# Graph-based root cause analysis for service-oriented and microservice architectures

作者：Álvaro Brandóna,∗ , Marc Solé

### 大意

对于困境的描述：

Service-oriented architectures and [microservices](https://www.sciencedirect.com/topics/computer-science/microservices) define two ways of designing software with the aim of dividing an application into loosely-coupled services that communicate among each other. This translates into rapid development, where each service is developed and deployed by small teams, enabling continuous shipping of new features and fast-evolving applications. However, the underlying complexity of this type of architecture can hinder observability and maintenance by the user. In particular, identifying the root cause of an anomaly detected in the application can be a difficult and time-consuming task, considering the numerous services and connections to be examined.

实验比较的结果是：基于图的比较比基于机器学习的方法更为有效。

基于对架构的图表示，建立了一个根因分析框架。该框架收集RCA(root cause analysis)过程真正重要的信息，如架构的拓扑图或者系统不同element之间的connections。

建立框架的目的是：1. 让用户更好的理解当前系统的行为。 2. 将现在系统的异常情况(anomalous situation)和以前被诊断的situation进行匹配.

### 相关工作

#### RCA

相关综述：

* Wong, W.E., Gao, R., Li, Y., Abreu, R., Wotawa, F., 2016. A survey on software fault localization. IEEE Trans. Softw. Eng. PP (99), 1. doi:10.1109/TSE.2016.2521368.
* Solé, M., Muntés-Mulero, V., Rana, A. I., Estrada, G., 2017. Survey on models and  
  techniques for root-cause analysis. arXiv:1701.08546.

基于模型的诊断框架：

* Joshi, K.R., Hiltunen, M.A., Sanders, W.H., Schlichting, R.D., 2005. Automatic model–  
  driven recovery in distributed systems. In: Reliable Distributed Systems, 2005.  
  SRDS 2005. 24th IEEE Symposium on. IEEE, pp. 25–36.
* 应用于web服务当中：Mayer, W., Friedrich, G., Stumptner, M., 2010. Diagnosis of service failures by trace analysis with partial knowledge. In: International Conference on Service-Oriented Computing. Springer, pp. 334–349.

基于分类的方法：  
（使用ML的方法进行诊断）

* Demetgul, M., 2013. Fault diagnosis on production systems with support vector  
  machine and decision trees algorithms. Int. J. Adv. Manuf. Technol. 67 (9-12),  
  2183–2194.

使用图的方法：

* 基于Probabilistic Graphical Models (PGMs)：Schoenfisch, J., Meilicke, C., von Stülpnagel, J., Ortmann, J., Stuckenschmidt, H.,2017. Root cause analysis in it infrastructures using ontologies and abduction in Markov logic networks. Inf. Syst..
* Akoglu, L., Chau, D.H., Vreeken, J., Tatti, N., Tong, H., Faloutsos, C., 2013. Mining connection pathways for marked nodes in large graphs. In: Proceedings of the 2013  
  SIAM International Conference on Data Mining. SIAM, pp. 37–45.

# Survey on Models and Techniques for Root-Cause Analysis

### 基本信息

##### 作者：

Marc Sol ́e…

##### 研究问题和特点：

* 关注IT系统RCA中用到的模型（model）和模型使用的生成（generation）和推断（inference）算法
* 在性能和可扩展性上对不同需求的的技术进行分析。
* 引导，如何针对于特定的系统选择合适的RCA策略。

##### 文章结构

* Introduction
* 概念和术语介绍
* 主要的models for RCA
* 用于模型的inference algorithm
* 总结

### 重点内容

##### 核心概念

causality（因果关系 ）和 explanation（ 解释，说明，理由，原因，）通常被认为是RCA的输出（output）

##### 术语Terminology

Event: an exceptional condition

Fault/problems/root causes: 是指源头的Events，只有它造成别的event，而非被别event导致。分为三类—permanent/ intermittent(间歇的) /transient

Error：由一个或者多个Fault造成

Failure：从系统外可以观测的Error

Symptom：一个Error的外部表征（manifestation)

Root Cause Analysis：也可以称为fault localization、fault isolation、alarm/event correlation。含义是：the process of inferring the set of faults that generates the given set of synptoms。

