复用云技术

叶坤宇

定义

从最基本的层面上来讲，容器让你可以将更多的计算工作负载塞入到单单一台服务器上，并且让你可以在一瞬间为新的计算任务提高增加容量。从理论上来说，这意味着你可以购买较少的硬件，自建或租赁较少的数据中心场地，并且雇用较少的人手来管理这些设备。

具体来说，Linux 容器为服务器上运行的每个应用程序提供了独自、隔离的环境来运行，但是那些容器都共享主机服务器的操作系统。由于容器没很必要装入操作系统，你可以在一瞬间为虚拟机建立容器，而不是要花数分钟。如果应用程序遇到业务活动突然猛增的情况，比如人们进行更多次搜索或订购更多个产品，这样的速度让数据中心能够非常迅速地响应。

对容器技术的理解

正如集装箱彻底颠覆了全球运输业和世界经济，容器将统治世界。

《集装箱改变世界》这本书生动的讲述了一个集装箱改变全球运输业的生动故事，最终集装箱彻底改变了世界经济的运输方式。

集装箱最大的成功在于其产品的标准化以及由此建立的一整套运输体系。无论货物的体积、形状差异有多大，最终都被装载进集装箱里。由于要实现标准尺寸集装箱的运输，堆场、码头、起吊、船舶、汽车乃至公路桥梁、隧道等，都必须适应它在全球范围内的应用而逐渐加以标准化，形成影响国际贸易的全球物流系统。由此带来的是系统效率大幅度提升，运输费大幅度下降，地球上任何一个地方生产的产品都可以快速而低廉地运送到有需求的地方。

下图是集装箱与容器的类比

集装箱	类比	容器技术(Docker)
发货商	<=>	应用的发布者，现实中多为应用的生产方，即开发者
客户	<=>	使用应用的互联网用户
货物	<=>	构成应用的代码、组件、依赖等
集装箱	<=>	Docker容器
装卸货	<=>	应用的发布、撤销
码头工人	<=>	实际操作应用发布过程的人，现实中多为运维人员
散件装卸、运输方式	<=>	应用发布过程中逐个安装部署代码、组件、依赖、配置环境等
集装箱装卸、运输方式	<=>	把应用运行所需的外部环境、内部代码、组件、依赖打包放进容器，应用发布以容器为单位
港口的码头、起重机、集装箱堆场	<=>	应用发布所需的基础设施与工具
轮船/轮船公司	<=>	容器运行平台，如可以运行容器的云计算平台

关键技术

隔离

每个容器内都包含一个独享的完整用户环境空间，并且一个容器内的变动不会影响其他容器的运行环境。为了能达 到这种效果，容器技术使用了一系列的系统级别的机制诸如利用 Linux namespaces 来进行空间隔离，通过文件系统的挂载点来决定容器可以访问哪些文件，通过 cgroups 来确定每个容器可以利用多少资源。此外容器之 间共享同一个系统内核，这样当同一个库被多个容器使用时，内存的使用效率会得到提升。

安全

现在发表的有关在一台服务器上并排运行多个(比如 1200 个)容器的安全性方面的研究并不多。一个运行中的容器无法闯入或窥视另一个容器已分配的内存空间。

但是，如果允许两个容器彼此对话，其中一个容器被装入了恶意代码，窥视被允许查看的数据当中的加密密钥，那又会怎样?由于共享内存里面存在太多的变数，宝贵数据(比如用户 ID、密码和加密密钥)迟早会落到恶意软件的手中。

恶意代码还可能逐渐了解大体情况，知道一个或多个关联的容器在干什么。从理论上来说，这不会发生，因为容器旨在确保每个应用程序相互隔离。但是没有人确信计算机科学家是否想到并杜绝了可能出现某种恶意软件窥视行为的每一种情况。

彼此邻近的容器共享处理器、内存和磁盘等资源，这种行为让安全专业人员深为担忧。有人可能找到一种方法，让一个容器中的代码设法窥视或窃取另一个容器中的数据，尽管之前还没有人这么做过。

