



数据分析领域的黑马 --ClickHouse



Power Your Data



新浪-高鹏-2017年11月





“世上无难事，只要有捷径”



“工具选的好，下班回家早”



目录

- 自我介绍
- 数据分析面临的问题
- ClickHouse原理、架构
- ClickHouse在新浪的实践与经验
- ClickHouse案例、生态

关于我

我是谁？

我是干啥的？

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



关于我

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



DBA

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



关于我

DBA

DA

关于我们

Data Analyst
Data Translator

致力于**运维大数据**
挖掘与分析

可视化、报警、数据分析

AI-OPS



“表哥” “表姐”们

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



我们需要什么样的工具？

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE





Excel?





也用





Hadoop Spark Hive



?





But,
Hadoop这玩意,
不是一天就
能玩得转的啊~~



Google用Hadoop





多数人用Hadoop





太重了~



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



一切以需求作为第一位~

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE





一切以需求作为第一位~



快速~ 好用~ 体量够用~





一切以需求作为第一位~



快速~ 好用~ 体量够用~



好维护!!!





对结构化的数据



快速给出聚合/过滤结果



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



We Need

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



SQL

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



Fast SQL

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE





Fast Complex



SQL





没有什么数据统计是一个SQL解决不了的。

如果有，那就2个





俄罗斯搜索巨头Yandex开源

异步复制 OLAP

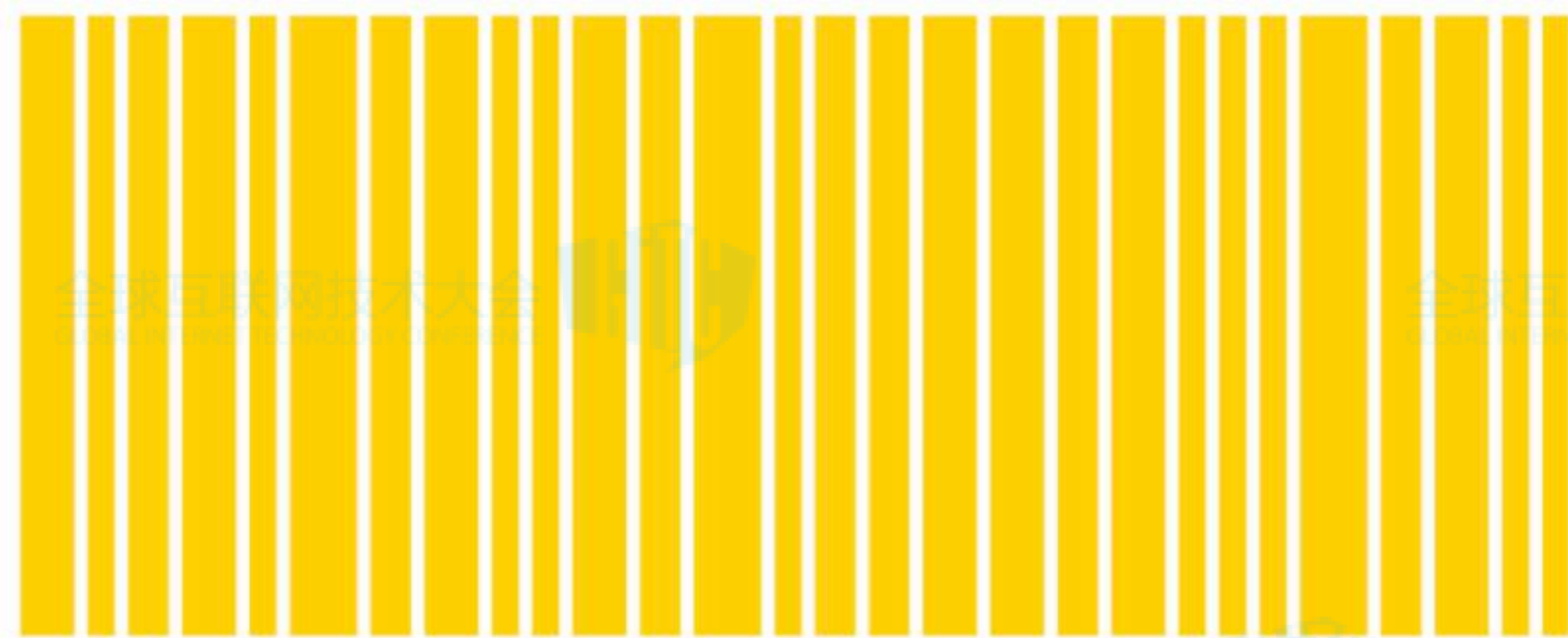
SQL

最终一致



统计函数

压缩



列式存储

PB级别

集群

驱动丰富



updated in real time

超高性能

线性扩展



跨数据中心

然鹅，

不支持事务

不支持update/delete

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



But,

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



查询‘巨’快

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



超大容量

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE





Let's Begin



部署：单机

部署

1. 官方提供Ubuntu包
2. 第三方rpm包
2. Docker镜像

需要注意：

1. 修改网络，默认监控IPv4/v6
2. 自定义数据目录，修改官方启动脚本
3. Docker修改时区

```
mysql> :) select version();
SELECT version()
+-----+
| version() |
+-----+
| 1.1.54289 |
+-----+
1 rows in set. Elapsed: 0.003 sec.

mysql> :) show tables ;
SHOW TABLES
+-----+
| name |
+-----+
| test |
+-----+
1 rows in set. Elapsed: 0.002 sec.
```

部署：单机

是不是很SQL

部署：单机

蚝，

我们来压测一下~

Talk is cheap. Show
me the code.

Linus Torvalds

quote fancy

部署：单机

数据源

USA civil flights data
since 1987 till 2015

contains **166 millions** rows
63 GB of uncompressed data

```
# 下载数据
for s in `seq 1987 2017`
do
for m in `seq 1 12`
do
wget http://transtats.bts.gov\
/PREZIP/On_Time_On_Time_Performance_${s}_${m}.zip
done
done

# 解压
for i in `ls *.zip`; do unzip -o $i;done

# 插入数据
for i in `ls *.csv`
do
echo '-----'
echo $i
du -sh $i
wc -l $i
time cat $i | sed 's/\.00//g' | sed 1d | clickhouse-client \
-h 127.0.0.1 --port 9000 -d gaopeng4 \
--query="INSERT INTO ontime FORMAT CSVWithNames";
echo '-----'
sleep 2
done
```


部署：单机

数据源

USA civil flights data
since 1987 till 2015

contains **166 millions** rows
63 GB of uncompressed data

数据大小

173MB

文件行数

436951

插入耗时

4.731 Sec

平均速度

9.3 W/Sec

压缩率

5倍

部署：单机

并发5个进程

```
while read a b c d e ;  
do  
echo $a $b $c $d $e;  
  
time cat $a | sed 's/\.\00//g' | sed 1d | clickhouse-client -h 127.0.0.1 \  
--port 9000 -d gaopeng4 --query="INSERT INTO ontime FORMAT CSVWithNames" &  
  
time cat $b | sed 's/\.\00//g' | sed 1d | clickhouse-client -h 127.0.0.1 \  
--port 9000 -d gaopeng4 --query="INSERT INTO ontime FORMAT CSVWithNames" &  
  
time cat $c | sed 's/\.\00//g' | sed 1d | clickhouse-client -h 127.0.0.1 \  
--port 9000 -d gaopeng4 --query="INSERT INTO ontime FORMAT CSVWithNames" &  
  
time cat $d | sed 's/\.\00//g' | sed 1d | clickhouse-client -h 127.0.0.1 \  
--port 9000 -d gaopeng4 --query="INSERT INTO ontime FORMAT CSVWithNames" &  
  
time cat $e | sed 's/\.\00//g' | sed 1d | clickhouse-client -h 127.0.0.1 \  
--port 9000 -d gaopeng4 --query="INSERT INTO ontime FORMAT CSVWithNames" &  
  
sleep 5  
done <<EOF  
`ls *.csv | xargs -n 5`  
EOF
```

机器负载

```
1 [|||||] 42.2% 7 [|||||] 61.0% 13 [|||||] 84.8% 19 [|||||] 78.0%  
2 [|||||] 57.4% 8 [|||||] 65.6% 14 [|||||] 58.9% 20 [|||||] 43.0%  
3 [|||||] 81.6% 9 [|||||] 85.7% 15 [|||||] 23.0% 21 [|||||] 31.2%  
4 [|||||] 35.7% 10 [|||||] 60.2% 16 [|||||] 68.2% 22 [|||||] 45.9%  
5 [|||||] 45.4% 11 [|||||] 95.7% 17 [|||||] 61.1% 23 [|||||] 27.8%  
6 [|||||] 77.5% 12 [|||||] 72.9% 18 [|||||] 43.5% 24 [|||||] 38.7%  
Mem[|||||] 11183/48092MB  
Swp[|||||] 354/8191MB  
Tasks: 71, 180 thr; 12 running  
Load average: 20.75 12.84 8.17  
Uptime: 45 days, 03:42:23
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
18534	clickhous	20	0	8834M	6160M	13140	S	199.12.8	20:40.99		clickhouse-server --daemon --pid-file=/var/run/clickhouse-server/clickhouse-server.pid --config-file

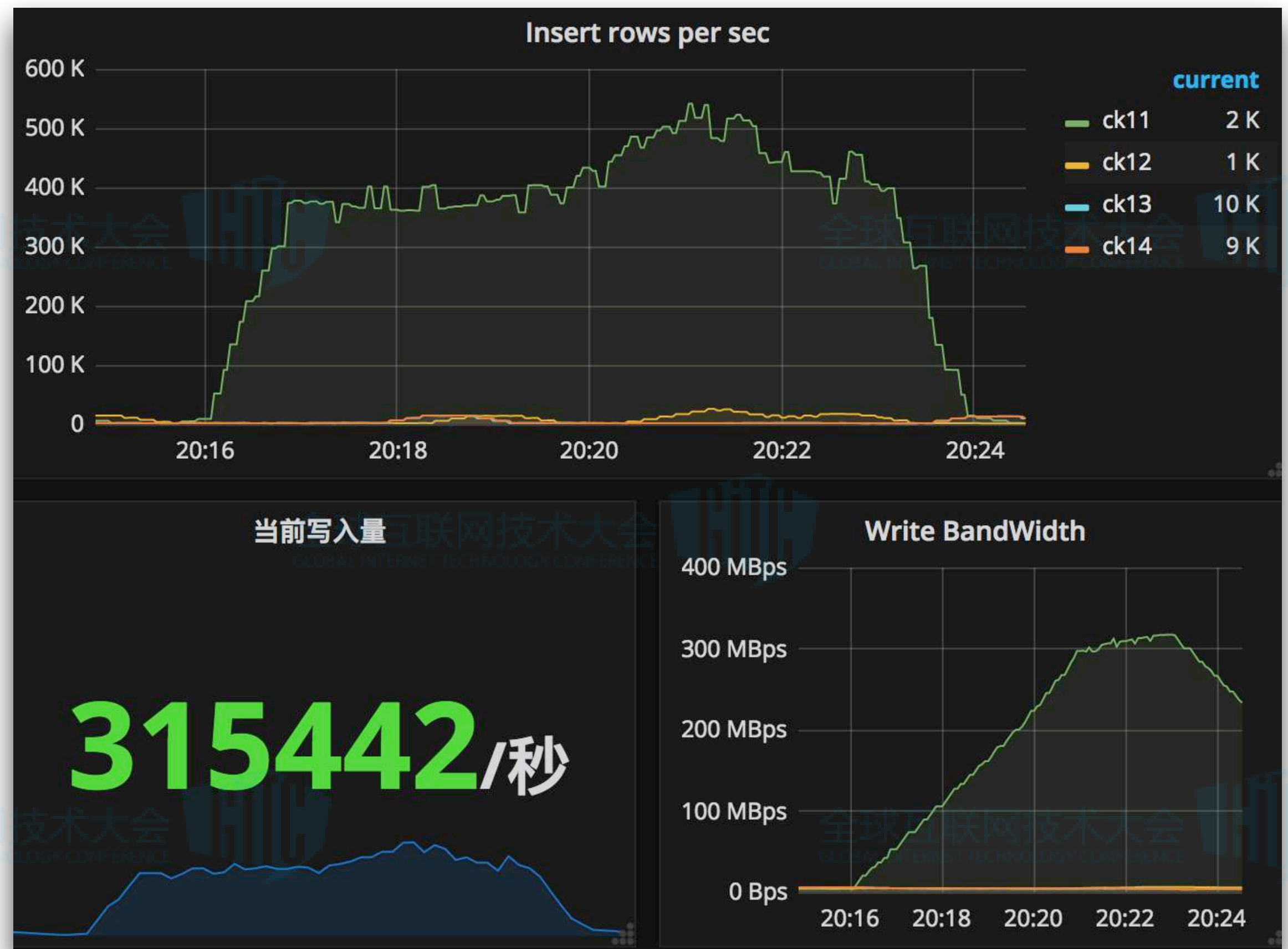
部署：单机

响应时间

```
SELECT elapsed  
FROM processes  
ORDER BY elapsed DESC  
LIMIT 10
```

elapsed
5.291229166
5.287489023
5.285718683
5.284233673
5.282636967
0.000210788

峰值50W QPS



部署：单机

查询类型

1. 查询总量

2. 简单group by

```
:) select count(*)/100000000 from ontime ;  
  
SELECT count(*) / 100000000  
FROM ontime  
┌─divide(count(), 100000000)─┐  
│ 1.71412868 │  
└──────────────────────────┘  
  