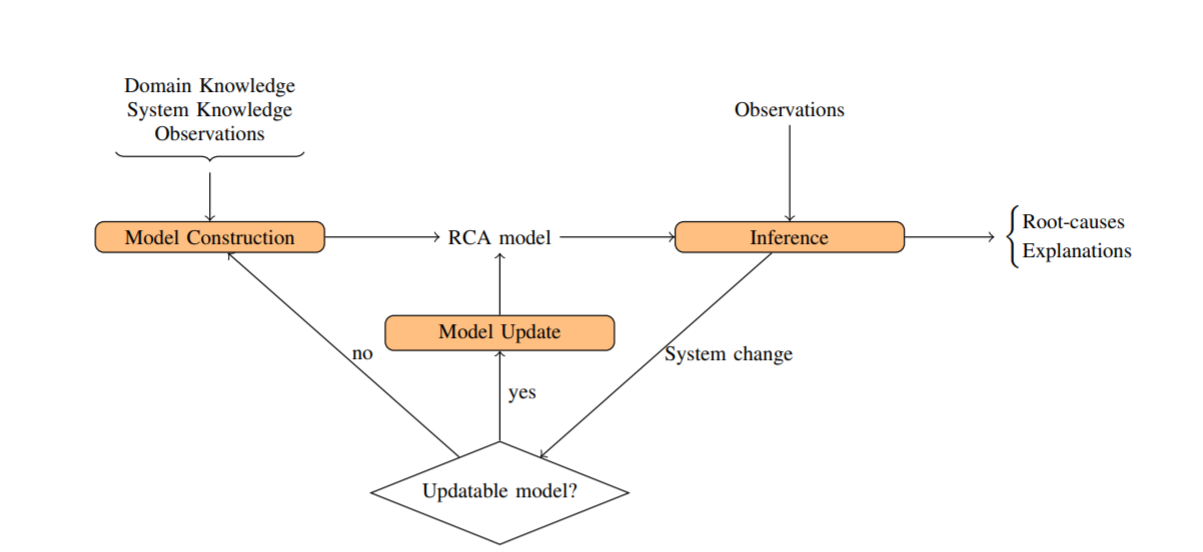
##### 系统对于RCA的影响因素

Analysis intent：意图，是需要获得观测到的symptoms的root cause，还是需要解释root cause如何和symptom产生联系。

Analysis time：非功能性需求，即算法能够用来进行推断的最大时间，分为两类场景---real-time diagnosis（响应时间很关键，解决方法是预先计算部分或者全部inference process），post-morten diagnosis（时间限制并不重要）

Complexity：复杂度，包含System Size、Data Size、Inference Length(从symptom到fault被遍历到的最大组件数)、Effect propagation time（一个组件状态的改变需要一定时间才能影响到其他临近组件，有时被忽略或认定为即时的，有时它决定了相关探测的window,决定了Observation Data Size）、Evolution rate（被诊断的系统的变化速度和程度）、Domain knowledge required、System knowledge required（诊断系统对被诊断系统所需要的访问级别）

