

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.

Новые возможности в MySQL 5.6

Дмитрий Ленев, Света Смирнова, Москва, 2012



<http://www.devconf.ru>

Актуальные версии MySQL

- 5.1 “старая” стабильная ветка
 - готова к использованию с ноября 2008
 - последняя версия 5.1.63 (апрель 2012)
- 5.5 “новая” стабильная ветка
 - готова к использованию с декабря 2010
 - последняя версия 5.5.24 (май 2012)
- 5.6 текущая development ветка
 - 5.6.5-m8 - milestone release (апрель 2012)
- Версии на MySQL Labs

Основные направления улучшений в MySQL 5.6

- Оптимизатор/выполнение запросов
- InnoDB
- Репликация
- Расширение типов и новые операции
- performance_schema
- Memcached API для INNODB

Улучшения в оптимизаторе/выполнении запросов

- Удобство в использовании
- Скорость

Улучшения в оптимизаторе/выполнении запросов (удобство в использовании)

- EXPLAIN для INSERT/UPDATE/DELETE
 - mysql> explain insert into salaries(emp_no, from_date) select emp_no, min(from_date) from titles group by emp_no\G
 - ***** 1. row *****
 - id: 1
 - select_type: SIMPLE
 - table: titles
 - type: index
 - possible_keys: NULL
 - key: emp_no

Улучшения в оптимизаторе/выполнении запросов (удобство в использовании)

- EXPLAIN для INSERT/UPDATE/DELETE
 - key_len: 4
 - ref: NULL
 - rows: 444058
 - Extra: Using index
 - 1 row in set, 2 warnings (0.00 sec)
 - Warning (Code 1364): Field 'salary' doesn't have a default value
 - Warning (Code 1364): Field 'to_date' doesn't have a default value

Улучшения в оптимизаторе/выполнении запросов (удобство в использовании)

- Optimizer Tracing
 - INFORMATION SCHEMA.OPTIMIZER_TRACE
 - Переменные:
 - optimizer_trace="enabled=on|off,one_line=off|on"
 - optimizer_trace_features="greedy_search=on,range_optimizer=on,dynamic_range=on,repeated_subselect=on"
 - optimizer_trace_limit=1
 - optimizer_trace_max_mem_size=16384
 - optimizer_trace_offset=-1
 - end_markers_in_json=0|1

Улучшения в оптимизаторе/выполнении запросов (удобство в использовании)

- Optimizer Tracing
 - # Turn tracing on (it's off by default):
 - SET optimizer_trace="enabled=on";
 - SELECT ...; # your query here
 - SELECT * FROM
INFORMATION_SCHEMA.OPTIMIZER_TRACE;
 - # possibly more queries...
 - # When done with tracing, disable it:
 - SET optimizer_trace="enabled=off";
 - http://forge.mysql.com/wiki/MySQL_Internals_Optimizer_tracing#Typical_Usage

Улучшения в оптимизаторе/выполнении запросов (удобство в использовании)

- Optimizer Tracing

```
– mysql> SELECT * FROM INFORMATION_SCHEMA.OPTIMIZER_TRACE\G
– ***** 1. row *****
–           QUERY: select emp_no, min(from_date) from titles group by emp_no limit 10
–           TRACE: {
–     "steps": [
–       {
–         "join_preparation": {
–           "select#": 1,
–           "steps": [
–             {
–               "expanded_query": "/* select#1 */ select `titles`.`emp_no` AS
– `emp_no`,min(`titles`.`from_date`) AS `min(from_date)` from `titles` group by
– `titles`.`emp_no` limit 10"

```

Улучшения в оптимизаторе/выполнении запросов (удобство в использовании)

- Optimizer Tracing
 - join_preparation
 - join_optimization
 - table_dependencies
 - rows_estimation
 - considered_execution_plans
 - attaching_conditions_to_tables
 - clause_processing
 - refine_plan
 - reconsidering_access_paths_for_index_ordering
 - join_execution

Улучшения в оптимизаторе/выполнении запросов (удобство в использовании)

- EXPLAIN в JSON формате
 - `mysql> explain format=json insert into salaries(emp_no, from_date) select emp_no, min(from_date) from titles group by emp_no\G`
 - `***** 1. row *****`
 - EXPLAIN: {
 - "query_block": {
 - "select_id": 1,
 - "grouping_operation": {
 - "using_filesort": false,
 - "table": {
 - "table_name": "titles",

