数据风云 十年变迁

DTCC 第十届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2019

IT168.com ChinaUnix ITPUB

2019 — 05 — 08-10

北京新云南皇冠假日酒店

MySQL全量SQL分析与审计平台

王栋

内容

背景介绍

日常运维中会碰到一些问题：

- CPU使用率瞬间升高
- MySQL并发线程数瞬间升高
- QPS瞬间飙升
- 慢查询突然增多
- 网卡流量突然增大
- 磁盘的IOPS突然升高
- …

排障方法

- 查看性能监控
- show processlist
- show engine innodb status
- pt-stalk
- …

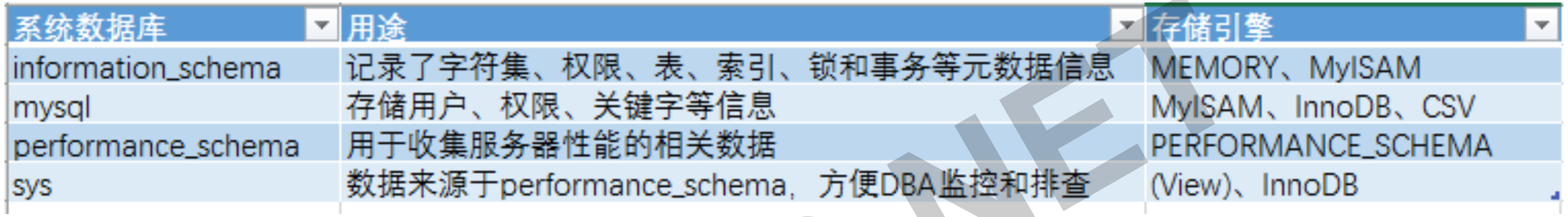
持续时间短，很难抓到现场

数据库分钟级别的监控已不能满足我们的要求

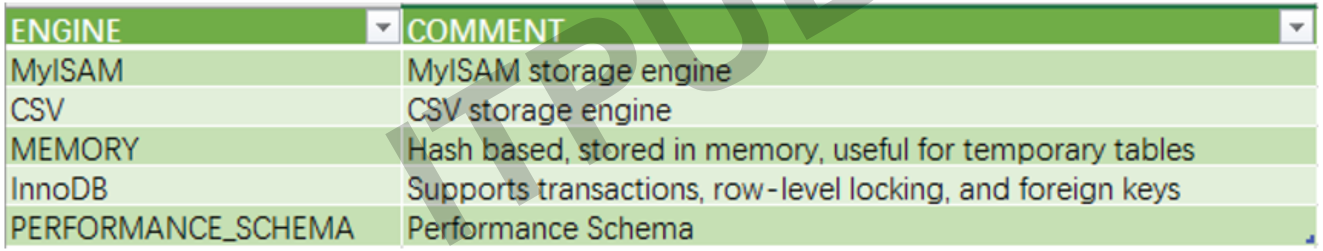
方案选型

- Tcpdump
- Proxy
- Performance schema
- …

系统数据库



系统数据库	用途	存储引擎
information_schema	记录了字符集、权限、表、索引、锁和事务等元数据信息	MEMORY、MyISAM
mysql	存储用户、权限、关键字等信息	MyISAM、InnoDB、CSV
performance_schema	用于收集服务器性能的相关数据	PERFORMANCE_SCHEMA
sys	数据来源于performance_schema，方便DBA监控和排查	(View)、InnoDB



ENGINE	COMMENT
MyISAM	MyISAM storage engine
CSV	CSV storage engine
MEMORY	Hash based, stored in memory, useful for temporary tables
InnoDB	Supports transactions, row-level locking, and foreign keys
PERFORMANCE_SCHEMA	Performance Schema

MySQL 5.7 Performance Schema相关表

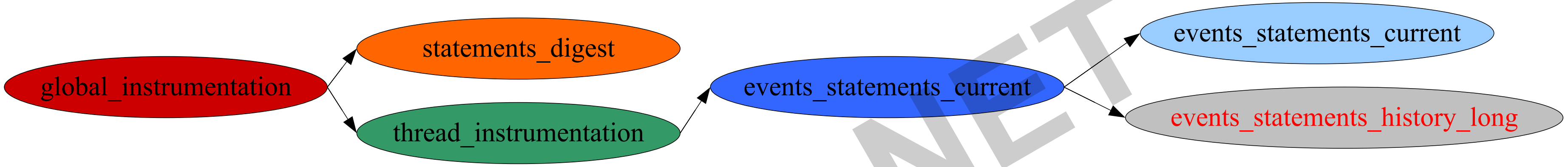
Instrumentation Setup

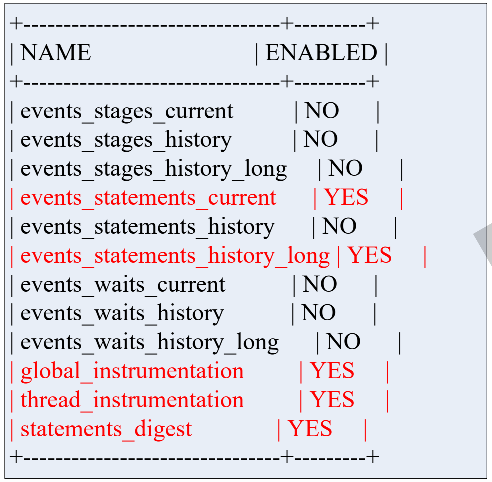
- setup_instruments
- setup_consumers
- setup_actors
- setup_objects
- setup_timers

Statements

- events_statements_current
- events_statements_history
- events_statements_history_long
- events_statements_summary_by_digest
- events_statements_summary_by_program
- events_statements_summary_global_by_event_name
- events_statements_summary_by_thread_by_event_name
- events_statements_summary_by_account_by_event_name
- events_statements_summary_by_host_by_event_name
- events_statements_summary_by_user_by_event_name

setup_consumers 层次图


global_instrumentation
statements_digest
thread_instrumentation
events_statements_current
events_statements_current
events_statements_history_long



```
+--------------------------------+---------+
| NAME                           | ENABLED |
+--------------------------------+---------+
| events_stages_current          | NO      |
| events_stages_history          | NO      |
| events_stages_history_long     | NO      |
| events_statements_current      | YES     |
| events_statements_history      | NO      |
| events_statements_history_long | YES     |
| events_waits_current           | NO      |
| events_waits_history           | NO      |
| events_waits_history_long      | NO      |
| global_instrumentation         | YES     |
| thread_instrumentation         | YES     |
| statements_digest              | YES     |
+--------------------------------+---------+
```

开启 performance_schema

- mysql5.6.6 以上版本默认开启，my.cnf中配置

```
[mysqld]
performance_schema=ON
```

- UPDATE setup_consumers SET ENABLED='YES' WHERE NAME='events_statements_history_long';

- UPDATE setup_instruments SET enabled='NO',TIMED='NO' WHERE NAME IN ('statement/com/InitDB','statement/com/Ping','statement/com/Quit','statement/sql/commit','statement/com/Prepare','statement/sql/show_warnings');

```
+----------------------------+---------+-------+
| NAME                       | ENABLED | TIMED |
+----------------------------+---------+-------+
| statement/sql/select       | YES     | YES   |
| statement/sql/alter_table  | YES     | YES   |
| statement/sql/update       | YES     | YES   |
| statement/sql/insert       | YES     | YES   |
| statement/sql/delete       | YES     | YES   |
| statement/sql/truncate     | YES     | YES   |
| statement/sql/drop_table   | YES     | YES   |
+----------------------------+---------+-------+
```

