# 22-面对海量数据，如何才能查得更快-

你好，我是李玥。

我们接着上节课的话题，来继续说海量数据。上节课我们讲了，如何来保存原始数据，那我们知道，原始数据的数据量太大了，能存下来就很不容易了，这个数据是没法直接来给业务系统查询和分析的。有两个原因，一是数据量太大了，二是也没有很好的数据结构和查询能力，来支持业务系统查询。

所以一般的做法是，用流计算或者是批计算，把原始数据再进行一次或者多次的过滤、汇聚和计算，把计算结果落到另外一个存储系统中去，由这个存储再给业务系统提供查询支持。这里的“流计算”，指的是Flink、Storm这类的实时计算，批计算是Map-Reduce或者Spark这类的非实时计算。

上节课我们说过，像点击流、监控和日志这些原始数据是“海量数据中的海量数据”，这些原始数据经过过滤汇总和计算之后，大多数情况下数据量会有量级的下降，比如说从TB级别的数据量，减少到GB级别。

有的业务，计算后的数据非常少，比如说一些按天粒度的汇总数据，或者排行榜类的数据，用什么存储都能满足要求。那有一些业务，没法通过事先计算的方式解决全部的问题。原始数据经过计算后产生的计算结果，数据量相比原始数据会减少一些，但仍然是海量数据。并且，我们还要在这个海量数据上，提供性能可以接受的查询服务。

今天这节课我们就来聊一聊，面对这样的海量数据，如何才能让查询更快一些。

## 常用的分析类系统应该如何选择存储?

查询海量数据的系统，大多都是离线分析类系统，你可以简单地理解为类似于做报表的系统，也就是那些主要功能是对数据做统计分析的系统。这类系统是重度依赖于存储的。选择什么样的存储系统、使用什么样的数据结构来存储数据，直接决定了数据查询、聚合和分析的性能。

分析类系统对存储的需求一般是这样的：

1. 一般用于分析的数据量都会比在线业务大出几个数量级，这需要存储系统能保存海量数据；
2. 能在海量的数据上做快速的聚合、分析和查询。注意这里面所说的“快速”，前提是处理GB、TB甚至PB级别的海量数据，在这么大的数据量上做分析，几十秒甚至几分钟都算很快了，和在线业务要求的毫秒级速度是不一样的；
3. 由于数据大多数情况下都是异步写入，对于写入性能和响应时延，一般要求不高；
4. 分析类系统不直接支撑前端业务，所以也不要求高并发。

然后我们看有哪些可供选择的存储产品。如果你的系统的数据量在GB量级以下，MySQL仍然是可以考虑的，因为它的查询能力足以应付大部分分析系统的业务需求。并且可以和在线业务系统合用一个数据库，不用做ETL（数据抽取），省事儿并且实时性好。这里还是要提醒你，最好给分析系统配置单独的MySQL实例，避免影响线上业务。

如果数据量级已经超过MySQL极限，可以选择一些列式数据库，比如：HBase、Cassandra、ClickHouse，这些产品对海量数据，都有非常好的查询性能，在正确使用的前提下，10GB量级的数据查询基本上可以做到秒级返回。高性能的代价是功能上的缩水，这些数据库对数据的组织方式都有一些限制，查询方式上也没有MySQL那么灵活。大多都需要你非常了解这些产品的脾气秉性，按照预定的姿势使用，才能达到预期的性能。

另外一个值得考虑的选择是Elasticsearch（ES），ES本来是一个为了搜索而生的存储产品，但是也支持结构化数据的存储和查询。由于它的数据都存储在内存中，并且也支持类似于Map-Reduce方式的分布式并行查询，所以对海量结构化数据的查询性能也非常好。

最重要的是，ES对数据组织方式和查询方式的限制，没有其他列式数据库那么死板。也就是说，ES的查询能力和灵活性是要强于上述这些列式数据库的。在这个级别的几个选手中，我个人强烈建议你优先考虑ES。但是ES有一个缺点，就是你需要给它准备大内存的服务器，硬件成本有点儿高。

