Clickhouse 实践总结

詹晓辉-20201027

目录

CONTENT

现状和规划

ClickHouse 机器与配置

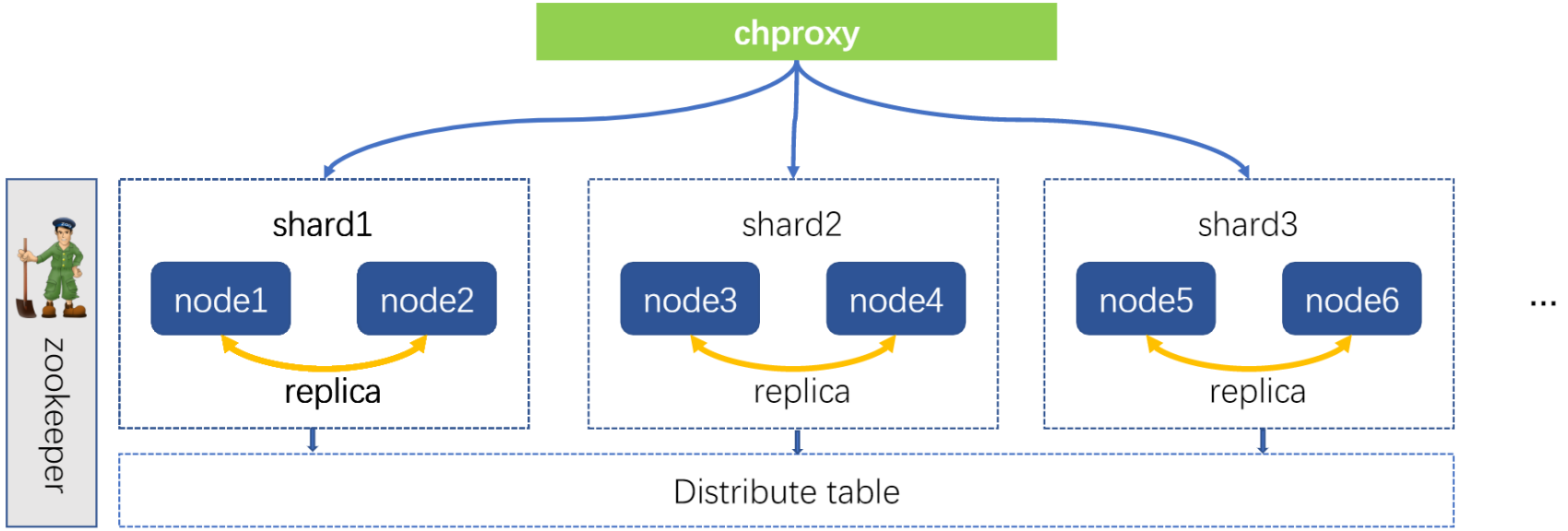
CPU：32核64线程 每次执行查询默认使用一半核进行SSE加速

mem：192G 内存越大 page cache加速越明显

disk：4块硬盘raid5，有条件可以做SSD/NVME+HDD 混合 io明显上升

zookeeper：zookeeper 单独机器/ssd硬盘/单独磁盘

部署图


chproxy
zookeeper
shard1
node1
node2
replica
shard2
node3
node4
replica
shard3
node5
node6
replica
...
Distribute table

基础参数优化

max_memory_usage：限制单次查询的最大内存

max_execution_time： 限制sql的最大执行时间

background_pool_size： 后台文件写入合并线程数，提高线程数加快初始化加载速度

background_pool_size： 后台文件写入合并线程数，提高线程数加快初始化加载速度

max_bytes_before_external_sort：当sort操作内存超过该值时会写盘

max_bytes_before_external_group_by：当group by 操作内存超过该值时会写盘 建议2*max_memory

问题与解决方法

1) Cannot allocate block number in ZooKeeper: Coordination::Exception. Too many parts (1472). Merges are processing significantly slower than inserts.

无法分配节点的原因是zknode节点产生的原因过快，后台速度合并太慢，解决方法加大batch数量，降低写入的频率可以从1s->5s

2) Zookeeper Session expire

Zookeeper gc 时间过长，node 总数过多(建议5M以下)都会导致该问题的发生，建议采用jdk11 zgc 降低gc 时间，同时增加maxSessionTimeout

3) 数据重复

clickhouse 不是完全事务的数据库，为了避免数据重复需要采用 replicateReplacingMergeTree 最好能指定主键进行合并代替

问题与解决方法

4) 重启加载速度慢，加载时间过长

增加backgroup_pool数目同时对partition数据过多的表进行重新设计partition key

5）执行group by速度慢，中间需要传输大量的数据，操作内存超出最大内存。

set distributed_group_by_no_merge=1

使用分治方法在每个shard 上执行，最后将group by 的结果再做一次整体合并，减少数据传输时间。

6)执行group by or order by 的sql 抛出oom 错误

内存溢出原因是group by 之前汇聚数据数据量超过内存可以采用堆外排序

set max_bytes_before_external_group_by=30000000000;

distributed_aggregation_memory_efficient=1;

问题与解决方法

7) 极端环境下断网断电，重启后分布式失效，或者Block structure mismatch in MergingSorted stream: different names of columns

a) 执行 select * from system.replication_queue 查看表中有多少 last_exception 不为空的表和报错异常 存在Found parts with the same 或No active replica has part 证明shard 内部 replication 复制存在异常

b) SELECT replica_path || '/queue/' || node_name FROM system.replication_queue JOIN system.replicas USING (database, table) WHERE create_time < now() - INTERVAL 3 DAY AND type = 'GET_PART' AND last_exception LIKE '%No active replica has part%'. 查出所有无效的part queue在zookeeper上的路径并删除

c) 对于Found parts with the same min block and with the same max block as the missing part 1599618600_0_225_24. Hoping that it will eventually appear as a result of a merge：采取dettache partition 后重新attach

d) SYSTEM RESTART REPLICAS

监控

基于Prometheus的监控系统

Promtail: 采集clickhouse-server error日志

Loki: 收集clickhouse-server日志

Clickhouse_exporter: 收集clickhouse-server的sql执行情况

Zookeeper_exporter+jmx_exporter:收集zookeeper metrics

Grafana:展示告警查询

监控

Clickhouse 关键指标

clickhouse_ephemeral_node：存活节点

clickhouse_query_memory_usage： sql执行所消耗的内存

clickhouse_query_duration_ms： sql执行所消耗的时间主要检查慢查询

clickhouse_read_bytes： sql执行读取的数据量

clickhouse_query_memory_usage{queryType=~“ExceptionBeforeStart|ExceptionWhileProcessing”}： sql执行失败的语句

展望

不足：

1）分布式功能欠缺，无法快速扩展和rebalance数据

2） 对zookeeper的依赖极为严重，zk负担过重。

规划：

1）将clickhouse-server 与k8s 整合能够快速扩展集群。同时将集群存储与计算分离

2）给clickhouse增加reshard功能在节点发生改变时执行system reshard 可以进行rebalance

THANKS