

## 03 | 二八原则：有针对性地处理好系统的“热点数据”

2018-10-03 许令波

如何设计一个秒杀系统

[进入课程 >](#)



讲述：秭明

时长 11:05 大小 5.08M



假设你的系统中存储有几十亿上百亿的商品，而每天有千万级的商品被上亿的用户访问，那么肯定有一部分被大量用户访问的热卖商品，这就是我们常说的“热点商品”。

这些热点商品中最极端的例子就是秒杀商品，它们在很短时间内被大量用户执行访问、添加购物车、下单等操作，这些操作我们就称为“热点操作”。那么问题来了：这些热点对系统有啥影响，我们非要关注这些热点吗？

### 为什么要关注热点

我们一定要关注热点，因为热点会对系统产生一系列的影响。

首先，热点请求会大量占用服务器处理资源，虽然这个热点可能只占请求总量的亿分之一，然而却可能抢占 90% 的服务器资源，如果这个热点请求还是没有价值的无效请求，那么对系统资源来说完全是浪费。

其次，即使这些热点是有效的请求，我们也要识别出来做针对性的优化，从而用更低的代价来支撑这些热点请求。

既然热点对系统来说这么重要，那么热点到底包含哪些内容呢？

## 什么是“热点”

热点分为**热点操作**和**热点数据**。所谓“热点操作”，例如大量的刷新页面、大量的添加购物车、双十一零点大量的下单等都属于此类操作。对系统来说，这些操作可以抽象为“读请求”和“写请求”，这两种热点请求的处理方式大相径庭，读请求的优化空间要大一些，而写请求的瓶颈一般都在存储层，优化的思路就是根据 CAP 理论做平衡，这个内容我在“减库存”一文再详细介绍。

而“热点数据”比较好理解，那就是用户的热点请求对应的数据。而热点数据又分为“静态热点数据”和“动态热点数据”。

所谓“静态热点数据”，就是能够提前预测的热点数据。例如，我们可以通过卖家报名的方式提前筛选出来，通过报名系统对这些热点商品进行打标。另外，我们还可以通过大数据分析来提前发现热点商品，比如我们分析历史成交记录、用户的购物车记录，来发现哪些商品可能更热门、更好卖，这些都是可以提前分析出来的热点。

所谓“动态热点数据”，就是不能被提前预测到的，系统在运行过程中临时产生的热点。例如，卖家在抖音上做了广告，然后商品一下就火了，导致它在短时间内被大量购买。

由于热点操作是用户的行为，我们不好改变，但能做一些限制和保护，所以本文我主要针对热点数据来介绍如何进行优化。

## 发现热点数据

前面，我介绍了如何对单个秒杀商品的页面数据进行动静分离，以便针对性地对静态数据做优化处理，那么另外一个关键的问题来了：如何发现这些秒杀商品，或者更准确地说，如何发现热点商品呢？

你可能会说“参加秒杀的商品就是秒杀商品啊”，没错，关键是系统怎么知道哪些商品参加了秒杀活动呢？所以，你要有一个机制提前来区分普通商品和秒杀商品。

我们从发现静态热点和发现动态热点两个方面来看一下。

### 发现静态热点数据

如前面讲的，静态热点数据可以通过商业手段，例如强制让卖家通过报名参加的方式提前把热点商品筛选出来，实现方式是通过一个运营系统，把参加活动的商品数据进行打标，然后通过一个后台系统对这些热点商品进行预处理，如提前进行缓存。但是这种通过报名提前筛选的方式也会带来新的问题，即增加卖家的使用成本，而且实时性较差，也不太灵活。

不过，除了提前报名筛选这种方式，你还可以通过技术手段提前预测，例如对买家每天访问的商品进行大数据计算，然后统计出 TOP N 的商品，我们可以认为这些 TOP N 的商品就是热点商品。

### 发现动态热点数据

我们可以通过卖家报名或者大数据预测这些手段来提前预测静态热点数据，但这其中有一个痛点，就是实时性较差，如果我们的系统能在秒级内自动发现热点商品那就完美了。

能够动态地实时发现热点不仅对秒杀商品，对其他热卖商品也同样有价值，所以我们需要想办法实现热点的动态发现功能。

这里我给出一个动态热点发现系统的具体实现。

1. 构建一个异步的系统，它可以收集交易链路上各个环节中的中间件产品的热点 Key，如 Nginx、缓存、RPC 服务框架等这些中间件（一些中间件产品本身已经有热点统计模块）。
2. 建立一个热点上报和可以按照需求订阅的热点服务的下发规范，主要目的是通过交易链路上各个系统（包括详情、购物车、交易、优惠、库存、物流等）访问的时间差，把上游已经发现的热点透传给下游系统，提前做好保护。比如，对于大促高峰期，详情系统是最早知道的，在统一接入层上 Nginx 模块统计的热点 URL。
3. 将上游系统收集的热点数据发送到热点服务台，然后下游系统（如交易系统）就会知道哪些商品会被频繁调用，然后做热点保护。