1 rows in set. Elapsed: 0.051 sec Processed 171.41 million rows, 171.41 MB (3.35 billion rows/s., 3.35 GB/s )  
  
:) select Year, count(*) as c1 from ontime group by Year limit 3  
:-] ;  
  
SELECT  
  Year,  
  count(*) AS c1  
FROM ontime  
GROUP BY Year  
LIMIT 3  
  
┌─Year─┐┌─c1─┐  
│ 1988 ││ 5202084 │  
│ 1989 ││ 5041188 │  
│ 1990 ││ 5270881 │  
└───┬──┘└───┬──┘  
  
3 rows in set. Elapsed: 0.208 sec. Processed 171.41 million rows, 342.83 MB (825.43 million rows/s., 1.65 GB/s.)
```

部署：单机

查询类型

- 条件查询, 聚合, 排序

```
SELECT
  DestCityName,
  uniqExact(OriginCityName) AS u
FROM ontime
WHERE (Year >= 2000) AND (Year <= 2010)
GROUP BY DestCityName
ORDER BY u DESC
LIMIT 10
```

DestCityName	u
Atlanta, GA	193
Chicago, IL	167
Dallas/Fort Worth, TX	161
Minneapolis, MN	138
Cincinnati, OH	138
Detroit, MI	130
Houston, TX	129
Denver, CO	127
Salt Lake City, UT	119
New York, NY	115

10 rows in set. Elapsed: 1.185 sec. Processed 72.79 million rows, 3.37 GB (61.45 million rows/s., 2.85 GB/s.)

部署：单机

查询类型

- 复杂查询

```
SELECT
  min(Year),
  max(Year),
  Carrier,
  count(*) AS cnt,
  sum(ArrDelayMinutes > 30) AS flights_delayed,
  round(sum(ArrDelayMinutes > 30) / count(*), 2) AS rate
FROM ontime
WHERE (DayOfWeek NOT IN (6, 7)) AND (OriginState NOT IN ('AK', 'HI', 'PR', 'VI')) AND (DestState NOT IN ('AK', 'HI', 'PR', 'VI')) AND (FlightDate < '2010-01-01')
GROUP BY Carrier
HAVING (cnt > 100000) AND (max(Year) > 1990)
ORDER BY rate DESC
LIMIT 1000
```

min(Year)	max(Year)	Carrier	cnt	flights_delayed	rate
2003	2009	EV	1454777	237698	0.16
2003	2009	B6	683874	103677	0.15
2006	2009	YV	740606	110389	0.15
2003	2009	FL	1082489	158748	0.15
2006	2009	XE	1016010	152431	0.15
2003	2005	DH	501056	69833	0.14
2001	2009	MQ	3238137	448037	0.14
2004	2009	OH	1195868	160071	0.13
2003	2006	RU	1007247	126733	0.13
1988	2009	UA	9593281	1197052	0.12
2003	2006	TZ	136735	16496	0.12
1988	2009	AA	10600421	1185336	0.11
1988	2009	CO	6029147	673863	0.11
1988	2001	TW	2659963	280741	0.11
1988	2009	DL	11869418	1156256	0.1
2003	2009	OO	2654259	257069	0.1
2007	2009	9E	577223	59437	0.1
1988	2009	NW	7601726	725460	0.1
1988	2009	US	10276931	991016	0.1
1988	2009	AS	1506003	146920	0.1
1988	2009	WN	12722172	1107840	0.09
1988	1991	PA	206841	19465	0.09
1988	2005	HP	2607603	235675	0.09
2005	2009	F9	307569	28679	0.09

24 rows in set. Elapsed: 1.094 sec. Processed 128.68 million rows, 1.57 GB (117.57 million rows/s., 1.44 GB/s.)

部署：单机

使用

1. 启动Server
2. use db, create table
3. 尽情select
4. 推荐引擎：MergeTree

```
CREATE TABLE apm.apm_msg (_clientip String, _data_size Float32,  
    date Date, ts DateTime, hour Int8, minute Int8)  
ENGINE = MergeTree(date, (minute, hour, date), 8192);
```

分区

主键

稀疏索引粒度

总结

优点：

1. 部署简单
2. 全部CPU打满，查询效率极高

问题：

1. 性能依赖单机（scale up路线）
2. 存在单点故障风险（宕机数据全丢）

MergeTree

写

- 如何写的快?
- 是否可压缩?

类似LSM Tree, 但是**没有内存表**, 不记录log

直接落磁盘, 按照**主键**排序, 分块写入

异步merge, 与写不冲突, 最大merge到月纬度

不支持删除、修改

primary.idx+*.bin+*.mrk+checksums.txt+columns.txt

读

- 如何快速查找?
- 数据量大, 如何适应内存?

• 主键查询:

eg: (x, y, z, date)

最左原则

• 其他列查询:

稀疏索引定位区间: 不适合点对点查询, 适合范围查询

颗粒度N: 默认8192

查询问题: 会带来过多的IO



缺乏：



扩展性



可靠性



如何获得：

扩展性

可靠性

部署：‘分布式’

■ 概括

假的‘scale out’

借助于特殊引擎实现

借助配置文件

部署：‘分布式’

Distributed引擎：

1. 本身不存储数据
2. 被写入，做转发
3. 查询，作为中间件，聚合后返回给用户

```
CREATE TABLE apm.apm_msg (_clientip String, _data_size Float32,  
    date Date, ts DateTime, hour Int8, minute Int8)  
ENGINE = MergeTree(date, (minute, hour, date), 8192);
```

分区

主键

颗粒度

```
CREATE TABLE apm.apm_msg_all  
(_clientip String, _data_size Float32,  
    date Date, ts DateTime, hour Int8, minute Int8)  
ENGINE = Distributed(bip_ck_cluster, apm, apm_msg, rand());
```

集群名称

库

表

分布算法



部署：‘分布式’

■ 分布式如何做到的

```
<clickhouse_remote_servers>
  <bip_ck_cluster>
    <shard>
      <internal_replication>true</internal_replication>
      <replica>
        <host>ck11. [REDACTED].com.cn</host>
        <port>9000</port>
      </replica>
    </shard>
    <shard>
      <replica>
        <internal_replication>true</internal_replication>
        <host>ck12. [REDACTED].com.cn</host>
        <port>9000</port>
      </replica>
    </shard>
    <shard>
      <internal_replication>true</internal_replication>
      <replica>
        <host>ck13. [REDACTED].com.cn</host>
        <port>9000</port>
      </replica>
    </shard>
    <shard>
      <internal_replication>true</internal_replication>
      <replica>
        <host>ck14. [REDACTED].com.cn</host>
        <port>9000</port>
      </replica>
    </shard>
  </bip_ck_cluster>
</clickhouse_remote_servers>
```

分片1

分片2

分片3

分片4

部署：‘分布式’

■ 写操作

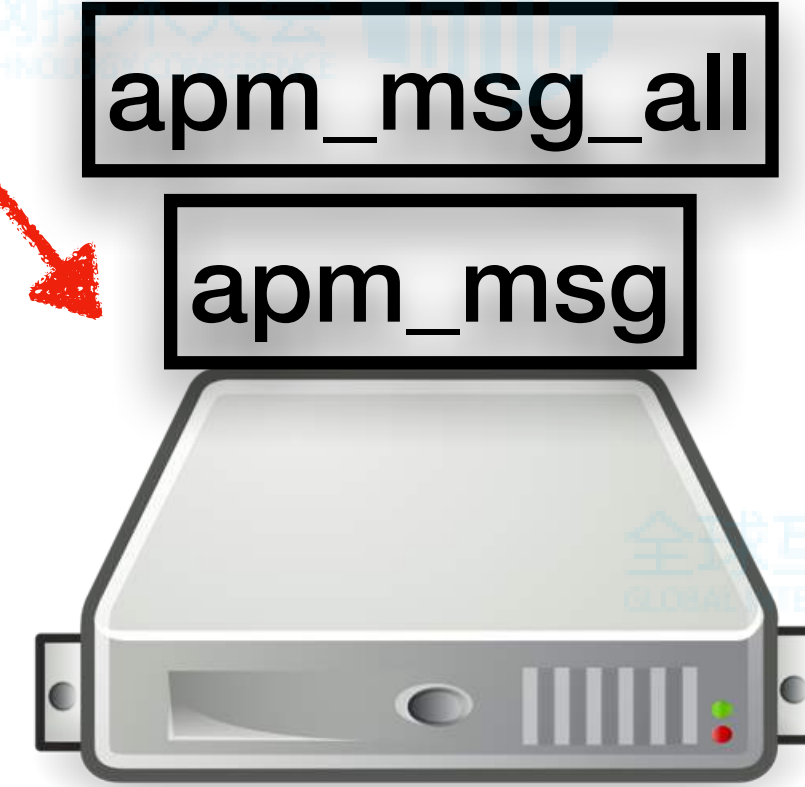
通过域名，写本地表

```
insert into apm_msg values()
```

总QPS=Σ单机QPS

ClickHouse-xxxxxx.sina.com.cn

写



部署：‘分布式’

■ 读操作

通过域名，读**分布式表**

