

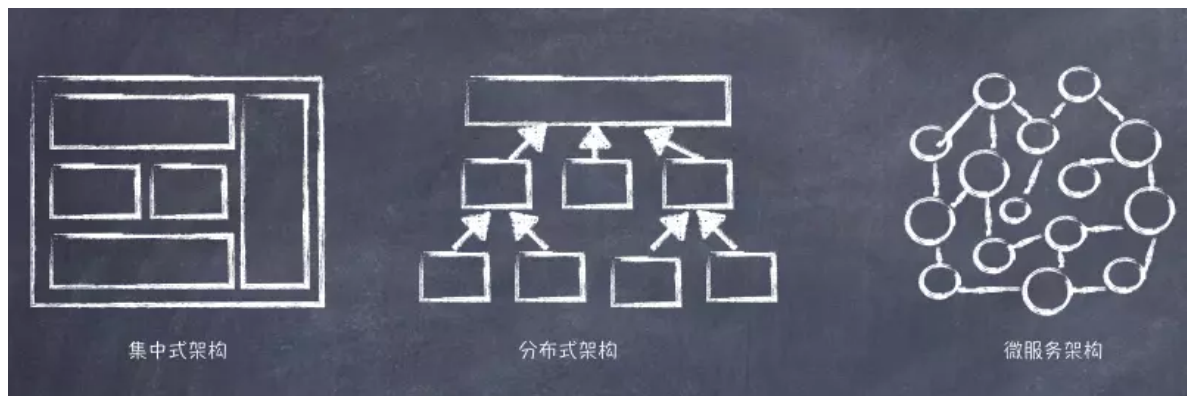
# 再谈Docker-微服务的场景化应用

我的名字是滨田宏,我发明了一些自认为很神奇的东西,希望你们能喜欢.这是一个微型机器人,它看起来微不足道,但当它和其他小伙伴们团结起来时,就变得有趣多了。它们由这个神经发射器控制,我想让它们做什么,它们就照做。这项创造的应用是无止境的。建筑物,曾经需要大队人马,人工建造数月或数年,现在只要一个人就可以完成。这仅仅是九牛一毛,可不可以用在交通运输上,微型机器人可以轻松移动任何物体至任何地方,只有想不到,没有做不到。你的思维局限是它唯一的限制。

看过《超能陆战队》的朋友可能仍然对于电影中的男主角介绍和演示自己发明的微型机器人的场景记忆犹新。“它”看起来只是一跟带有磁性的小小的金属部件。但是它是一个独立的个体,自己能够独立的大脑,同时,和同伴之间有相互的接口你行链接。能够通讯。能够随意的组合成任意功能的物体。通过类比,我们很容易由硬件领域想到软件领域。譬如软件系统的架构,一直都是伴随着几种主流的模式,集中式,分布式以及最近才开始流行起来的“微服务”。滨田宏发明的微型机器人,其实和微服务的思想很类似。一个微小的服务实体,有对外的接口与外部通讯。彼此之间能够快速组合成新的服务。其架构松散耦合。接下来的内容,会和大家一起分享微服务的特点以及所拥有的一些神奇的魔法。

## 什么是微服务?

微服务,至少我目前也没有找到一个很精确的标准化解释。所以我们首先从字面上来理解。既然是服务,那一定是一个能够实现某个功能的实体。光有功能,是不能成为一下服务的,因为还需要有途径和外部交互。让外部的实体能够获取服务。譬如web服务,通过http协议和浏览器或者app进行交互。所以微服务,一般来说,是有一套和外部通讯的标准接口的,譬如REST API。名字带了一个“微”字,说明提供的功能很小,或者很弱。但是一个非常小,或者非常弱的功能,是无法构成一个系统的,因此,他们之间,必须是能够相互组合的。在软件领域,一般把它理解成一种新的架构设计模式。可以和我们通常所熟知的软件架构做类比,譬如集中式架构,分布式架构。既然是一个很抽象的概念,那我这里也用一幅很抽象的图来表示。



