



2016 杭州·云栖大会
THE COMPUTING CONFERENCE

云栖社区
yq.aliyun.com

AliSQL 内核定制方案

2016
The Computing Conference

主办单位:



战略合作伙伴:



襄洛
阿里云资深开发工程师



扫码观看大会视频

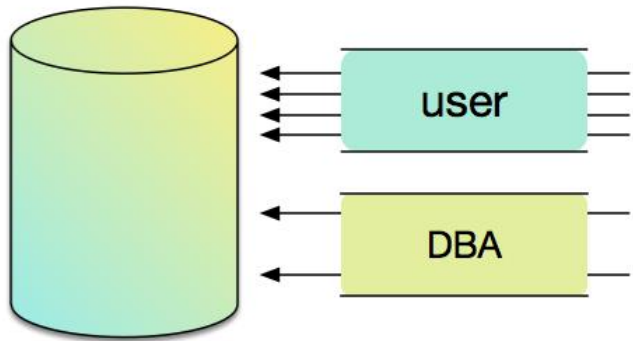
- 一、并发控制
- 二、诊断信息
- 三、语法定制
- 四、备库延迟



一、并发控制



1、连接分组

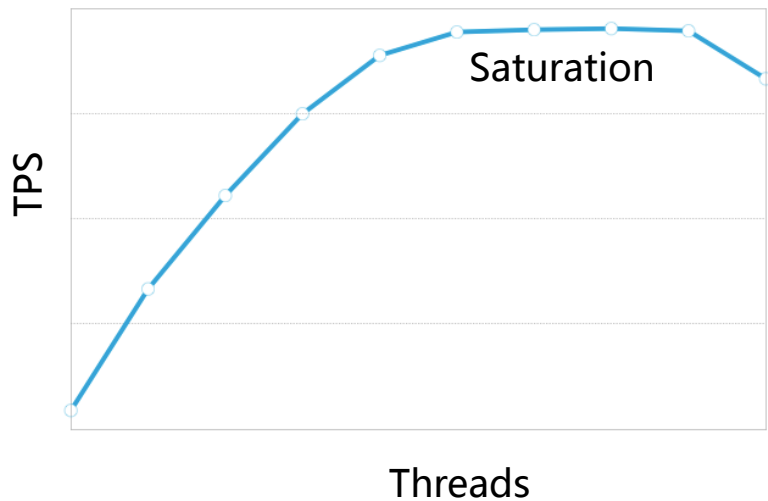


根据用户角色，对其连接进行分组隔离。

- 用户连接资源不会被后台系统占用
- 后台系统不受用户使用影响



2、高水位限流



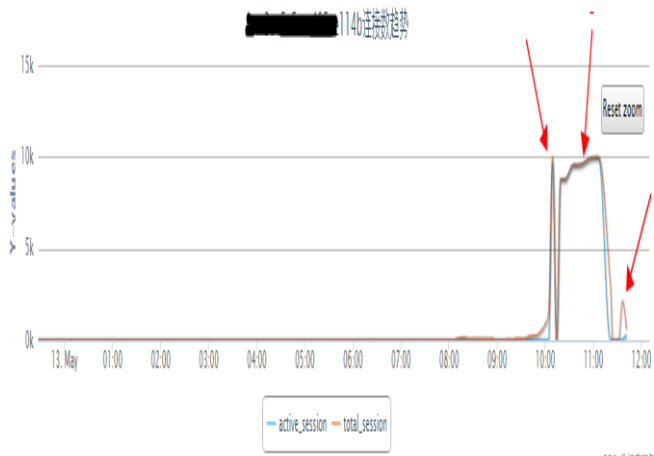
DB 负载过高时，拒绝部分流量，进行降级保护，保证吞吐能力。

1.rds_threads_running_high_watermark: [N]

2.rds_threads_running_ctl_mode: [SELECTS | ALL]



2、高水位限流 (Cont.)



ins	upd	del	sel	iud	lor	hit	run	con	cre
0	0	0	0	0	0	100.00	0	0	0
6	1	0	16998	7	6	100.00	2665	4208	0
208	-14	-23	255516	171	5752	99.91	2659	4148	234
24	3	-7	1314	20	7198	100.00	2645	4274	126
26	-3	-3	754	20	316	100.00	2644	4399	125
19	2	0	385	21	2321	99.87	2643	4476	77
62	211	77	36062	1350	109602	100.00	4433	4673	196
19	188	28	230720	235	3685	100.00	2602	4678	53
35	13	-3	518068	45	1309	100.00	2595	4215	127
33	-1	-3	22913	29	69	100.00	2590	4113	0
44	0	0	698	44	241	100.00	2584	4351	185
58	16	6	30677	80	2247	100.00	3984	4885	535
167	32	7	1377057	206	3653	100.00	2603	4087	122
40	62	1	36782	103	7233	100.00	2584	4117	81
31	27	6	38132	64	3403	100.00	2600	4112	48

