



基于大规模运维数据的云原生软件多维度分析

彭鑫 复旦大学计算机科学技术学院

pengxin@fudan.edu.cn

<http://www.se.fudan.edu.cn>

<https://cspengxin.github.io>

基于大规模运维数据的云原生软件多维度分析

目录 CONTENTS

- ① 云原生软件生态系统
- ② 软件分析的挑战与机遇
- ③ 基于运维数据的多维度分析
- ④ 总结与展望

基于大规模运维数据的云原生软件多维度分析

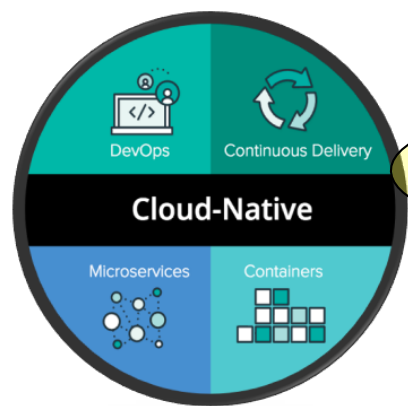
云原生软件生态系统

大规模云原生软件正逐步演化为复杂
软件生态系统

01

基于大规模运维数据的云原生软件多维度分析

云原生（Cloud Native）软件



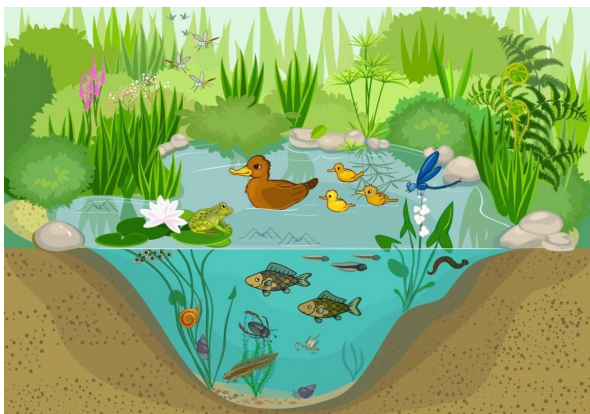
据IDC预测，到2022年，90%的应用程序将采用微服务架构；35%的产品应用将是云原生的。

云原生：应用的设计、实现及部署运行方式充分适应云计算平台的特点（如分布式、弹性伸缩）

- ✓ 微服务：物理隔离，轻量级分布式通信，独立部署、独立更新、独立伸缩
- ✓ DevOps：开发运维一体化，持续集成与自动化发布
- ✓ 持续交付：频繁发布、快速交付、快速反馈、快速试错
- ✓ 容器化：服务无差别封装在容器中，运维无需关心服务技术栈

基于大规模运维数据的云原生软件多维度分析

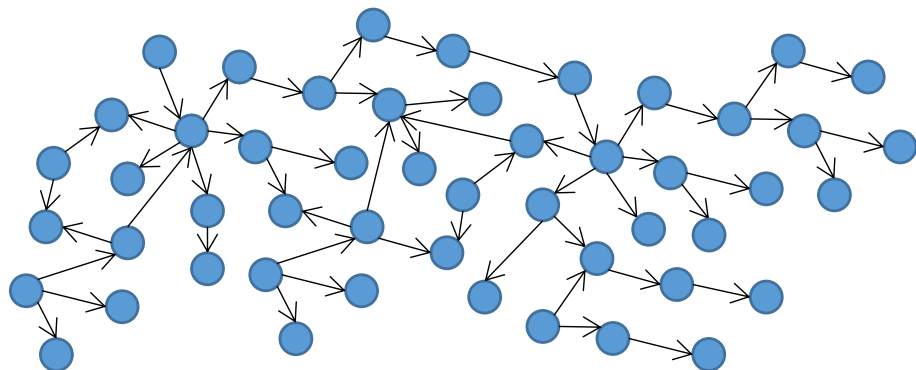
生态系统



自然生态系统



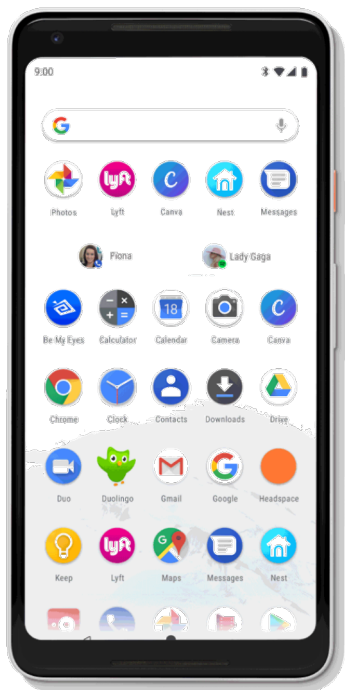
城市生态系统



规模大、组成成分多、关系复杂
组成成分之间相互依存又相对独立
动态变化、自主生长、持续演化

基于大规模运维数据的云原生软件多维度分析

软件生态系统



安卓生态系统



苹果生态系统



支持定制化开发的
软件产品线

应用共享并依赖于平台及开发框架

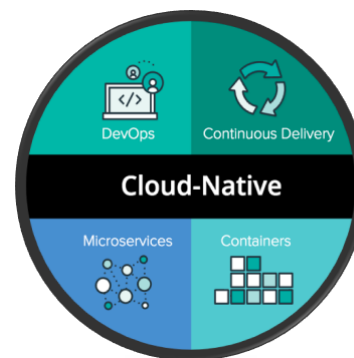
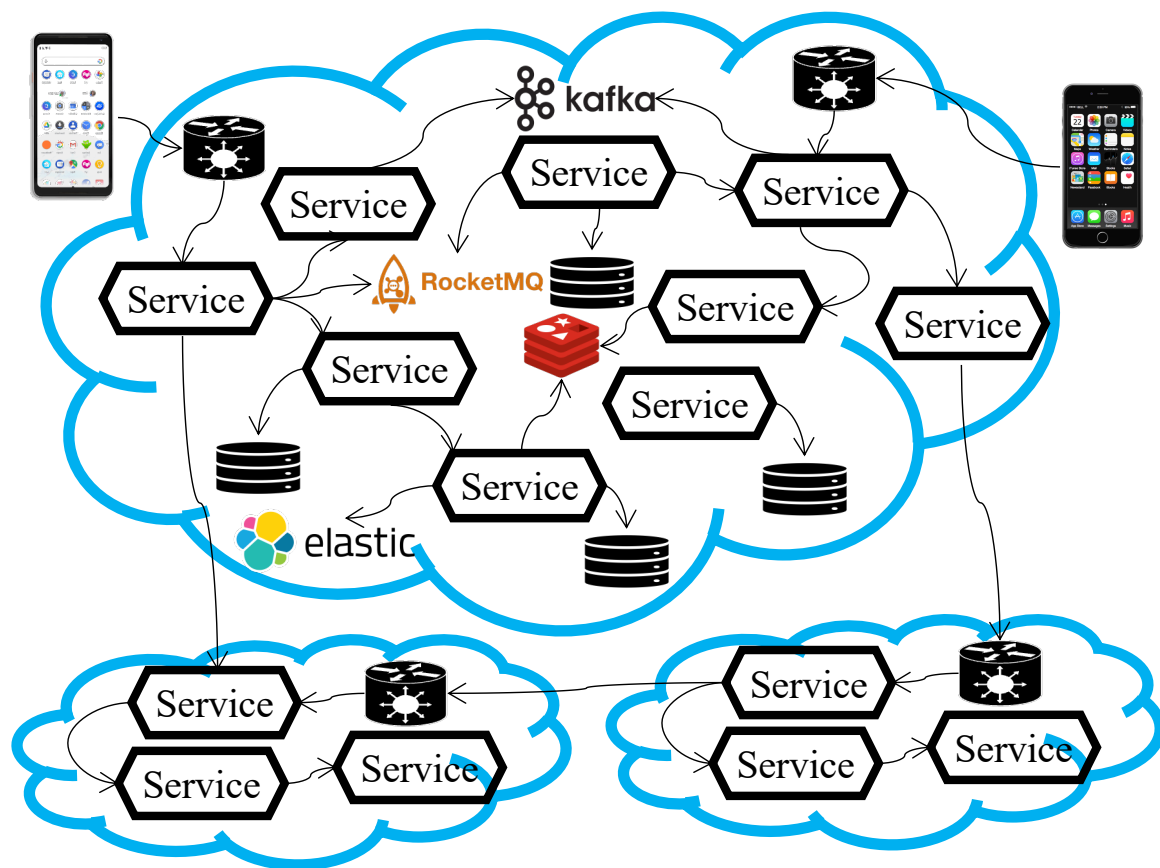
应用间可互联互通

