新浪微博如何应对极端峰值下的弹性扩容挑战?

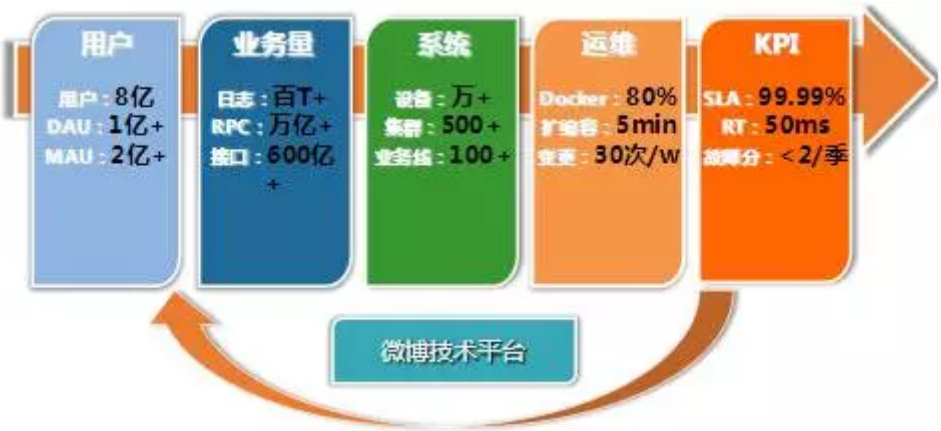
原创：付稳 51CTO技术栈 2017-07-19

新浪微博在 2015 年春晚便通过 Docker 实现了私有云平台的弹性调度能力，随着公有云技术的成熟，我们发现原有私有云比较困难的问题在公有云上能够比较容易的解决，例如突发峰值情况下弹性资源的成本，小业务快速试错等场景。

在 2016 年，微博完成了利用 Docker 构建混合云架构，本文将分享安全、网络、资源管理、调度管理、跨云服务发现等方面的一些实践经验。

新浪微博庞大的数据背后是持续不断的技术挑战

微博的数据量，在国内的社交媒体中排行位居前列。如下图：



百亿级 PV、千亿级数据、万台以上的服务器规模、数百以上服务模块、千台以上的 Docker 混合云集群等等，这些庞大数据背后，是持续不断的技术挑战。

在如此大的业务体系下，业务流量有很明显的特征：

- 每年春晚当天流量会达到一年中峰值，这样就会面临机架位和上千台服务器库存不足的情况，因此采购成本巨大且周期长，运行三个月仅仅只为一晚的流量峰值。
- 白百何出轨、李晨范冰冰在一起等热点事件，突发性强无预期、无准备，瞬时极端峰值、互动周期短，在这样的情况下，Push 常规化，短时间内大量扩容的需求会很迫切。

传统的服务扩容流程非常繁琐，整套流程要经历项目评审、设备申请、入 CMDB、装机上架、初始化、服务部署和报修下架等过程。

如何在十分钟内完成 1000 节点扩容能力

那么，如何才能快速顺利的应对各种流量峰值呢？首要解决的是如何在十分钟内完成 1000 节点扩容能力。

如下图，是应对极端峰值问题的解决思路。

基于混合云弹性调度可伸缩的特性，可以保证成本业务快速迭代的情况下，实现弹性快速的扩缩容。选择混合云是因为安全，具备可扩展性，成本相对较低。还有 Docker、Mesos 等容器新技术使大规模动态调度成为可能。

新浪微博 DCP 设计与实现

Docker化

在说 DCP 之前，我们先来了解一下 Docker。微博业务部署涉及 Java、PHP 等不同的语言，且存在环境差异，如依赖 OS、JDK、Nginx 等操作环境，还涉及依赖脚本、基础环境配置（启动脚本、定时任务）、目录结构等。

一旦数据量来临时，就要统一调配，导致整体的运维和研发效率在环境差异和不同语言下，非常低。这是决定做 Docker 的主要原因。

微博 DCP 技术架构演进

从 2014 年开始，新浪微博做单机容器化和在线 Docker 集群。2015 年，基于 Docker 的思维做弹性调度，服务发现与私有云建设。

2016 年，开始做混合云的部署，当下在做混合云与机器学习的支持，同时混合云 DCP 技术进行开源 OpenDCP：https://github.com/weibocom/opendcp。

混合云 DCP 技术架构

如下图的 DCP 架构:

DCP 架构底层私有云采用的是基于 OpenStack，公有云是和阿里云合作。整个架构从上到下分为编排、调度和主机三层。

当流量来临时，主机层通过私有云和公有云的 SDK 进行主机的创建，之后做初始化达到快速上线的目的。初始化主要做运维环境和 Docker 环境的安装。

初始化之后，编程可运行 Docker 环境，在经过容器调度和服务编排之后，会自动被服务发现模块引入流量，进行上线。

当然，整个大的体系还需依赖基础设施，如镜像中心，监控中心和容量评估等。

混合云 DCP 技术栈

如下图，是混合云 DCP 技术栈。

Docker 为什么会用 Host 模式？因为微博高并发的特性，在最开始验证时，性能的消耗非常大，所以选用 Host。

混合云 DCP 核心设计思想

什么是共享池？就是在私有云内部，不同的业务方，在不同的时间内，资源利用率会有所差别，把资源利用率低的共享到共享池，提供给资源利用率高的服务池进行使用。

DCP 大规模集群扩容方式有私有云弹性扩容、公有云弹性扩容和两者同时弹性扩容。

混合云 DCP 设计的核心思想是如何解决设备从哪里来的问题，当设备到位，如何进行一体化扩容，来快速应对峰值流量。如下图是具体的设备方案：

具体设备方案是内网共享池+公有云 = BufferPool。如下图，是 DCP 资源共享示例。

混合云 DCP 扩容流程

如下图，是混合云 DCP 整个扩容流程。

混合云 DCP 整个扩容流程分为主机申请、初始化、动态调度、服务发现和下线五大步骤。

综上是混合云 DCP 的整个实践流程，下面主要分享十分钟内完成 1000 节点扩容能力所带来的问题，主要涉及基础设施和弹性调度两方面。

微博 DCP 之基础设施

统一资源管理

如下图，是微博统一资源管理图。

基础设施部分，不仅包括内网集群、ECS 集群，SLB 等物理资源，还包括新建 IDC 资源时，所依赖 yum 源，镜像仓库、DNSServer 等，从 Docker 层以下都归类为基础设施。

单机部部署方案

如下图，是单机部署采用镜像的方式。

运用 Docker 做统一化部署，把代码、运维组件、监控组件等全部封装到容器中，这样一来，就打通了差异化，不管是 Openstack、还是阿里云机器都可直接使用。

这里遇到一个很大的问题，就是镜像仓库。假设一个镜像是 1G，如果十分钟之内扩容 1000 台，那就是 1000G 都需要做镜像拉取。

但任何一个分布式存储或镜像仓库都无法满足。微博通过镜像分层服务、优化带宽等方法来应对。

镜像分层服务

把镜像进行分层，逐层复用。底层部分放入不会变的镜像，如阿里云、Openstack 的操作系统、JDK、Tomcat 镜像。这样做会使得环境构建的速度大大加快，剩下的可变代码和配置部分只有约三百兆左右。

Docker Registry

根据多次大规模扩容经验来看，做镜像分层之后仓库还是扛不住，带宽是瓶颈。故构建私有 RegistryHub，在内网和阿里云分别搭建镜像缓存 Mirror。

如，阿里云端用户进行扩容时，Docker Client 就可以直接拉取镜像，而不用穿过内网的 IDC。同时，内网和阿里云的镜像缓存 Mirror 都支持分布式存储。

如下图，是 Docker Registry 部署架构：

通过这样的架构流程，300 兆的镜像拉取，500 台服务器在 1-2 分钟之内就可以完成。

DCP 中 SLB 的应用

在混合云端，经过实践经验，选择使用 SLB 来做负载均衡。

如上图，是通过 SLB 做负载均衡的流程，红包飞业务就是通过 SLB 来做的负载均衡。

DNS 智能解析

如上图，是 DNS 智能解析流程图，阿里云所有域名解析都会在阿里云完成，不会穿到内网，这样一来，加大了域名解析的性能。

专线网络架构

如上图，通过路由配置分散两条专线压力，可随时切换。还有 VPN 做备用以及不同业务划分网段，便于监控专线带宽的使用情况。

DCP 的弹性扩容

第一步：主机申请

做好镜像仓库、SLB 和基础网络之后，就可以进行主机申请了。步骤如下：

首选向内网私有云进行申请，如共享池（离线集群，低负载集群，错峰）不足，再向阿里云申请。

第二步：初始化

主机申请下来之后，进行初始化，具体操作如下图：

初始化主要做的两件事情就是运维环境安装和 Docker 环境安装。在初始化过程中，阿里云配置管理选择的是 Ansible，因为 Ansible 并发性能差，初始化流程将需要数分钟。

