



17 | 高性能缓存架构

李运华

- 00:00 / 10:33

虽然我们可以通过各种手段来提升存储系统的性能，但在某些复杂的业务场景下，单纯依靠存储系统的性能提升不够的，典型的场景有：

- 需要经过复杂运算后得出的数据，存储系统无能为力

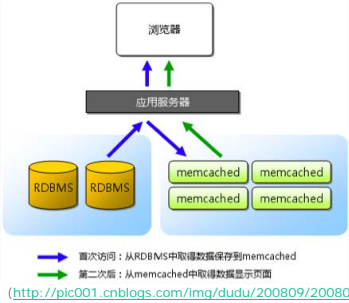
例如，一个论坛需要在首页展示当前有多少用户同时在线，如果使用MySQL来存储当前用户状态，则每次获取这个总数都要“count(*)”大量数据，这样的操作无论怎么优化MySQL，性能都不会太高。如果要实时展示用户同时在线数，则MySQL性能无法支撑。

- 读多写少的数据，存储系统有心无力

绝大部分在线业务都是读多写少。例如，微博、淘宝、微信这类互联网业务，读业务占了整体业务量的90%以上，以微博为例：一个明星发一条微博，可能几千万人来浏览。如果使用MySQL来存储微博，用户写微博只有一条Insert语句，但每个用户浏览时都要select一次，即使有索引，几千万条select语句对MySQL数据库的压力也会非常大。

缓存就是为了弥补存储系统在这些复杂业务场景下的不足，其基本原理是将可能重复使用的数据放到内存中，一次生成、多次使用，避免每次使用都去访问存储系统。

缓存能够带来性能的大幅提升，以Memcache为例，单台Memcache服务器简单的key-value查询能够达到TPS 50000以上，其基本的架构是：



缓存虽然能够大大减轻存储系统的压力，但同时也给架构引入了更多复杂性。架构设计时如果没有针对缓存的复杂性进行处理，某些场景下甚至会导致整个系统崩溃。今天，我来逐一分析缓存的架构设计要点。

缓存穿透

缓存穿透是指缓存没有发挥作用，业务系统虽然去缓存查询数据，但缓存中没有数据，业务系统需要再次去存储系统查询数据。通常情况下有两种情况：

1. 存储数据不存在

第一种情况是被访问的数据确实不存在。一般情况下，如果存储系统中没有某个数据，则不会在缓存中存储相应的数据，这样就导致用户查询的时候，在缓存中找不到对应的数据，每次都要去存储系统中再查询一遍，然后返回数据不存在。缓存在这个场景中并没有起到分担存储系统访问压力的作用。

通常情况下，业务上读取不存在的数据的请求量并不会太大，但如果出现一些异常情况，例如被黑客攻击，故意大量访问某些读取不存在数据的业务，有可能会将存储系统拖垮。

这种情况的解决办法比较简单，如果查询存储系统的数据没有找到，则直接设置一个默认值（可以是空值，也可以是具体的值，存到缓存中，这样第二次读取缓存时就会返回到默认值）。

值，而不会继续访问存储系统。

2.缓存数据生成耗费大量时间或者资源

第二种情况是存储系统中存在数据，但生成缓存数据需要耗费较长时间或者耗费大量资源。如果刚好在业务访问的时候缓存失效了，那么也会出现缓存没有发挥作用，访问压力全部集中在存储系统上的情况。

典型的就是电商的商品分页，假设我们在某个电商平台选择“手机”这个类别查看，由于数据巨大，不能把所有数据都缓存起来，只能按照分页来进行缓存，由于难以预测用户到底会访问哪些分页，因此业务上最简单的就是每次点击分页的时候按分页计算和生成缓存。通常情况下这样实现是基本满足要求的，但是如果被竞争对手用手爬虫来遍历的时候，系统性能就可能出现问题。

具体的场景有：

- 分页缓存的有效期设置为1天，因为设置太长时间的话，缓存不能反应真实的数据。
- 通常情况下，用户不会从第1页到最后1页全部看完，一般用户访问集中在前10页，因此第10页以后的缓存过期失效的可能性很大。
- 竞争对手每周来爬取数据，爬虫会将所有分类的所有数据全部遍历，从第1页到最后1页全部都会读取，此时很多分页缓存可能都失效了。
- 由于很多分页都没有缓存数据，从数据库中生成缓存数据又非常耗费性能（order by limit操作），因此爬虫会将整个数据库全部拖慢。