应用持续增长并自主演化

特点：技术依赖、离线演化

基于大规模运维数据的云原生软件多维度分析

云原生软件生态系统



服务独立开发、部署、演化

服务持续交付与更新

新的业务和特性不断生长

特点：持续运转、在线演化

基于大规模运维数据的云原生软件多维度分析

软件分析的挑战与机遇

传统的软件分析技术已经无法支撑云原生软件分析的需要

02

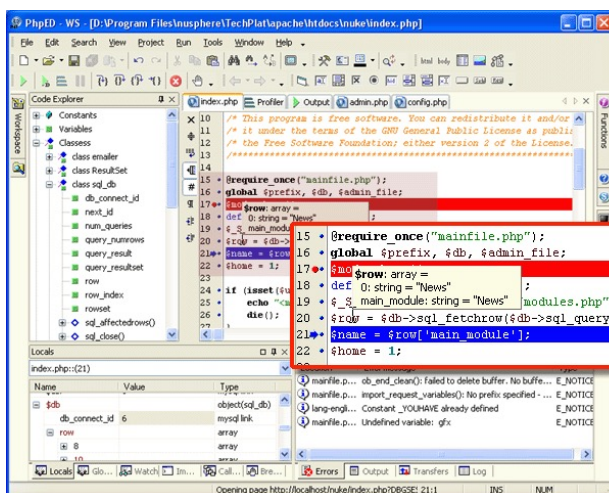
基于大规模运维数据的云原生软件多维度分析

软件分析的目的

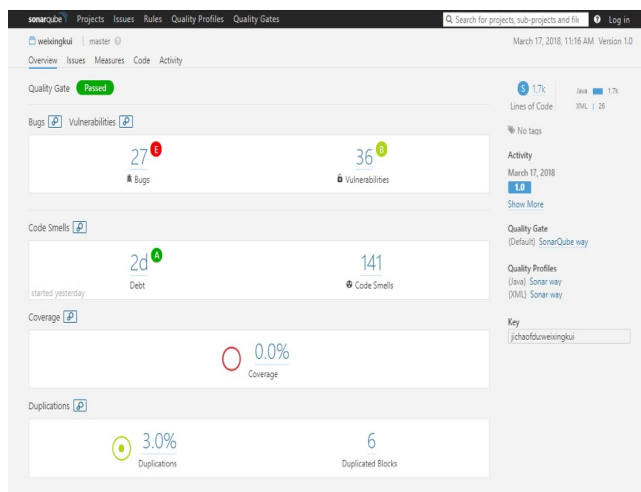


基于大规模运维数据的云原生软件多维度分析

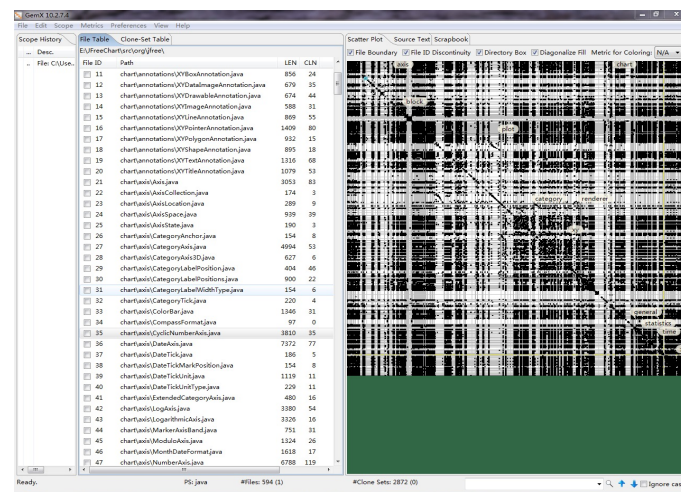
传统软件分析：代码调试与质量分析



代码调试



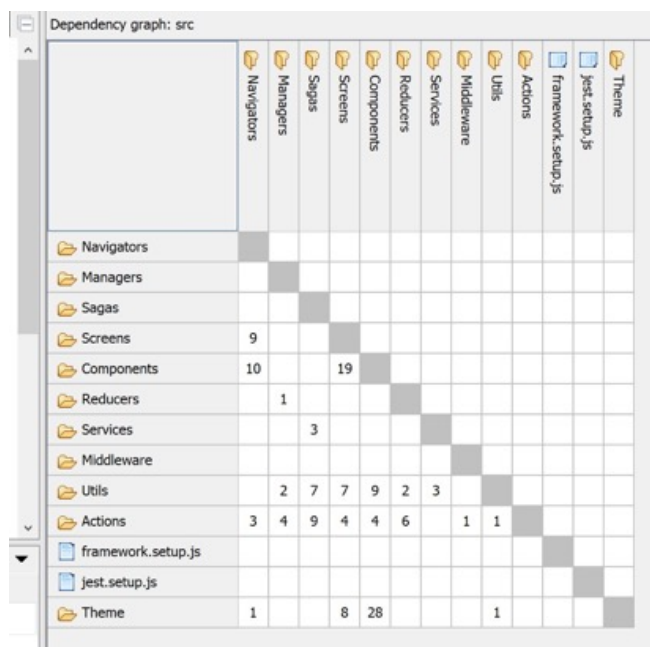
代码质量静态扫描



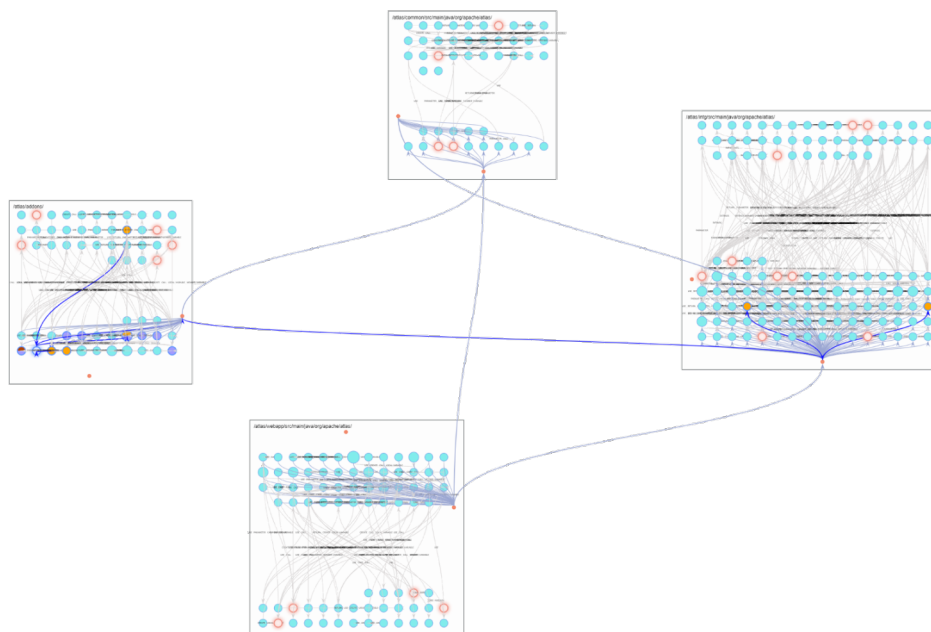
代码克隆检测

基于大规模运维数据的云原生软件多维度分析

传统软件分析：设计逆向分析与评估



依赖关系分析



设计异味检测

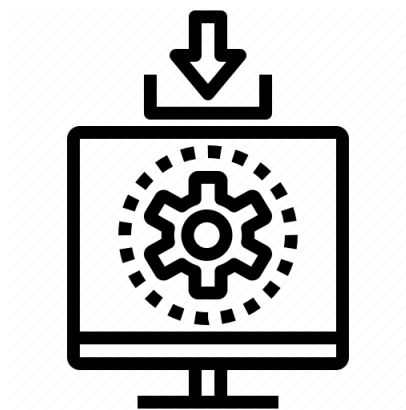


基于大规模运维数据的云原生软件多维度分析

传统软件分析技术



代码静态分析



代码动态分析



代码演化历史分析

基于大规模运维数据的云原生软件多维度分析

云原生软件分析的挑战

服务独立开发和部署

服务成为基本开发和运行单元，代码及演化历史分析只能应用于服务内部分析

运行环境复杂多变

运行环境配置及相互影响复杂多变，难以掌控和模拟

自主生长、独立演化

大量内部及第三方团队参与，演化过程难以通过自顶向下的方式进行集中控制

系统行为高度动态

服务和系统行为受到部署和运维配置及运行环境影响，具有高度的动态性

开发

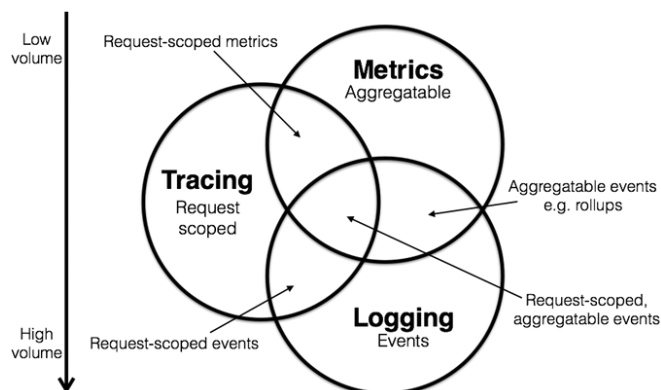


运维



基于大规模运维数据的云原生软件多维度分析

云原生软件分析的机遇

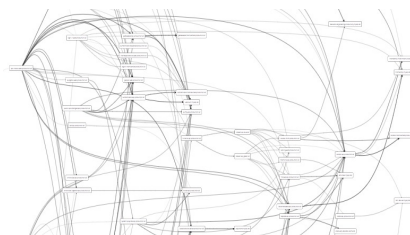


可观察性

Peter Bourgon. Metrics, tracing, and logging.
<https://peter.bourgon.org/blog/2017/02/21/metrics-tracing-and-logging.html>

