

Dependency-aware and Predictive Microservices Auto-Scaling in Cost-effective Platform-as-a-Service Clouds

彭青蓝¹

(1. 浙江大学 软件学院, 宁波 315000)

摘 要: 为了更好地把握微服务架构中各个服务的负载趋势, 以便做出及时准确的微服务伸缩策略, 提出了一种基于多元时间序列的负载预测方法。充分考虑到具有相互依赖的微服务之间负载存在关联, 采用相空间重构将多个具有依赖关系的微服务的负载历史数据映射到一个动力系统中, 近似地恢复多个微服务之间的多维非线性关系。使用 RBF 神经网络预测模型, 实现微服务负载的动态多部预测, 并根据预测负载生成微服务实例伸缩策略。

关 键 词: 微服务; 自动扩容; 多元时间序列预测; 相空间重构; RBF 神经网络

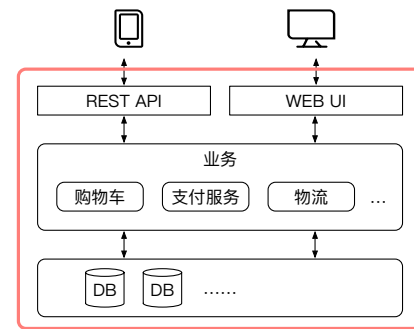
文 章 编 号: 001

1 从单体式架构到微服务架构

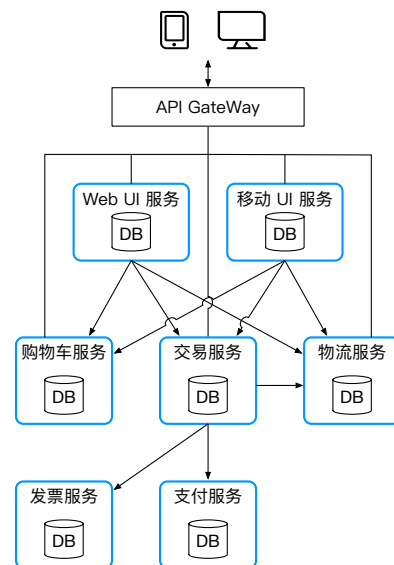
传统的单体式应用存在应用内部结构复杂、人员协调困难、可靠性低的问题。随着信息技术的飞速发展, 互联网企业又面临着新的挑战: 1) 用户规模大, 对分布式要求高; 2) 迭代周期短, 每天都要发布多个新版本; 3) 技术迭代快, 不同团队采用的技术不同。

幸运的是, 敏捷开发方法创始人之一, ThoughtWorks 首席科学家 Martin Fowler 与 2014 年提出了微服务式架构, 恰逢容器技术发展元年, 随着容器技术的飞速发展, 微服务架构逐渐落地生根。国外的包括 Netflix, Amazon 和 eBay 在内的大多数大型网站已经从单一的体系结构演变为微服务体系结构。国内的知乎、豆瓣已经完全演化为微服务架构, 阿里、京东也进行了一些微服务实践。

图 1(a) 是一个电商平台的单体式架构, 尽管采用了模块化逻辑, 但是最终还是会被打包并部署为单体式应用。



(a) 单体应用架构



(b) 微服务架构

图 1 一个电商平台的不同架构

同样的案例采用微服务式架构，如图 1(b) 所示，对于每一个应用功能点都使用微服务完成。不像传统多个服务共享一个数据库（这里指 SOA 架构），微服务架构每个服务都有自己的数据库。

