

PHP 2017·北京

全球开发者大会

高可用的 PHP

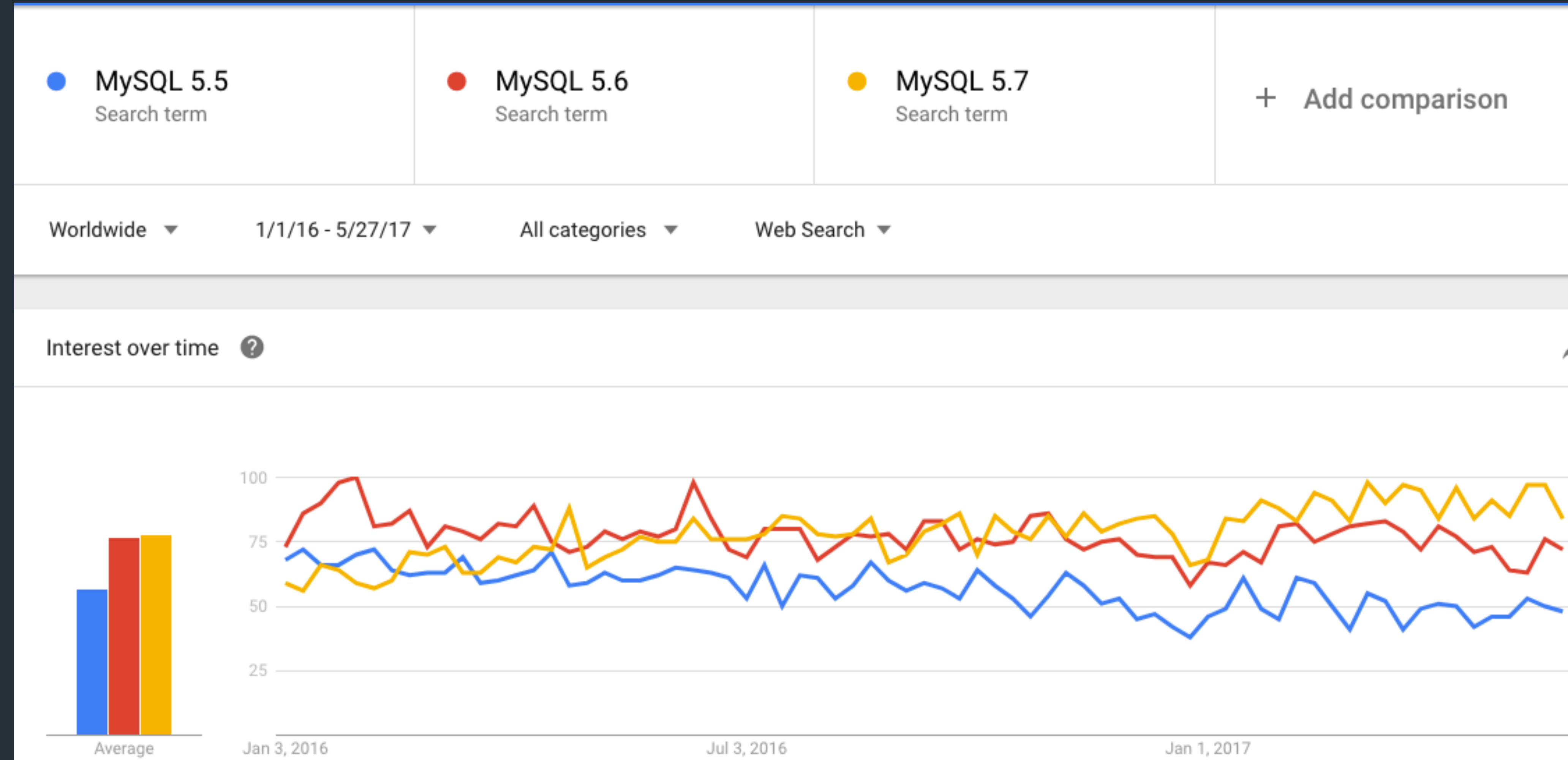
MySQL 5.7优化不求人

知数堂联合创始人 叶金荣

- 知数堂联合创始人
- Oracle MySQL ACE
- MySQL布道师
- 微信：4700963
- 公众号：老叶茶馆
- 专注培养靠谱的互联网攻城狮

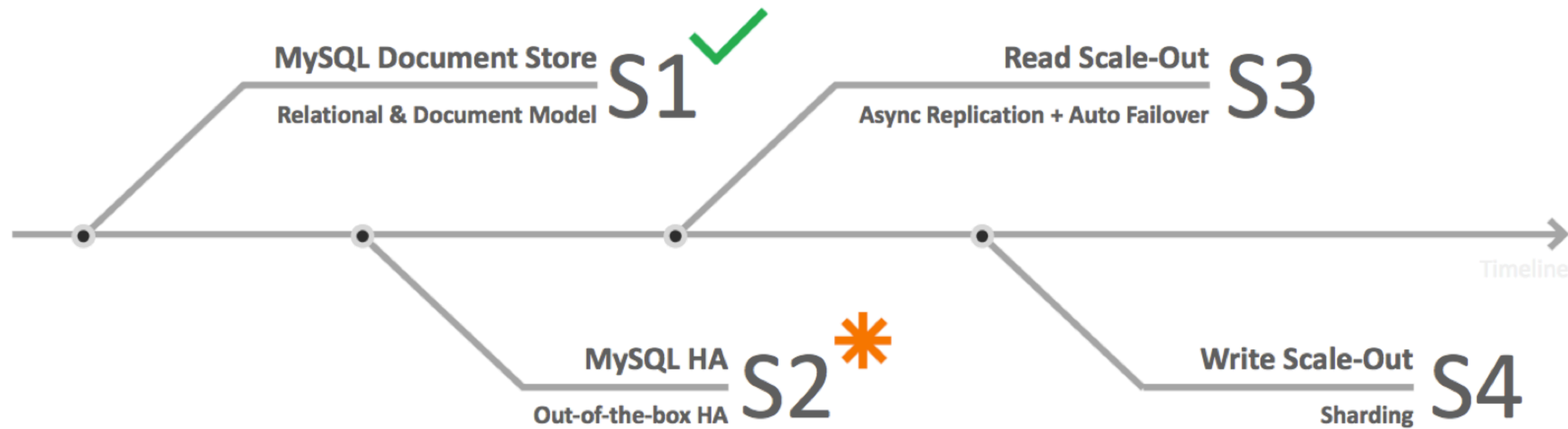


MySQL 5.7搜索趋势



- 2016.11月开始, 5.7占绝对优势
- 2016.12月, 5.7.17版本发布, 支持group replication
- 事实上, 2013.4.23发布5.7, 至今已过去4年多

