GIAC | 全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

预览MySQL Server 8.0新功能

杜修文

2017.12.22

2017.thegiac.com

ORACLE

MySQL

Safe Harbor Statement

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.

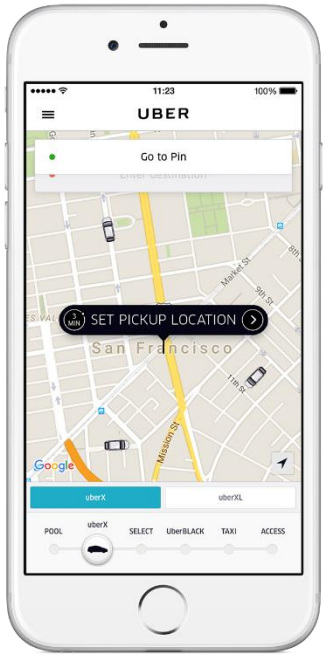
MySQL 5.7 – 全面的改进

200+ new features In total!

- Replication
- InnoDB
- Optimizer
- Security
- Performance Schema
- GIS

- Triggers
- Partitioning
- New! SYS Schema
- New! JSON
- Performance

MySQL 5.7 的故事 *(续.)*



8.0 – 让大家过上好日子!

- 目录下内容的变化
- 让设计师的日子更好过
  - 应因现代网路应用的字符集
  - 改进UUID的储存效率
  - 更完整的JSON函式
  - 更广泛的支持标准SQL
  - 更能应付秒杀场景
- 让DBA的日子更好过
  - 看不见的索引
  - 简化授权
  - 减少监看的代价
  - 更多的监看工具
  - DDL具ACID性
  - 在线设定持久化组态
  - 解决长久以来AUTO_INCREMENT的问题
  - 更好的成本模型带来更好的性能

展开MySQL 8 的datadir之下有什么?

```
-rw-r----- 1 ubuntu ubuntu       56 Sep  6 17:53 auto.cnf
-rw-r----- 1 ubuntu ubuntu     4310 Sep  6 17:58 ib_buffer_pool
-rw-r----- 1 ubuntu ubuntu 12582912 Sep  8 23:06 ibdata1
-rw-r----- 1 ubuntu ubuntu 50331648 Sep  8 23:06 ib_logfile0
-rw-r----- 1 ubuntu ubuntu 50331648 Sep  6 17:53 ib_logfile1
-rw-r----- 1 ubuntu ubuntu 12582912 Sep  9 10:54 ibtmp1
drwxr-x--- 2 ubuntu ubuntu     4096 Sep  6 17:53 mysql/
-rw-r----- 1 ubuntu ubuntu 22020096 Sep  8 23:06 mysql.ibd
-rw-r----- 1 ubuntu ubuntu      257 Sep  6 17:53 performance_sche_3.sdi
drwxr-x--- 2 ubuntu ubuntu     4096 Sep  6 17:53 performance_schema/
drwxr-x--- 2 ubuntu ubuntu     4096 Sep  8 23:05 sakila/
-rw-r----- 1 ubuntu ubuntu      246 Sep  8 23:05 sakila_7.sdi
drwxr-x--- 2 ubuntu ubuntu     4096 Sep  6 17:53 sys/
-rw-r----- 1 ubuntu ubuntu      242 Sep  6 17:53 sys_4.sdi
-rw-r----- 1 ubuntu ubuntu      154 Sep  8 23:02 tablespaces.open.1
-rw-r----- 1 ubuntu ubuntu      390 Sep  8 23:05 tablespaces.open.2
-rw-r----- 1 ubuntu ubuntu 11534336 Sep  8 23:06 undo_001
-rw-r----- 1 ubuntu ubuntu 10485760 Sep  8 23:06 undo_002
```

文件类型的说明

- Undo log由ibdata1中分离
- SDI (Serialized Dictionary Information)
  - 以序列化格式出现的数据字典物件. SDI以JSON format格式存在.
  - SDI的出现为InnoDB表空间文件提供 metadata 的冗余. SDI 可用idb2sdi工具自InnoDB表空间萃取出来.
  - MyISAM 表的SDI存于数据库目录的.sdi超数据文件. 做IMPORT TABLE操作时需要SDI超数据文件.

1. On the source database

```
mysql> FLUSH TABLES hr.employees WITH READ LOCK;
shell> cd export_basedir/data/hr
shell> cp employees_125.sdi /tmp/export
shell> cp employees.{MYD,MYI} /tmp/export
```

