

MariaDB 10.1

ЧТО НОВОГО

Сергей Петруня, MariaDB

 Devconf '2016

<http://www.devconf.ru>

Стабильные релизы MySQL/MariaDB

- MySQL 5.7: 21 октября 2015
- **MariaDB 10.1: 17 октября 2015**
- MariaDB 10.0: 31 марта 2014
- MySQL 5.6: 5 февраля 2013

Основные фишки в MariaDB 10.1

- **Встроенный Galera Cluster**
- Шифрование данных
- Улучшения в Roles
- Оптимизатор
 - EXPLAIN и ANALYZE
 - Другие улучшения
- Оптимистичная параллельная репликация

Встроенный Galera Cluster

- MariaDB 5.5, 10.0: -galera дерево
- MariaDB 10.1
 - Поддержка Galera в главном дереве
 - Собранные Galera packages.

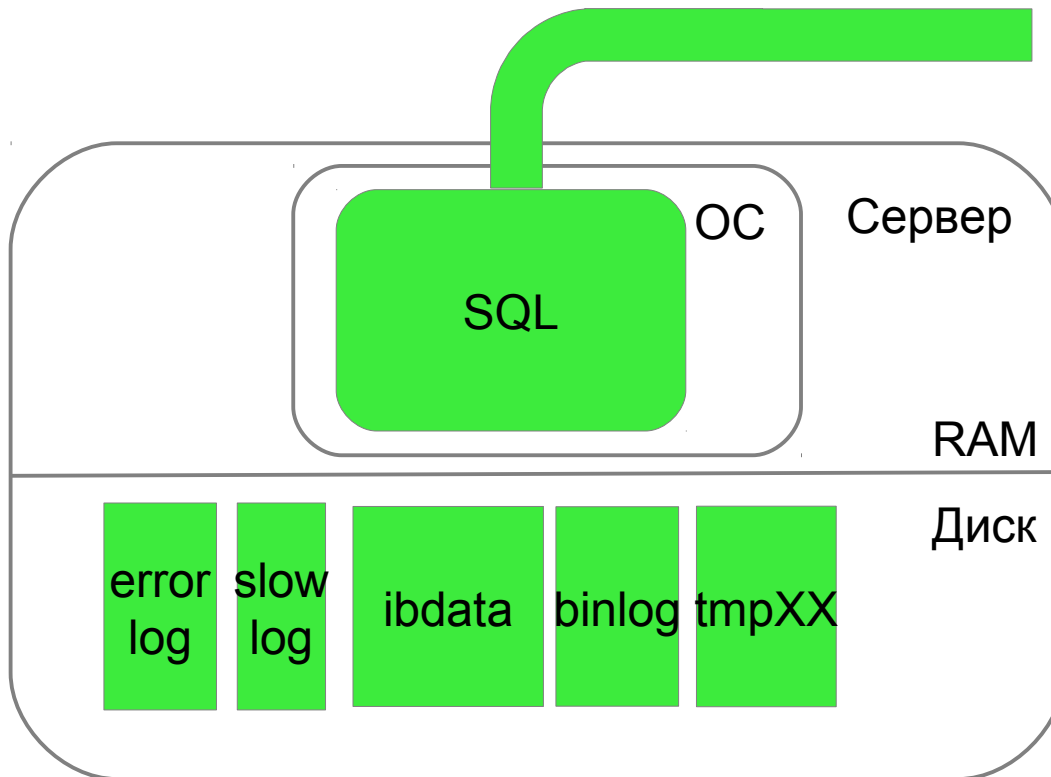
Главные фишки в MariaDB 10.1

- Встроенный Galera Cluster
- **Шифрование данных**
- Улучшения в Roles
- Оптимизатор
 - EXPLAIN и ANALYZE
 - Другие улучшения
- Оптимистичная параллельная репликация

