

基于 MySQL 与 ClickHouse 的混合存储架构在亿级数据查询中的优化实践

王佩佩

南京中新赛克软件有限责任公司

摘要: 本文主要研究了 MySQL 和 ClickHouse 混合存储架构在处理大规模数据查询时的优化方法。MySQL 擅长处理需要严格事务保证的操作, 而 ClickHouse 在快速分析大量数据方面更有优势, 把这两者结合起来, 就能形成一套新的存储方案。从实际应用的角度来看, 这种混合架构的设计需要考虑几个核心问题。比如怎么让两个数据库之间的数据保持同步, 如何设计合理的数据存储模型, 还有选择什么样的分区方式来提升查询效率。这些都是在处理亿级数据时必须解决的关键环节。通过对数据模型、分区策略和查询技术的调整, 这套混合架构能让系统在处理大量数据查询时, 响应速度和整体处理能力都有明显提升。

关键词: MySQL; ClickHouse; 混合存储架构; 亿级数据查询; 查询性能优化

一、MySQL 与 ClickHouse 混合存储架构概述

1.1 MySQL 与 ClickHouse 的基本特性

MySQL 是目前用得比较多的开源关系型数据库管理系统, 它的特点是运行稳定、不容易出问题, 而且功能也很全面。它支持标准的 SQL 语言, 能处理的数字、文字等数据类型很丰富, 还能把多个操作当成一个整体来处理, 不会出现中间出错的情况。不管是简单的信息记录, 还是复杂的业务流程里的数据存储和管理, 它都能满足需求。

ClickHouse 则是另一种开源数据库, 它专门用来做在线分析查询。它的查询速度特别快, 尤其适用于大数据量下的实时分析和多维分析, 其独特的列式存储和向量化执行引擎使得它在处理海量数据时能够展现出卓越的性能优势。

1.2 混合存储架构的设计理念

混合存储架构的设计思路, 核心是把 MySQL 和 ClickHouse 各自的长处都用起来, 以此解决亿级数据查询里遇到的各种难题。具体来说, 就是让 MySQL 当主数据库, 专门管事务性操作, 保证数据不会乱; ClickHouse 则负责分析, 专门处理大量数据的实时分析和从多个角度展开的分析任务。这种架构把事务处理和分析处理分开了, 两边不会互相影响。另外, 通过合理安排数据存在哪里、让查询走正确的路径, 数据访问效率变高了, 查询性能也得到了优化。

二、混合存储架构的搭建与配置

2.1 MySQL 与 ClickHouse 的集成方法

实现 MySQL 与 ClickHouse 的集成可以借助中间件完成。比如说用 Canal 这类工具去监听 MySQL 的 binlog 文件, 把数据的变化实时同步到 ClickHouse 里。这种做法能让两边的数据保持一致, 而且不会太影响 MySQL 的运行速度。也可以用

ETL 工具来做。定期从 MySQL 里把需要的数据抽出来, 经过一定的格式转换后再加载到 ClickHouse 中。这种方式更适合那些对数据同步速度要求不那么高的场景。另外还有一种办法, 就是用一些开源的连接器。直接在应用程序这一层实现两个数据库之间的交互, 根据具体的业务需求灵活选择该把数据存在哪个库、该用哪个库来查询。

在实际操作的时候, 选中间件得考虑几个点。比如它能不能稳定运行、处理数据的速度怎么样、有没有活跃的社区提供技术支持等。拿 Canal 来说, 它能解析 MySQL 的 binlog, 还提供了很多可以调整的设置, 能应对不同场景下的使用需求。ETL 工具的选择则要结合数据量大小、转换过程的复杂程度以及抽取数据的频率来定, 选对了工具才能让数据同步更顺畅。至于开源连接器, 得重点看看它能不能和应用程序很好地兼容, 以及是否支持业务需要的功能, 这样才能保证集成过程顺利完成。

2.2 数据同步与一致性保障策略

在数据同步的实际操作里, 要让数据保持一致, 得用一些具体的办法来保障。比如说, 有些业务对数据更新的速度要求特别高, 这时候可以优先用基于 binlog 的同步方式, 像用 Canal 工具盯着 MySQL 的 binlog 变化, 再实时把数据同步到 ClickHouse 里, 这样能尽量缩短数据更新的时间差, 让两边的数据保持一致。另外, 要是业务对实时性没那么严格, 也可以用 ETL 工具定期把数据抽出来、转成需要的格式再加载进去。不过这时候得把同步的时间间隔设置合理, 还得盯着同步过程中有没有数据对不上的地方, 发现问题就及时补上。还有一些辅助的办法, 比如给数据做校验和、给不同版本的数据做标记, 这些机制能在数据同步完之后, 再检查一

