DTCC

2017第八届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2017

你知我知的SQL优化

陈兴隆
15811381141
771980961@qq.com

SQL优化的正确姿势

- 需要优化的SQL在哪里？
- SQL优化原则
- SQL优化案例
- 全量统计优化/MariaDB ColumnStore

需要优化的SQL在哪儿

- 数据库版本MariaDB 10.1.21/ColumnStore 1.0.8
- my.cnf

  slow_query_log=1

  long_query_time=0.1
- Tools

  (1) pt-query-digest slow.log > slow_report.log

  (2) SQL> EXPLAIN SELECT * FROM <table_name>

  (3) SQL> SET PROFILING = 1;

  SQL> SELECT * FROM <table_name>;

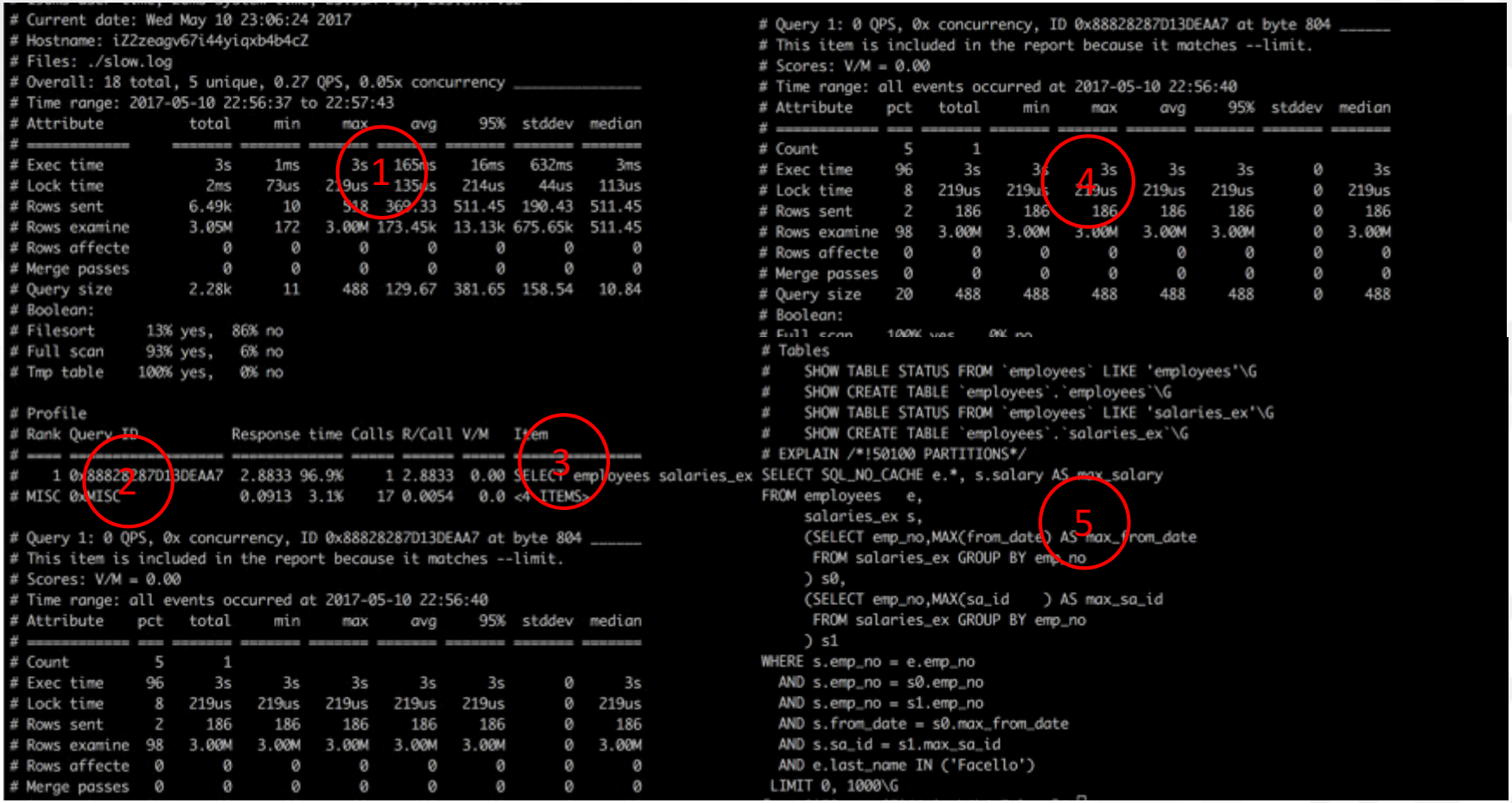
  SQL> SHOW PROFILE;

  SQL> SELECT * FROM information_schema.PROFILING;

