Gdevops

Global DevOps Summit

全球敏捷运维峰会

基于ClickHouse+StarRocks
构建支撑千亿级数据量的高可用查询引擎

演讲人：蔡岳毅

1. 为什么选择ClickHouse/StarRocks；

2. ClickHouse/StarRocks的高可用架构；

3. 如何合理的应用ClickHouse的优点，StarRocks 如何来补充ClickHouse 的短板；

4. ClickHouse的调优，运维介绍；

5. 应用总结；

根据实际业务场景需要来选择

1. 不固定的查询条件，不固定的汇总条件；
2. 数据量日益增量，每天要更新的数据量也不断增大；
3. 业务场景不断增多，涉及面越来越广；
4. 需要保证高可用并秒出；
5. 从Sql，Es， CrateDB， Kylin，Ingite，MongoDB，Hbase 不断的研究，实践；

ClickHouse 的特点

优点：

1. 数据压缩比高，存储成本相对非常低；
2. 支持常用的SQL语法，写入速度非常快，适用于大量的数据更新；
3. 依赖稀疏索引，列式存储，cpu/内存的充分利用造就了优秀的计算能力，并且不用考虑左侧原则；

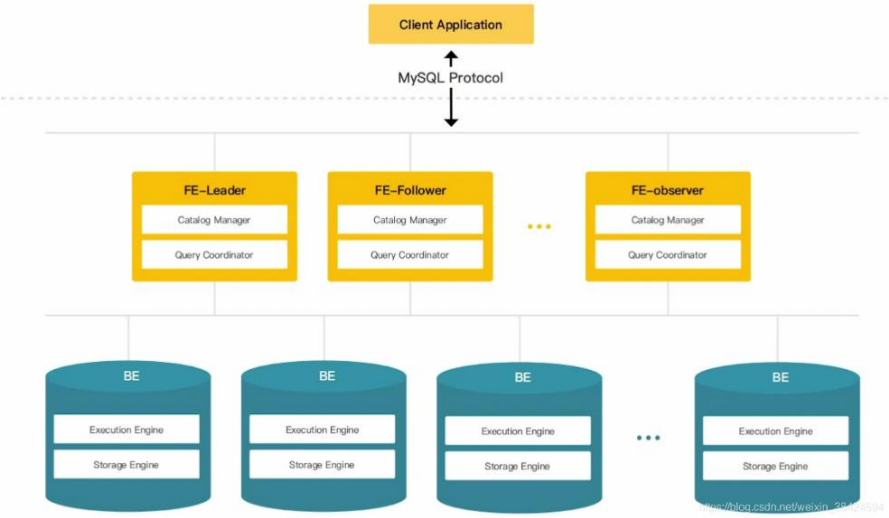
缺点：

1. 不支持事务，没有真正的update/delete；
2. 不支持高并发，可以根据实际情况修改qps相关配置文件；

StarRocks的特点

优点：

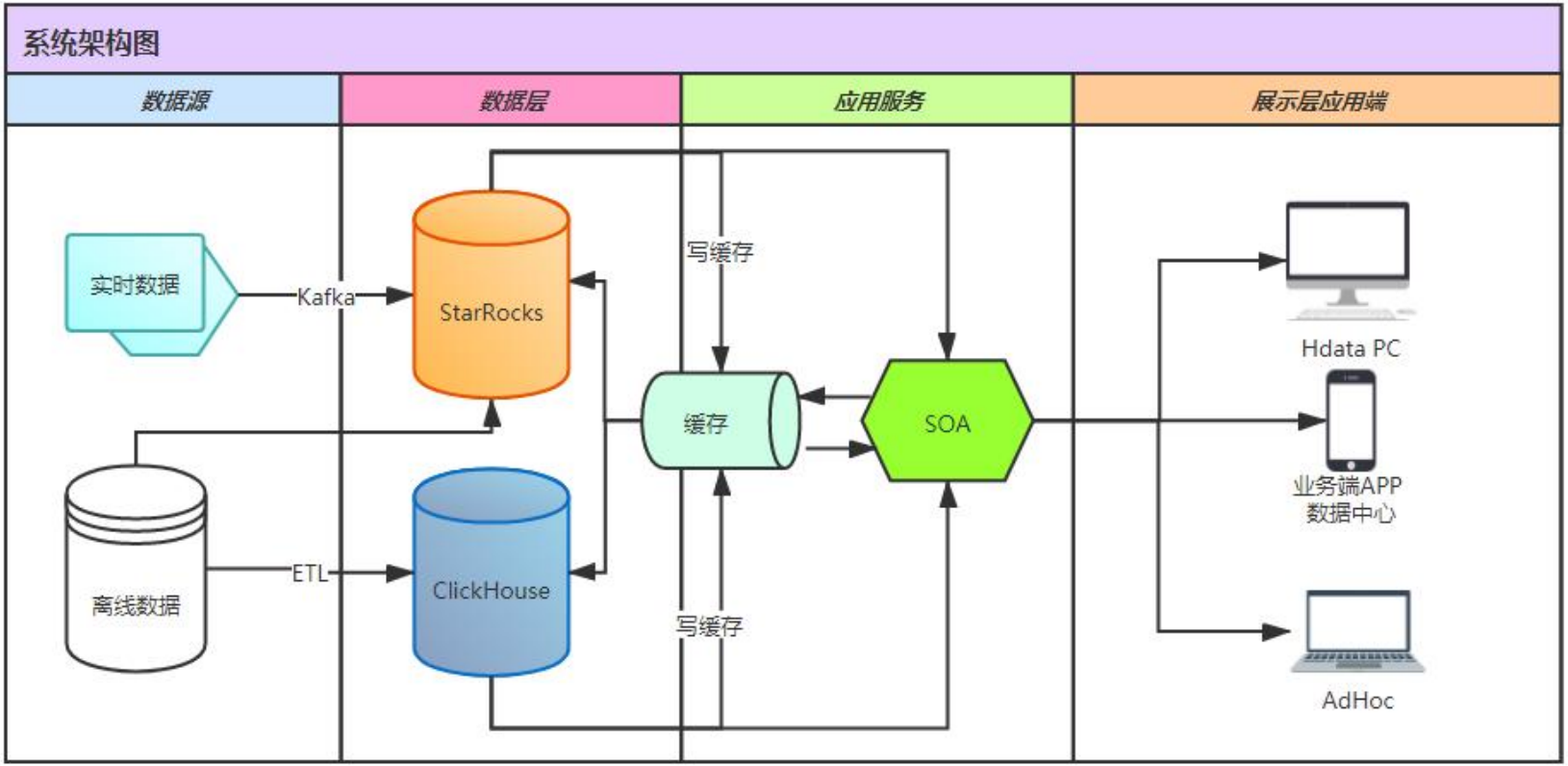
1. 支持标准的SQL语法，兼容MySql协议；
2. MPP架构，扩缩容非常简单方便；
3. 支持高并发查询；
4. 跨机房部署，实现最低成本的DR