## 微服务的特点

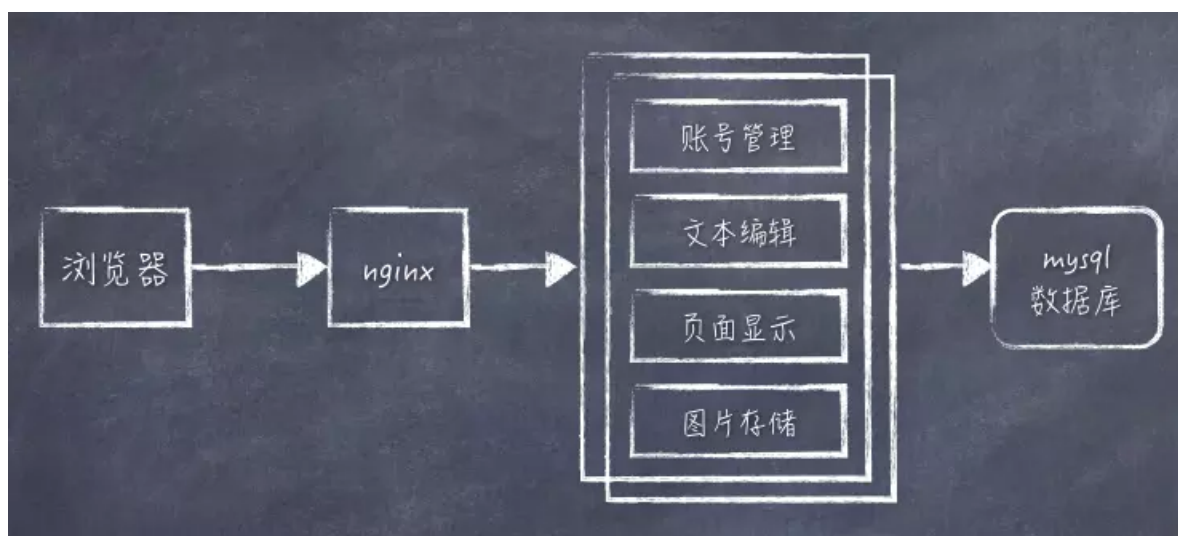
**彼此独立:** 既然是一个独立的服务,那必然是一个完整的自治系统,不依赖外部的东西就能够提供服务。有自己一整套的完整的运行机制,有和外部通讯的标准化接口。就像《超能特工队》里面滨田宏发明的微型机器人,它就是一个独立的小机器人。可以和其他的机器人通过磁性相互吸引,可以探测到彼此的存在。离开了其他个体,一样能够运转,只是功能比较单一。

**原子化**：作为一个微服务，一定是一个原子化的服务。也就是说服务不能再划分成更小的服务了。世界上的一些事物都是有原子构成的。它为什么能构成所有的物体，正是由于它足够的基础。如果一个服务还能划分成几个小的服务，那我们就不能称之为一个微服务，它其实可以通过几个微服务组合成的一个系统。

**组合和重构**：如果是最原子的服务，那一定是没有任何用处的。微服务之所以神奇，在于它能快速的组合和重构。彼此组合成一个系统。系统里面所有的实体在概念上是对等的。因此它的结构相对简单化。是一种松散耦合的结构，这样的系统，往往具有更强的可扩展性和鲁棒性。

