Localizing Failure Root Causes in a Microservice through Causality Inference

Yuan Meng[1], Shenglin Zhang[2], Yongqian Sun[2]
Ruru Zhang[2], Zhilong Hu[2], Yiyin Zhang[3], Chenyang Jia[3], Zhaogang Wang[3], Dan Pei[1]

清華大學 Tsinghua University [1]
南開大學 Nankai University [2]
阿里巴巴 Alibaba.com [3]

IEEE/ACM International Symposium on Quality of Service 15-17 June 2020 // Virtual Conference

Outline

Background

Algorithm

Evaluation

Case Studies

Outline

Background

Algorithm

Evaluation

Case Studies

Failures in Microservice

- **Microservice** has gained an increasing popularity in recent years.

- **Performance quality** of microservice is of vital importance to the Internet company and the users
  - On May 18 in 2020, Zoom, experienced **a wide range of failures**. The COVID-19 official Briefing of British Government was forced to cancel.
  - On March 26 in 2020, Google service broke down **for 20 minutes**.
  - Netflix **reduced stream quality** to meet additional demand

- **Efficient root cause localization** of online failures in microservice enables rapid service recovery and loss mitigation.

Microservice architecture

- In the microservice architecture, an application is **decoupled** into multiple microservices.

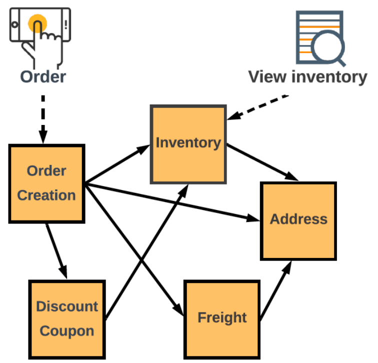


Fig.1. The call graph of microservices in the process of placing an order

Cross-microservices root cause localization tries to understand how a failure is propagated across microservices and aims to localize the ***root cause microservice,*** e.g. the Address microservice.

The failure root cause in Address may be the CPU, network, memory, etc.

- In the literature, only the **cross-microservices** root cause localization has been investigated.
- Failure root causes **within a microservice** is still not clear for the operators.

A Microservice

- How does a microservice work?