  (4) shell&&orzdba&&doDBA

  (5) MariaDB ColumnStore

需要优化的SQL在哪儿



```
# Current date: Wed May 10 23:06:24 2017
# Hostname: iZ2zeagv67i44yiqxb4b4cZ
# Files: ./slow.log
# Overall: 18 total, 5 unique, 0.27 QPS, 0.05x concurrency _____________
# Time range: 2017-05-10 22:56:37 to 22:57:43
# Attribute          total     min     max     avg     95%  stddev  median
# ============     ======= ======= ======= ======= ======= ======= =======
# Exec time             3s     1ms      3s   165ms    16ms   632ms     3ms
# Lock time             2ms    73us   219us   135us   214us    44us   113us
# Rows sent           6.49k      10     518  369.33  511.45  190.43  511.45
# Rows examine        3.05M     172   3.00M 173.45k  13.13k 675.65k  511.45
# Rows affecte            0       0       0       0       0       0       0
# Merge passes            0       0       0       0       0       0       0
# Query size          2.28k      11     488  129.67  381.65  158.54   10.84
# Boolean:
# Filesort       13% yes,  86% no
# Full scan      93% yes,   6% no
# Tmp table     100% yes,   0% no

# Profile
# Rank Query ID           Response time Calls R/Call V/M   Item
# ==== ================== ============= ===== ====== ===== ==============
#    1 0x88828287D13DEAA7  2.8833 96.9%     1 2.8833  0.00 SELECT employees salaries_ex
# MISC 0xMISC              0.0913  3.1%    17 0.0054   0.0 <4 ITEMS>

# Query 1: 0 QPS, 0x concurrency, ID 0x88828287D13DEAA7 at byte 804 ______
# This item is included in the report because it matches --limit.
# Scores: V/M = 0.00
# Time range: all events occurred at 2017-05-10 22:56:40
# Attribute    pct   total     min     max     avg     95%  stddev  median
# ============ === ======= ======= ======= ======= ======= ======= =======
# Count          5       1
# Exec time     96      3s      3s      3s      3s      3s       0      3s
# Lock time      8   219us   219us   219us   219us   219us       0   219us
# Rows sent      2     186     186     186     186     186       0     186
# Rows examine  98   3.00M   3.00M   3.00M   3.00M   3.00M       0   3.00M
# Rows affecte   0       0       0       0       0       0       0       0
# Merge passes   0       0       0       0       0       0       0       0
```

```
# Query 1: 0 QPS, 0x concurrency, ID 0x88828287D13DEAA7 at byte 804 ______
# This item is included in the report because it matches --limit.
# Scores: V/M = 0.00
# Time range: all events occurred at 2017-05-10 22:56:40
# Attribute    pct   total     min     max     avg     95%  stddev  median
# ============ === ======= ======= ======= ======= ======= ======= =======
# Count          5       1
# Exec time     96      3s      3s      3s      3s      3s       0      3s
# Lock time      8   219us   219us   219us   219us   219us       0   219us
# Rows sent      2     186     186     186     186     186       0     186
# Rows examine  98   3.00M   3.00M   3.00M   3.00M   3.00M       0   3.00M
# Rows affecte   0       0       0       0       0       0       0       0
# Merge passes   0       0       0       0       0       0       0       0
# Query size    20     488     488     488     488     488       0     488
# Boolean:
# Full scan     100% yes    0% no
# Tables
#    SHOW TABLE STATUS FROM `employees` LIKE 'employees'\G
#    SHOW CREATE TABLE `employees`.`employees`\G
#    SHOW TABLE STATUS FROM `employees` LIKE 'salaries_ex'\G
#    SHOW CREATE TABLE `employees`.`salaries_ex`\G
# EXPLAIN /*!50100 PARTITIONS*/
SELECT SQL_NO_CACHE e.*, s.salary AS max_salary
FROM employees   e,
     salaries_ex s,
     (SELECT emp_no,MAX(from_date) AS max_from_date
      FROM salaries_ex GROUP BY emp_no
     ) s0,
     (SELECT emp_no,MAX(sa_id    ) AS max_sa_id
      FROM salaries_ex GROUP BY emp_no
     ) s1
WHERE s.emp_no = e.emp_no
  AND s.emp_no = s0.emp_no
  AND s.emp_no = s1.emp_no
  AND s.from_date = s0.max_from_date
  AND s.sa_id = s1.max_sa_id
  AND e.last_name IN ('Facello')
 LIMIT 0, 1000\G
```

SQL优化原则

- **善用索引**
  1. 避免索引列计算
  2. 选择过滤性好的索引
  3. 表连接字段创建索引
  4. 尽量使用索引覆盖
  5. ……
- **用表连接代替代替子查询**
  1. 用连接替换的案例
- **注意数据类型转换**
  1. SQL参数添加双引号
  2. 表连接字段类型一致
  3. 计算时合理利用数据类型
- **减少函数计算**
  1. 减少函数的使用，加快查询结果的返回
  2. 相同功能的函数效率对比

SQL优化案例

```
-> #QUERY-1
-> SELECT SQL_NO_CACHE e.*, s.salary AS max_salary
-> FROM employees   e,
->      salaries_ex s,
->      (SELECT emp_no,MAX(from_date) AS max_from_date
->       FROM salaries_ex GROUP BY emp_no
->      ) s0,
->      (SELECT emp_no,MAX(sa_id    ) AS max_sa_id
->       FROM salaries_ex GROUP BY emp_no
->      ) s1
-> WHERE e.emp_no = s.emp_no
->   AND s.emp_no = s0.emp_no
->   AND s.emp_no = s1.emp_no
->   AND s.from_date = s0.max_from_date
->   AND s.sa_id = s1.max_sa_id
->   AND e.last_name IN ('Facello')
-> ;
```