```
1 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: Starting DemoApplication on my-m1 with PID 10422
2 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: No active profile set, falling back to default
3 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: Toms: Initializing with profile: demo
4 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: Starting service engine: Apache
5 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: Starting service engine: Apache
6 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: Starting service engine: Apache
7 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: Starting service engine: Apache
8 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: Starting service engine: Apache
9 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: Starting service engine: Apache
10 2020-02-14 22:04:22.720 INFO [main] com.example.demo.DemoApplication: Starting service engine: Apache
```

运行日志



调用链路



指标度量

显式的软件体系结构

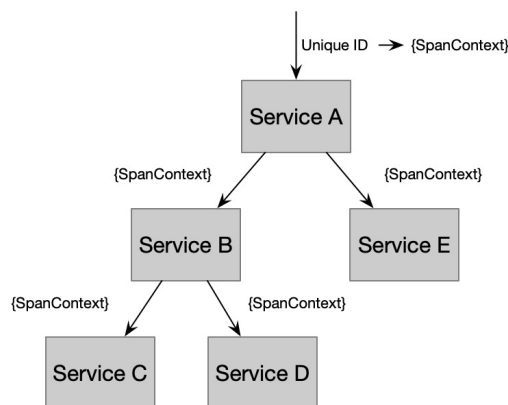
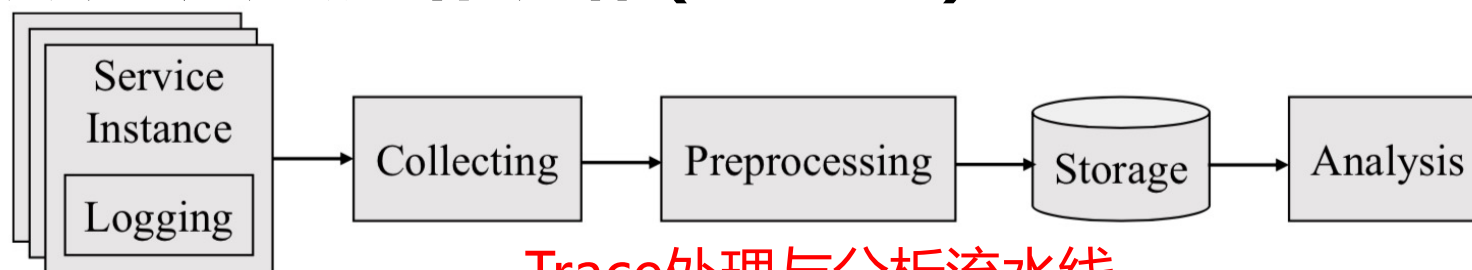
外化的交互行为