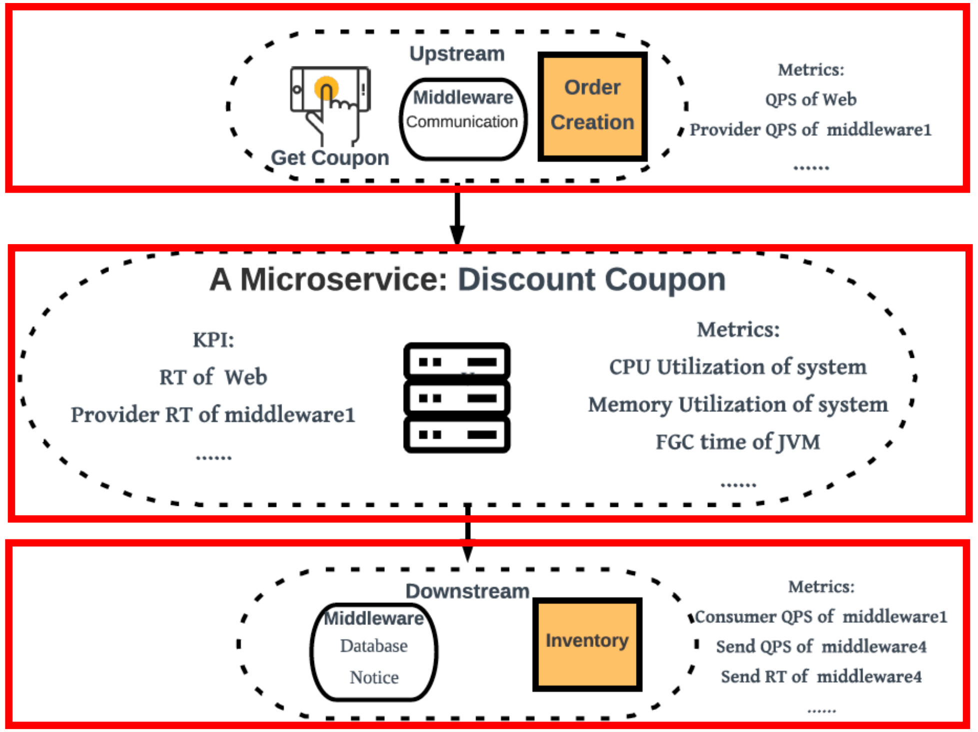


Fig.1. An example of the microservice

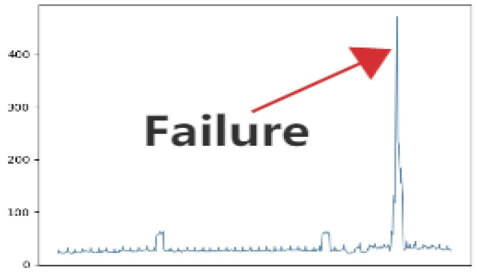
KPI(key **p**erformance **I**ndicator):a user-perceived indicator that
directly reflects the quality of service.

Metric: an indicator indicates the status of a microservice' s underlying component.

Potential root causes

Problem Statement

- **Failure ticket**
  - a microservice ID indicating where the failure occurs
  - a KPI representing which KPI becomes anomalous when this failure occurs
  - a timestamp showing when this failure happens.
  - E.g. {Microservice A, RT of Web, 17:18}
- **Problem definition**



Failure ticket

Top N root causes

Rank	Metrics
1	Web QPS
2	JVM YGC Time
...	...

Related work

Type	Model	Relationship learning	Root cause inference
Cross-microservice	Microscope[SOC18]	PC	Pearson correlation
	CloudRanger[CCGGRID18]	PC	Second order random walk
	MonitorRank[SIGMETRICS13]	Hadoop tools	random walk
	TON18	OpenStack APIs	random walk
Intra-microservice	**MicroCause (our method)**	**PCTS**	**TCORW**

Data analysis

System tools

Challenges

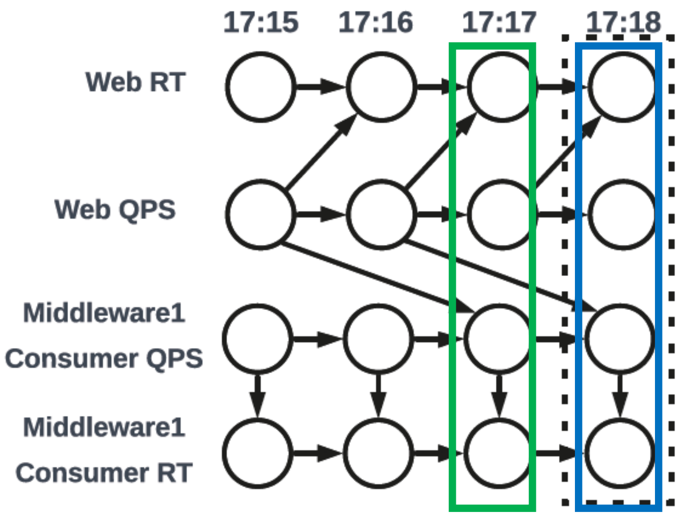


Fig.1:Causal relationship among a KPI and three metrics. A circle denotes a time point of a KPI/metric, and an arrow represents a causal relationship

