

04 | 标准化体系建设（下）：如何建立基础架构标准化及服务化体系？

2017-12-27 赵成

赵成的运维体系管理课

[进入课程 >](#)



讲述：黄洲君

时长 09:52 大小 4.52M



前面我们一起讨论了为什么要做标准化，标准化的套路是什么，并按照套路进行了基础设施和应用的标准化示例。我想这些内容可以帮助我们举一反三，尝试着应用到实际工作中了。

今天，我继续跟你聊基础架构标准化的问题，但是今天我计划不谈如何进行架构标准化的细节，而是想强调一下基础架构标准化的重要性，因为从我个人的经历和我实际观察到的情况来看，这块的问题会更普遍一些，而这一部分又影响着后续一系列效率和稳定性平台的建设方案。

同时，如果说上次我们讲的基础设施和应用标准化是运维团队职责的话，那今天的内容就是架构、开发和运维共同的职责。

常见的分布式基础架构组件

让我们先一起列一下，微服务的分布式架构下，涉及到的主要基础架构组件有哪些。

分布式服务化框架，业界开源产品比如 Dubbo、Spring Cloud 这样的框架；

分布式缓存及框架，业界如 Redis、Memcached，框架如 Codis 和 Redis Cluster；

数据库及分布式数据库框架，这两者是密不可分的，数据库如 MySQL、MariaDB 等，中间件如淘宝 TDDL（现在叫 DRDS）、Sharding-JDBC 等。当前非常火热的 TiDB，就直接实现了分布式数据库的功能，不再额外选择中间件框架；

分布式的消息中间件，业界如 Kafka、RabbitMQ、ActiveMQ 以及 RocketMQ 等；

前端接入层部分，如四层负载 LVS，七层负载 Nginx 或 Apache，再比如硬件负载 F5 等。

上面是几类主要的基础架构组件，为了便于理解我以开源产品举例。但在实际场景中，很多公司为了满足业务上的个性化需求，会自己研发一些基础组件，比如服务化框架、消息中间件等，这个情况在有一定技术实力的公司里比较常见。不过大部分情况下，我们会基于这些开源产品做一些封装或局部的改造，以适应我们的业务。

基础架构组件的选型问题

关于基础架构组件，业界可供我们选择的解决方案和产品非常多，但是选择多了就容易挑花眼，反而不知道从何入手。我们大概都会遇到同样的问题，**是自研还是选择开源产品？有这么多的开源产品到底该选哪一个？**

按正常的思路，一定是先组织选型调研，然后进行方案验证和对比，最后确认统一的解决方案。

但是，由于开源产品的便利性，以及开发同学对技术探索的好奇心，实际情况往往是，整个大的技术团队中，不同的开发团队，甚至不同的开发人员，会根据开发的需要或个人喜好，选择不同的开源产品，在没有严格限制的情况下，甚至会尝试去自研。

按照我的观察，这个问题特别容易出现在微服务架构引入初期。在这个阶段，团队组织架构按照业务领域进行切分，产生一个个与业务架构匹配的小规模技术团队。

每个小团队所负责的业务相对独立，自主权就会变大，如果这个时候整个团队中没有一个强有力的架构师角色去做端到端的约束，就极其容易出现上面的这个问题，并且会一直扩散蔓延下去。

相比之下，成规模的大公司在这一点上做得就相对严格一些，当然也可能是因为之前尝过苦头，所以后来变得越来越规范了。所以这一点也是每个技术团队在引入微服务架构时要提前关注的。

我们以分布式服务化框架为例，我之前遇到的一个实际情况就是，整个大的技术团队选型时以 Java 技术栈为主，毕竟这块有很多的业界经验和产品可以借鉴参考。但是有的团队对 PHP 特别精通熟悉，就想用 PHP 去做微服务，有的团队对 Go 感兴趣，就想尝试 Go 的微服务。

从单纯的技术选型上来看，选择什么语言并没有严格的标准。而且在技术团队中，我们也应该鼓励技术多样性和尝试新技术。不过这里要有个度，我暂时先不细说这个度在哪里，我们先来看看，假设没有统一标准的约束会带来什么问题。

技术的应用，一般都会随着应用场景的逐步深入和业务体量的增长，逐步暴露出各种各样的问题，我们分两个层面来看。

1. 开发层面

业务开发同学将大量的精力投入到基础组件和开源产品的研究、研发以及规模化之后的运维上，再加上产品形态的不统一，导致需要在技术层面的协作上做大量适配工作，而且经验还无法互通。