系统持续在线运行

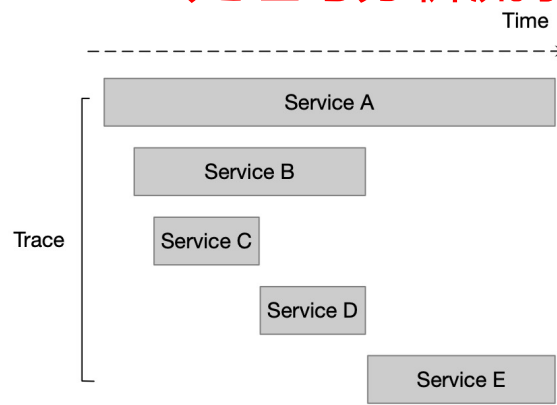
完善的基础设施支持

基于大规模运维数据的云原生软件多维度分析

微服务调用链路数据 (Trace)



Trace拓扑结构



Trace的时间线表示

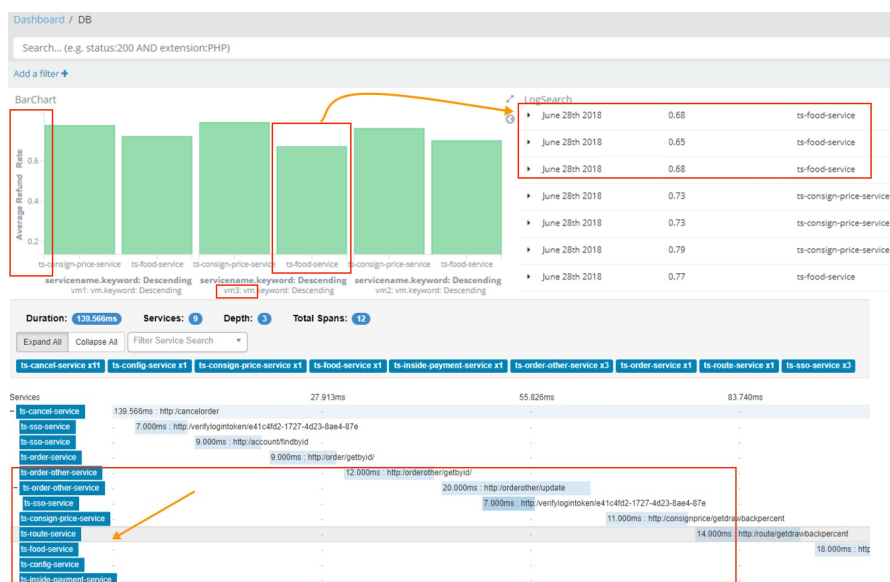
```
1 {
2   "traceId": "5982fe77008310cc80fda5e10147517",
3   "spanId": "be2d01e33cc78d97",
4   "parentSpanId": "ebf33e1a81dc6f71",
5   "timestamp": 1458702548786000,
6   "duration": 13000,
7   "localEndpoint": {
8     "serviceName": "client",
9     "ipv4": "192.168.1.2",
10    "port": 9411,
11    "remoteEndpoint": {
12      "serviceName": "server",
13      "ipv4": "127.0.0.1",
14      "port": 3306,
15      "annotations": [{
16        "timestamp": 1458702548786000,
17        "value": "cs"
18      }],
19      "timestamp": 1458702548799000,
20      "value": "cr"
21    },
22    "timestamp": 1458702548788000,
23    "value": "sr"
24  },
25  "timestamp": 1458702548796000,
26  "value": "ss"
27 }
```

Span Log

Bowen Li, Xin Peng, Qilin Xiang, Hanzhang Wang, Tao Xie, Jun Sun, Xuanzhe Liu: Enjoy your observability: an industrial survey of microservice tracing and analysis. Empir. Softw. Eng. 27(1): 25 (2022)

基于大规模运维数据的云原生软件多维度分析

微服务故障分析与调试企业实践

[illegible]

原始日志分析

- 典型工具：命令行工具
- 使用占比：23%

日志统计可视化分析

- 典型工具：ELK，即（Logstash + ElasticSearch + Kibana）
- 使用占比：23%

执行轨迹可视化分析

- 典型工具：Zipkin
- 使用占比：31%

Xiang Zhou, Xin Peng, Tao Xie, Jun Sun, Chao Ji, Wenhai Li, Dan Ding: Fault Analysis and Debugging of Microservice Systems: Industrial Survey, Benchmark System, and Empirical Study. *IEEE Trans. Software Eng.* 47(2): 243-260 (2021)

基于大规模运维数据的云原生软件多维度分析

基于运维数据的多维度分析

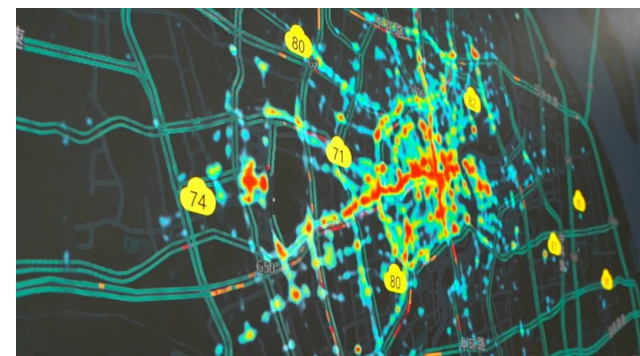
基于运维大数据分析支持故障定位、
设计评估及业务理解等多种分析目的

03

基于大规模运维数据的云原生软件多维度分析

类比：城市运行数字体征系统

对城市**生命体**进行“24小时×365天”的“**全时智慧体检**”。相较于传统的静态城市运行体征，即将上线的上海城市运行数字体征系统是“**活**”的，以“**实时、鲜活**”的**多维、多源、多态**数据为基础...通过**数字体征**来呈现城市正在发生什么、需要关注和处置什么...更好地研判城市运行的**趋势和规律**，及时发现潜在的**风险**，助推城市管理，将被动处置变为“**治未病**”...