The Road Ahead

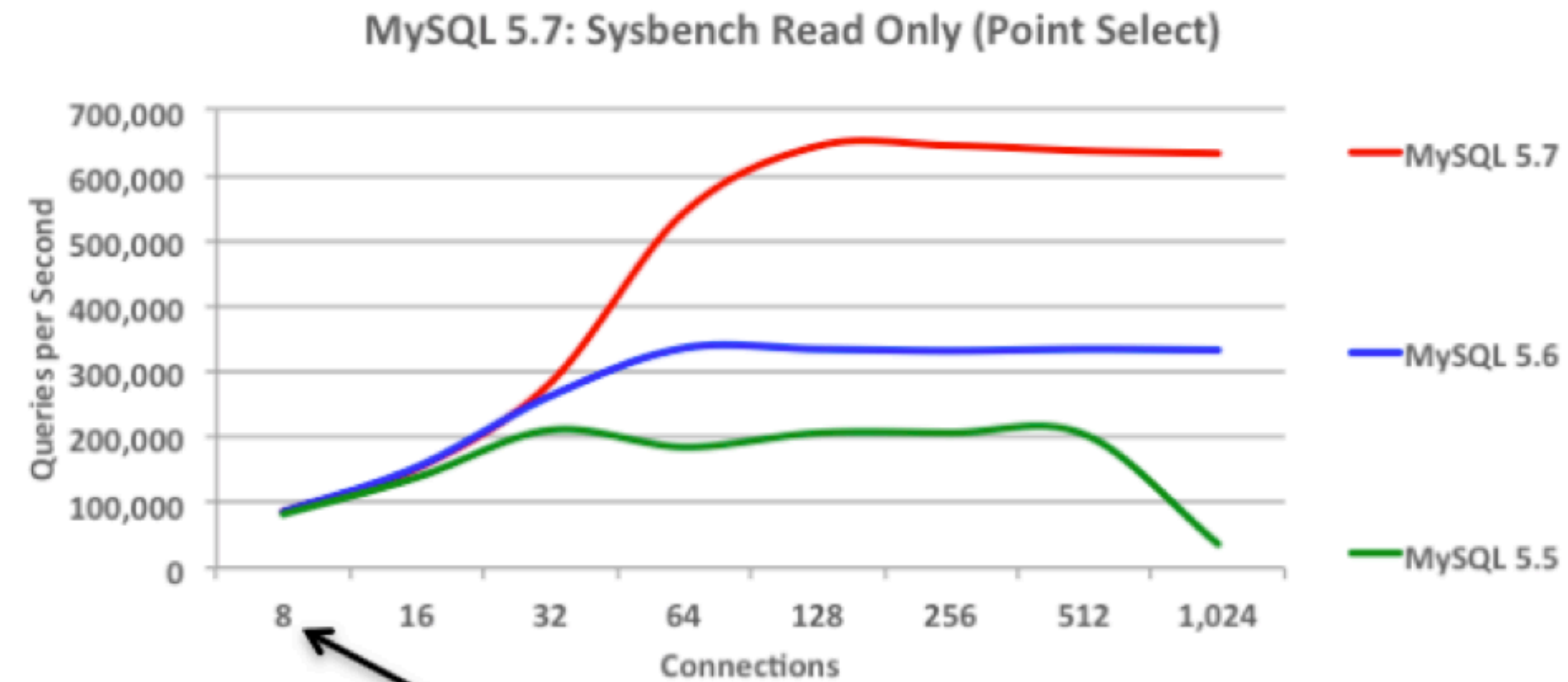




官方号称5.7比5.6性能提升3倍以上

2x Faster than MySQL 5.6