2. On the destination database

```
mysql> CREATE SCHEMA hr;
mysql> IMPORT TABLE FROM '/tmp/mexport/employees.sdi‘;
```

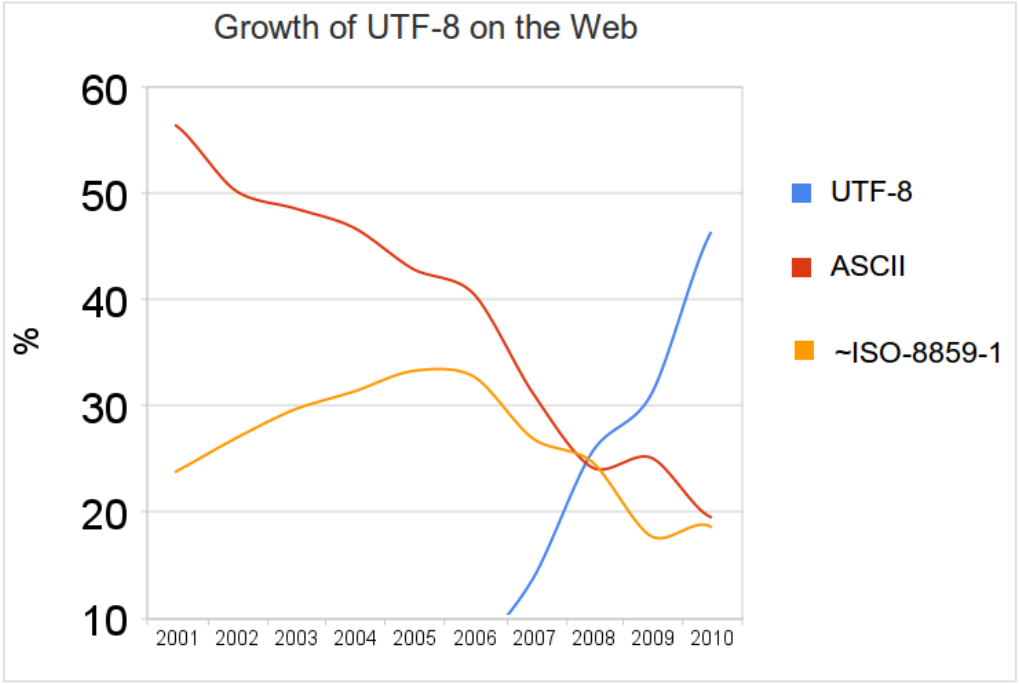
文件类型的说明

- 在回复时用表空间map 文件(tablespaces.open.1 tablespaces.open.2) used during recovery
  - 当文件损坏时可用innodb_scan_ditectories (option of mysqld) 重生成他们

UTF-8

用于网路的字符集

- 当今应用中大部份时UTF-8 字符集
- CJK 用户采用UTF-8
  - 因为要处理外来语的”借用字”
- EN用户采用UTF-8
  - 因为要处理emojis(心情符号)



https://en.wikipedia.org/wiki/UTF-8

UTF-8 (续)

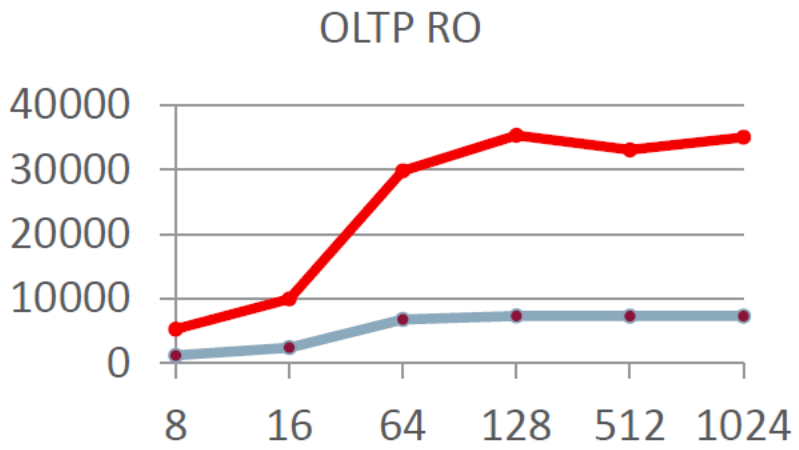
MySQL 8.0

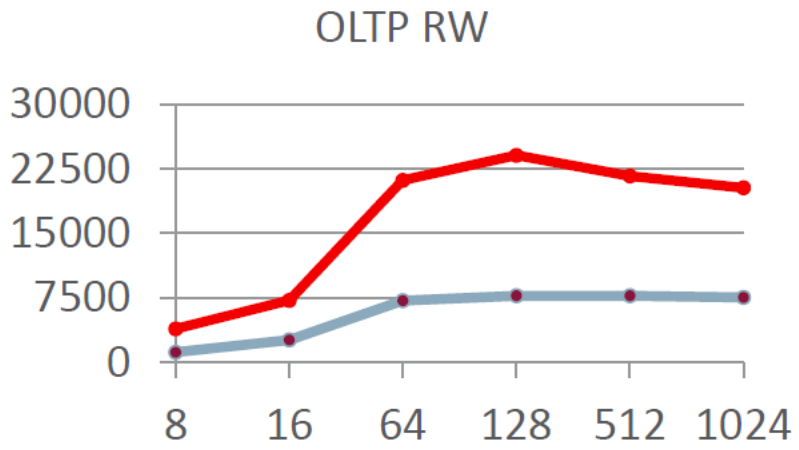
- New! Support for the latest Unicode 9.0 (emoji)

```
select * from information_schema.SCHEMATA;
```