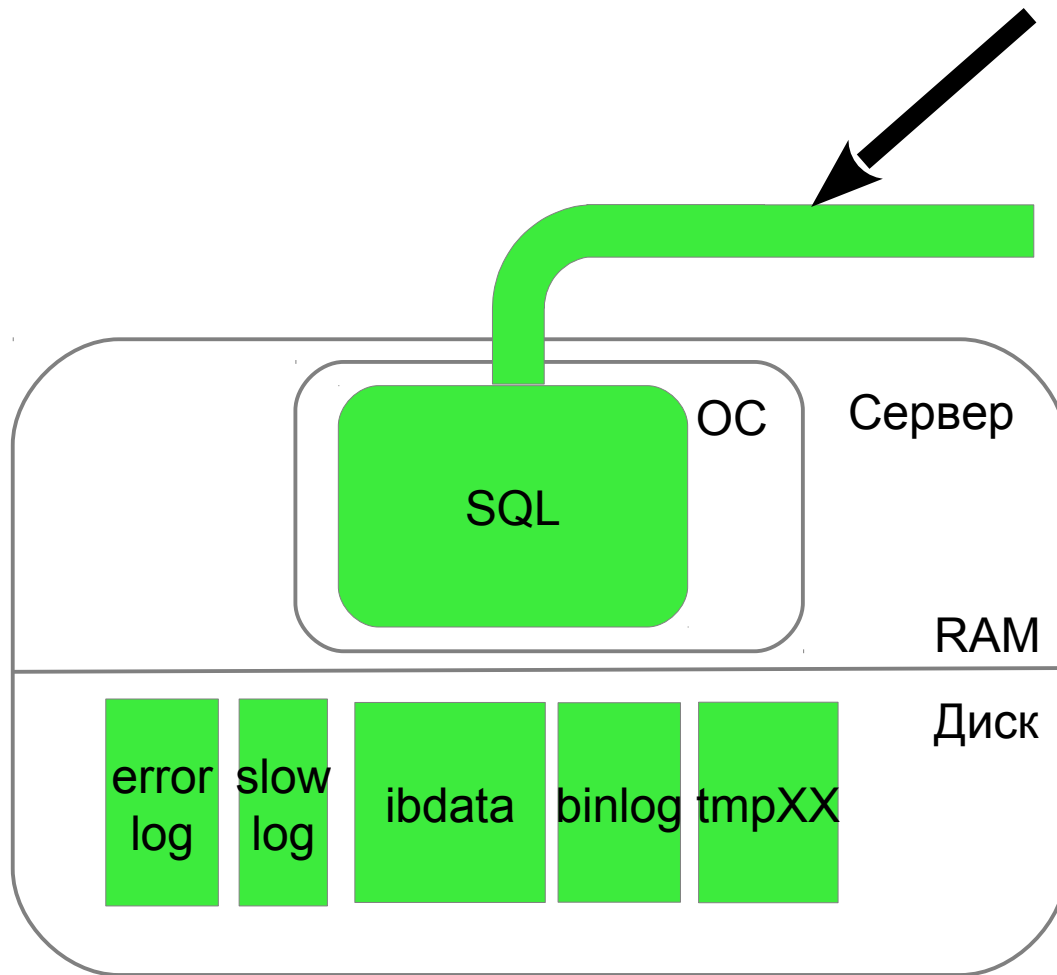
Шифрование в MariaDB

- Основано на
 - патче от Google
 - Патче от Eperi
- “Data-at-rest encryption”