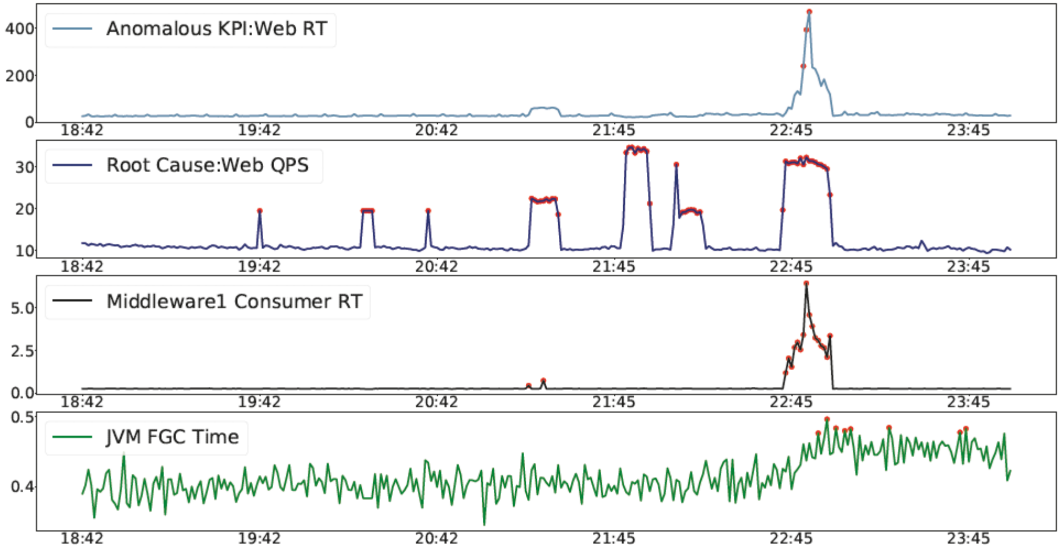


Fig.2: Monitoring indicators of failure case {Microservice A, Web RT, 22:45-22:55}

Challenge 1: *iid* based causal graph(e.g. PC) cannot capture propagation delays.

Challenge 2: Correlation based random walk may not accurately localize root cause.

Outline

Background

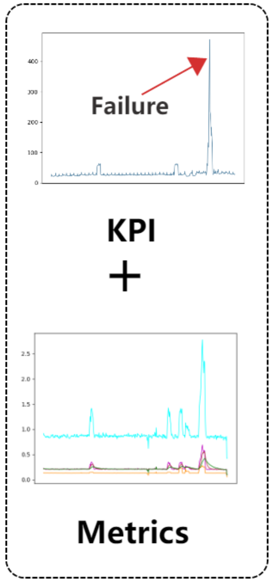
Algorithm

Evaluation

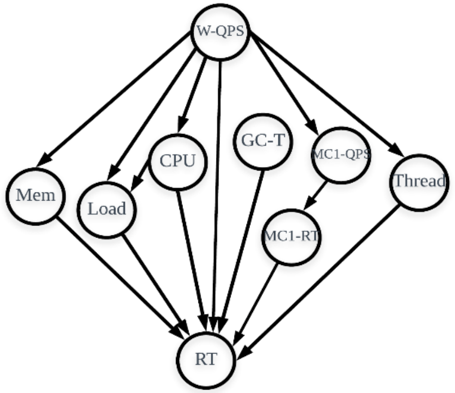
Case Studies

Model Architecture

Failure ticket X:
{Microservice A, RT of Web, 17:18}



Failure Casual Graph Learning



Temporal Cause Oriented Random Walk

Anomaly Detection

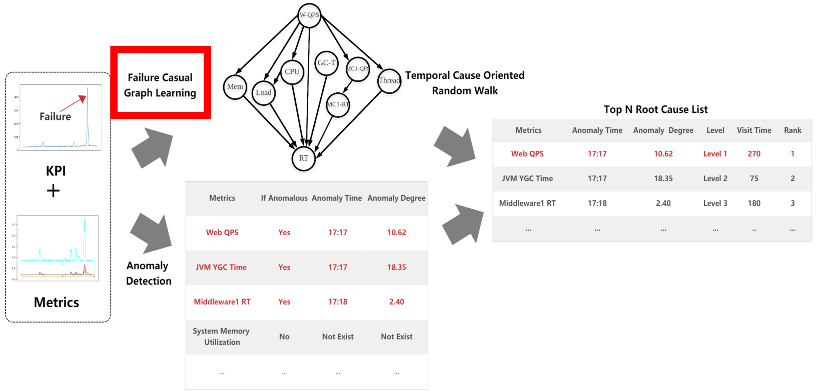
Metrics	If Anomalous	Anomaly Time	Anomaly Degree
Web QPS	Yes	17:17	10.62
JVM YGC Time	Yes	17:17	18.35
Middleware1 RT	Yes	17:18	2.40
System Memory Utilization	No	Not Exist	Not Exist
...	...	...	...