## 微服务之于实践

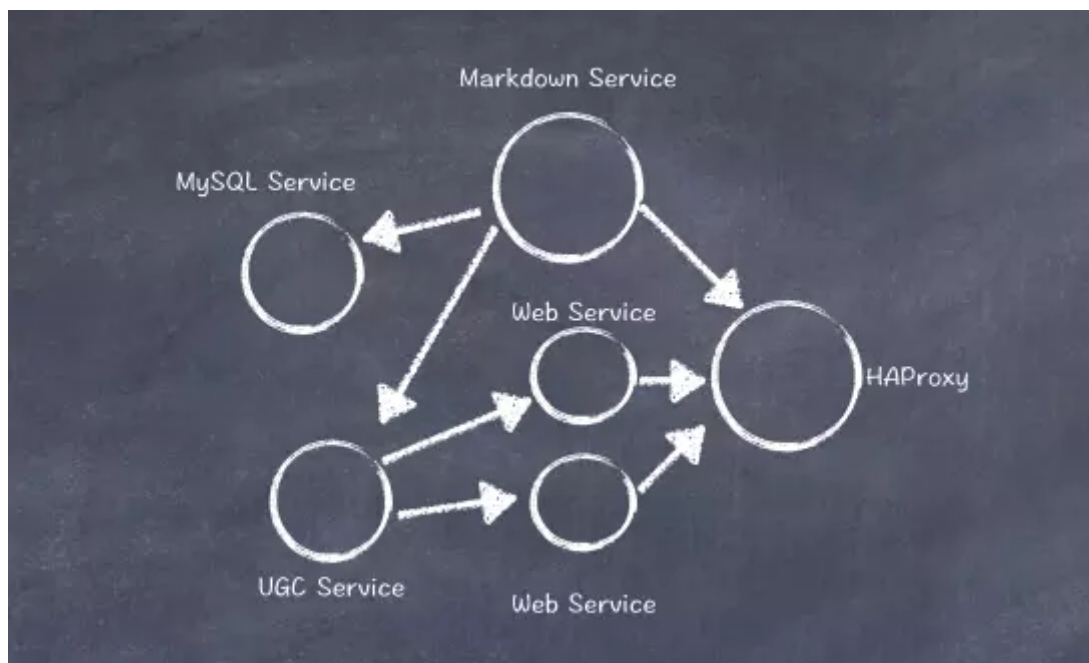
前面谈了这么多，可能大部分人还是没有明白微服务是个什么东西。我们试着可以通过一些的东西来描述。例如，我们使用ghost搭建了一套个人博客的系统。如果使用传统的架构，我们可能以模块的视角来划分，譬如可以分为“用户管理”，“文章编辑”，“页面显示”，“图片存储”，“文章分享”等几个模块，抽象成的架构图如下所示：