DTCC 2019 第十届中国数据库技术大会 DATABASE TECHNOLOGY CONFERENCE CHINA 2019

events_statements_history_long表

```
CREATE TABLE `events_statements_history_long` (
  `THREAD_ID` bigint(20) unsigned NOT NULL,
  `EVENT_ID` bigint(20) unsigned NOT NULL,
  `END_EVENT_ID` bigint(20) unsigned DEFAULT NULL,
  `EVENT_NAME` varchar(128) NOT NULL,
  `SOURCE` varchar(64) DEFAULT NULL,
  `TIMER_START` bigint(20) unsigned DEFAULT NULL,
  `TIMER_END` bigint(20) unsigned DEFAULT NULL,
  `TIMER_WAIT` bigint(20) unsigned DEFAULT NULL,
  `LOCK_TIME` bigint(20) unsigned NOT NULL,
  `SQL_TEXT` longtext,
  `DIGEST` varchar(32) DEFAULT NULL,
  `DIGEST_TEXT` longtext,
  `CURRENT_SCHEMA` varchar(64) DEFAULT NULL,
  `OBJECT_TYPE` varchar(64) DEFAULT NULL,
  `OBJECT_SCHEMA` varchar(64) DEFAULT NULL,
  `OBJECT_NAME` varchar(64) DEFAULT NULL,
  `OBJECT_INSTANCE_BEGIN` bigint(20) unsigned DEFAULT NULL,
  `MYSQL_ERRNO` int(11) DEFAULT NULL,
  `RETURNED_SQLSTATE` varchar(5) DEFAULT NULL,
  `MESSAGE_TEXT` varchar(128) DEFAULT NULL,
  `ERRORS` bigint(20) unsigned NOT NULL,
  `WARNINGS` bigint(20) unsigned NOT NULL,
  `ROWS_AFFECTED` bigint(20) unsigned NOT NULL,
  `ROWS_SENT` bigint(20) unsigned NOT NULL,
  `ROWS_EXAMINED` bigint(20) unsigned NOT NULL,
  `CREATED_TMP_DISK_TABLES` bigint(20) unsigned NOT NULL,
  `CREATED_TMP_TABLES` bigint(20) unsigned NOT NULL,
  `SELECT_FULL_JOIN` bigint(20) unsigned NOT NULL,
  `SELECT_FULL_RANGE_JOIN` bigint(20) unsigned NOT NULL,
  `SELECT_RANGE` bigint(20) unsigned NOT NULL,
  `SELECT_RANGE_CHECK` bigint(20) unsigned NOT NULL,
  `SELECT_SCAN` bigint(20) unsigned NOT NULL,
  `SORT_MERGE_PASSES` bigint(20) unsigned NOT NULL,
  `SORT_RANGE` bigint(20) unsigned NOT NULL,
  `SORT_ROWS` bigint(20) unsigned NOT NULL,
  `SORT_SCAN` bigint(20) unsigned NOT NULL,
  `NO_INDEX_USED` bigint(20) unsigned NOT NULL,
  `NO_GOOD_INDEX_USED` bigint(20) unsigned NOT NULL,
  `NESTING_EVENT_ID` bigint(20) unsigned DEFAULT NULL,
  `NESTING_EVENT_TYPE` enum('TRANSACTION','STATEMENT','STAGE','WAIT') DEFAULT NULL,
  `NESTING_EVENT_LEVEL` int(11) DEFAULT NULL,
  `CLIENT_USER` varchar(128) DEFAULT NULL,
  `CLIENT_HOST` varchar(128) DEFAULT NULL,
  `RU_UTIME` bigint(20) unsigned DEFAULT NULL,
  `RU_STIME` bigint(20) unsigned DEFAULT NULL,
  `LOGIC_READ` bigint(20) unsigned NOT NULL,
  `PHYSIC_READ` bigint(20) unsigned NOT NULL,
  `PAGE_WRITE` bigint(20) unsigned NOT NULL
) ENGINE=PERFORMANCE_SCHEMA DEFAULT CHARSET=utf8
```

关键字段

字段名	描述
TIMER_START	事件开始时间(picoseconds)
TIMER_END	事件结束时间(picoseconds)
TIMER_WAIT	语句执行持续时间(picoseconds)
LOCK_TIME	等待表锁的时间(picoseconds)
SQL_TEXT	SQL语句
DIGEST_TEXT	标准化SQL语句
DIGEST	标准化SQL语句的MD5 HASH值
ROWS_AFFECTED	SQL执行的影响行数
ROWS_SENT	SQL执行返回的行数
ROWS_EXAMINED	SQL执行从存储引擎读取的行数
CLIENT_USER	访问用户
CLIENT_HOST	访问来源
RU_UTIME	用户态CPU使用时间(microseconds)
RU_STIME	系统态CPU使用时间(microseconds)
LOGIC_READ	逻辑读次数
PHYSIC_READ	物理读次数

```
       SQL_TEXT: SELECT * FROM testdb.sbtest1 where k>1648952 ORDER BY rand() LIMIT 10
         DIGEST: 6ea1437fc7c276d66f7a98d06f938527
    DIGEST_TEXT: SELECT * FROM `testdb` . `sbtest1` WHERE `k` > ? ORDER BY `rand` ( ) LIMIT ?
     EVENT_NAME: statement/sql/select
     START_TIME: 2019-04-22 15:12:15.040178
       END_TIME: 2019-04-22 15:12:15.615435
  TIMER_WAIT_MS: 575.3
CURRENT_SCHEMA: testdb
  ROWS_AFFECTED: 0
      ROWS_SENT: 10
 ROWS_EXAMINED: 417278
```

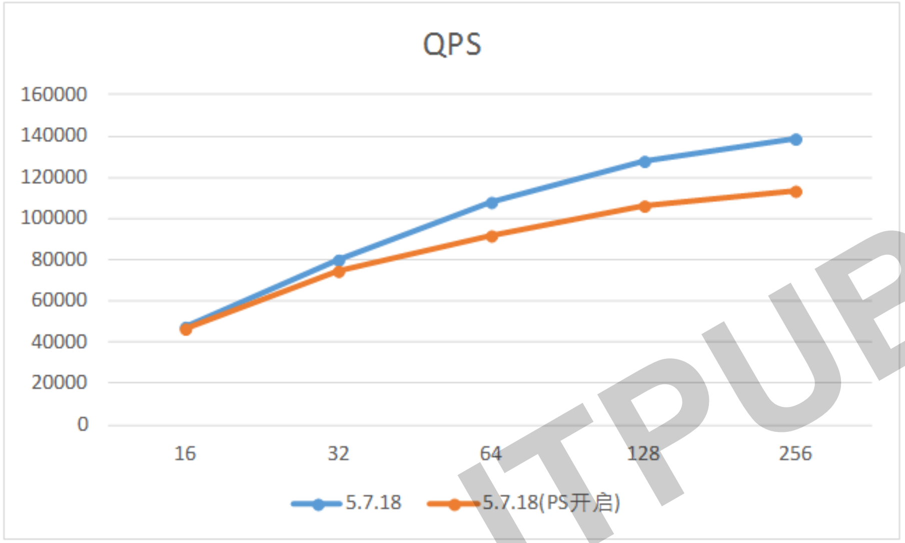
MySQL源码修改

- 在events_statements_history_long的基础上增加

  语句的访问来源CLIENT_USER和CLIENT_HOST
  精确的度量CPU消耗RU_UTIME和RU_STIME
  精确的度量IO消耗LOGIC_READ和PHYSIC_READ