这种情况并没有太好的解决方案，因为爬虫会遍历所有的数据，而且什么时候来爬取也是不确定的，可能是每天都来，也可能是每周，也可能是一个月来一次，我们也不可能为了应对爬虫而将所有数据永久缓存。通常的应对方案要么是识别爬虫然后禁止访问，但这可能会影响SEO和推广；要么就是做好监控，发现问题后及时处理，因为爬虫不是攻击，不会进行暴力破坏，对系统的影响是逐步的，监控发现问题后有时间进行处理。

缓存雪崩

缓存雪崩是指当缓存失效（过期）后引起系统性能急剧下降的情况。当缓存过期被清除后，业务系统需要重新生成缓存，因此需要再次访问存储系统，再次进行运算，这个处理步骤耗时几十毫秒甚至上百毫秒。而对于一个高并发的业务系统来说，几百毫秒内可能会接到几百上千个请求。由于旧的缓存已经被清除，新的缓存还未生成，并且处理这些请求的线程都不知道另外有一个线程正在生成缓存，因此所有的请求都会去重新生成缓存，都会去访问存储系统，从而对存储系统造成巨大的性能压力。这些压力又会拖慢整个系统，严重的会造成数据库宕机，从而形成一系列连锁反应，造成整个系统崩溃。

缓存雪崩的常见解决方法有两种：**更新锁机制**和**后台更新机制**。

1.更新锁

对缓存更新操作进行加锁保护，保证只有一个线程能够进行缓存更新，未能获取更新锁的线程要么等待锁释放后重新读取缓存，要么就返回空值或者默认值。

对于采用分布式集群的业务系统，由于存在几十上百台服务器，即使单台服务器只有一个线程更新缓存，但几十上百台服务器一起算下来也会有几十上百个线程同时来更新缓存，同样存在雪崩的问题。因此分布式集群的业务系统要实现更新锁机制，需要用到分布式锁，如ZooKeeper。

2.后台更新

由后台线程来更新缓存，而不是由业务线程来更新缓存，缓存本身的有效期设置为永久，后台线程定时更新缓存。

后台定时机制需要考虑一种特殊的场景，当缓存系统内存不够时，会“踢掉”一些缓存数据，从缓存被“踢掉”到下一次定时更新缓存的这段时间内，业务线程读取缓存返回空值，而业务线程本身又不会去更新缓存，因此业务上看到的现象就是数据丢了。解决的方式有两种：

- 后台线程除了定时更新缓存，还要频繁地去读取缓存（例如，1秒或者100毫秒读取一次），如果发现缓存被“踢了”就立刻更新缓存，这种方式实现简单，但读取时间间隔不能设置太长，因为如果缓存被踢了，缓存读取间隔时间又太长，这段时间内业务访问都拿不到真正的数据而是一个空的缓存值，用户体验一般。
- 业务线程发现缓存失效后，通过消息队列发送一条消息通知后台线程更新缓存。可能会出现多个业务线程都发送了缓存更新消息，但其实对后台线程没有影响，后台线程收到消息后更新缓存前可以判断缓存是否存在，存在就不执行更新操作。这种方式实现依赖消息队列，复杂度会高一些，但缓存更新更及时，用户体验更好。

后台更新既适应单机多线程的场景，也适合分布式集群的场景，相比更新锁机制要简单一些。

后台更新机制还适合业务刚上线的时候进行缓存预热。缓存预热指系统上线后，将相关的缓存数据直接加载到缓存系统，而不是等待用户访问才来触发缓存加载。

缓存热点

虽然缓存系统本身的性能比较高，但对于一些特别热点的数据，如果大部分甚至所有的业务请求都命中同一份缓存数据，则这份数据所在的缓存服务器的压力也很大。例如，某明星微博发布“我们”来宣告恋爱了，短时间内上千万的用户都会来围观。

