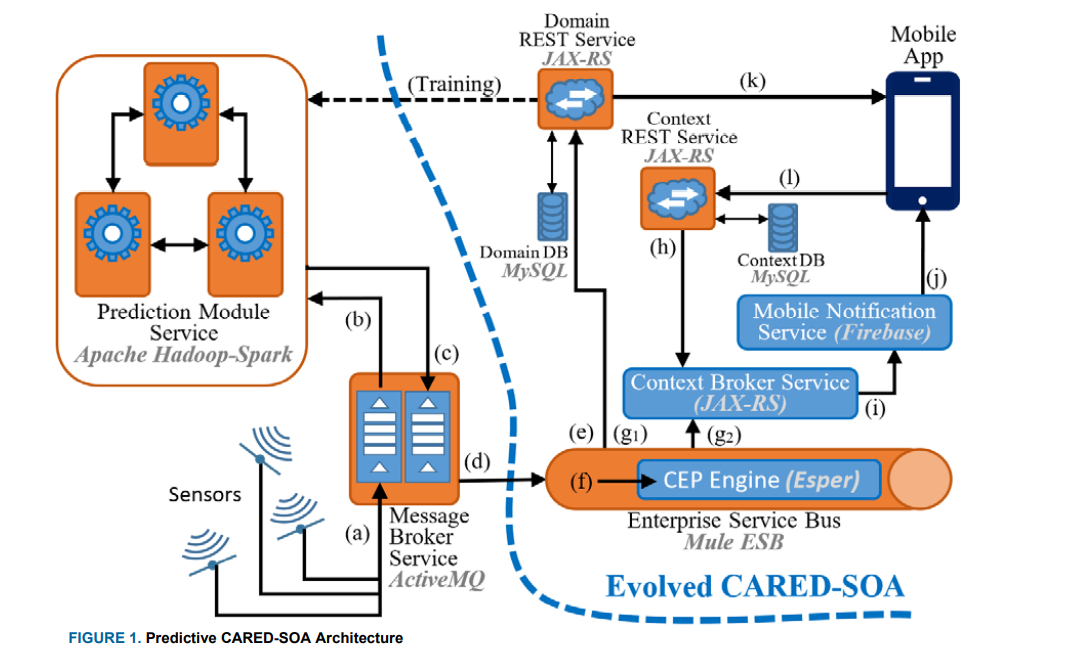
# 关键论文笔记阅读

## 1. Real-Time Context-Aware Microservice Architecture for Predictive Analytics and Smart Decision-Making

### 研究的问题

物联网环境下，大数据分析、实时的基于上下文感知预测和决策执行，与目前思考的方向比较贴近。基于微服务架构进行搭建。



### 方法

其整体的思路还是比较简单的，从传感器检测的信息到Message Broker Service对其进行分类，然后放到预测模块处理，其依赖的数据来源于后续Domain Service的反馈。

### 思考

感觉我们也可以采取类似的方式处理，如果确立感知范围是通过挖掘子图的方式确定的，将其感知范围与挖掘的子图的映射关系对立起来，通过选择监控的数据应该走向哪个服务。

## 2. To Adapt or Not to Adapt? Technical Debt and Learning Driven Self-Adaptation for Managing Runtime Performance

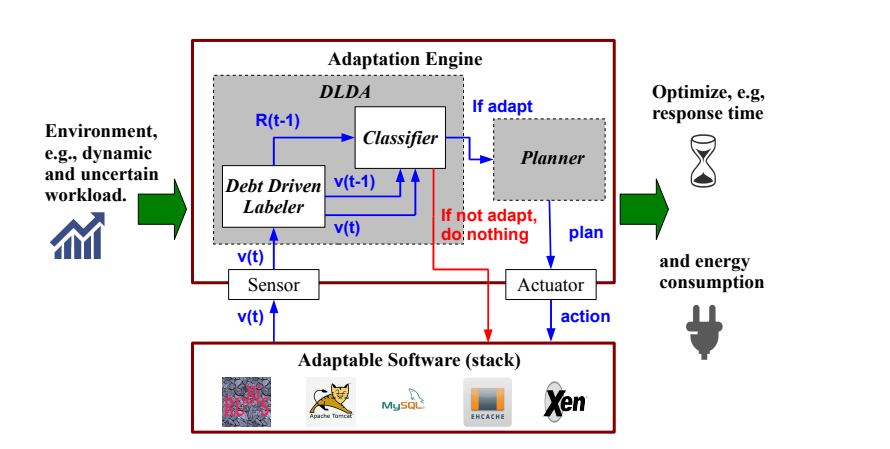
### 研究的问题

当系统需要进行自适应时，是否应该进行自适应。

### 方法

首先是对指标进行了定义：

* Principal: 一次性投资
* Interest：运行时消耗
* Revenue：收益



### 思考

能否采用上述的方式作为我们感知的决策呢，确实可以维持一个相似的东西来判断，作为是否给后面决策层处理。

## FacGraph: Frequent Anomaly Correlation Graph Mining for Root Cause Diagnose in Micro-Service Architecture

本文提出了基于异常相关图的频率模式挖掘算法，发现故障、根因。关键点在于关联图的构建、频率子图的挖掘。

从本文中可以看到关键点在于首先建立起来对于全局的监控，然后根据异常检测的结果再加上一层动态监控。全局监控初步锁定异常发生的区域。动态监控用于更细粒度的对故障的根因进行监控。以及对历史异常节点进行更具有定制化的细粒度感知监控。

## MicroRCA: Root Cause Localization of Performance Issues in Microservices

检测到异常之后、根因分析系统构构建了属性图G，表示异常的传播范围。然后提取出异常的子图SG，推断出某个最有可能的异常。收集系统数据metric。首先，根据监控数据分为两类：一类数据用于检测请求的响应事件，判断有无异常的发生。另一部分用于在异常发生之后构建关系图，即整个请求涉及到的传播区域。然后再通过子图提取进一步的缩小异常区域。

本文在属性图的构建过程中，不但把请求涉及到的服务纳入考虑，并且把与此服务相关的其他服务纳入建图，以及所有此服务的物理节host。因为服务试运行再host上的。服务的故障是与host节点有关的。构建了联通子图。

**想法**

主要的灵感在于感知不能局限于调用请求的调用链。调用链作为事件的边界是不足的。应该考虑更多的东西比如调用边界与其他服务的耦合、服务所在的节点感知等。

# 设计

整体设计如下图所示



1. 首先监测到异常指标判断异常事件产生，通过关系模块系统调用图进行挖掘并结合模块已经存在的感知范围确定事件的影响范围，产生其影响的服务调用链。当产生了影响范围后，应该与感知范围进行对比，判断目前的感知范围应该增加还是减少。
2. 当生成了影响范围，即服务调用链，其中存在服务的调用关系以及服务之间的状态信息。首先，将广义上的服务调用链传递给决策判断子模块，如《To Adapt or Not to Adapt? Technical Debt and Learning Driven Self-Adaptation for Managing Runtime Performance》论文所展示的方法，结合历史信息设置做决策所需要的一次性cost，和运行时的cost，以及决策后可能得到的收益（可以与知识图结合起来），进而判断是否应该将该事件传入决策层。同时，决策层所执行事件后产生的结果也应反馈到决策判断子模块中。
3. 同时，服务调用链也传递给预测子模块，预测子模块判断下一时刻系统事件发生的概率。我们看了比较多的自适应事件驱动方面的预测的论文，一般都是为了解决proactive执行决策。其一般是通过时间序列预测去分析。如果说此处要去分析下一时刻异常事件发生的概率，本来调用链就比较丰富，此处没有一个比较好的思路。