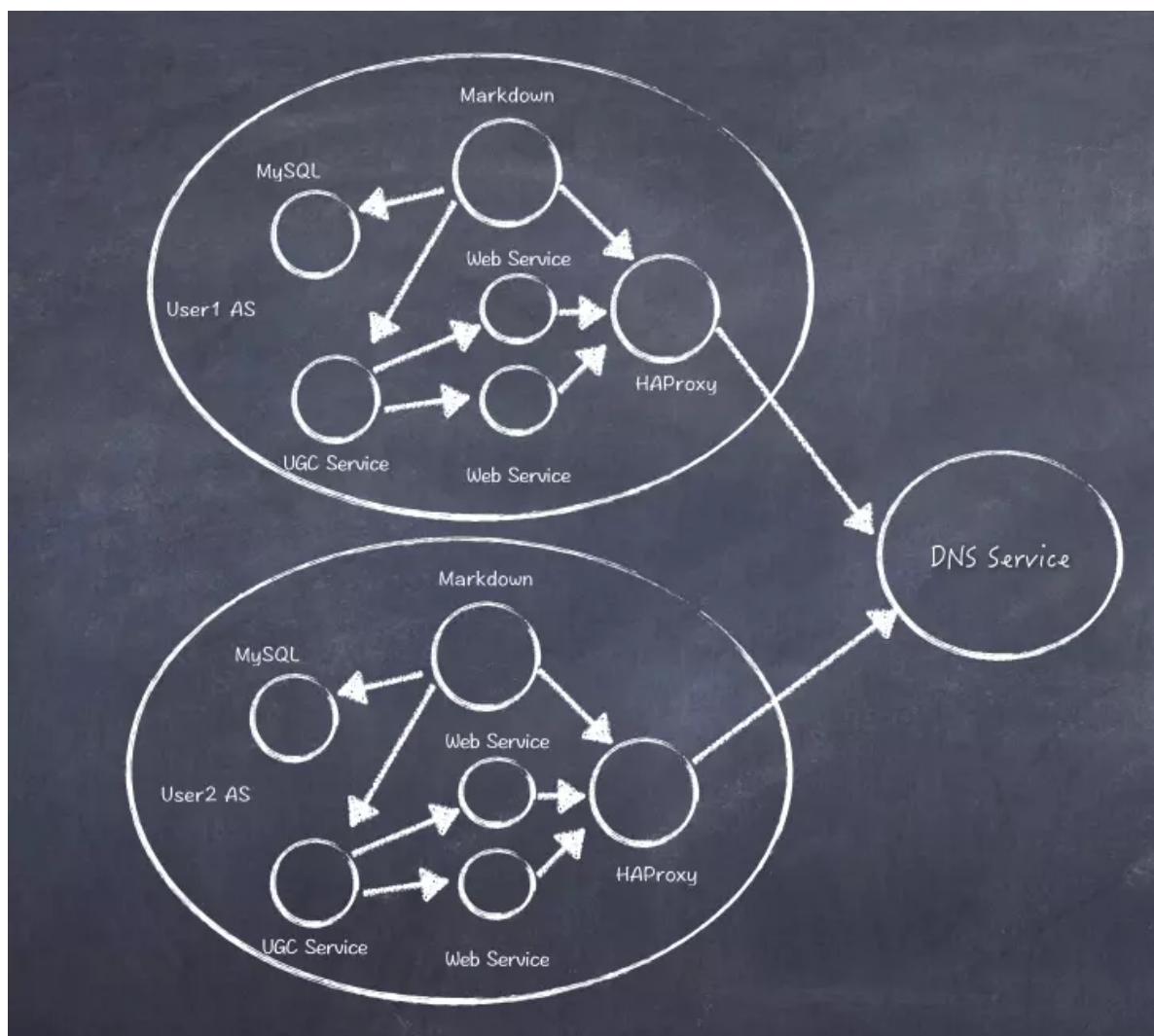
换一个视角，我们可以从服务的角度来思考。未来简单起见，我们先考虑单租户的场景。：

- **Markdown Service**
- **Web Service**
- **UGC Service**
- **MySQL Service**

基于微服务的架构，可能是下面这样一个图：



我们再想想，如果要提供多租户的服务呢？我们把数据库变大，存储多个用户的信息？这的确是一种思路，但是其思想有点和我们的微服务的思想背道而驰了。我们为什么不为用户配备这样一套服务呢，只要每个服务足够的微小，其实是没有太多的浪费的。上面图里构成的一套系统我们可以作为单独服务一个用户的自治系统。当用户增多时，就呈现出了一套去中心化的云服务的雏形。



## Docker在微服务系统中所扮演的角色

在Docker出现之前，虽然我们谈论微服务架构，但是其实是很难实现的。微服务要运行，首先需要一套执行的环境。这套环境不能对外部有依赖性。同时，执行环境的粒度又必须足够的小，这样才能称之为“微”，否则必然是对资源的巨大浪费。一个微服务可以跑在一台虚拟机上面，但是虚拟机粒度太大，即使最小的虚拟机，也至少也有1个核。正如我们上面的ghost博客的例子，服务一个用户的服务，显然用不了一个核。同时，虚拟机有没有一套方便的管理机制，能够快速的让这些服务之间能够组合和重构。Docker出现以后，我们看到了微服务的一个非常完美的运行环境。

- **独立性**：一个容器就是一个完整的执行环境，不依赖外部任何东西。
- **细粒度**：一台物理机器可以同时运行成百上千个容器。其计算粒度足够的小。
- **快速创建和销毁**：容器可以在秒级进行创建和销毁，非常适合服务的快速构建和重组。
- **完善的管理工具**：数量众多的容器编排管理工具，能够快速的实现服务的组合和调度。