缓存热点的解决方案就是复制多份缓存副本，将请求分散到多个缓存服务器上，减轻缓存热点导致的单台缓存服务器压力。以微博为例，对于粉丝数超过100万的明星，每条微博都可以生成100份缓存，缓存的数据是一样的，通过在缓存的Key里面上加编号进行区分，每次读缓存时都随机读取其中某份缓存。

缓存副本设计有一个细节需要注意，就是不同的缓存副本不要设置统一的过期时间，否则就会出现所有缓存副本同时生成同时失效的情况，从而引发缓存雪崩效应。正确的做法是设定一个过期时间范围，不同的缓存副本的过期时间是指定范围内的随机值。

实现方式

由于缓存的各种访问策略和存储的访问策略是相关的，因此上面的各种缓存设计方案通常情况下都是集成在存储访问方案中，可以采用“程序代码实现”的中间层方式，也可以采用独立的中间件来实现。

小结

今天我为你讲了高性能架构设计中缓存设计需要注意的几个关键点，这些关键点本身在技术上都不复杂，但可能对业务产生很大的影响，轻则系统响应变慢，重则全站宕机，架构师在设计架构的时候要特别注意这些细节，希望这些设计关键点和技术方案对你有所帮助。

这就是今天的全部内容，留一道思考题给你吧，分享一下你所在的业务发生过哪些因为缓存导致的线上问题？采取了什么样的解决方案？效果如何？

欢迎你把答案写到留言区，和我一起讨论。相信经过深度思考的回答，也会让你对知识的理解更加深刻。（编辑乱入：精彩的留言有机会获得丰厚福利哦！）



bluefantasy	2018-06-05
我们的系统就出现过类似的问题，开始的时候没有缓存，每次做活动访问量大的时候就会导致反应特别慢。后来通过加redis缓存解决了问题。	
对于缓存雪崩问题，我们采取了双key策略：要缓存的key过期时间是t，key1没有过期时间。每次缓存读取不到key时就返回key1的内容，然后触发一个事件。这个事件会同时更新key和key1。	
作者回复	2018-06-05
很有创意◆◆◆◆	
loveluckystar	2018-06-05
讲一个头两天发生的事情，我们的一个业务背后是es做db，之前是通过redis做缓存，缓存一段时间后失效再从es读取，是业务访问加载缓存的方式。有一天线上es集群机器单台出现问题，返回慢，由于分布式的缘故，渐渐拖满了所有请求，缓存失效来查询es发生了超时，加载失败，于是下次访问还是直接访问es。最终缓存全部失效，qps翻了好多倍，直接雪崩，es集群彻底没有响应了。。。之后我们只好先下线这个缓存加载功能，让集群活过来，最终改造缓存加载方式，用后台进程去更新缓存，而不用业务访问加载。	
作者回复	2018-06-05
现学现用的案例，很赞◆◆◆◆	
mapping	2018-06-05
计算机界两大难题：命名和缓存过期。没用缓存的时候，想着怎么用缓存提升性能，用了缓存又担心数据更新不及时。技术上希望所有的请求都能命中缓存，业务上又恨不得数据实时最新。所以就会引入各种缓存过期策略，如设置过期时间，按规则删除，打版本。这些应该在前期设计缓存系统时规划好，我们最早是将 sql md5 作 key 查询结果存入缓存，结果业务系统数据不一致，要清除缓存简直是噩梦，只能祭出绝招重启 memcache，后面改成按规则删除。在 key 中加上业务和用户的前缀，可以很方便删除某个业务或某个用户的缓存。以上过期策略在前端浏览器也是这样，最简单就是 web 服务器设置静态资源缓存过期时间，如果业务频繁发新版本，过期时间不宜设置太长，但其实每次变动的文件很少，这种策略会导致大部分缓存命中率不高。按规则删除，早期很多网站上会有诊断助手类的东西，页面加载错误点下诊断助手就帮你清除缓存，原理就是对静态文件逐一带上 no-cache 请求头发送 ajax 请求强制覆盖缓存（跟 DevTools 中 disable cache 原理一样）。打版本其实就相当于让浏览器请求一个新版本文件，对于老版本文件就让它在缓存中自生自灭。	
作者回复	2018-06-05
你们的缓存设计有点复杂，还不如调整业务，越复杂的方案越容易出错，参考架构设计原则	
三月沙@wecatch	2018-06-05
好的缓存方案应该从这几个方面入手设计： 1.什么数据应该缓存 2.什么时候触发缓存和以及触发方式是什么 3.缓存的层次和粒度（网关缓存如 nginx，本地缓存如单机文件，分布式缓存如redis cluster，进程内缓存如全局变量） 4.缓存的命名规则和失效规则 5.缓存的监控指标和故障应对方案 6.可视化缓存数据如 redis 具体 key 内容和大小	
作者回复	2018-06-05
确实，细节不少，可以写本书了◆◆	
◆◆夏天的味道	2018-06-06
线上遇到的一个错误：业务查询的结果序列化后放到redis，下次从redis取出来时报错。原来是结果类虽然实现了Serializable接口，但是没有重写SerialVersionUID，导致不能成功反序列化。	
假强小◆◆	2018-06-05
我们系统是做美术馆的3d展示的，后台配置的数据有点多，画框数据有几m，单个模型数据有几百k，最早直接mysql直接读取，后来用redis但是感觉效果不理想，又引入ehcache，发现经过网络传输性能很好，经过网络传输后网络一般还是卡的很，后来又在前端用local storage进行缓存，后端通过rabbitmq进行消息通知前端清除本地缓存，缓存设置有效期都是一天。请问下我们这种情况有更好的方案吗	
作者回复	2018-06-05
前后端分离，在node上缓存和渲染试试	
刘磊	2018-06-05
对缓存的key也需要进行valid，避免无效的key查询缓存	
脆米豆发	