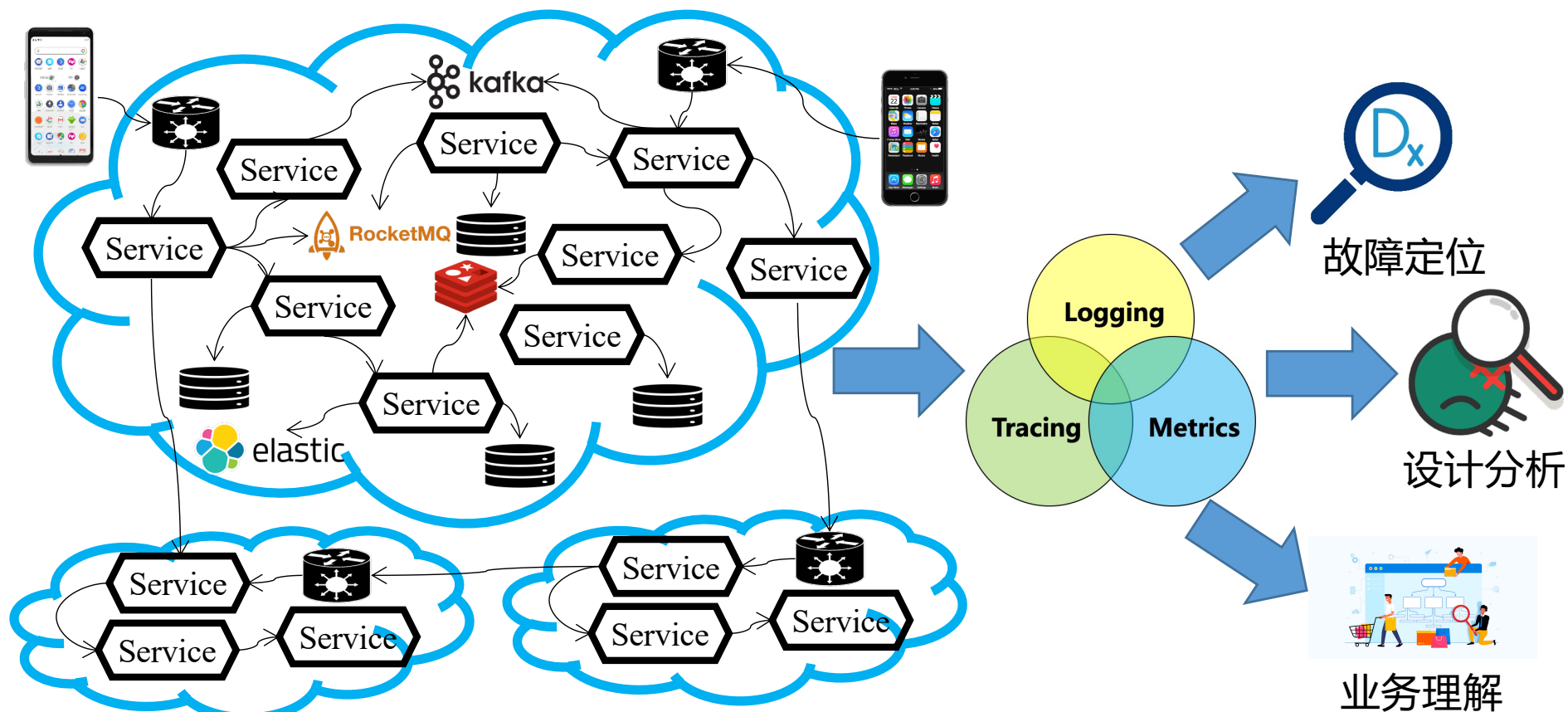


解放日报：上海城市运行数字体征系统将上线

http://www.xinhuanet.com/info/2021-05/12/c_139940069.htm

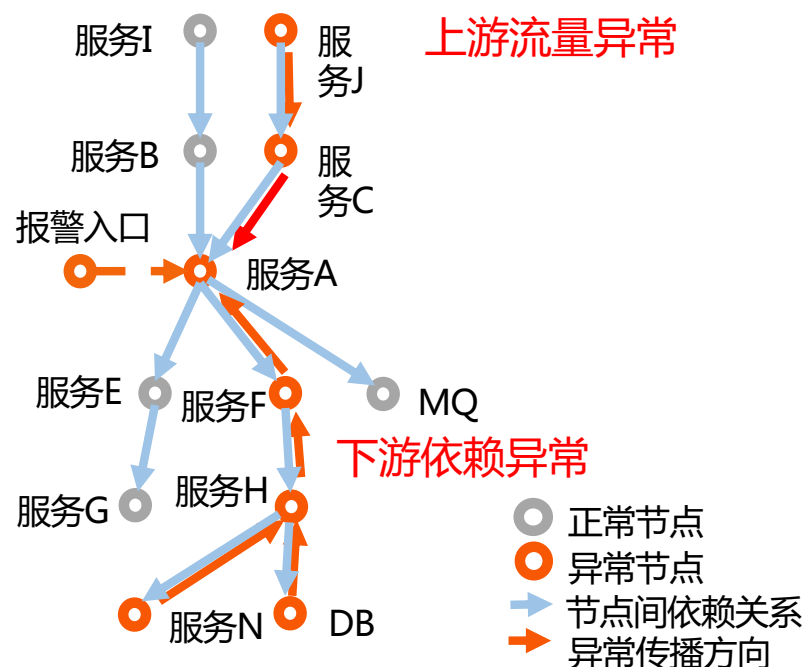
基于大规模运维数据的云原生软件多维度分析

基于运维数据的云原生软件多维度分析



基于大规模运维数据的云原生软件多维度分析

基于运行时服务依赖图的故障定位



三类异常问题：可用性（错误率）、性能（响应时间）、流量异常

基本思想：从入口点开始沿着服务依赖图游走进行故障根因定位

关键问题：游走方式、剪枝策略、单点异常判断、结果排序

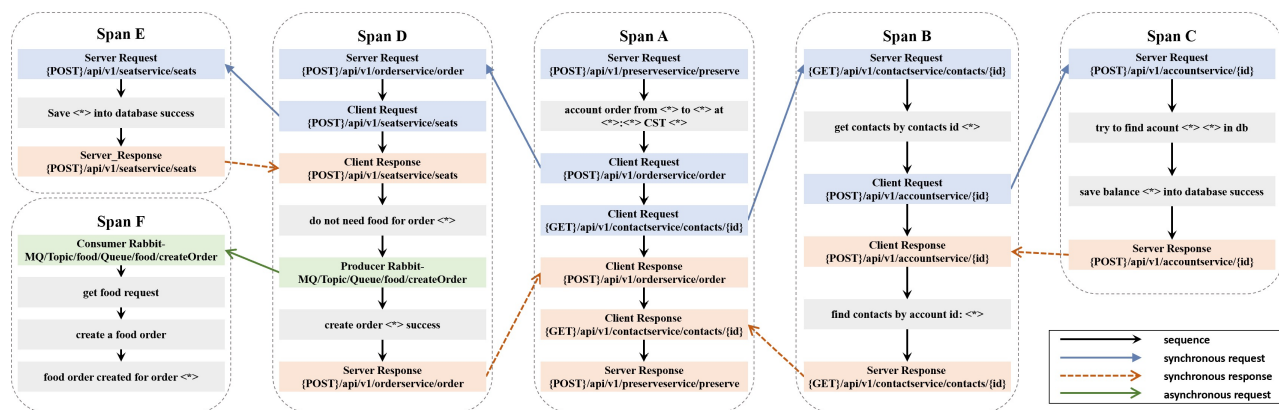
基于运行时服务依赖图分析不同类型异常的传播路径，实现根因定位

Dewei Liu, Chuan He, Xin Peng, Fan Lin, Chenxi Zhang, Shengfang Gong, Ziang Li, Jiayu Ou, Zheshun Wu: MicroHECL: High-Efficient Root Cause Localization in Large-Scale Microservice Systems. ICSE 2021 (SEIP Track).

