AHPA 阅读报告

AHPA: Adaptive Horizontal Pod Autoscaling Systems on Alibaba Cloud Container Service for Kubernetes

邹翔宇，陈锦新

2024-12-09

BackGround

HPA

Horizontal Pod Autoscaling(HPA) 是 Kubernetes 中的一种自动缩放机制。它能够自动更新工作负载资源，通过部署更多 **Pod** 来应对增加的负载，实现水平扩展，以匹配需求。HPA 主要基于以下几种指标进行工作。

- 资源指标
  - CPU 利用率、内存使用率 .....
- 自定义指标
  - 每秒请求数、队列长度 .....
- 集群系统外部指标

HPA Simple Survery

Auto-scaling Techniques

- Threshold-Based Rules
  - 根据特定性能指标（如 CPU 使用率）的预定义阈值来触发伸缩操作
- Reinforcement Learning
  - 在线学习模型并适应不断变化，但开销较大
- Queuing Theory
  - 基于排队论的模型，适用于特定场景
- Control Theory
  - 通过控制论来调整系统参数，适用于连续控制
- Time Series Analysis
  - 基于时间序列的分析，适用于周期性负载

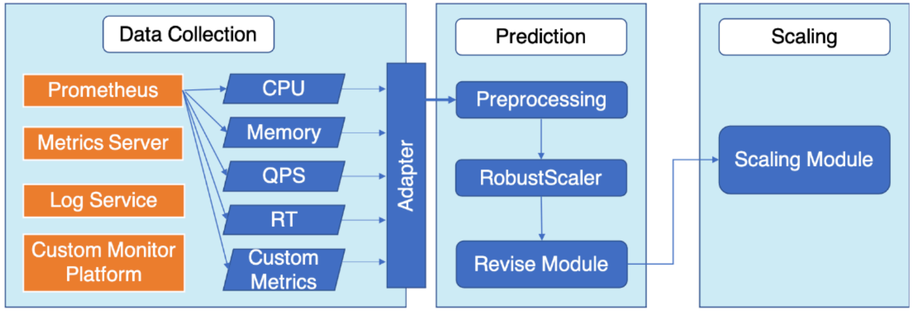
以往方法存在的问题

- 固定数量实例
  - 资源浪费：在业务需求低谷期，由于实例数量固定不变，会导致资源大量浪费。
- HPA（Horizontal Pod Autoscaling）
  - 响应滞后：在需求发生变化后才调整实例数量，对需求波动的响应存在滞后性
- CronHPA
  - 依赖人工经验
  - 缺乏动态调整能力

HPA Simple Survery
— 以往方法存在的问题

AHPA

AHPA 系统架构



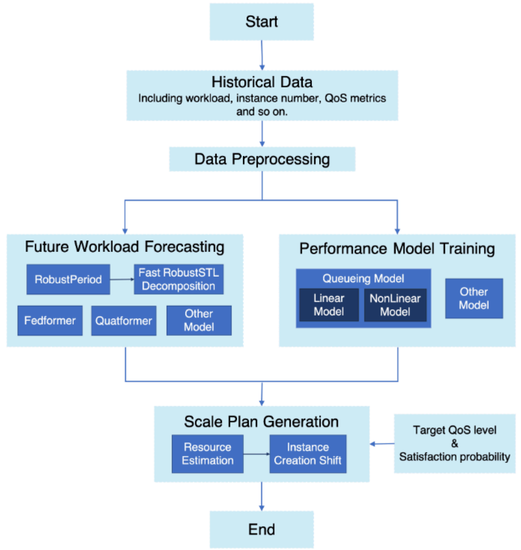
1. 数据采集，收集系统各种指标数据

2. 预测，根据历史数据预测未来负载

3. 扩展，根据预测结果调整 **Pod** 数量

AHPA 具体工作流程

- 收集历史数据
- 数据预处理
- 未来负载预测
- 性能模型训练
- 生成扩展计划



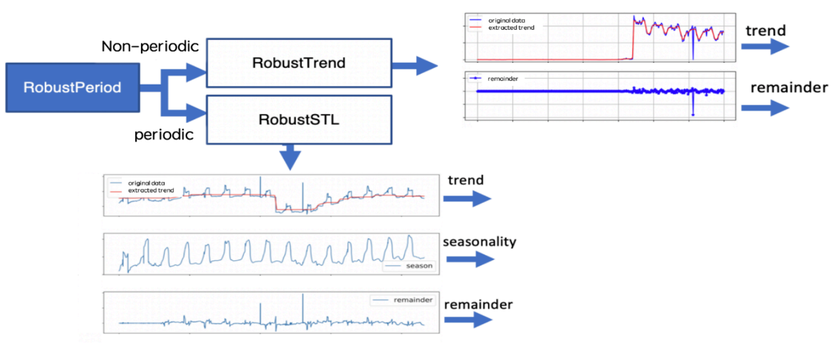
时序 STL 分解

STL(Seasonal-Trend decomposition procedure based on Loess) 分解方法会将原始时间序列分解为 trend、seasonality 和 residual terms。通过对时间序列的分解，分别对各成分进行预测，然后再将预测结果组合起来，得到对原始时间序列的预测值。

