文章编号：1005-8451（2015）11-0014-03

集群节点动态调整技术在互联网分区集群中的研究

苗　凡，朱建军，戴琳琳
（中国铁道科学研究院　电子计算技术研究所，北京　100081）

摘　要：针对售票高峰时期对计算资源的需求紧迫，而非高峰时期的需求不突出的现状，提出集群弹性计算的优化方案，满足互联网分区服务器资源充分利用的需求，并对该方案的实现与可行性进行深入的分析与研究。

关键词：集群；负载均衡；自动扩展；容器
中图分类号：U293.22：TP39　　**文献标识码：**A

Dynamic regulation technology of cluster node in Internet partition cluster

MIAO Fan, ZHU Jianjun, DAI Linlin
(Institute of Computing Technologies, China Academy of Railway Sciences, Beijing　100081, China)

Abstract: During holidays, computing resources for INETIS were urgent in the rush hour of ticket sale, while during workdays, demands for computing resources were not very prominent. According to these current situations, the article put forward a solution of cluster elastic computing to make full use of resources, made an in-depth analysis for the implementation and feasibility of the solution.

Key words: cluster; load balance; auto scaling; container

集群节点动态调整技术，即弹性计算，就是在负载均衡的基础上根据负载的大小自动增加或减少节点的技术。与传统的手工增删节点相比，弹性计算具有响应时间短，运行维护成本低，稳定性高等特点而受到业界的广泛重视。

新一代中国铁路客票发售和预订系统（简称：客票系统）经过 3 年的发展，已经成功完成 3 次春运大考，单日售票量突破 1 000 万，互联网渠道超过 600 万。在旅客畅享回家团圆的背后，客票系统的扩展性，稳定性，容灾能力，运行维护能力，紧急故障处理能力承受了严峻的考验。春运结束后，当初以峰值配置的服务器出现 CPU、内存、网络使用率较低的现象。如果引入弹性计算将可以有效地管理服务器资源，提高资源利用率，将闲置的计算资源投入到新的业务上，让传统的人工新增服务器上线扩容逐渐过渡到全自动维护，节约硬件成本与运行维护人力成本，对新一代客票系统的建设具有重要意义。

收稿日期：2015-04-10
基金项目：中国铁路总公司科技研究计划项目（2013X012-A-1，2013X012-A-2，2014X008-A）。
作者简介：苗　凡，助理研究员；朱建军，副研究员。

1　研究内容

新一代客票系统中的应用服务器 INETIS（互联网分区服务器），主要负责将互联网、手机等渠道的业务请求准确分发到各数据节点上，同时提供一个安全、高性能、高扩展、稳定可靠的中间件服务。作为新一代客票系统的核心模块，INETIS 的可靠性和可用性直接影响着系统的交易性能和用户体验。由于 INETIS 的业务多且用户基数大，在节假日高峰，每个放票时间点都会产生犹如洪水般的网络流量，固定数目的物理节点无法支撑不可预见的并发量。本文将研究一种基于 Mesos、Marathon 的高可用 Docker 集群架构，来保证压力暴增 10 倍的情况下 INETIS 仍能正常提供服务。它具有以下功能：

（1）自动实时，无感知服务刷新；
（2）支持任意多台 Docker 主机；
（3）支持负载均衡，故障迁移；
（4）具备资源弹性，自动增删节点；

（5）具备健康检测功能，支持高可靠性。

2　关键技术

2.1　Docker

随着企业服务的大规模部署，单台应用服务器的CPU、内存、网络连接的处理能力已成为瓶颈，难以满足大规模、高并发的需求，这就需要服务器集群。在X86时代，虚拟化技术因能提高企业资源的利用率，而成为当时构建大规模集群的主流技术。随着云计算时代的到来，企业对应用服务器的安全性、隔离性要求越来越高，对于部署的标准化以及虚拟机的性能要求越来越高，容器技术的出现解决了以上问题。

Docker是以Linux容器（LXC，Linux Container）为基础，实现轻量级的虚拟化解决方案。在LXC的基础上Docker进行了进一步的封装，让用户不需要去关心容器的管理，使得操作更为简便，用户操作Docker容器就像操作一个快速轻量级的虚拟机一样简单。