```
select * from apm_msg_all  
where xxx=yyy
```

原则：确保其他节点返回的数据，自己还可以聚合，如top/group by逻辑就不同

ClickHouse-xxxxxx.sina.com.cn

读



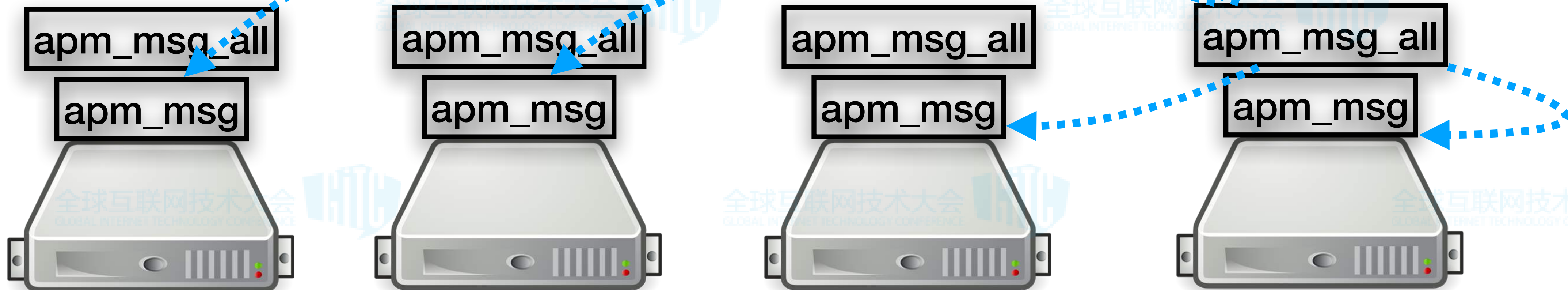
部署：‘分布式’

■ 读操作

ClickHouse-xxxxxx.sina.com.cn

收到请求的节点，
拿到所有节点数据
(各自计算一次)

本地汇总后

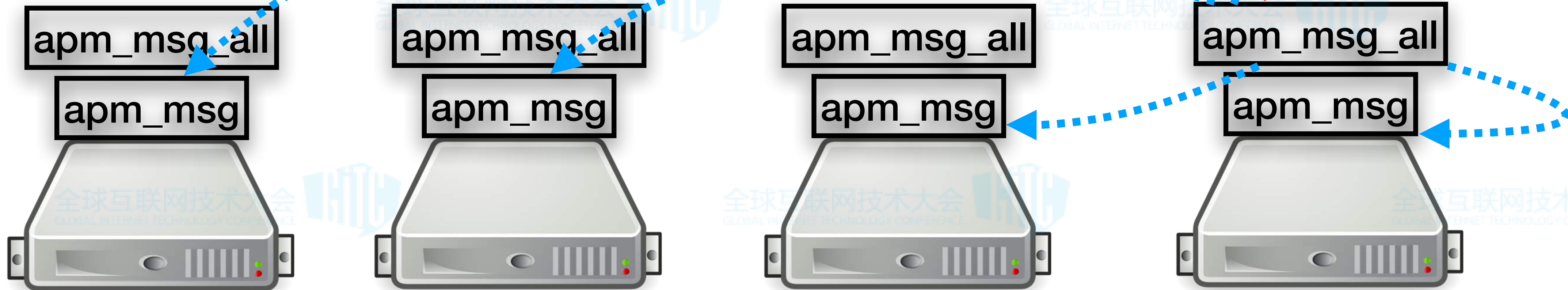


部署：‘分布式’

■ 读操作

返回给客户端

ClickHouse-xxxxxx.sina.com.cn



部署：‘分布式’

通过全局配置文件，达到集群相互知晓

■ 总结

各自维护各自的数据，让用户自己写入

水平**扩展性**很好

查询/写入能力随机器数**线性**增加

cluster config updated **on the fly**

部署：‘分布式’

1. 直接写分布式表，造成数据不均匀

问题

2. 域名映射的IP只有在初始解析

3. 新增节点，历史数据不会搬迁，造成不均衡

4. 过度的group by，导致大量数据交换

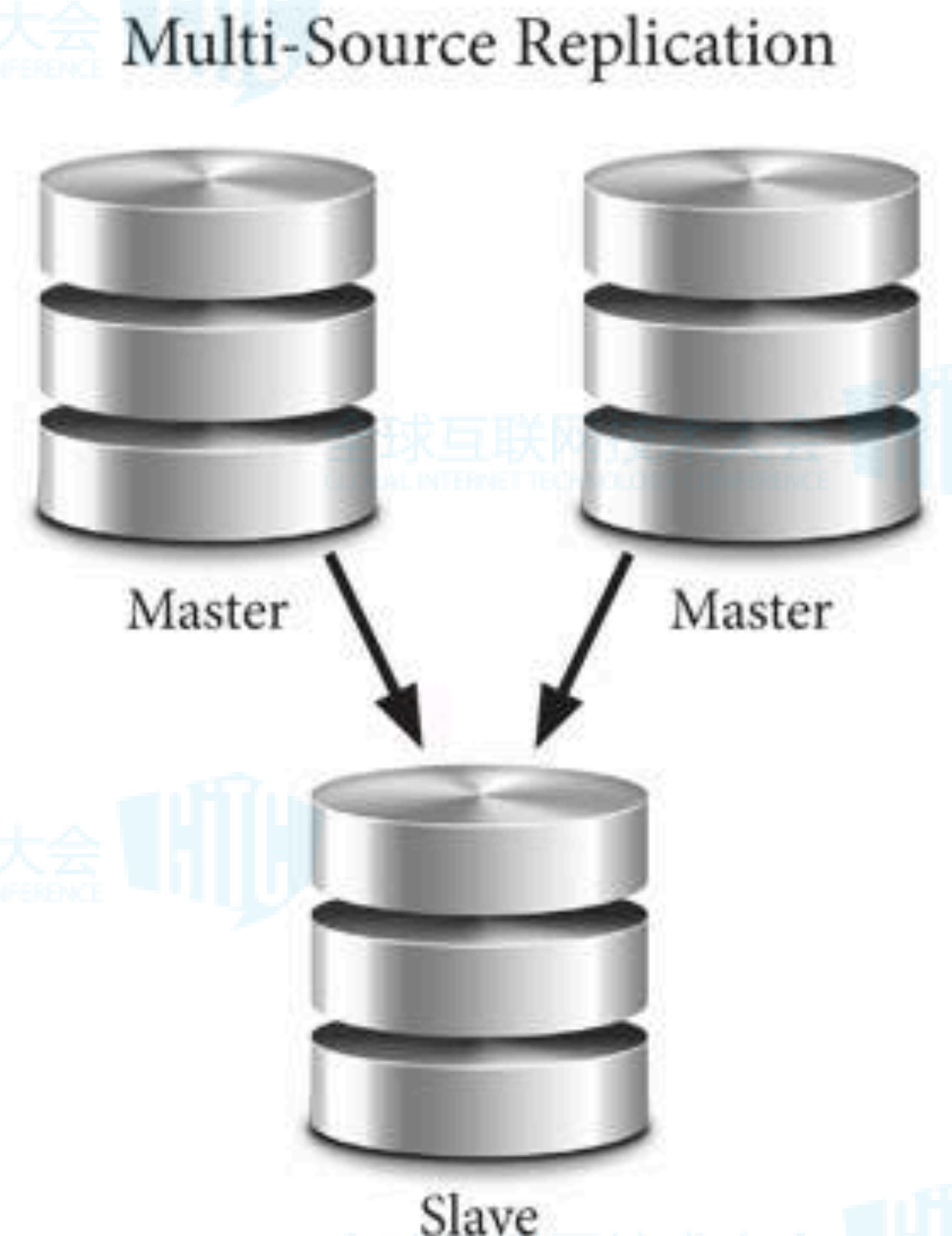
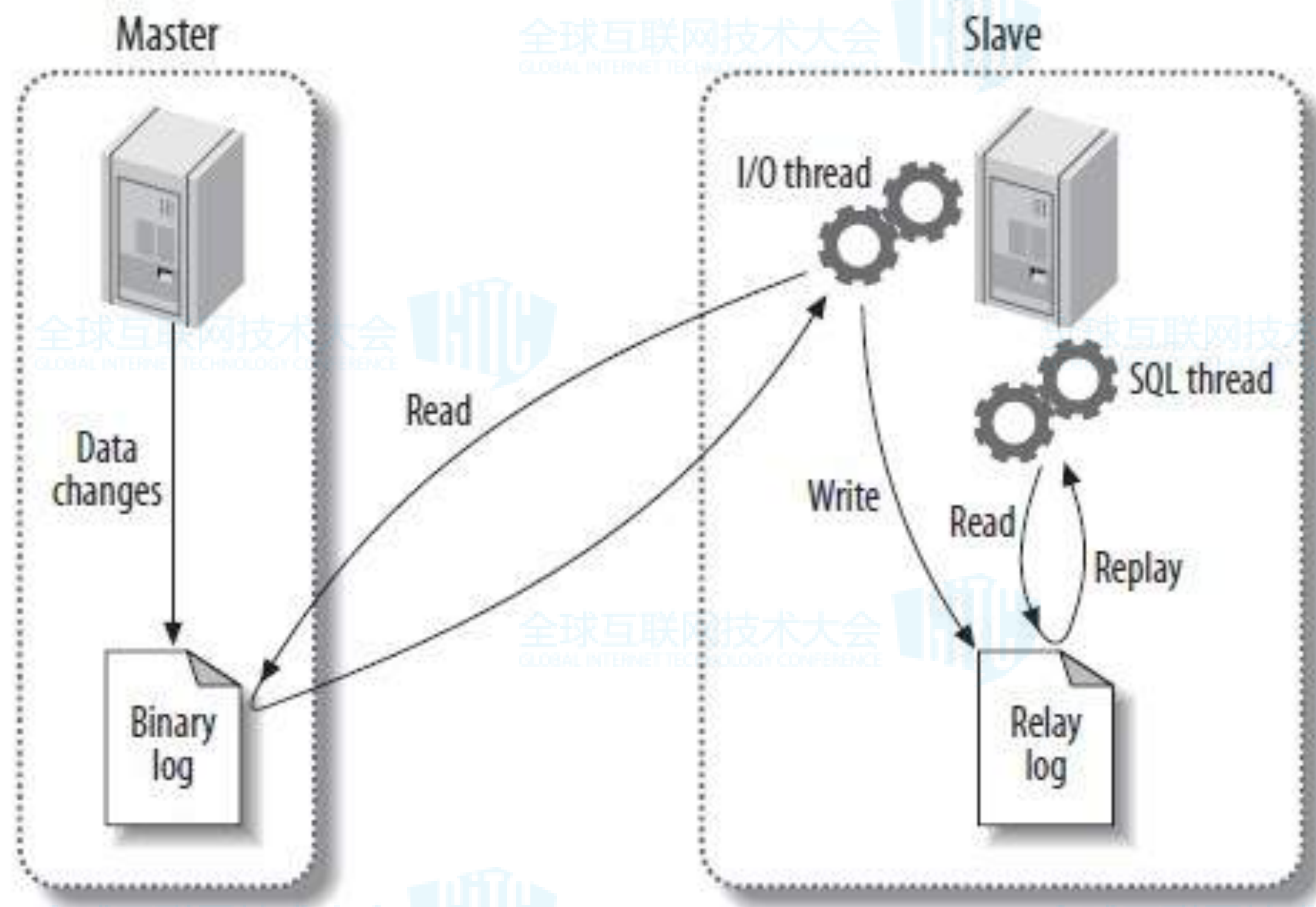
5. 数据分片依赖单机稳定性，缺乏可靠性

如何获得：

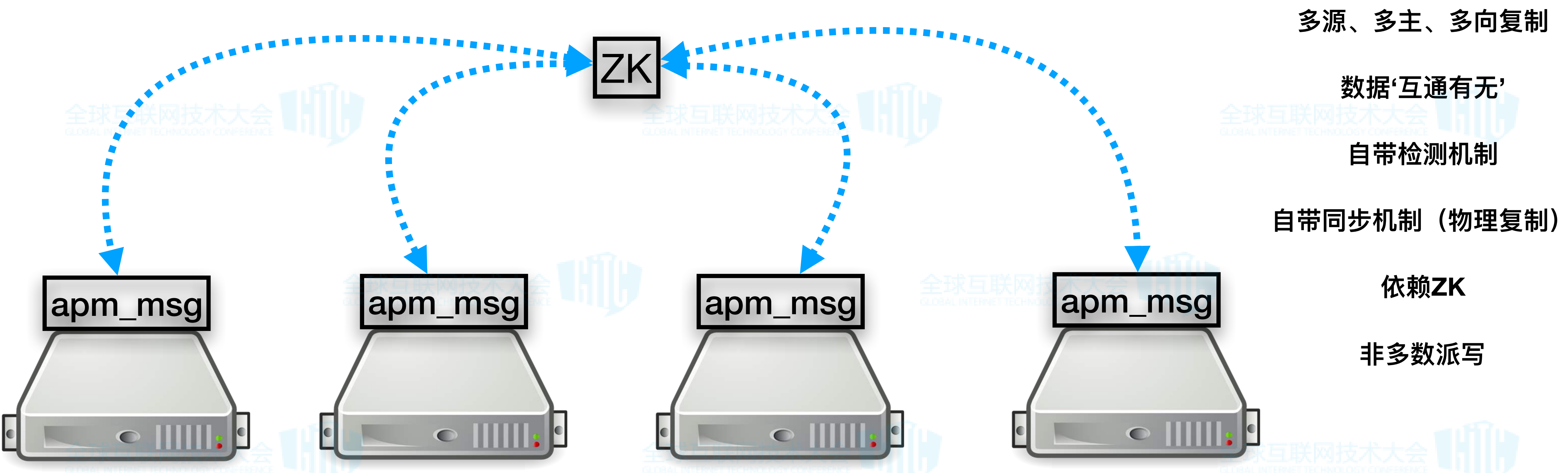
扩展性

可靠性

我们知道的复制：



ClickHouse的复制



多源、多主、多向复制

数据‘互通有无’

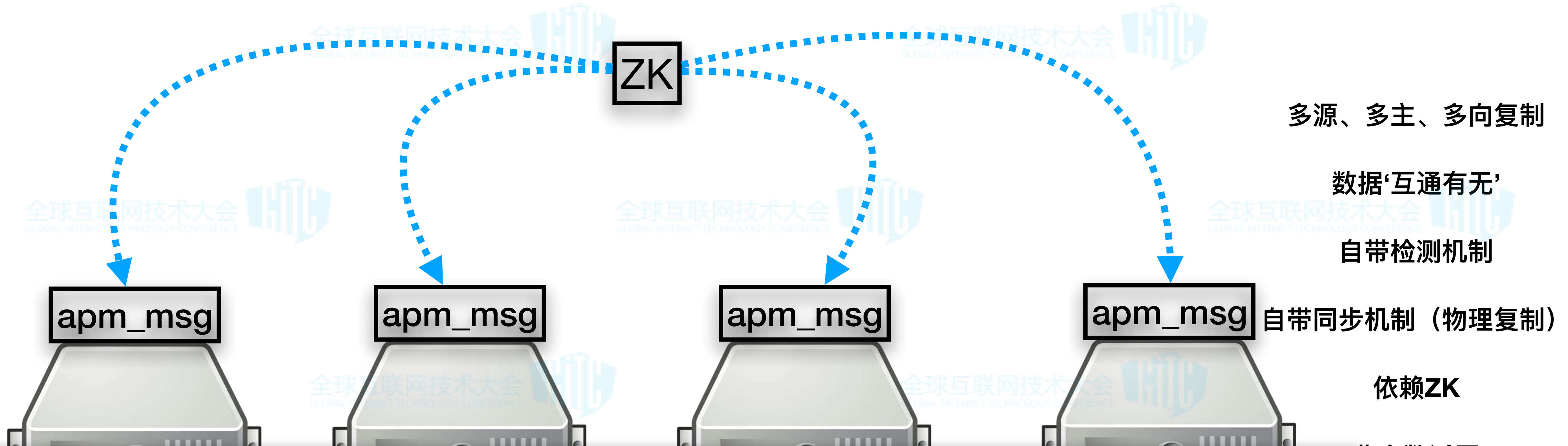
自带检测机制

自带同步机制（物理复制）

依赖ZK

非多数派写

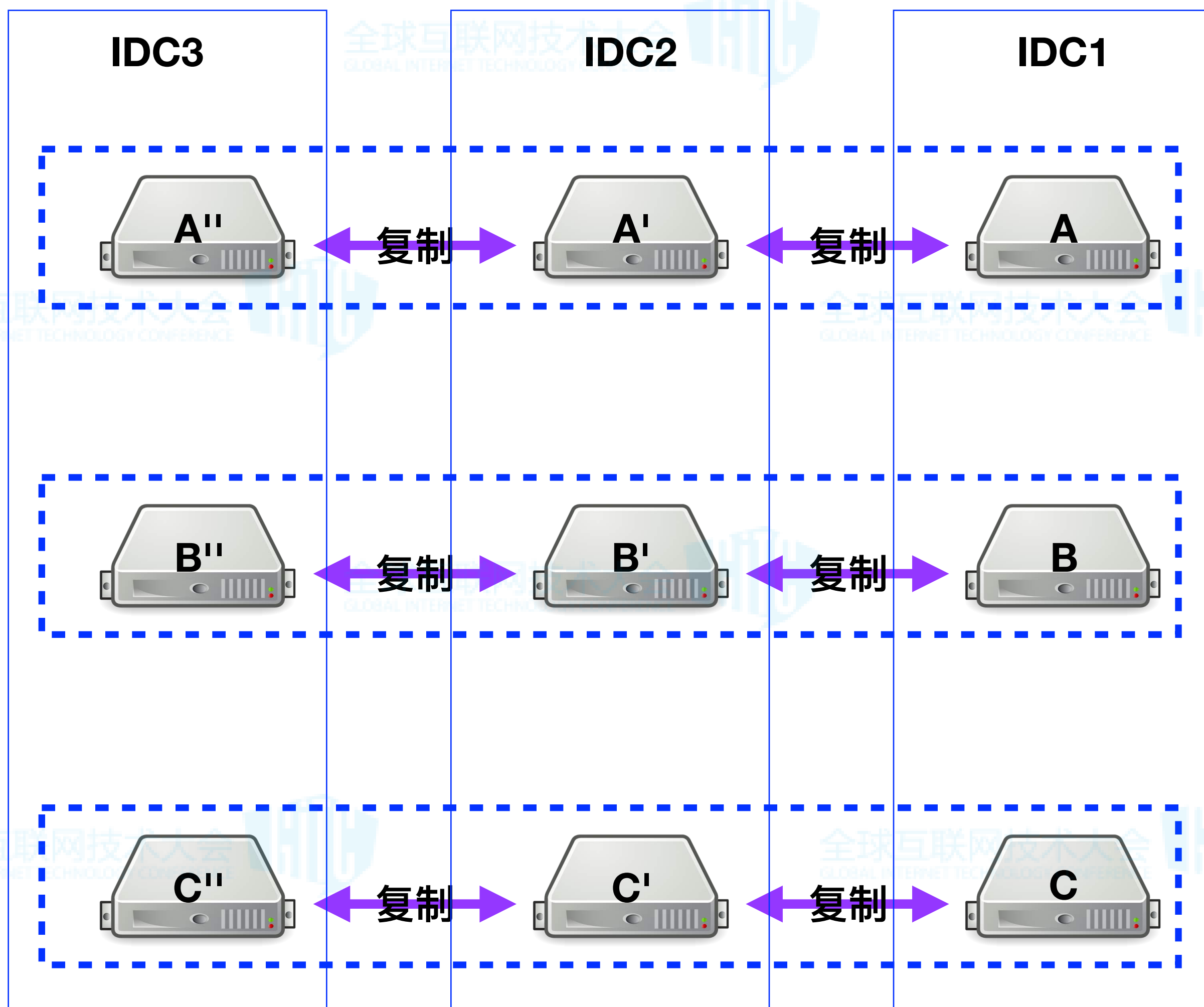
部署：复制