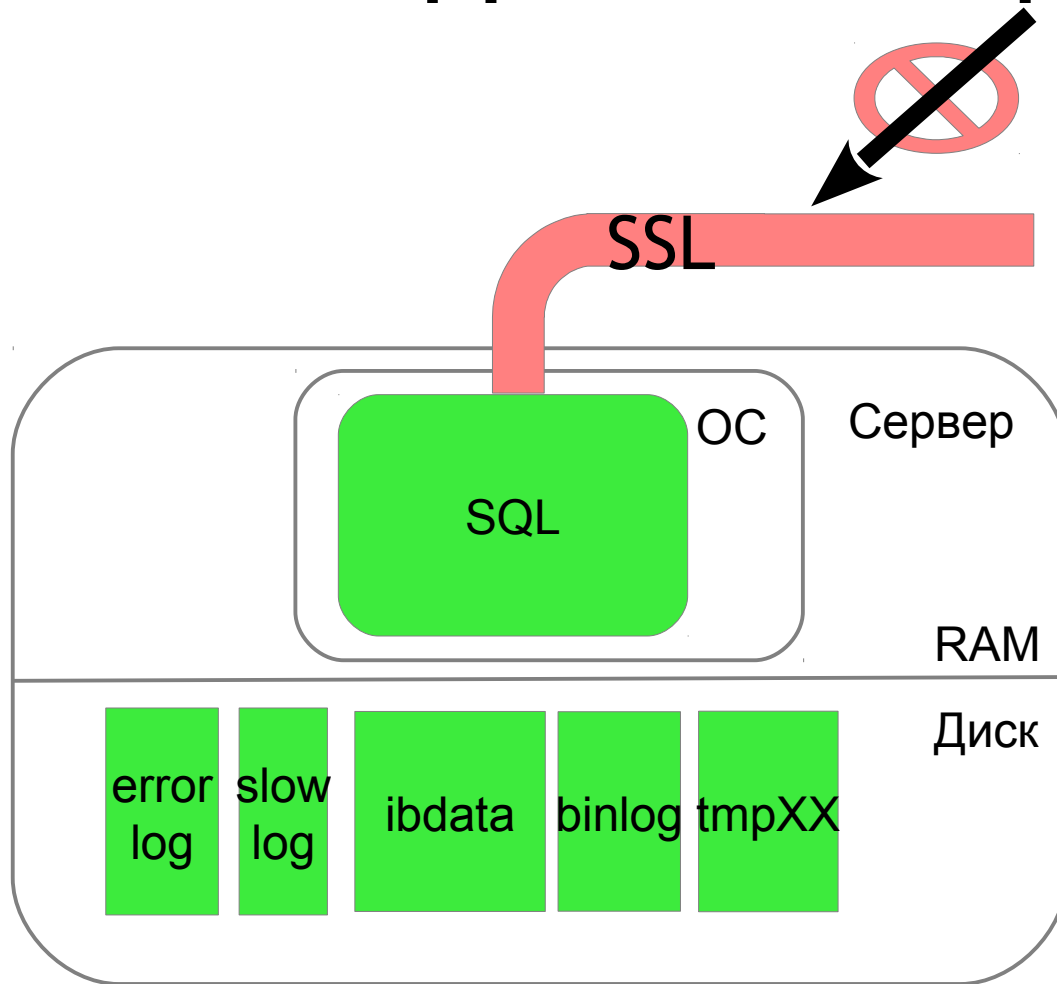
Сервер с данными



Прослушивание сети

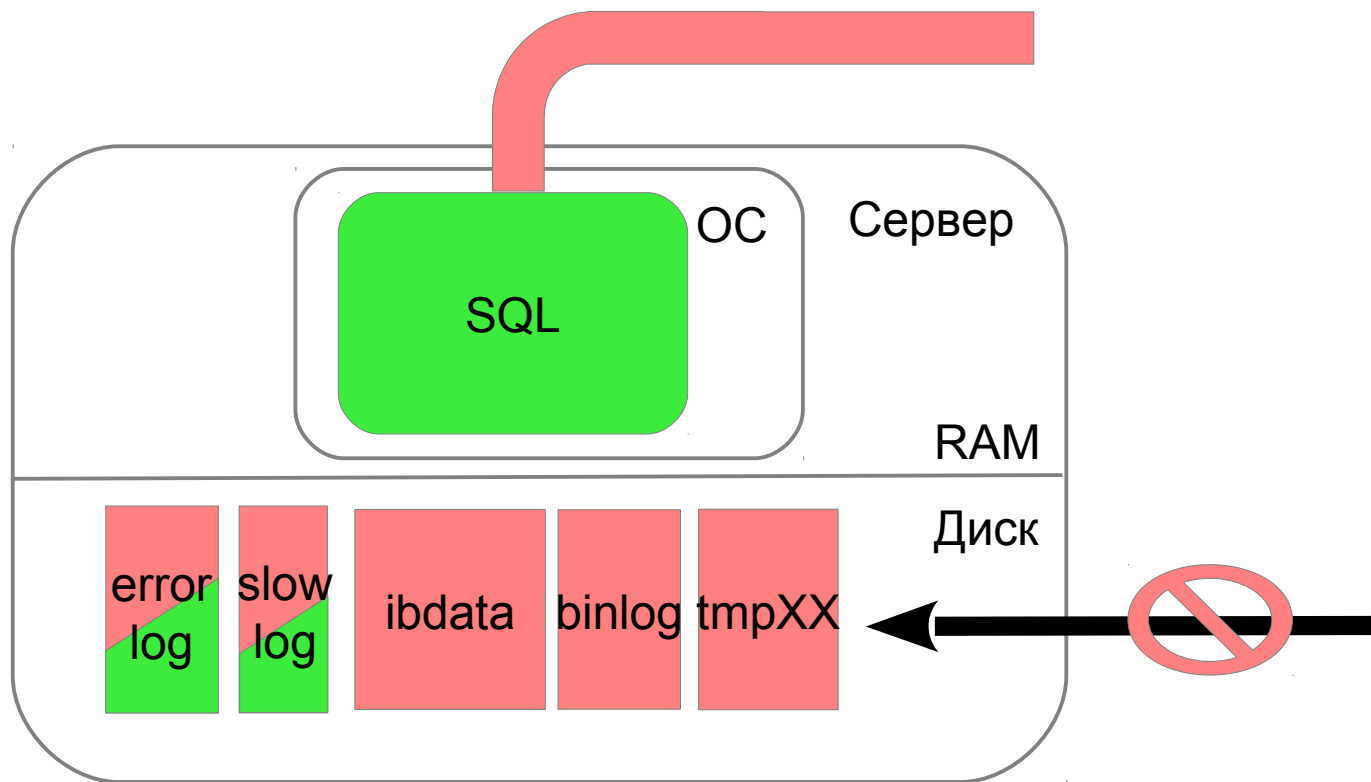


SSL-шифрование трафика



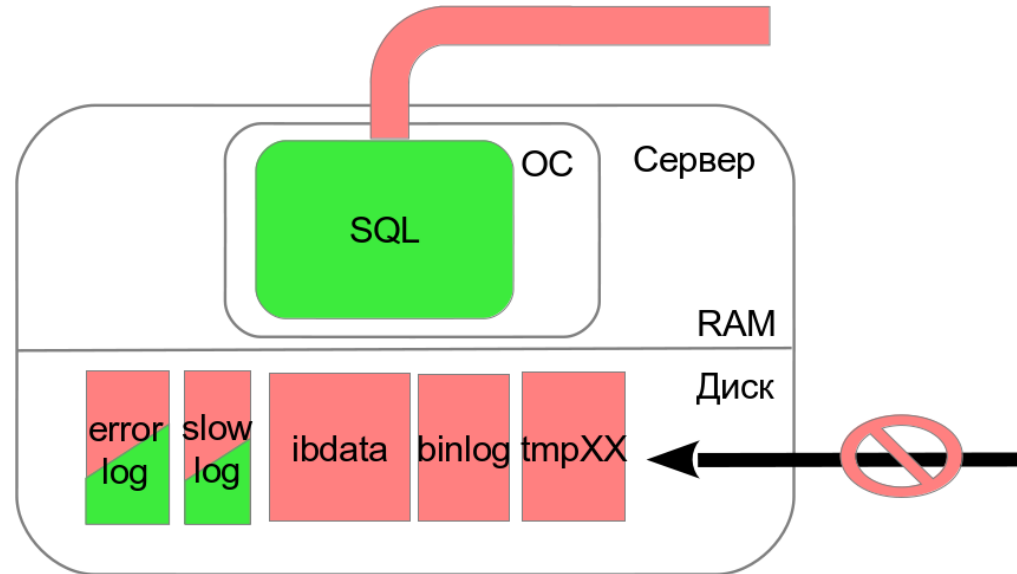
Data-at-rest encryption

- Шифруем всё, что на диске.



