

海尔电商峰值系统架构设计

刘志斌 架构师 2015-12-21

架构师 (JiaGouX)

我们都是架构师！

导语

本文重点介绍了海尔电商平台的架构方案，也用不少篇幅阐述面临的场景和挑战，以及在架构方案决策过程中的关注点。其实作为一个优秀的电商平台，提供极致的用户体验、让技术最大化地创造价值，才是架构的终极目标。

多数电商平台都会经历相似的过程，流量和业绩每年以几倍至十几倍的速度增长，每年都要接受几次大规模、全方位的系统检阅，例如双11、周年庆等购物狂欢节，期间流量和订单可能是日常的十几倍甚至几十倍，产生的峰值对平台形成极其强烈的冲击，对电商平台的架构带来巨大的考验。因此，对电商平台的规划和架构工作不仅要高瞻远瞩，而且要细致入微，否则将导致平台无法满足高速增长的业务发展，细微处的失误也可能造成严重后果，不仅影响业务指标的实现，还可能导致对系统进行重新架构，劳时费力又伤钱。

从2012年开始，海尔进入了网络化发展阶段，企业平台化、用户个性化和员工创客化的“三化”做法为电商的蓬勃发展提供了很好的土壤，也是海尔在面对互联网转型时的一个重点。海尔电商平台在发展过程中也同样经历了上述的问题。下面就抛砖引玉，为大家分享海尔电商平台应对电商峰值的架构设计经验。

站在巨人肩膀上的SOA架构

随着电商业务开展和业绩增长，系统结构和逻辑变得越来越复杂。为应对业务规模和复杂性的增长，需要将系统按照细分专业领域拆分；为应对流量和交易的增长，需要将网站进行大量子站拆分。这种状况下，SOA在保持清晰的系统结构和良好的逻辑组织方面提供了有力保障，为业务优化调整及新业务的开展带来巨大收益。

通过服务封装和严格分离，为电商平台实现高伸缩性打下坚实基础。实现高伸缩性的主要工作集中在服务内部，对客户端影响的评估和改造工作也变得非常清晰。这将大大降低了实现高伸缩性的难度、工作量和实施周期。

Dubbo是阿里提供的一个优秀的开源服务框架，在高并发情况下具有优秀的性能表现，海尔电商的SOA架构全面基于Dubbo服务框架。关于Dubbo框架的详细介绍可以参考GitHub上的Dubbo项目文档。下面对Dubbo框架工作机制进行简单介绍。

如图1所示，每个服务提供者启动时都会注册到注册中心，并且通过长连接与注册中心保持心跳检测。这样注册中心就拥有一份完整、可用的服务提供者清单，某个服务提供者下线或由于故障中断，注册中心都能感知到并从清单中删除这个提供者。消费者启动时从注册中心获得服务提供者清单，并与提供者建立长连接，后续直接调用服务提供者，不再经过注册中心，避免注册中心成为瓶颈。每个消费者同样与注册中心保持长连接，这样有新的提供者注册或者某个提供者下线，都由注册中心通知到每个消费者。消费者在调用服务提供者时支持各种负载均衡和故障容错策略。监控中心则负责运行状态统计，例如每分钟的调用次数和平均耗时等。

图1 Dubbo服务部署架构示意图

Dubbo框架不仅实现了高性能、高可用性，而且使用方便，扩展性非常好。海尔电商所有服务都基于Dubbo框架开发，图2是系统整体SOA架构情况。

图2 海尔电商SOA架构示意图

鱼与熊掌兼得的产品服务架构

面临的挑战

产品的检索和展示在电商平台中具有举足轻重的地位，贯穿用户浏览、购物整个过程，以及订单交付全流程。产品服务需要为整个平台提供数据请求和检索服务，而各品类的产品差异性非常大，这给产品服务设计带来了巨大的挑战。

负载权重高。电商平台中几乎每一个前台页面都与产品展示和检索相关，产品服务的负载在整个平台中占比非常高，对产品服务的请求量可能达到整站流量的几倍、几十倍。在电商活动高峰期间，核心系统中首当其冲的便是产品服务。因此，产品服务的设计必须满足高可用性，并且实现良好的性能和高伸缩性。

产品差异性大。不同品类的产品具有不同维度的属性和规格参数，产品结构的设计必须具备足够的通用性和灵活性，才能良好地满足电商平台多品类运营的要求，以及在平台、品类扩展时可以提供快速的响应支持。

全方位检索、排序。让用户方便快捷地在大量产品中找到自己满意的产品，是电商平台用户体验和信息架构中非常关键的一点。除了关键词搜索、按类目检索浏览之外，还需要提供按常用属性进行检索。在深入优化用户体验时，可能会提出更复杂的检索处理逻辑，例如组合属性检索，自动根据检索结果反过滤掉无结果的类目和属性，展示符合各个属性条件的商品个数，以及实时地结合大数据分析结果添加更多自动化、智能化的策略等。