数据量级超过TB级的时候，对这么大量级的数据做统计分析，无论使用什么存储系统，都快不到哪儿去。这个时候的性能瓶颈已经是磁盘IO和网络带宽了。这种情况下，实时的查询和分析肯定做不了。解决的办法都是，定期把数据聚合和计算好，然后把结果保存起来，在需要时对结果再进行二次查询。这么大量级的数据，一般都选择保存在HDFS中，配合Map-Reduce、Spark、Hive等等这些大数据生态圈产品做数据聚合和计算。

### 转变你的思想：根据查询来选择存储系统

面对海量数据，仅仅是根据数据量级来选择存储系统，是远远不够的。

经常有朋友会问：“我的系统，每天都产生几个GB的数据量，现在基本已经慢得查不出来了，你说我换个什么数据库能解决问题呢？”那我的回答都是，对不起，换什么数据库也解决不了你的问题。为什么这么说呢？

因为在过去的几十年里面，存储技术和分布式技术，在基础理论方面并没有什么本质上突破。技术发展更多的是体现在应用层面上，比如说，集群管理简单，查询更加自动化，像Map-Reduce这些。不同的存储系统之间，并没有本质的差异。它们的区别只是，存储引擎的数据结构、存储集群的构建方式，以及提供的查询能力，这些方面的差异。这些差异，使得每一种存储，在它擅长的一些领域或者场景下，会有很好的性能表现。

比如说，最近很火的RocksDB、LevelDB，它们的存储结构LSM-Tree，其实就是日志和跳表的组合，单从数据结构的时间复杂度上来说，和“老家伙”MySQL采用的B+树，有本质的提升吗？没有吧，时间复杂度都是O(log n)。但是，LSM-Tree在某些情况下，它利用日志有更好的写性能表现。没有哪种存储能在所有情况下，都具有明显的性能优势，所以说，**存储系统没有银弹，不要指望简单地更换一种数据库，就可以解决数据量大，查询慢的问题。**

但是，在特定的场景下，通过一些优化方法，把查询性能提升几十倍甚至几百倍，这个都是有可能的。这里面有个很重要的思想就是，**根据查询来选择存储系统和数据结构**。我们前面的课程《06 | 如何用Elasticsearch构建商品搜索系统》，就是把这个思想实践得很好的一个例子。ES采用的倒排索引的数据结构，并没有比MySQL的B+树更快或者说是更先进，但是面对“全文搜索”这个查询需求，选择使用ES的倒排索引，就比使用其他的存储系统和数据结构，性能上要高出几十倍。

再举个例子，大家都知道，京东的物流速度是非常快的。经常是，一件挺贵的衣服，下单之后，还没来得及后悔，已经送到了。京东的物流之所以能做到这么快，有一个很重要的原因是，它有一套智能的补货系统，根据历史的物流数据，对未来的趋势做出预测，来给全国每个仓库补货。这样京东就可以做到，你下单买的商品，很大概率在离你家几公里那个京东仓库里就有货，这样自然很快就送到了。这个系统的背后，它需要分析每天几亿条物流数据，每条物流数据又细分为几段到几十段，那每天的物流数据就是几十亿的量级。

这份物流数据，它的用途也非常多，比如说，智能补货系统要用；调度运力的系统也要用；评价每个站点

儿、每个快递小哥的时效达成情况，还要用这个数据；物流规划人员同样要用这个数据进行分析，对物流网络做持续优化。

那用什么样的存储系统保存这些物流数据，才能满足这些查询需求呢？显然，任何一种存储系统，都满足不了这么多种查询需求。我们需要根据每一种需求，去专门选择合适的存储系统，定义适合的数据结构，各自解决各自的问题。而不是用一种数据结构，一个数据库去解决所有的问题。

对于智能补货和运力调度这两个系统，它的区域性很强，那我们可以把数据按照区域（省或者地市）做分片，再汇总一份全国的跨区物流数据，这样绝大部分查询都可以落在一个分片上，查询性能就会很好。