ins	upd	del	sel	iud	lor	hit	run	con	cre	cac
0	0	0	0	0	0	100.00	0	0	0	0
415	183	519	111212	1117	641424	100.00	299	2387	27	99
589	104	702	135282	1395	536131	100.00	1248	2353	5	93
449	54	449	100077	952	370036	100.00	302	2267	18	11
653	173	812	58709	1638	397282	100.00	703	2034	0	35
539	83	599	121055	1221	148946	100.00	727	1880	0	99
674	88	767	93719	1529	311084	100.00	555	642	0	99
1188	209	1337	67006	2734	311836	100.00	304	355	0	99
813	199	991	26069	2003	224129	100.00	304	401	62	56
216	820	966	12757	2002	886850	100.00	36	184	0	20
0	87	94	18234	181	1151678	100.00	304	670	467	0
496	109	591	96988	1196	386145	100.00	456	582	6	6
877	301	1182	33187	2360	373603	100.00	300	488	0	14



3、SQL 语句并发控制

VARIABLE_NAME	VARIABLE_VALUE
rds_sql_update_filter	
rds_sql_select_filter	
rds_sql_delete_filter	
rds_reset_all_filter	OFF
rds_filter_key_cmp_in_order	OFF

控制特定模式的 SQL 语句的并发度。

- SET GLOBAL rds_sql_select_filter = '+,{CONC},KEY1,KEY2...';
- SET GLOBAL rds_sql_select_filter = '-,id1,id2...'
- SELECT * FROM INFORMATION_SCHEMA.SQL_FILTER_INFO;
- SET GLOBAL rds_filter_key_cmp_in_order = 1;
- SHOW SQL_FILTERS;
- SET GLOBAL rds_reset_all_filter = 1;



3、SQL 语句并发控制 (Cont.)

```
mysql> SET GLOBAL rds_sql_select_filter = '+,5,a=1,t1';
```

```
mysql> SET GLOBAL rds_sql_select_filter = '+,5,a=1,t2';
```

```
mysql> SHOW SQL_FILTERS;
```

type	item_id	cur_conc	max_conc	key_num	key_str
SELECT	6	0	5	1	+,5,a=1,t2
SELECT	5	0	5	1	+,5,a=1,t1

添加两条规则

```
mysql> SET GLOBAL rds_sql_select_filter = '-,6';
```

```
mysql> SHOW SQL_FILTERS;
```

type	item_id	cur_conc	max_conc	key_num	key_str
SELECT	5	0	5	1	+,5,a=1,t1

去掉一条规则





二、诊断信息



1、表和索引的统计

```
mysql> SELECT * FROM INFORMATION_SCHEMA.TABLE_STATISTICS;
```

TABLE_SCHEMA	TABLE_NAME	ROWS_READ	ROWS_CHANGED	ROWS_CHANGED_X_INDEXES	ROWS_INSERTED	ROWS_DELETED	ROWS_UPDATED
test	sbtest1	1076141	17846	35692	2536	2536	12774

```
mysql> SELECT * FROM INFORMATION_SCHEMA.INDEX_STATISTICS;
```

TABLE_SCHEMA	TABLE_NAME	INDEX_NAME	ROWS_READ
test	sbtest1	k_1	66
test	sbtest1	PRIMARY	1076075

Table 统计：核心业务的读写比例如何，后期的扩展性是读写分离，还是水平拆分哪个更适合。

Index 统计：索引使用的频率和效率，哪些索引根本没有使用，可以 DROP 掉。



2、SQL 内存/IO/临时空间

```
Id: 39
User: xiangluo
Host: 30.40.13.11:56566
db: test
Command: Query
Time: 0
State: updating
Info: UPDATE sbtest1 SET c='64862974741'
Memory_used: 2175648
Memory_used_by_query: 8248
Logical_read: 12246
Physical_sync_read: 0
Physical_async_read: 0
Temp_user_table_size: 0
Temp_sort_table_size: 0
Temp_sort_file_size: 0
```

内存统计：

- Memory_used
- Memory_used_by_query

IO统计：

- Logical_read
- Physical_sync_read
- Physical_async_read

临时空间统计

- Temp_user_table_size
- Temp_sort_table_size
- Temp_sort_file_size



3、审计日志

[SQL日志明细](#)
[SQL日志报告](#)
[SQL注入检测](#)
[慢SQL统计\(按天\)](#)
[慢SQL明细](#)
[主键统计](#)
[索引统计](#)
[存储引擎](#)
[大表优化](#)
[缺失索引](#)
[其它](#)

日期时间: - 关键字:

SQL语句	执行时间	数据库	用户名	客户端IP	消耗时间(微秒)	返回记录
SET NAMES 'utf8'	2016-09-29 16:51:26	wwb	wwb	10.47.160.210	32	0