Top N Root Cause List

Metrics	Anomaly Time	Anomaly Degree	Level	Visit Time	Rank
Web QPS	17:17	10.62	Level 1	270	1
JVM YGC Time	17:17	18.35	Level 2	75	2
Middleware1 RT	17:18	2.40	Level 3	180	3
...	...	...	...	..	...

MicroCause

MicroCause



- **Failure Causal Graph Learning**
  - ➢ **PCTS**

Challenge 1

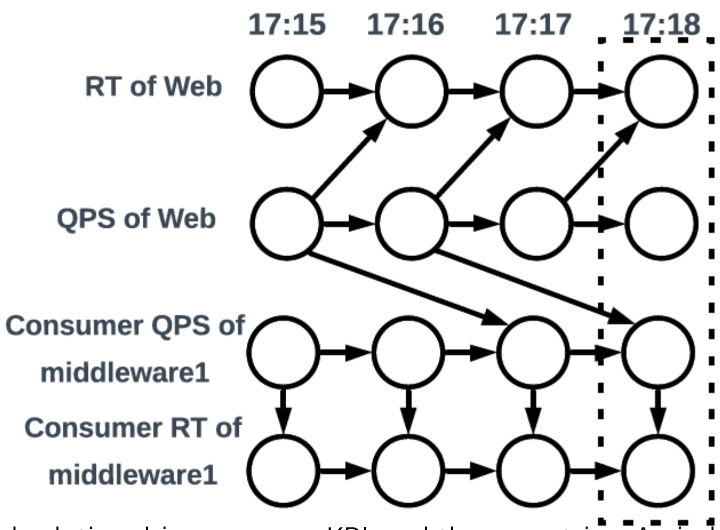


Fig.1:Causal relationship among a KPI and three metrics. A circle denotes a time point of a KPI/metric, and an arrow represents a causal relationship

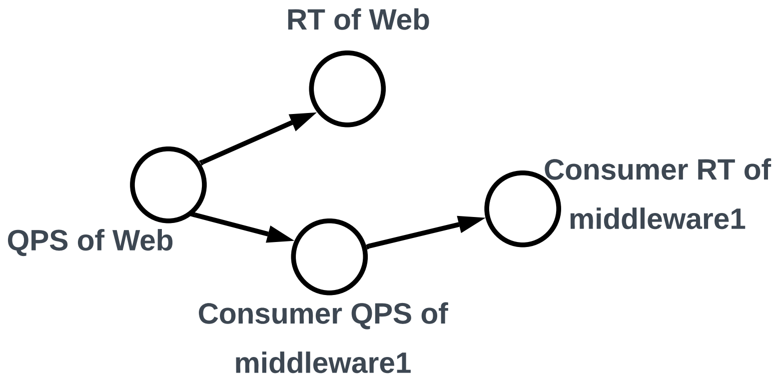


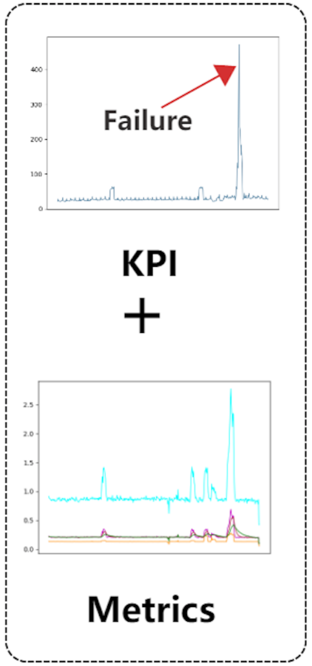
Fig.2: Failure causal graph between a KPI and three metrics