```
CREATE TABLE apm.apm_msg (_clientip String, _data_size Float32,  
    date Date, ts DateTime, hour Int8, minute Int8)  
ENGINE = ReplicatedMergeTree('/clickhouse/tables/apm_msg', '195', date, (minute, hour, date), 8192);
```

ReplicatedMergeTree('zk路径', '副本名称', 日期列, (其他列, 日期列), 索引粒度)

部署：最佳架构



3个IDC使用复制机制做互备

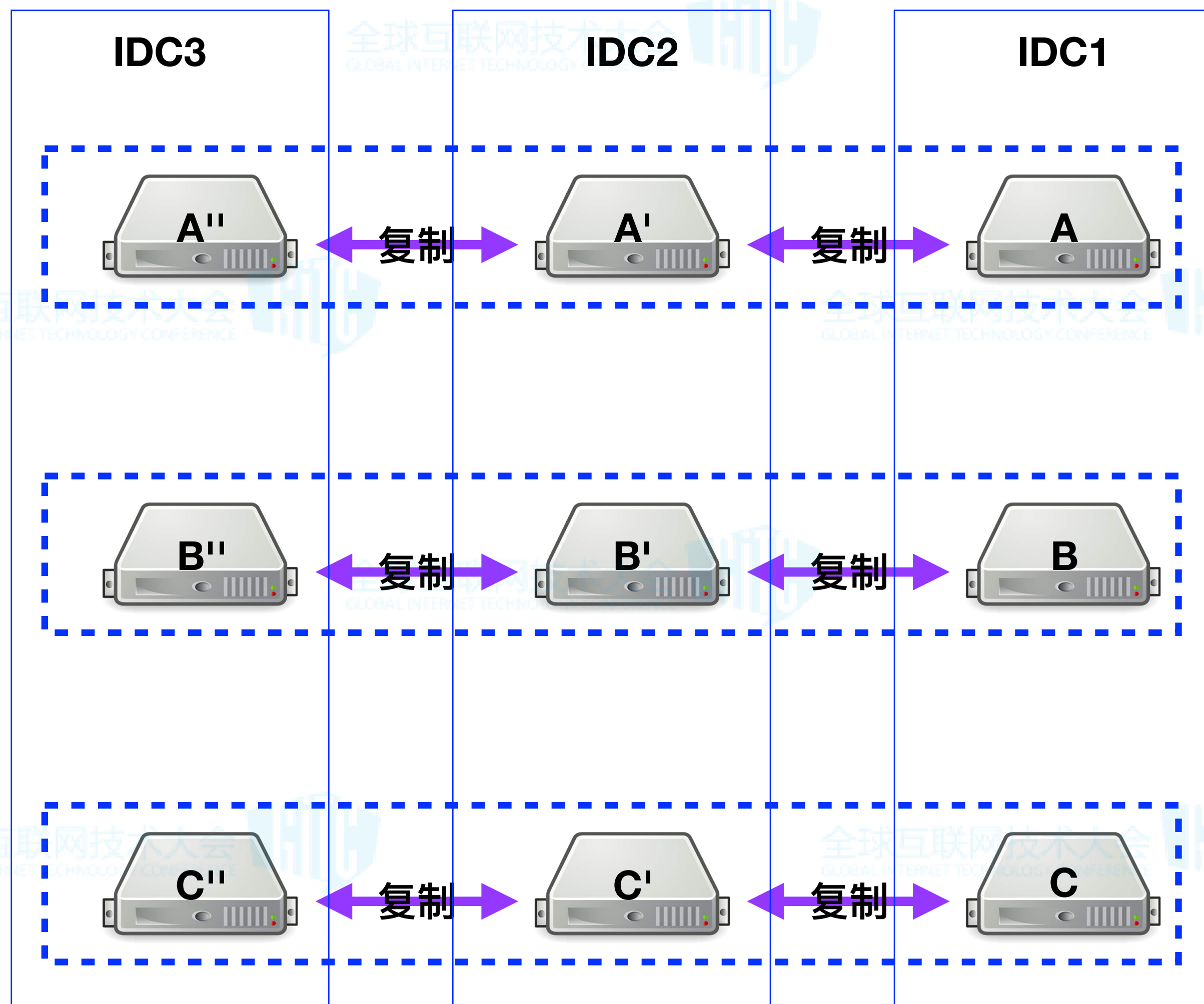
每个IDC 3个节点，做分布式表，分担查询压力

如：

A/B/C 分别是数据的3个分片，各自承担1/3的数据

查询性能：A+B+C

部署：最佳架构



宕机分析

1. 3个IDC，挂掉2个
不影响读写
恢复只需重启实例
ClickHouse自动完成数据同步
2. 挂掉某个实例，如A
切换读写到其他IDC，恢复实例A即可
3. 挂掉多个实例，A、B同时挂掉
处理方式同上

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



集群的ClickHouse有多快？

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



我们的架构

CPU E5-2620 @ 2.00GHz
开启超线程后24core

48G内存

3T*12 Raid5

X 4

表1:
14字段 900亿 4T

表2:
94字段 500亿 15T

请开始你的表演：

```
;) select count(*) from apm_msg_all;  
SELECT count(*)  
FROM apm_msg_all  
  
count(*)  
| 30249469866 |  
  
1 rows in set. Elapsed: 0.927 sec. Processed 30.25 billion rows, 30.25 GB (32.63 billion rows/s., 32.63 GB/s.)  
;) |
```

select count(*)

300亿

0.9秒

请开始你的表演：

```
:) select date, count(*)/100000000 from apm_msg_all group by date order by date desc limit 10 ;
```

```
SELECT  
  date,  
  count(*) / 100000000  
FROM apm_msg_all  
GROUP BY date  
ORDER BY date DESC  
LIMIT 10
```

date	divide(count(), 100000000)
2017-10-09	2.95357962
2017-10-08	5.25704122
2017-10-07	5.02549947
2017-10-06	4.92416848
2017-10-05	4.98019803
2017-10-04	4.68522239
2017-10-03	4.81681714
2017-10-02	4.95085341
2017-10-01	4.78589495
2017-09-30	4.90037846

```
10 rows in set. Elapsed: 9.872 sec. Processed 30.27 billion rows, 60.53 GB (3.07 billion rows/s., 6.13 GB/s.)
```

select date, count(*) from xx
group by date

300亿group by日期

9.8秒

请开始你的表演：

```
        _request_url,
        count(1) AS hit_2hour
    FROM apm.apm_msg_all
    WHERE (date = toString(today())) AND (hour = (toInt8(substring(toString(now()), 12, 2)) - 2)) AND (_ek = '_error') AND (url_type = 'PIC') AND (_http_code IN ('403', '404'))
    GROUP BY _request_url
)
ALL INNER JOIN
(
    SELECT
        _request_url,
        round(MAX(hit) / 60, 2) AS maxhit_per_s,
        sum(hit) AS hit_hour
    FROM
    (
        SELECT
            _request_url,
            minute,
            COUNT(1) AS hit
        FROM apm.apm_msg_all
        WHERE (date = toString(today())) AND (hour = (toInt8(substring(toString(now()), 12, 2)) - 1)) AND (_ek = '_error') AND (url_type = 'PIC') AND (_http_code IN ('403', '404'))
        GROUP BY
            _request_url,
            minute
    ) AS a
    GROUP BY _request_url
) USING (_request_url)
GROUP BY
    _request_url,
    maxhit_per_s,
    hit_2hour,
    hit_hour
USING (_request_url)

Progress: 3.42 million rows, 213.70 MB (28.50 million rows/s., 1.78 GB/s.) 21%
Progress: 79.78 million rows, 6.18 GB (97.48 million rows/s., 7.55 GB/s.) 92%
```

太复杂了



10 rows in set. Elapsed: 1.140 sec. Processed 114.07 million rows, 9.46 GB (100.06 million rows/s., 8.29 GB/s.)

请开始你的表演：

```
SELECT
  hour,
  one_hit,
  two_hit,
  one_error_hit,
  two_error_hit,
  round((one_error_hit / ((one_hit / 0.3) + one_error_hit)) * 100, 2) AS one_error_rate,
  round((two_error_hit / ((two_hit / 0.3) + two_error_hit)) * 100, 2) AS two_error_rate
FROM
(
  SELECT
    hour,
    sum(caseWithoutExpr((area IN ('北京', '上海')) OR (city_name IN ('深圳', '广州')), 1, 0)) AS one_hit,
    sum(caseWithoutExpr((area NOT IN ('北京', '上海')) AND (city_name NOT IN ('深圳', '广州')), 1, 0)) AS two_hit
  FROM apm.apm_msg_all
  WHERE (date = toString(yesterday())) AND (_ek = '_network') AND (_snet != 'wifi') AND (country = '中国')
  GROUP BY hour
)
ANY LEFT JOIN
(
  SELECT
    hour,
    sum(caseWithoutExpr((area IN ('北京', '上海')) OR (city_name IN ('深圳', '广州')), 1, 0)) AS one_error_hit,
    sum(caseWithoutExpr((area NOT IN ('北京', '上海')) AND (city_name NOT IN ('深圳', '广州')), 1, 0)) AS two_error_hit
  FROM apm.apm_msg_all
  WHERE (date = toString(yesterday())) AND (_ek = '_error') AND (_snet != 'wifi') AND (country = '中国')
  GROUP BY hour
) USING (hour)
Progress: 558.25 million rows, 42.98 GB (102.60 million rows/s., 7.90 GB/s.) 89%
```

更复杂了

支持Join



24 rows in set. Elapsed: 14.543 sec. Processed 1.23 billion rows, 94.76 GB (84.60 million rows/s., 6.52 GB/s.)

请开始你的表演：

官方demo

```
SELECT  
  count() as visits,  
  sum(PageViews) as hits,  
  uniq(UserID) as users  
FROM visits_all  
WHERE StartDate > today() - 7;  
  
SELECT  
  count() AS visits,  
  sum(PageViews) AS hits,  
  uniq(UserID) AS users  
FROM visits_all  
WHERE StartDate > (today() - 7)
```

visits	hits	users
12106577368	46414744272	1080418023

1 rows in set. Elapsed: 6.245 sec. Processed 13.23 billion rows, 185.18 GB (2.12 billion rows/s., 29.65 GB/s.)

