

19 | 跨系统实时同步数据，分布式事务是唯一的解决方案吗？

李玥 · 后端存储实战课



你好，我是李玥。

我们在《[🔗 15 | MySQL 存储海量数据的最后一招：分库分表](#)》这节课中讲过，数据量太大的时候，单个存储节点存不下，那就只能把数据分片存储。

数据分片之后，我们对数据的查询就没那么自由了。比如订单表如果按照用户 ID 作为 Sharding Key 来分片，那就只能按照用户维度来查询。如果我是一个商家，我想查我店铺的订单，对不起，做不到了。（当然，强行查也不是不行，在所有分片上都查一遍，再把结果聚合起来，又慢又麻烦，实际意义不大。）

对于这样的需求，普遍的解决办法是用空间换时间，毕竟现在存储越来越便宜。再存一份订单数据到商家订单库，然后以店铺 ID 作为 Sharding Key 分片，专门供商家查询订单。

另外，之前我们在《[🔗 06 | 如何用 Elasticsearch 构建商品搜索系统](#)》这节课也讲到过，同样一份商品数据，如果我们是按照关键字搜索，放在 ES 里就比放在 MySQL 快了几个数量级。

原因是，数据组织方式、物理存储结构和查询方式，对查询性能的影响是巨大的，而且海量数据还会指数级地放大这个性能差距。

所以，在大厂中，对于海量数据的处理原则，都是根据业务对数据查询的需求，反过来确定选择什么数据库、如何组织数据结构、如何分片数据，这样才能达到最优的查询性能。同样一份订单数据，除了在订单库保存一份用于在线交易以外，还会在各种数据库中，以各种各样的组织方式存储，用于满足不同业务系统的查询需求。像 BAT 这种大厂，它的核心业务数据，存个几十上百份是非常正常的。

那么问题来了，如何能够做到让这么多份数据实时地保持同步呢？

我们之前讲过分布式事务，可以解决数据一致性的问题。比如说，你可以用本地消息表，把一份数据实时同步给另外两、三个数据库，这样还可以接受，太多的话也是不行的，并且对在线交易业务还有侵入性，所以分布式事务是解决不了这个问题的。

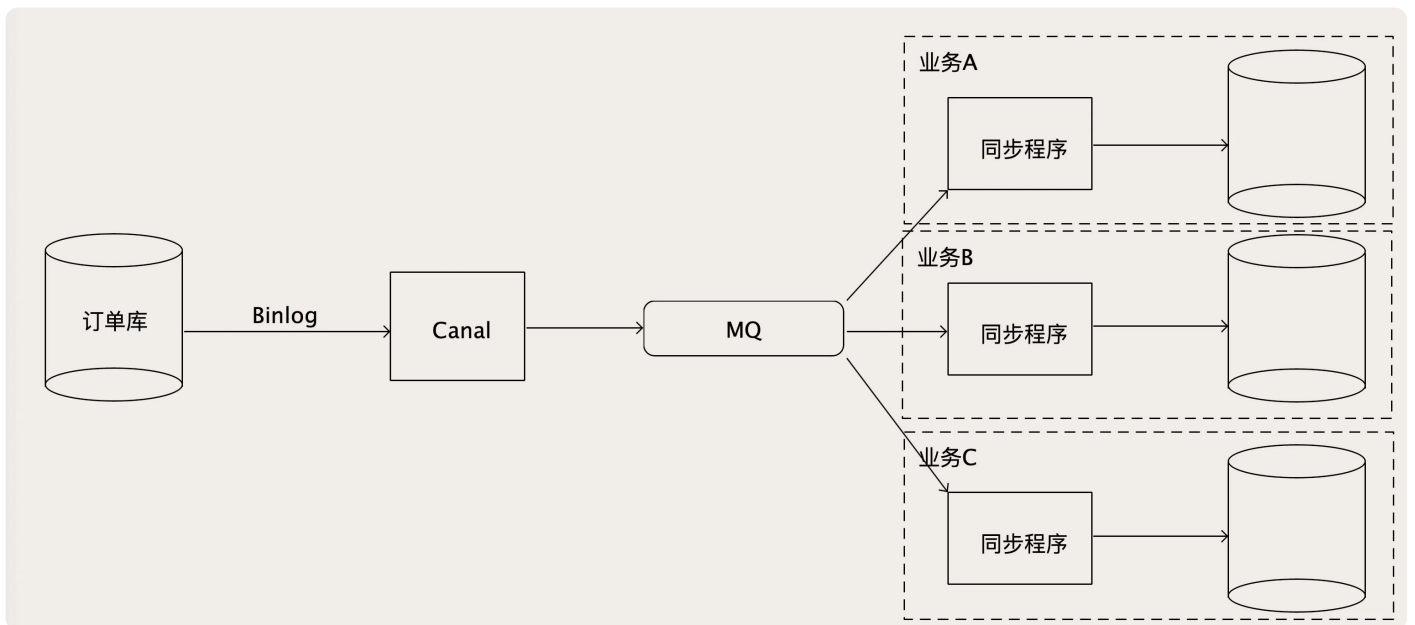
今天这节课我们就来说一下，如何把订单数据实时、准确无误地同步到这么多异构的数据中去。