Улучшения в оптимизаторе/выполнении запросов (удобство в использовании)

- EXPLAIN в JSON формате
 - "access_type": "index",
 - "key": "emp_no",
 - "key_length": "4",
 - "rows": 444058,
 - "filtered": 100,
 - "using_index": true
 - }
 - }
 - }
 - }

Улучшения в оптимизаторе/выполнении запросов (удобство в использовании)

- EXPLAIN в JSON формате
 - 1 row in set, 2 warnings (0.01 sec)
 -
 - Warning (Code 1364): Field 'salary' doesn't have a default value
 - Warning (Code 1364): Field 'to_date' doesn't have a default value
- Определение стратегии обработки ORDER BY на этапе оптимизации

Улучшения в оптимизаторе/выполнении запросов (скорость)

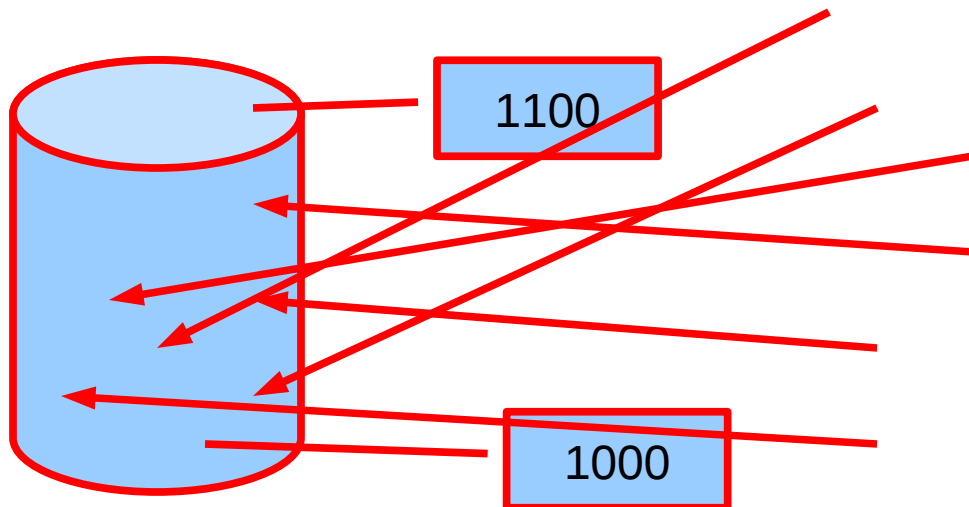
- Index Condition Pushdown
 - CREATE TABLE person (
 - personid INTEGER PRIMARY KEY,
 - firstname CHAR(20), lastname CHAR(20),
 - postalcode INTEGER, age INTEGER,
 - KEY k1 (postalcode,age)) ENGINE=InnoDB;
 - SELECT lastname, firstname FROM person
 - WHERE postalcode BETWEEN 5000 AND 5500 AND age BETWEEN 21 AND 22;

Улучшения в оптимизаторе/выполнении запросов (скорость)

- Disk Sweep Multi-Range Read (MRR)
 - CREATE TABLE items
 - (id INTEGER PRIMARY KEY, attr1 CHAR(20),
 - keypart1 INTEGER,
 - keypart2 INTEGER,
 - KEY k1 (keypart1, keypart2)) ENGINE=InnoDB;
 - SELECT * FROM items
 - WHERE keypart1 BETWEEN 1000 AND 1100 AND
keypart2 = 1200;

