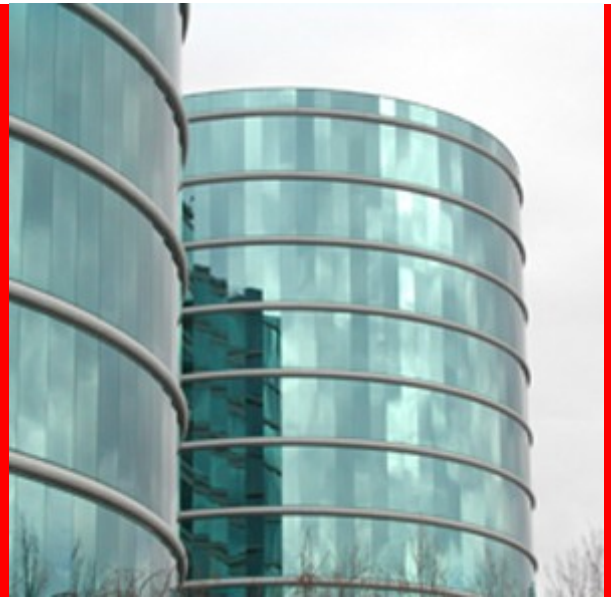


MySQL™

ORACLE®

# MySQL: проблемы, решения и последствия.

Дмитрий Ленев, Ноябрь 2012, Санкт-Петербург

# Почему?

# Краткая история MySQL
## (с точки зрения разработчика сервера)

- 80-ые –

  Monty Widenius разрабатывает Unireg, ISAM СУБД с не-SQL интерфейсом в виде экранных форм
- 1995/96 –

  начало работы над MySQL, релизы для внутреннего использования, первый публичный релиз, нет и не планируется транзакций, GRANT операторов.
- 1998/99 –

  Несколько стабильных релизов ветки 3.*, реализация MyISAM SE
- 2000 –

  Переход на GPL, начало работы над транзакционными SE

# Краткая история MySQL
## (с точки зрения разработчика сервера)

- 2001 –

  3.23.31 – Heap, BDB SE (транзакции, page-locks), Full-text операции, репликация

- 2002 –

  Первая стабильная версия InnoDB SE (Innobase Oy, транзакции, row-locks, MVCC).

- 2003 –

  4.0.12, Query Cache, Full-text индексы, embedded библиотека,  TRUNCATE, UNION, …

  MySQL покупает NDB Cluster (Alzato, дочка Ericcson, in-memory, shared nothing)

- 2004 –

  4.1.7, подзапросы, Unicode, GIS, OpenSSL

# Краткая история MySQL
## (с точки зрения разработчика сервера)

- 2005 –
  - Oracle покупает Innobase Oy, разработчика главного транзакционного SE – InnoDB
  - Концепция Pluggable SE, работа над Maria (Monty), Falcon (Jim Starkey)
  - 5.0.15, хранимые процедуры, представления, триггеры, I_S, XA, Federated SE, Archive SE (критика за качество)
- 2007 –
  - Выделение MySQL Cluster в отдельный продукт
  - 5.1.22-rc – partitioning, RBR, events, log tables, cluster replication, cluster disk storage, XML support
  - Начало работы над веткой 6.0
- 2008 –
  - MySQL куплен Sun Microsystems
  - 5.1.30 – GA (спустя год после rc!)
  - Осознание что 6.0 невозможен, переход к новой модели разработки

# Краткая история MySQL
## (с точки зрения разработчика сервера)

- 2009 –

  Oracle покупает Sun Microsystems (мы теперь одна компания с InnoDB! Maria и Falcon заморожены).

