关于性能优化的一点思考

作者：Not-Found
原文链接：https://ld246.com/article/1578211117915
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

背景

  最近，在做一个项目需要对某些数据进行处理并且提供API接口查询。该项目天的数据量，经过压缩后大概是10G左右。而且这个项目对于实时性要求很高，不宜直接使用Mysql读写数据。

  于是采用了Redis的集群，把每天的实时数据存在Redis里面，再通过别的方将旧数据存放在Mysql里面做一个备份。

第一次尝试：数据量测试

  把存储数据的方式确定为Redis后，我们开始了第一次的尝试，对数据的容量一次测试，对每天的数据都是通过JSON的形式存在Redis中。本以为40G内存的Redis完全可以应付天的数据。结果，一天的数据还没接收完，Redis就因为可使用的内存不足导致宕机。没办法，可用内存有限，不得不将数据进行压缩。

使用Protobuf

  经过一次数据容量的测试，发现以JSON的数据结构存在Redis中太浪费内存，于是又经过一轮讨论，我们决定使用Protobuf，将数据通过Protobuf序列化存在Redis中。改造完，又进行了一次压力测试，终于Redis集群没有再次宕机。并且，将原来每天最少40G的数据量，成的减少到了10G，也就是说减少了4倍的内存占用。

第二次尝试：API接口的压力测试

  搞定了接收数据的程序，并且也搞定了第一版的API接口，于是我们对API进了一次压力测试。

第一次测试

API服务器配置为8核32G，压测服务器8和32G，使用wrk工具进行简单的压力测试

一共有3个接口，QPS测试结果（大致数据）：

- 当日数据接口（217/QPS）
- 五日数据接口（30/QPS）
- 详细数据接口（4000+/QPS）

初次测试的结果并不是很理想，特别是当日数据和五日数据接口，都没有达到测试的标准。并且这两接口每次请求都会对数据进行一次计算，于是继续改造。将需要算的数据，都放在接收数据的程序中API只是获取数据。

第二次测试

把API计算的部分放在了接收的程序中，让接收程序进行计算，不让API每次都进行计算。并且再次进了一次对当日数据和五日数据接口的压力测试，测试结果如下：

- 当日数据接口（340+/QPS）
- 五日数据接口（300+/QPS）

通过这次的压力测试，可以发现这两个接口的有一定的提升，也就是说这次优化是有效的。但，我依不够满意。我发现，这些数据有一个特点，就是旧的数据它不会变，那么就可以对旧的数据在本地内中进行缓存，减少从Redis获取数据发起的网络请求。

第三次测试

  将API接口改为了将历史数据缓存在本地内存中，并且减少了发起网络请求的数。再次进行压力测试，测试结果如下：

- 当日数据接口（10000+/QPS）
- 五日数据接口（6400+/QPS）

可以发现，对旧的历史数据缓存，减少发起网络请求的优化之后，QPS有明显的提升。

总结

在对程序性能的分析时，使用了阿里开源的Arthas工具，发现最影响程序性能的地方，就是从Redis取数据时发起的网络请求最耗时。

经过这次的优化中，我有几点总结：

- 直接将JSON数据存在Redis太浪费，并且API获取数据时，需要的时间会明显的高于Protobuf序列后存在Redis中的数据获取时间。
- 减少API对数据计算的次数，最好不计算，把需要计算的数据放在别的地方，并且存起来
- 减少发起网络请求的次数，因为在这次测试的结果中，经过分析后发现最耗时间也最浪费性能的地就是获取Redis数据时，需要发起的网络请求。

blog:https://www.codedream.xin