1

2

3

SQL优化案例

```
sys_root@localhost[employees]22:07:35>EXPLAIN
    -> SELECT SQL_NO_CACHE e.*, s.salary AS max_salary
    -> FROM employees   e,
    ->      salaries_ex s,
    ->      (SELECT emp_no,MAX(from_date) AS max_from_date
    ->       FROM salaries_ex GROUP BY emp_no
    ->      ) s0,
    ->      (SELECT emp_no,MAX(sa_id    ) AS max_sa_id
    ->       FROM salaries_ex GROUP BY emp_no
    ->      ) s1
    -> WHERE s.emp_no = e.emp_no
    ->   AND s.emp_no = s0.emp_no
    ->   AND s.emp_no = s1.emp_no
    ->   AND s.from_date = s0.max_from_date
    ->   AND s.sa_id = s1.max_sa_id
    ->   AND e.last_name IN ('Facello')
    -> ;
+------+-------------+------------+--------+---------------------+-------------+---------+-----------------------------------+---------+--------------------------+
| id   | select_type | table      | type   | possible_keys       | key         | key_len | ref                               | rows    | Extra                    |
+------+-------------+------------+--------+---------------------+-------------+---------+-----------------------------------+---------+--------------------------+
|    1 | PRIMARY     | e          | ref    | PRIMARY,idx_lastname | idx_lastname | 50      | const                             |     186 | Using index condition    |
|    1 | PRIMARY     | <derived2> | ref    | key0                | key0        | 4       | employees.e.emp_no                |      15 | Using where              |
|    1 | PRIMARY     | s          | ref    | PRIMARY,emp_no      | emp_no      | 7       | employees.e.emp_no,s0.max_from_date |       1 | Using index              |
|    1 | PRIMARY     | <derived3> | ref    | key1                | key1        | 9       | employees.e.emp_no,employees.s.sa_id |      10 |                          |
|    3 | DERIVED     | salaries_ex | index | NULL                | emp_no      | 11      | NULL                              | 2837536 | Using index              |
|    2 | DERIVED     | salaries_ex | range | NULL                | emp_no      | 4       | NULL                              |  709385 | Using index for group-by |
+------+-------------+------------+--------+---------------------+-------------+---------+-----------------------------------+---------+--------------------------+
6 rows in set (0.00 sec)
```

SQL优化案例

```
-> #QUERY-2
-> SELECT SQL_NO_CACHE e.*, s0.salary AS max_salary
-> FROM employees e,
->     (SELECT emp_no,
->      SUBSTRING_INDEX(MAX(
->      CONCAT(from_date,LPAD(sa_id, 19, '0'), ',', salary)
->      ), ',', -1) salary
->      FROM salaries_ex GROUP BY emp_no
->     ) s0
-> WHERE e.emp_no = s0.emp_no
->   AND e.last_name IN ('Facello')
-> ;
```

1

2

3

```
    -> WHERE e.emp_no = s0.emp_no
    ->   AND e.last_name IN ('Facello')
    -> ;
+------+-------------+-------------+-------+--------------------+-------------+--------+-------------------+---------+-----------------------+
| id   | select_type | table       | type  | possible_keys      | key         | key_len | ref               | rows    | Extra                 |
+------+-------------+-------------+-------+--------------------+-------------+--------+-------------------+---------+-----------------------+
|    1 | PRIMARY     | e           | ref   | PRIMARY,idx_lastname | idx_lastname | 50     | const             |     186 | Using index condition |
|    1 | PRIMARY     | <derived2>  | ref   | key0               | key0        | 4      | employees.e.emp_no |      10 |                       |
|    2 | DERIVED     | salaries_ex | index | NULL               | emp_no      | 11     | NULL              | 2837536 | Using index           |
+------+-------------+-------------+-------+--------------------+-------------+--------+-------------------+---------+-----------------------+
3 rows in set (0.00 sec)
```

SQL优化案例

T1&&T2			
PK1	Col_id	Col_2	Col_3
1	100	A	CCC
2	101	B	XXX
3	100	C	LLL
4	102	D	XXX
5	100	E	YYY
过程如下			
STEP2:	SUBSTRING_INDEX(MAX(		
STEP1:	CONCAT(Col_2,LPAD(PK1, 19, '0'), ',', Col_3)		
STEP3:	), ',', -1)		

SQL优化案例

```
-> #QUERY-3
-> SELECT SQL_NO_CACHE e.*, s1.salary AS max_salary
-> FROM employees e
-> INNER JOIN salaries_ex s1 ON   e.emp_no = s1.emp_no
-> LEFT   JOIN salaries_ex s2 ON s1.emp_no = s2.emp_no
->      AND s1.from_date < s2.from_date AND s1.sa_id < s2.sa_id
-> WHERE s2.sa_id IS NULL
->      AND s2.from_date IS NULL
->      AND e.last_name IN ('Facello')
-> ;
```