环境的一致性

开发工程师完成应用开发后 build 一个 docker image，基于这个 image 创建的 container 像是一个集装箱，里面打包了各种“散件货物”(运行应用所需的程序，组件，运行环境，依赖)。无论这个集装箱在哪里：开发环境、测试环境、生产

环境，都可以确保集装箱里面的“货物”种类与个数完全相同，软件包不会在测试环境缺失，环境变量不会在生产环境忘记配置，开发环境与生产环境不会因为安装了不同版本的依赖导致应用运行异常。这样的一致性得益于“发货”（build docker image）时已经密封到”集装箱“中，而每一个环节都是在运输这个完整的、不需要拆分合并的”集装箱“。

Build Once, Run Everywhere

这个特性着实吸引了我，“货物”（应用）在“汽车”，“火车”，“轮船”（私有云、公有云等服务）之间迁移交换时，只需要迁移符合标准规格和装卸方式的“集装箱”（docker container），削减了耗时费力的人工“装卸”（上线、下线应用），带来的是巨大的时间人力成本节约。这使未来仅有少数几个运维人员运维超大规模装载线上应用的容器集群成本可能，如同 60 年代后少数几个机器操作员即可在几小时内连装带卸完一艘万级集装箱船。

挑战

容器技术在传统的虚拟化技术当中引入了大量的最新发展趋势，比如云计算、在应用程序开发方面进行的相应调整以及像 Docker 这样强大的新型容器架构。在奥兰多举行的 2015 Gartner IT 运维战略和解决方案峰会上，Gartner 副总裁兼著名分析师 Thomas Bittman 以容器技术为主题进行了演讲。Bittman 的演讲列举了一些容器技术的优势，但是同时也指出了其存在的很多缺陷。下面我们将逐个分析这些缺陷并且讨论如何进行解决。

不能应用在所有场景当中

Bittman 认为虽然容器技术拥有很强的兼容性，但是仍然不能完全取代现有的虚拟机环境。就像虚拟化技术刚刚出现的时候，一些传统的应用程序更加适合运行在物理环境当中一样，现在，一些应用程序并不适合运行在容器虚拟化环境当中。

比如，容器技术非常适合用于开发微服务类型的应用程序——这种方式将复杂的应用程序拆分为基本的组成单元，每个组成单元部署在独立的容器当中，之后将相关容器链接在一起，形成统一的应用程序。可以通过增加新的组成单元容器的方式对应用程序进行扩展，而不再需要对整个应用程序进行重新开发。

但是另一方面，一些应用程序只能以统一整体的形式存在——它们在最初设计时就采用了这种方式，很难实现高扩展性和快速部署等特性。对于这种情况来说，容器技术反而会对应用负载造成限制。最好的检验方式就是进行大量试验，查看哪种现有应用程序能够通过容器技术发挥最大优势。一般来说，新的应用程序研发过程很可能从容器技术当中获益。而那些不能被容器化的应用程序仍然可以运行在传统 hypervisor 的全功能虚拟机当中。一位来自知名保险提供商的 IT 架构师表示应该放缓应用程序容器化趋势。“虽然容器技术非常具有吸引力，但是软件开发团队需要一段时间及时跟进，才能够真正地高效利用容器技术所带来的优势。”

难以解决依赖关系问题

大多数虚拟机都是相对独立的，每台虚拟机都包含自己的操作系统、驱动和应用程序组件。只要拥有合适的 hypervisor，还可以将虚拟机迁移到其他任何虚

拟化平台当中。但是对比来说，容器运行在物理操作系统之上，相互之间共享大量底层的操作系统内核、库文件以及二进制文件。Bittman 进一步解释说容器之间的现有依赖关系可能会限其在服务器之间的可移植性。比如，位于 Linux 操作系统上的 Docker 容器就不能运行在当前版本的 Windows Server 操作系统上。

对于这种问题来说，当前的解决方案并不止一种——容器可以在数秒钟之内完成复制过程，操作系统也在不断发展，开始提供“micro OS”和“nano OS”等多种类型，提供了高稳定性以及快速重启等特性。从容器自身的角度来说其更加适合于这些环境，只要数据中心当中的其他服务器可用，仍然能够对其进行迁移。

