# ClickHouse基准性能测试报告

## 前言

本次测试主要评估ClickHouse的整体性能，量化当前ClickHouse的性能指标，对各种场景下ClickHouse性能表现进行评估，为业务应用提供参考。

本报告主要介绍此次测试的基本条件、ClickHouse在各种测试场景下的性能指标（主要包括单次请求平均延迟和系统吞吐量）以及对应的资源利用情况，并对各种测试结果进行分析。

## 2.测试环境

### 2.1硬件配置

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IP | CPU型号 | 核心数 | 内存(GB) | 磁盘(TB) | 以太网(Mbps) |
| 192.168.1.143 | Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz | 8 | 64 | 10 | 100 |
| 192.168.1.144 | Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz | 8 | 64 | 10 | 100 |
| 192.168.1.142 | Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz | 8 | 64 | 10 | 100 |
| 192.168.1.17 | Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz | 12 | 64 | 10 | 100 |

### 2.2软件配置

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IP | 操作系统 | JVM | ZooKeeper | Clickhouse | 测试工具 |
| 192.168.1.143 | CentOS Linux release 7.4.1708 (Core) | openjdk version "1.8.0\_275" | 3.4.6 | 21.2.8.31 | YCSB 0.17.0 |
| 192.168.1.144 |

### 2.3 clickhouse核心配置

|  |  |
| --- | --- |
| 配置项 | 值 |
| max\_memory\_usage | 40000000000 |
| max\_connections | 4096 |
| max\_thread\_pool\_size | 10000 |
| mark\_cache\_size | 5368709120 |

### 2.4 测试原则

1. 可测试
2. 可重复
3. 可对比

### 2.5 测试指标

|  |  |
| --- | --- |
| 网络延迟 | 吞吐量 |
| 开始执行一个操作到操作完成中间的耗时 | 每秒钟执行操作的次数 |

### 2.6 测试变量

|  |  |
| --- | --- |
| 数据量 | 并发数 |
| 测试不同数据量下的读写性能差异 | 测试不同并发下读写性能的差异 |

## 3.测试场景

### 3.1测试场景一：位图计算效率测试

说明：模拟对多个位图进行计算，包括与、或、非运算，检测位图的运算效率。

场景：单机

查询测试：针对多个位图数据进行合并处理，分别处理10个，100个，1000个，10000个计算，计算时延、cpu占用、内存占用

压力测试：合并100个位图处理，线程数10个递增，查看线程数和时延变化

#### 3.1.1每个表不同预分区数量下的性能对比

##### 1)方案

|  |  |
| --- | --- |
| 配置项 | 说明 |
| 位图数量 | 10个，100个，1000个，10000个 |
| 操作次数 | 100000次 |
| 数据行数 | 24675455条 |
| 字段数量 | 1个 |
| 字段大小 | 10字节 |
| 请求分布模型 | zipfian |
| 线程数 | 1 |
| 场景 | 单机 |

##### 2)测试结果

###### 延迟结果

###### 吞吐量结果

##### 3)小结

1. 进行计算的位图数量与延迟成正相关，当位图数量到达千级数据量的情况下，平均延迟的变化幅度不大；当位图数量到达万级后，平均延时明显变大。
2. 进行计算的位图数量与吞吐量成负相关。当处理的位图数量只有几百个的时候，吞吐量的变化幅度不大；处理的位图数量到达千级后，吞吐量明显下降。

#### 3.1.2 不同并发下的混合读写性能对比

##### 1) 方案

|  |  |
| --- | --- |
| 配置项 | 说明 |
| 位图数量 | 1000个 |
| 操作次数 | 100000次 |
| 数据行数 | 24675455条 |
| 字段数量 | 1个 |
| 字段大小 | 10字节 |
| 请求分布模型 | zipfian |
| 线程数 | 10、100、1000、100000、1000000 |
| 场景 | 单机 |