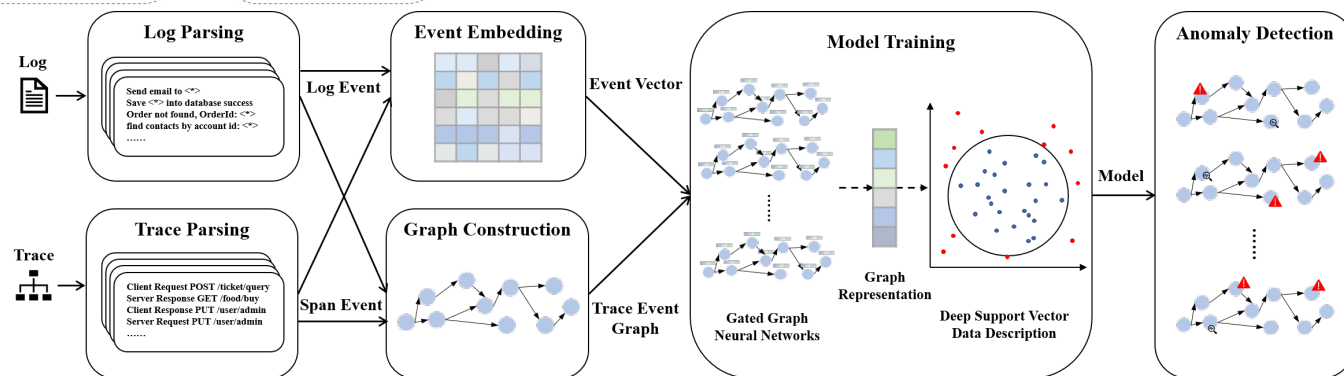
基于大规模运维数据的云原生软件多维度分析

基于日志和调用链融合分析的异常检测

在跨服务的日志消息链路关联的基础上，利用基于图的表示实现链路拓扑与日志事件的融合表示



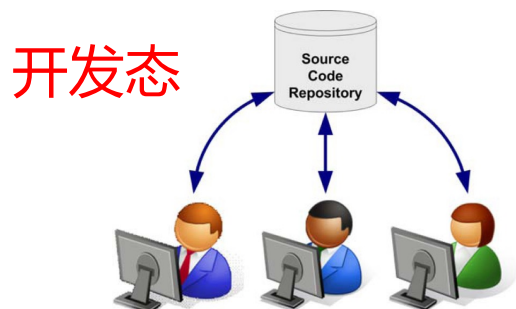
利用图神经网络训练异常检测模型，识别系统异常行为



Chenxi Zhang, Xin Peng, Chaofeng Sha, Ke Zhang, Zhenqing Fu, Xiya Wu, Qingwei Lin, Dongmei Zhang: DeepTraLog: Trace-Log Combined Microservice Anomaly Detection through Graph-based Deep Learning. ICSE 2022, accepted.

基于大规模运维数据的云原生软件多维度分析

设计评估：从开发态评估到运行态评估



基于代码和提交历史等开发态数据分析各种关联关系，例如：

- ✓ 调用关联（代码静态分析）
- ✓ 数据关联（代码静态分析）
- ✓ 主题关联（代码文本分析）
- ✓ 演化关联（提交历史分析）

运行态



基于部署和运维日志、调用链路等运行态数据分析各种关联关系，例如：

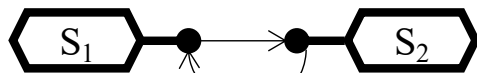
- ✓ 调用关联（调用链路分析）
- ✓ 数据关联（调用链路分析）
- ✓ 演化关联（部署历史分析）
- ✓ 指标关联（指标波动分析）