这里我给出了一个图，其中用户访问商品时经过的路径有很多，我们主要是依赖前面的导购页面（包括首页、搜索页面、商品详情、购物车等）提前识别哪些商品的访问量高，通过这些系统中的中间件来收集热点数据，并记录到日志中。

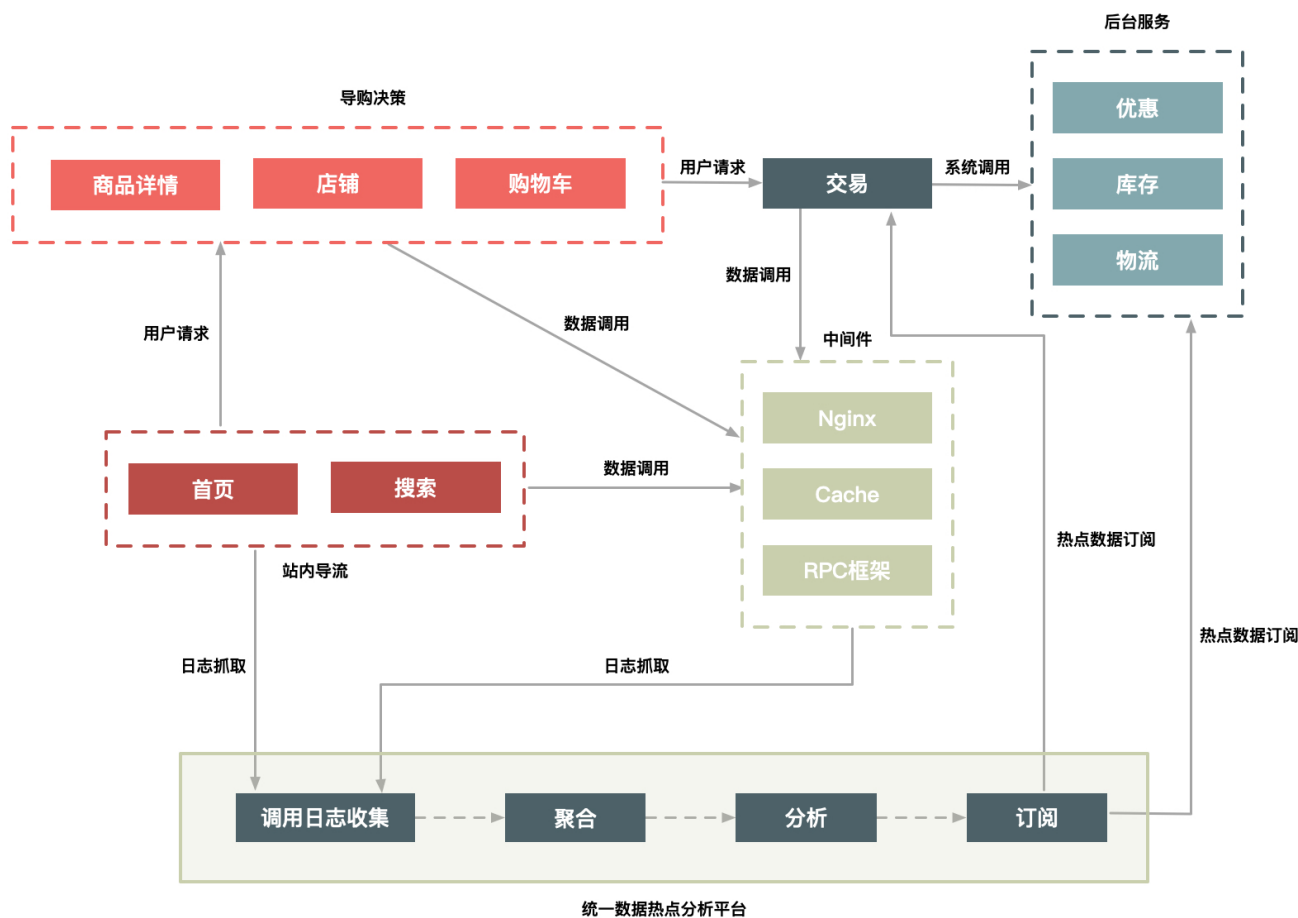


图 1 一个动态热点发现系统

我们通过部署在每台机器上的 Agent 把日志汇总到聚合和分析集群中，然后把符合一定规则的热点数据，通过订阅分发系统再推送到相应的系统中。你可以是把热点数据填充到 Cache 中，或者直接推送到应用服务器的内存中，还可以对这些数据进行拦截，总之下游系统可以订阅这些数据，然后根据自己的需求决定如何处理这些数据。

打造热点发现系统时，我根据以往经验总结了几点注意事项。

1. 这个热点服务后台抓取热点数据日志最好采用异步方式，因为“异步”一方面便于保证通用性，另一方面又不影响业务系统和中间件产品的主流程。
2. 热点服务发现和中间件自身的热点保护模块并存，每个中间件和应用还需要保护自己。热点服务台提供热点数据的收集和订阅服务，便于把各个系统的热点数据透明出来。

3. 热点发现要做到接近实时（3s 内完成热点数据的发现），因为只有做到接近实时，动态发现才有意义，才能实时地对下游系统提供保护。

## 处理热点数据

处理热点数据通常有几种思路：一是优化，二是限制，三是隔离。

先来说说优化。优化热点数据最有效的办法就是缓存热点数据，如果热点数据做了动静分离，那么可以长期缓存静态数据。但是，缓存热点数据更多的是“临时”缓存，即不管是静态数据还是动态数据，都用一个队列短暂地缓存数秒钟，由于队列长度有限，可以采用 LRU 淘汰算法替换。

再来说说限制。限制更多的是一种保护机制，限制的办法也有很多，例如对被访问商品的 ID 做一致性 Hash，然后根据 Hash 做分桶，每个分桶设置一个处理队列，这样可以把热点商品限制在一个请求队列里，防止因某些热点商品占用太多的服务器资源，而使其他请求始终得不到服务器的处理资源。

最后介绍一下隔离。秒杀系统设计的第一个原则就是将这种热点数据隔离出来，不要让 1% 的请求影响到另外的 99%，隔离出来后也更方便对这 1% 的请求做针对性的优化。