2 微服务架构的自动伸缩

2.1 服务可以独立伸缩

微服务架构在带来一个好处是每个服务可以独立地进行扩展，做到有针对性的扩容。

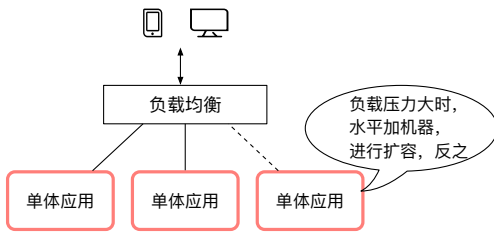


图 2 单体式架构的伸缩

例如针对双十一零时的用户高峰，图 2 是一个传统的单体式应用的扩容方案，采用了增加电商单体应用实例的方式来应对即将到来的高峰。

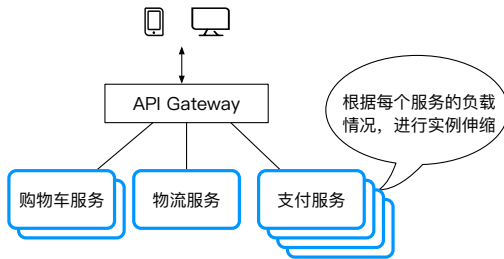


图 3 微服务架构的伸缩

图 3 是微服务架构的扩容方案。由于大家提前都将商品加入购物车中了，因此购物车服务在零点压力不太大，压力主要集中在支付服务。采用微服务架构后，就可以单独添加支付服务的实例来应对高峰，从而做到了有针对性的扩容，节省了计算资源。

2.2 服务之间存在依赖关系

采用微服务架构后，各个服务之间存在依赖关系。

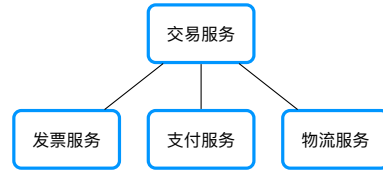


图 4 服务之间的依赖

如图 4 所示交易服务需要按照顺序调用发票服务、支付服务与物流服务，因此支付服务于这三个服务之间存在依赖关系。所以交易服务的负载变化明显会影响到发票、支付与物流服务的负载变化。反过来，当发票服务由于内部设计的缺陷或者是某种不确定因素导致负载上升时，这种变化也会影响到上层依赖它的交易服务。因此各个服务负载间的相互影响是存在的，并且呈现出一种非线性、动态多变的特点。服务实例的伸缩不能仅考虑当前服务的负载，应当将个依赖链的相互影响考虑在内。

3 基于服务依赖感知和预测的自动伸缩机制

该机制首先找出将微服务架构中的动力系统，然后使用张鹏程等人在 [1] 中提出的多元时间序列方法预测各个服务未来的负载，并根据该预测值作出相应的伸缩策略。

3.1 动力系统的划分

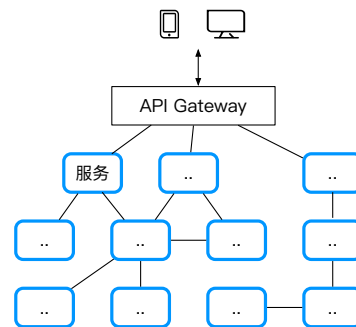


图 5 服务之间的拓扑结构

如图 5 所示，一个应用的微服务架构可以用一个图 $T = (V, E)$ 来表示，其中 $V = \{S_1, S_2, S_3, \dots, S_n\}$, $V = \{e_{i,j} \dots\}$ 。图中每个顶点表示一个独立的服务，它可以拥有多个实例，这些事例根据该服务的负载动态伸缩。边 $e_{i,j}$ 表示服务 S_i 和服务 S_j 之间存在依赖。图 T 可以从微服务架构中的注册中心获得。

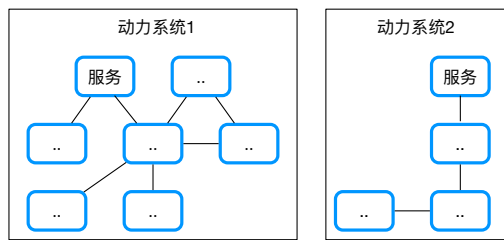


图 6 动力系统的划分

如图 6 所示，首先找出该图中的所有联通分量，每个联通分量中的服务之间存在着依赖关系。可以将每个联通分量看做一个独立的动力系统，每个动力系统元素之间以一种混沌的方式互相影响。

3.2 相空间重构

非线性时间序列可以看作由确定的非线性系统产生的，相空间 (Phase space) 是描述系统运动和演变最有力的工具之一，可以表示动力系统中所有可能的状态，其中每个状态具有对应的相位空间点，从而刻画了一个点随时间变化的情况。在得到一个动力系统中所有服务的负载时间序列后，就可以使用相空间重构来近似恢复序列构成的动力系统。