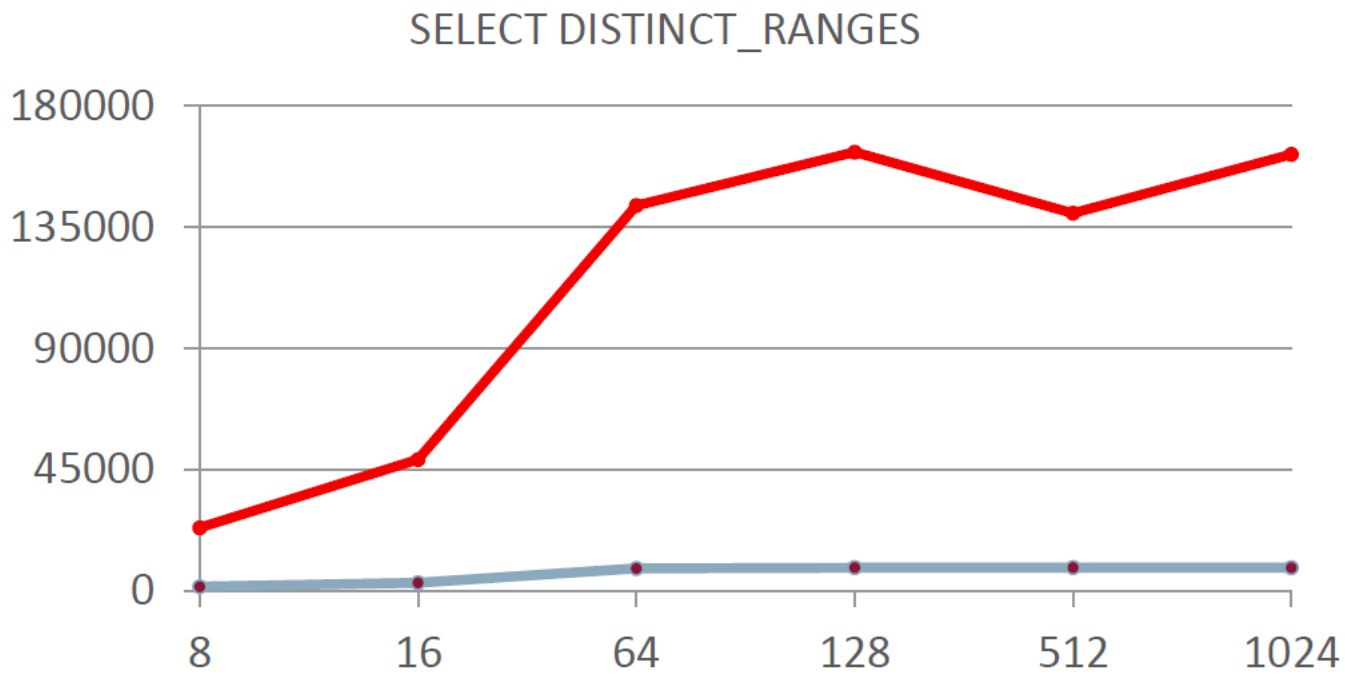
- We are working on per-country collation rules
  - Accent Sensitive
  - Case Sensitive
- Based on UCA DUCET
- **UTF8MB4 as the default character set**
  - Project started in MySQL 5.7
  - Many improvements to reduce performance impact

MySQL 8.0 *vs* MySQL 5.7 utf8mb4





+300-350% in OLTP RO
+176-233% in OLTP RW
+1500-1800% in SELECT DISTINCT_RANGES



新的! UUID 的改进

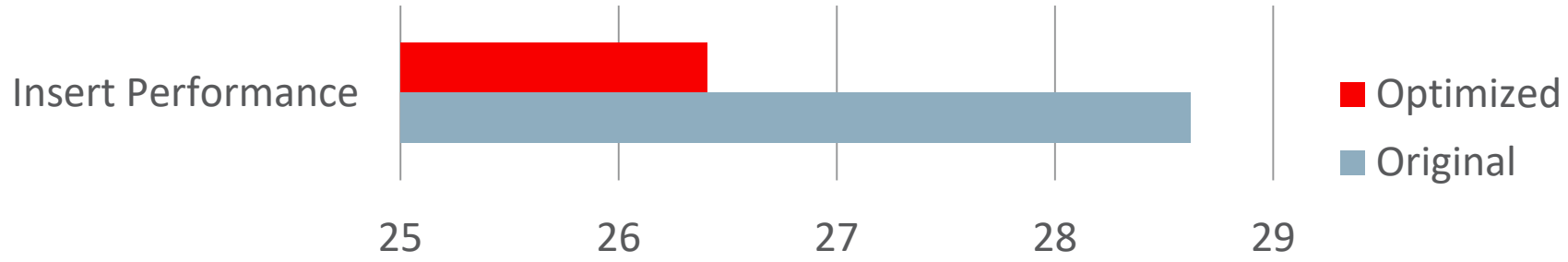
- 在UUID 和二元数据间转换的功能:
  - UUID_TO_BIN()
  - BIN_TO_UUID()
  - *plus* IS_UUID()
- 新功能! 对二元数据类做Bit-wise 操作
  - Designed with IPv6 in mind:
  - INET6_ATON(address) & INET6_ATON(network)

UUID_TO_BIN 的优化

- 现在的Binary格式较小存储也较有效率:

From **BINARY(36)** 53303f87-78fe-11e6-a477-8c89a52c4f3b

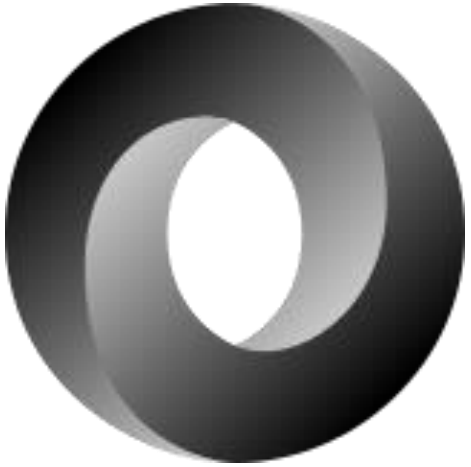
To **BINARY(16)** 11e678fe53303f87a4778c89a52c4f3b



持续改进JSON

MySQL 8.0

- MySQL Document Store
  - JSON_ARRAYAGG()
  - JSON_OBJECTAGG()
- 通过MySQL Shell管理MySQL



Common Table Expressions

- “With queries”
- 支持递归和非递归格式的命令
- 简化复杂的SQL:

```
WITH t1 AS (SELECT * FROM tblA WHERE a='b')
SELECT * FROM t1;
```

示范 – 用CTE做递归查询

```
mysql> show create table ctedemo.emp\G
*************************** 1. row ***************************
       Table: emp
Create Table: CREATE TABLE `emp` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `mgrid` int(11) DEFAULT NULL,
  `name` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `mgrid` (`mgrid`),
  CONSTRAINT `emp_ibfk_1` FOREIGN KEY (`mgrid`) REFERENCES `emp` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=33 DEFAULT CHARSET=utf8mb4
1 row in set (0.00 sec)
```

示范 – 用CTE做递归查询 (续)

```
mysql> select * from emp;
+----+-------+----------------------+
| id | mgrid | name                 |
+----+-------+----------------------+
|  2 |  NULL | CEO                  |
| 10 |     2 | president            |
| 27 |    10 | VP Production        |
| 28 |    10 | VP RD                |
| 29 |    10 | VP Sales             |
| 30 |     2 | CFO                  |
| 31 |    27 | Manufacture Manager  |
| 32 |    27 | Warehouse Manager    |
+----+-------+----------------------+
8 rows in set (0.00 sec)
```

示范 – 用CTE做递归查询 (续)

```
mysql> with recursive empstruct (id, mgrid, name, space, path) as
(select id, mgrid, name, cast('' as char(200)) space, cast(name as char(400)) path from
emp where mgrid is null
union all
select e.id ,e.mgrid, e.name, concat('  ',ep.space) space, concat(ep.path,e.name) path
from empstruct ep, emp e where ep.id = e.mgrid
)
select id, mgrid, concat(space,name) structure from empstruct order by path;
+------+-------+---------------------------+
| id   | mgrid | structure                 |
+------+-------+---------------------------+
|    2 |  NULL | CEO                       |
|   30 |     2 |   CFO                     |
|   10 |     2 |   president               |
|   27 |    10 |     VP Production         |
|   31 |    27 |       Manufacture Manager |
|   32 |    27 |       Warehouse Manager   |
|   28 |    10 |     VP RD                 |
|   29 |    10 |     VP Sales              |
+------+-------+---------------------------+
8 rows in set (0.01 sec)
```

Window Functions

- 汇总和非汇总函式对各行用其他行相关于该行的计算
- *over_clause*:

  {OVER (*window_spec*) | OVER *window_name*}

示范–汇总功能

```
mysql> SELECT year, country, product, profit, SUM(profit) OVER() AS total_profit,
SUM(profit) OVER(PARTITION BY country) AS country_profit
FROM sales
ORDER BY country, year, product, profit;
```

year	country	product	profit	total_profit	country_profit
2000	Finland	Computer	1500	7535	1610
2000	Finland	Phone	100	7535	1610
2001	Finland	Phone	10	7535	1610
2000	India	Calculator	75	7535	1350
2000	India	Calculator	75	7535	1350
2000	India	Computer	1200	7535	1350
2000	USA	Calculator	75	7535	4575
2000	USA	Computer	1500	7535	4575
2001	USA	Calculator	50	7535	4575
2001	USA	Computer	1200	7535	4575
2001	USA	Computer	1500	7535	4575
2001	USA	TV	100	7535	4575
2001	USA	TV	150	7535	4575

Window Functions –非汇总功能

Name	Description
CUME_DIST()	Cumulative distribution value
DENSE_RANK()	Rank of current row within its partition, without gaps
FIRST_VALUE()	Value of argument from first row of window frame
LAG()	Value of argument from row lagging current row within partition
LAST_VALUE()	Value of argument from first row of window frame
LEAD()	Value of argument from row leading current row within partition
NTH_VALUE()	Value of argument from N-th row of window frame
NTILE()	Bucket number of current row within its partition.
PERCENT_RANK()	Percentage rank value
RANK()	Rank of current row within its partition, with gaps
ROW_NUMBER()	Number of current row within its partition