1 2 3 4

```
sys_root@localhost[employees]22:12:15>EXPLAIN
    -> SELECT SQL_NO_CACHE e.*, s1.salary AS max_salary
    -> FROM employees e
    -> INNER JOIN salaries_ex s1 ON e.emp_no = s1.emp_no
    -> LEFT  JOIN salaries_ex s2 ON s1.emp_no = s2.emp_no
    ->       AND s1.from_date < s2.from_date AND s1.sa_id < s2.sa_id
    -> WHERE s2.sa_id IS NULL
    ->   AND s2.from_date IS NULL
    ->   AND e.last_name IN ('Facello')
    -> ;
+------+-------------+-------+------+----------------------+-------------+---------+-------------------+------+-----------------------------------------+
| id   | select_type | table | type | possible_keys        | key         | key_len | ref               | rows | Extra                                   |
+------+-------------+-------+------+----------------------+-------------+---------+-------------------+------+-----------------------------------------+
|    1 | SIMPLE      | e     | ref  | PRIMARY,idx_lastname | idx_lastname | 50      | const             |  186 | Using index condition                   |
|    1 | SIMPLE      | s1    | ref  | emp_no               | emp_no      | 4       | employees.e.emp_no |    4 | Using index                             |
|    1 | SIMPLE      | s2    | ref  | PRIMARY,emp_no       | emp_no      | 4       | employees.e.emp_no |    4 | Using where; Using index; Not exists    |
+------+-------------+-------+------+----------------------+-------------+---------+-------------------+------+-----------------------------------------+
3 rows in set (0.00 sec)
```

SQL优化案例

T1&&T2			
PK1	Col_id	Col_2	Col_3
1	100	A	CCC
2	101	B	XXX
3	100	C	LLL
4	102	D	XXX
5	100	E	YYY
过程如下			
STEP1:	T1 LEFT JOIN T2 ON T1.Col_id = T2.Col_id		
STEP2:	AND T1.Pk1 < T2.PK1		
STEP3:	WHERE t2.PK1 IS NULL		

SQL优化案例

```
-> #QUERY-4
-> SELECT SQL_NO_CACHE e.*, s.salary AS max_salary
-> FROM employees e
-> INNER JOIN salaries_ex s ON e.emp_no = s.emp_no
-> WHERE s.from_date = (
->    SELECT MAX(from_date) max_from_date
->    FROM salaries_ex s1 WHERE e.emp_no = s1.emp_no
->    AND s1.sa_id = (
->       SELECT MAX(sa_id) max_sa_id
->       FROM salaries_ex s2 WHERE s.emp_no =s2.emp_no
->    )
-> ) AND e.last_name IN ('Facello')
-> ;
```

1 2 3

```
+------+--------------------+-------+--------+----------------------+--------------+---------+-------------------+------+--------------------------+
| id   | select_type        | table | type   | possible_keys        | key          | key_len | ref               | rows | Extra                    |
+------+--------------------+-------+--------+----------------------+--------------+---------+-------------------+------+--------------------------+
|    1 | PRIMARY            | e     | ref    | PRIMARY,idx_lastname | idx_lastname | 50      | const             |  186 | Using index condition    |
|    1 | PRIMARY            | s     | ref    | emp_no               | emp_no       | 4       | employees.e.emp_no |    4 | Using where; Using index |
|    2 | DEPENDENT SUBQUERY | s1    | eq_ref | PRIMARY,emp_no       | PRIMARY      | 4       | func              |    1 | Using where              |
|    3 | DEPENDENT SUBQUERY | s2    | ref    | emp_no               | emp_no       | 4       | employees.s.emp_no |    4 | Using index              |
+------+--------------------+-------+--------+----------------------+--------------+---------+-------------------+------+--------------------------+
4 rows in set (0.00 sec)
```

SQL优化案例

#QUEYR 1

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	PRIMARY	e	ref	PRIMARY,idx_lastname	idx_lastname	50	const	186	Using index condition
1	PRIMARY	<derived2>	ref	key0	key0	4	employees.e.emp_no	15	Using where
1	PRIMARY	s	ref	PRIMARY,emp_no	emp_no	7	employees.e.emp_no,s0.max_from_date	1	Using index
1	PRIMARY	<derived3>	ref	key1	key1	9	employees.e.emp_no,employees.s.sa_id	10	
3	DERIVED	salaries_ex	index	NULL	emp_no	11	NULL	2837536	Using index
2	DERIVED	salaries_ex	range	NULL	emp_no	4	NULL	709385	Using index for group-by

1

#QUEYR 2

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	PRIMARY	e	ref	PRIMARY,idx_lastname	idx_lastname	50	const	186	Using index condition
1	PRIMARY	<derived2>	ref	key0	key0	4	employees.e.emp_no	10	
2	DERIVED	salaries_ex	index	NULL	emp_no	11	NULL	2837536	Using index

2

#QUEYR 3

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	e	ref	PRIMARY,idx_lastname	idx_lastname	50	const	186	Using index condition
1	SIMPLE	s1	ref	emp_no	emp_no	4	employees.e.emp_no	4	Using index
1	SIMPLE	s2	ref	PRIMARY,emp_no	emp_no	4	employees.e.emp_no	4	Using where; Using index; Not exists