缺点：

1. 不支持大规模的批处理；
2. 支持insert into，但最理想的是消费Kafka；

ClickHouse/StarRocks在酒店数据智能平台的架构



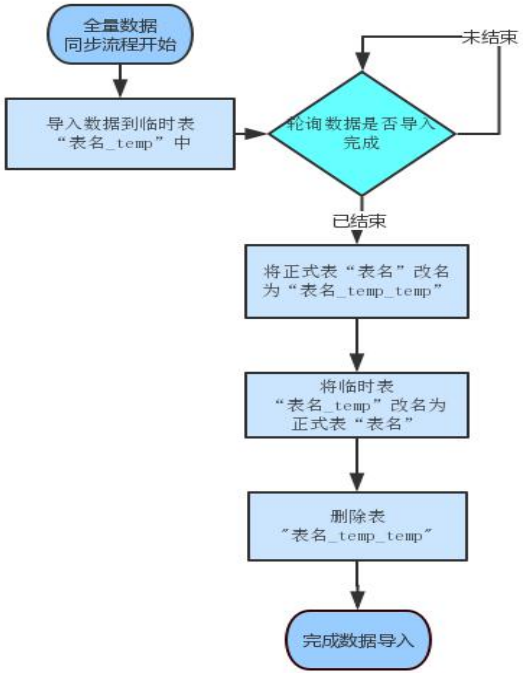
ClickHouse的全量数据同步流程

1. 清空A_temp表，将最新的数据从Hive通过ETL导入到A_temp表;

2. 将A rename 成A_temp_temp;

3. 将A_temp rename成 A;

4. 将A_ temp_temp rename成 A_temp;



其他方式：

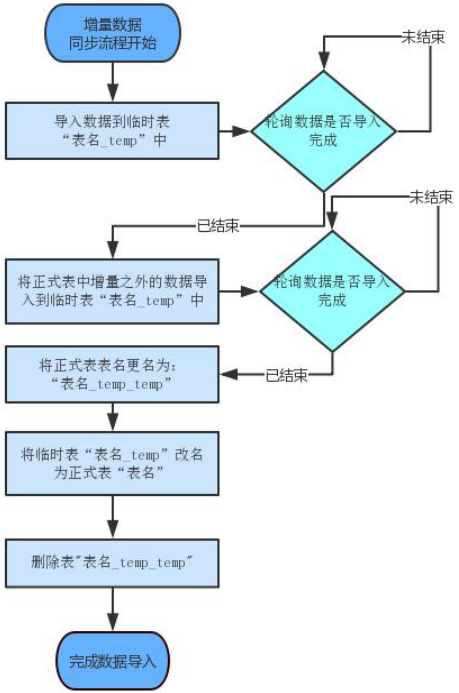
1. 采用 waterdrop 的方式大幅提升写入速度；

2. 直接读Hdfs文件的方式，但内存波动较大；

ClickHouse的增量数据同步流程

传统方式：

1. 将最近3个月的数据从Hive通过ETL入到A_temp表;
2. 将A表中3个月之前的数据select into到A_temp表；
3. 将A rename 成A_temp_temp;
4. 将A_temp rename成 A;
5. 将A_ temp_temp rename成 A_temp;

