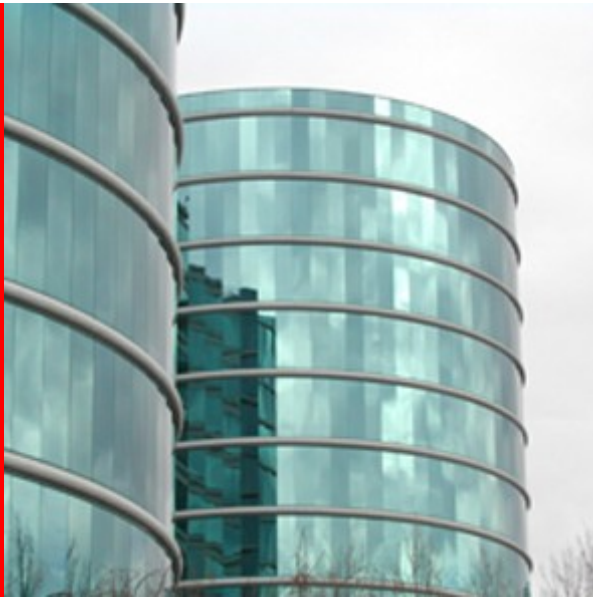


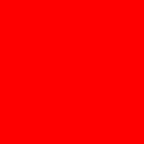
**ORACLE®**



**ORACLE®**

## **Новые возможности в MySQL 5.6**

Дмитрий Ленеv



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.

# Актуальные версии MySQL

- 5.1 – “старая” стабильная ветка
  - готова к использованию с ноября 2008
  - последняя версия 5.1.57, выпущена в мае 2011
- 5.5 – “новая” стабильная ветка
  - готова к использованию с декабря 2010
  - последняя версия 5.5.13, выпущен в мае 2011
- 5.6 – текущая development ветка
  - 5.6.2 – milestone release, выпущен в апреле 2011

# Основные направления улучшений в MySQL 5.6

- Оптимизатор/выполнение запросов
- InnoDB
- Репликация
- performance\_schema
- Partitioning
  
- Memcached API для InnoDB

# Улучшения в оптимизаторе/выполнении запросов.

- **Index Condition Pushdown** (storage engine при чтении по индексу проверяет часть условия которая зависит только от полей в индексе, не читая всю строку и не возвращая управление SQL-layer).

```
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.

- Disk Sweep – Multi-Range Read

- чтение строк по индексу, в порядке расположения их на диске
- пакетная обработка групп чтений по индексу

```
CREATE TABLE items (  
  id INTEGER PRIMARY KEY,  
  attr1 CHAR(20),  
  keypart1 INTEGER,  
  keypart2 INTEGER,  
  KEY k1 (keypart1, keypart2)  
) ENGINE=InnoDB;
```

Без MRR получим один промежуток вида  $1000 \leq \text{keypart1} \leq 1100$ .

С MRR получим 100 промежутков для  $\{\text{keypart1}, \text{keypart2}\} - \{1000, 1200\}, \{1001, 1200\} \dots \{1100, 1200\}$ .

```
SELECT * FROM items
```

```
WHERE keypart1 BETWEEN 1000 AND 1100 AND keypart2 = 1200;
```

# Улучшения в оптимизаторе/выполнении запросов.

- Улучшение filesort
  - для запросов с ORDER BY non\_indexed\_column и LIMIT x
  - и случаев когда x строк помещаются в sort buffer

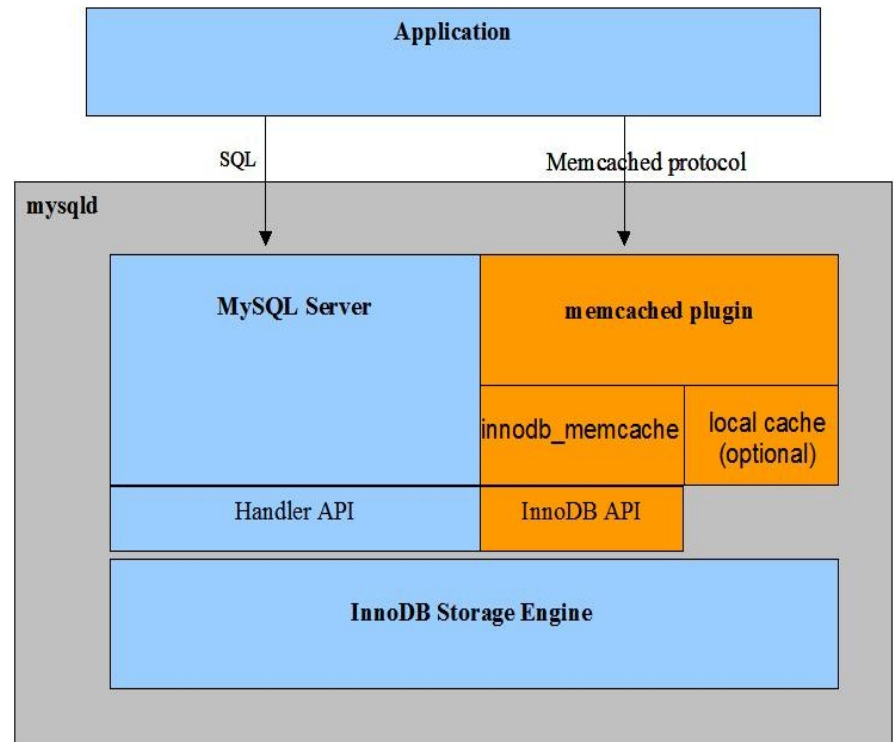


# Улучшения в InnoDB

- Хранимая статистика для таблиц (сохраняется при рестарте, перестраивается ANALYZE TABLE, опции управления качеством).
- Новые таблицы в INFORMATION\_SCHEMA (INNODB\_METRICS, INNODB\_SYS\_\*, INNODB\_BUFFER\_\*)
- Поддержка ICP и MRR
- Kernel mutex разбит на несколько mutexов
- Несколько purge ниток (--innodb\_purge\_threads, <=32)
- Отдельная нить для выталкивания страниц из buffer pool (вместо master thread)
- 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
- Опционально поддерживает локальное кэширование
- Поддерживает batch операции для чтения и записи
- Доступен как MySQL Labs релиз



## Улучшения в репликации

- Поддержка частичных образов в RBR
- Многопоточный slave (параллелизм для БД, MySQL Labs релиз)
- Crash-safe slave (автоматическое восстановление репликации и master/relay.info после падения)
- Поддержка контрольных сумм в binary log
- Репликация с задержкой
- Информационные события для RBR (запросы)
- UUID для идентификации серверов
- Прямая работа с binary log для удаленного сервера

# Новое в performance\_schema

- Инструментирован ввод-вывод в таблицы (wait/io/table/sql/handler)