Data-at-rest encryption

- Защищает от
 - Копирования или кражи диска
 - Не-root доступа к серверу
- Не защищает от
 - SQL-инъекций
 - Взлома приложения (wordpress'a и тд)
 - Взлома сервера с получением root'a.



Ключи шифрования

- File key management plugin
 - Поставляется с MariaDB
 - Ключ в файле, в открытом виде
 - /dev/shm...
- AWS key management plugin
 - Исходники - в MariaDB
 - Пакеты? Только MariaDB Enterprise
- Eperi Gateway for Databases
 - Коммерческий продукт с закрытым кодом
 - Eperi.de

AWS Key Management Plugin

- AWS Key Management Service
 - Генерирует и хранит ключи
 - Выдает приложениям по требованию
 - Лог выдачи
 - Отзыв права получить ключ
 - ...
- Доступ ко ключам
 - В EC2 - Amazon знает, кто вы
 - Вне EC2: надо аутентифицироваться в Amazon Identity Service.

А что у Oracle?

- Шифрование появилось в MySQL 5.7.11
 - Выпущено 5 февраля 2016 (?)
- Ключами управляют
 - MySQL Community: плагин `keyring_file`
 - MySQL Enterprise Edition: Oracle Key Vault.

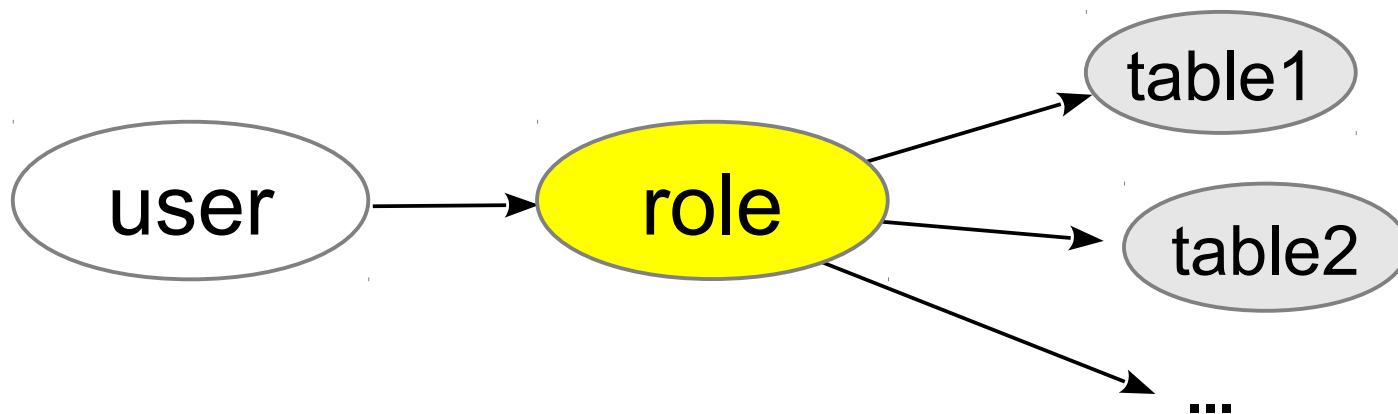
Шифрование - выводы

- Защищает от копирования/кражи диска
 - Полезно для security compliance
 - Не защищает от других атак
- Key Management
 - Ключ из файла
 - Ключ из AWS Key Management Service
 - Ключ из Eperl Gateway

Основные фишки в MariaDB 10.1

- Встроенный Galera Cluster
- Шифрование данных
- **Улучшения в Roles**
- Оптимизатор
 - EXPLAIN и ANALYZE
 - Другие улучшения
- Оптимистичная параллельная репликация

Роли в MariaDB 10.0



```

root> CREATE ROLE developer;
root> GRANT ... TO developer;
root> GRANT developer TO 'user1'@'host';
  
```

```

user1> SET ROLE developer;
user1> DELETE FROM table1; ...
  
```

Обязательно!
В каждом connection!

MariaDB 10.1: роль по умолчанию

```
root> SET DEFAULT ROLE developer FOR 'user1'@'host';
```

```
user1> connect
```

```
user1> DELETE FROM table1;
```

- SET ROLE не требуется!
- Несколько ролей по умолчанию?
 - GRANT role1 TO user1_default_role
 - GRANT role2 TO user1_default_role
 - SET DEFAULT ROLE user1_default_role FOR user1@host

Роли - выводы

- Роли появились в MariaDB 10.0
- MariaDB 10.1 - роль по умолчанию
 - Роли теперь прозрачны для приложений.

