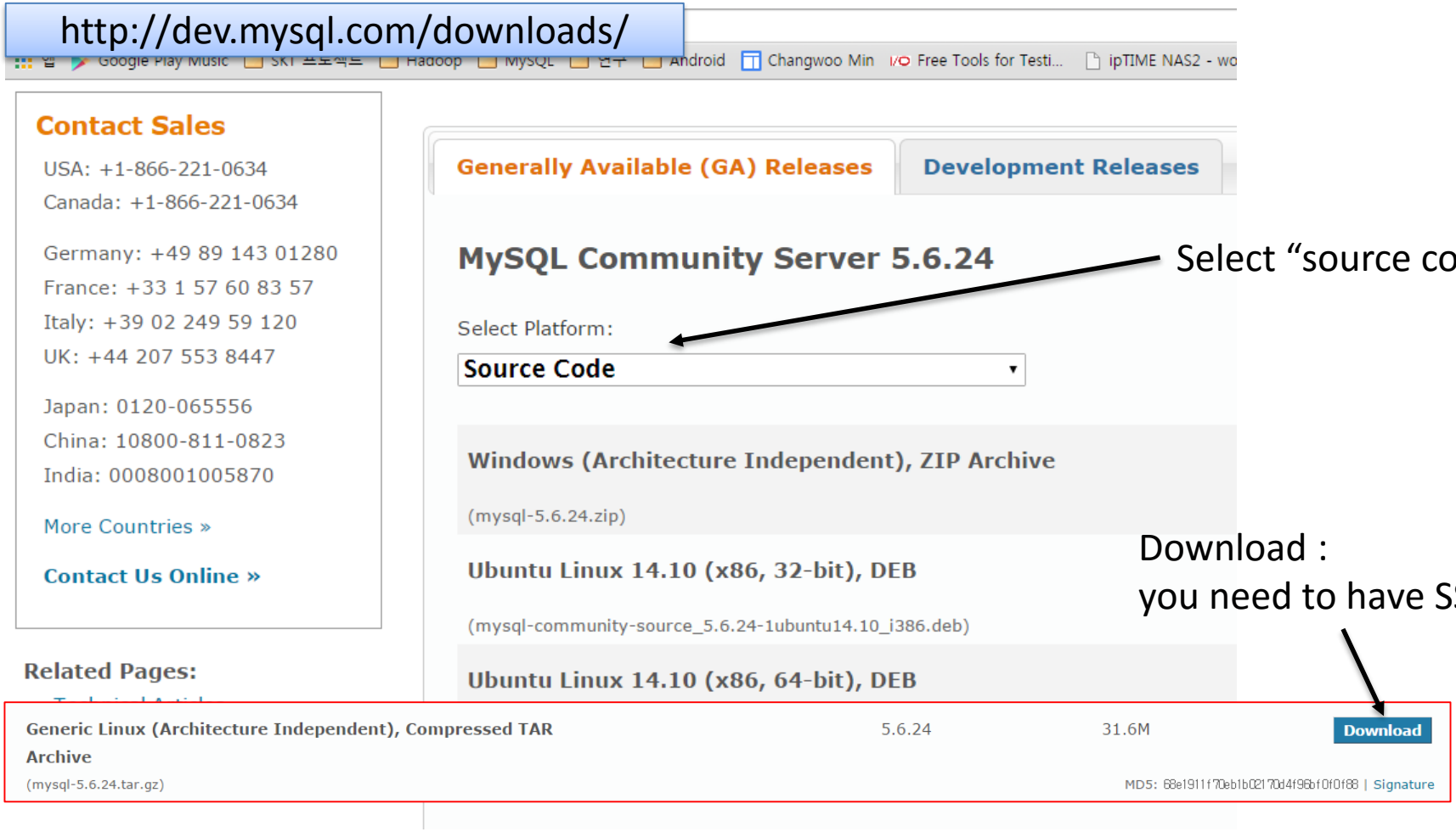
Transaction Log

- ARIES Style Logging and Recovery
- Write Ahead Log (WAL)
- REDO
  - Physiological logging
- UNDO
  - Rollback segments

RUN DBMS AND TOOLS

Download Source Code

http://dev.mysql.com/downloads/



Select “source code”

Download :
you need to have SSO ID

Build & install

- Pre-requisites
  - libreadline
  - libaio
- cmake
  - Change default install directory
    - cmake -DCMAKE_INSTALL_PREFIX=/path/to/dir
- make -j8 (8 : # of cores in your machine)
- make install

Run DBMS and Tools

- Create database files
  - need a configuration file
  - default config file path
    - /etc/mysql/my.cnf
    - $HOME/.my.cnf
  - use install db script
- Run database server
  - mysqld_safe --defaults-file=/path/to/my.cnf
- Tools
  - Performance stats