- 将导出表数据的方式改为动态设置系统变量刷新数据文件

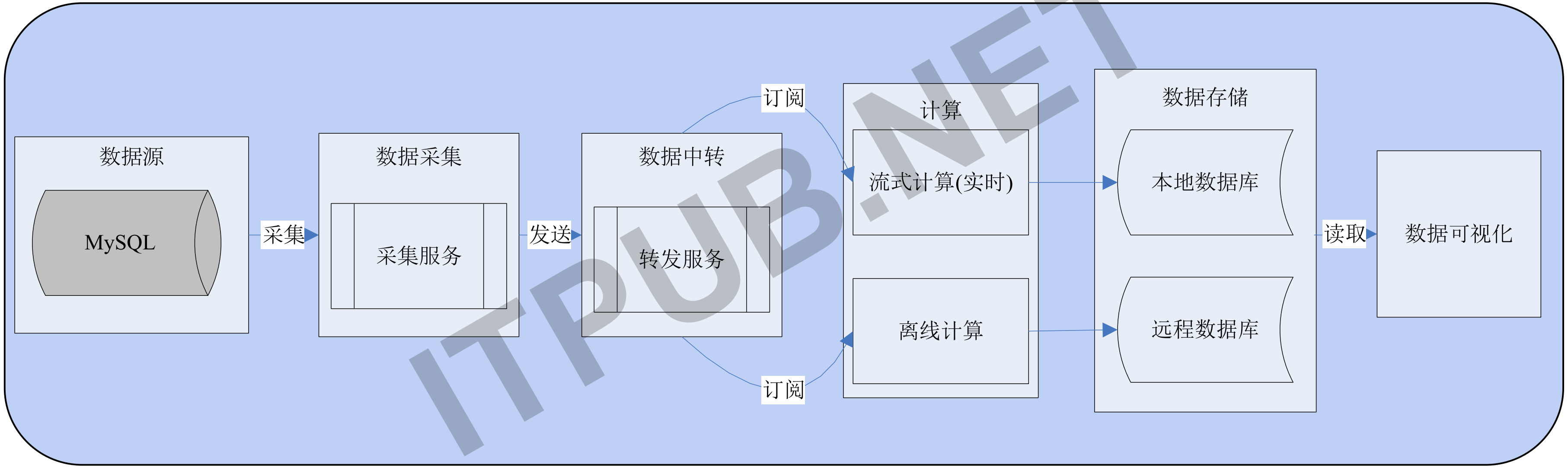
开启PS后的性能影响



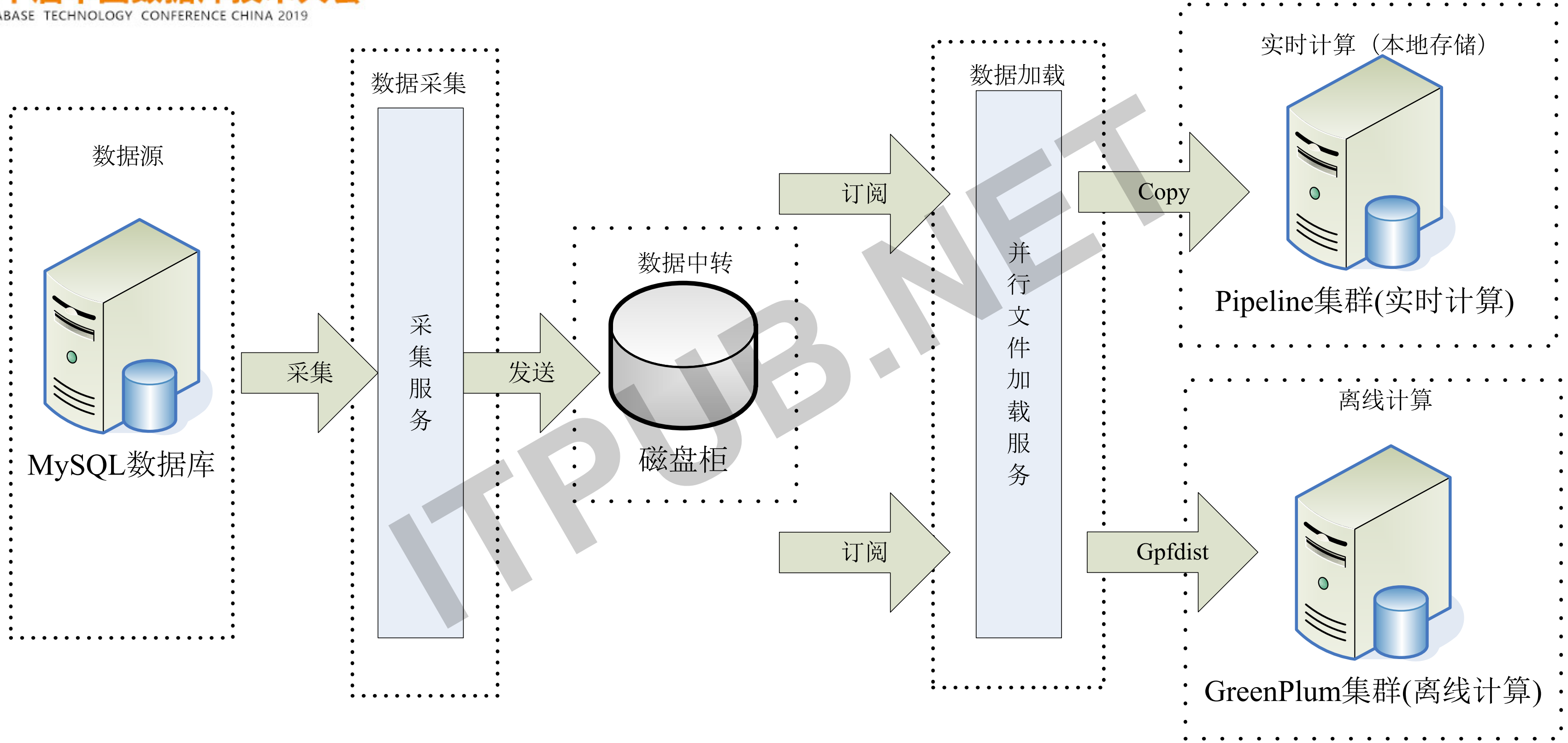
混合读写			
Thread/QPS	5.7.18	5.7.18(PS开启)	PS开启后性能下降(%)
16	47760	46791	2.03%
32	80401	74750	7.03%
64	107810	92121	14.55%
128	127944	106624	16.66%
256	138589	113614	18.02%

开启performance_schema后，随着QPS访问量的增加，对MySQL处理性能的影响逐渐增加。当QPS小于4万时（符合我们大部分的生产环境）开启PS带来的实际性能影响不大，约2%左右

设计思路


数据源
MySQL
采集
数据采集
采集服务
发送
数据中转
转发服务
订阅
订阅
计算
流式计算(实时)
离线计算
数据存储
本地数据库
远程数据库
读取
数据可视化