- 2010 –

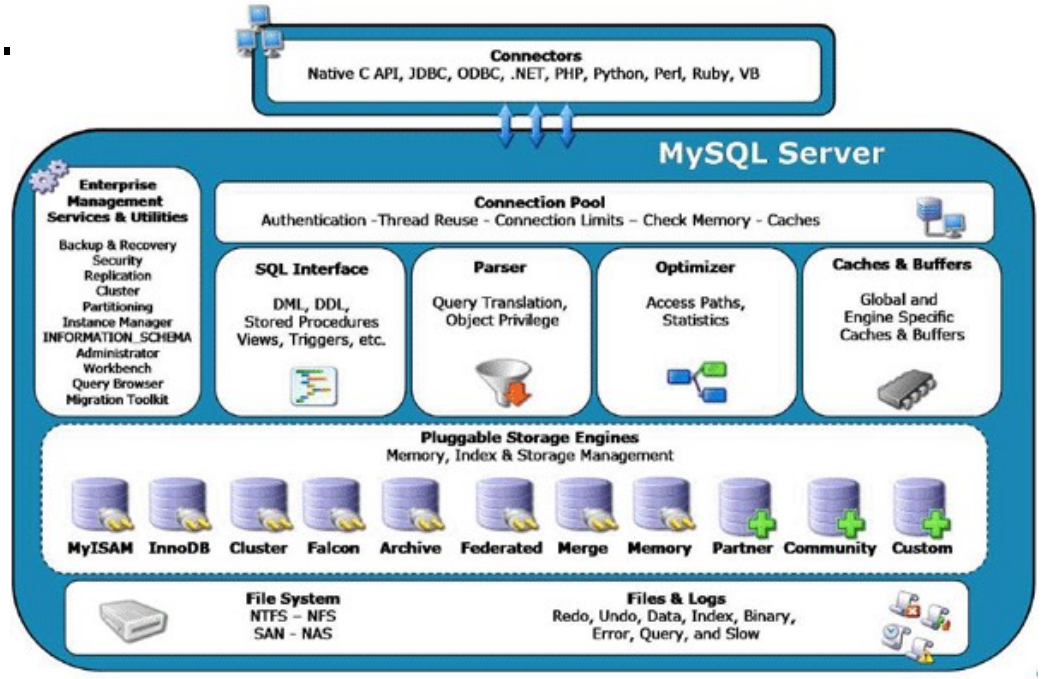
  5.5.8, масштабируемость и производительность, диагностика и мониторинг, решение проблем 5.0, полусинхронная репликация, коммерческие расширения...

- 2012 –

  5.6.7-rc, улучшения в оптимизаторе, EXPLAIN, в InnoDB (масштабируемость и производительность, FTS, TT), в безопасности (plugins), репликации (MTS, GTID)

# Проблема #1: Отсутствие транзакций

- MyISAM – не транзакционный, табличные блокировки
- Решение: Storage Engine API и сторонние Storage Engines.



# Проблема #1: Отсутствие транзакций

- Дополнительные плюсы:
  - Гибкость (InnoDB, CSV, Archive, Federated, ...)
  - Экосистема (Tokutek, NitroDB, Sphinx, Custom SE)
  - Дивиденды от маркетинга!

# Проблема #1: Отсутствие транзакций

- Проблемы:
  - Транзакционность сильное свойство которое нужно учитывать при проектировании (ACID!, проблемы со стандартом, архитектурой, репликацией, с DD и DDL, deadlock detector, тянутся до сих пор, LOCK TABLES, prelocking)
  - Зачет по последнему (GTID и MyISAM, FK).
  - API как барьер (3 LockMan, интерфейс как зеркало организации, дублирование данных и кода)

# Проблема #1: Отсутствие транзакций

- Уроки:
  - Сильные свойства лучше учитывать с самого начала
  - Альтернатива производить глобальное перепроектирование при их добавлении
  - Отказ от гибкости иногда имеет смысл (FK и DD в PostgreSQL).
- Текущий тренд:
  - Ориентация на InnoDB

# Проблема #2: Совместимость

- Обратная (backward):
  - SET sql_mode=...

    REAL_AS_FLOAT, PIPES_AS_CONCAT, ANSI_QUOTES, IGNORE_SPACE, ONLY_FULL_GROUP_BY, NO_UNSIGNED_SUBTRACTION, NO_DIR_IN_CREATE, POSTGRESQL, ORACLE, MSSQL, DB2, MAXDB, NO_KEY_OPTIONS, NO_TABLE_OPTIONS, NO_FIELD_OPTIONS, MYSQL323, MYSQL40, ANSI, NO_AUTO_VALUE_ON_ZERO, NO_BACKSLASH_ESCAPES, STRICT_TRANS_TABLES, STRICT_ALL_TABLES, NO_ZERO_IN_DATE, NO_ZERO_DATE, ALLOW_INVALID_DATES, ERROR_FOR_DIVISION_BY_ZERO, TRADITIONAL, NO_AUTO_CREATE_USER, HIGH_NOT_PRECEDENCE, NO_ENGINE_SUBSTITUTION, PAD_CHAR_TO_FULL_LENGTH
  - опции запуска (--new, --explicit_defaults_for_timestamp)
- Снизу вверх (forward): Комментарии вида /*!52301 … */

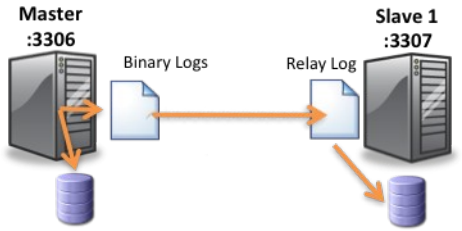
```
CREATE TABLE `t1` (
  `a1` int(11) NOT NULL,
  `a2` varchar(256) DEFAULT NULL,
  `a3` blob,
  PRIMARY KEY (`a1`)
) /*!50100 TABLESPACE ts1 STORAGE DISK */ ENGINE=ndbcluster DEFAULT CHARSET=latin1
```

# Проблема #2: Совместимость

- Проблемы:
  - Хорошо работают в простых случаях
  - Плохо работают с репликацией
  - Создают проблемы для сложных объектов
  - Увеличивают размер кода и его сложность
  - Источник ошибок
  - Проблемы при тестировании
- Стараемся избегать новых sql_mode
- Помогает хорошая и аггрессивная deprecation policy
- Чтите стандарт!