基于大规模运维数据的云原生软件多维度分析

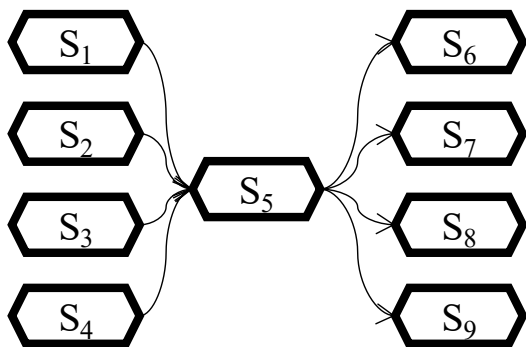
基于运维数据的云原生软件设计评估



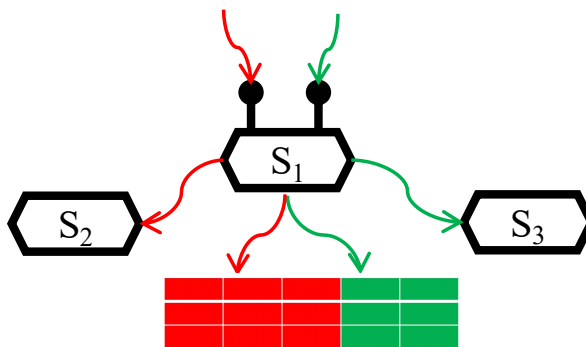
弱循环调用



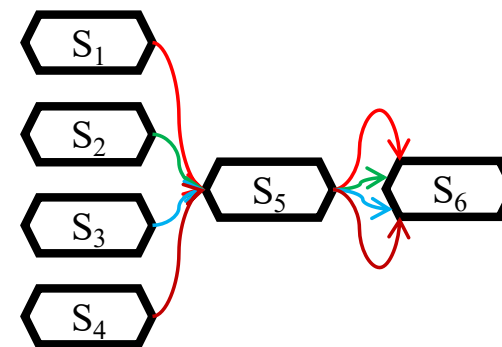
强循环调用



枢纽式服务



服务内聚度弱



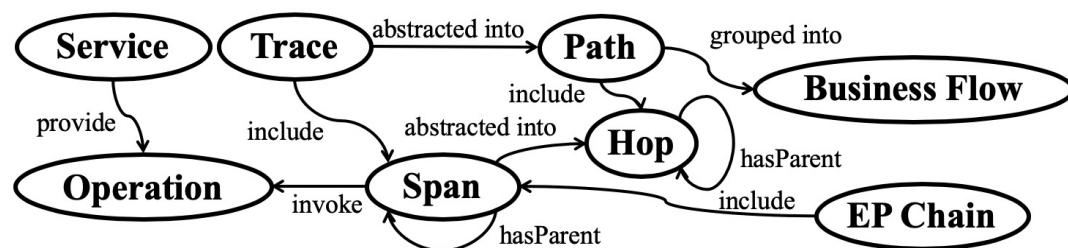
服务间耦合度高

基于调用链路等运维数据实现对于服务拆分、分布式通信、可用性保障等方面的设计评价

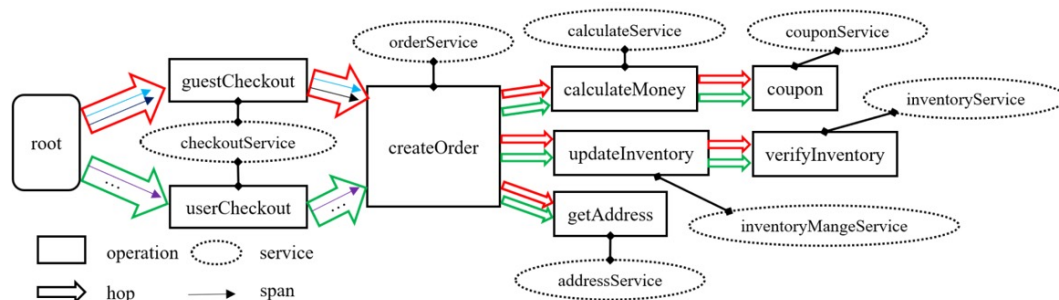
针对架构退化进行早期预警

基于大规模运维数据的云原生软件多维度分析

运行时调用链路聚合分析

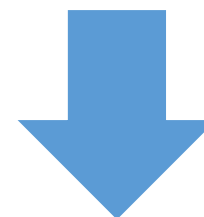


业务流分析概念模型



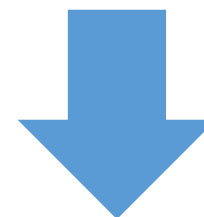
业务流分析示例

执行轨迹



按轨迹拓扑
结构聚合

执行路径



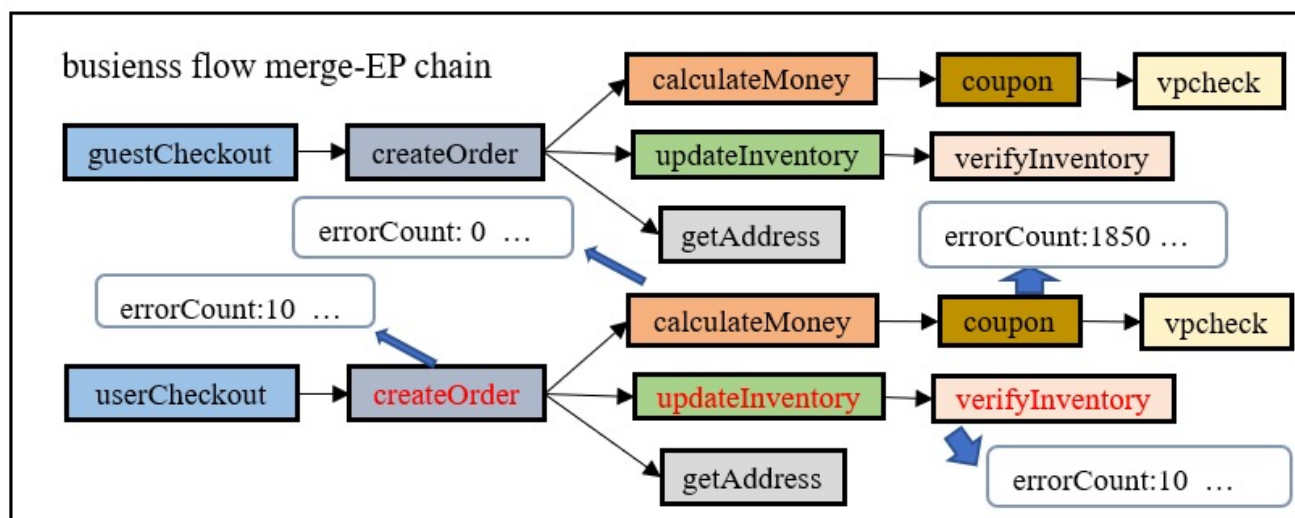
按途径的关键
服务聚合

业务流

Xiaofeng Guo, Xin Peng, Hanzhang Wang, Wanxue Li, Huai Jiang, Dan Ding, Tao Xie, Liangfei Su: Graph-based trace analysis for microservice architecture understanding and problem diagnosis. ESEC/SIGSOFT FSE 2020 (Industry Track).

基于大规模运维数据的云原生软件多维度分析

基于调用链路聚合的故障传播链可视化分析

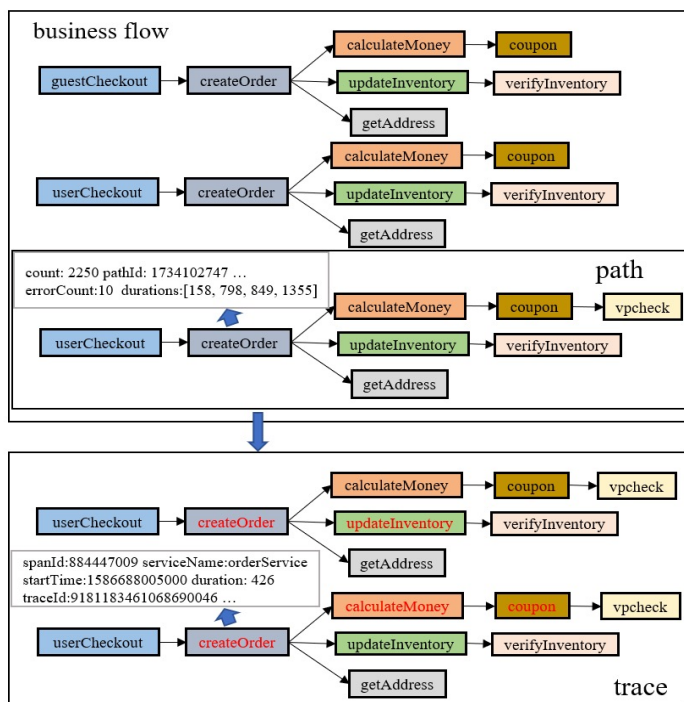


通过聚合后的调用链路
叠加出错指标度量，可
视化分析故障传播链

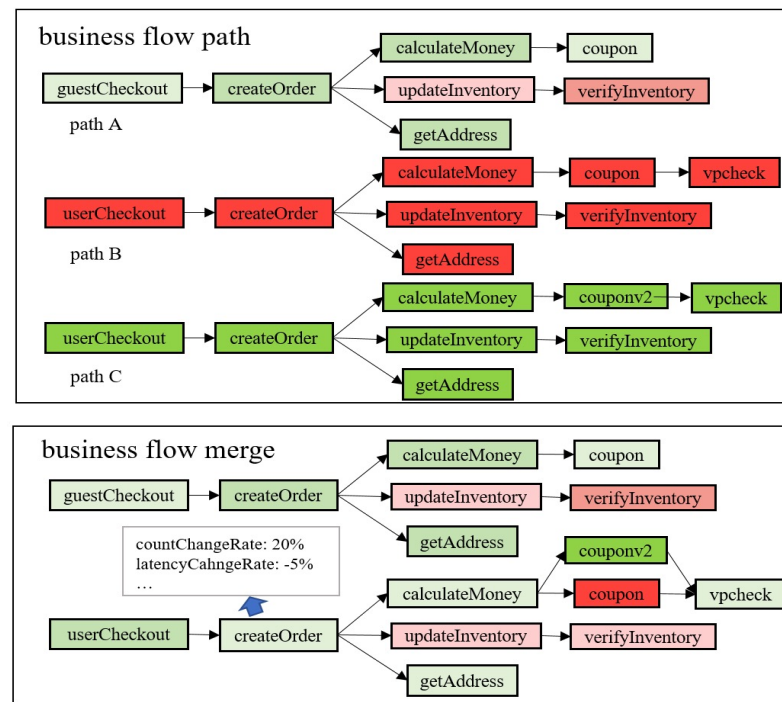
Xiaofeng Guo, Xin Peng, Hanzhang Wang, Wanxue Li, Huai Jiang, Dan Ding, Tao Xie, Liangfei Su: Graph-based trace analysis for microservice architecture understanding and problem diagnosis. ESEC/SIGSOFT FSE 2020 (Industry Track).