3x Faster than MySQL 5.5

645,000 QPS

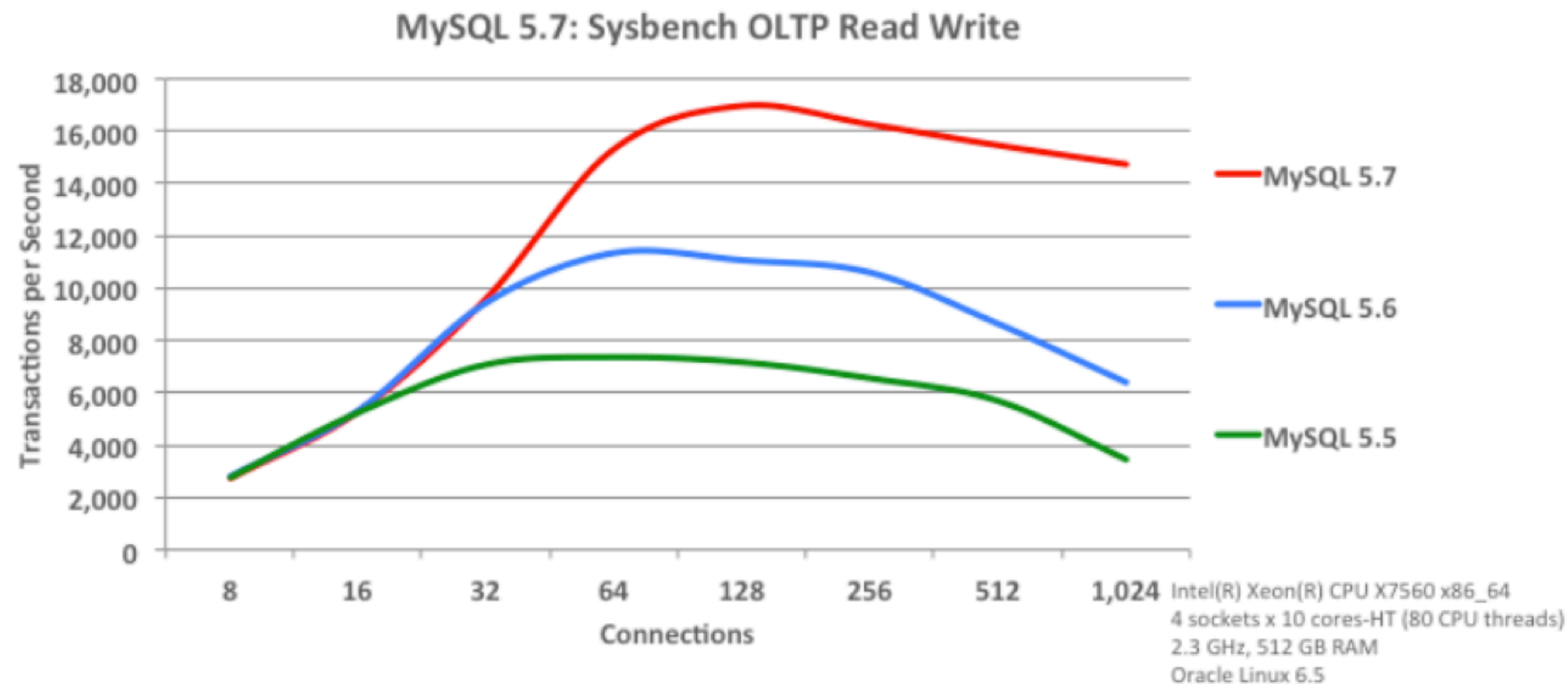


Starts with 8 Threads
What about 2-4 threads?