##### 测试结果

###### 延迟结果

###### 吞吐量结果

##### 小结

1. 平均延迟有明显波动，线程数从零到千级的时候，平均延迟上升，线程数到千级后，平均延迟呈下降趋势，到十万级后趋于平稳。
2. 线程数与吞吐量成负相关。当线程数从千级到万级变化时，吞吐量的下降幅度较大；线程数量到达万级后，吞吐量的没有明显变化，稳定在一个值附近。

### 3.2测试场景二：本地表查询性能

说明：查询一个区域某小时的用户数与当天累计用户数

查询测试：针对多个区域数据进行查询，分别查询10个地理索引，100个，1000个，10000个计算，计算时延、cpu占用、内存占用

场景：单机

压力测试：查询100个地理索引，线程数10个递增，查看时延变化

查询100个地理索引，线程数10个，数据量按照100万递增，查看时延变化

#### 3.2.1 查询不同数目地图索引个数时的性能对比

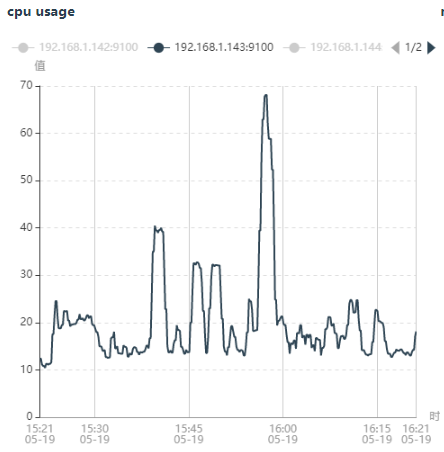
##### 1）方案

|  |  |
| --- | --- |
| 配置项 | 说明 |
| S2地理索引数量 | 10个，100个，1000个，10000个 |
| 操作次数 | 100000次 |
| 数据行数 | 50000000条 |
| 字段数量 | 2个 |
| 字段大小 | 10字节 |
| 请求分布模型 | zipfian |
| 线程数 | 1 |
| 测试场景 | 单机 |

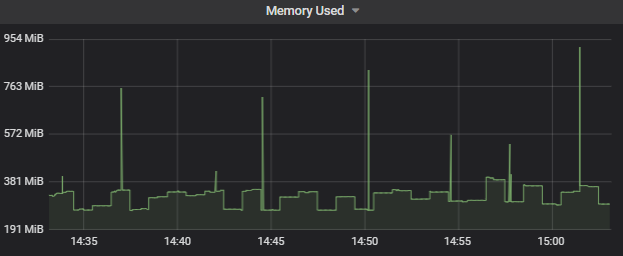
##### 2）测试结果

###### 延迟和吞吐量

###### CPU使用率



###### 内存使用



##### 3）小结

1. 吞吐量

当查询的S2地理索引的个数在千级以内时，吞吐量变化较小，控制在25ops/sec以内。当查询的S2地理索引的个数在万级以上时，吞吐量大幅度下降，当S2地理索引的个数达到10万时，吞吐量下降为10ops/sec以内。

1. 延迟

延迟随查询S2地理索引的个数增加而呈现增长趋势。S2地理索引个数从零到千级的时候，延迟变化较小；S2地理索引到万级后，延迟增长幅度较大，到十万级后趋于平稳。

1. CPU使用率

单个查询CPU使用率在40%以内，CPU一般在50%左右会出现查询波动，达到70%会出现大范围的查询超时，CPU是关键的指标。

1. 内存使用

进行简单查询，只返回少数列，且返回的结果集较小，内存使用在最大可用的内存范围内。

#### 3.2.2 不同并发下的混合读写性能对比

##### 1）方案

|  |  |
| --- | --- |
| 配置项 | 说明 |
| S2地理索引数量 | 10个，100个，1000个，10000个 |
| 操作次数 | 100000次 |
| 数据行数 | 50000000条 |
| 字段数量 | 2个 |
| 字段大小 | 10字节 |
| 请求分布模型 | zipfian |
| 线程数 | 10、20、30、40、50、100、200、1000 |
| 场景 | 单机 |