```
SELECT  
  uniq(UserID) AS users,  
  if(IsMobile, if(IsTablet, 'tablet', 'phone'), 'desktop') AS device  
FROM visits_all  
WHERE StartDate > (today() - 7)  
GROUP BY device
```

users	device
465363568	phone
548496870	desktop
70608335	tablet

3 rows in set. Elapsed: 7.650 sec. Processed 13.23 billion rows, 158.71 GB (1.73 billion rows/s., 20.75 GB/s.)



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



快

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



快

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



为啥这么快?

1. 优秀的代码编写，强大的底层优化，严格的单元测试
2. A vector engine & Code generation
3. CPU底层指令集的使用
4. C++新特性

为啥这么快?

Vectorized processing

Data is represented as small single-dimensional arrays (vectors), easily accessible for CPUs.

The percentage of instructions spent in interpretation logic is reduced by a factor equal to the vector-size

The functions that perform work now typically process an array of values in a tight loop

Tight loops can be optimized well by compilers, enable compilers to generate SIMD instructions automatically.

Modern CPUs also do well on such loops, out-of-order execution in CPUs often takes multiple loop iterations into execution concurrently, exploiting the deeply pipelined resources of modern CPUs.

It was shown that vectorized execution can improve data-intensive (OLAP) queries by a factor 50.

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

ClickHouse函数：

3000+

ClickHouse高级函数：

■ 统计类

`quantile(level)(x)`

`quantiles(level1, level2, ...)(x)`

`median(X)`

■ URL截取

`cutQueryString(X)`

`domain(X)`

■ 其他

`today()-1`

`yesterday()`

`substring(s, offset, length)`

`IPv4NumToStringClassC`

`extract(haystack, pattern)`

高级函数举例：

■ 域名类

```
SELECT protocol('https://weibo.com/jackpgao/home?wvr=5') as protocol;
```

```
┌protocol┐  
| https |  
└────────┘
```

```
SELECT domain('https://weibo.com/jackpgao/home?wvr=5') as domain;
```

```
┌domain──┐  
| weibo.com |  
└────────┘
```

```
SELECT domainWithoutWWW('https://weibo.com/jackpgao/home?wvr=5') as domainWithoutWWW;
```

```
┌domainWithoutWWW┐  
| weibo.com |  
└────────────────┘
```

```
SELECT cutQueryString('https://weibo.com/jackpgao/home?wvr=5') as cutQueryString;
```

```
┌cutQueryString──┐  
| https://weibo.com/jackpgao/home |  
└────────────────┘
```

```
SELECT path('https://weibo.com/jackpgao/home?wvr=5') as path;
```

```
┌path──┐  
| /jackpgao/home |  
└────────┘
```

```
SELECT pathFull('https://weibo.com/jackpgao/home?wvr=5') as pathFull;
```

```
┌pathFull──┐  
| /jackpgao/home?wvr=5 |  
└──────────┘
```


高级函数举例：

IP归类

```
SELECT
  IPv4NumToStringClassC(IPv4StringToNum(clientip)) AS k,
  count()
FROM xxx
WHERE (date = toString(today())) AND (hour = 12)
GROUP BY k
ORDER BY count() DESC
LIMIT 10
```

k	count()
118.190.218.xxx	99353
59.63.248.xxx	64207
122.224.52.xxx	49156
117.136.79.xxx	47350
120.241.4.xxx	47041
59.63.249.xxx	45717
115.124.31.xxx	40806
111.13.31.xxx	38786
223.104.3.xxx	37271
0.0.0.xxx	36596

来点干货

目录结构

```
├─ config-preprocessed.xml
├─ config.xml
├─ users-preprocessed.xml
├─ users.xml
├─ metrika.xml
```

配置文件

```
├─ cores
├─ flags
├─ status
```

状态文件

```
├─ tmp
```

```
├─ log
├─ │ error.log
├─ │ server.log
├─ │ stderr
├─ │ stdout
```

日志

```
├─ metadata
├─ │ apm.sql
├─ │ default
├─ │ gaopeng4
├─ │ │ apm_msg.sql
├─ │ system
```

元数据

```
├─ data
├─ │ default
├─ │ gaopeng4
├─ │ │ apm_msg
├─ │ │ │ 20171001_20171001_0_9_2
├─ │ │ │ │ checksums.txt
├─ │ │ │ │ columns.txt
├─ │ │ │ │ date.bin
├─ │ │ │ │ date.mrk
├─ │ │ │ │ minute.bin
├─ │ │ │ │ minute.mrk
├─ │ │ │ │ primary.idx
├─ │ │ │ detached
├─ │ system
```

数据目录

配置文件：

config.xml

```
<?xml version="1.0"?>
<yandex>
  <logger>
    <level>trace</level>
    <log>/data1/clickhouse/log/server.log</log>
    <errorlog>/data1/clickhouse/log/error.log</errorlog>
    <size>1000M</size>
    <count>10</count>
  </logger>
  <http_port>8123</http_port>
  <tcp_port>9000</tcp_port>
  <interserver_http_port>9009</interserver_http_port>
  <interserver_http_host>ck21. [REDACTED].com.cn</interserver_http_host>
  <listen_host>0.0.0.0</listen_host>
  <max_connections>4096</max_connections>
  <keep_alive_timeout>3</keep_alive_timeout>
  <max_concurrent_queries>100</max_concurrent_queries>
  <uncompressed_cache_size>8589934592</uncompressed_cache_size>
  <mark_cache_size>10737418240</mark_cache_size>
  <path>/data1/clickhouse/</path>
  <tmp_path>/data1/clickhouse/tmp/</tmp_path>
  <users_config>users.xml</users_config>
  <default_profile>default</default_profile>
  <log_queries>1</log_queries>
  <default_database>default</default_database>
  <remote_servers incl="clickhouse_remote_servers" />
  <zookeeper incl="zookeeper-servers" optional="true" />
  <macros incl="macros" optional="true" />
  <builtin_dictionaries_reload_interval>3600</builtin_dictionaries_reload_interval>
  <max_table_size_to_drop>0</max_table_size_to_drop>
  <include_from>/data1/clickhouse/metrika.xml</include_from>
</yandex>
```

日志

本节点信息
本机域名

本地配置

集群相关配置

配置文件：

metrika.xml

```
<yandex>
<clickhouse_remote_servers>
  <bip_ck_cluster>
    <shard>
      <internal_replication>true</internal_replication>
      <replica>
        <host>ck11[REDACTED].com.cn</host>
        <port>9000</port>
      </replica>
    </shard>
    <shard>
      <replica>
        <internal_replication>true</internal_replication>
        <host>ck12[REDACTED].com.cn</host>
        <port>9000</port>
      </replica>
    </shard>
    <shard>
      <internal_replication>true</internal_replication>
      <replica>
        <host>ck13[REDACTED].com.cn</host>
        <port>9000</port>
      </replica>
    </shard>
    <shard>
      <internal_replication>true</internal_replication>
      <replica>
        <host>ck14[REDACTED].com.cn</host>
        <port>9000</port>
      </replica>
    </shard>
  </bip_ck_cluster>
</clickhouse_remote_servers>
```

配置文件：

metrika.xml

```
<zookeeper-servers>
  <node index="1">
    <host>1. [REDACTED].sina.com.cn</host>
    <port>2181</port>
  </node>
  <node index="2">
    <host>2. [REDACTED].sina.com.cn</host>
    <port>2181</port>
  </node>
  <node index="3">
    <host>3. [REDACTED].sina.com.cn</host>
    <port>2181</port>
  </node>
</zookeeper-servers>
```

```
<macros>
  <replica>ck11</replica>
</macros>
```

```
<networks>
  <ip>::/0</ip>
</networks>
```

```
<clickhouse_compression>
<case>
  <min_part_size>10000000000</min_part_size>
  <min_part_size_ratio>0.01</min_part_size_ratio>
  <method>lz4</method>
</case>
</clickhouse_compression>
</yandex>
```

配置文件：

user.xml

```
<profiles>
  <default>
    <max_memory_usage>10000000000</max_memory_usage>
    <use_uncompressed_cache>0</use_uncompressed_cache>
    <load_balancing>random</load_balancing>
  </default>
  <readonly>
    <max_memory_usage>10000000000</max_memory_usage>
    <use_uncompressed_cache>0</use_uncompressed_cache>
    <load_balancing>random</load_balancing>
    <readonly>1</readonly>
  </readonly>
</profiles>

<quotas>
  <!-- Name of quota. -->
  <default>
    <interval>
      <duration>3600</duration>
      <queries>0</queries>
      <errors>0</errors>
      <result_rows>0</result_rows>
      <read_rows>0</read_rows>
      <execution_time>0</execution_time>
    </interval>
  </default>
</quotas>
```

配置文件：

user.xml

```
<users>
  <default>
    <password_sha256_hex>967f3bf355dddafbfca1c9f5cab39352b2ec1cd0b05f9e1e6b8f629705fe7d6e</password_sha256_hex>
    <networks incl="networks" replace="replace">
      <ip>::/0</ip>
    </networks>
    <profile>default</profile>
    <quota>default</quota>
  </default>
  <ck>
    <password_sha256_hex>967f3bf355dddafbfca1c9f5cab39352b2ec1cd0b05f9e1e6b8f629705fe7d6e</password_sha256_hex>
    <networks incl="networks" replace="replace">
      <ip>::/0</ip>
    </networks>
    <profile>readonly</profile>
    <quota>default</quota>
  </ck>
</users>
```