1、最早也是采用后台用数据库，前台用关系型数据库+被动缓存的模式。结果是经常的性能抖动，且缓存一致性问题很难解决。后来我们的多数系统，都采用了前台+后台的模式，前台用redis

数据仍然是关系型数据库，前台使用缓存作为数据源，两者之间数据实时同步+定时同步+人工触发结合。这个缓存仍然需要了空盘，所以还是需要数据冗余，所以还是需要数据冗余，所以还是需要数据冗余。具备相对可靠的持久化机制+运维体系。

2、遇到大量数据写入，感知到写入性能下降，需要扩容，扩容后写入性能提升，写入性能提升，写入性能提升。这个相对容易，可以对key进行分片，使用hashtag机制分片。第二种情况，数据分布不均匀，较少遇到，遇到相对比较棘手。第三种情况，数据分片不均匀，但访问不均匀，可以增加副本数量。

2018-06-05

2018-06-17

2018-06-17

2018-06-17

2018-06-19

2018-06-19

2018-06-06

2018-06-06

2018-06-06

2018-06-06

2018-06-05

2018-06-05

2018-06-06

2018-06-06

2018-06-05

2018-06-05

2018-06-06

2018-06-06

2018-06-05

2018-06-05

2018-06-05

2018-06-05

2018-06-05

2018-06-05

2018-06-05

2018-06-05

2018-06-11

2018-06-11

2018-06-12

2018-06-12

2018-06-08

2018-06-08

2018-06-06

2018-06-06

2018-06-07

2018-06-07

2018-06-06

2018-06-06

2018-06-06

2018-06-06

2018-06-06

遇到过一次缓存失效导致缓存穿透，很多请求的压力直接到了db。为了处理这种情况，我采用的方案就是：key永不过期，后台有个进程定时更新所有缓存。文中提到的组合键队列来更新具有奇效，当然像看到大神的几个lock key value，也是很有用的。不过版本太高了，相当于成本翻倍。	
作者回复	
除非特殊场景，一般我还是建议尽量用简单直观甚至粗暴的方案💎💎	2018-06-06
jacy	2018-06-05
商品列表中，像商品描述等信息，缓存更新不及时影响不大，但某些重要数据，如价格，需要及时更新的数据，有什么好的办法做到刷新。对于价格这种关键数据，不缓存，直接从数据库查询，是否可行。或者在用户查看商品详情时再去数据库查询价格，但可能出现列表中的价格和详情页中的价格不一致。	
作者回复	2018-06-06
通常有几种做法： 1. 同步刷新缓存：当更新了某些信息后，立刻让缓存失效。 这种做法的优点是用户体验好，缺点是修改一个数据可能需要让很多缓存失效 2. 适当容忍不一致：例如某东的商品就是这样，我查询的时候显示有货，下单的时候提示我没货了 3. 关键信息不缓存：库存，价格等不缓存，因为这类信息查询简单，效率高，关系数据库查询性能也很高	
王磊	2018-06-05
'后台更新既适应单机多线程的场景，也适合分布式集群的场景，相比更新锁机制要简单一些' - 没觉得简单啊，因为如文中所说，后台更新也可能会出现多个业务线程都发送了缓存更新消息，这种情况我理解也需要有分布式就来避免多个相同请求查询存储系统，否则在判断的时候，因为缓存在构建还没有准备好。	