系统架构

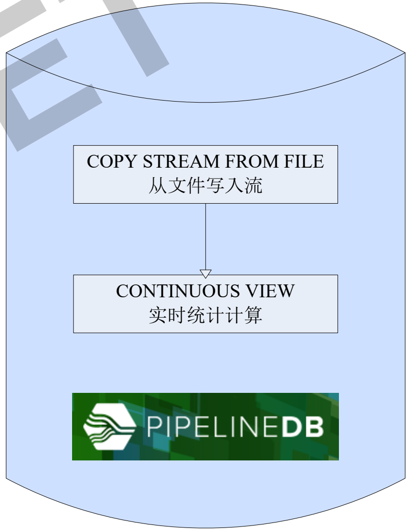

数据源
MySQL数据库
采集
数据采集
采集服务
发送
数据中转
磁盘柜
订阅
订阅
数据加载
并行文件加载服务
Copy
Gpfdist
实时计算（本地存储）
Pipeline集群(实时计算)
离线计算
GreenPlum集群(离线计算)

实时计算

基于PostgreSQL的流式数据库

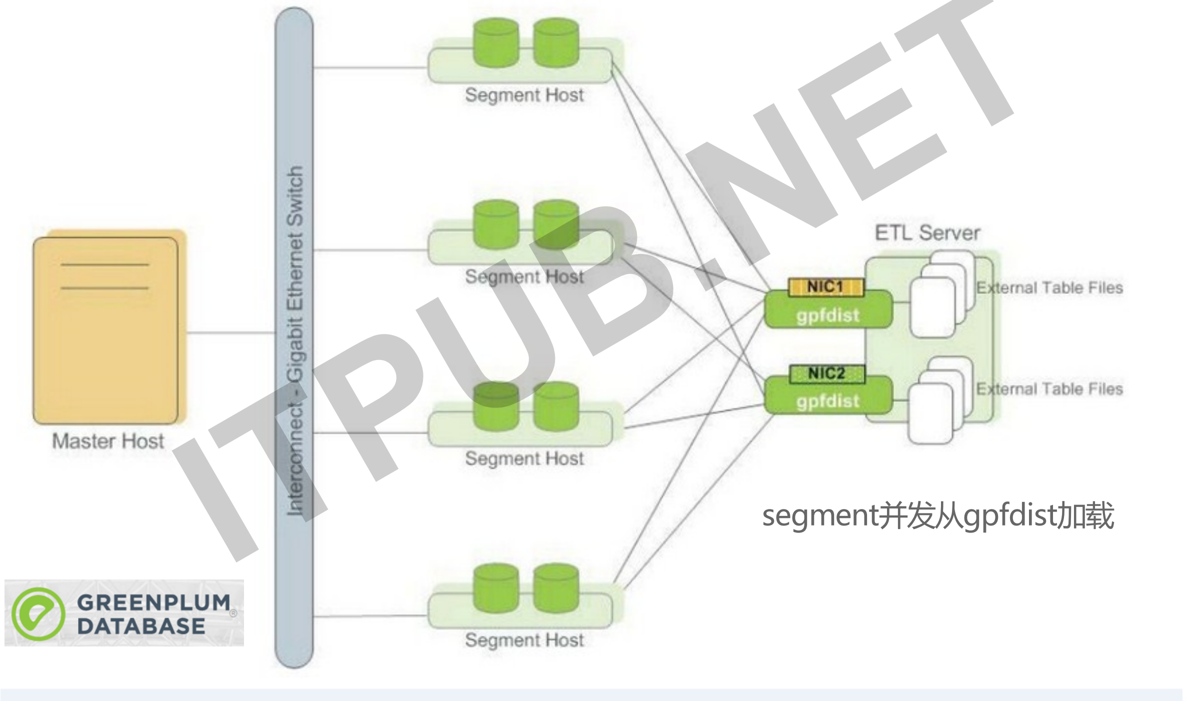
Continuous Aggregations

Continuously aggregate, filter, and distill streaming data into summary data in realtime with continuous SQL queries and store the results in PipelineDB.



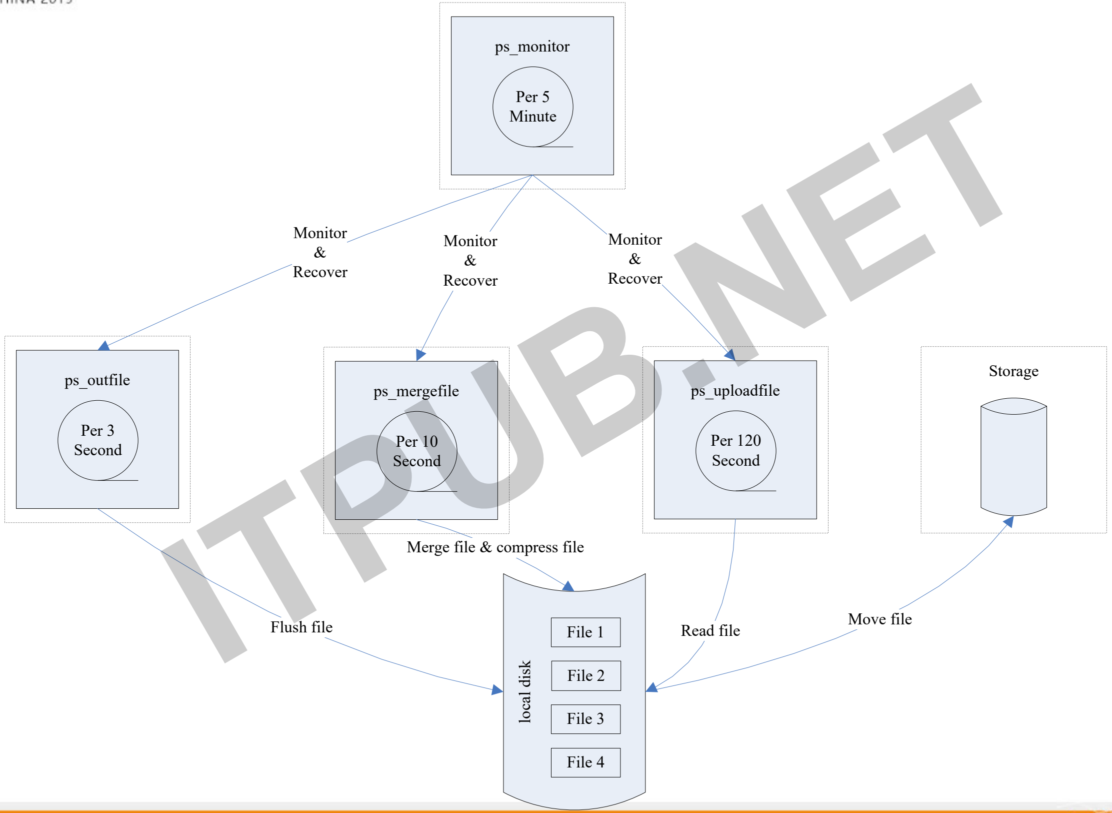
离线计算

基于PostgreSQL的MPP数据仓库



segment并发从gpfdist加载

数据采集服务


ps_monitor
Per 5 Minute
Monitor & Recover
Monitor & Recover
Monitor & Recover
ps_outfile
Per 3 Second
ps_mergefile
Per 10 Second
ps_uploadfile
Per 120 Second
Storage
Merge file & compress file
Flush file
Read file
Move file
local disk
File 1
File 2
File 3
File 4