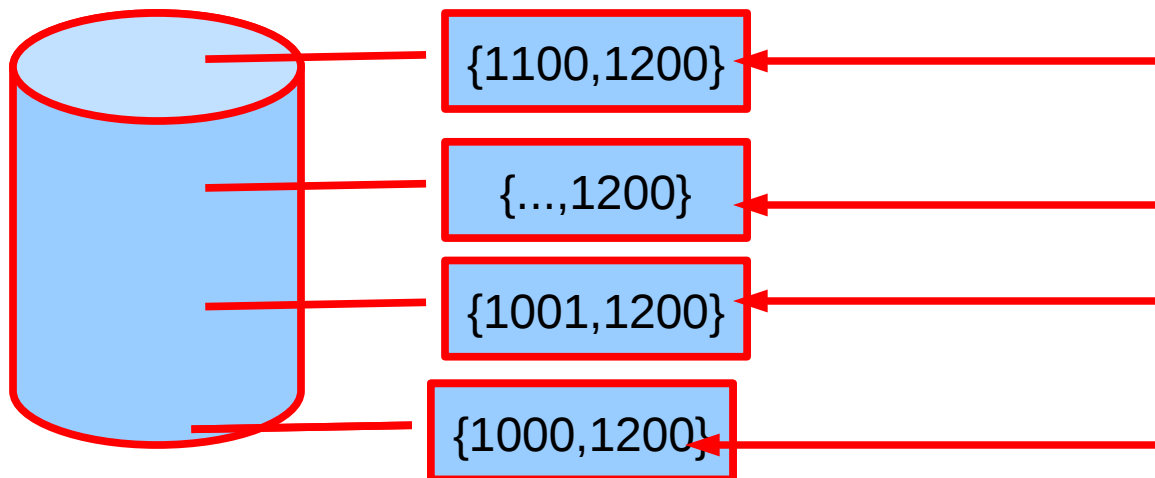
Улучшения в оптимизаторе/выполнении запросов (скорость)

- Disk Sweep Multi-Range Read (MRR)
 - Без MRR получим один промежуток вида $1000 \leq \text{keypart1} \leq 1100$.



Улучшения в оптимизаторе/выполнении запросов (скорость)

- Disk Sweep Multi-Range Read (MRR)
 - С MRR получим 100 промежутков для {keypart1, keypart2} – {1000, 1200}, {1001, 1200} ... {1100, 1200}.



Улучшения в оптимизаторе/выполнении запросов (скорость)

- Batched Key Access (ВКА) для соединений (JOINS)
 - SET
`optimizer_switch='mrr=on,mrr_cost_based=off,batched_key_access=on';`
 - Для соединений с доступом к одной из табличек по ключу
 - Накопить строки из первой таблички в буфере
 - Запросить строки из второй таблички, с ключем, одним запросом, используя DS MRR
 - Получить строки по одной от storage engine для соединения (с ссылкой на элемент в буфере)

Улучшения в оптимизаторе/выполнении запросов (скорость)

- Оптимизации выполнения подзапросов
 - Через материализацию
 - Через semi-join
 - `SELECT * FROM nation WHERE n_regionkey IN (SELECT r_regionkey FROM region WHERE r_name='AFRICA');`
 - Отложенная материализация для подзапросов из FROM/views
 - `EXPLAIN SELECT * FROM (SELECT * FROM t1) t1 WHERE 1=2;`

Улучшения в оптимизаторе/выполнении запросов (скорость)

- Оптимизации выполнения подзапросов
 - Игнорирование клауз подзапроса типа IN/ALL/SOME/EXISTS, не влияющих на результат
 - DISTINCT
 - GROUP BY, при отсутствии HAVING и агрегирующих функций
 - ORDER BY, так как LIMIT всё равно не поддерживается для этих типов подзапросов

Улучшения в оптимизаторе/выполнении запросов (скорость)

- Улучшение выполнения запросов с ORDER BY и LIMIT (filesort)
 - SELECT ... FROM single_table ... ORDER BY non_index_column [DESC] LIMIT [M,]N;
 - sort_buffer_size
- Оптимизированный анализ селективности для IN
 - eq_range_index_dive_limit=10 (0..4294967295)
 - col_name IN(val1, ..., valN)
 - col_name = val1 OR ... OR col_name = valN

Новые возможности в InnoDB

- Поддержка полнотекстового поиска
 - MATCH (col1,col2,...) AGAINST (expr [search_modifier])
 - set global innodb_ft_aux_table = 'db_name/tbl_name';
 - Переменные innodb_ft_*
 - innodb_ft_user_stopword_table
 - innodb_ft_server_stopword_table
 - Таблицы INFORMATION_SCHEMA.INNODB_FT_*
 - INNODB_FT_DEFAULT_STOPWORD
 - INNODB_FT_CONFIG
 - INNODB_FT_INDEX_CACHE

Новые возможности в InnoDB

- Новые таблицы в INFORMATION_SCHEMA
 - INNODB_METRICS
 - INNODB_SYS_*
 - INNODB_BUFFER_*
- Появилась возможность журналирования всех deadlock-ов
 - innodb_print_all_deadlocks
- Хранимая статистика для таблиц (сохраняется при рестарте, перестраивается ANALYZE TABLE, опции управления качеством)
 - innodb_analyze_is_persistent