除了Docker生态系统之内的一些工具，包括Serf之类的服务自发现技术的发展，可以让微服务能够自动化的感知其关联的其他服务，实现系统的自我构建。我记得2014年早些时候，centurylinklabs里面有一篇文章，讲述了如何通过FIG，Serf，HAProxy构建一个自动负载均衡的Docker应用。其实，这篇文章所蕴含的思想，就是一种微服务架构的概念。

[AUTO-LOADBALANCING DOCKER WITH FIG, HAPROXY AND SERF WITH DOCKER](#)


## 去中心化的云服务

---

最近一段时间，“场景化”是一个频繁出现的词汇。在这里，我也套用一下这个词，“什么是微服务的场景化应用？”。去中心化的云服务，是一个非常典型的应用场景。什么是去中心化的云服务呢？这里做一个类比，譬如家里的供暖，可以采用集中化的供暖方式。由电厂或者钢铁厂统一提供供暖服务。当然，也有的家庭自己会建设一套中央空调系统进行供暖。云服务，也会有类似的趋势。目前云计算的发展比较低级。主要是以托管为主，因此大部分还是中心化的云服务。随着云计算的应用越来越垂直化，必然也会出现越来越多的去中心化的应用场景。去年 iCloud 爆出了被黑客攻击。黑客攻破一家服务商，就直接窃取了所有用户的资料。这就是一种中心化的云服务带来的一些不利的因素。既然我们可以由统一的服务商来提供云服务。我们能否实现一套去中心化的服务呢？就拿个人云存储来举例。每个人都有一套个人的云的存储系统。这套系统运行在任意的提供“水和电”的基础云服务商的系统之上。并且可以任意在不同的服务商之间迁移和部署。不同的用户，可能位于不同的服务商之上。完全由自己控制的一套系统。每一套系统，都是一系列微小的服务组合而成。虽然底层也依赖基础云服务商，但是他们的作用更像水和电一样。

国内以微服务为基础的去中心化的云服务也已经有一些实践的例子，譬如terminal.com, dianCloud.com等，逐渐呈现出一部分这样的思想。借助于这样的服务，用户能够快速的构建一套属于自己的ghost博客系统，或者采用开源软件ownCloud搭建的个人云存储系统。选购他们，就像在商店里面选购商品一样，拿回家，插上电就可以用了。这种模式，也给开源软件找到了一个非常好的商业化的机制。我相信这种机制未来会越来越流行。






29.00元起


Ghost博客

[立即开通](#) [观看介绍](#)



WordPress - 功能强大的博客

[即将上线](#)