示范 –非汇总功能

```
mysql> SELECT val, ROW_NUMBER() OVER w AS 'row_number', CUME_DIST() OVER w AS
'cume_dist', PERCENT_RANK() OVER w AS 'percent_rank'

FROM numbers

WINDOW w AS (ORDER BY val);
+------+------------+--------------------+--------------+
| val  | row_number | cume_dist          | percent_rank |
+------+------------+--------------------+--------------+
| 1    | 1          | 0.2222222222222222 |            0 |
| 1    | 2          | 0.2222222222222222 |            0 |
| 2    | 3          | 0.3333333333333333 |         0.25 |
| 3    | 4          | 0.6666666666666666 |        0.375 |
| 3    | 5          | 0.6666666666666666 |        0.375 |
| 3    | 6          | 0.6666666666666666 |        0.375 |
| 4    | 7          | 0.8888888888888888 |         0.75 |
| 4    | 8          | 0.8888888888888888 |         0.75 |
| 5    | 9          | 1                  |            1 |
+------+------------+--------------------+--------------+
```

示范 –非汇总功能带上排序

```
mysql> SELECT year, country, product, profit,
ROW_NUMBER() OVER(PARTITION BY country) AS row_num1,
ROW_NUMBER() OVER(PARTITION BY country ORDER BY year, product) AS row_num2 F
ROM sales;
+------+---------+------------+--------+----------+----------+
| year | country | product    | profit | row_num1 | row_num2 |
+------+---------+------------+--------+----------+----------+
| 2000 | Finland | Computer   |   1500 |        2 |        1 |
| 2000 | Finland | Phone      |    100 |        1 |        2 |
| 2001 | Finland | Phone      |     10 |        3 |        3 |
| 2000 | India   | Calculator |     75 |        2 |        1 |
| 2000 | India   | Calculator |     75 |        3 |        2 |
| 2000 | India   | Computer   |   1200 |        1 |        3 |
| 2000 | USA     | Calculator |     75 |        5 |        1 |
| 2000 | USA     | Computer   |   1500 |        4 |        2 |
| 2001 | USA     | Calculator |     50 |        2 |        3 |
| 2001 | USA     | Computer   |   1500 |        3 |        4 |
| 2001 | USA     | Computer   |   1200 |        7 |        5 |
| 2001 | USA     | TV         |    150 |        1 |        6 |
| 2001 | USA     | TV         |    100 |        6 |        7 |
+------+---------+------------+--------+----------+----------+
```

MySQL 8.0: Better Handing of Hot Rows


```
SELECT seat_no
 FROM seats
 JOIN seat_rows USING ( row_no )
WHERE seat_no IN (3,4)
 AND seat_rows.row_no IN (12)
 AND booked = 'NO'
FOR UPDATE OF seats SKIP LOCKED
FOR SHARE OF seat_rows NOWAIT;
```

Non deterministically skip over locked rows

Error immediately if a row is already locked

新功能! 不可见索引

- 索引对优化器是“隐藏” 的
  - 和以前MyISAM的“disabled indexes”不一样
  - 在DML会维护其内容
- 两个用例:
  - 软删除*(Recycle Bin)*
  - 分阶段推出

软删除

Example Usage

- I don’t think this index is used any more:
```
ALTER TABLE Country ALTER INDEX c INVISIBLE;
```
- I need to revert:
```
ALTER TABLE Country ALTER INDEX c VISIBLE;
```
- It is now safe to drop:
```
ALTER TABLE Country DROP INDEX c;
```

分阶段推出

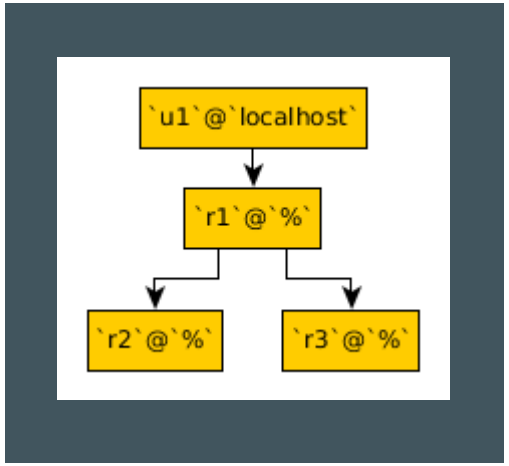
- Adding any new index can change existing execution plans.
- All change introduces risk of regression
- Invisible indexes allows you to stage all changes
  - i.e. put the database in a “prepared” state
  - Turn on changes at an opportune time

```
ALTER TABLE Country ADD INDEX c (Continent) INVISIBLE;
# after some time
ALTER TABLE Country ALTER INDEX c VISIBLE;
```

```
SELECT * FROM information_schema.statistics WHERE is_visible='NO';
*************************** 1. row ***************************
TABLE_CATALOG: def
 TABLE_SCHEMA: world
   TABLE_NAME: Country
   NON_UNIQUE: 1
 INDEX_SCHEMA: world
   INDEX_NAME: c
 SEQ_IN_INDEX: 1
  COLUMN_NAME: Continent
    COLLATION: A
  CARDINALITY: 7
     SUB_PART: NULL
       PACKED: NULL
     NULLABLE:
   INDEX_TYPE: BTREE
      COMMENT: disabled
INDEX_COMMENT:
   IS_VISIBLE: NO
```