Основные фишки в MariaDB 10.1

- Встроенный Galera Cluster
- Шифрование данных
- Улучшения в Roles
- **Оптимизатор**
 - EXPLAIN и ANALYZE
 - Другие улучшения
- Оптимистичная параллельная репликация

EXPLAIN FORMAT=JSON

- Появился в MySQL 5.6
- Имел явные недостатки

Bug#69567: EXPLAIN FORMAT=JSON lists subquery in optimized_away_subqueries, but it is run

Bug#69795: EXPLAIN FORMAT=JSON doesn't show Using filesort for UNION

Bug#74462: EXPLAIN FORMAT=JSON produces ordering_operation when no ordering takes place

Bug#74661: EXPLAIN FORMAT=JSON says two temptables are used, execution shows just one

Bug#74744: EXPLAIN FORMAT=JSON produces duplicates_removal where there is none

[no bug#]: EXPLAIN FORMAT=JSON shows the same subquery as two different subqueries

...

- В MariaDB переписали
 - Показывает правду
 - Больше информации, чем в 5.6
 - Читательнее

ANALYZE для команд

- EXPLAIN показывает план запроса

```

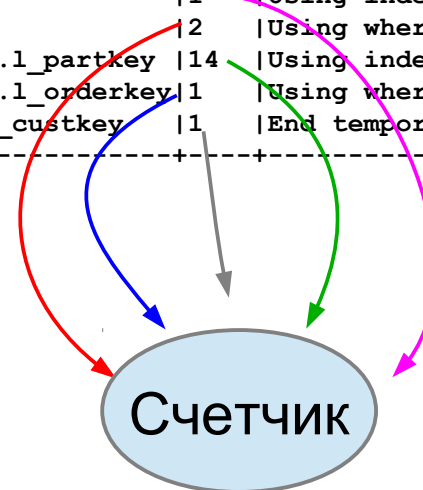
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|id  |select_type|table  |type|possible_keys|key      |key_len|ref              |rows  |Extra          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  1 |SIMPLE     |orders |ALL |PRIMARY,...  |NULL    |NULL   |NULL            |1507320|Using where|
|  1 |SIMPLE     |lineitem|ref |PRIMARY,...  |PRIMARY |4      |orders.o_orderkey|1      |Using where|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

- А что было в реальности ?

EXPLAIN и выполнение

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	PRIMARY	orders	const	PRIMARY	PRIMARY	4	const	1	Using index
1	PRIMARY	lineitem	ref	PRIMARY,i_...	PRIMARY	4	const	2	Using where; Start temporary
1	PRIMARY	lineitem	ref	PRIMARY,i_... i_suppkey		5	lineitem.1_partkey	14	Using index
1	PRIMARY	orders	eq_ref	PRIMARY,i_...	PRIMARY	4	lineitem.1_orderkey	1	Using where
1	PRIMARY	customer	eq_ref	PRIMARY	PRIMARY	4	orders.o_custkey	1	End temporary



- Slow query log: Rows_examined
- SHOW STATUS LIKE 'Handler%'
- SHOW (TABLE|INDEX)_STATISTICS
- performance_schema.table_io_waits_summary_by_table

ANALYZE для команд

```
analyze select *
from lineitem, orders
where o_orderkey=l_orderkey and
      o_orderdate between '1990-01-01' and '1998-12-06' and
      l_extendedprice > 1000000
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	r_rows	filtered	r_filtered	Extra
1	SIMPLE	orders	ALL	PRIMARY,i_...	NULL	NULL	NULL	1504278	1500000	50.00	100.00	Using where
1	SIMPLE	lineitem	ref	PRIMARY,i_...	PRIMARY	4	orders.o_orderkey	2	4.00	100.00	0.00	Using where

Новые колонки: r_ (real)

- r_rows

- r_filtered

- orders:

r_rows ≈ rows, OK

r_filtered = 2 * filtered, сойдет

- lineitem:

r_rows = 2 * rows, OK

r_filtered = 0.00% ??