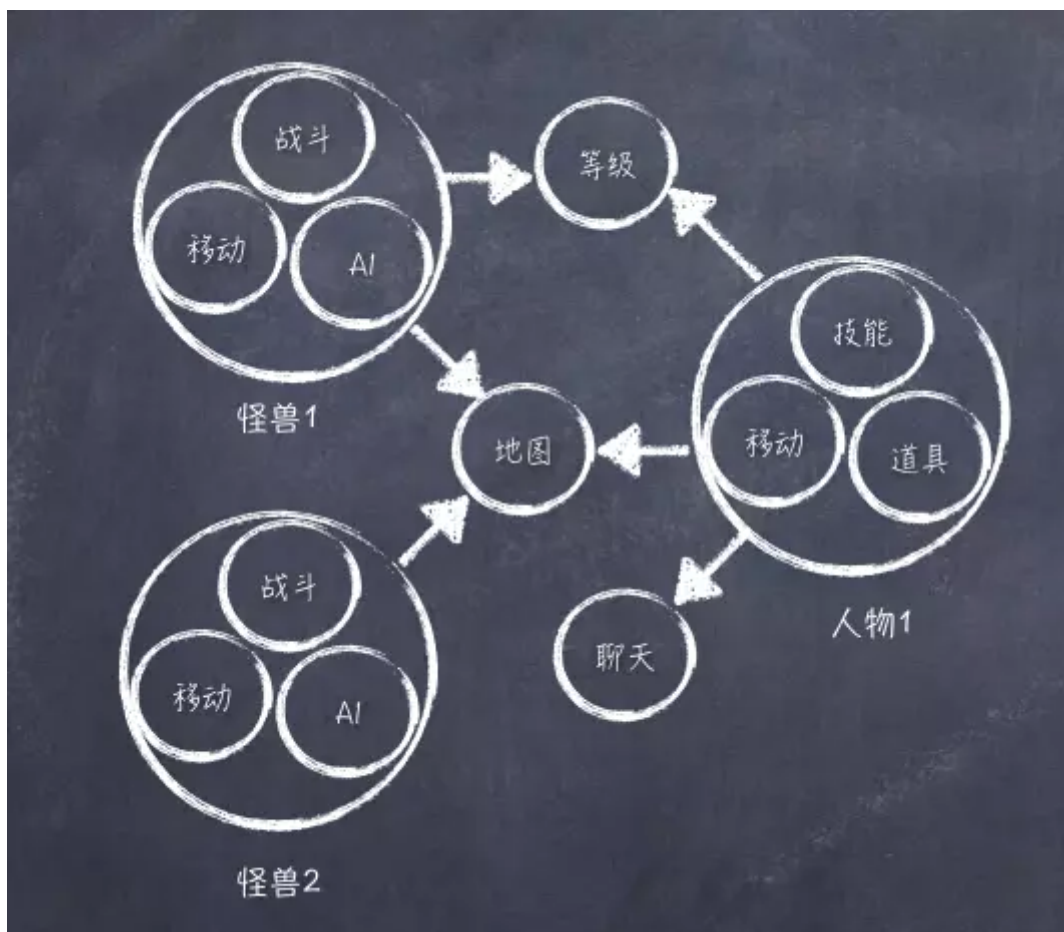


TeamToy - 创新团队的效率工具

[即将上线](#)

## 一个游戏架构的应用场景

游戏是一个比较特殊的行业。在国内，应该算是比较早拥抱云计算的一个行业，但是也是架构相对保守的行业。大部分的游戏架构非常简单。分布式的架构使用并不是太普遍，大部分是单区单服，一台强大的机器，运行若干个游戏服（游戏世界）。这并不是游戏架构落后，而是游戏本身的特点决定的。游戏一般以游戏服来划分，每个游戏服是一个独立的游戏世界。里面有一定数量的玩家。不能太多，也不能太少（总用户量一定的情况下，单服人数和总服的数量决定了游戏收入的最大化），两个游戏世界之间，数据不需要互通。因此通常都是一个进程搞定一个游戏服。其实这种模式下，微服务也是一个非常好的应用场景。我们知道，游戏其实有非常复杂的逻辑，譬如有控制人物移动的逻辑，控制道具，控制战斗，同时，游戏中还有成百上千的电脑控制的角色，每个角色都需要有自己智能。为什么我们不将这些细小的功能通过微服务来实现呢？譬如游戏中的一个单独的怪兽，可以由自己微服务构成的小的自治系统来控制。它可以完全独立，接收外部信息，做出反应。未来游戏公司可以复用这些单独的小系统。换上不同的皮肤，就可以用于不同的游戏。同时游戏其他的逻辑，都可以通过一些独立的微服务来构成。这些微服务可以借助Docker之类的系统，运行在容器中。能够快速的自动化的构建出一个完整的游戏世界。



## 后记

最近，基于Docker的创业公司不停的涌现，大家一夜之间似乎都在谈论Docker。但是我想说的是，Docker只是一项新的技术，消费者只会为服务买单，不会为技术买单。何况，对于圈子之外的大部分的消费者，云已经是其能理解的技术极限了，再来一个Docker，基本是无法理解的。因此如果想在Docker领域创业。停止谈论Docker，思考Docker技术之上的丰富的场景化的应用，才是关键。同样，微服务也只是一种架构思想。基于这种架构所带来的神奇的应用场景才是未来。

by: yongfeng