具体到“秒杀”业务，我们可以在以下几个层次实现隔离。

1. **业务隔离**。把秒杀做成一种营销活动，卖家要参加秒杀这种营销活动需要单独报名，从技术上来说，卖家报名后对我们来说就有了已知热点，因此可以提前做好预热。
2. **系统隔离**。系统隔离更多的是运行时的隔离，可以通过分组部署的方式和另外 99% 分开。秒杀可以申请单独的域名，目的也是让请求落到不同的集群中。
3. **数据隔离**。秒杀所调用的数据大部分都是热点数据，比如会启用单独的 Cache 集群或者 MySQL 数据库来放热点数据，目的也是不想 0.01% 的数据有机会影响 99.99% 数据。

当然了，实现隔离有很多种办法。比如，你可以按照用户来区分，给不同的用户分配不同的 Cookie，在接入层，路由到不同的服务接口中；再比如，你还可以在接入层针对 URL 中的不同 Path 来设置限流策略。服务层调用不同的服务接口，以及数据层通过给数据打标来区分等等这些措施，其目的都是把已经识别出来的热点请求和普通的请求区分开。

## 总结一下



本文与数据的动静分离不一样，它从另外一个维度对数据进行了区分处理。你要明白，区分的目的主要还是对读热点数据加以优化，对照“4 要 1 不要”原则，它可以减少请求量，也可以减少请求的路径。因为缓存的数据都是经过多个请求，或者从多个系统中获取的数据经过计算后的结果。

热点的发现和隔离不仅对“秒杀”这个场景有意义，对其他的高性能分布式系统也非常有价值，尤其是热点的隔离非常重要。我介绍了业务层面的隔离和数据层面的隔离方式，最重要最简单的方式就是独立出来一个集群，单独处理热点数据。

但是能够独立出来一个集群的前提还是首先能够发现热点，为此我介绍了发现热点的几种方式，比如人工标识、大数据统计计算，以及实时热点发现方案，希望能够给你启发。

最后，欢迎你在留言区和我交流，你也可以说说在实际工作中，还有哪些发现和解决热点问题的不同思路或方案，非常期待。



# 如何设计一个秒杀系统

大并发高可用秒杀系统的设计之道

许令波 前阿里巴巴 高级技术专家



新版升级：点击「🔗 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 02 | 如何才能做好动静分离？有哪些方案可选？