Intel(R) Xeon(R) CPU E7-4860 x86_64
4 sockets x 10 cores-HT (80 CPU threads)
2.3 GHz, 512 GB RAM
Oracle Linux 6.5

1.5x Faster than MySQL 5.6
2.5x Faster than MySQL 5.5

17,000 TPS



InnoDB引擎增强

1

性能提升

只读事务性能提升；临时表性能提升；page clean效率提升；索引更新效率提升

2

功能提升

buffer pool online resize；varchar in-place enlarge；设备原子写特性检测；

3

其他增强

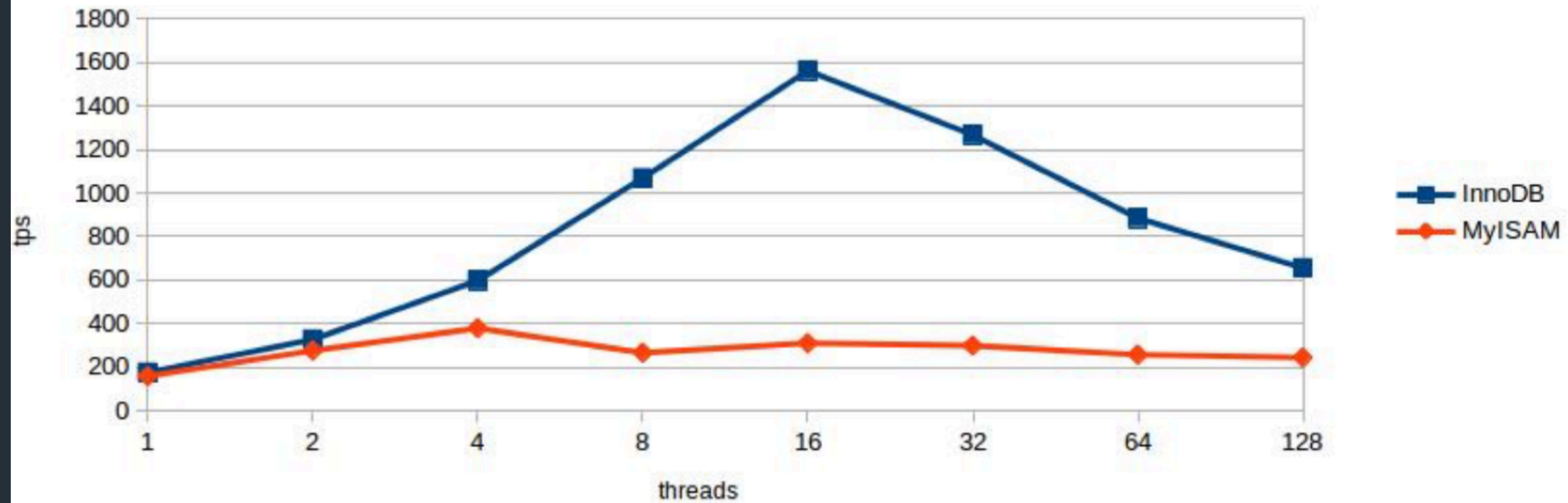
monitor增强；undo log truncate；原生支持表分区；支持通用表空间；



是时候放弃MyISAM了

Feature	MyISAM	InnoDB
Full Text Indexes	yes	Since MySQL 5.6
Portable tables (tablespaces)	yes	Since MySQL 5.6
Spatial Indexes/RTREE (GIS)	yes	<i>Since MySQL 5.7</i>
Last update for table	yes	<i>Since MySQL 5.7</i> <i>(http://dev.mysql.com/worklog/task/?id=6658)</i>
Suitable for temp tables	yes	<i>Since MySQL 5.7</i> <i>Also complex selects uses</i> <i>InnoDB ondisk temp tables</i>
Fast count(*)	yes	<i>*Faster in MySQL 5.7 but</i> <i>does not store counter</i>

InnoDB vs MyISAM (as Optimizer SE)



InnoDB buffer pool功能增强

Online buffer pool resize

- 由小改大几乎无影响；
- 由大改小需要释放部分内存，不过影响也不大，秒级完成。

Buffer pool dump增强

- 设置innodb_buffer_pool_dump_pct，只导出最热那部分数据；
- 系统负载高的时候，会根据innodb_io_capacity设置自动限制buffer pool dump的速度。