Step1: Improved PC algorithm[1]

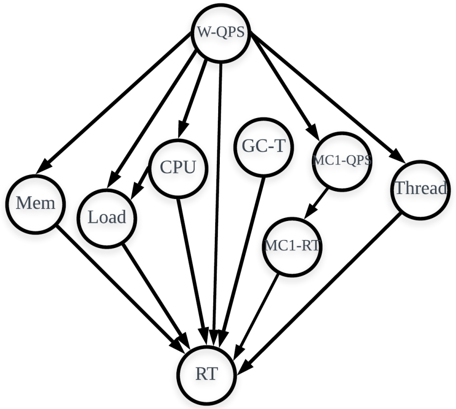
Step2: Generate failure causal graph

[1] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic, "Detecting and quantifying causal associations in large nonlinear time series datasets," Science Advances, vol. 5, no. 11, p. eaau4996, 2019.

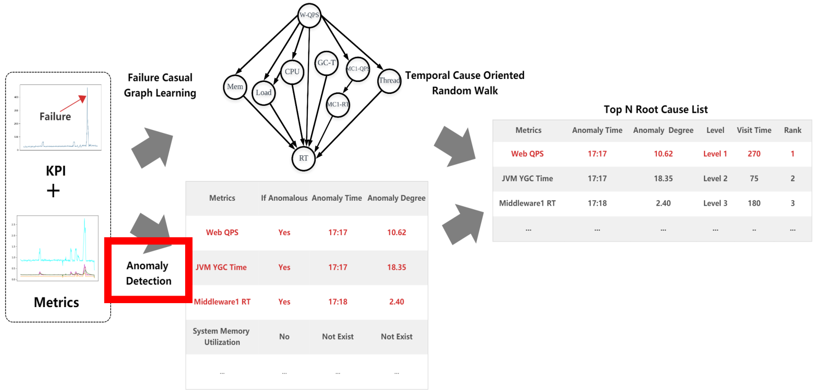
MicroCause



Failure Causal Graph Learning



MicroCause



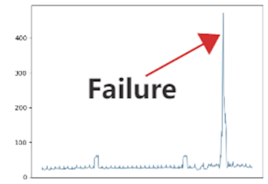
- **Anomaly detection**
  - ➢ **SPOT[KDD17]**
    - ➢ detects the sudden change in time series via the extreme value theory
  - ➢ **Anomaly degree**

$$\eta^i_{max} = \max_{k \in O} \frac{|M^i_k - \phi_{M^i_k}|}{\phi_{M^i_k}}$$

Challenge 2

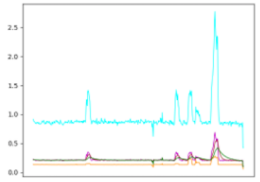
O is the index set of the anomaly point, $\phi_{M^i_t}$ is the threshold