新功能! 安控角色

- MySQL 8 provides
- 全功能,弹性的,适当架构的Roles
- DBA们能
  - 建立和删除Roles, 授权给Roles
  - 授权Roles给Roles, 授权Roles 给 Users
  - 限制那些主机可用那些roles, 定义默认的Roles
  - 决定在对话中可用的roles
  - 甚至以SQL函式ROLES_GRAPHML() 将Role视觉化



New! 权限的原子化

- Privilege Tables now 100% InnoDB
- User Management DDLs Atomic
  - CREATE USER
  - ALTER USER
  - RENAME USER
  - DROP USER
  - GRANT
  - REVOKE

降序索引

- DESC in an index definition to store of key values in descending order
- Searching descending index in forward order
- Optimizer uses multiple-column indexes when the most efficient scan order mixes ascending order for some columns and descending order for others

```
CREATE TABLE t (
  c1 INT, c2 INT,
  INDEX idx1 (c1 ASC, c2 ASC),
  INDEX idx2 (c1 ASC, c2 DESC),
  INDEX idx3 (c1 DESC, c2 ASC),
  INDEX idx4 (c1 DESC, c2 DESC)
);
```

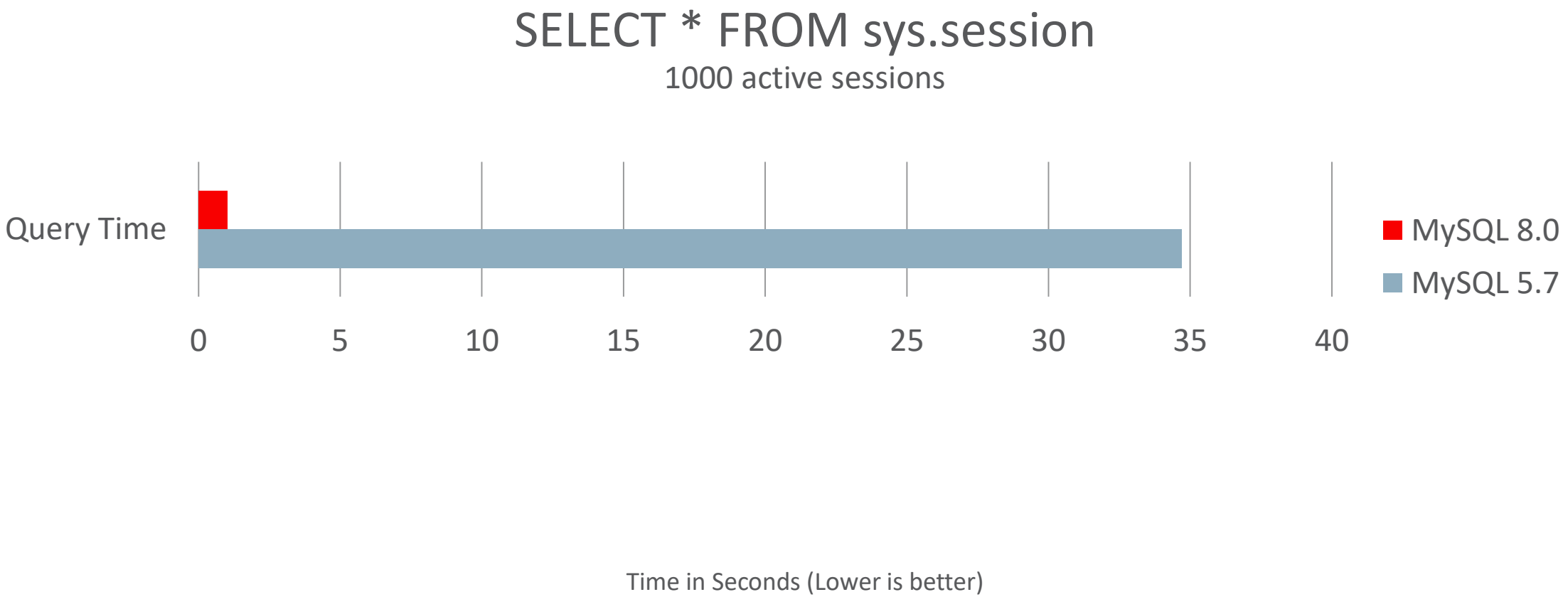
```
ORDER BY c1 ASC, c2 ASC
 -- optimizer can use idx1
ORDER BY c1 DESC, c2 DESC
 -- optimizer can use idx4
ORDER BY c1 ASC, c2 DESC
 -- optimizer can use idx2
ORDER BY c1 DESC, c2 ASC
 -- optimizer can use idx3
```

New! Performance Schema 的索引

- Allows for more efficient access to Performance Schema tables
- A total of **115 indexes** across **93 tables**
- Adds zero overhead
  - A physical index is not maintained internally
  - Implementation of indexes *tricks* the optimizer into better execution plan

Performance 对比

Over 30x faster!