3

5

#QUEYR 4

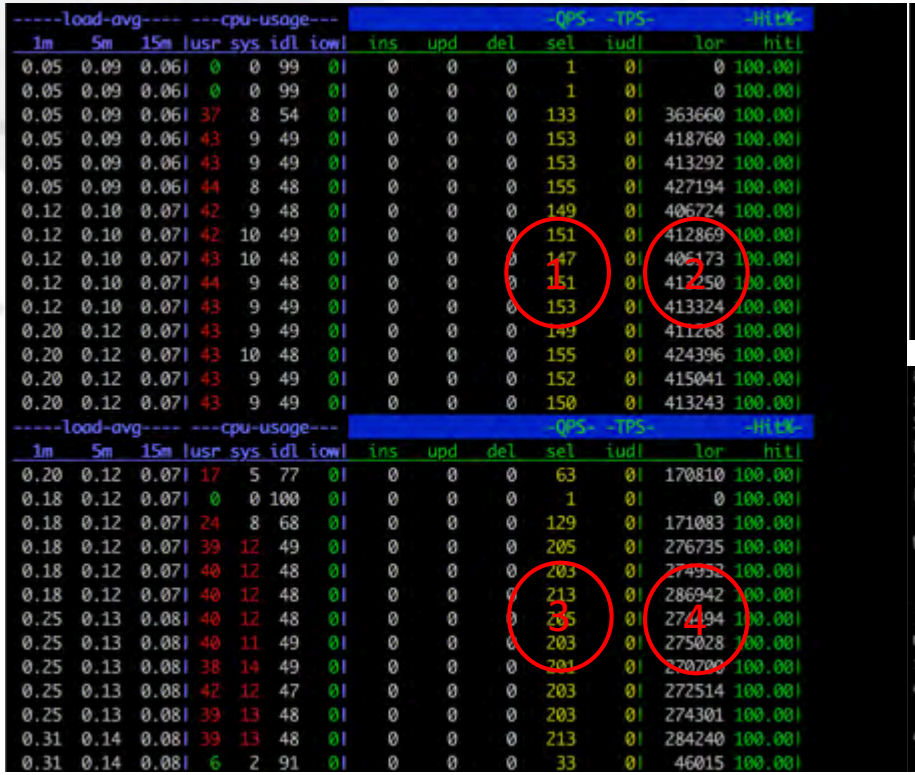
```
    -> ;
```

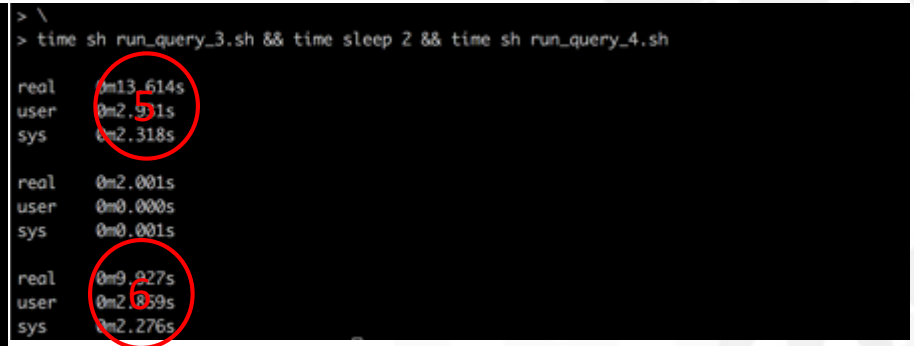
id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	PRIMARY	e	ref	PRIMARY,idx_lastname	idx_lastname	50	const	186	Using index condition
1	PRIMARY	s	ref	emp_no	emp_no	4	employees.e.emp_no	4	Using where; Using index
2	DEPENDENT SUBQUERY	s1	eq_ref	PRIMARY,emp_no	PRIMARY	4	func	1	Using where
3	DEPENDENT SUBQUERY	s2	ref	emp_no	emp_no	4	employees.s.emp_no	4	Using index

