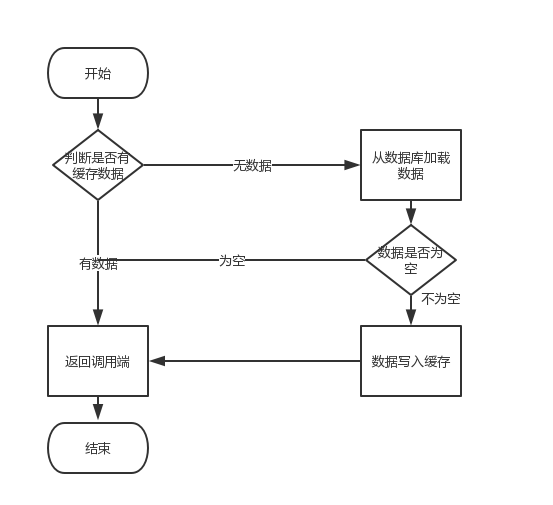
# [【原创】分布式之数据库和缓存双写一致性方案解析](https://www.cnblogs.com/rjzheng/p/9041659.html)

## **引言**

### **为什么写这篇文章？**

首先，缓存由于其高并发和高性能的特性，已经在项目中被广泛使用。在读取缓存方面，大家没啥疑问，都是按照下图的流程来进行业务操作。  
  
但是在更新缓存方面，对于更新完数据库，是更新缓存呢，还是删除缓存。又或者是先删除缓存，再更新数据库，其实大家存在很大的争议。

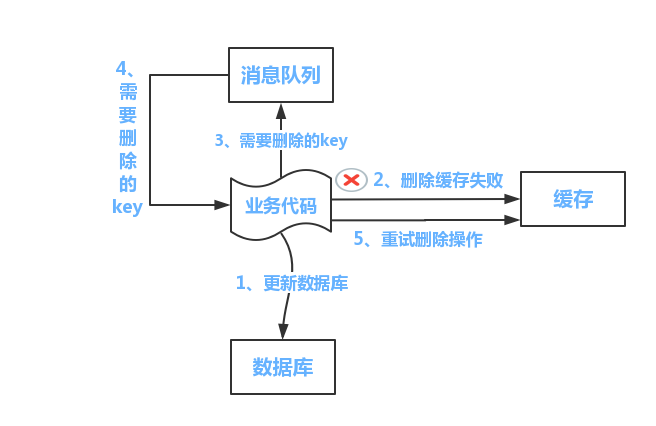
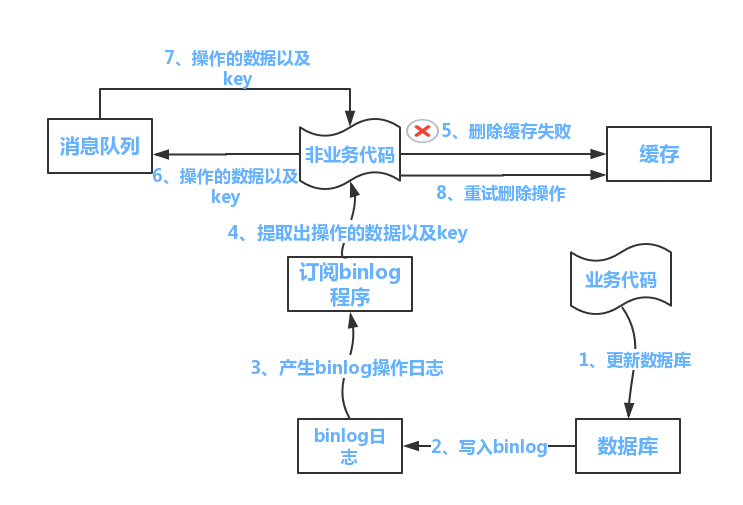
### **(1)先删缓存，再更新数据库 （不推荐）**