VARCHAR in-place enlarge

```
mysql> ALTER TABLE t1 ALGORITHM=INPLACE,  
      CHANGE COLUMN c1 c1 VARCHAR(255);
```

- 255字节长度是个门槛；
- 不跨越255长度门槛即可在线调整；
- 这使得增加VARCHAR列长度毫无压力；
- 也就没必要再为VARCHAR列预留太大长度了；
- 不过，不支持VARCHAR列长度in-place缩减。

其他InnoDB增强

临时表增强

1

临时表使用独立表空间、不记录redo、没有change buffer、锁更少

UNDO log 清除

2

执行innodb_undo_log_truncate=1选项，当undo log超过innodb_max_undo_log_size时自动truncate

page clean 效率提升

3

支持多个page cleaner线程，提高dirty page flush效率

其他增强

4

索引批量更新效率提升3倍；InnoDB原生支持分区，效率更高；自动检测设备是否支持原子写

MySQL复制增强

多源复制

1

支持一从多主，由多个主服务器汇聚到一个从服务器，可用于数据分析或异地容灾、集中备份

并行复制

2

基于LOGICAL_CLOCK时序组提交的并行复制，同时进入PREPARE状态的事务都可以在SLAVE并行应用

组复制

3

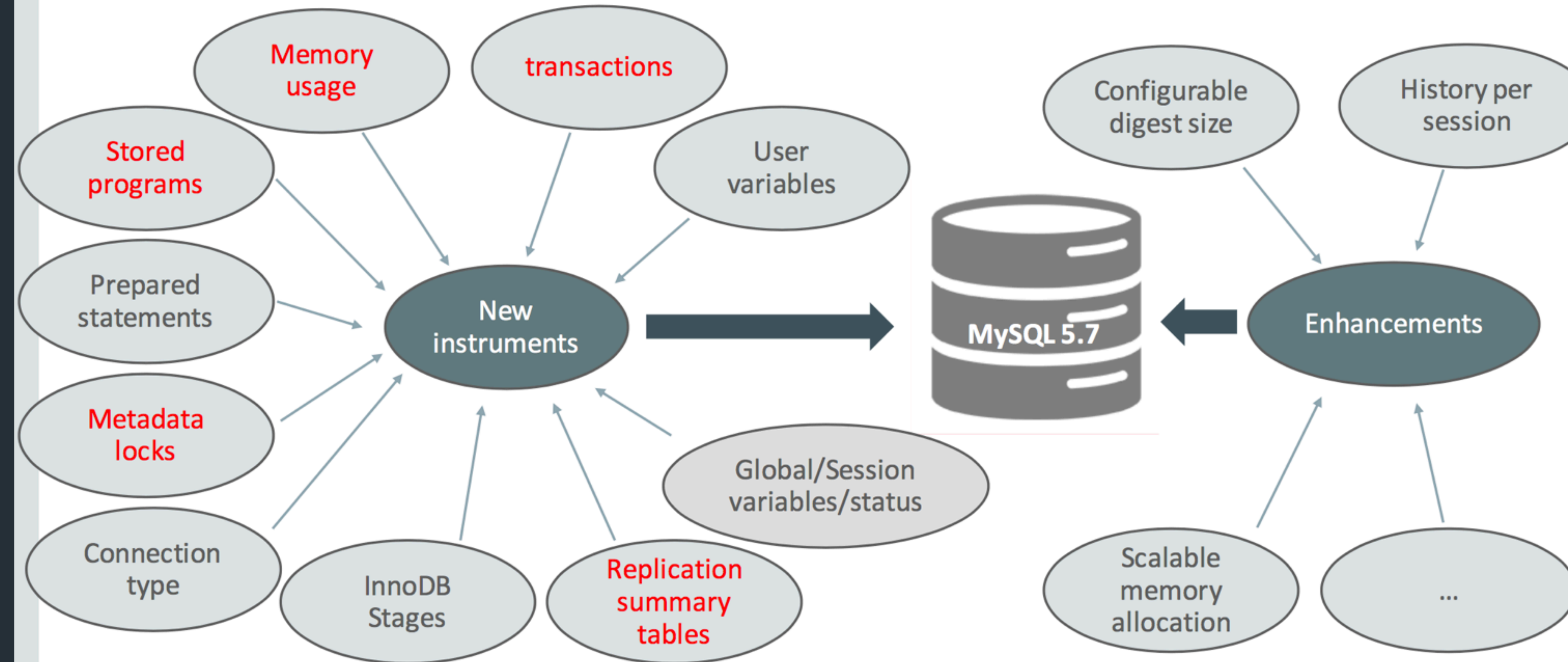
类似Galera Cluster (PXC) 架构，支持多点并行写入，同时提供读负载均衡，只支持InnoDB引擎，所有表必须都有主键，RBR