```
SELECT THREAD_ID, EVENT_NAME, TIMER_WAIT, OBJECT_SCHEMA, OBJECT_NAME, INDEX_NAME, OPERATION FROM performance_schema.events_waits_history_long LIMIT 10,1;
```

THREAD_ID	EVENT_NAME	TIMER_WAIT	OBJECT_SCHEMA	OBJECT_NAME	INDEX_NAME	OPERATION
16	wait/io/table/sql/handler	12542508	test	marker	NULL	insert

- Агрегирование данных о вводе-выводе в таблицы и индексы (table\_io\_waits\_summary\_by\_table/index\_usage)

```
SELECT OBJECT_SCHEMA, OBJECT_NAME, COUNT_INSERT, SUM_TIMER_INSERT, AVG_TIMER_INSERT FROM performance_schema.table_io_waits_summary_by_table LIMIT 5, 1;
```

OBJECT_SCHEMA	OBJECT_NAME	COUNT_INSERT	SUM_TIMER_INSERT	AVG_TIMER_INSERT
test	marker	14	279283356	19948500

## Новое в performance\_schema

- Инструментированы блокировки таблиц (wait/lock/table/sql/handler)
- Агрегирование статистики для блокировок (table\_lock\_waits\_summary\_by\_table)
- Фильтрация по пользователям/сессиям (setup\_actors, threads)
- Глобальное агрегирование по типу объектов (objects\_summary\_global\_by\_type)

# Новое в performance\_schema (MySQL Labs)

- Инструментированы операторы (events\_statements\_\*, events\_statements\_summary\_\*)

```
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

```
SELECT THREAD_ID, EVENT_NAME, SUM_TIMER_WAIT, AVG_TIMER_WAIT,  
       SUM_ROWS_SENT, SUM_NO_INDEX_USED FROM  
performance_schema.events_statements_summary_by_thread_by_event_name WHERE  
THREAD_ID=16 LIMIT 1;
```

THREAD_ID	EVENT_NAME	SUM_TIMER_WAIT	AVG_TIMER_WAIT	SUM_ROWS_SENT	SUM_NO_INDEX_U SED
16	statement/sql/select	1062565000	265641000	1	1

## Новое в performance\_schema (MySQL Labs)

- Инструментированы этапы выполнения операторов (event\_stages\_\*, event\_stages\_summary\_\*)

```
SELECT EVENT_NAME, SOURCE, TIMER_START, TIMER_WAIT FROM
performance_schema.events_stages_history WHERE THREAD_ID=16 AND
NESTING_EVENT_ID=343 ORDER BY TIMER_WAIT DESC LIMIT 5;
```

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

- Доступно содержимое host cache

# 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 e EXCHANGE PARTITION p0 WITH TABLE e2;
```



- Подробнее о MySQL 5.6:
  - <http://dev.mysql.com/doc/refman/5.6/en/mysql-nutshell.html>
  - <http://dev.mysql.com/tech-resources/articles/whats-new-in-mysql-5.6.html>
- Загрузить MySQL 5.6.2-m5:
  - <http://dev.mysql.com/downloads/mysql/>
- MySQL Labs релизы:
  - <http://labs.mysql.com/>

**ORACLE®**