目录

[一、 ClickHouse安装 1](#_Toc1648730795)

[(1) 安装clickhouse 1](#_Toc620903412)

[(2) 修改配置文件 1](#_Toc1352463157)

[(3) 启动clickhouse 2](#_Toc1385900453)

[二、 ClickHouse文件写入性能测试(4core + 16G) 2](#_Toc1235883725)

[(1) Parquet数据写入测试 2](#_Toc133799449)

[(2) JSON数据写入测试 3](#_Toc353080858)

[(3) sparkstreaming 写clickhouse性能测试和异常测试 4](#_Toc1868017409)

[三、 ClickHouse查询性能测试 4](#_Toc327240425)

[(1) 漏斗分析(7.9亿+条数据) 4](#_Toc545752096)

[(2) 事件分析(7.9亿+条数据) 5](#_Toc1314383180)

[(3) 标签查询(4000w+条数据) 7](#_Toc1974337126)

[(4) 联表查询(大表1亿+条数据，小表37W+条数据) 7](#_Toc1830490037)

1. **ClickHouse安装**
2. 安装clickhouse

|  |
| --- |
| sudo apt-get install apt-transport-https ca-certificates dirmngr  sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv E0C56BD4  echo "deb https://repo.clickhouse.com/deb/stable/ main/" | sudo tee /etc/apt/sources.list.d/clickhouse.list  sudo apt-get update  #安装服务端及客户端  sudo apt-get install -y clickhouse-server clickhouse-client |

1. 修改配置文件
2. 在/etc/clickhouse-server/下新建自定义配置文件metrika.xml并输入配置

|  |
| --- |
| <yandex>  <!-- 集群配置 -->  <clickhouse\_remote\_servers>  <!-- 集群名称(自定义) -->  <ck\_cluster>  <shard>  <internal\_replication>true</internal\_replication>  <replica>  <host>instance-1</host>  <port>9000</port>  </replica>  <replica>  <host>instance-2</host>  <port>9000</port>  </replica>  </shard>  </ck\_cluster>  </clickhouse\_remote\_servers>  <!-- zookeeper配置 -->  <zookeeper\_servers>  <node index="1">  <host>triwin-bd-zookeeper</host>  <port>2181</port>  </node>  <node index="2">  <host>triwin-bd-zookeeper</host>  <port>2182</port>  </node>  <node index="3">  <host>triwin-bd-zookeeper</host>  <port>2183</port>  </node>  </zookeeper\_servers>  <!-- clickhouse建表宏配置 -->  <macros>  <shard>shard01</shard>  <replica>replica01</replica>  </macros>  </yandex> |

1. 修改/etc/clickhouse-server/config.xml 配置

|  |
| --- |
| #第615行插入配置  <include\_from>/etc/clickhouse-server/metrika.xml</include\_from>  <zookeeper incl="zookeeper\_servers" optional="true" />  <macros incl="macros" optional="true" />  <remote\_servers incl="clickhouse\_remote\_servers" optional="true" />  #取消下列配置的注释  <listen\_host>::</listen\_host> |

1. 启动clickhouse

|  |
| --- |
| sudo service clickhouse-server start |

1. **ClickHouse文件写入性能测试(4core + 16G)**
2. Parquet数据写入测试
3. 事件数据

|  |  |
| --- | --- |
| 数据格式 | Parquet，8个字段 |
| 数据大小 | 157G、7.97亿行、181个文件，每个文件888M，每个文件行数440W行左右 |
| 写入方式 | cat clickhouse.parquet | clickhouse-client --query="INSERT INTO default.user\_event\_history\_local FORMAT Parquet" |
| 写入速率 | 总耗时3065s，平均每秒写入大约26万行 |
| 数据压缩率 | 默认deflate压缩格式，压缩前162.06G，压缩后15.06G，压缩率大约90% |
| data列map格式存储 | 压缩前149.14 G，压缩后16.58 G，压缩率大约90% |
| 资源消耗 | 平均值：CPU：120%，内存：30% |
| 数据完整性 |  |
| 复制集完整性 |  |

1. 基础标签宽表数据

|  |  |
| --- | --- |
| 数据格式 | Parquet，193个字段 |
| 数据大小 | 5.43G、4135万行、59个文件，每个文件95M，每个文件行数70万行左右 |
| 写入方式 | cat clickhouse.parquet | clickhouse-client --query="INSERT INTO default.user\_basic\_tags\_local FORMAT Parquet" |
| 写入速率 | 总耗时684s，平均每秒写入大约6万行 |
| 数据压缩率 | 默认deflate压缩格式，压缩前15.37G，压缩后4.25G，压缩率大约72% |
| 资源消耗 | 平均值：CPU：200%，内存：45% |
| 数据完整性 |  |
| 复制集完整性 |  |