运行日志

```
2019-04-04 16:59:08 [Note] ********************************************ps_monitor.sh(begin)************************************************
2019-04-04 16:59:08 [Note] process (ps_monitor.sh),has been running 3 times.
2019-04-04 16:59:08 [Note] psoutpath dir '/tmp/ps' has used (3%)
2019-04-04 16:59:08 [Note] remotepath dir '/home/op1/share' has used (73%)
2019-04-04 16:59:08 [Note] set events_statements_history_long enabled.
2019-04-04 16:59:08 [Note] process (ps_outfile.sh) has been started.
2019-04-04 16:59:08 [Note] performance_schema outfile 'ps_outfile' process is starting,scan one time per (3) second,flush table size is (6000).
2019-04-04 16:59:08 [Note] set performance_schema.setup_instruments success.
2019-04-04 16:59:08 [Note] process (ps_mergefile.sh),pid 15199 has been running for 64 seconds.
2019-04-04 16:59:08 [Note] mysql version is 5.6.21-ctrip-log,mysql starttime is 2019-04-04 16:48:21.000000.
2019-04-04 16:59:08 [Note] process (ps_uploadfile.sh),pid 15229 has been running for 64 seconds.
2019-04-04 16:59:08 [Note] ********************************************ps_monitor.sh(end)**************************************************
2019-04-04 17:12:46 [Note] outfile 'test_20190404_171246',size is 10000
2019-04-04 17:13:15 [Note] outfile 'test_20190404_171315',size is 10000
2019-04-04 17:13:18 [Note] outfile 'test_20190404_171318',size is 10000
2019-04-04 17:13:21 [Note] outfile 'test_20190404_171321',size is 10000
2019-04-04 17:13:24 [Note] outfile 'test_20190404_171324',size is 10000
2019-04-04 17:13:27 [Note] outfile 'test_20190404_171327',size is 10000
2019-04-04 17:13:43 [Note] outfile 'test_20190404_171343',size is 10000
2019-04-04 17:13:46 [Note] outfile 'test_20190404_171346',size is 10000
2019-04-04 17:13:59 [Note] outfile 'test_20190404_171359',size is 10000
2019-04-04 17:13:59 [Note] compress file test_20190404_171246.psmerge.tgz finish
2019-04-04 17:14:02 [Note] outfile 'test_20190404_171402',size is 10000
2019-04-04 17:14:05 [Note] outfile 'test_20190404_171405',size is 10000
2019-04-04 17:14:06 [Note] upload file /tmp/ps/mergefiles/test_20190404_171246.psmerge.tgz success.
```

注意的问题

- 超长SQL截断

SQL_TEXT列/ DIGEST_TEXT列标准化SQL长度字节数
performance_schema_max_sql_text_length=1024
performance_schema_max_digest_length=1024

- 导出SQL语句分隔符

FIELDS TERMINATED BY '|&*|' LINES TERMINATED BY '|&&|'

踩过的坑

- SQL执行时间

✓ TIMER_START、TIMER_END，单位皮秒（10-12秒），相对MySQL服务启动时间

✓ 8字节 bigint unsigned, 0到18446744073709551615

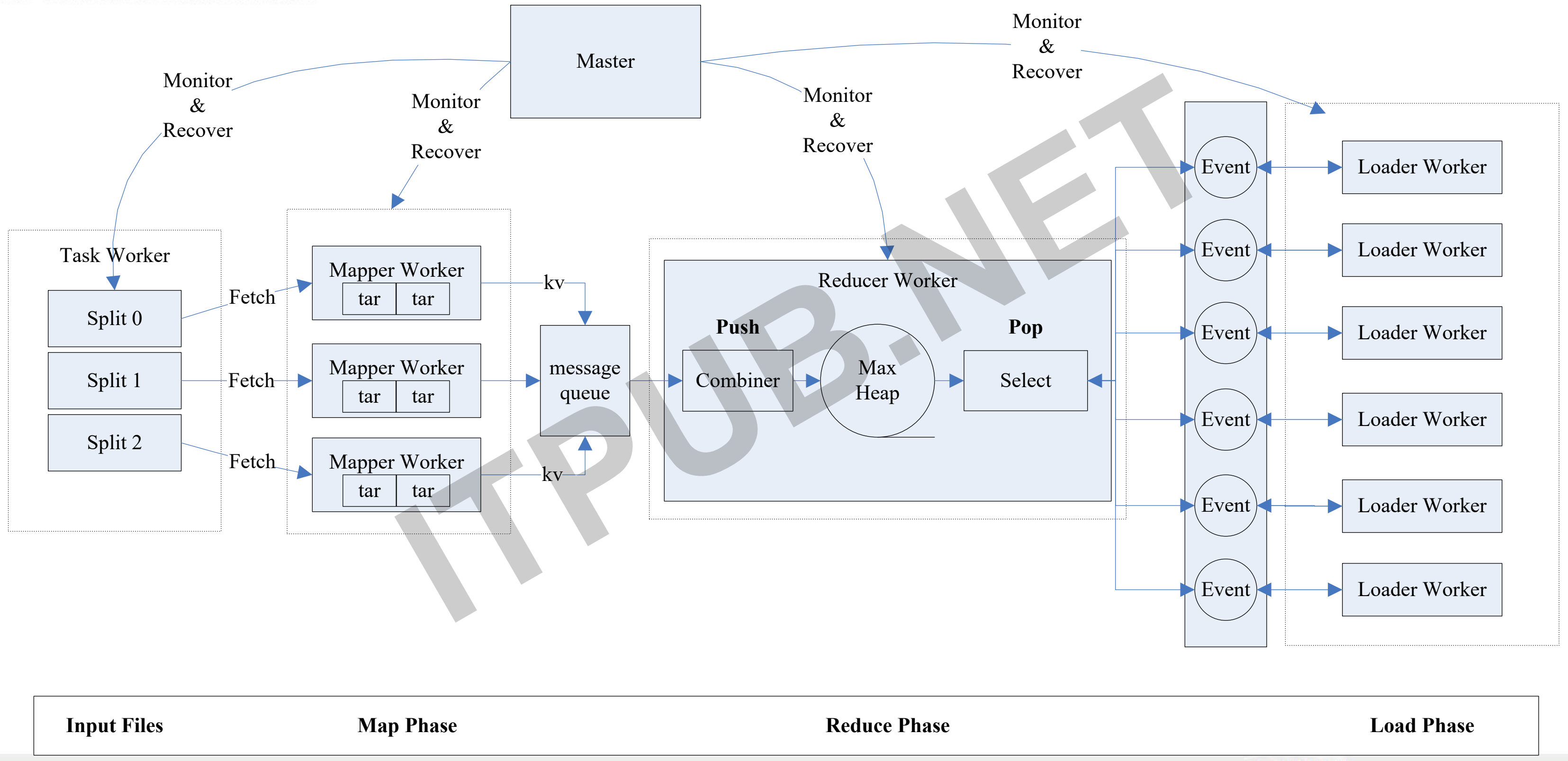
✓ 18446744073709551615*10e-13/(60*60*24)=213.5天