Configuration File

- default-storage-engine = innodb
- basedir = /path/to/msyql_bin/dir
- datadir = /path/to/data/dir
- ##settings for data file
- innodb_data_file_path=ibdata1:1G:autoextend
- innodb_file_per_table=1 #file per table ON
- innodb_buffer_pool_size=4GB #buffer settings
- innodb_buffer_pool_instances=4
- innodb_log_file_size=2G #transaction log settings
- innodb_log_files_in_group=3
- # 0:every 1 seconds, 1:fsync on commits, 2:writes on commits
- innodb_log_buffer_size=32M
- innodb_flush_method=O_DIRECT
- innodb_use_native_aio=true #AIO control
- #Log group path (iblog0, iblog1)
- innodb_log_group_home_dir=/path/to/log/dir/

Create Default Database Files

- echo $MYSQL_BIN
  - /home/woonhak/bin/mysql/bin
- $> cd $MYSQL_BIN
- $> ./scripts/mysql_install_db --defaults-file=~/path/to/my.cnf
  - During this command
    - create system table space and log files
  - It takes a few minutes

```
To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

  /home/woonhak/bin/mysql//bin/mysqladmin -u root password 'new-pass
  /home/woonhak/bin/mysql//bin/mysqladmin -u root -h woonhak-ututntu

Alternatively you can run:

  /home/woonhak/bin/mysql//bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default.  This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:

  cd . ; /home/woonhak/bin/mysql//bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl

  cd mysql-test ; perl mysql-test-run.pl

Please report any problems with the ./bin/mysqlbug script!

The latest information about MySQL is available on the web at

  http://www.mysql.com

Support MySQL by buying support/licenses at http://shop.mysql.com
```

Run Database Server

- Startup
  - $>mysqld_safe --defaults-file=/path/to/my.cnf

```
woonhak@woonhak-utuntu:~/bin/mysql$ cd
woonhak@woonhak-utuntu:~$ mysqld_safe --defaults-file=./.my.cnf
150427 23:50:36 mysqld_safe Logging to '/home/woonhak/mysql_data/mysql_error.log'.
150427 23:50:36 mysqld_safe Starting mysqld daemon with databases from /home/woonhak/mysql_data/
```

- Connect
  - $>mysqld_safe --defaults-file=/path/to/my.cnf

```
woonhak@woonhak-utuntu:~$ mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.2-m12 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

root:(none)>
```

Monitoring Tools

- SHOW ENGINE INNODB STATUS\G
  - InnoDB Monitor
- Performance schema for InnoDB
- Information schema tables
  - Information schema metrics table
  - Information schema for InnoDB system tables
  - Information schema for InnoDB buffer pool

InnoDB show engine status

- Monitor output
  - Background thread
  - Semaphores
  - LATEST FOREIGN KEY ERROR
  - LATEST DETECTED DEADLOCK
  - TRANSACTIONS
  - FILE I/O
  - LOG
  - BUFFER POOL AND MEMORY
  - ROW OPERATIONS

```
root:(none)> show engine innodb status \G
*************************** 1. row ***************************
  Type: InnoDB
  Name:
Status:
=====================================
2015-04-27 23:01:43 0x7f8b04db3700 INNODB MONITOR OUTPUT
=====================================
Per second averages calculated from the last 50 seconds
-----------------
BACKGROUND THREAD
-----------------
srv_master_thread loops: 594 srv_active, 0 srv_shutdown, 4 srv_idle
srv_master_thread log flush and writes: 598
----------
SEMAPHORES
----------
OS WAIT ARRAY INFO: reservation count 7869168
--Thread 140235068241664 has waited at btr0cur.cc line 789 for 0.00 seconds the se
SX-lock on RW-latch at 0x7f8a4c025be0 created in file dict0dict.cc line 2521
a writer (thread id 140235181713152) has reserved it in mode  SX
number of readers 11, waiters flag 1, lock_word: fffffff5
Last time read locked in file btr0cur.cc line 810
Last time write locked in file /home/woonhak/workspace/mysql-5.7.2-m12/storage/inn
--Thread 140235075430144 has waited at btr0cur.cc line 789 for 0.00 seconds the se
SX-lock on RW-latch at 0x7f8a4c025be0 created in file dict0dict.cc line 2521
a writer (thread id 140235181713152) has reserved it in mode  SX
number of readers 11, waiters flag 1, lock_word: fffffff5
Last time read locked in file btr0cur.cc line 810
Last time write locked in file /home/woonhak/workspace/mysql-5.7.2-m12/storage/inn
OS WAIT ARRAY INFO: signal count 1954822
Mutex spin waits 0, rounds 0, OS waits 0
RW-shared spins 2955267, rounds 4358927, OS waits 1403660
RW-excl spins 70420, rounds 284549975, OS waits 150531
RW-sx spins 88218, rounds 5244320, OS waits 160491
Spin rounds per wait: 0.00 mutex, 1.47 RW-shared, 4040.76 RW-excl, 59.45 RW-sx
------------------------
LATEST FOREIGN KEY ERROR
```