半同步增强

4

设置master事务提交点AFTER_SYNC 或 AFTER_COMMIT，提高复制可靠性；接收、发送信号线程分离（串行变并行），提高复制效率

MySQL 5.7 : Performance Schema



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

39

sys schema增强

schema table相关统计视图

```
yejr@zhishutang.com (15:24)[sys]> show tables like 'schema%';  
+-----+  
| Tables_in_sys (schema%) |  
+-----+  
| schema_auto_increment_columns |  
| schema_index_statistics |  
| schema_object_overview |  
| schema_redundant_indexes |  
| schema_table_lock_waits |  
| schema_table_statistics |  
| schema_table_statistics_with_buffer |  
| schema_tables_with_full_table_scans |  
| schema_unused_indexes |  
+-----+
```


sys schema增强

table DML统计

```
yejr@zhishutang.com[sys]> select table_name, rows_fetched, rows_inserted, rows_updated, rows_deleted,
                                io_read_requests, io_read, io_write_requests, io_write from schema_table_statistics
                                where table_schema = 'world' and table_name = 'city';
```

table_name	rows_fetched	rows_inserted	rows_updated	rows_deleted	io_read_requests	io_read	io_write_requests	io_write
city	17037	4079	0	0	16	1.49 KiB	67	660.73 KiB

sys schema增强

table index usage统计

```
yejr@zhishutang.com[sys]> select index_name, rows_selected, rows_inserted, rows_updated,
                                rows_deleted from schema_index_statistics where
                                table_schema = 'world' and table_name = 'city' and
                                index_name = 'CountryCode';
```

index_name	rows_selected	rows_inserted	rows_updated	rows_deleted
ID	18131	0	0	0
CountryCode	2	0	2	0

sys schema增强

redundant indexes

```
yejr@zhishutang.com[sys]> select * from schema_redundant_indexes\G
***** 1. row *****
      table_schema: world
      table_name: countrylanguage
    redundant_index_name: CountryCode
    redundant_index_columns: CountryCode
    redundant_index_non_unique: 1
      dominant_index_name: PRIMARY
    dominant_index_columns: CountryCode,Language
    dominant_index_non_unique: 0
      subpart_exists: 0
    sql_drop_index: ALTER TABLE `world`.`countrylanguage` DROP INDEX `CountryCode`
```

sys schema增强

unused indexes

```
yejr@zhishutang.com[sys]> select * from schema_unused_indexes;
```

object_schema	object_name	index_name
sakila	actor	idx_actor_last_name
sakila	address	idx_fk_city_id
sakila	address	idx_location
sakila	city	idx_fk_country_id
sakila	customer	idx_fk_store_id
sakila	customer	idx_fk_address_id
sakila	customer	idx_last_name
sakila	film	idx_title
sakila	film	idx_fk_language_id
sakila	film	idx_fk_original_language_id
sakila	film_actor	idx_fk_film_id
sakila	film_category	fk_film_category_category
sakila	film_text	idx_title_description

sys schema增强

row lock wait

```
yejr@imysql.com [sys]>select * from sys.innodb_lock_waits\G
***** 1. row *****
      wait_started: 2017-02-13 15:32:54
      wait_age: 00:00:30
      wait_age_secs: 30
      locked_table: `test`.`t1`
      locked_index: PRIMARY
      locked_type: RECORD
      waiting_trx_id: 248632
      waiting_trx_started: 2017-02-13 15:32:54
      waiting_trx_age: 00:00:30
      waiting_trx_rows_locked: 1
      waiting_trx_rows_modified: 0
      waiting_pid: 24
      waiting_query: select * from t1 where c1=2 for update
      waiting_lock_id: 248632:293:3:3
      waiting_lock_mode: X
      blocking_trx_id: 248631
      blocking_pid: 22
      blocking_query: NULL
      blocking_lock_id: 248631:293:3:3
      blocking_lock_mode: X
      blocking_trx_started: 2017-02-13 15:32:37
      blocking_trx_age: 00:00:47
      blocking_trx_rows_locked: 1
      blocking_trx_rows_modified: 0
      sql_kill_blocking_query: KILL QUERY 22
      sql_kill_blocking_connection: KILL 22
```