✓ 一直增长，达到上限后，重新计算

SQL开始执行时间(MySQL5.6)

```
SELECT
DATE_SUB(NOW(), INTERVAL (VARIABLE_VALUE - ((18446744073709551615)*10e-13)*
(VARIABLE_VALUE div (18446744073709551615*10e-13))- TIMER_START*10e-13) second)
as START_TIME
,SQL_TEXT
,DIGEST
,DIGEST_TEXT
,ROWS_AFFECTED
,ROWS_SENT
,ROWS_EXAMINED
FROM performance_schema.events_statements_history_long
JOIN information_schema.global_status
WHERE variable_name = 'UPTIME' limit 1\G
```

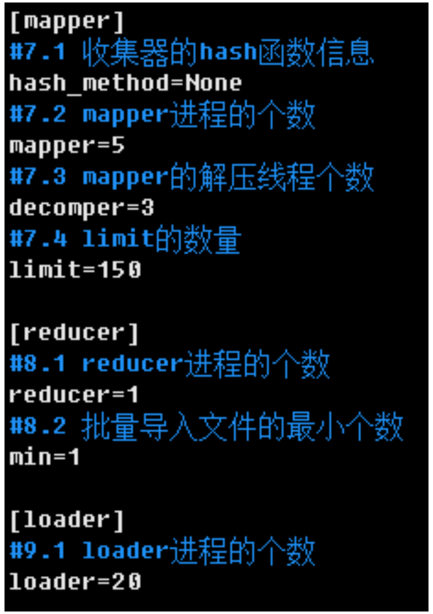
数据加载服务


Master
Monitor & Recover
Monitor & Recover
Monitor & Recover
Monitor & Recover
Task Worker
Split 0
Split 1
Split 2
Fetch
Fetch
Fetch
Mapper Worker
tar
tar
kv
kv
message queue
Reducer Worker
Push
Pop
Combiner
Max Heap
Select
Event
Loader Worker
Input Files
Map Phase
Reduce Phase
Load Phase

服务配置

四种类型worker

- 1个tasker，收集导入的文件信息，生成hash分配给mapper
- 5个mapper（每个mapper内有3个解压线程）
  拷贝解压文件，并将文件key-file键值传递给reducer
- 1个reducer，mapper的消费者
  生成优先队列，将高优先级的文件key-table传递给loader
- 20个loader(每个loader使用一个数据库长连接)
  将数据文件载入到对应的数据表中



```
[mapper]
#7.1 收集器的hash函数信息
hash_method=None
#7.2 mapper进程的个数
mapper=5
#7.3 mapper的解压线程个数
decomper=3
#7.4 limit的数量
limit=150

[reducer]
#8.1 reducer进程的个数
reducer=1
#8.2 批量导入文件的最小个数
min=1

[loader]
#9.1 loader进程的个数
loader=20
```

运行情况

- 每日加载的数据量在15T左右
- 单个文件(20万数据)的加载速度平均400ms
- 每日累计处理消息数量在120亿左右
- 提供1分钟聚合服务，数据延迟在2分钟内

```
Ysera: Monitor Process
Ysera: Tasker Process collect 5/(175 uncopy) tasks from /data/loader/trace
Ysera: cmapper0 Process 9/14 tasks begin
Ysera: cmapper1 Process 9/21 tasks begin
Ysera: cmapper2 Process 9/15 tasks begin
Ysera: cmapper3 Process 0/0 tasks begin
Ysera: cmapper4 Process 0/0 tasks begin
Ysera: creducer Process loading rate:420.15 MB/s(total 12191.26GB today)
Ysera: cloader0 Process idle
Ysera: cloader1 Process load 3 files into test
Ysera: cloader2 Process idle
Ysera: cloader3 Process load 6 files into test1
Ysera: cloader4 Process idle
Ysera: cloader5 Process load 4 files into test2
```

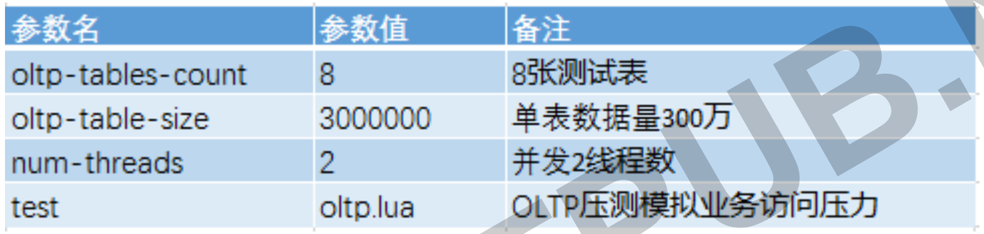
平台功能

- 性能分析
- SQL审计
- 执行SQL明细查询
- SQL优化建议
- 健康检查

性能分析案例

慢查询SQL导致CPU飙升的问题

- sysbench压测工具模拟正常业务访问



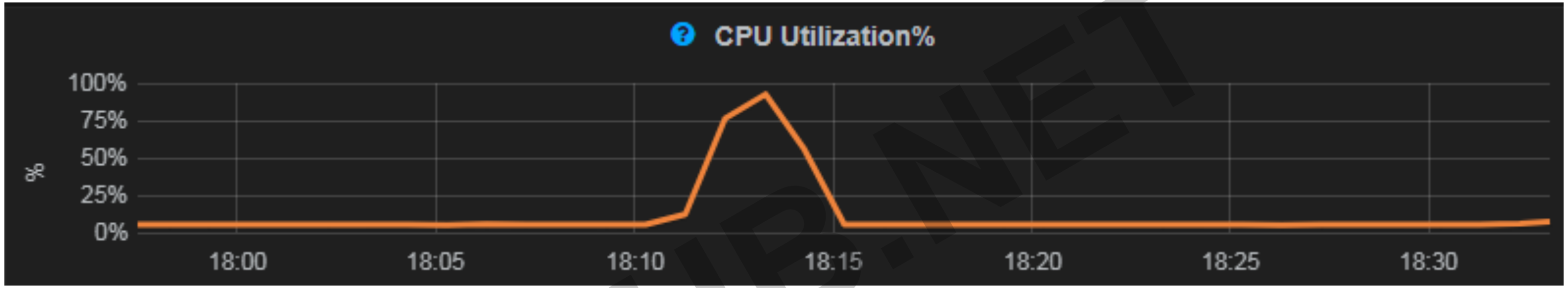
参数名	参数值	备注
oltp-tables-count	8	8张测试表
oltp-table-size	3000000	单表数据量300万
num-threads	2	并发2线程数
test	oltp.lua	OLTP压测模拟业务访问压力