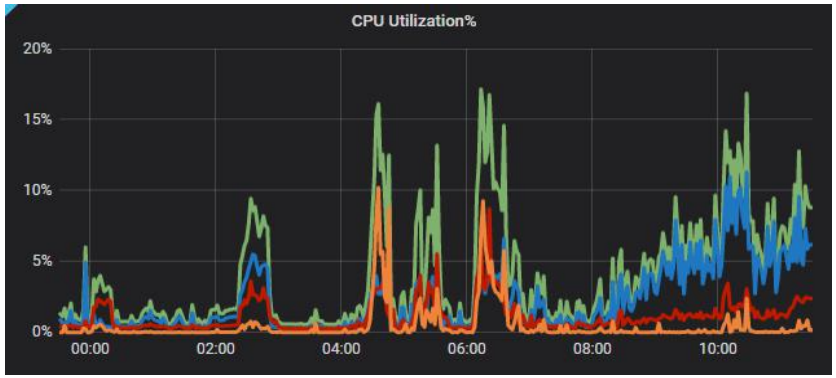


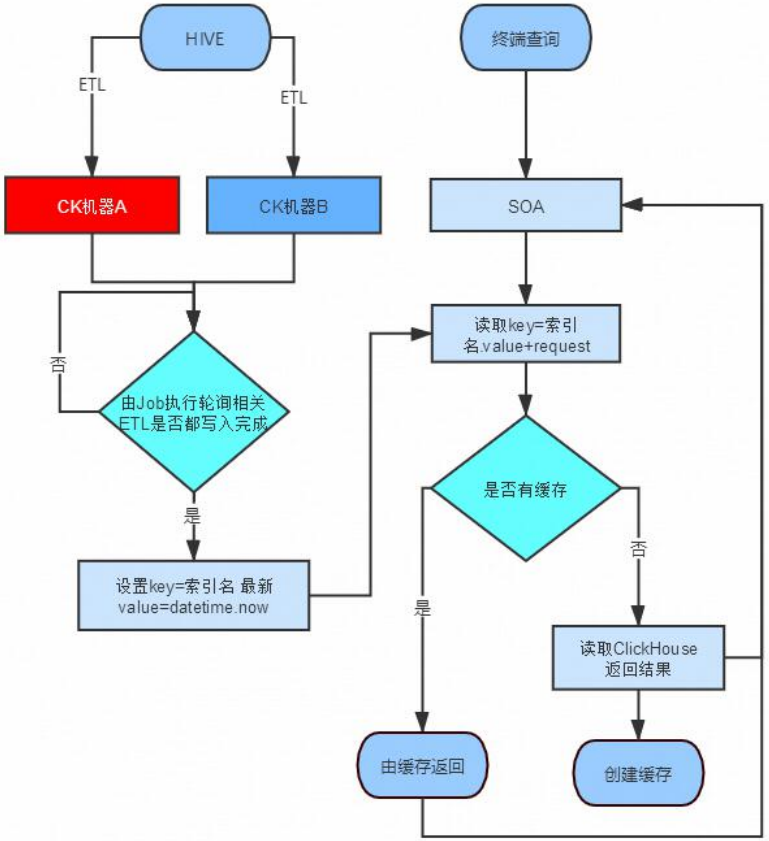
非传统： 在第2步采用：

ALTER TABLE A REPLACE PARTITION 分区名 FROM A_temp

针对ClickHouse的保护机制

1. 被动缓存；
2. 主动缓存；



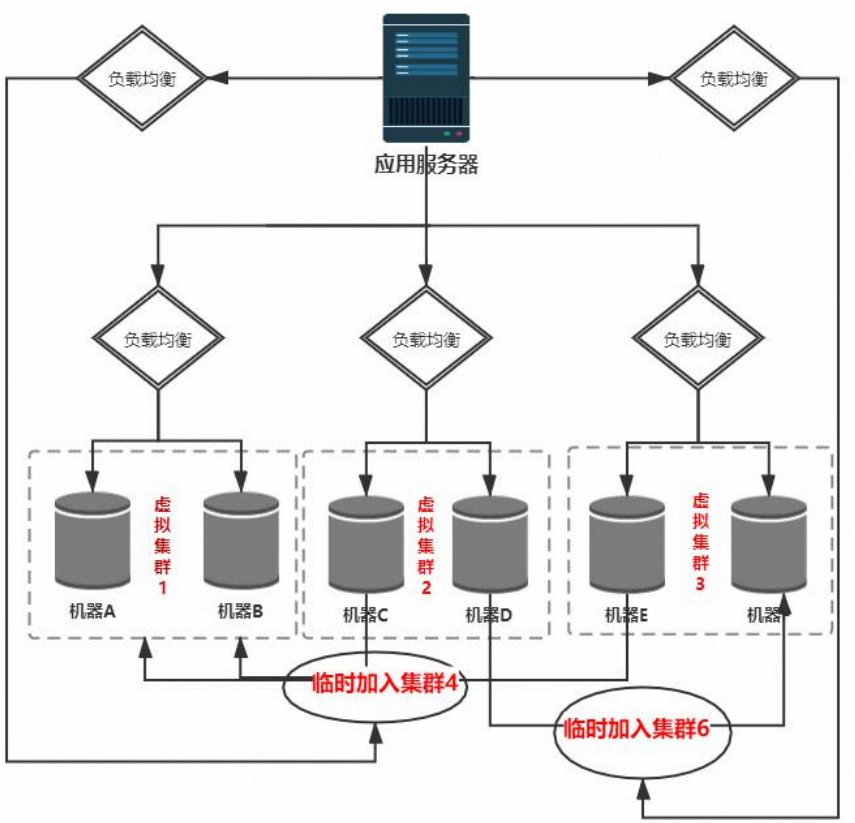


ClickHouse集群架构

- 虚拟集群最少两台机器在不同的机房；
- 数据独立，多写，相互不干扰；
- 数据读取通过应用程序做负载平衡；
- 灵活创建不同的虚拟集群用于适当的场合；
- 随时调整服务器，新增/缩减服务器；

分布式：

k8s的集群式部署



采用ClickHouse后平台的查询性能

system.query_log表，记录已经执行的查询记录

query：执行的详细SQL，查询相关记录可以根据SQL关键字筛选该字段

query_duration_ms：执行时间

memory_usage：占用内存

read_rows和read_bytes ：读取行数和大小

result_rows和result_bytes ：结果行数和大小

以上信息可以简单对比SQL执行效果

```
1  select * from `system`.query_log
2  where query_start_time > '2021-08-05 09:41:00'
3  and type='QueryFinish' --ExceptionBeforeStart
4  and query like '%-- chen_sf_test_202108061030%'
5  and query not like '%query_log%'
6  order by query_start_time
7
8  select-- chen_sf_test_202108061030