```
PASSWORD=$(base64 < /dev/urandom | head -c8);
echo "$PASSWORD"; echo -n "$PASSWORD" | sha256sum | tr -d '-'
```

ClickHouse的问题：

1. insert into xx (a, b, c) values ('a', 'b', 'c') 只能是**单引号**

2. 如果是int插入的是string，报错（不确定是否有类似SQL_MODE的参数）

3. 删除只支持到**月纬度**的分区

4. 改造官方的启动脚本，**不要用root**直接启动

谁在用

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



Yandex Metrika

Features

Resources

Pricing

Partners

Get in touch

En ▾

Sign up

Log in

All-Round Web Analytics

From traffic trends to mouse movements – get a comprehensive understanding of your online audience and drive business growth.

Get started

Try live demo



谁在用

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



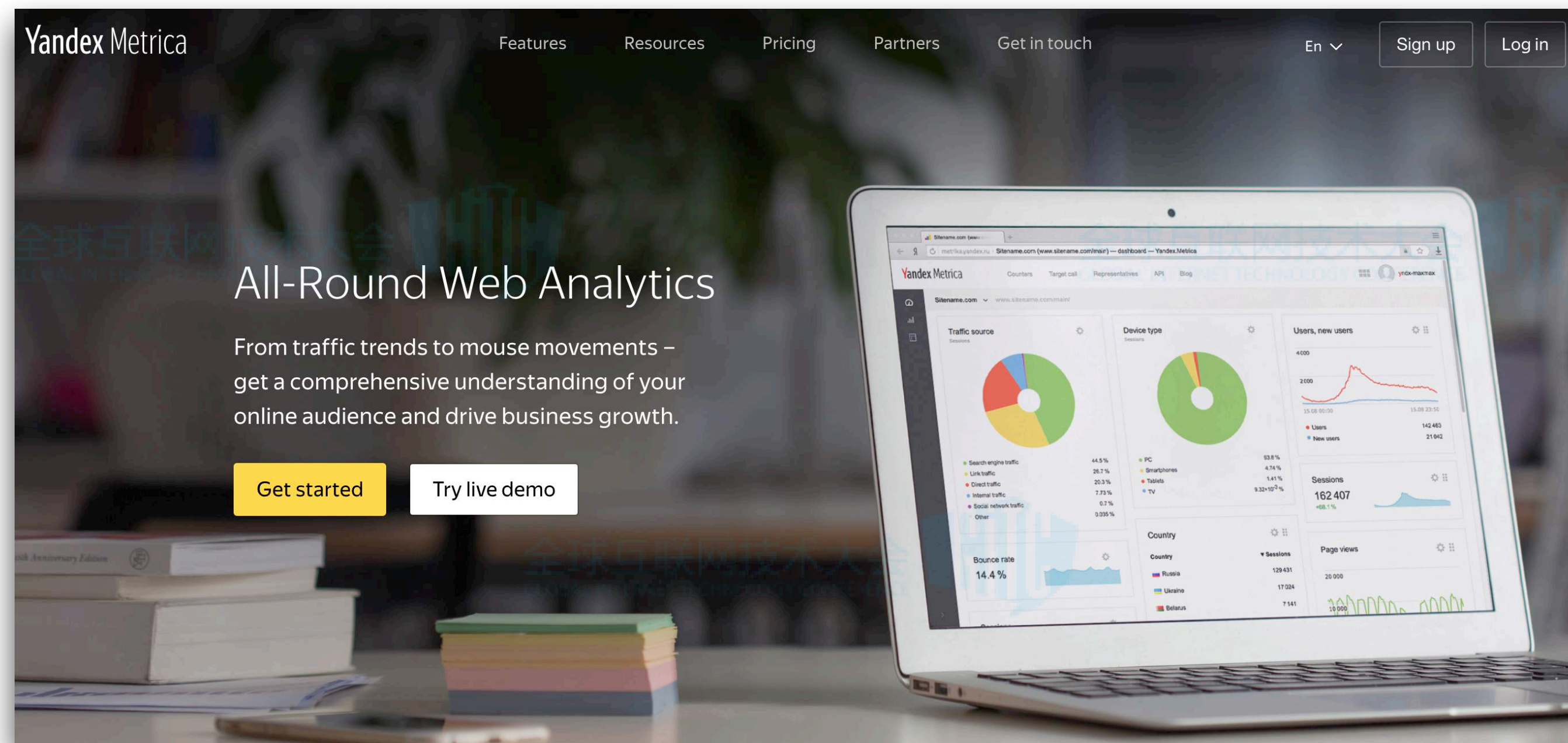
全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



374 servers

store over 20.3 trillion rows

17 PB uncompressed data TSV

2 PB without counting duplication and replication

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



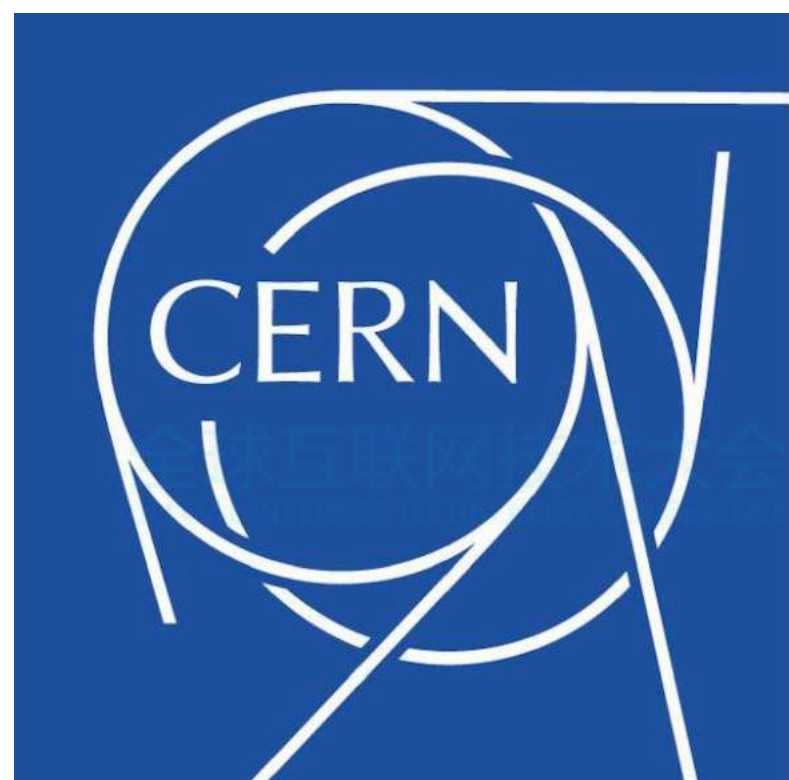
全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



谁在用



欧洲原子能研究机构

store and process
metadata on 10 billion
events with over
1000 attributes per event



Tinkoff Bank

Innovative provider of online retail
financial services in Russia



CLOUDFLARE

《How Cloudflare analyzes **1M** DNS queries per second》

Multi-tenant ClickHouse cluster

33

Nodes

8M+

Row Insertion/s

4GB+

Insertion Throughput/s

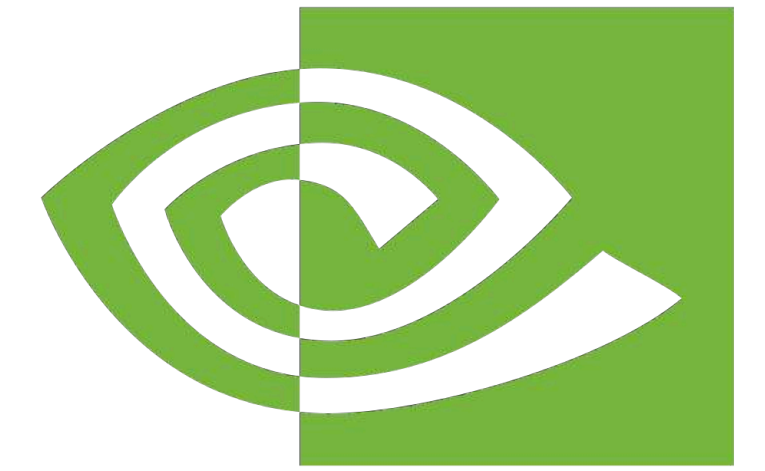
2PB+

Raid-0 Spinning Disks

谁在用



- 做存储的公司
- Python驱动: [infi.clickhouse_orm](https://github.com/infinidat/infi.clickhouse_orm)



NVIDIA

CARTO

- **CARTO** (formerly CartoDB) is a Software as a Service (SaaS) cloud computing platform that provides GIS and web mapping tools for display in a web browser.

谁在用

- **Altinity is the leading service provider for ClickHouse**
- **Altinity Provides Cloud Version of ClickHouse on Kodiak Data MemCloud™**
- **Kodiak Data is an edge-cloud infrastructure company that allows customers to easily deploy and scale big data infrastructure in both public and private clouds**



Altinity

谁在用

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



LAUNCH ANALYTICS TO **A NEW LEVEL** WITH CLICKHOUSE

FUNCTIONAL DEMOS | HANDS-ON PRIVATE DEMOS | POC | BENCHMARKS | DATASET DEMOS

LEARN MORE



谁在用

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



PMM Roadmap

- Alerting
- QAN for MongoDB
- MySQL -> **ClickHouse** for QAN datastore
- Plugins and Integrations
- Long term metrics storage (past 30 days)
- One-click ticket submission*
- Standardised data collection for tickets*

Any feedback of what you'd like to see in PMM?

* for Percona Subscribers (Customers) only



PERCONA
Monitoring and Management



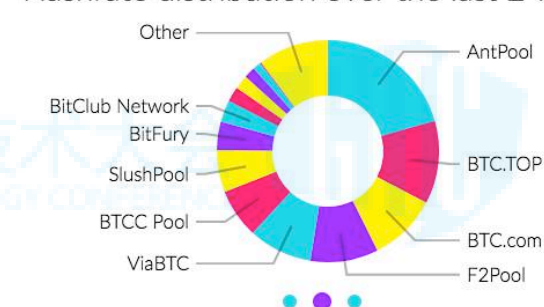
BLOCKCHAIR

Search Bitcoin and Bitcoin Cash blockchains for anything...

Examples: [Hello World](#) [The biggest blocks](#) [~\\$1M transactions](#) [1dice8EMZmqKvrGE4Qc9bUff9PX3xaYDp](#) [satoshi.id](#)