随着操作系统的逐渐发展，这些依赖关系问题也在不断得到解决。比如，Windows Server 2016 承诺同时支持 Docker 和原生 Hyper-V 容器。除了 Docker 之外，还有许多其他容器平台可供选择，比如 LXC、Parallels Virtuozzo、Joyent、Canonical LXD、Spoon 等等，VMware 也有可能随时加入到竞争行列中来。

较差的隔离性

基于 hypervisor 的虚拟机拥有完善的隔离特性，由于系统硬件资源完全是虚拟的，由 hypervisor 分配给虚拟机使用，因此 bug、病毒或者入侵有可能影响一台虚拟机，但是不会蔓延到其他虚拟机上。

容器的隔离性较差因为其共享同一个操作系统内核以及其他组件，在开始运行之前就已经获得了统一的底层授权（对于 Linux 环境来说通常是 root 权限）。因此，漏洞和攻击更加有可能进入到底层的操作系统，或者转移到其他容器当中——潜在的传播行为远比最初的事件更加严重。

尽管容器平台也在不断发展，开始隔离操作系统权限、减少脆弱的安全特性等，但是 Bittman 仍然推荐管理员通过在虚拟机当中运行容器来提升安全性。比如，可以在 Hyper-V 当中部署一台 Linux 虚拟机，在 Linux 虚拟机当中安装 Docker 容器。这样即便虚拟机当中的容器出现问题，这种漏洞也只存在于当前虚拟机当中——限制了潜在的受攻击范围。

潜在的蔓延问题

就像虚拟机生命周期管理对于 hypervisor 环境来说十分重要一样，生命周期管理对于容器来说也是至关重要的。容器可以被大量快速复制，这是容器技术的重要优势之一，但是也有可能在管理员没有注意到的情况下消耗大量计算资源。如果应用程序所在的容器不再使用时能够被及时删除，那么情况还不算太坏。但是如果对一个容器化应用程序进行扩展之后忘记将其缩减回之前的规模，那么将会为企业带来大量的（并且不必要的）云计算开销。Bittman 还表示云提供商十分高兴看到这种情况发生——因为他们就是通过出租计算资源而获利的——因此用户需要自己关注容器的部署情况。

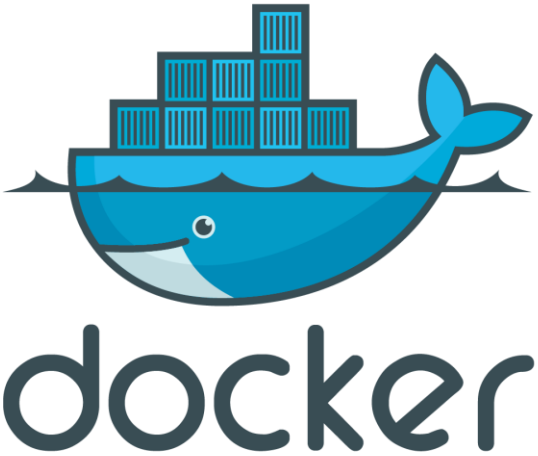
缺乏工具

对于这个行业来说，用于监控和管理容器的工具种类仍然十分缺乏。这并不是一种最近产生的现象，在基于 hypervisor 虚拟化的早期也曾经出现过可用工具十分匮乏的情况。就像优秀的虚拟机监控和管理工具逐渐增多一样，容器管理领域也在不断出现新的工具。其中包括谷歌的开源 Docker 管理工具 Kubernetes，此外 DockerUI 使用基于 web 的前端界面替换 Linux 的命令行功能，Logspout 能够将容器日志汇集到一个集中位置。

Bittman 建议管理员可以通过将容器运行在虚拟机当中缓解容器管理工具缺乏的问题，这样就可以使用虚拟机工具来完成一些监控和管理功能了。因为虚拟机工具更加成熟和多样化，因此在容器工具逐渐成熟之前，可以将其作为临时的替代产品。