作者回复	2018-06-06
通常情况下，业务线程数量要远远大于后台更新线程数量。假设20台64核机器，每台机器256线程，业务线程就是5120个，后台缓存更新线程数量一般8~32就足够了。 如果缓存设计是只能一个线程更新，那确实也只能用锁了	
王磊	2018-06-05
经常我说到缓存的时候，面试官问我，数据库自身不是有缓存吗，标准答案是怎么回击他？	
作者回复	2018-06-06
我对mysql比较熟，以下仅限mysql： 1. mysql第一种缓存叫sql语句结果缓存，但条件比较苛刻，程序很不可控，我们的dba线上都关闭这个功能，具体实现可以查一下 2. mysql第二种缓存是innodb buffer pool，缓存的是磁盘上的分页数据，不是sql的查询结果，sql的执行过程省不了。而mc，redis这些实际上都是缓存sql的结果，两种缓存方式，性能差很远。	
因此，可控性，性能是数据库缓存和独立缓存的主要区别	
cqc	2018-06-05
1个问题：关于后台更新，既然缓存服务器内存不足，需要删除数据，那么后台更新再次触发查询，是否又会导致其他一些缓存数据被剔除，这感觉像是陷入一个循环了 1个思路：我们之前的web项目，对于缓存热点数据，为了减少服务器的压力，在客户端引入了缓存：CDN+local storage，感觉对服务器端压力分散还是很有效果的。	
作者回复	2018-06-05
1. 是的，所以加内存才是根本解决方式 2. 这是分级缓存策略	
byte	2018-06-05
业务上通过Redis集群缓存网络数据，分布在北上广三地，3个数据中心，集群规模是80台物理机共1000个左右实例1主3从。线上出现过跨地域集群数据频繁全量同步打爆交换机的问题，导致整个服务不可用。通过排查发现是网络延时导致频繁全量同步以及服务器电源过热导致服务器频繁重启。解决方案是跨地数据同步通过kafka试下，再接入各地Redis集群，电源问题通过更换硬件解决。	
作者回复	2018-06-05
缓存集群间一般不要跨数据中心同步，存储可以用跨数据中心同步	
Snway	2018-06-05
订单设置的缓存有效期是永久的，但再一次上线后，更新缓存的后台线程出问题，导致用户查看不了最新订单数据，引起大量客诉！	
作者回复	2018-06-05
重启缓存服务器💎💎	
姜洋昌	2018-06-05
目前我们还没用上缓存技术，所有请求都直接访问了数据库，导致数据库集群压力大，cpu、内存、IO常常过高，请问项目初用缓存，比如redis，有什么指导原则或技巧吗？是缓存热点sql还是热点功能还是根据其他原则缓存呢？	
作者回复	2018-06-05
数据库压力大就可以考虑缓存了	
krugle	2018-07-16
我们的缓存都是下载业务代码里面，有数据不一致的地方要到处翻代码，有什么好的方法解决这个吗，百度了好多没结果	
作者回复	2018-07-17
缓存框架，由框架统一管理缓存,看着Ehcache	
家榆	2018-07-05
我是实现了个框架:https://github.com/qiujiayu/AutoLoadCache，用于解决一问题： 1. 缓存操作与业务代码耦合问题； 2. 缓存穿透问题； 3. 异步在缓存快要过期时，异步刷新缓存； 4. 使用“拿来主义机制”，降低回源并发量；	
Simple_Zhang	2018-07-04
华哥，请问如何保证缓存和数据库的一致性，例如用户修改了一项配置，是先更新缓存还是先更新库？如何保证缓存和库都更新一致呢💎💎	
作者回复	2018-07-04