使用 Binlog 和 MQ 构建实时数据同步系统

早期大数据刚刚兴起的时候，大多数系统还做不到异构数据库实时同步，那个时候普遍的做法是，使用 ETL 工具定时同步数据，在 T+1 时刻去同步上一个周期的数据，然后再做后续的计算和分析。定时 ETL 对于一些需要实时查询数据的业务需求就无能为力了。所以，这种定时同步的方式，基本上都被实时同步的方式给取代了。

怎么来做这么大数据量、这么多个异构数据库的实时同步呢？你还记得我在《[@17 | 大厂都是怎么做 MySQL to Redis 同步的](#)》这节课中讲到的方法吧？利用 Canal 把自己伪装成一个 MySQL 的从库，从 MySQL 实时接收 Binlog 然后写入 Redis 中。把这个方法稍微改进一下，就可以用来做异构数据库的同步了。

为了能够支撑下游众多的数据库，从 Canal 出来的 Binlog 数据肯定不能直接去写下游那么多数据库，一是写不过来，二是对于每个下游数据库，它可能还有一些数据转换和过滤的工作要做。所以需要增加一个 MQ 来解耦上下游。



Canal 从 MySQL 收到 Binlog 并解析成结构化数据之后，直接写入到 MQ 的一个订单 Binlog 主题中，然后每一个需要同步订单数据的业务方，都去订阅这个 MQ 中的订单 Binlog 主题，消费解析后的 Binlog 数据。在每个消费者自己的同步程序中，它既可以直接入库，也可以做一些数据转换、过滤或者计算之后再入库，这样就比较灵活了。

如何保证数据同步的实时性

这个方法看起来不难，但是非常容易出现性能问题。有些接收 Binlog 消息的下游业务，对数据的实时性要求比较高，不能容忍太高的同步时延。比如说，每个电商在大促的时候，都会有一个大屏幕，实时显示现在有多少笔交易，交易额是多少。这个东西都是给老板们看的，如果说大促的时候，你让老板们半小时之后才看到数字，那估计你就得走人了。

大促的时候，数据量大、并发高、数据库中的数据变动频繁，同步的 Binlog 流量也非常大。为了保证这个同步的实时性，整个数据同步链条上的任何一个环节，它的处理速度都必须得跟得上才行。我们一步一步分析可能会出现性能瓶颈的环节。

源头的订单库，如果它出现繁忙，对业务的影响就不只是大屏延迟了，那就影响到用户下单了，这个问题是数据库本身要解决的，这里我们不考虑。再顺着数据流向往下看，Canal 和 MQ 这两个环节，由于没什么业务逻辑，性能都非常好。所以，**一般容易成为性能瓶颈的就是消费 MQ 的同步程序**，因为这些同步程序里面一般都会有一些业务逻辑，而且如果下游的数据库写性能跟不上，表象也是这个同步程序处理性能上不来，消息积压在 MQ 里面。

那我们能不能多加一些同步程序的实例数，或者增加线程数，通过增加并发来提升处理能力呢？这个地方的并发数，还真不是随便说扩容就可以就扩容的，我来跟你讲一下为什么。

我们知道，MySQL 主从同步 Binlog，是一个单线程的同步过程。为什么是单线程？原因很简单，在从库执行 Binlog 的时候，必须按顺序执行，才能保证数据和主库是一样的。**为了确保数据一致性，Binlog 的顺序很重要，是绝对不能乱序的。**严格来说，对于每一个 MySQL 实例，整个处理链条都必须是单线程串行执行，MQ 的主题也必须设置为只有 1 个分区（队列），这样才能保证数据同步过程中的 Binlog 是严格有序的，写到目标数据库的数据才能是正确的。