对于站点儿和人的时效达成情况，这种业务的查询方式以点查询为主，那可以考虑事先在计算的时候，按照站点儿和人把数据汇总好，存放到一些分布式KV存储中，基本上可以做到毫秒级查询性能。而对于物流规划的查询需求，查询方式是多变的，可以把数据放到Hive表中，按照时间进行分片。

我们之前也讲到过，按照时间分片是对查询最友好的分片方式。物流规划人员可以在上面执行一些分析类的查询任务，一个查询任务即使是花上几个小时，用来验证一个新的规划算法，也是可以接受的。

## 小结

海量数据的主要用途就是支撑离线分析类业务的查询，根据数据量规模不同，由小到大可以选择：关系型数据库，列式数据库和一些大数据存储系统。对于TB量级以下的数据，如果可以接受相对比较贵的硬件成本，ES是一个不错的选择。

对于海量数据来说，选择存储系统没有银弹，重要的是转变思想，根据业务对数据的查询方式，反推数据应该使用什么存储系统、如何分片，以及如何组织。即使是同样一份数据，也要根据不同的查询需求，组织成不同的数据结构，存放在适合的存储系统中，才能在每一种业务中都达到理想的查询性能。

## 思考题

今天的课后思考题是这样的，我们要做一个日志系统，收集全公司所有系统的全量程序日志，给开发和运维人员提供日志的查询和分析服务，你会选择用什么存储系统来存储这些日志？原因是什么？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。

## 精选留言：

- 李玥 2020-04-16 11:17:40
  Hi，我是李玥。

  这里回顾一下上节课的思考题：

  课后请你想一下，为什么 Kafka 能做到几倍于 HDFS 的吞吐能力，技术上的根本原因是什么？

  答案：

  这个问题的最根本原因是，对于磁盘来说，顺序读写的性能要远远高于随机读写，这个性能差距视不同的磁盘，大约在几十倍左右。Kafka是为顺序读写设计的，儿HDFS是为随机读写的设计的，所以在顺序写入的时候，Kafka的性能会更好。 [1赞]

- 1 2020-04-16 18:03:20
  我对内存数据库有个疑问，是启动之后他会把放到硬盘的数据放到内存里？还是查询过一次之后把结果放到内存里

- 一步 2020-04-16 11:37:13
  对于思考题，会选择 ES 作为存储系统，这里是因为是
  1: 日志一般是根据时间线来保存的，而且不用保存历史的数据，只需要保存最近 15天或者 7天的数据就可以满足要求，数量量不是很大
  2: 查看日志的时候一般都会使用全文搜索， ES 可以高效的支持全文搜索

- hello 2020-04-16 11:02:07
  时序数据库，如Influxdb

- 那时刻 2020-04-16 10:27:19
  我们采取的方式是，最近的三天日志存在es里，旧的数据存在S3，查询的时候使用spectrum

- Wind 2020-04-16 09:33:55
  如果数据量不是太大，我会选ELK

- leslie 2020-04-16 09:09:16
  DB和OPS从业多年越来越觉得很难单一用某套系统去解决问题，记得老师在消息队列的课程提及过；消息队列的作用是削峰填谷，最近在思考"中台"真实的作用是什么？是不是真的偶然？是不是就是由于现在单一无力解决而造就了中台的诞生。
  mysql早期的分引擎处理其实比现在更合理，5.7开始读写都一套引擎反而限制了；虽然业界普遍认为8更好，可是个人觉得早期的做法避免了跨库；读写全部依赖一种东西是不可能真的做到平衡的。就像老师文中提及“需求决定数据库的选择”，日志系统其实合适的很多关键还是要看怎么操作；记得曾经听说过有一套数据库是基本不做DML的，只做查询性能极其好。
  "需求决定选择"我觉得这才是现在对于DB这块最合理的选择。
  谢谢老师的分享，期待后续的课程。

- 墨雨 2020-04-16 08:46:18
  还是看量级,课程中已有说到，少－mysql,中－ES,多－HDFS