先更新新库好些，因为更新库成功后即使更新缓存失败，缓存也有过期时间。 如果要保证一致性，更新库前先删除缓存，然后更新，再更新缓存，否则做缓存，只能做到读缓存和库不一致，因此要做读缓存一致是很复杂的，需要用到4k这类协调软件，一般不建议这么做，没必要	
梦里	2018-07-04
做游戏开发，数据先写入redis，用一个线程去读，然后写入mysql。遇到redis满了的情况，就是读的速度赶不上写的。很崩溃 作者回复	2018-07-04
读的时候可以一次读取多个，进行批量操作，性能会好很多	2018-07-04
郭柱	2018-06-27
华哥，我现在来阿里云工作了，有机会希望能拜访华哥，和你分享一下之前我在上家公司做的一个系统。 之前做过一个销售规则系统。这个系统服务于产品查询和销售。查询TPS峰值过万，查定比在100到200之间。规则系统的规则放到缓存中，同时支持规则实时发布到生产系统。性能测试指标是50ms就可以加载到生产系统。上线当晚，我就不停的在做规则修改，几分钟后查询系统变慢，开始大面积报警，当晚查询系统也在上线，后来过了一段时间发现是我们拖慢了整个业务链条，因为我们是新系统被迫下线改造。后来把缓存系统设计为AB缓存，每次加载新规则就加载到备用缓存中假设当前为B缓存，然后从日志中取1000生产查询请求，去查询最新更新的缓存，当100%成功后，将生产请求切换到B缓存，实际时间也就是秒级切换完成，这样保障了销售规则系统正常运转下去的，目前也还是这个架构。这种不考虑成本的方案是不是只有在国企才会用到？ 作者回复	2018-06-28
互联网用得更多啊，现在内存不值钱了，缓存大胆💎💎	2018-06-28
孙振超	2018-06-24
在对qps有高要求的系统通常会采用缓存，正所谓解决了一个问题之后就会有有一个新问题出现，引入缓存提升qps后最大的问题在于缓存db数据一致性问题。使用缓存的通常做法是先读缓存，缓存没有命中，就查询db，后将查询到的数据添加到缓存中。在这种使用方法下导致缓存db不一致的主要情况是db和缓存没有同步更新，比如先更新db后更新缓存可能会出现db更新成功而缓存更新失败的情况，如果是先更新缓存后更新db则可能会出现缓存更新成功而db更新失败的情况。而解决这个问题可能要引入消息中间件或者利用分布式锁来保证，这样又给系统带来了复杂性。	2018-06-24
想飞的鱼	2018-06-22
我 off 业务中涉及到很多金融各个分类的数据(股票，基金，债券，期货，宏观...)，每个分类又有不同的业务类别数据，例如，股票有基础行情数据，还有各个上市公司的基础信息数据，财务数据.....，由于业务场景，我们做缓存用了主动后台程序缓存，可以缓解数据库压力。但是这样造成了部分及时的行情数据无法及时更新到缓存中，后面我们部分业务改成了近期查库，及时查缓存，从一定程度解决问题。 但是感觉现在开发业务复杂度变大了，需要主动缓存，业务程序查历史缓存，最新数据，进行合并。 我们深知还需要继续学习，寻找更好的解决方案。感谢老师的分享。 作者回复	2018-06-23