ANALYZE FORMAT=JSON

EXPLAIN
FORMAT=JSON

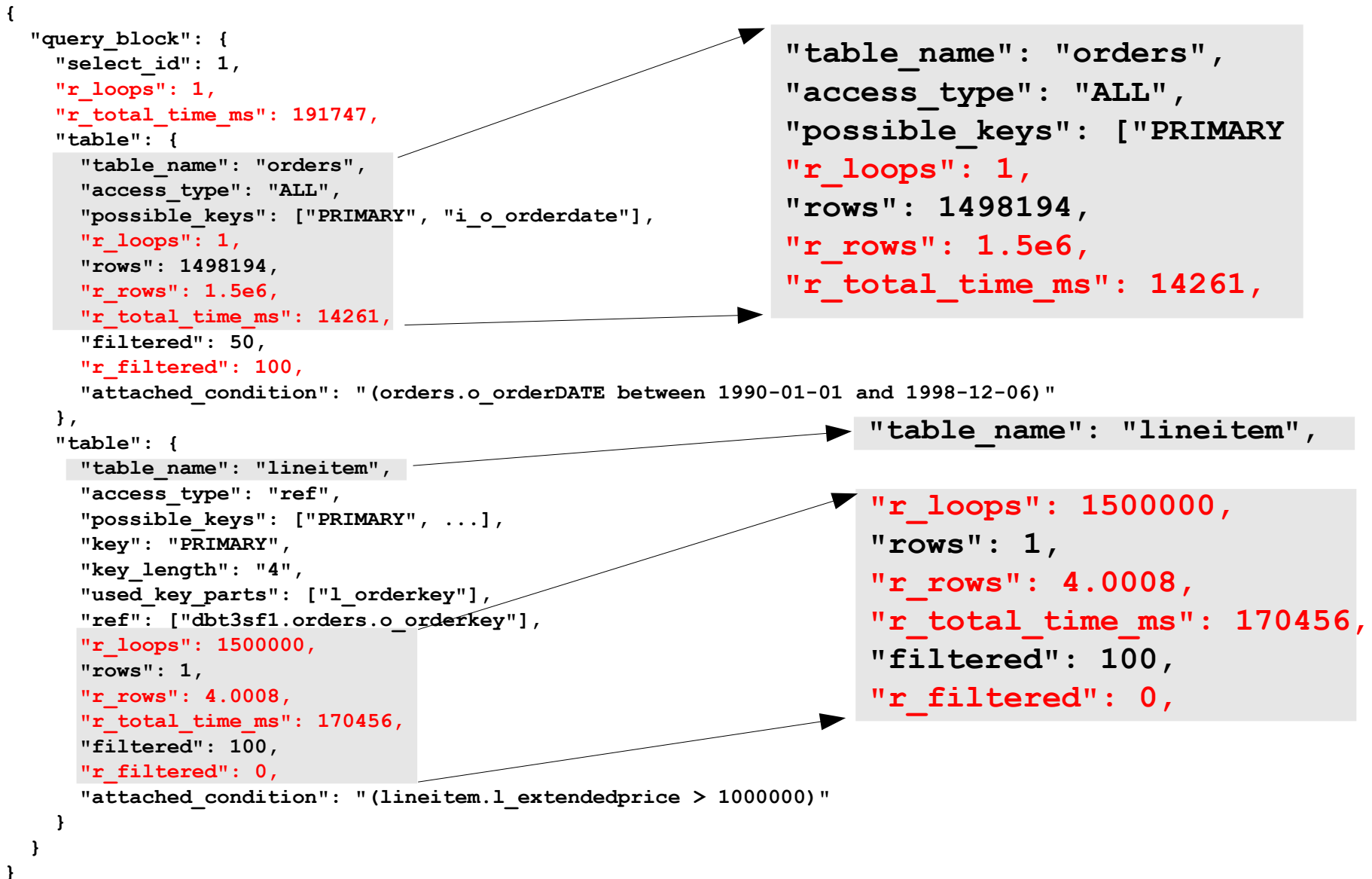
+

ANALYZE

=

ANALYZE FORMAT=JSON

```
{
  "query_block": {
    "select_id": 1,
    "r_loops": 1,
    "r_total_time_ms": 191747,
    "table": {
      "table_name": "orders",
      "access_type": "ALL",
      "possible_keys": ["PRIMARY", "i_o_orderdate"],
      "r_loops": 1,
      "rows": 1498194,
      "r_rows": 1.5e6,
      "r_total_time_ms": 14261,
      "filtered": 50,
      "r_filtered": 100,
      "attached_condition": "(orders.o_orderDATE between 1990-01-01 and 1998-12-06)"
    },
    "table": {
      "table_name": "lineitem",
      "access_type": "ref",
      "possible_keys": ["PRIMARY", ...],
      "key": "PRIMARY",
      "key_length": "4",
      "used_key_parts": ["l_orderkey"],
      "ref": ["dbt3sf1.orders.o_orderkey"],
      "r_loops": 1500000,
      "rows": 1,
      "r_rows": 4.0008,
      "r_total_time_ms": 170456,
      "filtered": 100,
      "r_filtered": 0,
      "attached_condition": "(lineitem.l_extendedprice > 1000000)"
    }
  }
}
```



Поля в ANALYZE FORMAT=JSON

- Элементы плана
 - **r_loops** – число выполнений
 - **r_total_time_ms** – общее время
 - **r_rows** – число записей
- Условия
 - **r_filtered** - %записей после проверки условия
- Join buffer, filesort
 - **r_buffer_size** – Размер буфера
- filesort
 - **r_limit**
 - **r_output_rows**
 - **r_priority_queue_used**
- Subquery cache
 - **r_hit_ratio**
- Range checked for each record
 - **r_indexes**
- ...