该方案会导致不一致的原因是。同时有一个请求A进行更新操作，另一个请求B进行查询操作。那么会出现如下情形:  
（1）请求A进行写操作，删除缓存  
（2）请求B查询发现缓存不存在  
（3）请求B去数据库查询得到旧值  
（4）请求B将旧值写入缓存  
（5）请求A将新值写入数据库  
上述情况就会导致不一致的情形出现。而且，如果不采用给缓存设置过期时间策略，该数据永远都是脏数据。

### **(2)先更新数据库，再删缓存（推荐）**

首先，先说一下。老外提出了一个缓存更新套路，名为[《Cache-Aside pattern》](https://docs.microsoft.com/en-us/azure/architecture/patterns/cache-aside)。其中就指出

* **失效**：应用程序先从cache取数据，没有得到，则从数据库中取数据，成功后，放到缓存中。
* **命中**：应用程序从cache中取数据，取到后返回。
* **更新**：先把数据存到数据库中，成功后，再让缓存失效。

另外，知名社交网站facebook也在论文[《Scaling Memcache at Facebook》](https://www.usenix.org/system/files/conference/nsdi13/nsdi13-final170_update.pdf)中提出，他们用的也是先更新数据库，再删缓存的策略。  
**这种情况不存在并发问题么？**  
不是的。假设这会有两个请求，一个请求A做查询操作，一个请求B做更新操作，那么会有如下情形产生  
（1）缓存刚好失效  
（2）请求A查询数据库，得一个旧值  
（3）请求B将新值写入数据库  
（4）请求B删除缓存  
（5）请求A将查到的旧值写入缓存  
ok，如果发生上述情况，确实是会发生脏数据。  
**然而，发生这种情况的概率又有多少呢？**  
发生上述情况有一个先天性条件，就是步骤（3）的写数据库操作比步骤（2）的读数据库操作耗时更短，才有可能使得步骤（4）先于步骤（5）。可是，大家想想，数据库的读操作的速度远快于写操作的（不然做读写分离干嘛，做读写分离的意义就是因为读操作比较快，耗资源少），因此步骤（3）耗时比步骤（2）更短，这一情形很难出现。  
假设，有人非要抬杠，有强迫症，一定要解决怎么办？  
**如何解决上述并发问题？**  
首先，给缓存设有效时间是一种方案。其次，采用策略（2）里给出的异步延时删除策略，保证读请求完成以后，再进行删除操作。  
**还有其他造成不一致的原因么？**  
有的，这也是缓存更新策略（2）和缓存更新策略（3）都存在的一个问题，如果删缓存失败了怎么办，那不是会有不一致的情况出现么。比如一个写数据请求，然后写入数据库了，删缓存失败了，这会就出现不一致的情况了。这也是缓存更新策略（2）里留下的最后一个疑问。  
**如何解决？**  
提供一个保障的重试机制即可，这里给出两套方案。  
**方案一**：  
如下图所示  
  
流程如下所示  
（1）更新数据库数据；  
（2）缓存因为种种问题删除失败  
（3）将需要删除的key发送至消息队列  
（4）自己消费消息，获得需要删除的key  
（5）继续重试删除操作，直到成功  
然而，该方案有一个缺点，对业务线代码造成大量的侵入。于是有了方案二，在方案二中，启动一个订阅程序去订阅数据库的binlog，获得需要操作的数据。在应用程序中，另起一段程序，获得这个订阅程序传来的信息，进行删除缓存操作。  
**方案二**：  
  
流程如下图所示：  
（1）更新数据库数据  
（2）数据库会将操作信息写入binlog日志当中  
（3）订阅程序提取出所需要的数据以及key  
（4）另起一段非业务代码，获得该信息  
（5）尝试删除缓存操作，发现删除失败  
（6）将这些信息发送至消息队列  
（7）重新从消息队列中获得该数据，重试操作。

**备注说明：**上述的订阅binlog程序在mysql中有现成的中间件叫canal，可以完成订阅binlog日志的功能。至于oracle中，博主目前不知道有没有现成中间件可以使用。另外，重试机制，博主是采用的是消息队列的方式。如果对一致性要求不是很高，直接在程序中另起一个线程，每隔一段时间去重试即可，这些大家可以灵活自由发挥，只是提供一个思路。