# Проблема #3: Репликация

- Изначально Statement Based
  - Просто реализовать
  - Маленький журнал
  - Поддерживает трюки (blackhole)
- Проблемы:
  - Требует репликации среды и применима только при полной детерминированности
  - Плохо работает для сложных конструкций и ситуаций (unsafe конструкции, конфликты, частичная репликация, SF, не serializable режимы, KILL)



# Проблема #3: Репликация

- Решение Row Based
  - Простая модель которая работает во всех ситуациях
  - Журнал может быть гораздо больше
- Промежуточное решение Mixed mode
- Выводы:
  - Иногда стоит предпочитать менее “умные” но более надежные решения
  - Возможно с самого начала стоило бы рассмотреть интеграцию журнала репликации и storage engine

# Проблема #4: Диагностика

SHOW операторы и status variables.

Проблемы:

- Далеко не все покрыто
- SHOW дает результат в форме текста
- Проблемы с фильтрацией

```
SHOW ENGINE INNODB STATUS\G
...
----------
SEMAPHORES
----------
OS WAIT ARRAY INFO: reservation count 413452, signal count 378357
--Thread 32782 has waited at btr0sea.c line 1477 for 0.00 seconds the
semaphore: X-lock on RW-latch at 41a28668 created in file btr0sea.c line 135
a writer (thread id 32782) has reserved it in mode wait exclusive
number of readers 1, waiters flag 1
Last time read locked in file btr0sea.c line 731
Last time write locked in file btr0sea.c line 1347
Mutex spin waits 0, rounds 0, OS waits 0
RW-shared spins 108462, OS waits 37964; RW-excl spins 681824, OS waits
375485
...
------------
TRANSACTIONS
------------
Trx id counter 0 290328385
Purge done for trx's n:o < 0 290315608 undo n:o < 0 17
History list length 20
Total number of lock structs in row lock hash table 70
LIST OF TRANSACTIONS FOR EACH SESSION:
---TRANSACTION 0 290328384, ACTIVE 0 sec, process no 3205, OS thread id
38929 inserting
1 lock struct(s), heap size 320
MySQL thread id 29, query id 4668736 localhost heikki update
insert into speedc values (1519229,1, 'hgjhjgghggjgjgjgjgjgjgjgjgjgjgjg
jlhhgghgggggghhjhghgggggghjhghghghghghhhhghghghjhhjghjghjkghjghjghjghjfhjfh
...
----------------------
BUFFER POOL AND MEMORY
----------------------
Total memory allocated 84966343; in additional pool allocated 1402624
Buffer pool size   3200
Free buffers       110
Database pages     3074
Modified db pages  2674
Pending reads 0
Pending writes: LRU 0, flush list 0, single page 0
Pages read 171380, created 51968, written 194688
28.72 reads/s, 20.72 creates/s, 47.55 writes/s
Buffer pool hit rate 999 / 1000
...
```

# Проблема #4: Диагностика

- Текущий подход P_S и I_S таблицы:
  - Систематическое покрытие и разные типы событий
  - Универсальность результатов
  - Легкость настройки и фильтрации
  - Сравнительно малые накладные расходы (около 5%)

# Проблема #4: Диагностика

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;

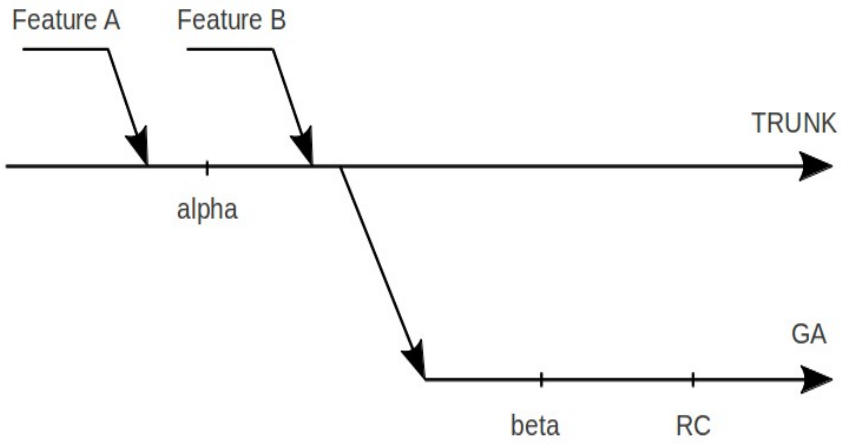
| EVENT_NAME | SOURCE | TIMER_START | TIMER_WAIT |
|---|---|---|---|
| stage/sql/init | sql_parse.cc:936 | 552715025748 | 66623746 |
| stage/sql/Opening tables | sql_base.cc:4805 | 552788106302 | 29843681 |
| stage/sql/preparing | sql_select.cc:1977 | 552876811982 | 21863187 |
| stage/sql/init | sql_select.cc:3588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