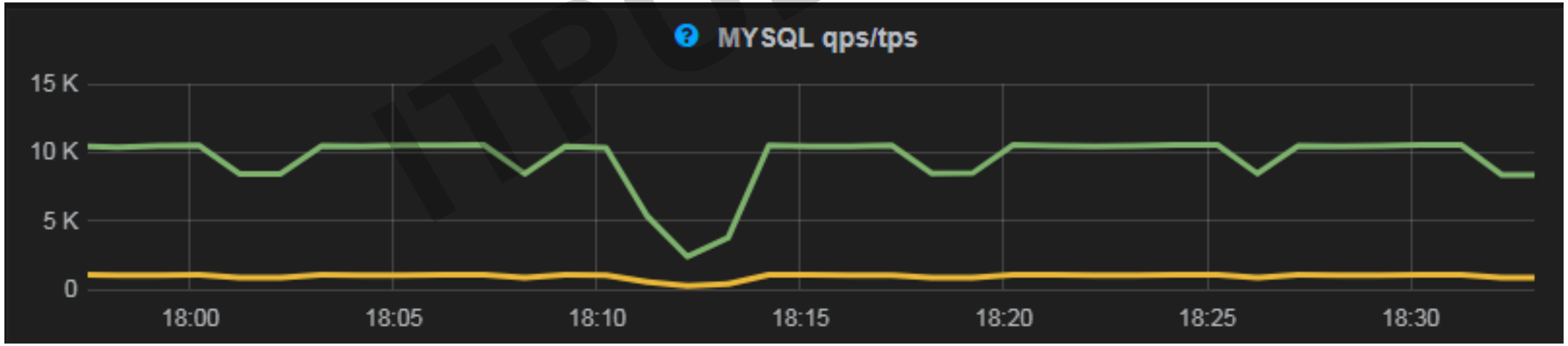
- mysqlslap工具并发50个线程模拟慢查询的业务访问

```
SELECT * FROM testdb.sbtest1 where k>1648952 ORDER BY rand() LIMIT 10;
```

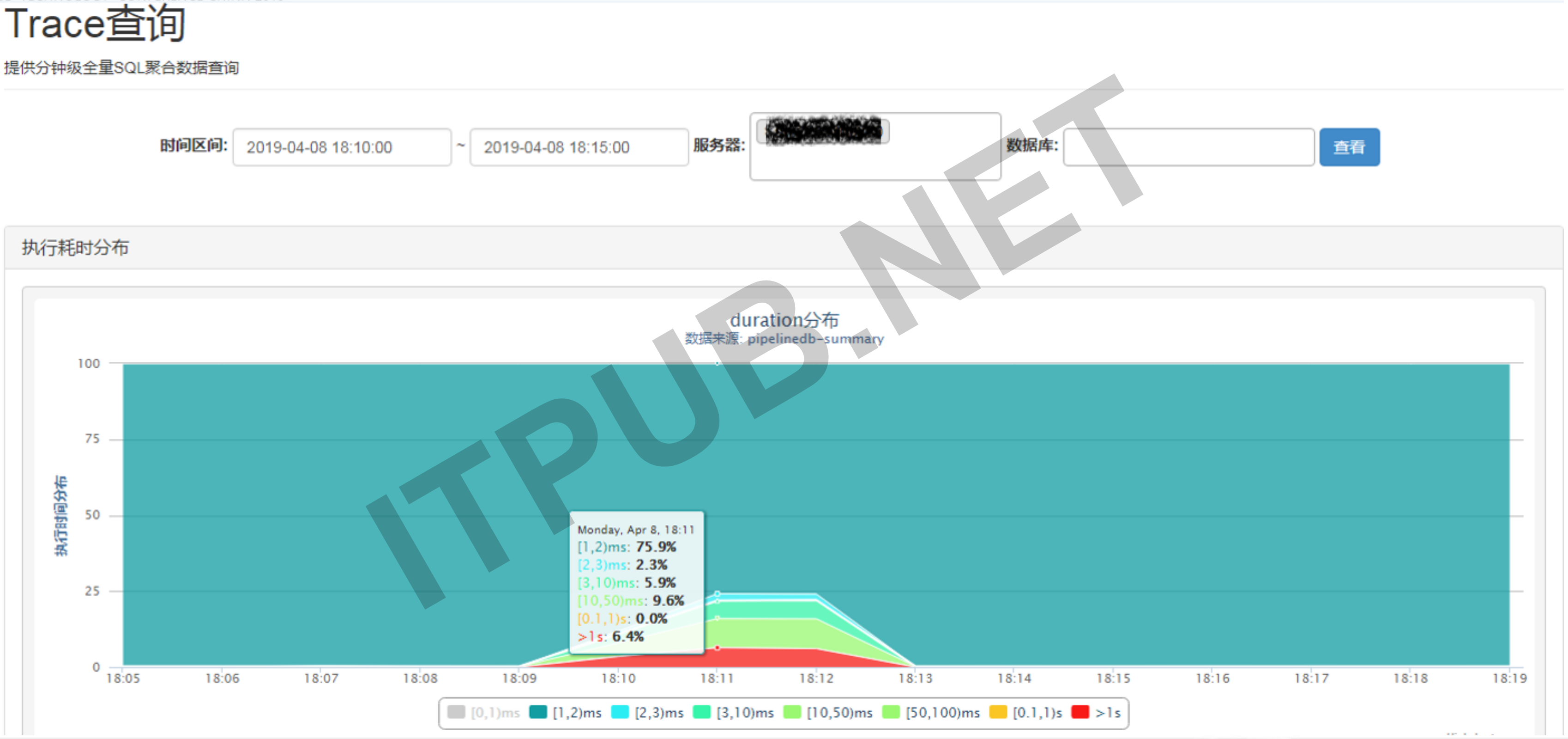
现象

18:10-18:15之间CPU飙升90%以上，QPS却大幅下降





SQL执行耗时分布



Trace查询

提供分钟级全量SQL聚合数据查询

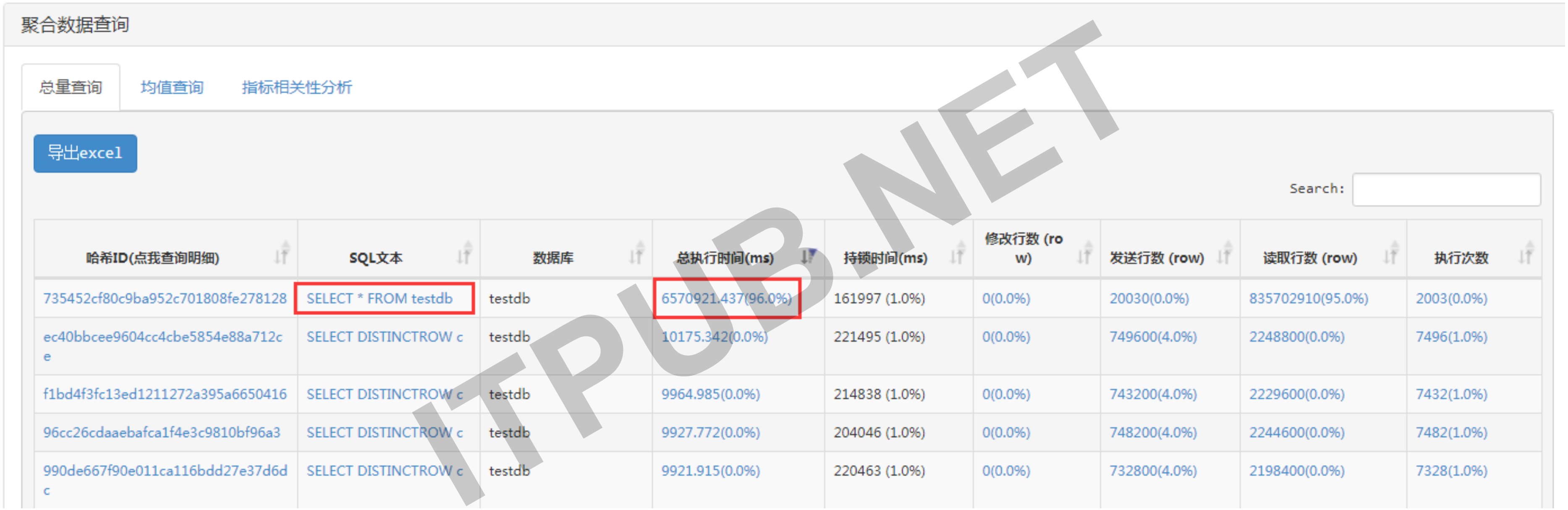
时间区间: 2019-04-08 18:10:00 ~ 2019-04-08 18:15:00 服务器: [redacted] 数据库: 查看