好不容易在一个产品上摸索了很长时间，踩了很多坑，积累了宝贵的经验，结果发现另外一个产品也要经历同样的一个过程，积累的经验依然不能互通和传递。

2. 运维层面

当我们考虑建设一个统一的效率和稳定体系时，发现基础组件不统一，这个时候就需要做大量的针对不同组件的适配工作。

比如我们要在发布系统中做服务上下线处理，就要针对多个微服务化框架做适配。再举个稳定性上全链路跟踪的例子，为了在分布式复杂调用场景下的链路跟踪和问题定位，我们会在服务化框架中统一做打点功能，这样才不需要侵入业务逻辑。

就这样一个工作，如果服务化框架不统一，就需要到每个框架里都去开发一遍。不过现实中遇到的实际问题是，整个链路就是会有这样那样的情况而串联不起来。

如果你有过类似的经历，一定深有感受。其实各种各样奇葩的问题还远不止这些，继续演化下去，就是我们所说的架构失控了。

当我们把业务开发资源消耗在与业务开发无关的事情上，业务开发就很难聚焦于业务架构，并能够更快、更多、更好地完成业务需求，这就与公司对业务开发的诉求背道而驰了。

同时还会出现维护投入不足，那就必然导致故障频发等一系列问题，团队内部也会因为问题定位不清楚而形成扯皮推诿的不良氛围。

所以，这个时候我们需要做的，就是**对基础架构有统一的规划和建设**。原则上，每种基础组件只允许一种选型，至少就能满足 90% 甚至更多的应用场景。

比如数据库就只允许使用 MySQL，然后版本统一，同时配套的中间件也必须统一，其它的关系型数据库没有特殊情况坚决不允许使用，如果遇到特殊情况具体分析。

这里就举个特殊的小例子。

为了更好地满足业务个性化需求，我们的消息中间件在早期选择了自研，业务上也要求各个应用使用我们统一的服务。但是对于大数据的业务来说，很多开源产品如 Spark，都是原生与 Kafka 配套的，同时很多的新特性也都是基于 Kafka 去做的，在这种情况下就不能很生硬地要求大数据业务必须按照我们的标准来，这里还是得遵守大数据生态本身的标准才可以。

所以选型问题还是要看具体的业务和应用场景，这里只介绍大致的原则，至于具体应该如何标准化，你可以参考我们前面讲到的标准化套路去尝试梳理，先看看你梳理出来的标准化体系是什么样的，后面我也会针对案例进行分享。

基础架构的服务化

我们对基础架构组件做了统一标准之后，下一步要做的就是服务化。因为这些组件都只提供了简单的维护功能，还有很多都是命令行层面的维护，这时我们要做的就是把这些组件提供的维护 API 进行封装，以提供更加便捷的运维能力。

这里以 Redis 缓存为例。

创建和容量申请；

容量的扩容和缩容，新增分片的服务发现及访问路由配置；

运行指标监控，如 QPS、TPS、存储数据数量等等；

主备切换能力等等。

以上这些，假设都依赖 Redis 提供的原生能力来做，基本是不可维护的。所以必须要**基于这些原生能力进行封装，结合运维场景，将能力服务化，这样就大大提升了使用方的便利性。**

同时，我们也可以看到，这个服务化的过程其实就是 PaaS 化的过程。换言之，**如果我们能把基础架构组件服务化完成，我们的 PaaS 平台也就基本成型了。**

运维的职责是什么？

总结上面的过程，我们要做的事情，可以归纳为两步：**第一步是基础架构标准化，第二步是基础架构服务化。**

所以这个时候，运维必须要有意识去做的两件事情。

1. **参与制定基础架构标准，并强势地约束。**在这里运维作为线上稳定的 Owner，发挥约束作用有可能会比业务架构师这样的角色更为有效。另外，由于历史原因或其他种种因素造成的已有架构标准不统一的问题，是需要开发和运维共同合作去改造的。这里面如何保持良好的协作，制定统一的路线图也是非常重要的。所以这里强制约束是一方面，同时也要提供工具化的手段来支持开发的改造，也就是下面这个动作。
2. **基础架构的服务化平台开发，目标是平台自助化，让开发依赖平台的能力自助完成对基础组件的需求，而不是依赖运维的人。**这个事情是驱动运维转型和改进的动力，也是运维能够深入了解架构组件细节的有效途径。同时，要注意到，如果不朝着服务化方向发展，运维将始终被拖累在这些基础组件的运维操作上。