但实际情况不允许，所以针对这个问题，我们做了异步列队、高并发下水平扩容、分布式改造等优化。

微博 DCP 的弹性调度

申请主机，经过初始化之后，批量的 Docer 资源已经进入 Docker 调度池了，接下来要做的事情就是对容器进行调度。

弹性调度是混合云 DCP 整个扩容流程的第三步，是重中之重。

新浪微博的诉求是服务池动态跨池缩容、容量评估，多机型署等，所以资源调度框架架构的设计目标是可以实现快速迭代，内网计算资源统一管理调配，公有云上获得计算资源，快速自动化资源调度与应用部署。

弹性调度架构的选型

业界很多大会都在讲 Swarm、Mosos 和 K8s 这三个弹性调度架构。我们综合资源利用、业务压力指标等考量后：

- 初期阶段，为了快速上手，响应应用，新浪微博选用 Docker 原生的 Swarm。
- 中期阶段，随着业务发展、Swarm 调度性能、高可用及调度算法表现不足，需要做一系列改造，才能应对当时需求。同时也选择使用 Mesos 对非容器进行管理。
- 后期阶段，因为对 Swarm 的源码改动太多，所以被放弃。之后底层选用 OpenStack，容器调度方面选用新浪自研的 Dispatch 做任务调度。

任务调度框架 Dispatch

Dispatch 调度框架的主要特点是使用任务模板，主要原因是容器启动之后，不是简单的上线，而是要先预热，对整个容器按步骤进行编排之后再上线。具体编程流程，如下图：

对每台机器布设一个 agant，在启动之后，进行容器编排，经过预热、健康检查、服务发现流量、引入等步骤。同时会向主进程进行汇报，汇报过程中进行严格批次，按照不同概念去执行。

弹性扩容流程

回顾整个弹性扩容的流程，如下图：

向混合云平台发布请求，做资源评估，如私有云资源不足，则向公有云进行申请。之后进行初始化，做容器调度、部署服务，最后进行服务注册，整个服务要在 10 分钟之内完成。

微博 DCP 之服务发现

完成申请、初始化、调度之后，进入第四步服务发现，这里要做的是，找到新扩容的节点，做好流量的迁移。

如何把流量快速、安全的切换到弹性节点呢？如下图：

微博有十几个严格的服务池，这些服务池按照复杂的规则进行划分。这里涉及到很多服务器的流量调动，所以需要有服务发现体系来支持。

面对 Reload 损耗的问题，开源解决方案大多利用 Nginx 的 Reload 机制。但在请求量方面，普通 Reload 会导致吞吐量下降 10%。微博的应对方案是 nginx-upsync-module，如下图：

当 Docker 启动之后，支持基于 Consul 的自动服务发现。同时 Core-module 模块会从配置中心把 Docker 节点自动拉入，做平滑 reload。这样一来，可减少扩容时性能的波动。

新浪微博为什么要 OpenDCP?

当下，对于一些创业公司使用 Docker 相比较难，现在乃至未来，微博要把 DCP 整套技术体系进行开源。

由于微博业务的特殊性带来很大压力，流量成倍增长，短时间内要扩大到超大规模，会带来很多技术问题或难点需要应对。开源是希望可以把这些经验做输出，使得更多人得以借鉴。

OpenDCP 地址：https://github.com/weibocom/opendcp，在这里也欢迎大家一起来建设。

以上内容由编辑王雪燕根据付稳老师在 WOTA2017 “容器技术实践”专场的演讲内容整理。

作者：付稳
编辑：王雪燕、陶家龙、孙淑娟
技术编辑：王雪燕，关注架构、算法，运维等技术领域，有投稿、寻求报道意向技术人请联络 wangxy@51cto.com

付稳

新浪微博技术专家

微博混合云 DCP 项目技术负责人，借助公有云弹性计算资源平台应对爆发式峰值流量，基于 Docker、Swarm 等容器云技术体系实现分钟级千台规模机器创建及服务部署自动化运维体系。参与微博混合云、Feed 混合云多机房部署改造、微博春晚保障工作、Feed 性能优化、HBase 改造等重量级架构改造项目，对高可用架构、混合云平台建设、多机房部署、应用性能跟踪及分析、业务技术保障等方面有深入研究。

精彩文章推荐：

- 新浪微博应对弹性扩容的架构演进
- 揭秘百亿级云客服实时分析架构是怎么炼成的?
- 微服务架构的两大解耦利器与最佳实践