执行耗时分布

```
Monday, Apr 8, 18:11
[1,2)ms: 75.9%
[2,3)ms: 2.3%
[3,10)ms: 5.9%
[10,50)ms: 9.6%
[0.1,1)s: 0.0%
>1s: 6.4%
```

DTCC 2019
第十届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2019

SQL聚合数据



聚合数据查询

总量查询 均值查询 指标相关性分析

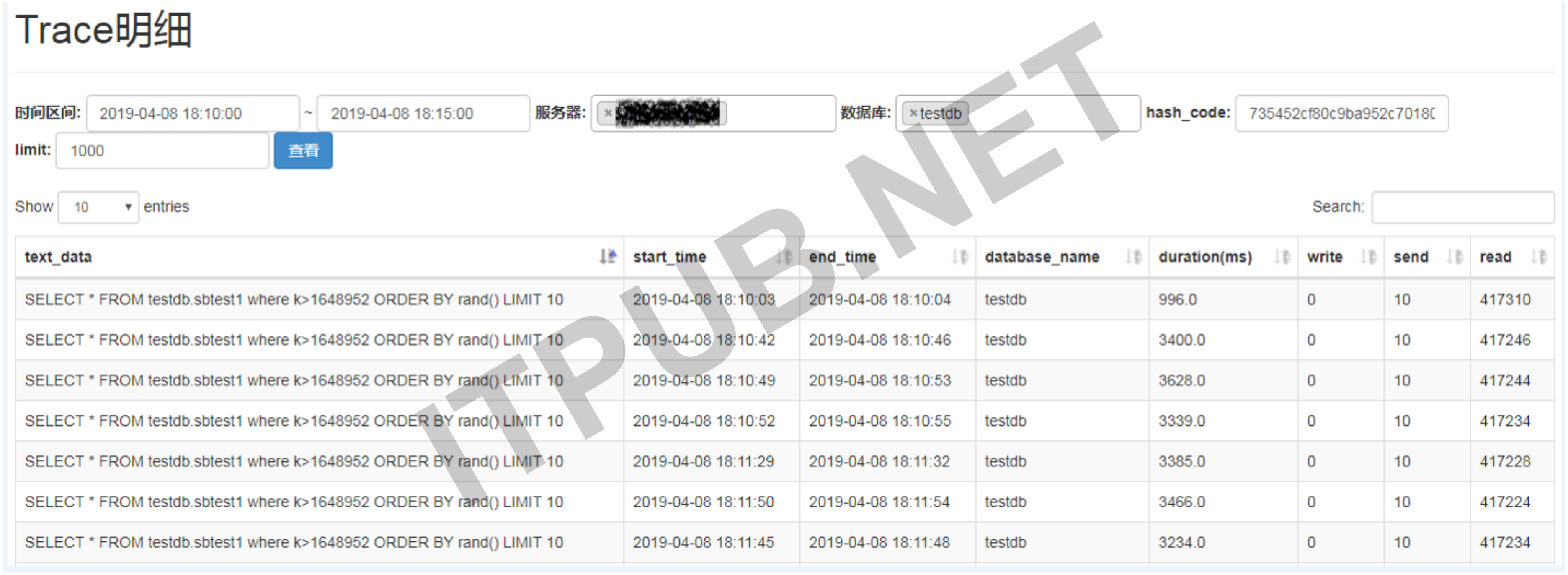
导出excel

Search:

哈希ID(点我查询明细)	SQL文本	数据库	总执行时间(ms)	持锁时间(ms)	修改行数 (row)	发送行数 (row)	读取行数 (row)	执行次数
735452cf80c9ba952c701808fe278128	SELECT * FROM testdb	testdb	6570921.437(96.0%)	161997 (1.0%)	0(0.0%)	20030(0.0%)	835702910(95.0%)	2003(0.0%)
ec40bbcee9604cc4cbe5854e88a712ce	SELECT DISTINCTROW c	testdb	10175.342(0.0%)	221495 (1.0%)	0(0.0%)	749600(4.0%)	2248800(0.0%)	7496(1.0%)
f1bd4f3fc13ed1211272a395a6650416	SELECT DISTINCTROW c	testdb	9964.985(0.0%)	214838 (1.0%)	0(0.0%)	743200(4.0%)	2229600(0.0%)	7432(1.0%)
96cc26cdaaebafca1f4e3c9810bf96a3	SELECT DISTINCTROW c	testdb	9927.772(0.0%)	204046 (1.0%)	0(0.0%)	748200(4.0%)	2244600(0.0%)	7482(1.0%)
990de667f90e011ca116bdd27e37d6dc	SELECT DISTINCTROW c	testdb	9921.915(0.0%)	220463 (1.0%)	0(0.0%)	732800(4.0%)	2198400(0.0%)	7328(1.0%)

DTCC 2019 第十届中国数据库技术大会 DATABASE TECHNOLOGY CONFERENCE CHINA 2019

SQL执行明细



Trace明细

时间区间: 2019-04-08 18:10:00 ~ 2019-04-08 18:15:00 服务器: [redacted] 数据库: ×testdb hash_code: 735452cf80c9ba952c7018(

limit: 1000 查看

Show 10 entries Search:

text_data	start_time	end_time	database_name	duration(ms)	write	send	read
SELECT * FROM testdb.sbtest1 where k>1648952 ORDER BY rand() LIMIT 10	2019-04-08 18:10:03	2019-04-08 18:10:04	testdb	996.0	0	10	417310
SELECT * FROM testdb.sbtest1 where k>1648952 ORDER BY rand() LIMIT 10	2019-04-08 18:10:42	2019-04-08 18:10:46	testdb	3400.0	0	10	417246
SELECT * FROM testdb.sbtest1 where k>1648952 ORDER BY rand() LIMIT 10	2019-04-08 18:10:49	2019-04-08 18:10:53	testdb	3628.0	0	10	417244
SELECT * FROM testdb.sbtest1 where k>1648952 ORDER BY rand() LIMIT 10	2019-04-08 18:10:52	2019-04-08 18:10:55	testdb	3339.0	0	10	417234
SELECT * FROM testdb.sbtest1 where k>1648952 ORDER BY rand() LIMIT 10	2019-04-08 18:11:29	2019-04-08 18:11:32	testdb	3385.0	0	10	417228
SELECT * FROM testdb.sbtest1 where k>1648952 ORDER BY rand() LIMIT 10	2019-04-08 18:11:50	2019-04-08 18:11:54	testdb	3466.0	0	10	417224
SELECT * FROM testdb.sbtest1 where k>1648952 ORDER BY rand() LIMIT 10	2019-04-08 18:11:45	2019-04-08 18:11:48	testdb	3234.0	0	10	417234

数据库运维经验分享

THANKS