##### 测试结果

##### 3）小结

1. 延迟

线程数个数在100以内时，平均延迟变化较小，呈现持续小幅度增长的趋势；线程数继续增加大于100个小于200并发时，平均延迟骤变，增长幅度较大，到达峰值；线程数大于200后，平均延迟又下降。

2、吞吐量

线程数个数在100以内时，吞吐量变化较小，缓慢下降趋势；当线程个数到达200并发后，吞吐量急速下降。

#### 3.2.3 固定地图索引数目不同数据量的性能对比

##### 1）方案

|  |  |
| --- | --- |
| 配置项 | 说明 |
| S2地理索引数量 | 100个 |
| 操作次数 | 100000次 |
| 数据行数 | 50000000条，数据量按照100万递增 |
| 字段数量 | 2个 |
| 字段大小 | 10字节 |
| 请求分布模型 | zipfian |
| 线程数 | 10 |
| 场景 | 单机 |

##### 2）测试结果

##### 小结

1. 延迟

随着数据量的增加，平均延迟呈现缓慢上升趋势，数据量从5千万增加到2亿，平均延迟的变化依旧较小，可见亿级别数据量对延迟的影响较小。

1. 吞吐量

随着数据量的增加，吞吐量上下波动，波动范围不大。

### 3.3测试场景三：分布式表查询性能

说明：查询一个区域某小时的用户数与当天累计用户数

场景：分布式

查询测试：针对多个区域数据进行查询，分别查询10个地理索引，100个，1000个，10000个计算，计算时延、cpu占用、内存占用

压力测试：查询100个地理索引，线程数10个递增，查看时延变化

#### 3.3.1 查询不同数目地图索引个数时的性能对比

##### 方案

|  |  |
| --- | --- |
| 配置项 | 说明 |
| 地理索引数量 | 10个，100个，1000个，10000个 |
| 操作次数 | 100000次 |
| 数据行数 | 50000000条 |
| 字段数量 | 2个 |
| 字段大小 | 10字节 |
| 请求分布模型 | zipfian |
| 线程数 | 1 |
| 测试场景 | 集群 |

##### 测试结果

##### 小结

1. 延迟

S2地理索引的个数增长对平均延迟的影响较小，S2地理索引个数从零到千级的时候，延迟变化较小；S2地理索引从千级到万级后，延迟增长幅度较大。

1. 吞吐量

当查询的S2地理索引的个数在千级以内时，吞吐量呈现上升趋势，峰值到达766.353ops/sec。当查询的S2地理索引的个数在前级以上时，吞吐量大幅度下降，降至210.35ops/sec。

#### 3.3.2 不同并发下的混合读写性能对比

##### 方案

|  |  |
| --- | --- |
| 配置项 | 说明 |
| 地理索引数量 | 10个，100个，1000个，10000个 |
| 操作次数 | 100000次 |
| 数据行数 | 50000000条 |
| 字段数量 | 2个 |
| 字段大小 | 10字节 |
| 请求分布模型 | zipfian |
| 线程数 | 10、20、30、40、50、100、200、1000 |
| 测试场景 | 集群 |

##### 测试结果

##### 小结

1. 延迟

当线程数在100以内，平均延迟随线程数的增加缓慢增长，幅度较小，最大值为15.556ms；当线程数大于100时，平均延迟随线程数的增加急速增长，幅度较大。

1. 吞吐量

随着线程数的增加，吞吐量呈波动趋势。当线程数为100时，趋于稳定。

### 3.4测试场景四：ck点查的效率

#### 3.4.1 本地表点查性能

说明：从海量数据中查询少量数据的效率

场景：单机

源数据量：1亿、5亿、10亿

查询数据量：1条、100条、1000条、10000条、100000条

度量：时延、吞吐量

##### 1）方案

|  |  |
| --- | --- |
| 配置项 | 说明 |
| 源数据量 | 1亿、5亿、10亿 |
| 操作次数 | 100000次 |
| 查询数据量 | 1条、100条、1000条、10000条 |
| 字段数量 | 2个 |
| 字段大小 | 10字节 |
| 请求分布模型 | zipfian |
| 线程数 | 1 |
| 测试场景 | 单机 |