将页面或者部分页面的静态化是一种非常有效的优化方式，可以极大地降低对后台服务和数据的请求。但静态化带来的最大弊端就是服务端丧失了控制力，使得一些深入的自动化、智能化策略难以应用。因此，我们希望通过提升服务端的性能和伸缩性，来避免静态化的方案。

性能和伸缩性是电商平台的关键指标。为了保障系统性能和伸缩性，不少时候我们需要牺牲或者完全拒绝某些功能，或者降低系统的灵活性和扩展性等。在产品服务架构设计阶段，我们努力思考和研究着一种可以鱼和熊掌兼得的解决方案。

解决方案

如图3所示，在数据库层允许复杂的产品存储结构设计，以细粒度、深度优化的关系模型充分实现产品数据模型的通用性、可扩展性。在数据模型设计时完全不用关心客户端检索查找的复杂性和性能问题。

图3 产品服务逻辑架构示意图

产品查询引擎将复杂的数据存储模型封装成一个简单的逻辑模型。这个逻辑模型实现的效果，完全等同于产品的所有属性都存储在同一张数据库表中，逻辑模型的每个属性对应数据库表中的一个字段。在这个逻辑模型的基础上实现了一个简洁的DSL，供客户端进行检索查询。客户端工作在逻辑模型和DSL之上，检索查询简单、灵活，同样完全不用关心产品数据存储模型的复杂性和性能问题。

产品查询语言DSL

产品查询语言DSL的语法类似SQL中的where条件语法，任何一个开发人员都很容易掌握。客户端将DSL表达式传给服务端，即可得到满足条件的产品列表及相关属性数据（图4）。

图4 查询语言DSL工作原理

DSL还支持中文语法，更方便使用，尤其对于业务人员进行复杂的后台检索查询，或者为前台页面及栏位设置产品展示的过滤条件等情况。

产品查询引擎

图5描述了查询引擎的核心组件及关键的执行流、数据流。编译器基于Antlr开发，职责是将DSL表达式编译为语法树，并完成一系列编译优化操作。执行引擎使用语法树逐个对产品进行匹配，得到符合条件的产品列表。智能排序引擎基于产品综合竞争力评估模型，为结果集进行排序，实现最大化提升转换率的目的。结果构造器则根据客户端在调用服务时指定的要求，将客户端所需属性加载到结果集中。

图5 查询引擎工作机制

在服务启动时将产品数据缓存到内存中，通过订阅MQ消息队列，在数据发生变化时刷新有变化的数据。

产品服务架构

产品服务分不同集群进行部署，面向Web应用和其他服务的集群在运行期间几乎不会产生数据库请求，因此不管网站访问量和交易量多高，数据库都不会产生压力瓶颈。在系统峰值期间，只需为Web和服务添加服务器即可，实现了高伸缩目标。

效果

性能：最高峰值2.6亿次/天，平均耗时60毫秒/次，后续对编译器和执行引擎进行优化，性能还有更大的提升空间。

伸缩性：在一定条件下接近线性伸缩，所有使用产品服务的地方无须出于性能和系统压力原因额外设计其他方案，直接调用产品服务即可。

通用性：不会因为电商平台性能和伸缩性要求而受到任何限制，可以像开发内部管理系统PDM一样设计产品数据模型，并且直接用于其他在线服务和前台Web应用，尽

可能达到通用灵活的目的。

扩展性：通过逻辑模型屏蔽了底层的数据模型，将数据模型的优化、扩展工作量以及影响范围降低到最小限度，提升了电商平台中产品服务的可维护性和扩展性。

以查询引擎为核心的产品服务是一个鱼与熊掌兼得的架构设计案例，通用性、扩展性、伸缩性等电商平台中相互制约、矛盾的一组核心架构目标全部得到满足。

.....

作者刘志斌，海尔电商首席架构师

来源：CSDN

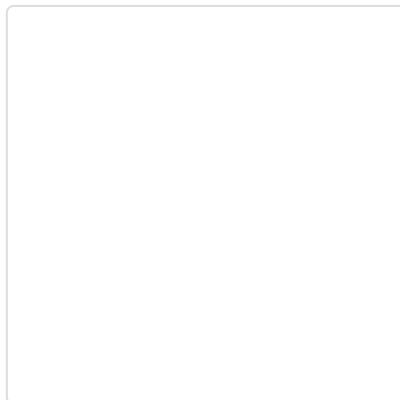
原文：<http://kb.cnblogs.com/page/511435/>

转载文章，向原作者致敬！如有侵权或不周之处，敬请劳烦联系若飞（微信：13511421494）马上删除，谢谢！

·END·

架构师

我们都是架构师！



架构师订阅号，关注获取更多技术分享

现已开通多个微信群，有兴趣交流学习的同学

可加若飞微信：13511421494进群

合作邮箱：admin@137x.com