通过虚拟机搭建的集群自动扩展会面临许多问题：虚拟机提供的是完整的操作系统环境，迁移的时候包含了大量类似硬件驱动，虚拟处理器，网络接口等并不需要的信息。虚拟机启动时间长，同时也会消耗大量的内存及CPU资源等。

Docker运行起来就和一个常规程序差不多，与虚拟机相比就显得非常轻量级，而且解决了虚拟机面临的以下问题：

（1）用户可以根据自己的需要定制服务所依赖的环境，同时保持环境统一；

（2）Docker镜像（image）的Tag功能，有助于知道各个镜像的功能和内容，使得部署和升级更加方便；

（3）Docker创建一个镜像和制作一个系统快照只需要几秒钟，还具有运行性能高，启动速度快的优点。

2.2　Mesos和Marathon

Mesos是一款开源的集群资源管理平台，它的主要作用是将集群中的所有机器抽象成一个大的计算机，运行在Mesos上的所有服务，都不需要关心自身运行在哪个机器上，而只需要关心集群中的资源是否充足即可。

Marathon是一个基于Mesos的轻量级调度框架，它随着Mesos一起运行，并且在运行工作负载的同时提供了更高的可用性。它具有良好的扩展性，支持RESTful api来创建和管理应用服务器，自动为应用服务器做容错迁移。Marathon还支持在同一组服务器上运行多种类型的分布式系统如Hadoop，Spark等，并提供失败检测、任务发布、任务跟踪、任务监控、资源管理与资源共享等功能。

3　系统架构

3.1　总体结构

基于现有的INETIS集群的新一代客票系统总体结构如图1所示。

Mesos与Marathon管理并提供统一的计算资源池；互联网用户请求发送给Haproxy，并通过haproxy-marathon-bridge实现服务发现与负载均衡；Mesos master cluster负责管理Docker集群；Docker registry为私人Docker库，Docker通过Marathon api发布到新的计算节点上；Docker cluster负责所有的INETIS业务。

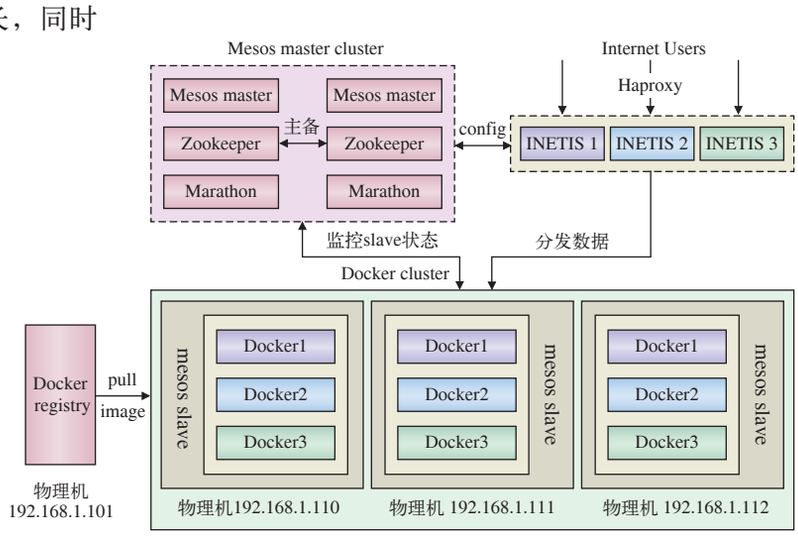

图1　新一代客票系统总体结构

从图1可以看出，Mesos是一个master/slave结构，其中Mesos master与Zookeeper，Marathon一起安装在多台物理机上。Mesos master使用Zookeeper进行服务选举和发现，来解决Mesos master的单点

故障，Zookeeper 将保证 Mesos 存在多个 masters，且在 masters 中选取一个作为 active 的 master，当其出现故障而无法工作时，能选取另一个备用的 master 让 Mesos 的 slave 连接到新的 master，让 Mesos cluster 继续提供服务。