ANALYZE - ВЫВОДЫ

- ANALYZE = EXPLAIN + данные от выполнения запроса
- ANALYZE FORMAT=JSON - много информации о выполнении
 - r_total_time_ms!
- Замедление от ANALYZE < 5%.
- Легко найти проблемы в планах запросов

Главные фишки в MariaDB 10.1

- Встроенный Galera Cluster
- Шифрование данных
- Оптимизатор
 - EXPLAIN and ANALYZE
 - Другие улучшения
- Оптимистичная параллельная репликация

SET STATEMENT

```
MariaDB> SET STATEMENT sysvar=... FOR SELECT ...
```

- “Hints” для оптимизатора:

```
SET STATEMENT optimizer_switch=... FOR SELECT ...
```

```
SET STATEMENT  
optimizer_switch=...,  
join_buffer_size=1024*1024*1024  
FOR  
SELECT ...
```

Оптимизации для ORDER BY

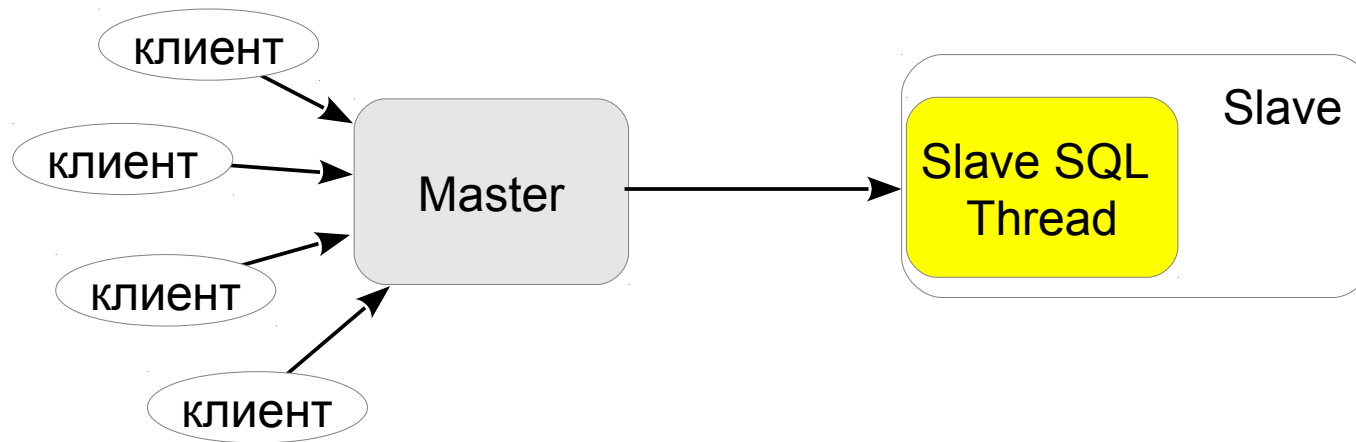
- Несколько фиксов для пользователей/клиентов
- Проблемы с ORDER/GROUP BY?
- Просто попробуйте MariaDB 10.1
- MariaDB 10.1.15+:

```
set optimizer_switch='orderby_uses_equalities=on'
```
- Все еще не работает? Пишите!

Основные фишки в MariaDB 10.1

- Встроенный Galera Cluster
- Шифрование данных
- Оптимизатор
 - EXPLAIN and ANALYZE
 - Другие улучшения
- Оптимистичная параллельная репликация

Параллельная репликация

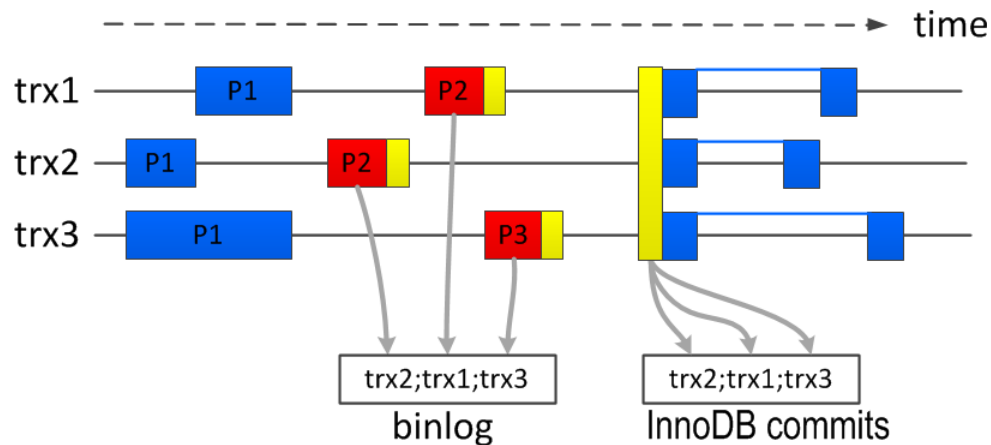


- На Master - много транзакций параллельно
- На Slave - один SQL thread
 - Использует один CPU
 - I/O: коммитит транзакции по одной
 - mk-slave-prefetch
 - “Replication Booster”

Параллельный slave

- MySQL 5.6
 - Транзакции из разных databases
- MariaDB 10.0
 - Маркеры Group Commit
- **MariaDB 10.1**
 - **Оптимистичная параллельная**
- MySQL 5.7
 - Маркеры Group Commit