MicroCause



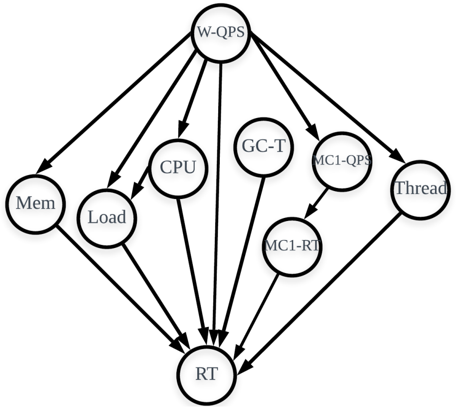
KPI

+



Metrics

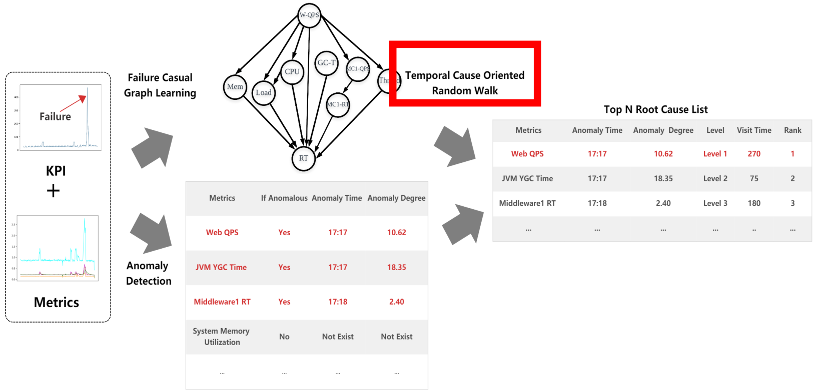
Failure Causal Graph Learning



Anomaly detection

Metrics	If Anomalous	Anomaly Time	Anomaly Degree
QPS of Web	Yes	17:17	10.62
YGC Time of JVM	Yes	17:17	18.35
RT of Middleware1	Yes	17:18	2.40
Memory Utilzation of System	No	Not Exist	Not Exist
...	...	...	...

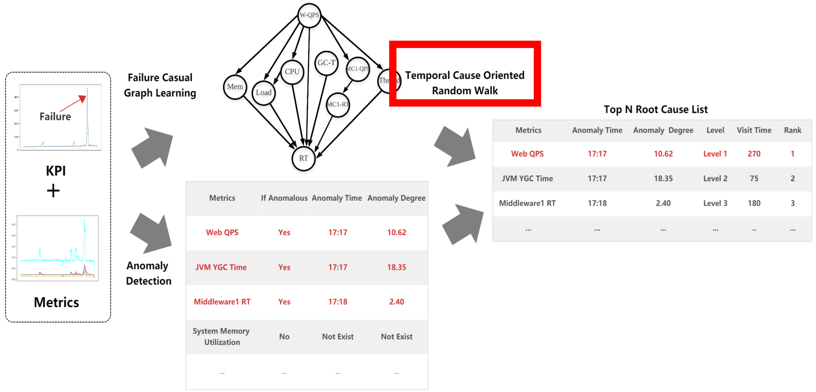
MicroCause



- **Temporal Cause Oriented Random Walk(TCORW)**
  - ➢ Step one: Cause oriented random walk (causal relationship)
  - ➢ Step two: Potential root cause score (+ anomaly degree)
  - ➢ Step three: Rank the root causes (+ priority, anomaly time)

Challenge 2

MicroCause



- **Temporal Cause Oriented Random Walk(TCORW)**
  - **Step one: Cause oriented random walk (causal relationship)**
    - *Partial correlation*
    - Forward step (walk from effect indicator to cause indicator)

$$Q_{ij} = R_{pc}(v_{ak}, v_j | Pa(v_{ak}) \setminus v_j, Pa(v_j))$$

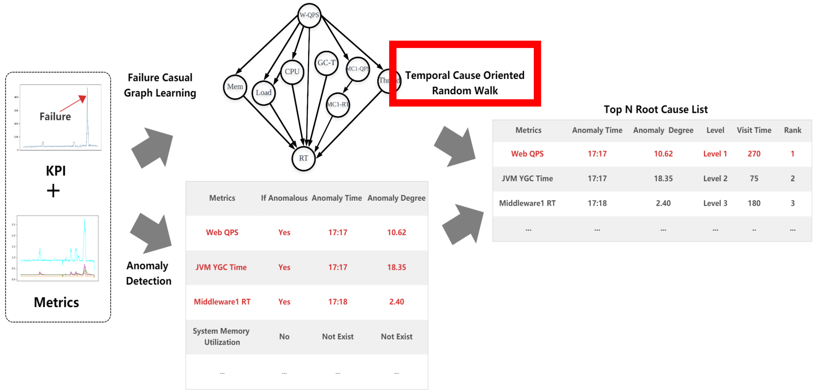
    - Backward step (walk from cause indicator to effect indicator)