Performance Schema

- mysql> use performance_schema;

```
root:(none)> use performance_schema ;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
root:performance_schema> show tables ;
+----------------------------------------------------+
| Tables_in_performance_schema                       |
+----------------------------------------------------+
| accounts                                           |
| cond_instances                                     |
| events_stages_current                              |
| events_stages_history                              |
| events_stages_history_long                         |
| events_stages_summary_by_account_by_event_name     |
| events_stages_summary_by_host_by_event_name        |
| events_stages_summary_by_thread_by_event_name      |
| events_stages_summary_by_user_by_event_name        |
| events_stages_summary_global_by_event_name         |
| events_statements_current                          |
| events_statements_history                          |
| events_statements_history_long                     |
| events_statements_summary_by_account_by_event_name |
| events_statements_summary_by_digest                |
| events_statements_summary_by_host_by_event_name    |
| events_statements_summary_by_program               |
| events_statements_summary_by_thread_by_event_name  |
| events_statements_summary_by_user_by_event_name    |
| events_statements_summary_global_by_event_name     |
| events_waits_current                               |
| events_waits_history                               |
| events_waits_history_long                          |
| events_waits_summary_by_account_by_event_name      |
| events_waits_summary_by_host_by_event_name         |
| events_waits_summary_by_instance                   |
| events_waits_summary_by_thread_by_event_name       |
| events_waits_summary_by_user_by_event_name         |
| events_waits_summary_global_by_event_name          |
| file_instances                                     |
| file_summary_by_event_name                         |
| file_summary_by_instance                           |
| host_cache                                         |
```

https://dev.mysql.com/doc/refman/5.6/en/performance-schema.html

Performance Schema

- Example)
  - SELECT DISTINCT(name) FROM threads WHERE name LIKE "%innodb%";

```
root:performance_schema> SELECT DISTINCT(name) FROM threads WHERE name LIKE "%innodb%";
+---------------------------------------+
| name                                  |
+---------------------------------------+
| thread/innodb/io_ibuf_thread          |
| thread/innodb/io_log_thread           |
| thread/innodb/io_read_thread          |
| thread/innodb/io_write_thread         |
| thread/innodb/srv_error_monitor_thread |
| thread/innodb/srv_lock_timeout_thread |
| thread/innodb/srv_monitor_thread      |
| thread/innodb/srv_master_thread       |
| thread/innodb/srv_purge_thread        |
| thread/innodb/page_cleaner_thread     |
+---------------------------------------+
10 rows in set (0.00 sec)
```

Performance Schema in InnoDB

```
root:test> SHOW ENGINE PERFORMANCE_SCHEMA STATUS
    -> ;
+--------------------+-------------------------------------------------------------+-----------+
| Type               | Name                                                        | Status    |
+--------------------+-------------------------------------------------------------+-----------+
| performance_schema | events_waits_current.size                                   | 184       |
| performance_schema | events_waits_current.count                                  | 2268      |
| performance_schema | events_waits_history.size                                   | 184       |
| performance_schema | events_waits_history.count                                  | 3780      |
| performance_schema | events_waits_history.memory                                 | 695520    |
| performance_schema | events_waits_history_long.size                              | 184       |
| performance_schema | events_waits_history_long.count                             | 1000      |
| performance_schema | events_waits_history_long.memory                            | 184000    |
| performance_schema | (pfs_mutex_class).size                                      | 256       |
| performance_schema | (pfs_mutex_class).count                                     | 200       |
| performance_schema | (pfs_mutex_class).memory                                    | 51200     |
| performance_schema | (pfs_rwlock_class).size                                     | 320       |
| performance_schema | (pfs_rwlock_class).count                                    | 30        |
| performance_schema | (pfs_rwlock_class).memory                                   | 9600      |
| performance_schema | (pfs_cond_class).size                                       | 256       |
| performance_schema | (pfs_cond_class).count                                      | 80        |
| performance_schema | (pfs_cond_class).memory                                     | 20480     |
| performance_schema | (pfs_thread_class).size                                     | 192       |
| performance_schema | (pfs_thread_class).count                                    | 50        |
| performance_schema | (pfs_thread_class).memory                                   | 9600      |
| performance_schema | (pfs_file_class).size                                       | 320       |
| performance_schema | (pfs_file_class).count                                      | 50        |
| performance_schema | (pfs_file_class).memory                                     | 16000     |
| performance_schema | mutex_instances.size                                        | 128       |
| performance_schema | mutex_instances.count                                       | 5918      |
| performance_schema | mutex_instances.memory                                      | 757504    |
| performance_schema | rwlock_instances.size                                       | 192       |
```

https://dev.mysql.com/doc/refman/5.6/en/performance-schema.html

Information scheme

- INFORMATION_SCHEMA tables in InnoDB
  - Data Dictionary related
  - FTS related
  - Compression related
  - Buffer Pool related
  - Locks / Transactions
  - General Statistics gold mine (metrics table)