Group Commit маркеры

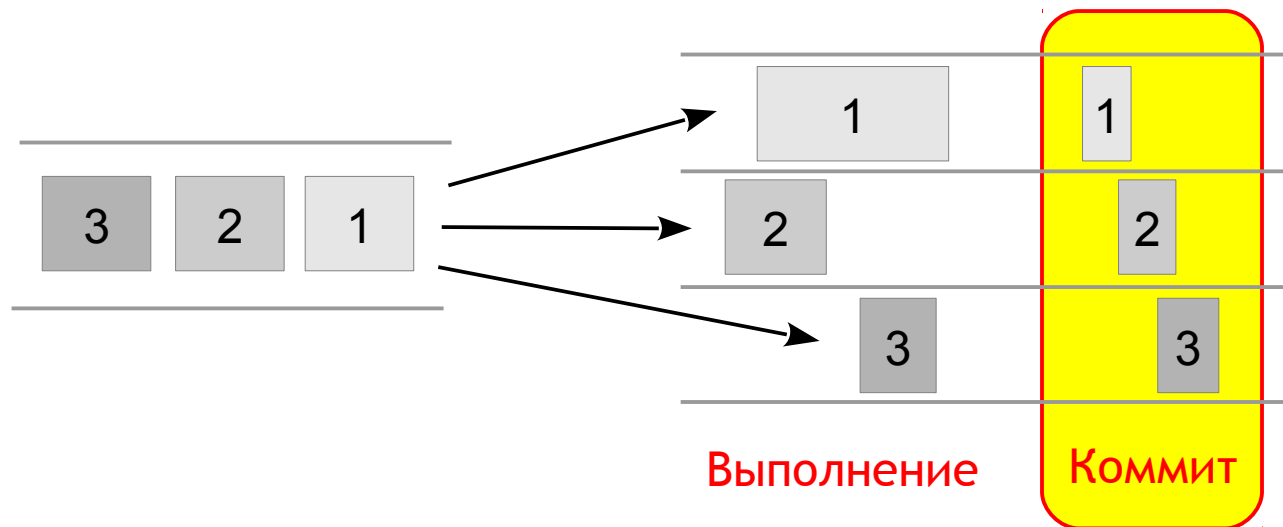


- Master коммитит транзакции группами
- Внутри группы конфликтов нет
 - И на slave не будет*
- Пусть master пишет group marker в binlog
 - А slave применяет группы параллельно

Group Commit маркеры

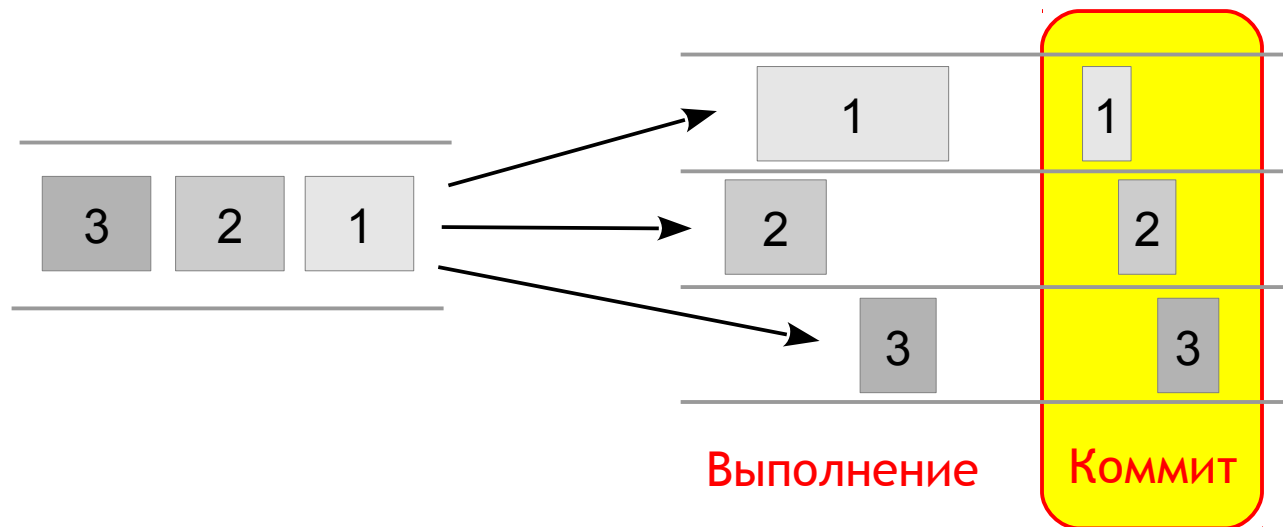
- Размер групп определяется master'ом
 - Не оптимально для slave
 - Быстрый master
 - Последовательное выполнение
- Проблема с SBR
 - Нет конфликтов на master, но есть на slave

Оптимистичный параллельный slave



- Применяем транзакции параллельно
- Коммит группами, в исходном порядке
- Конфликт?
 - Более поздняя транзакция откатывается
 - И запускается повторно

Оптимистичный параллельный slave



- Применяем транзакции параллельно
- Коммит группами, в исходном порядке
- Конфликт?
 - Более поздняя транзакция откатывается
 - И запускается повторно

Параллельный slave - выводы

- MySQL 5.6 - разделение по базе данных
- MariaDB 10.0 - на основе Group Commit
 - MySQL 5.7 - то же самое
- MariaDB 10.1
 - Параллельный slave безо всяких условий.

MariaDB 10.1 - ВЫВОДЫ

- Встроенный Galera Cluster
- Шифрование данных
- Улучшения в Roles
- Оптимизатор
 - EXPLAIN FORMAT=JSON
 - ANALYZE [FORMAT=JSON]
 - Другие улучшения
- Оптимистичная параллельная репликация

Спасибо за внимание!

Вопросы-ответы