Mesos slave 的主要功能是向 Mesos master 汇报任务的状态并为每个 INETIS Docker 实例提供计算资源。Mesos 还配置了 Docker registry 的信息。Mesos slave 具有恢复机制，即使一个 Mesos slave 死机了，用户的任务还是能够继续运行，Mesos slave 将一些关键点信息如任务信息，状态更新持久化到本地磁盘上，重新启动时可以从磁盘上恢复运行这些任务。

最下面一层为集群资源池，里面所有的物理机器均一样，包括硬件和软件，如 CPU、内存、网络、存储、操作系统等。每一个物理机器上跑着各个类型的 INETIS Docker 实例，通过不同的端口暴露给负载均衡。

3.2 弹性计算

Marathon 负责所有 INETIS Docker 容器的生命周期，计算节点的就绪，退出，出错都会触发一个事件，通过监听这些事件并向中心控制器发送指令即可实现弹性计算，流程如图 2 所示，当 INETIS 负载较高需要增加计算节点时，系统自动完成以下步骤：

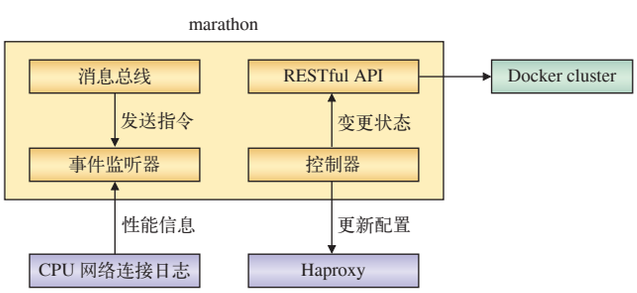

图2 弹性计算流程图

（1）检测负载，检测负载的方法有多种：a. 统计单位时间内 INETIS 应用程序的日志量；b. 获取 INETIS 实时的连接数或者是等待队列占用的大小，再将这部分数据实时更新到日志系统。通过收集到的数据，可以很容易就得到 INETIS 负载的每秒请求数（QPS）、响应时间等指标，对比设置的阈值就可以判断出是否要增加节点。

（2）设置阈值，同检测负载的方法类似，可以从不同方面设置参考值。如果根据连接的缓冲队列大小和每个请求的响应时间来设置阈值，则可设置阈值为队列长度的 30% 或超时请求超过 10%。如根据单位时间写的日志量来设置阈值，则可设置单位时间日志增量阈值为 50%。当检测到负载较大时，控制中心会调用 Marathon 的 RESTful API, 请求其增加计算节点，Marathon 会自动向 Mesos 请求资源，并根据应用程序的配置来创建 Docker 容器节点。

（3）更新负载均衡配置，当自动添加的节点就绪时，事件监听器就会收到就绪事件，控制器会通过自动化的部署方案，更新负载均衡的配置文件，通过调用 RESTful API，在集群中自动重启一个同样的计算节点。

当高峰过后，INETIS 应用服务器的 CPU 负载较低时，控制中心会调用 Marathon 的 API 请求减少节点，步骤如下：a. 从负载均衡的配置文件去掉富余节点的地址与端口；b. 调用 Marathon 的 API 回收节点。

一般情况下为了保证服务的稳定性，都会设置一个最少的计算节点数目，防止因为负载或调试的原因，导致所有计算节点都被回收。

4 结束语

本文研究了 Docker 在客票系统中的应用，通过 Docker 来部署、管理、升级、维护 INETIS 集群，并结合 Mesos 与 Marathon 构建了基于 Docker 进行弹性计算的架构。经测试，方案的实现可以极大地节省部署时间，有效利用计算资源。但同时发现 Docker 存在版本不稳定、Bug 较多等问题，在生产环境的推广使用还存在较大的风险，建议在开发和测试环境中继续进行研究试用。

参考文献：

[1] 唐海东，武延军．分布式同步系统 zookeeper 的优化 [J]. 计算机工程，2014（4）.

[2] 刘冉冉．基于任务分配的数据库集群模型研究 [D]. 武汉：华中科技大学，2007（4）.

责任编辑 方 圆