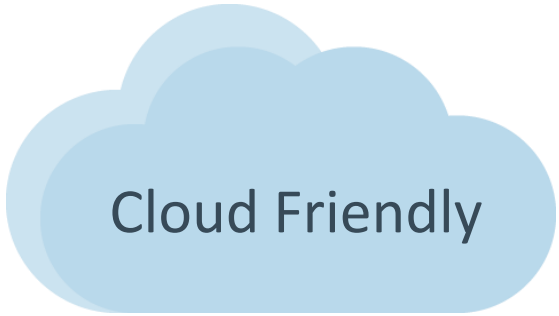
新功能! Performance Schema Instrumenting SQL Errors

Aggregation	Table Name
By Account	events_errors_summary_by_account_by_error
By Host	events_errors_summary_by_host_by_error
By Thread	events_errors_summary_by_thread_by_error
By User	events_errors_summary_by_user_by_error
Global	events_errors_summary_global_by_error

```
SELECT * FROM test.no_table;
ERROR 1146 (42S02): Table 'test.no_table' doesn't exist

SELECT * FROM performance_schema.events_errors_summary_global_by_error
WHERE sum_error_handled > 0 OR SUM_ERROR_RAISED > 0\G
*************************** 1. row ***************************
     ERROR_NUMBER: 1146
       ERROR_NAME: ER_NO_SUCH_TABLE
        SQL_STATE: 42S02
 SUM_ERROR_RAISED: 1
SUM_ERROR_HANDLED: 0
       FIRST_SEEN: 2016-09-11 20:52:42
        LAST_SEEN: 2016-09-11 20:52:42
1 row in set (0.00 sec)
```

新功能! 组态持久化



- Persist GLOBAL Dynamic Server Variables
  - `SET PERSIST sql_mode='STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION';`
- Examples Include:
  - SQL Mode
  - Offline Mode
  - Read Only
- Requires no filesystem access

新功能! Variables Info

Find the source of variables changed on your installation

```
SELECT * FROM performance_schema.variables_info
WHERE variable_source != 'COMPILED';
+-------------------------+-----------------+----------------------+-----------+-----------+
| VARIABLE_NAME           | VARIABLE_SOURCE | VARIABLE_PATH        | MIN_VALUE | MAX_VALUE |
+-------------------------+-----------------+----------------------+-----------+-----------+
| basedir                 | COMMAND_LINE    |                      | 0         | 0         |
| bind_address            | EXPLICIT        | [..]/my.sandbox.cnf  | 0         | 0         |
| datadir                 | COMMAND_LINE    |                      | 0         | 0         |
| foreign_key_checks      | DYNAMIC         |                      | 0         | 0         |
| log_error               | COMMAND_LINE    |                      | 0         | 0         |
| lower_case_table_names  | EXPLICIT        | [..]/my.sandbox.cnf  | 0         | 2         |
| pid_file                | COMMAND_LINE    |                      | 0         | 0         |
| plugin_dir              | COMMAND_LINE    |                      | 0         | 0         |
| port                    | COMMAND_LINE    |                      | 0         | 65535     |
| socket                  | COMMAND_LINE    |                      | 0         | 0         |
| tmpdir                  | EXPLICIT        | [..]/my.sandbox.cnf  | 0         | 0         |
+-------------------------+-----------------+----------------------+-----------+-----------+
11 rows in set (0.00 sec)
```

新功能! 支持ACID的数据词典

- Increased Reliability
- Using InnoDB internally for data dictionary
  - No FRM files
  - No DB.OPT files
  - No TRG files
  - No TRN files
  - No PAR files
- MySQL 8.0 default install no longer contains MyISAM tables.