[Blocks mined by Slush](#) [Tx paid over \\$5 in fees](#) [Addresses starting with 1dice](#) [Marry me](#) [BIP 100 blocks](#)
[Tx destroyed the most coindays](#) [Mempool transactions](#) [Happy Birthday](#) [Blocks collected over 2.5 BTC in fees](#) [Bailout](#)
[Coinbase transactions](#) [123456](#)

Bitcoin / Blocks

Hashrate distribution over the last 2 weeks

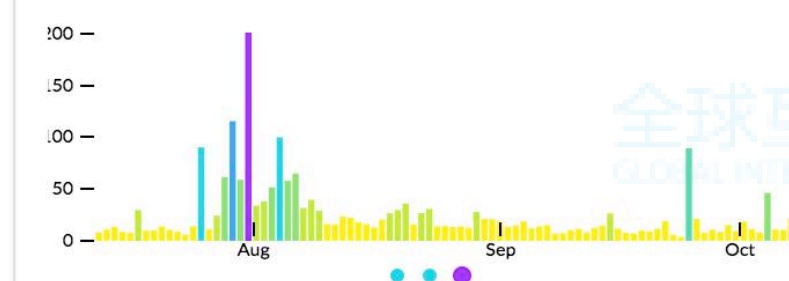


Useful links:

- Empty blocks
- SegWit blocks / Unlimited blocks

Bitcoin / Transactions

Coindays destroyed (millions) by day over last 3 mos.

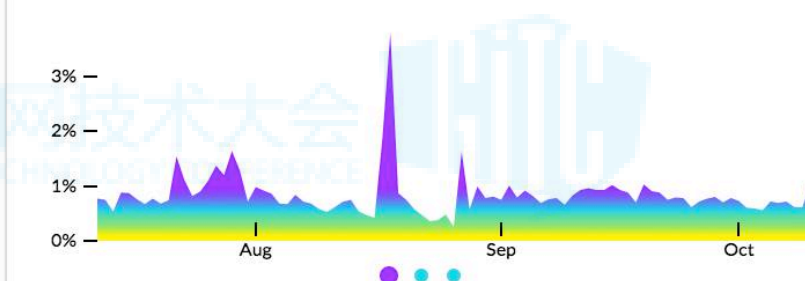


Useful links:

- Coinbase transactions
- Transactions which destroyed 0 coindays

Bitcoin / Outputs

Percentage of non-monetary outputs over last 3 mos.



Useful links:

- UTXO set (unspent outputs)
- OP_RETURN (nulldata) outputs

Blockchair is a blockchain search and analytics engine or you can also say it's a blockchain explorer on steroids.

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



谁在用

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

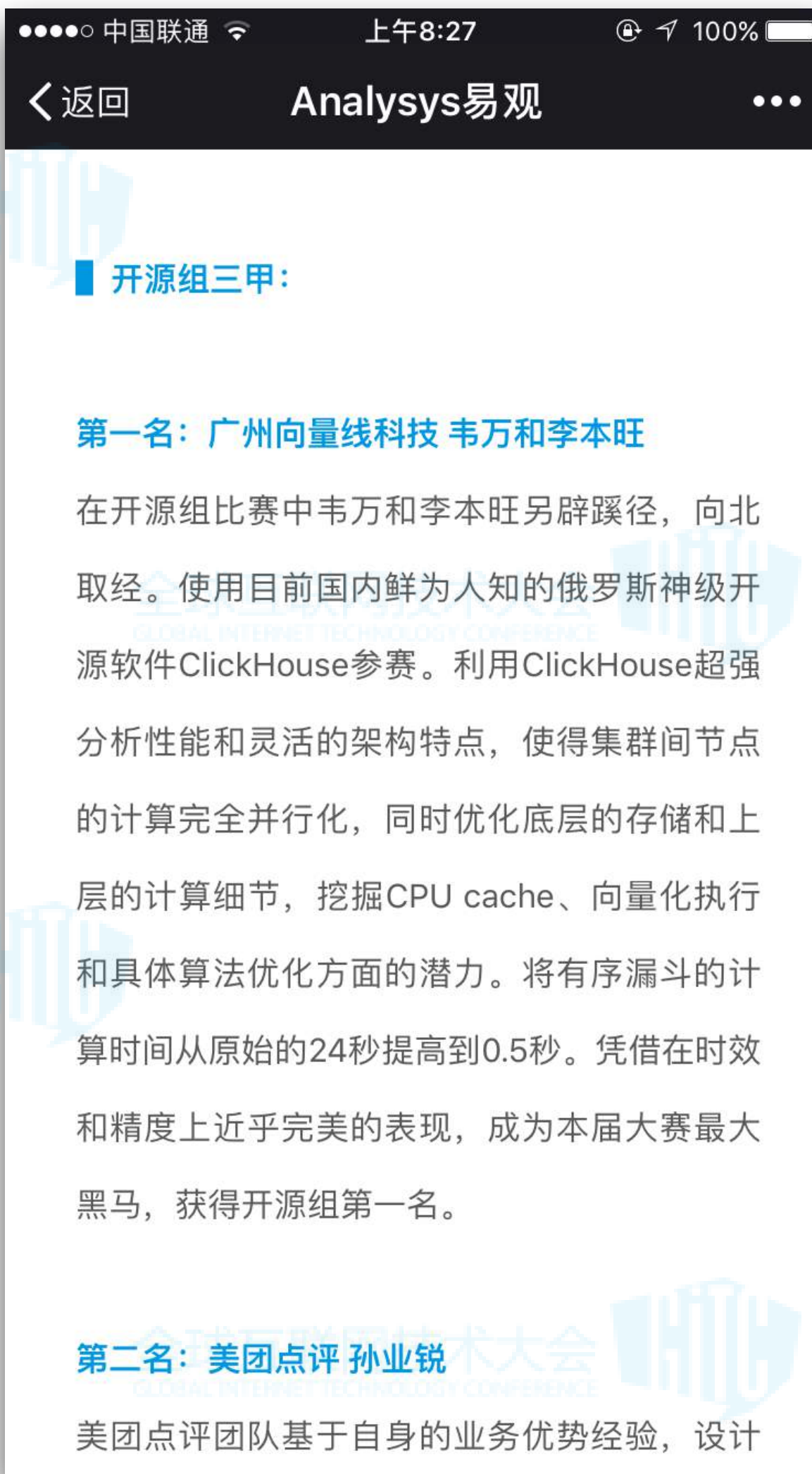


全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



我浪怎么用？

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



搭配

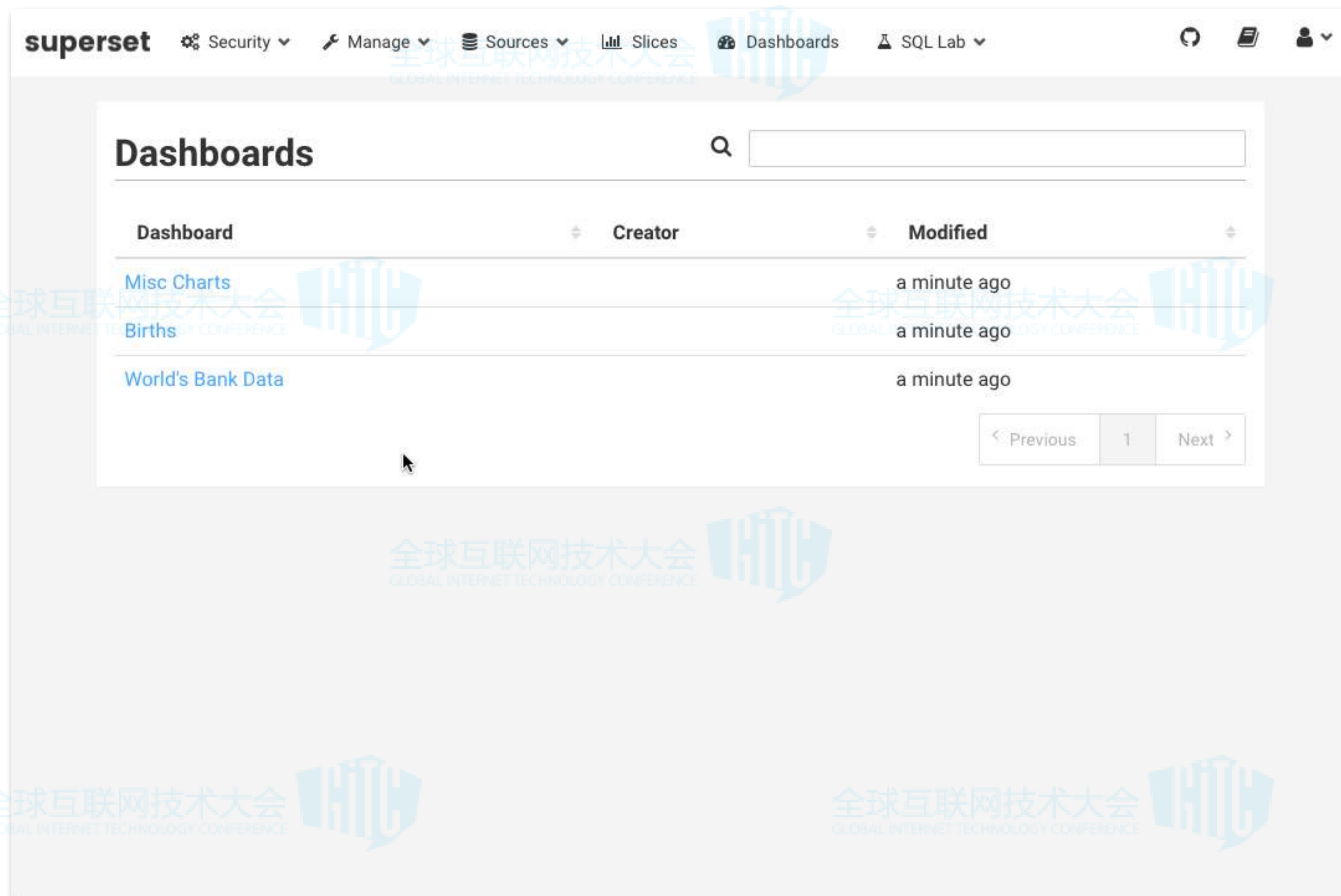
- Superset

- Airbnb开源Python项目

- Apache孵化

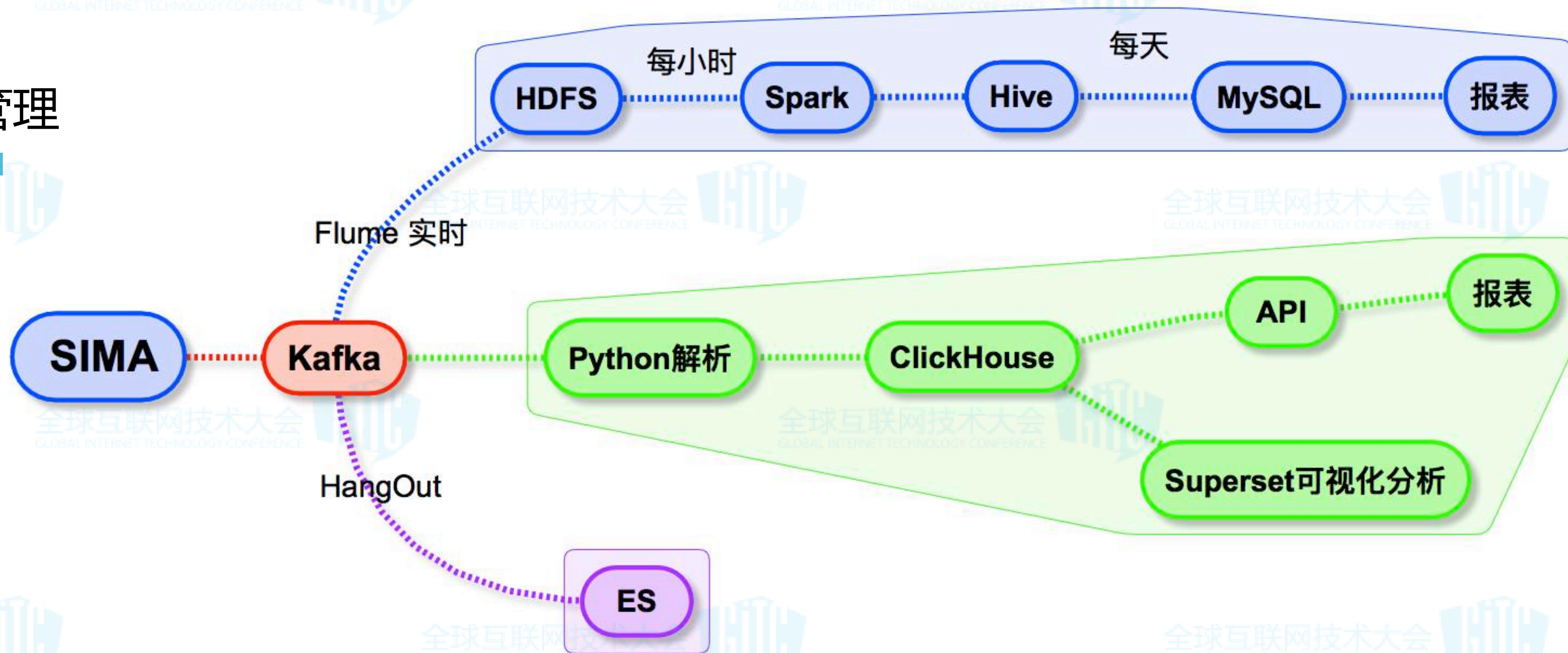
- 快速实现可视化

- 完美对接ClickHouse ,
方便分析师进行问题排查与分析

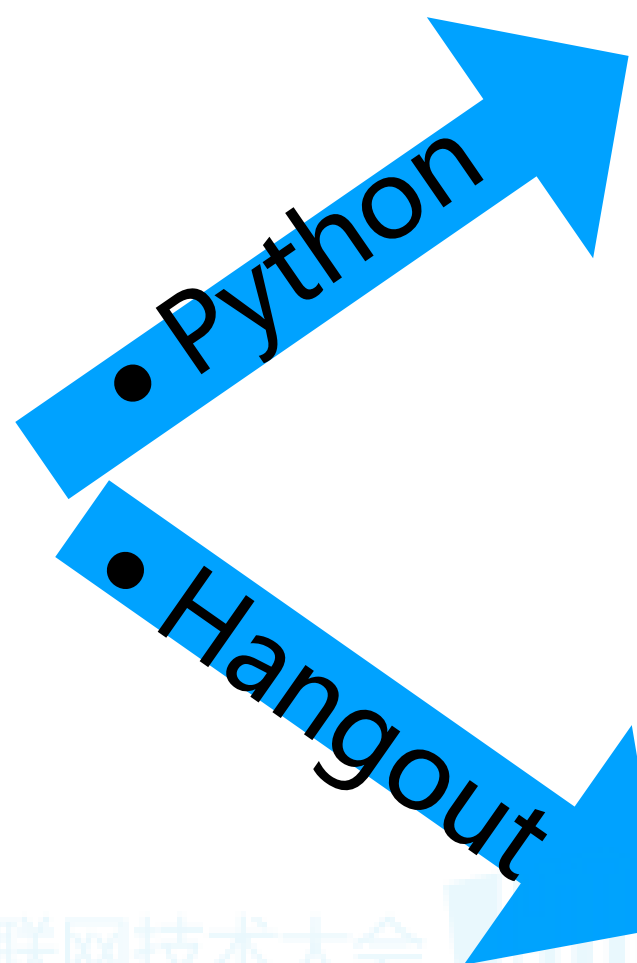


案例

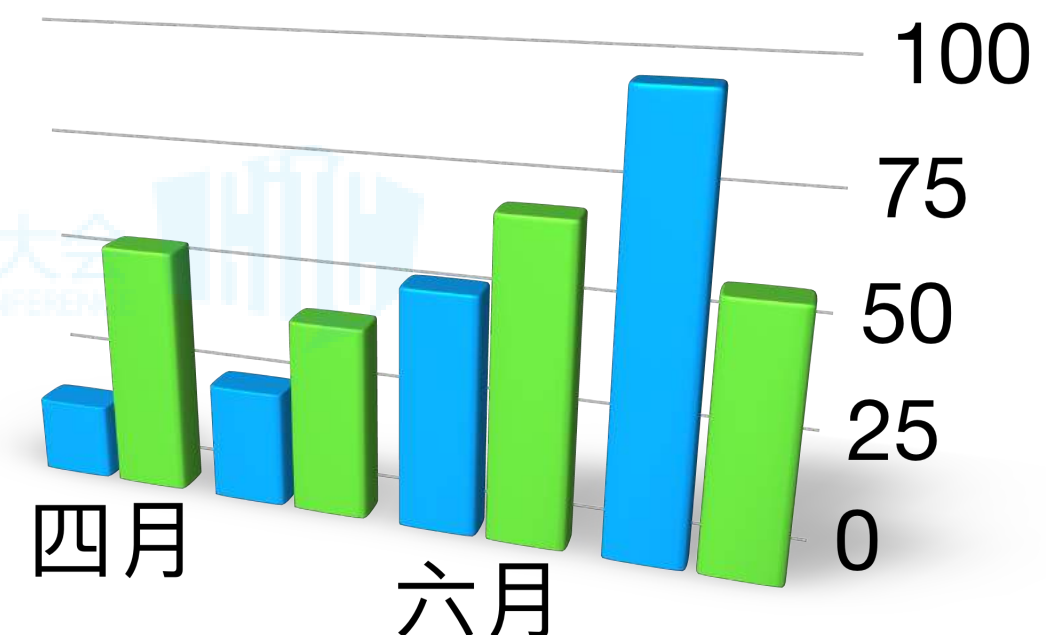
- 某APP APM管理



案例



• ClickHouse



- 图表、邮件、报警等
- Superset可视化工作台



• ES

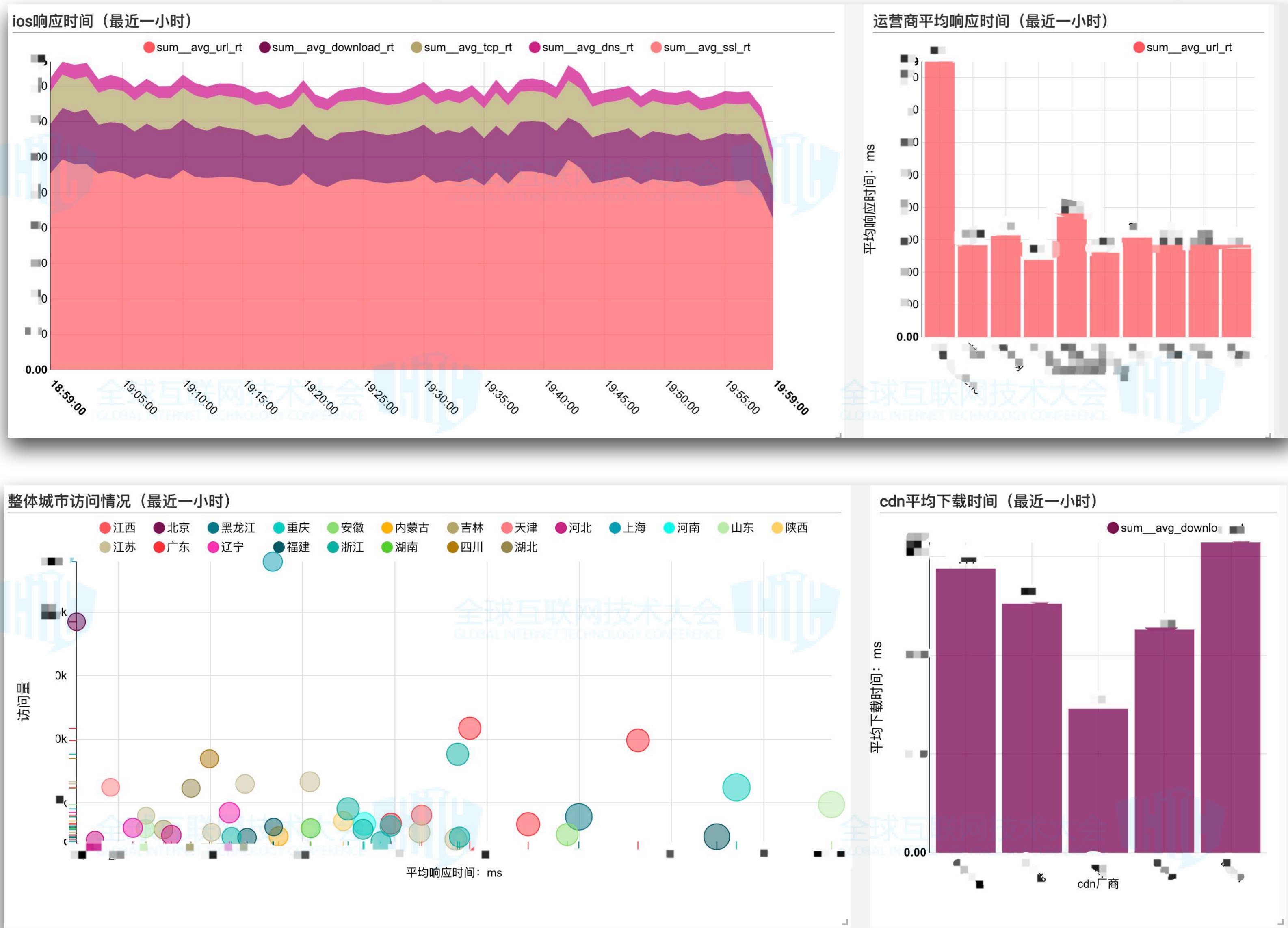


- 问题排查, Trace

- 数据处理链路短
- 数据实时可见, 及时Trace
- 如何快速数据变现

案例

● 某APP APM管理

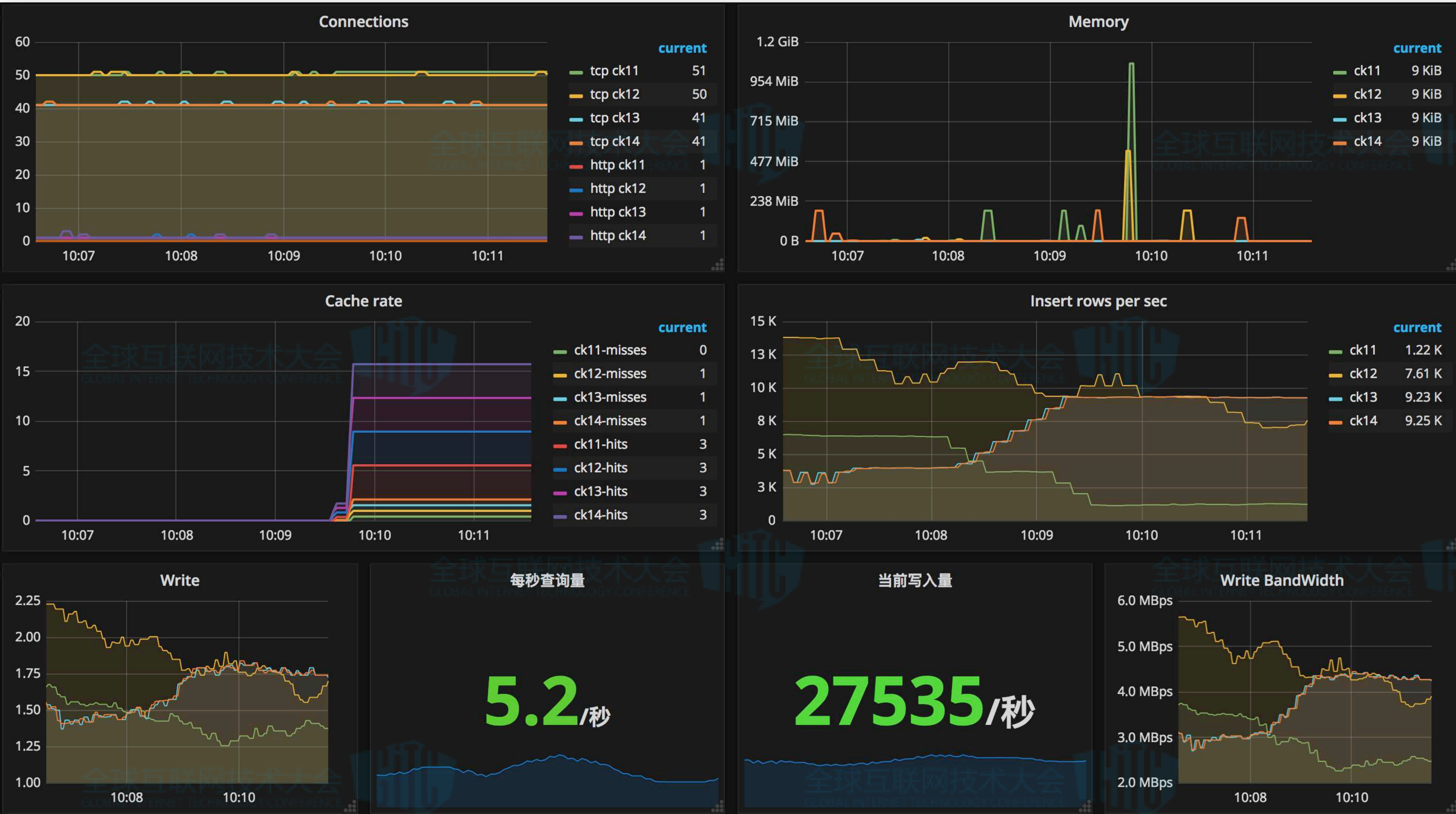


CK监控

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



Prometheus
+
ClickHouse exporter
+
Grafana

最佳实践

运维

1. CPU: 多核优于主频

(SSE 4.2 instruction set need)

开启超线程、性能模式

2. 小磁盘多机器 > 大磁盘少机器

3. Raid-10 is better

If Raid5/6/50, increase stripe_cache_size

4. 内存: 越大越好, 留给page cache

禁用swap/透明大页/NUMA

5. CentOS7/Ext4/复制带宽问题

使用

1. batch insert 2K 起步

2. 用域名写本地表, 读分布式表

3. 如果是Docker, 注意修改时区

4. clickhouse-client在Docker里, 中文乱码

5. 拒绝 `select *`

6. 无Decimal, 乘以倍率, 用Uint64存

总结

■ 使用场景

大容量

结构化

SQL方式日志分析

广告

IOT

监控

■ 对比

MySQL：复杂查询、容量，重点不在事务

HDFS生态：简单、易用、查询快，规模稳定性有待验证

Druid：保留原始数据，防止预先设定不满足需求



大PK



对比

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网
GLOBAL INTERNET

对比项

ES

ClickHouse

数据接入

插件丰富：hangout/logstash/filebeats
无需特别指明字段类型，兼容性好

需要自己开发：JDBC、clickhouse-client、Python导入

[Hangout to ClickHouse插件](#)

查询

原生查询方式不灵活
SQL插件复杂度有限
大范围查询性能差

支持复杂查询
高级函数多
支持连表查询

扩展性

旗鼓相当，ES略轻松，ClickHouse需要管理表维度

其他

原生API
社区健壮

原生API
社区正在发展

语言

Java/Python

Java/Python/R/PHP

对比ES

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

对比



- GPU Database for Fast, Interactive Visual Analytics

kinetica