那单线程处理速度上不去，消息越积压越多，这不无解了吗？其实办法还是有的，但是必须得和业务结合起来解决。

还是拿订单库来说啊，其实我们并不需要对订单库所有的更新操作都严格有序地执行，比如说 A 和 B 两个订单号不同的订单，这两个订单谁先更新谁后更新并不影响数据的一致性，因为这两个订单完全没有任何关系。但是同一个订单，如果更新的 Binlog 执行顺序错了，那同步出来的订单数据真的就错了。

也就是说，我们只要保证每个订单的更新操作日志的顺序别乱就可以了。这种一致性要求称为**因果一致性（Causal Consistency）**，有因果关系的数据之间必须要严格地保证顺序，没有因果关系的数据之间的顺序是无所谓的。

基于这个理论基础，我们就可以并行地来进行数据同步，具体的做法是这样的。

首先根据下游同步程序的消费能力，计算出需要多少并发；然后设置 MQ 中主题的分区（队列）数量和并发数一致。因为 MQ 是可以保证同一分区内，消息是不会乱序的，所以我们需要把具有因果关系的 Binlog 都放到相同的分区中去，就可以保证同步数据的因果一致性。对应到订单库就是，相同订单号的 Binlog 必须发到同一个分区上。

这是不是和之前讲过的数据库分片有点儿像呢？那分片算法就可以拿过来复用了，比如我们可以用最简单的哈希算法，Binlog 中订单号除以 MQ 分区总数，余数就是这条 Binlog 消息发往的分区号。

Canal 自带的分区策略就支持按照指定的 Key，把 Binlog 哈希到下游的 MQ 中去，具体的配置可以看一下 [Canal 接入 MQ 的文档](#)。

小结

对于海量数据，必须要按照查询方式选择数据库类型和数据的组织方式，才能达到理想的查询性能。这就需把同一份数据，按照不同的业务需求，以不同的组织方式存放到各种异构数据库中。因为数据的来源大多都是在线交易系统的 MySQL 数据库，所以我们可以利用 MySQL 的 Binlog 来实现异构数据库之间的实时数据同步。

为了能够支撑众多下游数据库实时同步的需求，可以通过 MQ 解耦上下游，Binlog 先发送到 MQ 中，下游各业务方可以消费 MQ 中的消息再写入各自的数据库。

如果下游处理能力不能满足要求，可以增加 MQ 中的分区数量实现并发同步，但需要结合同步的业务数据特点，把具有因果关系的数据哈希到相同分区上，才能避免因为并发乱序而出现数据同步错误的问题。

思考题

在我们这种数据同步架构下，如果说下游的某个同步程序或数据库出了问题，需把 Binlog 回退到某个时间点然后重新同步，这个问题该怎么解决？欢迎你在留言区与我讨论。

感谢你的阅读，如果你觉得今天的内容对你有帮助，也欢迎把它分享给你的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (25)



李玥 置顶

2020-04-09

Hi，我是李玥。

这里回顾一下上节课的思考题：

对象存储并不是基于日志来进行主从复制的。假设我们的对象存储是一主二从三个副本，采用半同步方式复制数据，也就是主副本和任意一个从副本更新成功后，就给客户端返回成功响应。主副本所在节点宕机之后，这两个从副本中，至少有一个副本上的数据是和宕机的主副本上是一样的，我们需要找到这个副本作为新的主副本，才能保证宕机不丢数据。但是没有了日志，如果这两个从副本上的数据不一样，我们如何确定哪个上面的数据是和主副本一样新呢？

这个问题有些同学已经在留言区给出了答案，一般都是基于版本号来解决，在Leader上，KEY每更新一次，KEY的版本号就加1，版本号作为KV的一个属性，一并复制到从节点上，通过比较版本号就可以知道哪个节点上的数据是最新的。

另外，有的同学提出用比较时间戳的方式来解决这个问题。这个方法理论上可行，但实际上非常难实现，因为它要求集群上的每个节点的时钟都必须时刻保持同步，这个要求往往非常难达到。