1. JSON数据写入测试

|  |  |
| --- | --- |
| 数据格式 | JSON格式，3个字段，其中一个字段为STRUCT类型 |
| 数据文件大小 | 10.8G、2420万行、6个文件，每个文件1.8G，每个文件行数400万行左右 |
| 写入方式 | clickhouse-client --query="INSERT INTO default.user\_define\_tags FORMAT JSONEachRow" < clickhouse.json |
| 写入速率 | 总耗时90s，平均每秒写入大约26.8万行 |
| 数据压缩率 | 默认deflate压缩格式，压缩前11G，压缩后426M，压缩率大约96% |
| 资源消耗 | 平均值：CPU：150%，内存：20% |
| 数据完整性 |  |
| 复制集完整性 |  |

1. sparkstreaming 写clickhouse性能测试和异常测试

|  |  |
| --- | --- |
| 源数据格式 | Json数据类型 |
| 数据量 | 2千万条数据 |
| 表类型 | 副本表 |
| 写入方式 | Clickhosue jdbc（8123端口）方式，集群模式（url中多个实例ip），并设置insert\_quorum=2（写多个副本才算写入成功） |
| SparkStreaming 资源 | 4 excutor 、6 G memory /excutor 、3 core / executor、driver memory 2G |
| 写入速率 | Sparkstreaming 设置的批次时间为8s一个批次，实际每批次平均耗时11s，每批次平均处理数据140000 条，平均12727条/s |
| 资源消耗 | 平均值：CPU：5.06%，内存：4.9%，磁盘吞吐量：6.33M/s |
| 异常场景测试 | 程序在写clickhouse 过程中不断重启其中一台clickhouse server服务 |
| 异常场景测试结果 | 通过设置insert\_quorum>1 并在程序中对未写入多个副本的异常进行捕获后重新写入，写入clickhouse 集群数据为 2千万条，不存在数据丢失和数据重复情况 |
| 数据完整性 |  |
| 复制集完整性 |  |

1. **ClickHouse查询性能测试**
2. 漏斗分析(7.9亿+条数据)

* 事件无筛选条件漏斗

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 事件时间范围 | 查询耗时 | 漏斗窗口大小 | 漏斗层数 |
| ClickHouse | 7天 | 11s | 1天 | 9层 |
| BigQuery | 7天 | 29s | 1天 | 9层 |
| 结果对比一致 |  | | | |

* 事件+筛选条件漏斗

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 事件时间范围 | 查询耗时 | 漏斗窗口大小 | 漏斗层数 |
| ClickHouse | 7天 | 13s | 1天 | 7层 |
| BigQuery | 7天 | 22s | 1天 | 7层 |
| 结果对比一致 |  | | | |

1. 事件分析(7.9亿+条数据)

* spin事件分析

|  |  |  |  |
| --- | --- | --- | --- |
|  | 事件时间范围 | 查询耗时 | 指标数量 |
| ClickHouse | 7天 | 10s | 7 |
| BigQuery | 7天 | 4s | 7 |
| 结果对比一致 |  | | |

* recharge事件分析

|  |  |  |  |
| --- | --- | --- | --- |
|  | 事件时间范围 | 查询耗时 | 指标数量 |
| ClickHouse | 7天 | 1.4s | 8 |
| BigQuery | 7天 | 2s | 8 |
| 结果对比一致 |  | | |

* spin(加条件)事件分析

|  |  |  |  |
| --- | --- | --- | --- |
|  | 事件时间范围 | 查询耗时 | 指标数量 |
| ClickHouse | 7天 | 8.9s | 5 |
| BigQuery | 7天 | 3.4s | 5 |
| 结果对比一致 |  | | |

1. 标签查询(4000w+条数据)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 查询耗时 | 筛选条件(and 和or条件) | 结果对比一致 |
| ClickHouse(宽表) | 0.691s | 5个 |  |
| BigQuery(宽表) | 1s | 5个 |  |
| BigQuery(函数) | 2.8s | 5个 |  |

1. 联表查询(大表1亿+条数据，小表37W+条数据)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 查询耗时 | 关联方式 | 结果对比一致 |
| ClickHouse | 8.5s | 小表 left join 大表 |  |
| BigQuery | 1.4s |
| ClickHouse | 3.8s | 大表 left join 小表 |  |
| BigQuery | 1.7s |
| ClickHouse | 3.6s | 大表 join 小表 |  |
| BigQuery | 1s |
| ClickHouse | 4.7s | 大表 full join 小表 |  |
| BigQuery | 1.3s |