基于大规模运维数据的云原生软件多维度分析

基于调用链路聚合的业务流可视化分析



业务流可视化

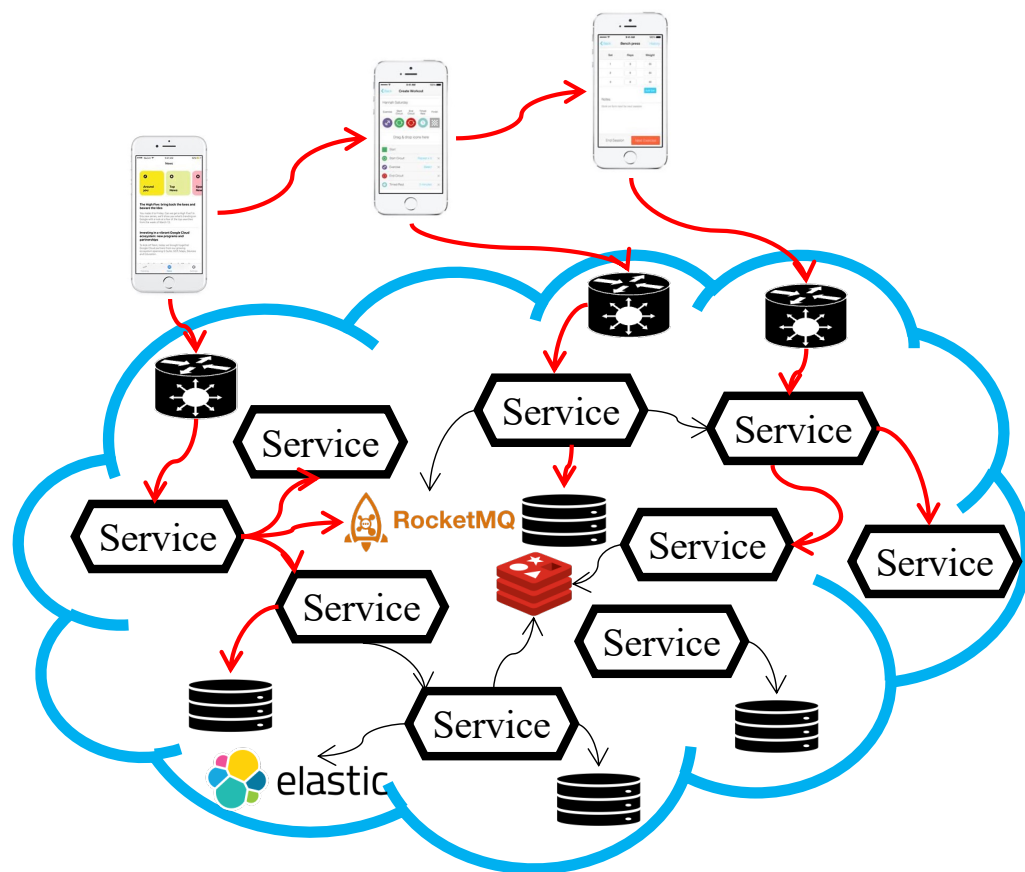


业务流变化比较

Xiaofeng Guo, Xin Peng, Hanzhang Wang, Wanxue Li, Huai Jiang, Dan Ding, Tao Xie, Liangfei Su: Graph-based trace analysis for microservice architecture understanding and problem diagnosis. ESEC/SIGSOFT FSE 2020 (Industry Track).

基于大规模运维数据的云原生软件多维度分析

基于前后端融合分析的业务视图恢复



问题：客户端逻辑复杂且持续变化，服务端调用链路难以反映完整的用户业务过程，难以获得全局的业务视图

思路：融合客户端界面导航结构及服务请求信息、客户端操作日志、服务端调用链路，实现基于运维数据的业务视图恢复

总结与展望

针对云原生软件系统持续进行全方位感知，实现全面大数据分析

04

基于大规模运维数据的云原生软件多维度分析

总结

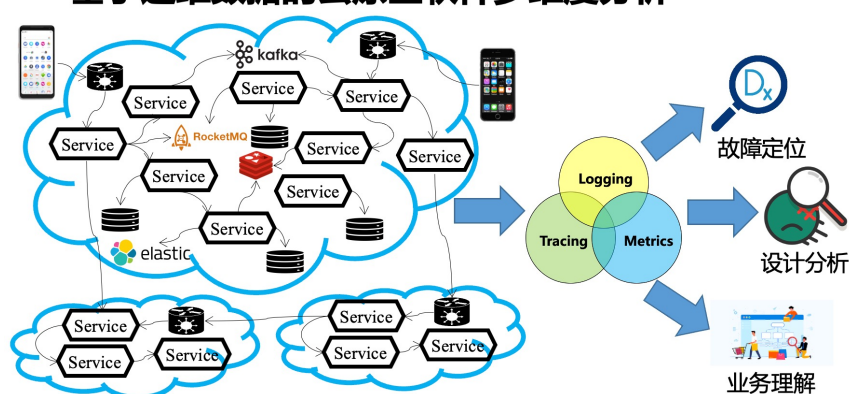
云原生软件生态系统



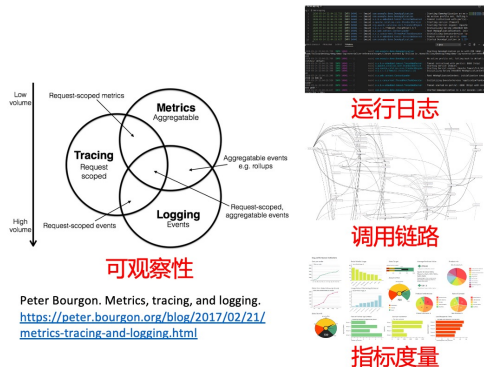
云原生软件分析的挑战



基于运维数据的云原生软件多维度分析



云原生软件分析的机遇



显式的软件体系结构
外化的交互行为
系统持续在线运行
完善的基础设施支持

基于大规模运维数据的云原生软件多维度分析

展望

云原生软件生态系统

越来越多的云原生软件系统会逐步演化成为复杂生态系统，新业务和特性不断生长，系统复杂度不断提高

丰富多元的监控数据采集

在当前以应急处理为目的的运行日志、调用链路、指标度量等监控数据收集基础上，面向云原生软件系统数字化体征分析的需要实现更加丰富多元的监控数据采集



开发运维一体化向纵深迈进

运维在关注应急处理的基础上越来越重视系统的长期健康演化和可持续发展，系统业务及技术运转状况的分析和理解越来越依赖于运行时数据分析

融合数据分析成为新的挑战

如何融合运行日志、调用链路、指标度量等多维、多源、多态运维数据，通过关联分析等手段获得更加全面、准确的运维分析视图

基于大规模运维数据的云原生软件多维度分析



CodeWisdom

Thanks

复旦大学

CodeWisdom



基于大规模运维数据的云原生软件多维度分析