今天我们讨论的这个话题，我也和很多同行、专家交流过，发现大家都有相同的痛点，但是业界的架构资料和图书中很少涉及到这一部分的内容。我觉得根本上还是经验意识上的缺失，所以结合自己的经验专门整理出来，也很期待听到你的经验和想法。

如果今天的内容对你有帮助，也请你分享给身边的朋友。

欢迎你留言与我一起讨论。



赵成的运维体系管理课

带你直击运维的本质

赵成 美丽联合集团技术服务经理



拼课微信：1716143661

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 03 | 标准化体系建设（上）：如何建立应用标准化体系和模型？

下一篇 05 | 如何从生命周期的视角看待应用运维体系建设？

精选留言 (6)

 写留言



赵成 置顶
2017-12-29

 8

本篇的重点，标准化的目的是为了更好提供地提供服务，而服务化又是标准化的技术落地形式。



宵伯特

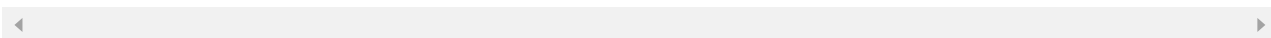
2017-12-28

👍 6

亦如编程的过程就是了解工具掌握工具规范化使用工具的过程，编程中的自由并非为了达到目的就可以胡作非为，而是在一定的约束下，通过有限且有效的步骤来实现业务逻辑，而这些约束或者是编程范式，或是设计模式，或是代码规范，总之想要提升效率就需要制定标准规范，遵守契约，以流程化，标准化的方式进行高效的协作。在架构中规范协议，制定标准，避免过多精力耗费在组织之间的协作沟通上，从而提高研发效率。

展开 ▾

作者回复: 优质评论！赞



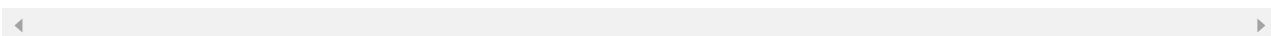
白开水

2018-01-06

👍 3

正在梳理公司的运维标准及规范。公司以前也有，就是比较零散，现在发现缺乏一个方法，把这些零散的文档串起来。就是说，希望能找到一个方法，能整理出我们需要哪些标准规范，具体要求是什么。缺少的，我们补充；不达标的，我们完善。请问有什么建议吗？

作者回复: 从你实际的运维对象出发，应用，服务器等等，这个需要你去识别，同时后面几篇文章也可以看看，有介绍从生命周期的角度入手的套路。



james

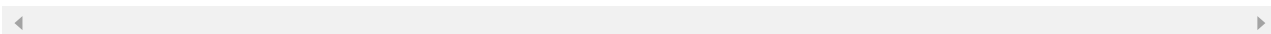
2018-03-09

👍 1

不知道用java技术栈能否开发出您说的这类运维平台，而无需为每个被管理对象安装一个代理，比如：开机关机，当然是针对虚机或docker，还有控制redis的启停啊，rabbitMQ的启停啊，当时的消息的消费情况什么的，等等，相信这个团队要有相当的技术实力，比如阿里云的所有服务都通过网页可以操作

作者回复: 语言不重要，关键是理清需求和架构，各个公有云平台其实提供了很好的借鉴。

技术实力是一方面，确实需要积累，只能一步步来





付盼星

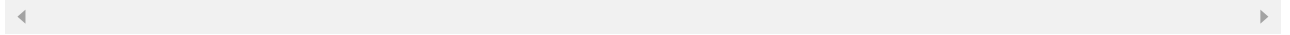
2018-03-04

👍 1

服务化也是运维能力的输出，当然它的核心目的是标准化的落地。

展开 ▾

作者回复: 讲的很好，不过我觉着反过来理解会更好，标准化是手段，运维能力输出是目的。



keeper

2018-09-25

👍

后面提到的基础架构服务化“基础架构的服务化平台开发，目标是平台自助化，让开发依赖平台的能力自助完成对基础组件的需求，而不是依赖运维的人”，不是很理解这里的意思，意思是后面运维工作交给开发？

展开 ▾