记录每一条 SQL 语句的执行和使用的资源。

- host, user, db, tid, SQL
- 事务/语句开始时间、执行时间 (RT)、等锁时间

- MEM / IO / 临时表(文件)
- error_code



4、IO Latency

```
mysql> SELECT * FROM INFORMATION_SCHEMA.INNODB_IO_STATUS;
```

type	request	max_time
Normal aio read	5	4111
Normal aio write	0	0
Ibuf aio read	0	0
Log aio	0	0
Sync aio	0	0

```
InnoDB: [RDS Diagnose] Write len: 16384, latency: 47us.
InnoDB: [RDS Diagnose] Write len: 163328, latency: 108us.
InnoDB: [RDS Diagnose] Log flush len: 163328, latency: 5594us.
Binlog: [RDS Diagnose] Log flush len: 135322, latency: 6443us.
Semisync: [RDS Diagnose] Wait pos: 272979, latency: 2419us.
InnoDB: [RDS Diagnose] Write len: 16384, latency: 60us.
InnoDB: [RDS Diagnose] Write len: 16384, latency: 48us.
InnoDB: [RDS Diagnose] Write len: 16384, latency: 712us.
InnoDB: [RDS Diagnose] AIO write len: 16384, latency: 47us.
InnoDB: [RDS Diagnose] AIO write len: 16384, latency: 63us.
```

I_S 表和 error-log 中记录 IO 延迟信息，方便判断磁盘 IO 问题。





三、语法定制



1

SET RDS_RESETCONNECTION=1

2

**SELECT ... FOR UPDATE [WAIT [n]|NO_WAIT]
LOCK TABLE ... [WAIT [n]|NO_WAIT]**

3

ALTER TABLE t [WAIT [n]|NO_WAIT]



4

SELECT FROM UPDATE

5

UPDATE ... ON NON_AFFECTED ROWS INSERT VALUES ...

6

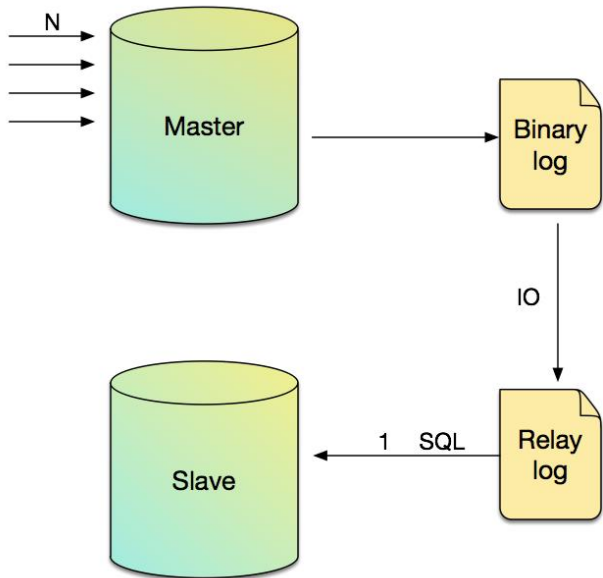
SET rds_sql_max_affected_rows=n



四、备库延迟



1、主备复制架构



主备线程 N : 1

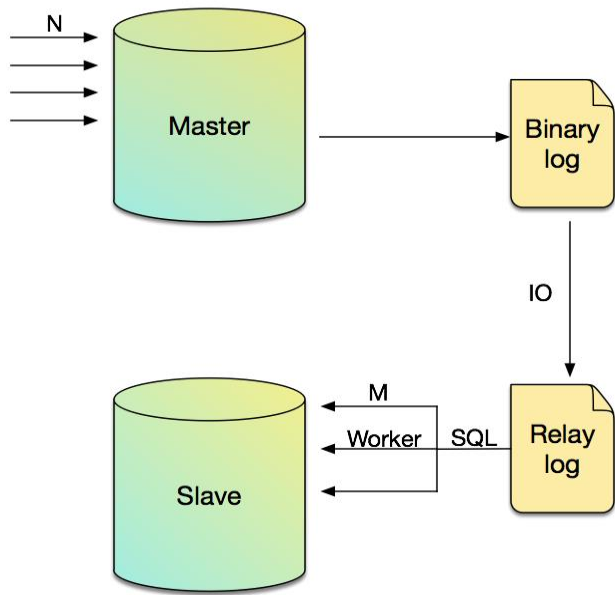
备库放开双 1 配置

`innodb_flush_log_at_trx_commit`

`sync_binlog`



2、并行复制



主备线程 $N : M$

`slave_parallel_workers`

`slave_pr_mode [SCHEMA|TABLE|TRX]`



3、隐含主键

```
mysql> SHOW CREATE TABLE t1\G
***** 1. row *****
      Table: t1
Create Table: CREATE TABLE `t1` (
  `id` int(11) DEFAULT NULL,
  `name` varchar(30) DEFAULT NULL,
  `__#alibaba_rds_row_id#` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'Implicit Primary Key by RDS',
  KEY `__#alibaba_rds_row_id#` (`__#alibaba_rds_row_id#`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```



2016 The
Computing
Conference
THANKS