$$y_t = \text{trend} + \text{seasonality} + \text{residual}$$

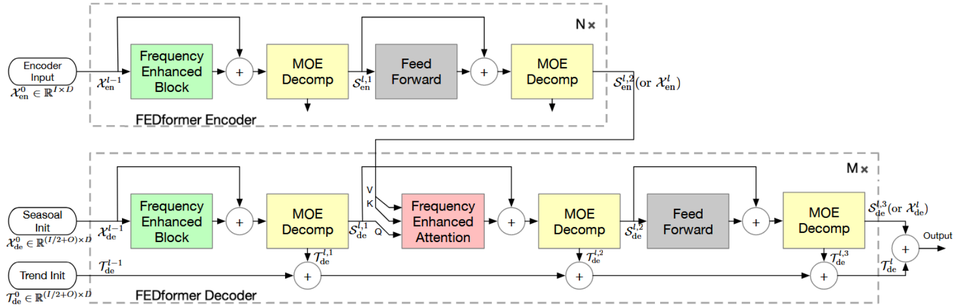
- trend，反映了数据在较长时间跨度内的总体走向
- seasonality，代表了数据中重复出现的周期性波动模式
- residual，。它包含了无法由趋势和季节性解释的随机波动和噪声。

AHPA 中的时序分解



达摩院团队基于 STL 分解的思路利用深度学习做了一系列的创新研究。分别有 RobustTrend,RobustSTL, 和 RobustPeriod, 最后以将上述算法集成的一个统一的异常检测系统。

FEDformer



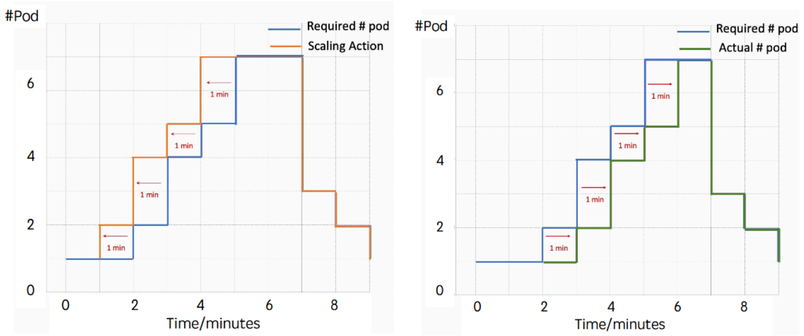
在数据较多的情况下，文中还提到了一种基于 FEDformer 的方法，通过引入 FEDformer 模型，可以更好地处理大量数据，提高模型的准确性。

AHPA Performance Model

AHPA 中主要采用运筹学中排队论的方法，包括两种不同的排队模型，*M/M/1* 队列和 *M/M/c* 队列。

- Pod 数量较多时：当系统中可调整的 Pod 数量较多时，应选择 M/M/1 队列模型，因其在这种情况下表现更优，能更简便地根据业务 QPS 等参数确定 Pod 数量。
- Pod 数量较少时：当可调整的 Pod 数量较少时，应选择 M/M/c 队列模型，它能更准确地考虑到 Pod 之间的相互影响和资源共享等情况，从而更好地满足平均响应时间要求，确定合适的 Pod 数量。

AHPA Scale Plan Generation



AHPA 采用了改进的预测偏移算法。通过提前规划来尽量抵消启动延迟带来的影响，保障业务在需要资源时能及时有足够可用的 Pod 来处理请求。

AHPA 的优势与不足

优势

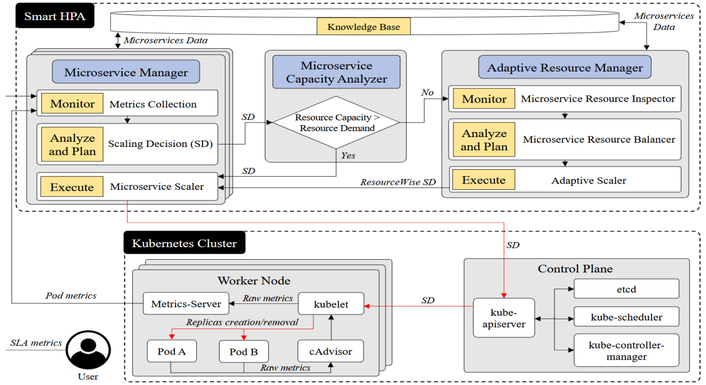
- 动态资源调整，能够提前感知业务变化趋势
- 优化资源分配
- 自动化与智能化

不足

- 依赖历史数据，对于新业务难以适应
- 面对复杂业务场景，时间序列模块和性能模型训练模块的准确性有待提高

Related Work

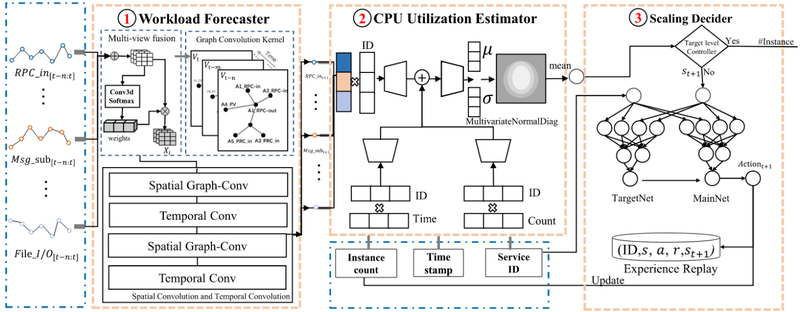
Smart HPA



Smart HPA 提出一种融合集中式和分散式架构风格的层次化架构. 特别适用于资源受限环境下的微服务资源管理。

Related Work
— Smart HPA

DeepScaling



DeepScaling 通过引入多种深度学习模型，利用时空图神经网络（STGNN），深度 Q 网络（DQN）模型等，实现了对于复杂业务场景的自适应调整。

AHPA

Thanks for your attention!