##### 2）测试结果

* 源数据量：1亿
* 源数据量：5亿
* 源数据量：10亿

##### 3）小结

1. 延迟

源数据量相同，查询不同的数据量时，当查询的数据量在万级以内时，平均延迟变化较小，控制在1ms以内；当查询的数据量到达万级以上时，平均延迟增长较大，达到8ms左右，仍然小于10ms。

源数据量不相同，查询的数据量相同时，平均延迟的变化趋势相同。都是当查询的数据量在万级以内时，延迟基本不变；查询的数据量大于万级时，延迟增长明显。

2. 吞吐量

源数据量相同，查询不同的数据量时,当查询的数据量在万级以内时,吞吐量呈现缓慢下降的趋势；当查询的数据量到达万级以上时，吞吐量急速下降，值小于200ops/sec。

源数据量不相同，查询的数据量相同时，吞吐量的变化趋势相同。都是当查询的数据量在万级以内时，吞吐量缓慢下降；查询的数据量大于万级时，吞吐量下降明显。

#### 3.4.2 分布式表点查性能

说明：从海量数据中查询少量数据的效率

场景：集群

源数据量：1亿、5亿、10亿

查询数据量：1条、100条、1000条、10000条、100000条

度量：时延、吞吐量

##### 1）方案

|  |  |
| --- | --- |
| 配置项 | 说明 |
| 源数据量 | 1亿、5亿、10亿 |
| 操作次数 | 10000次 |
| 查询数量 | 1条、10条、100条、1000条、10000条 |
| 字段数量 | 2个 |
| 字段大小 | 10字节 |
| 请求分布模型 | zipfian |
| 线程数 | 1 |
| 场景 | 集群 |

##### 2）测试结果

###### 延迟结果

* 源数据量：1亿
* 源数据量：5亿
* 源数据量：10亿

###### 吞吐量结果

* 源数据量：1亿
* 源数据量：5亿
* 源数据量：10亿

##### 3）小结

1. 在一亿、五亿、十亿的源数据量基础上，查询数量相同时，吞吐量的差异不大，并且都是在千级查询数量以后，有下降的趋势。

2. 在一亿的源数据量基础上，查询数量相同时，平均时延的差异不大，在百级查询数量以后，平均延迟呈上升趋势；在五亿、十亿的源数据量基础上，查询数量相同时，平均时延的差异不大，在千级查询数量以后，平均延迟呈上升趋势。

## 4 总结

1. 位图计算效率

位图个数在千级时，并发数在100以内时，延迟较小，吞吐量适中，效率最佳。

1. 本地表和分布式表查询性能。

简单查询时，分布式表的平均延迟大于本地表，吞吐量小于本地表。

较复杂查询时，分布式表的平均延迟小于本地表。可见集群模式时，会将计算压力分摊到多台服务器。由于ClickHouse会自动将查询拆解为多个task下发到集群中，然后进行多机并行处理，最后把结果汇聚到一起。极致的并行处理能力，极大的降低了查询延时。

1. 单点查询效率

适合在线查询，单点查询的延迟分析，意味着在没有对数据做任何预处理的情况下以极低的延迟处理查询并将结果加载到用户的页面中，且当只请求万级据量时，平均延迟可控制在2ms以内。

1. 处理短查询的延时时间

数据被page cache缓存的情况下，它的延迟应该小于50毫秒(最佳情况下应该小于10毫秒)。 否则，延迟取决于数据的查找次数。延迟可以通过以下公式计算得知： 查找时间（10 ms） \* 查询的列的数量 \* 查询的数据块的数量

1. CPU使用率

Clickhouse快是因为采用了并行处理机制，即使一个查询，也会用服务器一半的CPU去执行，所以ClickHouse不能支持高并发的使用场景，默认单查询使用CPU核数为服务器核数的一半。

1. 线程数

不支持高并发，官方建议qps为100，可以通过修改配置文件增加连接数，但是在服务器足够好的情况下。