希望对你有帮助💎💎	2018-06-23
孙晓明	2018-06-22
我们的一个系统缓存是这样的：在数据库中定义了视图，将视图的结果缓存，永不过期，如果数据发生变动，先更新视图，然后再将视图结果更新到缓存中。后来也有用存储过程定时统计复杂的数据，并将统计结果存储到临时表中，程序访问时读取临时表的数据。	2018-06-22
zwfec	2018-06-11
使用 update +1 的系统挂过，其实就是一个信息页统计访问量，修改了解决方法，使用s写入内存，再定时写入库	2018-06-11
大雁小鱼	2018-06-10
分布式缓存的设计关键点是什么？	2018-06-08
echo _ 陈	2018-06-09
关于缓存穿透，比如，对用户缓存，因为每个用户的访问时间是均匀分布的.....也不会同时缓存失效.....这时候缓存穿透问题不大吧 作者回复	2018-06-09
这种是正常的，只有短时间内大量数据都没命中缓存才算缓存穿透	2018-06-09
Mrsong	2018-06-07
在以往的开发中遇到过这样的问题，某个数据做了缓存，key为k1，在后面的业务中也发现也需要这个缓存数据，但是查询的key不是key1，而是key2，这种情况有什么好的解决方案么，曾经用过的方案是把key1与key2的对应关系也做了缓存，根据对应关系去缓存中查询想要的的数据。类似的业务需求，有好的解决方案么？ 作者回复	2018-06-07
没明白，直接查key1为何不行？	2018-06-07
正直D令狐勇	2018-06-06
老师这里把缓存击穿和缓存雪崩合并在一起讲了。 作者回复	2018-06-06
你是说爬虫的案例是缓存雪崩？我理解不是的，雪崩是由于缓存不存在且高并发访问，穿透是由于缓存不存在或生成缓存需要耗费大量资源	2018-06-06
顾海	2018-06-06
恶意攻击导致的缓存穿透，如果你的key不是非常多，可以使用布隆过滤器 作者回复	2018-06-06
实际应用比较难把所有key列出来	2018-06-06
june peng	2018-06-06
你好，在缓存雪崩后台更新策略里，比如1000个同时访问一个失效的缓存key，如果给这个key加读写锁，这样保证只有一个访问存储系统，其它999个人虽然慢点，但是至少能保证业务不会挂。如果用消息队列，就是前台只拿缓存key里的数据（不能访问存储系统），如果key不存在就发给消息队列更新，如果启多个进程去接受这些消息，依然不能避免后台击穿存储系统，难道只有启用一个进程？这样又太慢，可否进一步说明这种方案的复杂点在哪？谢谢 作者回复	2018-06-06
你的分析很对💎💎 具体在实现的时候，后台更新线程既不能只有一个，也不能和业务线程一样多，一般8~32个就差不多了，因为缓存更新并不会非常频繁。	2018-06-06