遍数据对不对、全不全，进一步确保数据的一致性。

三、数据模型与分区策略

3.1 亿级数据的模型设计

当需要处理海量数据时，模型搭建得同时考虑查询速度和存储开销。比如说，用星型或雪花结构来做维度表和事实表，通过关联字段减少数据重复，再结合 ClickHouse 的列式存储特点让查询更快。另外，对于经常用到的查询字段，可以提前做好汇总表或者物化视图，把常用的指标先算出来，这样就不用每次查询都实时计算了。从实际应用来看，合理调整数据的详细程度也很重要，在满足业务需求的基础上，适当合并数据来缩小总量，这对提升查询效率有帮助。

3.2 ClickHouse 分区键的选择与优化

在选择 ClickHouse 分区键时，要把数据分布情况、日常查询习惯和后期维护难度都考虑进去。比如说，优先挑那些取值范围广的字段，像日期、用户 ID 这类，这样能让数据均匀分到各个分区里，不会因为某块数据堆得太多影响查询速度。针对按时间记录的数据，按天或者按月分区是比较合适的做法。这样既能快速找到某个时间段的数据，也能避免单个分区的数据量太大，方便平时调整和管理。要是遇到需要同时用多个条件查询的情况，可以试试把几个字段组合起来分区。这种多级分区的方式能让数据划分得更细致，查询的时候能更快定位到目标数据。另外，还得隔一段时间检查一下分区键是不是还合适，根据业务变动和数据增加的情况调整策略，让系统一直保持稳定高效的状态。

四、查询性能优化实践

4.1 MySQL 查询优化技巧

对于 MySQL 查询优化，首先要注意索引的合理使用。得结合查询时常用的条件和数据本身的分布特点，给那些经常被用来查询的字段建合适的索引，这样就能避免因为要扫描整个表而拖慢速度。另外，还要定期看看哪些索引是常用的、哪些是没用的，把多余或者用不上的索引删掉，这样能减少维护索引的成本。优化 SQL 语句的写法也很关键。尽量别用太复杂的子查询或者关联查询，能拆成几个简单的语句来执行就拆。还有，对于那些经常执行、结果又不会变的查询，可以开启查询缓存功能，这样能让查询响应更快一些。数据库服务器的硬件配置和网络环境也不能忽略。得确保服务器有足够的资源来处理查询请求，同时尽量减少网络传输时的延迟对查询性能的影响。

在索引优化这一块，除了前面说的合理创建和删除索引，还可以用复合索引来应对多条件查询的情况。不过要注意复

合索引里字段的顺序，得把那些选择性高的字段放在前面。

SQL 语句优化方面，除了避免复杂查询和合理拆分，还可以用 EXPLAIN 命令来分析查询的执行计划，找出哪里有性能瓶颈，再针对性地优化。对于数据量很大的查询，可以采用分页的方式，每次只查一部分数据，这样能提高查询效率。硬件配置上，可以根据查询负载的特点，对服务器的 CPU、内存、磁盘等进行合理的配置和升级，满足查询性能的需求。

查询缓存的使用也有讲究。除了开启功能，还要合理设置缓存的大小，别让缓存太大占用太多内存。同时，要定期清理缓存里过期或者没用的数据。网络环境优化方面，可以用高速稳定的网络设备，合理规划网络的拓扑结构，减少网络节点和传输距离，还要对网络带宽进行合理分配和监控，确保查询请求在传输过程中不会被阻碍。另外，通过优化数据库的参数配置也能提升查询性能，比如调整缓冲池大小、排序缓冲区大小等。不过这些参数得根据服务器实际的硬件情况和查询负载来精细调整。

从数据类型选择的角度也能进行优化。创建表的时候，要根据实际需求选合适的数据类型，别用太大或者不必要的字段类型，这样能减少数据的存储空间，提高查询时读取数据的速度。对于经常作为查询条件的字段，除了建索引，还可以考虑把它单独拿出来建一张小表，通过关联查询来获取数据，这样查询时能减少扫描数据的范围。

处理关联查询时，要确保关联字段上有索引，而且关联条件要尽量简洁，别用复杂的函数或者表达式来关联，以免影响性能。一些复杂的统计查询，可以先把中间结果存在临时表里，再基于临时表进行后续查询，这样能避免重复计算，提高效率。还要定期对数据库进行维护，比如优化表结构、更新表的统计信息等，让数据库更了解数据的分布情况，从而生成更优的查询执行计划。

4.2 ClickHouse 查询性能调优

合理设计表结构能提升数据读取和查询的效率，比如选择合适的列存储格式。常用查询字段可以建立索引，这样查询时定位数据会更快。写查询语句时，别用太复杂的逻辑，尽量简化条件，少做不必要的计算和处理。ClickHouse 有并行查询的能力，调整好并行度，能让服务器的多核性能充分发挥，整体查询速度会提高。周期性的查询任务，可以提前把数据处理好存起来，以减少实时查询的计算量。

ClickHouse 的配置参数也能优化，比如调整内存分配的比例、优化磁盘读写的策略，这样能适应不同的查询负载情况。数据存储方面，根据重要性和访问频率分开存，把经常