共 7 条评论 >

👍 36



豆腐居士

2020-04-09

如果预估了分区（队列）数量之后 随着业务数据的增长 需要增加分区 提高并发 怎么去做扩容？

因为统一笔订单需要打到同一个分区上

- 作者回复: 1. 停掉Canel;
2. 等MQ中所有的消息都消费完了。
3. 扩容MQ分区数，增加消费者实例数量。
4. 重新启动Canel。

共 10 条评论 >

👍 39



木头发芽

2020-04-12

有点像cpu取指令的冒险，如果当前指令后n条指令跟当前指令没有上下文依赖就可以放入指令流水里并行执行力。计算机科学的各种理论真是到处都在用，学好基础是关键



👍 27



Geek_772139

2020-05-26

今把binlog回退到某个时间点开始重新同步，这个需要mq消费端的消费进度支持重置，重置到过去的某一个消费进度就可以了

作者回复: 本身row格式的binlog就是幂等的，mq也要求消费者必须具备幂等性。

所以，自然就支持重置。



17



Simon

2020-04-09

老师，请问下都用mq了还能是实时同步数据嘛？

作者回复: 一般使用MQ，也可以做到秒级延迟。



11



VincentJiang

2020-06-09

请问老师，如果应用跨云（AWS和阿里）部署，并且使用的数据库不是MySQL而是PG，有什么好方法可以实时这种跨云数据同步？

作者回复: PG也有WAL，和MySQL的Binlog是类似的。

你可以参考一下这个开源项目：<https://github.com/debezium/debezium>



10



飞翔

2020-04-11

老师 mq 可以有多个 sharding key 是订单号，这样同一个订单号就可以保证到同一个mq里边去，保证顺序，但是canal不还是必须只有一个 不会成为瓶颈嘛

作者回复: 一般Canal是不会成为瓶颈的，你想，MySQL的主从同步也是单线程的，正常情况下都不会有延迟的。

共 3 条评论 >

8



此方彼方Francis

2020-04-09

这节课感觉可以和MySQL同步数据到redis那节合起来。

共 1 条评论 >



7



陆老师

2020-04-24

越到后面，留言的越少了。下游的某个同步程序或数据库出了问题，可以抛出异常不确认消息，这样，等数据库好了，再次进行消费，不过这样性能会差点，数据也有延迟。如果不想影响多个系统共用的MQ，可以把数据再发送到某个业务系统单独的MQ中去，后续自己单独慢慢消费



5



阿甘

2022-01-21

> MySQL 主从同步 Binlog，是一个单线程的同步过程。为什么是单线程？原因很简单，在从库执行 Binlog 的时候，必须按顺序执行，才能保证数据和主库是一样的。为了确保数据一致性，Binlog 的顺序很重要，是绝对不能乱序的。

这是以前的MySQL了。从MySQL 5.6 版本起可以开启并行复制功能（`slave_parallel_workers > 0`）实现多库并行化复制；MySQL 5.7 引入了MTS（Enhanced Multi-threaded slaves）来实现并行复制，不再有基于库的并行复制限制，5.7.2 又进行了优化，增加了参数`slave_parallel_type`。



3



永不止步

2020-04-16

重新同步的话，下游的消费者需要满足幂等性，保证同一条记录只处理一次



2



那一刻

2020-04-09

使用MQ的消息回放功能？



2



凯文小猪

2021-12-15

二刷课程 谈一下心得体会：

这里canal分发mq主题采用的思想 和lamport老爷子的happened before是一样的：

- 1.两个共识或事件如果有因果关系 那么应该将其组织成偏序关系 发往一个队列 。这一点正巧和现代mq的分区思想吻合
- 2.没有因果关系的事件 因为空间时刻特性无法确定先后关系 所以干脆就并行分发到不同分区 释放掉压力



芋头

2023-03-27 来自广东

订单数据不适合存储在Es中直接查询吗，这样没必要做这么多同步了？



ifelse

2022-12-13 来自浙江

对于海量数据，必须要按照查询方式选择数据库类型和数据的组织方式，才能达到理想的查询性能。--记下来



大碗碗小婉婉

2021-12-30

同步es也用mq吗，看官网有一个adapter



凯文小猪

2021-04-15

感谢老师的精彩介绍 对于因果一致性有了全新的认识，对于解决工作中的实际问题有了自信。最后谈下老师的问题：实际上解决思路就和之前老师说的一样 就是定期做snapshot 再讲mq当做binlog 利用mq的消息回溯来来回回擦数据就行了。唯一要注意的是消费者要做好幂等 不过这应该是互联网开发最基本的要求了吧



djfhchdh

2020-09-20

这个下游的同步程序需要支持binlog同步的幂等操作，对于同一条binlog消息，要保证幂等性。

共 1 条评论 >



victor

2020-09-07

多张表合并成一张宽表是如何同步哪？



苗

2020-08-28

客户端的多个线程，每个线程绑定一个队列并行处理吗？