sys schema增强

metadata lock wait

```
yejr@zhishutang.com[sys]> select * from schema_table_lock_waits limit 4\G
***** 1. row *****
      object_schema: world
      object_name: city
      waiting_thread_id: 108
      waiting_pid: 72
      waiting_account: root@localhost
      waiting_lock_type: SHARED_READ
      waiting_lock_duration: TRANSACTION
      waiting_query: select * from city limit 4
      waiting_query_secs: 21
      waiting_query_rows_affected: 0
      waiting_query_rows_examined: 0
      blocking_thread_id: 106
      blocking_pid: 70
      blocking_account: root@localhost
      blocking_lock_type: SHARED_NO_READ_WRITE
      blocking_lock_duration: TRANSACTION
      sql_kill_blocking_query: KILL QUERY 70
      sql_kill_blocking_connection: KILL 70

yejr@zhishutang.com[sys]> show processlist;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Id | User | Host      | db  | Command | Time | State | Info |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 70 | root | localhost | world | Sleep   | 49 |      | NULL |
| 71 | root | localhost | sys  | Query   | 0  | starting | show processlist |
| 72 | root | localhost | world | Query   | 31 | Waiting for table metadata lock | select * from city limit 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```


sys schema增强

I/O写最多的文件

```
yejr@imysql.com [sys]>select * from x$io_global_by_file_by_bytes
-> order by total_written desc limit 1\G
***** 1. row *****
      file: /Volumes/DATA/mysql/test/sid.ibd
count_read: 3046
total_read: 49954816
  avg_read: 16400.1366
count_write: 4054
total_written: 95322112
  avg_write: 23513.1011
      total: 145276928
write_pct: 65.61
```

sys schema增强

热门SQL TOP 10

```
yejr@zhishuedu.com [sys]>select db,exec_count,query from statement_analysis order by exec_count desc limit 10;
```

db	exec_count	query
yejr	299993	SET @? = `round` (`rand` () * ?) ;
yejr	299987	SET @? = `round` (`rand` () * ?) ;
test	15997	SELECT @@`version_comment` LIMIT ?
test	14004	UPDATE `t1` SET `c4` = ? WHERE `c1` = ?
test	2000	INSERT INTO `t1` SELECT ?, ...
information_schema	92	SHOW ENGINE `innodb` STATUS
test	54	SELECT ?
sys	37	SELECT IF (`isnull` (`perfor ... _host_by_event_name` GROUP BY
NULL	32	SHOW PROCESSLIST
test	29	SHOW TABLES

sys schema增强

查看实例消耗内存

```
yejr@zhishutang.com[sys]> select * from memory_global_total;  
+-----+  
| total_allocated |  
+-----+  
| 3.48 GiB      |  
+-----+
```

sys schema增强

查看内部对象内存消耗

yejr@imysql> select event_name,SUM_NUMBER_OF_BYTES_ALLOC from

memory_summary_global_by_event_name

order by SUM_NUMBER_OF_BYTES_ALLOC desc LIMIT 10;

event_name	SUM_NUMBER_OF_BYTES_ALLOC
memory/innodb/mem0mem	23839472166
memory/sql/Filesort_buffer::sort_k	2754080008
memory/memory/HP_PTRS	2632950064
memory/sql/thd::main_mem_root	1830295888
memory/mysys/IO_CACHE	977629336
memory/sql/String::value	946131984
memory/sql/TABLE	409615108
memory/mysys/MY_DIR	300412224
memory/sql/test_quick_select	178511200
memory/sql/QUICK_RANGE_SELECT	173118272

sys schema增强

查看线程内存消耗

yejr@imysql>select event_name, SUM_NUMBER_OF_BYTES_ALLOC from
memory_summary_by_thread_by_event_name

order by SUM_NUMBER_OF_BYTES_ALLOC desc limit 20;

thread_id	event_name	SUM_NUMBER_OF_BYTES_ALLOC
1	memory/innodb/buf_buf_pool	139722752
1	memory/sql/Log_event	41576778
1	memory/sql/THD::Session_tracker	35384895
1	memory/sql/thd::main_mem_root	33832032
1	memory/sql/NET::buff	33402726
1239	memory/innodb/mem0mem	30407948
2327	memory/innodb/mem0mem	30407948
1220	memory/innodb/mem0mem	30407948

Optimizer增强

- UNION ALL不再产生临时表（除非需要排序）
- IN子查询效率提升
- 全文检索效率提升
- 排序效率提升

EXPLAIN增强

- 查看当前活跃SESSION的SQL执行计划
 - EXPLAIN FOR CONNECTION 3306
- JSON格式的输出结果中，还能看到执行计划代价信息
- 总是启用PARTITIONS/EXTENDED选项

JSON & Generated Columns