- GPU-accelerated analytics database for real-time insights on large and streaming datasets



BLAZINGDB

- High Performance GPU Database for Big Data SQL

brytlyt

- World's most **advanced** GPU based PostgreSQL Database

GPU产品

<http://tech.marksblogg.com/benchmarks.html>

对比



74x to 3,500x faster than CPU DBs.

The table is sorted by the fastest time query 1 finished in (measured in seconds).

Query 1 Query 2 Query 3 Query 4 Setup

0.021	0.053	0.165	0.51	<u>MapD & 8 Nvidia Pascal Titan Xs</u>
0.027	0.083	0.163	0.891	<u>MapD & 8 Nvidia Tesla K80s</u>
0.028	0.2	0.237	0.578	<u>MapD & 4-node g2.8xlarge cluster</u>
0.036	0.131	0.439	0.964	<u>MapD & 4 Nvidia Titan Xs</u>
0.051	0.146	0.047	0.794	<u>kdb+/q & 4 Intel Xeon Phi 7210 CPUs</u>
1.034	3.058	5.354	12.748	<u>ClickHouse, Intel Core i5 4670K</u>
1.56	1.25	2.25	2.97	<u>Redshift, 6-node ds2.8xlarge cluster</u>
2	2	1	3	<u>BigQuery</u>
4	4	10	21	<u>Presto, 50-node n1-standard-4 cluster</u>
6.41	6.19	6.09	6.63	<u>Amazon Athena</u>
8.1	18.18	n/a	n/a	<u>Elasticsearch (heavily tuned)</u>
10.19	8.134	19.624	85.942	<u>Spark 2.1, 11 x m3.xlarge cluster w/ HDFS</u>
11	10	21	31	<u>Presto, 10-node n1-standard-4 cluster</u>
14.389	32.148	33.448	67.312	<u>Vertica, Intel Core i5 4670K</u>
34.48	63.3	n/a	b/a	<u>Elasticsearch (lightly tuned)</u>
35	39	64	81	<u>Presto, 5-node m3.xlarge cluster w/ HDFS</u>
43	45	27	44	<u>Presto, 50-node m3.xlarge cluster w/ S3</u>
152	175	235	368	<u>PostgreSQL 9.5 & cstore_fdw</u>
264	313	620	961	<u>Spark 1.6, 5-node m3.xlarge cluster w/ S3</u>

对比

- **PG-Strom is an extension designed for PostgreSQL v9.5 or later, to off-load a part of CPU intensive workloads to GPU (Graphic Processor Unit) devices, and execute them in parallel asynchronously.**



对比

Commercial solutions – fast and expensive

- ◇ Vertica
- ◇ RedShift
- ◇ Teradata
- ◇ Etc
- ◇ **The cost scales with your data**

商业产品

Open Source:
somewhat slow,
sometime buggy.
But free

开源产品

- ◇ InfiniDB (now MariaDB ColumnStore)
- ◇ InfoBright
- ◇ GreenPlum (started as commercial)
- ◇ Hadoop systems
- ◇ Apache Spark

对比



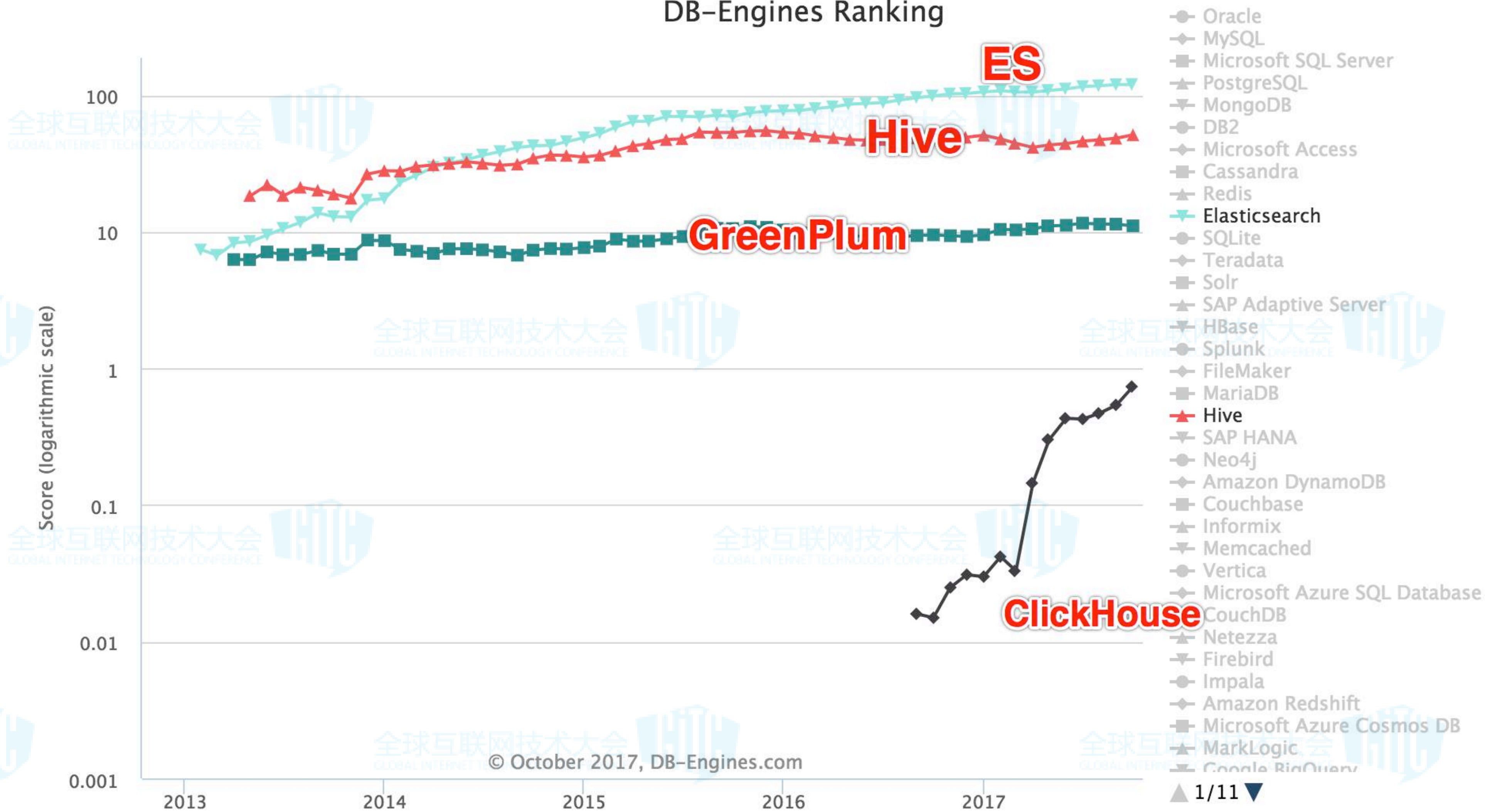
性能与成本的均衡

对比

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

DB-Engines Ranking



行业排名

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



结缘，



数据库工程师

“那年我还是个DBA，饱受业务复杂查询，也就是OLAP之苦”



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



一个
神奇的网站 :)



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



<https://www.percona.com/>

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE





不怕有坑?





不**试试**



怎么知道好用不好用



Summary

Try it now!

大容量结构化的数据

需要SQL

快速实现聚合、可视化





- But, 如果,
- 不好用, 别撕我~



资源推荐：



JackpGao



gaopeng4@staff.sina.com.cn

1. [官方文档](#) ★★★ 结构清晰，文档清楚，但是例子太少
2. [Percona ClickHouse Blog](#) ★★★★★
3. [github issue](#) & [ClickHouse Google Group](#)
4. [ClickHouse服务提供商Altinity](#) ★★★★★
5. [官方Meetup PPT](#) ★★★★★
6. [Altinity提供的rpm包](#) ★★★★★
7. [官方运维建议](#)
8. 个人推荐PPT合集：[百度网盘](#)，密码yv72