НОВЫЕ ВОЗМОЖНОСТИ В InnoDB

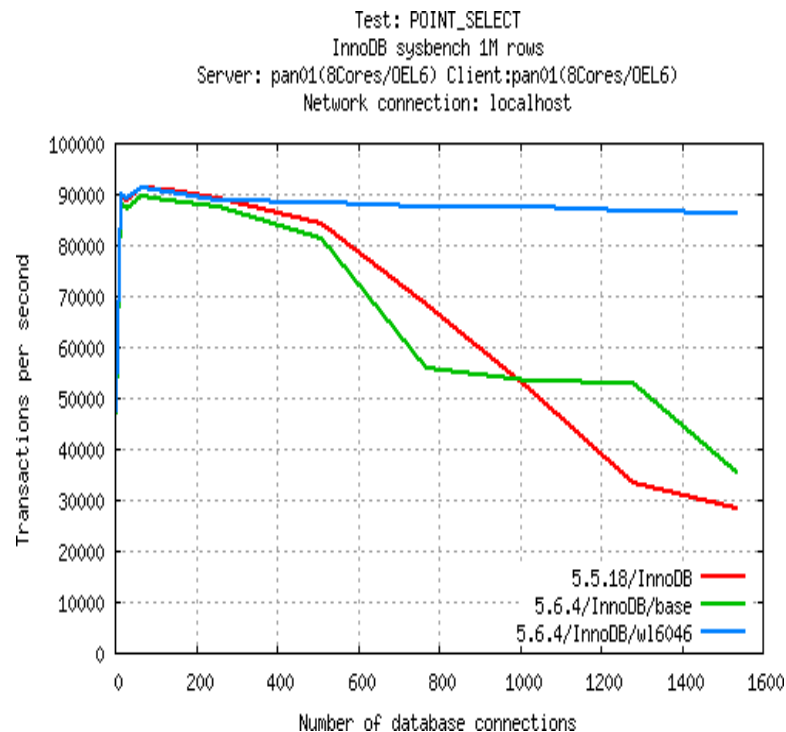
- Online ALTER
 - add/drop secondary index
 - drop foreign key
 - изменение auto_increment
 - Переименование колонок
 - Доступно только на labs.mysql.com
- Transportable tablespaces
 - Доступно только на labs.mysql.com

Новые возможности в InnoDB

- Поддержка опции DATA DIRECTORY
 - CREATE TABLE . . . DATA DIRECTORY = 'absolute path of data directory';
 - CREATE TABLE . . . PARTITION . . . DATA DIRECTORY = 'absolute path of data directory';
 - Доступно только на labs.mysql.com

Производительность и масштабируемость в InnoDB

- Оптимизация обработки read-only транзакций
 - START TRANSACTION READ ONLY
 - SET TRANSACTION READ ONLY
 - non-locking autocommit SELECTs
 - <http://blogs.innodb.com/wp/2011/12/better-scaling-of-read-only-workloads/>



Производительность и масштабируемость в InnoDB

- Поддержка страниц уменьшенного размера
 - innodb_page_size: 4K и 8K, 16K по умолчанию
 - **SSD**
- Сохранение/восстановление содержимого buffer pool
 - innodb_buffer_pool_dump_at_shutdown
 - innodb_buffer_pool_load_at_startup
 - innodb_buffer_pool_dump_now
 - innodb_buffer_pool_load_now

Производительность и масштабируемость в InnoDB

- Поддержка аппаратного вычисления контрольных сумм для страниц
 - `innodb_checksum_algorithm=crc32`
 - `innodb_checksum_algorithm=strict_crc32`
- Поддержка UNDO журнала в отдельных tablespaces
 - `innodb_undo_directory`
 - SSD
- Увеличен максимальный размер REDO журнала
 - `innodb_log_file_size`
 - 512Gb