$$Q_{ji} = \rho R_{pc}(v_{ak}, v_i | Pa(v_{ak}) \setminus v_i, Pa(v_i))$$

    - Self step (stay in the present node):

$$Q_{ii} = \max[0, R_{pc}(v_{ak}, v_i \mid Pa(v_{ak}) \setminus v_i, Pa(v_i)) - P_{pc}^{max}]$$
$$P_{pc}^{max} = \max_{k:e_{ki} \in E} R_{pc}(v_{ak}, v_k \mid Pa(v_{ak}) \setminus v_k, Pa(v_k))$$

MicroCause

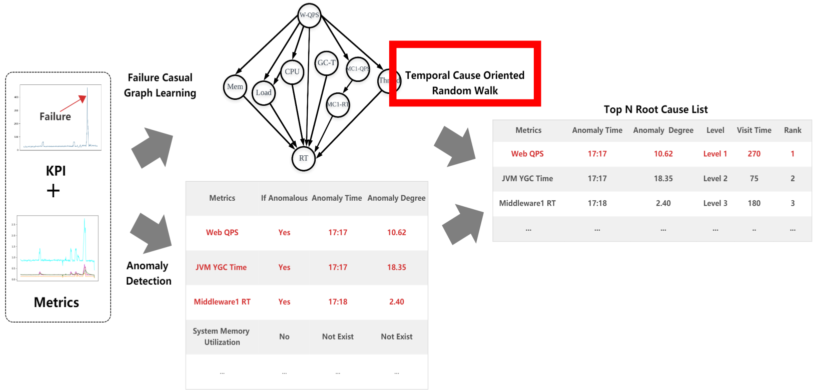


- **Temporal Cause Oriented Random Walk(TCORW)**
  - **Step two: Potential root cause score (+ anomaly degree)**

$$\gamma_i = \lambda \bar{c}_i + (1 - \lambda)\bar{\eta}^i_{max}$$

$\bar{c}_i$ is the normalized visit time c_i. $\bar{\eta}^i_{max}$ is the normalized anomaly degree η^i_{max}. λ controls the contribution of metric's causal relationship with the anomalous KPI and the anomaly degree of the metric.

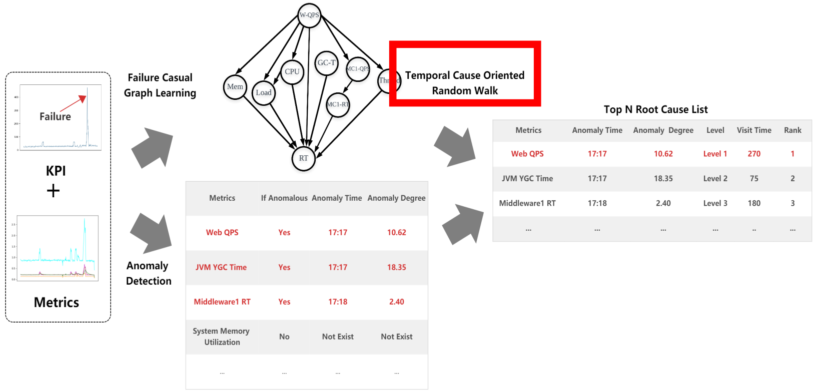
MicroCause



- **Temporal Cause Oriented Random Walk(TCORW)**
  - **Step three: Rank the root causes (+ priority, anomaly time)**