4

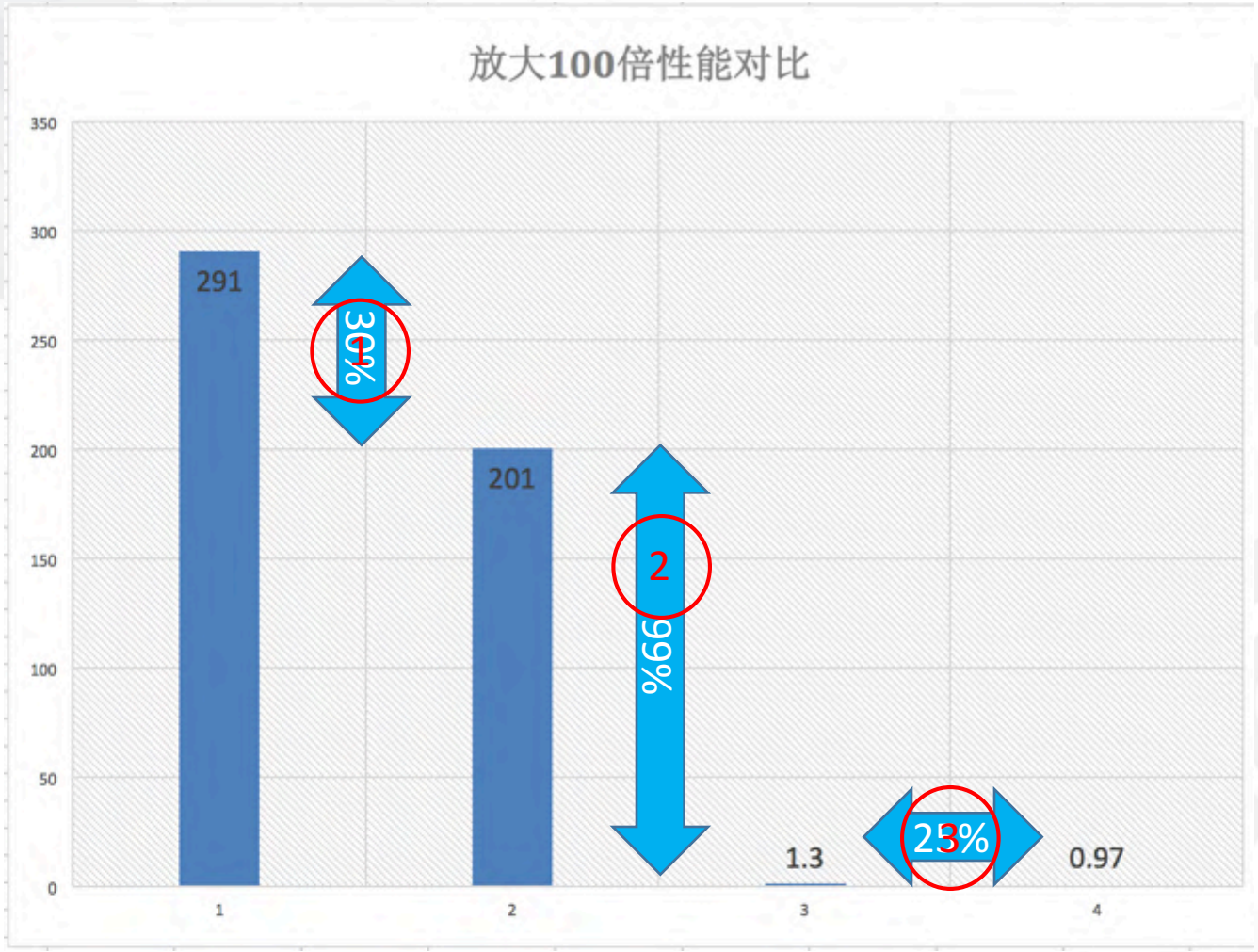
SQL优化案例

#把QUERY3和QUERY4放大1000倍，性能对比



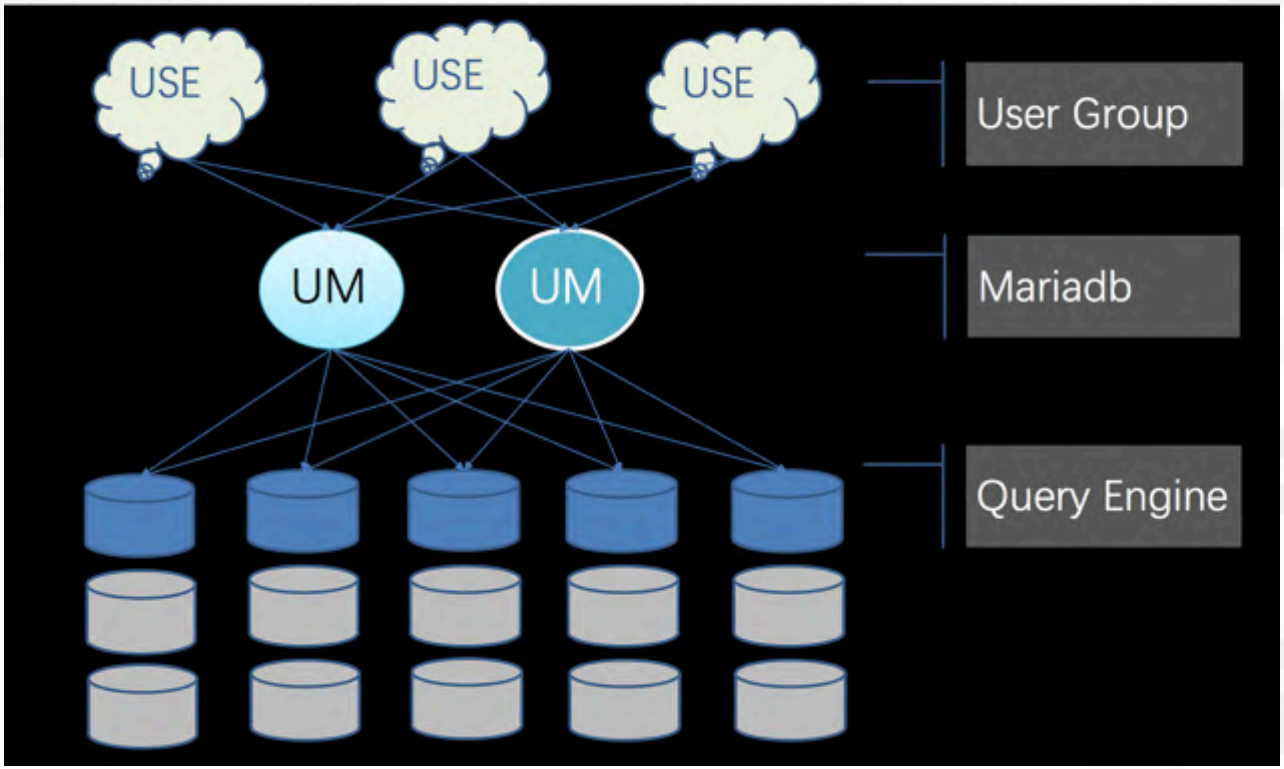


SQL优化案例


放大100倍性能对比
350
300
250
200
150
100
50
0
291
30%
201
2
99%
1.3
23%
0.97
1
2
3
4

MariaDB ColumnStore

全量数据查询统计怎么办？



MariaDB ColumnStore

同一颜色数据，位置相邻

ROW-oriented

id	Lname	Gender	ctime	utime
1	Li	0	20160808	20160809
2	Wang	1	20160808	20160809
3	Song	1	20160808	20160809
4	He	0	20160808	20160809
5	Zhang	1	20160808	20160809

Column-oriented

id	Lname	Gender	ctime	utime
1	Li	0	20160808	20160809
2	Wang	1	20160808	20160809
3	Song	1	20160808	20160809
4	He	0	20160808	20160809
5	Zhang	1	20160808	20160809

MariaDB ColumnStore

```
MariaDB [dtcc]> select count(*) from salaries_cstore;
+-----------+
| count(*)  |
+-----------+
| 100663296 |
+-----------+
1 row in set (0.65 sec)

MariaDB [dtcc]> update salaries_cstore set salary = salary + 1;
Query OK, 100663296 rows affected (3 min 21.76 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```

```
MariaDB [dtcc]> select left(from_date,1),count(*) from salaries_cstore\
    -> group by left(from_date,1);
+-------------------+----------+
| left(from_date,1) | count(*) |
+-------------------+----------+
| 1                 | 36825276 |
| 2                 | 63838020 |
+-------------------+----------+
2 rows in set (15.49 sec)
```

```
MariaDB [dtcc]> delete from salaries_cstore where sa_id > 34078569 limit 30000000;
Query OK, 30000000 rows affected, 1 warning (1 min 17.19 sec)

