### AIOps for a Cloud Object Storage Service

研究的问题：为IBM的云存储开发AIOps能力。需要数据驱动且能自动化。

选取两类Operational data：

1. logs：指每一个被调用的操作，包含operation type，the bucket and object names, HTTP return code, 操作的开始与结束时间，各种延迟信息
2. connectivity logs：通过后台产生的，关于server的连接信息

### Automatic and Generic Periodicity Adaptation for KPI Anomaly Detection

KPI—Key Performance Indicator，

**Due to the effects of work days, off days, festivals, and business activities on user behavior, KPIs may exhibit different patterns within different days, which we call periodicity profifilesof KPIs.**

提出了一个Period 框架，通过日常的subsequences clustering检测periodicity profiles，提升自动检测的性能。

### A Self Healing Microservice Architecture: A Case Study in Docker Swarm Cluster

研究问题：在自适应微服务架构中，动态应对系统环境中突然的、实时发生的改变，对系统中运行的服务的正常和异常行为进行区分。

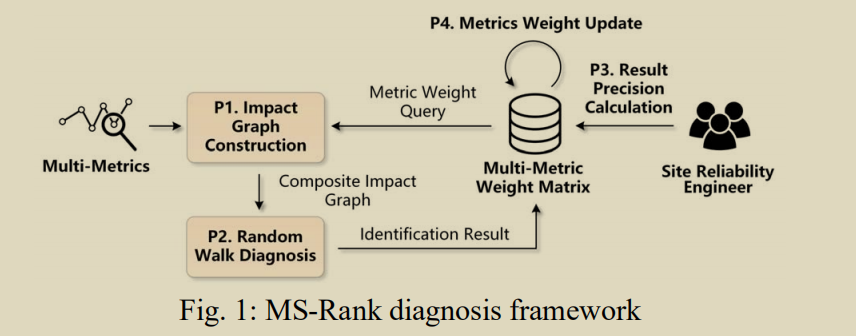
提出方法：一个self-healing微服务架构，特点——监控环境；检测异常行为；通过多维度的utility-based 模型提供合理的策略；对于系统的运行时突发改变进行计算资源的自我调整。

通过MAPE-K模型给微服务架构提供一个自适应的属性，主要贡献—自适应过程当中的效用函数。

### MS-Rank: Multi-Metric and Self-Adaptive Root Cause Diagnosis for Microservice Applications

利用MS-Rank框架，分析从系统中收集的多种metrics，分为四个阶段处理：

* Impact graph construction
* random walk diagnosis
* result precision calculation
* metrics weight update



文章提及：使用领域知识domain knowledge来提升框架的准确性。（将domain knowledge设计为五个方面：link service APIs；link known calling relationships；orient known direction；删除没有依赖的边；删除正常的服务）

* 在数据集很稀疏的时候，通过引入domain knowledge到impact graph 的构建中能取得更好的效果。

### Graph-Based Trace Analysis for Microservice Architecture Understanding and Problem Diagnosis

大量运行时产生的traces难以实时捕捉，难以进行故障的分析，因此提出一个基于图的微服务trace analysis方法—GTMA(Graph-based Microservice Trace Analysis System)，理解整个架构与诊断问题。

GTMA分为三个模块：

* processing：处理分布式追踪系统的span log等原始数据，将其整理成trace,path和business flow三个层次
* storage：结合图数据库+实时数据库+设计的存储结构
* access：三个层次的调用接口

### Optimal Resource Utilization Strategy for Cloud using MAPE-K model and Microservices on Kubernetes Federation

基础背景：IBM提出的MAPE-K循环，Netflix提出的Microservice。在云上，对于资源的动态自动扩展能力几乎决定了用户的QoS，传统的HPA无法应对突发的大量资源分配请求，无法最优化对于资源的利用，同时致使QoS下降。

研究的问题：是否能替代传统的HPA(Horizonal Pod Auto scaling)，转而使用MAPE-K循环和微服务来寻找更优化的资源分配策略？

提出的方法：MAPE-K 算法，在Knowledge base存在时，分为四个阶段MAPE，对文章指定的指标进行检测和计算。Knoelwdge base用于整合所有的阶段。

在不同的senarios下，进行测试和评估。

不足： 需要考虑更多的复杂集群环境和高强度的工作负载场景，算法需要更加精简。

K–Knowledge，表现为一个数据库，收集与存储在特定输入的场景下所获得的输出结果。

Knowledge : This can be regarded as the central element of the MAPE-K feedback loop  
as it works in integration with all the respective phases to deliver reliable results.

* Konwledge base作为一个数据库，需要基于不同的场景输出来收集数据，不断进行迭代，缺乏主动性，只能用来被动地提供消息。难以考虑所有的场景。