下一篇 04 | 流量削峰这事应该怎么做？

## 精选留言 (26)

写留言



A星空之夜

2018-10-03

6

如果是微博的热点数据，应该怎么优化读操作呢

展开

作者回复: 微博我没做过，但是我觉得大体思路应该是通用的

微博的内容基本是静态数据，发了就不能修改，所以cache应该没问题，而微博的热点比较明显，所以发现热点和对热点数据进行针对性的优化比较重要

至于要怎么进行优化还要看看微博的数据结构是怎么设计的，微博内容和评论的关系已经发微博的人和关注的人关系等



爱我曲

2018-10-04

4

例如对被访问商品的 ID 做一致性 Hash，然后根据 Hash 做分桶，每个分桶设置一个处理队列，这样可以把热点商品限制在一个请求队列里

作者，您好，这个分桶设置处理队列是怎么实现的？可以讲下实现过程吗？

展开

作者回复: 例如对每个请求的商品id取模，让后根据取模的结果分别设置多个linkedhashmap，每个map当做一个队列



放羊大王

2018-10-04

4

思路大概是通过日志和数据分析判断，某商品是热点，即将有大量请求到来。把商品请求移交给另外的秒杀系统，秒杀系统动静分离出html页面，和ajax请求。

展开

作者回复: 😊



慕-小坏

2018-10-08

👍 3

但是，缓存热点数据更多的是“临时”缓存，即不管是静态数据还是动态数据，都用一个队列短暂地缓存数秒钟，由于队列长度有限，可以采用 LRU 淘汰算法替换。

为什么选择用队列来缓存数据呢，队列如何做到快速读取缓存数据的呢？

展开 ▾

作者回复: 不一定非要用队列，也可以用其他内存集合如hashmap都行  
缓存数据，一般静态数据可以提前加载到内存，如通过配置系统推送的方式。动态数据可以在远程调用时零时缓存在本机中。



黑瞎子

2018-10-28

👍 1

关于隔离层次的一个小问题：想了下数据隔离好像不太好实现吧？如果都是只读的，那么还好办一些，开一些只读实例专用就好，但是这样的场景肯定会有写请求，但是写的记录要和其他数据互通，最后不还得打到数据库集群里么？这好像和“启用单独的 MySQL”相悖了.....请老师具体指教下，多谢！

展开 ▾

作者回复: 是不好做，隔离数据是按照一个纬度隔离的，例如隔离秒杀商品，是按照商品纬度，就是把一部分商品数据单独放到另外一个数据库，这样只要把一部分商品路由到新的隔离数据库就行了。