##### RCA workflow



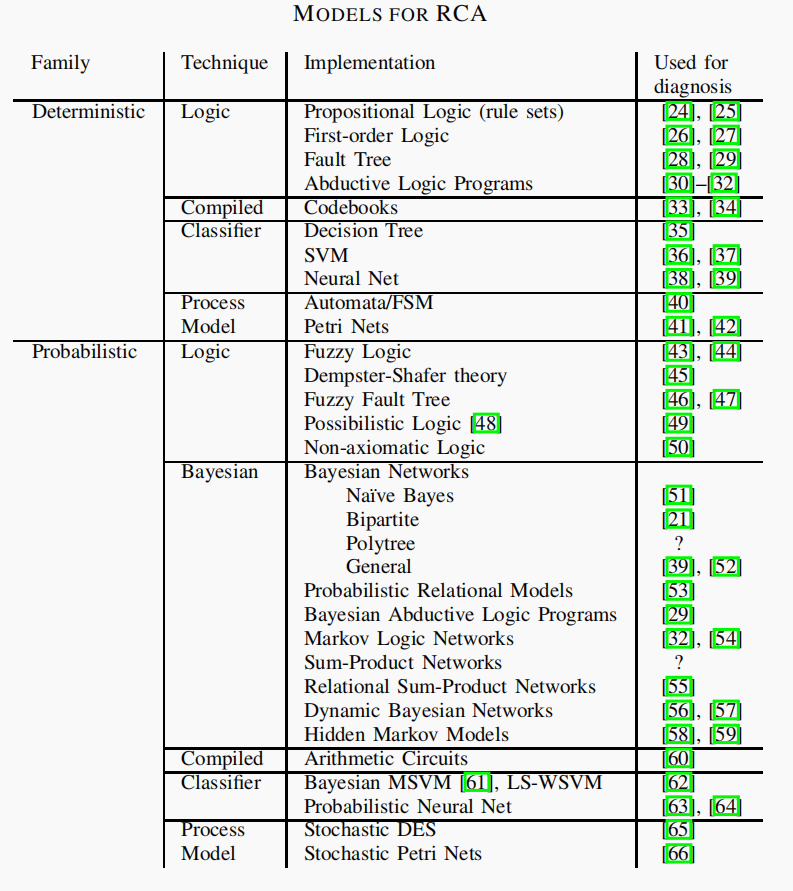
##### RCA模型的生成

inference process当中关键的一个特点是：Performance。

许多的推断模型都来自AI，大致分为两大类：

1. Deterministic确定式模型
2. Probabilistic概率式模型

图表简要展示不同类型的RCA模型：



为什么一些ML中的分类器没有占领RCA领域？

1. 它们通常只返回一个预测的root cause，无法获得解释
2. 很难和domain knowledge结合起来，不能产生逻辑规则
3. 它们往往被简化成了single-label classifier，当需要进行多种故障的检测时，需要和多种classifier协同工作

什么因素对诊断的性能影响最大？

* Size：可以理解为是基本的分析元素的数量，通常是组件component的数目，但是也可以有多个Size属性。
* Inference structure：它定义了不同的elemental analysis element是如何相互关联到一起的。通常利用**Inference Length**指标来进行判别。

##### 如何生成RCA模型？

模型的Size（用来构建系统的基本分析元素的多少，主要取决于系统建模时抽象的层次）和Inference structure（定义不同的基本分析元素是如何相互联系的）对整个模型很重要。

主要有三类方式：

1. Manual generation：专家来提供模型；模型会比较准确，但是知识的elicitation（引用）会很慢且对人来说比较复杂。
2. Assisted generation：根据相当数量的以子模型库的形式出现的domain knowledge 和向下的system information ，产生最后的诊断模型。
3. Automated generation

##### learning model for RCA

当没有domian knowledge和专家手工等一些协助方法可用时，只能采用learning algorithm，学习raw data of the system。

learning algorithm往往能够学习structure 和 internal parameter，

##### RCA模型的推断—Inference for RCA model

Inference指的是 从由故障产生出来的一系列给定症状当中获取对应的故障。

**对分类器的模型而言，推断的过程只涉及了使用模型对symptom进行分类。**

对于其他模型而已，存在不同的算法来进行实现。

分为两种算法：

* 提供具有不同性能的算法实现，但具有相同的结果
* 对于如何给出好的解释具有不同看法

贝叶斯网络 --- 一种概率图模型。  
特性：

1. 贝叶斯网络本身是一种不定性因果关联模型，它本身是将多元知识图解可视化的一种**概率知识表达与推理模型**。
2. 贝叶斯网络具有强大的不确定性问题处理能力  
   基于贝叶斯网络推理研究有两类近似推理算法：**基于仿真方法和基于搜索方法**，一般在故障搜索时采用搜索推理算法。  
   朴素贝叶斯(理想状况下才能发挥效果)、 二部贝叶斯网络（一部分节点表示故障原因，一部分表示错误症状，但是在遇到复杂的模型是难以进行转换）、

在性能和复杂度的角度来看，MPE（最大可能解释问题，在贝叶斯网络进行推理时，在变量状态满足条件时，关心概率最大的那一个，是MAP的特例）、MAP（最大后验假设问题，在所有可能的假设当中，找出后验概率最大的那一个）

如果开销太大，则计算近似值。

**近似算法**：其中一类是有关MAP的贝叶斯网络。

决策树模型：能够快速的训练出来，一种是使用打好标签的数据来将其训练成为分类器；另一种则只是将其作为有无错误出现的一个判别方式。

分类器，机器学习算法有很好的推断时复杂性表现。SVM代价是线性的。

Sum-Product和Arithmetic Circuits

逻辑相关的方法往往和表达性联系在一起

算法并行执行。

##### 总结

大范围的技术都在进行着tractability和expressiveness的权衡。

# Microservices The Journey So Far and Challenges Ahead

未来挑战：

* 服务的模块化与重构 Service Modularization and Refactoring
* 服务粒度 Service Granularity
* 前端集成 Front-end Integration
* Resource Monitoring and Management
* Failure, Recovery, Self-Repair
* Organizational Culture and Coordination
* Addressing the Chanllenges

软件架构正在经历从单体架构到微服务的转变，以实现软件开发的弹性、敏捷性和可扩展性。然而，由于技术异构、大量微服务以及软件功能和基础设施的频繁更新，微服务很难诊断性能问题。

通过将应用程序性能症状与相应的系统资源利用率相关联，实时推断根本原因，无需任何应用程序检测。

根本原因定位基于对跨服务和机器的异常传播进行建模的属性图。