Bittman 对于容器技术充满热情，认为其能够快速交付轻量级的应用程序，提升资源使用效率和扩展性；容器自身（非虚拟化 I/O）还能够实现更好的性能表现；已经拥有像 Docker 这样优秀的开发架构，像 GitHub 这样吸引广泛关注的共享和协作平台。但是容器并不是一种能够满足所有虚拟化任务的解决方案，只是虚拟化工具箱提供的另外一种工具——通常可以和传统虚拟机很好地协同工作。

容器技术方案



Docker 已成为容器的代名词，但它不是唯一的提供者。

Docker 这家公司提出了一种标准的方法来构建容器工作负载，那样工作负载就可以四处移动，但是在任何支持容器的环境中以易于预测的方式来运行。

无论是基于 Linux 的 Docker 容器、Solaris Zone 还是 FreeBSD Jail，所有容器都为在多应用程序主机上运行的应用程序提供某种隔离机制。既然如此，眼下我们为什么听到的除了 Docker 还是 Docker 呢?答案在于这个事实：Jail 和 Zone 确实是容器领域的开拓者，但是它们的采用很有限，这归咎于这个事实：使用 Solaris 和 FreeBSD 操作系统的公司比较少。它们仍在企业得到使用，但是在公有云环境下只是稍微得到使用。

谷歌和开发 Linux 控制组(Linux Control Groups)的开发人员成功地将容器功能添加到 Linux 内核中后，鉴于 Linux 几乎遍地开花的态势，容器立马进入到了每一个企业和政府数据中心。

Docker 这家公司大概也在同一时间成立。开发人员明白，如果有一种统一的方法来创建并移动容器，而不是有五花八门的容器格式化引擎，那么容器的实用性和移植性就会大大加强。眼下，Docker 就是那种事实上的标准。

正如 Docker 的首席执行官 Ben Golub 喜欢说的那样，Docker 就好比船运集装

箱。每家货运公司、铁路公司和船厂都知道如何装卸和移动标准的船运集装箱。Docker 容器在一系列广泛的计算环境下同样备受欢迎。

如何复用

我们以 LBE 推出的新应用——平行空间 (Parallel Space)为例，看一下容器技术在客户端的应用。

想同时登陆两个微信，很多人可能需要两个手机，但对于 Android 用户来说，你多了一种更加方便的选择，使用 LBE 推出的新应用——平行空间 (Parallel Space)。

它可以帮助用户在一部安卓手机中同时登陆两个社交或游戏账户，目前已经支持包括国内和海外的绝大多数安卓应用——微信、QQ、陌陌、微博、Facebook、WhatsApp、Instagram、Twitter 等社交应用，以及部落冲突、海岛奇兵、皇室战争等游戏应用。LBE 平行空间已经在国外获得大量用户，进入多个国家 Google Play 免费工具榜单 TOP10。

不过一提到虚拟化，就很容易想到在 Windows 上装个 VMware，对性能有一定影响。但平行空间使用了容器技术，其好处在于不依赖特定的硬件、操作系统的底层权限，而最大的好处就是轻，对手机性能的影响很小。

在平行空间的背后是 LBE 推出的 MultiDroid——移动计算平台上的虚拟化系统引擎。在 LBE CEO 张勇看来，平行空间只是 MultiDroid 引擎的应用之一，未来围绕 MultiDroid 会有更大的想象空间。

例如虚拟化技术一直应用于云端，而平行空间将其运用到消费者端，在后续平行空间会也会为用户提供一个安全环境，通过使用不可逆的加密方式，确保支付场景或其他场景的安全需求。

Reference

容器详解：你不可不知的九个基本事实
http://tech.idcquan.com/XuNi/74658.shtml
Docker 容器技术为何如此重要？
http://server.ctocio.com.cn/133/13583633.shtml
虚拟化 VS 容器化
http://www.oschina.net/news/61820/virtualization-vs-containerization
为什么容器技术将主宰世界

http://blog.csdn.net/gaoyingju/article/details/49616295
容器技术的五大缺陷
http://www.searchvirtual.com.cn/showcontent_89711.htm
拿了阿里腾讯小米投资，LBE 将容器技术运用到消费端，推出应用“平行空间”
http://36kr.com/p/5046431.html