Information scheme

- mysql$> use information_schema;

```
root:(none)> use information_schema;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
root:information_schema> select DISTINCT subsystem from innodb_metrics
ystem;
+---------------------+
| subsystem           |
+---------------------+
| adaptive_hash_index |
| buffer              |
| buffer_page_io      |
| change_buffer       |
| compression         |
| ddl                 |
| dml                 |
| file_system         |
| icp                 |
| index               |
| lock                |
| metadata            |
| os                  |
| purge               |
| recovery            |
| server              |
| transaction         |
+---------------------+
17 rows in set (0.00 sec)

root:information_schema>
```

Information scheme

- example) get avg performance counter
  - msqyl$> select name, subsystem, count, avg_count from information_schema.innodb_metrics order by name;

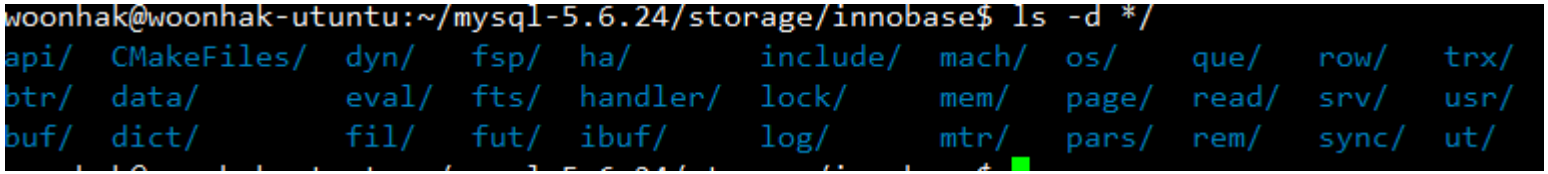
```
root:information_schema> root:information_schema> select name, subsystem, count, avg_count from inform
cs order by name;db_metri
+----------------------------------------+--------------------+-----------+----------------------+
| name                                   | subsystem          | count     | avg_count            |
+----------------------------------------+--------------------+-----------+----------------------+
| adaptive_hash_pages_added              | adaptive_hash_index |         0 |                 NULL |
| adaptive_hash_pages_removed            | adaptive_hash_index |         0 |                 NULL |
| adaptive_hash_rows_added               | adaptive_hash_index |         0 |                 NULL |
| adaptive_hash_rows_deleted_no_hash_entry | adaptive_hash_index |         0 |                 NULL |
| adaptive_hash_rows_removed             | adaptive_hash_index |         0 |                 NULL |
| adaptive_hash_rows_updated             | adaptive_hash_index |         0 |                 NULL |
| adaptive_hash_searches                 | adaptive_hash_index |         0 |                    0 |
| adaptive_hash_searches_btree           | adaptive_hash_index |         0 |                 NULL |
| buffer_data_reads                      | buffer             |    995328 |      681.264887063655 |
| buffer_data_written                    | buffer             |    505344 |    345.88911704312113 |
| buffer_flush_adaptive                  | buffer             |         0 |                 NULL |
| buffer_flush_adaptive_pages            | buffer             |         0 |                 NULL |
| buffer_flush_adaptive_total_pages      | buffer             |         0 |                 NULL |
| buffer_flush_avg_page_rate             | buffer             |         0 |                 NULL |
| buffer_flush_background                | buffer             |         0 |                 NULL |
```

MISC.

Source Tree

- innodb - storage/innodb/

Name	Description	Name	Description
btr	Btree/cursor	os	OS related (system call)
dict	Dictionary	page	Page layout
fil	File mgmt.	log	Log management
fsp	Free space mgmt.	mach	Memory Architecture (endian)
lock	Lock mgmt.	row	Row handle
mtr	Mini transaction	srv	InnoDB server management
trx	Transaction system		



Q&A

- Any Questions ?

Reference

- Understanding MySQL Internals, O'Reilly
- An Introduction to InnoDB Internals, Justin Swanhart, Percona Live
- Jeremy cole blog, MySQL expert, http://blog.jcole.us/ (innodb diagrams : https://github.com/jeremycole/innodb_diagrams)
- Source Code : MySQL Community Server 5.6.15