SELECT EVENT_NAME, COUNT_STAR, SUM_TIMER_WAIT, AVG_TIMER_WAIT FROM performance_schema.events_waits_summary_global_by_event_name WHERE EVENT_NAME IN ('wait/synch/mutex/sql/LOCK_open', 'wait/io/file/myisam/dfile');

| EVENT_NAME | COUNT_STAR | SUM_TIMER_WAIT | AVG_TIMER_WAIT |
|---|---|---|---|
| wait/synch/mutex/sql/LOCK_open | 3 | 581694 | 193764 |
| wait/io/file/myisam/dfile | 11 | 29418762 | 2674104 |

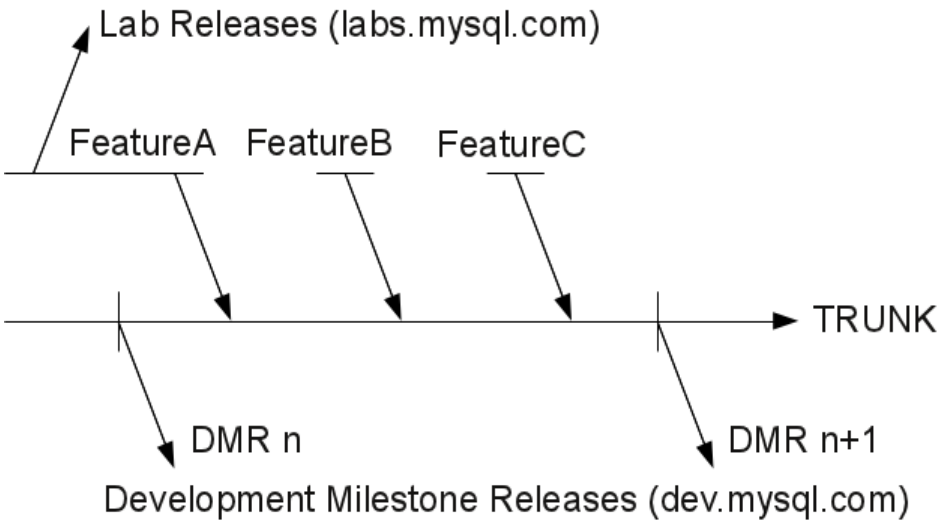
# Проблема #5: Цикл разработки

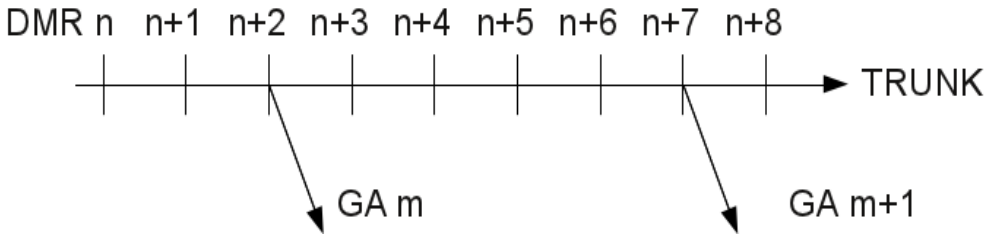
- Исторически сложившийся подход (до 5.3)
- Фокус QA на стадии alpha-betta-RC
- Опора на отчеты от сообщества и частично на внутренний QA
- Модель маршрутки
- Проблемы с качеством
- Нестабильность по срокам



# Проблема #5: Цикл разработки

- Development Milestone Releases модель (5.5+)
- Фокус на QA в Feature tree
- Регулярные (3-6 месяцев) DMR релизы с RC качеством
- Периодически DMR → RC → GA
- Дополнительное QA в транке перед DMR
- Больший упор на внутренний QA
- Модель поезда
- Выше качество, стабильнее по срокам



# Проблема #5: Цикл разработки

- Специальный случай – MySQL Cluster
- Главная аудитория – телекомы
- Требуется гораздо более частый выпуск стабильных major версии
- Результат – отдельный продукт со своим деревом и расписанием релизов
- Проблема – обмен изменениями в обе стороны


ORACLE®