Производительность и масштабируемость в InnoDB

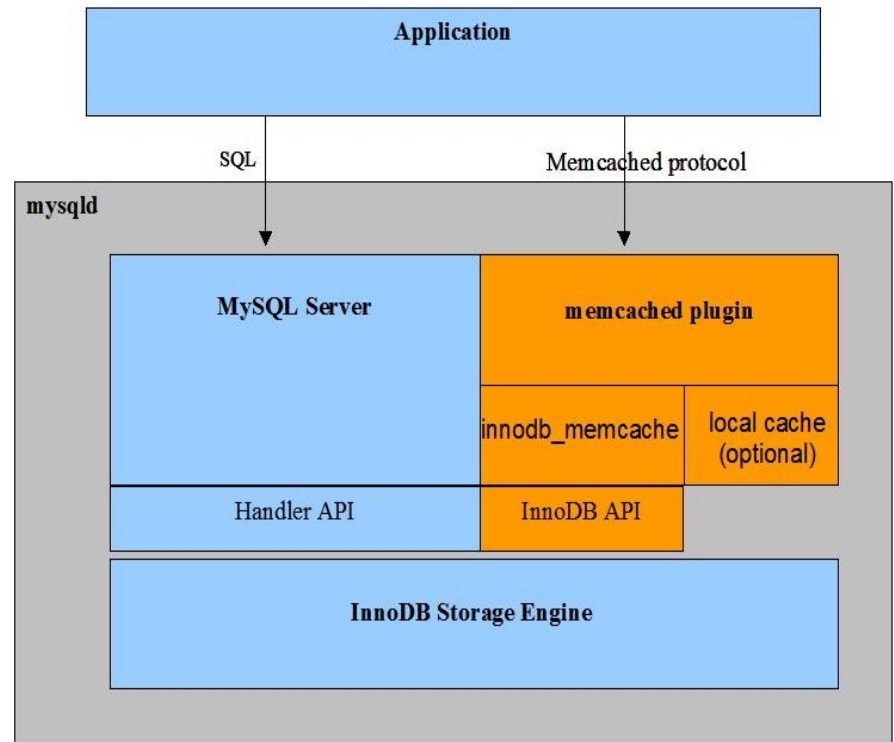
- Kernel mutex разбит на несколько mutex-ов
 - Транзакции и MVCC views: `trx_sys_t::lock (rw_lock)` и `trx_t::mutex`
 - Locking : `lock_sys_t::mutex` и `lock_sys_t::wait_mutex`
- Оптимизация использования блокировок при обращении к `buffer pool`
- Оптимизация использования блокировок во время `commit`

Производительность и масштабируемость в InnoDB

- Отдельная нить для выталкивания страниц из buffer pool (вместо master thread), новый алгоритм выталкивания
 - 5.6.5
 - <http://dimitrik.free.fr/blog/archives/2012/04/mysql-performance-improved-adaptive-flushing-in-56labs.html>
- Несколько purge ниток
 - `innodb_purge_threads <= 32`
- Очистка кэша InnoDB таблиц
 - `table_definition_cache`
 - LRU

Memcached API для InnoDB

- Daemon plugin для MySQL
- Прямой доступ к InnoDB
- Поддерживает бинарный и текстовый протокол
- Поддерживает значения из нескольких полей
- Тонкая настройка кэширования:
 - innodb_only
 - cache-only
 - caching



Memcached API для InnoDB

- Поддерживает batch операции для чтения и записи
- Поддерживает простую аутентификацию через SASL
- Поддерживает MySQL binary log
 - innodb_api_enable_binlog
- Доступно на <http://labs.mysql.com>

Улучшения в репликации: надёжность

- UUID для идентификации серверов
- Поддержка Global Transaction Identifiers
 - GTID = source_id:transaction_id
 - CHANGE MASTER TO MASTER_AUTO_POSITION=1
 - gtid_mode
- Crash-safe master и slave
 - автоматическое восстановление после падения
 - файлы состояния
 - хранение состояния в системных таблицах
 - Master: crash safe binary log

Улучшения в репликации: надёжность

- Поддержка контрольных сумм в binary log
 - `binlog-checksum={NONE|CRC32}`
- Информационные события для RBR
 - `binlog-rows-query-log-events`
- Прямая работа с binary log для удаленного сервера
 - `mysqlbinlog --read-from-remote-server --host=host_name --raw --stop-never binlog.000130`

Улучшения в репликации: производительность

- Многопоточный slave
 - `slave_parallel_workers=[0..1024]`
- Репликация с задержкой
 - `CHANGE MASTER TO MASTER_DELAY = N;`
- Поддержка частичных образов в RBR
 - `binlog_row_image={full|minimal|noblob}`