```
Create Table: CREATE TABLE `json_test` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `data` json DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8
```

```
yejr@imysql.com [test]>select * from json_test where data->'$.type' = 'line';
```

```
+----+-----+  
| id | data |  
+----+-----+  
|  2 | {"type": "line", "coordinates": [-87.9101245, 41.7585879]} |  
+----+-----+  
1 row in set (0.00 sec)
```

```
yejr@imysql.com [test]>select * from json_test where json_extract(data,'$.type') = 'line';
```

```
+----+-----+  
| id | data |  
+----+-----+  
|  2 | {"type": "line", "coordinates": [-87.9101245, 41.7585879]} |  
+----+-----+
```


JSON & Generated Columns

```
yejr@imysql.com [test]>explain select * from json_test where data->'$.type' = 'line'\G
***** 1. row *****
      id: 1
select_type: SIMPLE
      table: json_test
  partitions: NULL
        type: ALL
possible_keys: NULL
          key: NULL
       key_len: NULL
         ref: NULL
        rows: 2
   filtered: 100.00
      Extra: Using where
```

JSON & Generated Columns

```
yejr@imysql.com [test]>alter table json_test add data_type varchar(255) GENERATED ALWAYS AS (data->'$.type') VIRTUAL;  
Query OK, 0 rows affected (0.04 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
yejr@imysql.com [test]>alter table json_test add key (data_type);  
Query OK, 0 rows affected (0.03 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
yejr@imysql.com [test]>explain select * from json_test where data_type = 'line';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	json_test	NULL	ref	data_type	data_type	768	const	1	100.00	NULL

JSON & Generated Columns

```
Create Table: CREATE TABLE `sid` (  
  `id` int(10) unsigned NOT NULL DEFAULT '0',  
  `name` varchar(50) NOT NULL DEFAULT '',  
  `aid` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `nid` int(11) unsigned GENERATED ALWAYS AS ((`id` + 1)) VIRTUAL NOT NULL,  
  `nnid` int(11) unsigned GENERATED ALWAYS AS ((`id` + 1)) STORED NOT NULL,  
  PRIMARY KEY (`aid`),  
  KEY `name` (`name`),  
  KEY `id` (`id`),  
  KEY `nid` (`nid`)  
) ENGINE=InnoDB AUTO_INCREMENT=893210 DEFAULT CHARSET=utf8
```


JSON & Generated Columns

```
yejr@imysql.com [test]>desc select * from sid where id+1=1024\G
***** 1. row *****
      id: 1
select_type: SIMPLE
      table: sid
  partitions: NULL
        type: ref
possible_keys: nid
          key: nid
      key_len: 4
         ref: const
        rows: 8
   filtered: 100.00
      Extra: NULL
```

设定SELECT SQL超时

- `SELECT /*+ MAX_EXECUTION_TIME(1) */ * FROM t`
- 也可修改 `max_statement_time` 选项
- 这个特性非常实用，有效避免某个垃圾SQL引发雪崩
- 参考：MariaDB/Percona版本选项 `innodb_kill_idle_transaction`

其他新特性

单表支持多个触发器

- 可以设定 `trigger_order`, 调整不同触发器的优先级
- 兼容 `pt-osc` 等工具

默认启用严格SQL_MODE

- `STRICT_ALL_TABLES`、`STRICT_TRANS_TABLES`
- 规避一些容易混淆的操作, 比如超长内容自动被截断、除零、用 `'0000-00-00'` 表示日期、写入不同类型的数据、不能在 `GRANT` 中同时创建用户等

MySQL 8.0新特性

- Optimizer增强，如JOIN优化（可能支持hash join）、HINT增强
- 新增支持不可见索引、倒序索引
- CTE(公用表表达式，Common Table Expression)功能及windowing统计函数
- 直方图、UUID优化、减少handler API调用执行计划缓存（可能）
- 所有metadata全部存储到InnoDB中
- 增加user role特性，权限管理更方便
- InnoDB表自增列最大值持久化，再也不用担心实例重启后最大自增Id丢失了

MySQL 8.0新特性

- InnoDB memcached支持mget指令
- innodb_deadlock_detect在线动态调整，在高负载环境下，建议关闭死锁检测
- InnoDB加密除了支持data page外，增加支持redo log和undo log，提高保密性
- SELECT ... FOR SHARE/UPDATE增加NOWAIT 和 SKIP LOCKED选项
- JSON功能增强，JSON列更新、数据排序效率提升
- 默认使用utf8mb4字符集
- SET PERSIST语法支持修改options后持久化

PHP 2017·北京

全球开发者大会