假如8个线程后台更新也可能导致缓存雪崩，那就要做更多事情了，例如：后台线程更新前先读取一下缓存，存在就不更新。	MarksG	2018-06-05
老师请教一下，比如数据库连表查询的分页结果，这样的数据怎么设计存入缓存了？比如就用一个key，然后存储数据库查询结果的json吗？这个一直比较困惑。不知道key的设计怎么比较好，感谢	作者回复	2018-06-06
一般分页缓存，缓存id列表，不要缓存所有数据		2018-06-05
天天平安		2018-06-05
请问 阿里hybidbb for mysql,hybidbb for postgresql和maxcompute这三个产品有什么区别，分别在哪些情况选择对应的数据库合适？	qpm	2018-06-05
公司业务部署在某公司云上，有一次memcache故障，导致返回的数据总是为null，刚好系统设计是数据都先加载到缓存的，于是处理逻辑就认为系统里都没有数据，导致出现很大的数据问题。后来我们重新编写逻辑，不完全信任缓存进程，把一部分使用量较高的，数据不大的数据用JVM本地进行缓存。	作者回复	2018-06-05
这里主要的问题是缓存使用不当，正常缓存返回null后应该去存储查		2018-06-05
十七		2018-06-05
现在项目重度依赖redis做统计计算，在请求量非常大的情况下，达不到低延时要求，不知道行业内有没有性能更好的实现方案	作者回复	2018-06-05
之前看到有证券公司将依赖redis+lua的计算改为go去计算，redis只做存储	李唐	2018-06-05
曾经设计过某安卓游戏平台，前台采用redis+solr+mysql多级缓存方案，考虑过缓存穿透，后来被新来的架构师否决了。个人感觉，业务架构重于技术架构	作者回复	2018-06-05
否决的理由是什么？如果评估缓存穿透的可能性很低，不做确实也可以	yushing	2018-06-05
1、文本提到可以监控爬虫，发现问题后及时处理，请问具体是怎么处理呢？ 2、分页数据有很多排序规则，而且可能在某个时间点要上架新商品，就会有实时性的要求，请问这样使用分页缓存真的合适么？	作者回复	2018-06-05
1、监控数据库的各项指标，发现逐步变慢后看看是不是爬虫，只要系统还撑得住就让它爬，撑不住就不让它爬 2、缓存是为了解决性能问题，实时性要求很高就不能用缓存了，或者要做缓存及时更新机制	yushing	2018-06-05
业务系统中有使用reids存储商品list列表，每次添加、更新商品时，跟该商品有关的列表缓存都要跟着刷新；后来改成列表缓存中只保存商品id，查询时再关联商品信息，但这样只是在更新商品时不用更新商品列表缓存，在商品有添加、删除操作时，列表缓存更新的问题依然存在，请问对列表缓存的更新有什么好的方式呢？	作者回复	2018-06-05
我觉得这样设计没什么问题呀，除非你们的list很大，如果list很大，分段缓存就可以	sunny	2018-06-05
之前系统用阿里云memcached,有一段时候缓存总是失效。后来发现随着业务的增长，缓存的值越来越多。同一个value下的值过多会导致缓存系统存储失效。进而导致了文中提到的缓存血崩的情况。后来限制了缓存内容大小。解决了问题。	聊哈	2018-06-05
如果采用商品分页缓存，怎么实现更新呢，商品对实时性还是有要求的吧？比如更改价格，或者新上架的商品，怎么能比较实时的展现？还有商品页还是很复杂的，除了筛选还有很多的排序等等，这些场景怎么应用缓存吗？	作者回复	2018-06-05
缓存常见的列表操作，没法缓存所有的列表，因为搜索的条件太复杂了	byte	2018-06-05
想请教下对于流式音视频数据，类似优酷和爱奇艺这种网站后台的缓存系统如何设计比较合理？有没有推荐的方案或者书籍？谢谢！	作者回复	2018-06-05
如果是媒体内容的缓存，那是CDN流式缓存的范畴，普通的后台缓存没什么特别的	李志博	2018-06-05
最近有一个redis存储数据过大的问题，字符串key，大小2.5mb导致压测的时候发现带宽不够了，我优化的方式类似于后台更新的方式，启动的时候查一次，放到一个concurrent hash map上，然后写个定时器每隔半小时更新一次，业务线程直接读map里的数据，如果读不到，说明是新增的数据，我搞了个jdk自带sync开头的queue，先通过offer往queue里添加一个new object 触发一个异步线程更新缓存，同时业务线程在根据id查单条，我还利用开关的观察者机制，留了个后门，可以通过人工方式触发缓存更新	作者回复	2018-06-05
最好优化缓存设计，这么大的缓存，带宽都很浪费	haydenilu	2018-06-05
老师，关于缓存过期，我想问下，哪些场景需要加过期时间，那些不需要。另外，关于分布式锁，我想问下，基于zk的实现，和基于redis的实现，在阿里内部哪个用的比较多？为什么？	作者回复	2018-06-05
1、缓存的有效期和业务强相关 2、都有用，但哪个更多我没法统计，redis的简单，zk的功能强大，可靠性高	钱浩亮	2018-06-05

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

老师我想问一下，cndle有更细的ms，能否针对的讲一下，可以清晰作者回复	2018-06-05
redis的SETNX可以当锁来用，至于高并发是否可用，要看你期望达到多高的并发了，毕竟这是跨网络的访问，而且是单机支持，具体性能你要实测浪子恒心	2018-06-05
朗读者声音不错，很清晰！	2018-06-05
探索无止境	2018-06-05
文中提到了缓存穿透存储数据不存在的情况，存储空对象，但是如果黑客恶意发起连续的攻击，而且key不相同，那么也会让缓存服务器的内存被迅速占满，这个方案感觉还是有些问题的作者回复	2018-06-05
确实没办法💎💎💎💎	2018-06-05