用的热数据放在性能好的存储介质上, 查询响应会更快。定期维护 ClickHouse 的表也很重要, 比如执行优化表的操作, 整理数据碎片, 让数据存储更连续, 查询效率也会提高。还要监控查询的性能指标, 像响应时间、资源利用率这些, 根据监控结果调整优化策略。

分布式环境下, ClickHouse 的查询调优要考虑集群的负载均衡。设计好分片和副本的策略, 让查询请求均匀分到各个节点, 别让某个节点过载, 整体性能会更好。物化视图功能也能用上, 预先计算并存储复杂查询的结果, 查询响应会更快。特别复杂的查询, 可以拆成多个子查询, 在应用层合并结果, 这样单个查询的复杂度会降低。硬件上选高性能的 SSD 硬盘存数据, 能明显提升磁盘 I/O 性能, 查询速度也会加快。服务器内存也要足够, 这样 ClickHouse 才能用内存缓存数据、处理查询。

分析查询执行计划也能优化性能, 用 EXPLAIN 语句看看查询的执行路径, 找出全表扫描、低效连接这些瓶颈, 针对性优化。数据更新不频繁的话, 可以用字典表功能, 把常查的维度数据压缩存储, 减少读取量, 还能提高一致性和准确性。字典表的更新策略也要设置好, 保证数据时效性。大量数据的聚合查询, 可以用近似计算功能, 在保证精度的前提下减少计算量。比如用 uniqApprox 函数做近似去重计数, 比精确的 uniq 函数更快。日志配置也要优化, 设置合适的日志级别和存储路径, 别让太多日志影响性能, 定期清理旧日志释放空间。网络层面要保证集群节点间连接稳定、带宽充足, 减少延迟和丢包, 数据传输效率会提高, 尤其是分布式查询时, 网络对速度影响更大。还要关注版本更新, 及时升级到最新稳定版, 新版本会修复性能问题, 引入新的优化功能。

五、实时数据处理与分析

5.1 实时数据流的处理机制

在 MySQL 和 ClickHouse 混合存储的架构里, 实时数据流动的处理方式起着关键作用。要让数据既及时又准确, 系统就得有高效的接收管道, 能快速抓取来自各种源头的实时数据。这些数据经过初步整理和清理后, 会直接存到 ClickHouse 里。ClickHouse 的列式存储和多任务并行处理能力很强, 用它来做实时分析特别合适。系统还设计了灵活的路由策略, 能根据数据本身的特点和查询时的具体需求, 自动把数据分到最合适的存储节点上。这样做能让查询速度更快, 性能更好。另外, 考虑到数据流量可能突然变大, 系统还能动态调整规模、平衡负载。就算很多用户同时访问, 也能保持稳定的处理速度, 不会出问题。

5.2 ClickHouse 在实时数据分析中的应用

ClickHouse 在实时数据分析领域表现出很强的优势, 这主要得益于它采用的列式存储方式和高效的并行计算能力。它能让实时数据快速写入系统, 用户也能随时查询数据, 第一时间拿到最新的分析结果。利用 ClickHouse 的实时聚合功能, 系统可以对大量数据进行实时统计和分析, 给业务决策提供有力支持。而且, ClickHouse 还配备了丰富的函数库和查询接口, 用户用这些工具能更方便地做复杂的数据分析和挖掘, 让实时数据分析的灵活性和准确性都得到了提升。

结语

在当今数据驱动的时代, 很多业务场景都需要处理和查询海量数据, 这已经成了核心需求。我们搭建了 MySQL 和 ClickHouse 结合的混合存储架构, 这样既能用好 MySQL 在事务处理上的长处, 也能发挥 ClickHouse 在实时数据分析和大规模数据查询方面的好性能。这种新的架构设计, 让数据处理的速度变快了, 也给业务决策提供了更及时、更准确的数据支持。未来, 数据量会越来越大, 业务需求也会不断变化, 我们会继续研究和优化这种混合存储架构, 来应对更复杂、更多样的数据处理难题, 助力企业在激烈的市场竞争中脱颖而出。

参考文献

- [1] 桂智明; 孙忠祥; 黄舟. 区块链与数据库混合架构下的空间查询优化方法[J]. 北京大学学报(自然科学版). 2023, 59(02): 261-270.
- [2] FabricSQL: 区块链数据的关系查询[J]. 余涛; 牛保宁; 樊星. 计算机工程与设计, 2022(10): 2988-2995.
- [3] 杨育运. 基于区块链技术的分布式数据存储与处理研究[J]. 信息记录材料. 2024, 26(10): 142-144.
- [4] 孙伟. 新一代信息技术在大数据处理中的作用与挑战[J]. 石河子科技. 2024(05): 27-29.
- [5] 童晓涛. 基于大数据的数据处理与分析技术. 数字技术与应用, 2024(03): 208-210.