halo

2018-10-06

👍 1

图1 热点发现系统: 中间件nginx cache rpc 为什么不是后台服务的一部分呢？还是说中间件构成的异步监控系统用来日志记录？

展开 ▾

作者回复: 也可以把他当成后台系统的一部分，但是中间件本身就一直存在，并不是因为热点发现系统才加入的。



中间件记录热点大部分是记录到日志里面的，但也不绝对



小喵喵

2018-10-03

1

发现动态热点中的第二点，建立一个热点上报。。

怎么判断数据是不是热点呢？是一个具体的算法没有写出来吗？这个地方看了好几遍都看不明白，请指点。

展开

作者回复: 热点的统计可以很简单的对访问的商品进行访问计数，然后排序  
还有就是用通常的队列的淘汰算法如lru等都可以实现



Mr.钩

2018-10-03

1

虽然这个热点可能只占请求总量的亿分之一，但是可能占用服务器90%的资源

这句话是什么意思呢？可以详细解释一下吗？

作者回复: 亿分之一是指热点商品指占商品用量的比例很小

占用服务器资源是指一台服务器秒杀商品的qps占了大部分请求比例



iWill

2018-10-03

1

缓存数据，LRU淘汰算法，那么缓存的这些数据会不会频繁的移动、删除，并发比较大的时候会不会成为性能的瓶颈，想了解下。

展开

作者回复: 对秒杀这个场景，商品基本是固定的，基本都会一直在内存中



**Jefung**

2019-04-21



分桶限制还是无法理解, 这么处理才能把热点请求和正常请求区分开

---



**WL**

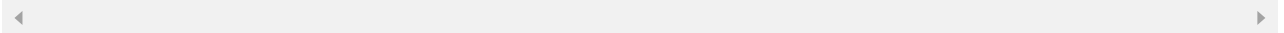
2019-04-20



请问下老师缓存数据，LRU淘汰算法的实现可不可以用redis加一个过期时间，如果可以的话是否需要动态的判断过期时间，如果不行的话是什么原因不可以？

展开 ∨

作者回复: 加过期时间也是一个好思路



**威猛小茂然**

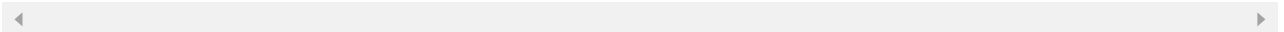
2019-03-18



写的挺好的

展开 ∨

作者回复: : )



**李白**

2019-02-28



再来说说限制。限制更多的是一种保护机制，限制的办法也有很多，例如对被访问商品的ID 做一致性 Hash，然后根据 Hash 做分桶，每个分桶设置一个处理队列，这样可以把热点商品限制在一个请求队列里，防止因某些热点商品占用太多的服务器资源，而使其他请求始终得不到服务器的处理资源。

这个具体怎么操作呢?nginx可以处理吗？

展开 ∨



**冉博哲 (Fr...)**

2019-02-27



DB 层，如何对热点数据进行优化呢？比如：a 表的某一行的库存，是热点更新？除了把数据拦在 DB 层之外的，进入 DB 层的热点更新请求，要如何给 DB 减压呢？想到给热点行的库存分成多个减压，当缺点太多，而且对于动态热点没用。其它方案都还有那几种？

展开 ▾

作者回复: 数据库层深层优化我不是专家, 建议可以参考一下MySQL的优化书籍 📖



**williamcai**

2019-01-27



热点发现系统, 通过日志分析出热点数据, 有没有判断标准说明该数据是热点, 比如搜索次数达到多少, 能不能给出具体的指标

展开 ▾

作者回复: 这个具体的指标要看情况, 不一定就是多少次就判定是热点, 也可能根据top多少来设置, 到底设置多少合理, 要根据其他一些因素来综合考虑, 例如缓存这些热点的cache空间大小以及你对这些热点的处理方式(例如可能会限流)都会影响。



**Andy\_Clai...**

2019-01-21



你好, 看到你处理热点数据的思路, 其中限制和隔离让我有点混淆, 感觉区别不大. 你给出限制的做法是根据id做hash, 然后用单独的队列处理. 我理解这也是一种维度(请求)的隔离.

作者回复: 从效果上来说, 都是为了达到保护的作用。限制是防止你越界, 而隔离是不让你越界 : )



**godtrue**

2018-11-13



这节的重点感觉是加深对二八原理的认识, 这个原理不知为何好像具有普适的感觉, 各个领域都有它的用武之地。

展开 ▾

作者回复: 😊



godtrue

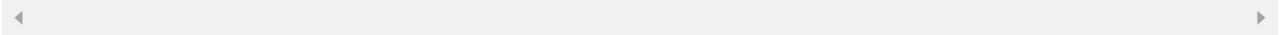
2018-11-13



热点数据-我们之前做过一个系统也用到热点数据，我们的思路比较简单，对于仓库生产，仓库的范围是有限的，具体到仓，从业务的角度出发北上广的仓库基本属于比较热的数据，于是就有了定时和手动往缓存中刷新热点数据的处理方式了。

展开 ▾

作者回复: 虽然土点，但是有效 😊



qingbushen...

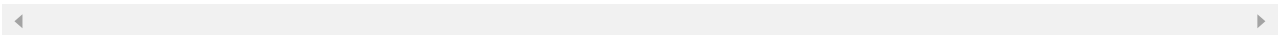
2018-11-10



解决快速发现热点数据的另一个思路：1.在前端nginx 通过商品、下单、购物车URI 做数据镜像 2 通过nginx+lua+redis 根据不同的业务key 做计数，如果超过设定阈值则采用redis 发布订阅机制或消息中间件 通知下游Cache 模块缓存数据。感谢老师点评

展开 ▾

作者回复: 😊



rango

2018-11-05



你好，想问一下通过异步采集日志的形式获取热点数据这方面的话，我想知道日志如何存储，是不是加一些特定字符串以便收集，采集的方法等技术，有没有思路或者推荐的文章等。

作者回复: 日志收集和存储的技术现在都比较成熟了，有很多开源的组件，例如zabbix可以学习一下他是如何实现的