数据词典ACID

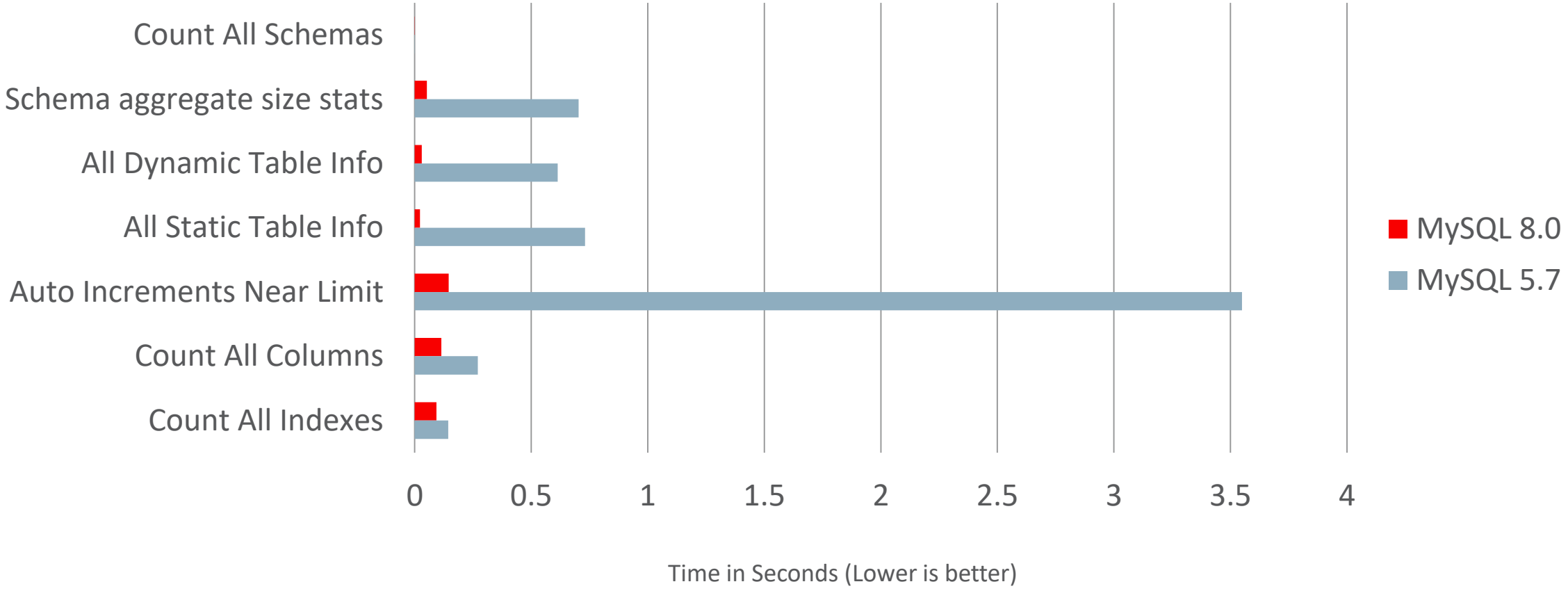
Additional Benefits

- Better cross-platform experience
  - No dependencies on filesystem semantics
- Atomic DDL
  - Better Replication
  - Simplifies server edge cases
- MDL for Foreign Keys
- Flexible Metadata API
  - Easier path to adding new features

Information Schema 的性能

100 schemas times 50 tables (5000 tables)

Already faster at **7/10 queries** in our test suite!



30x Faster

```
SELECT TABLE_SCHEMA,
 TABLE_NAME, TABLE_TYPE,
 ENGINE, ROW_FORMAT
FROM information_schema.tables
WHERE TABLE_SCHEMA LIKE 'db%';
```

Test Performed with 100 schemas, each with 50 tables.

InnoDB Auto Increment Persists

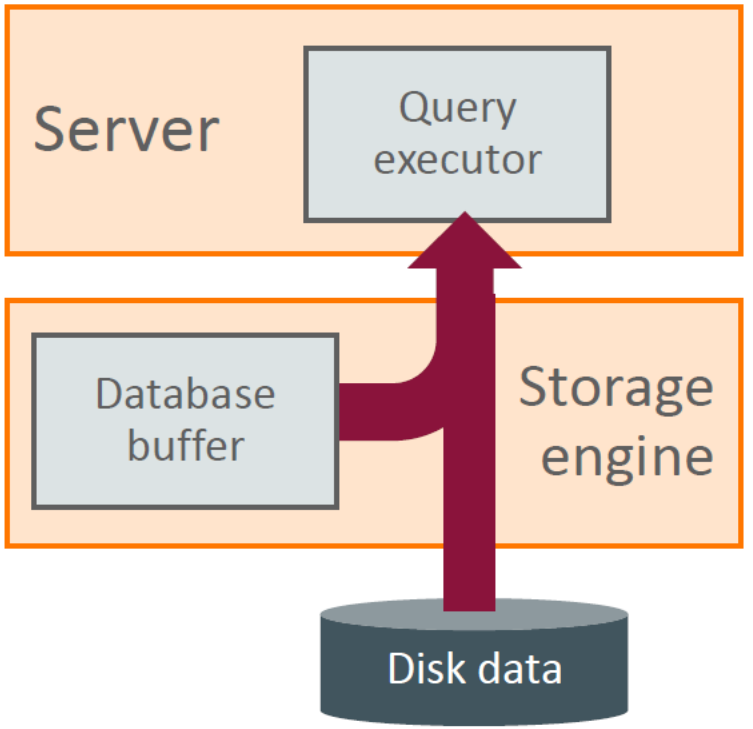
- First reported as BUG #199
- Auto increment counters are now written to the REDO log
- Allows for fast changing meta data

成本模型

- InnoDB buffer estimates for tables and indexes
- Dynamic cost based on memory fit
- Allows for the optimizer to make better query execution decisions

Memory Buffer Aware Cost Estimates

- Storage engines:
  - Estimate for how much of data and indexes are in a memory buffer
  - Estimate for hit rate for memory buffer
- Optimizer cost model:
  - Take into account whether data is already in memory or need to be read from disk



此外,还有更多...

- New! Source code now documented with Doxygen
- New! Plugin Infrastructure!
- Expanded GIS Support
- Expanded Query Hints Support
- Improved Scan Query Performance
- Improved BLOB Storage
- Improved Memcached Interface
- Scalability Improvements
- Parser Refactoring
- New! Document Store
- Improvements to Temporary Tables
- C++11 and Toolchain Improvements
- Replication Applier Thread Progress Reports
- GTID_PURGED always settable

8.0 – 让大家过上好日子!

- 目录下内容的变化
- 让设计师的日子更好过
  - 应因现代网路应用的字符集
  - 改进UUID的储存效率
  - 更完整的JSON函式
  - 更广泛的支持标准SQL
  - 更能应付秒杀场景
- 让DBA的日子更好过
  - 看不见的索引
  - 简化授权
  - 减少监看的代价
  - 更多的监看工具
  - DDL具ACID性
  - 在线设定持久化组态
  - 解决长久以来AUTO_INCREMENT的问题
  - 更好的成本模型带来更好的性能

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

扫码关注GIAC公众号