3.3 RBF 神经网络

RBF 神经网络由 Moody 和 Darken 根据人体大脑皮层的感知区域特点提出的一种前馈型神经网络，将一个动力系统的负载综合数据序列作为输入进行预测。然后分解得到每个服务的负载预测值，根据预测值再作出伸缩决策。以图 6 中的动力系统 2 为例，整个伸缩过程如图 7 所示。

4 实验设计

4.1 对比方法

实验就以下三种扩容机制进行对比：

- 1) 基于阈值触发的伸缩机制
- 2) 基于预测的伸缩机制
- 3) 基于服务依赖感知和预测的伸缩机制

4.2 数据集

目前没有找到微服务的公开数据集，可以使用微服务架构一个简单 web 应用，在 Docker 上进行部署。对于每个微服务实例单独进行压力测试，穷举其每个阶段的伸缩决策，收集数据。

4.3 实验步骤

针对一段用户请求波动较大的输入，分别使用三种策略在收集到的数据上回放伸缩过程。

4.4 预计结果

以下纯属猜想：1) 传统的基于阈值的伸缩策略在图像上表现僵硬，处于被动伸缩模式。

2) 基于预测的伸缩策略在图像上表现较为缓和，比阈值法优秀 [2]。

3) 基于服务依赖感知和预测的伸缩机制表现更好，可以更好地感知到高负载的来临。

参考文献

- [1] Web Service, Qos Forecasting, Approach Using, and Multivariate Time. 一种基于多元时间序列的 Web Service QoS 预测方法. (Lm):1-18, 2018.
- [2] Tao Li, Jingyu Wang, Wei Li, Tong Xu, and Qi Qi. Load prediction-based automatic scaling cloud computing. In *Networking and Network Applications (NaNA), 2016 International Conference on*, pages 330-335. IEEE, 2016.

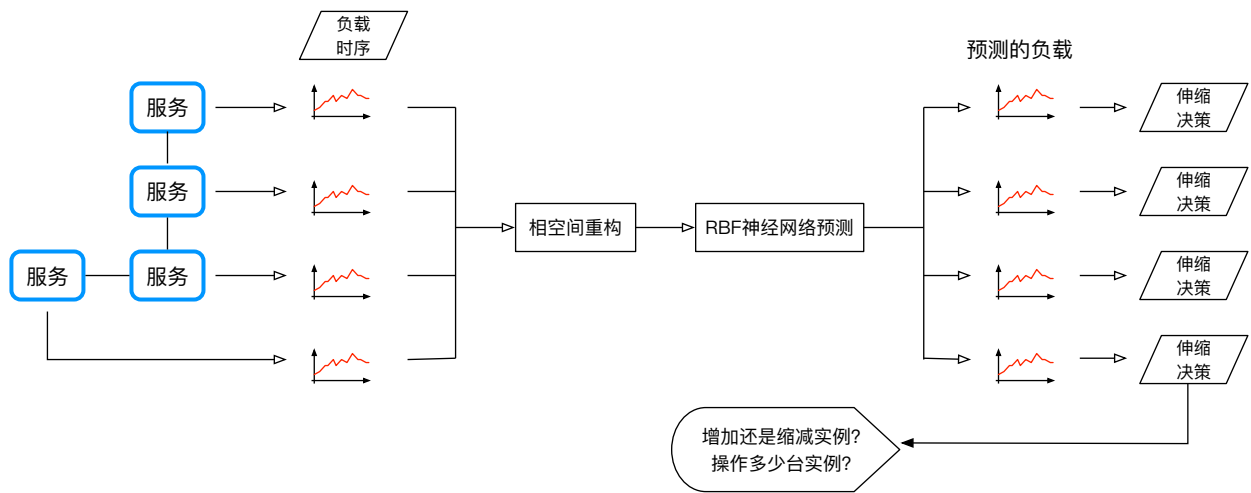


图 7 基于服务依赖感知和预测的自动伸缩机制