MariaDB [dtcc]> delete from salaries_cstore where sa_id > 34078569 limit 30000000;
Query OK, 30000000 rows affected, 1 warning (1 min 13.34 sec)

MariaDB [dtcc]> delete from salaries_cstore where sa_id > 34078569 limit 30000000;
Query OK, 7108864 rows affected, 1 warning (15.31 sec)

MariaDB [dtcc]> delete from salaries_cstore where sa_id > 34078569 limit 30000000;
Query OK, 0 rows affected, 1 warning (0.13 sec)
```

```
MariaDB [dtcc]> INSERT INTO salaries_cstore SELECT 102235708,\
    -> FLOOR(RAND()*1000000),FLOOR(RAND()*100000),\
    -> @i:=CURDATE()-INTERVAL FLOOR(RAND()*10000) DAY, \
    -> @i+INTERVAL FLOOR(RAND()*1000) DAY;
Query OK, 1 row affected, 1 warning (1.16 sec)
Records: 1  Duplicates: 0  Warnings: 1
```

批量速度较快，单条很慢
一次删除条数有限制

```
MariaDB [dtcc]> UPDATE salaries_cstore SET salary = 1000000, \
    -> emp_no = 100 WHERE sa_id = 10000000;
Query OK, 1 row affected (0.19 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [dtcc]> select * from salaries_cstore where sa_id = 10000000;
+----------+--------+---------+------------+------------+
| sa_id    | emp_no | salary  | from_date  | to_date    |
+----------+--------+---------+------------+------------+
| 10000000 |    100 | 1000000 | 2016-12-04 | 2018-06-30 |
+----------+--------+---------+------------+------------+
1 row in set (0.07 sec)

MariaDB [dtcc]> delete from salaries_cstore where sa_id = 10000000;
Query OK, 1 row affected (0.28 sec)

MariaDB [dtcc]> select * from salaries_cstore where sa_id = 10000000;
Empty set (0.08 sec)