9  hotel,
10 hotelname,
11 realstar as star
```

```
uery_log
here
elect * from `system`.query_log where query_start_  输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)
```

	event_time	query_start_time	query_duration_ms	read_rows	read_bytes	12
1	-08-06 10:32:52	2021-08-06 10:32:50	1,299	342,588,933	5,239,605,594	
2	-08-06 10:33:34	2021-08-06 10:33:33	831	84,875,637	1,133,445,817	
3	-08-06 13:59:13	2021-08-06 13:59:12	727	34,300,184	728,423,011	

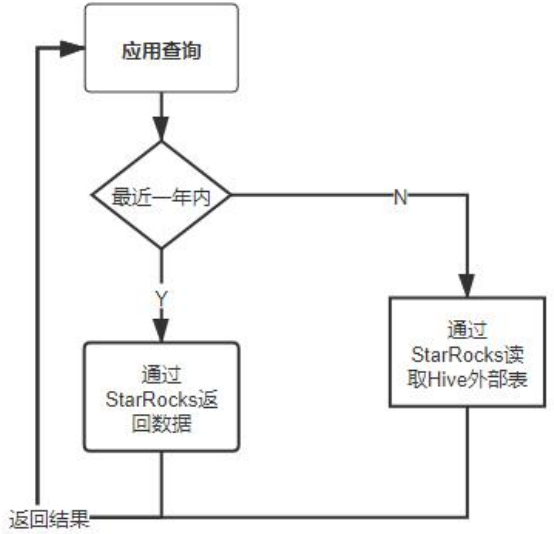
采用ClickHouse后平台的查询性能

ClickHouse应用小结

- 数据导入之前要评估好分区字段；
- 数据导入时根据分区做好Order By；
- 左右表join的时候要注意数据量的变化；
- 是否采用分布式；
- 监控好服务器的cpu/内存波动/`system`.query_log；
- 数据存储磁盘尽量采用ssd；
- 减少数据中文本信息的冗余存储；
- 特别适用于数据量大，查询频次可控的场景，如数据分析，埋点日志系统；

StarRocks应用小结

- 发挥分布式的优势，要提前做好分区字段规划；
- 支持各种join，语法会相对clickhouse简单很多；
- 一个sql可以多处用；
- 建立好守护进程以及节点监控；



devops

全球敏捷运维峰会

THANK YOU !