Расширения системы типов/операций

- Поддержка микросекунд для TIME, TIMESTAMP и DATETIME
- Поддержка DEFAULT NOW()/ON UPDATE NOW() для DATETIME и нескольких полей
- Поддержка точных OpenGIS операций для GIS данных
- Поддержка Unicode 5.2 Collation Algorithm и нескольких новых collations: Вьетнамская, Хорватская, Бенгальская и др.
- Поддержка UTF-16LE и Unicode в Windows CLI
- TO_BASE64()/FROM_BASE64() и WEIGHT_STRING()

Улучшения в обработке ошибок

- GET DIAGNOSTICS
 - GET DIAGNOSTICS rows = ROW_COUNT, conditions = NUMBER;
 - GET DIAGNOSTICS CONDITION 1 code = RETURNED_SQLSTATE, msg = MESSAGE_TEXT;
- Поведение обработчиков ошибок (HANDLER) в хранимых процедурах стало ближе к стандарту

Новые возможности partitioning

- Возможность явного задания partition в операторе
 - SELECT * FROM employees PARTITION (p0, p2);
 - DELETE FROM employees PARTITION (p0, p1);
 - UPDATE employees PARTITION (p0) SET store_id = 2 WHERE fname = 'Jill';
 - SELECT e.id, s.city FROM employees AS e JOIN stores PARTITION (p1) AS s ...;
- Импорт/Экспорт в/из таблиц с partitioning
 - ALTER TABLE partitioned EXCHANGE PARTITION p0 WITH TABLE not_partitioned;

Новое в PERFORMANCE_SCHEMA

- Инструментированы блокировки таблиц, ввод-вывод в таблицы и сетевой ввод-вывод
- Агрегирование статистики для блокировок и ввода-вывода
 - table_io_waits_summary_by_index_usage
 - table_io_waits_summary_by_table
 - table_lock_waits_summary_by_table
 - socket_summary_by_event_name
 - socket_summary_by_instance

Новое в PERFORMANCE_SCHEMA

- Инструментированы операторы
 - events_statements_*

```
SELECT THREAD_ID, EVENT_NAME, TIMER_WAIT, SQL_TEXT, ROWS_SENT, ROWS_EXAMINED,
CREATED_TMP_TABLES, NO_INDEX_USED FROM performance_schema.events_statements_history
LIMIT 3, 3;
```

THRE AD_ID	EVENT_NAME	TIMER_WAIT	SQL_TEXT	ROWS _SENT	ROWS_EX AMINED	CREATED T MP_TABLES	NO INDE X_USED
16	statement/sql/create_ table	1572757000	create table t1 (i int)	0	0	0	0
16	statement/sql/insert	1129054000	insert into t1 values (1), (2), (3)	0	0	0	0
16	statement/sql/select	678927000	select * from t1 where i = 2	1	3	0	1

Новое в PERFORMANCE_SCHEMA

- Инструментированы этапы выполнения операторов
– `events_stages_*`

EVENT_NAME	SOURCE	TIMER_START	TIMER_WAIT
stage/sql/init	sql_parse.cc:936	552715025748	66623746
stage/sql/Opening tables	sql_base.cc:4805	552788106302	29843681
stage/sql/preparing	sql_select.cc:1977	552876811982	21863187
stage/sql/init	sql_select.cc:3588	552831727471	20826223
stage/sql/System lock	lock.cc:324	552817949983	13777488

Новое в PERFORMANCE_SCHEMA

- Поддержка digests для операторов и агрегирование по НИМ
 - 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_thread_by_event_name
 - events_statements_summary_by_user_by_event_name
 - events_statements_summary_global_by_event_name

Новое в PERFORMANCE_SCHEMA

- Фильтрация по пользователям, сессиям и таблицам
 - `events_waits_summary_by_account_by_event_name`
 - `events_waits_summary_by_host_by_event_name`
 - `events_waits_summary_by_user_by_event_name`
- Таблица `HOST_CACHE`

Итоги

- Удобство использования
- Надёжность
- Масштабируемость
- Производительность

Ссылки

- Подробнее о MySQL-5.6:
 - <http://dev.mysql.com/doc/refman/5.6/en/mysql-nutshell.html>
 - <http://dev.mysql.com/doc/refman/5.6/en/news-5-6-x.html>
- Загрузить MySQL-5.6.5-m8:
 - <http://dev.mysql.com/downloads/mysql/#downloads>
- MySQL Labs релизы:
 - <http://labs.mysql.com/>

Спасибо!

?