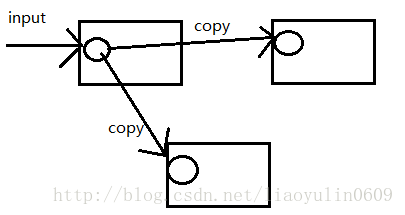
构建ehcache+redis两级缓存

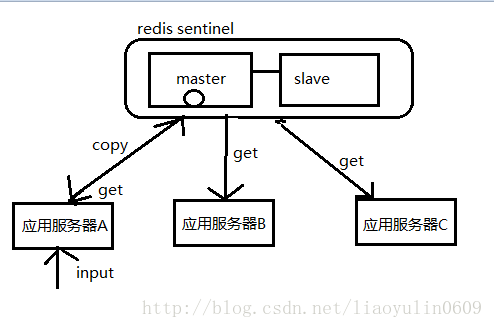
问题描述

场景：我们的应用系统是分布式集群的，可横向扩展的。应用中某个接口操作满足以下一个或多个条件：   
1. 接口运行复杂代价大，   
2. 接口返回数据量大，   
3. 接口的数据基本不会更改，   
4. 接口数据一致性要求不高（只需满足最终一致）。

此时，我们会考虑将这个接口的返回值做缓存。考虑到上述条件，我们需要一套高可用分布式的缓存集群，并具备持久化功能，备选的有**ehcache集群**或**redis主备（sentinel）**。

* ehcache集群因为节点之间数据同步通过组播的方式，可能带来的问题：节点间大量的数据复制带来额外的开销，在节点多的情况下此问题越发严重，N个节点会出现N-1次网络传输数据进行同步。（见下图，缓存集群中有三台机器，其中一台机器接收到数据，需要拷贝到其他机器，一次input后需要copy两次，两次copy是需要网络传输消耗的）   
  

redis主备由于作为中心节点提供缓存，其他节点都向redis中心节点取数据，所以，一次网络传输即可。（当然此处的一次网络代价跟组播的代价是不一样的）但是，随着访问量增大，大量的缓存数据访问使得应用服务器和缓存服务器之间的网络I/O消耗越大。（见下图，同样三台应用服务器，redis sentinel作为中心节点缓存。所谓中心，即所有应用服务器以redis为缓存中心，不再像ehcache集群，缓存是分散存放在应用服务器中，需要互相同步的，任何一台应用服务器的input，都会经过一次copy网络传输到redis，由于redis是中心共享的，那么就可以不用同步的步骤，其他应用服务器需要只需去get取即可。但是，我们会发现多了N台服务器的get的网络开销。）



提出方案

那么要怎么处理呢？所以两级缓存的思想诞生了，在redis的方案上做一步优化，在缓存到远程redis的同时，缓存一份到本地进程ehcache（此处的ehcache不用做集群，避免组播带来的开销），取缓存的时候会先取本地，没有会向redis请求，这样会减少应用服务器<–>缓存服务器redis之间的网络开销。（见下图，为了减少get这几条网络传输，我们会在每个应用服务器上增加本地的ehcache缓存作为二级缓存，即第一次get到的数据存入ehcache，后面output输出即可从本地ehcache中获取，不用再访问redis了，所以就减少了以后get的网络开销。get开销只要一次，后续不需要了，除非本地缓存过期需要再get。）   