MariaDB [dtcc]>
```

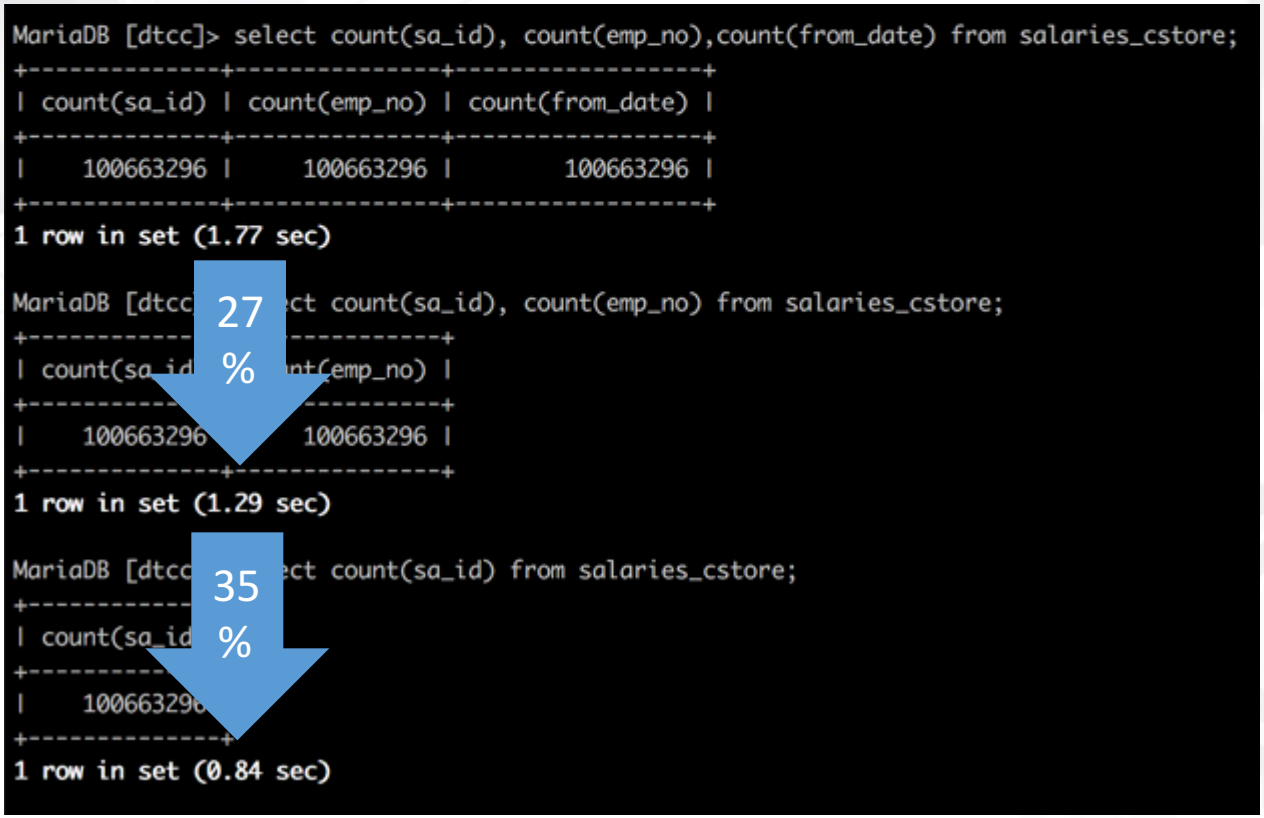
MariaDB ColumnStore

CStore表中比InnoDB表数据量多2倍

在全量统计下，性能依然飞升数倍

MariaDB ColumnStore

涉及的字段越少，性能越快



MariaDB ColumnStore

ColumnStore有自己的执行路径解析

```
MariaDB [dtcc]> EXPLAIN SELECT DATE_FORMAT(from_date,'%Y') years, COUNT(*) FROM salaries_cstore GROUP BY DATE_FORMAT(from_date,'%Y');
+------+-------------+-----------------+------+---------------+------+---------+------+------+---------------------------------+
| id   | select_type | table           | type | possible_keys | key  | key_len | ref  | rows | Extra                           |
+------+-------------+-----------------+------+---------------+------+---------+------+------+---------------------------------+
|    1 | SIMPLE      | salaries_cstore | ALL  | NULL          | NULL | NULL    | NULL | 2000 | Using temporary; Using filesort |
+------+-------------+-----------------+------+---------------+------+---------+------+------+---------------------------------+
MariaDB [dtcc]>
MariaDB [dtcc]> SELECT DATE_FORMAT(from_date,'%Y') years, COUNT(*) FROM salaries_cstore GROUP BY DATE_FORMAT(from_date,'%Y') LIMIT 1;
+-------+----------+
| years | COUNT(*) |
+-------+----------+
| 2000  |  3685713 |
+-------+----------+
1 row in set (7.91 sec)

MariaDB [dtcc]> SELECT LEFT(from_date, 4) years, COUNT(*) FROM salaries_cstore GROUP BY LEFT(from_date,4) LIMIT 1;
+-------+----------+
| years | COUNT(*) |
+-------+----------+
| 2000  |  3685713 |
+-------+----------+
1 row in set (16.03 sec)

MariaDB [dtcc]> SELECT YEAR(from_date) years, COUNT(*) FROM salaries_cstore GROUP BY YEAR(from_date) LIMIT 1;
+-------+----------+
| years | COUNT(*) |
+-------+----------+
|  2017 |  1327811 |
+-------+----------+
1 row in set (2.73 sec)
```

1

2

3

欢迎讨论你知道的SQL优化

更多SQL优化及ColumnStore实践，请联系我
陈兴隆
15811381131

THANKS

SequeMedia 盛拓传媒 ITPUB ChinaUnix