Metrics Type	Metrics	Priority
Upstream	QPS of Web; Provider QPS of middleware1; Receive QPS of middleware2; Receive QPS of middleware3	Level 1
Self	**Java virtual machine (JVM) related**: YGC count of JVM; YGC time of JVM; FGC count of JVM; FGC time of JVM; Max heap memory of JVM; Used heap memory of JVM; Usage heap memory of JVM; Max nonheap memory of JVM; Used nonheap memory of JVM; Usage metaspace pools of JVM memory; Usage code cache pools of JVM memory; Max mapped bufferpool of JVM; Used mapped bufferpool of JVM; Max direct bufferpool of JVM; Used direct bufferpool of JVM; Thread count of JVM; Deamon thread count of JVM; Deadlock thread count of JVM; Runnable thread count of JVM; File descriptor utilization of JVM;	Level 2
	System related: CPU utilization of system; CPU steal of system; Load1 utilization of system; Load5 utilization of system; Load15 utilization of system; Load1 of system; Load5 of system; Load15 of system; Memory utilization of system; Swap utilization of system; Net in of system; Net out of system; Net retran utilization of system; Net established of system; Disk utilization of system; Disk read of system; Disk write of system; Dish inode of system;	Level 2
Downstream	**Queries per second (QPS):** Consumer QPS of middleware1; Read QPS of middleware4; Write QPS of middleware1; Read QPS of middleware5; Write QPS of middleware5; Send QPS of middleware2; Send QPS of middleware3;	Level 2
	Response time (RT): Consumer RT of middleware1; Read RT of middleware4; Write RT of middleware4; Read RT of middleware5; Write RT of middleware5; Send RT of middleware2; Send RT of middleware3;	Level 3
	Success rate: Consumer success rate of middleware1; Read success rate of middleware4; Write success rate of middleware4; Read success rate of middleware5; Write success rate of middleware5; Send success rate of middleware2; Send success rate of middleware3;	Level 3

MicroCause



- **Temporal Cause Oriented Random Walk(TCORW)**
  - **Step three: Rank the root causes (+ priority, anomaly time)**

Algorithm 1: Rank the root cause

Input: 1 Levels of metrics, 2 γ_i of metric i, 3 anomaly time t_i of metric i

Output: RankResultSet

ResultSet $\leftarrow$ [ ]

for *j=1,2,3* **do**

  $R_j \leftarrow$ rank metrics in Level j by γ_i in descending order.

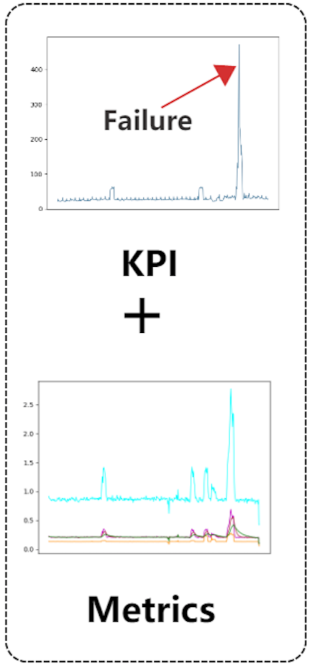
  ResultSet $\leftarrow$ append the top 2 result in R_j

end

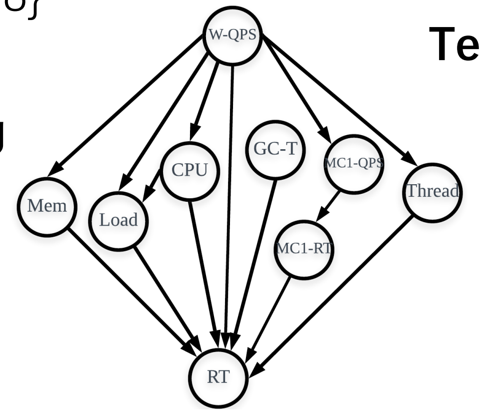
RankResultSet $\leftarrow$ rank ResultSet by t_i in ascending order

MicroCause

Failure ticket X:
{Microservice A, RT of Web, 17:18}



Failure Causal Graph Learning



Temporal Cause Oriented Random Walk

Anomaly detection

Metrics	If Anomalous	Anomaly Time	Anomaly Degree
QPS of Web	Yes	17:17	10.62
YGC Time of JVM	Yes	17:17	18.35
RT of Middleware1	Yes	17:18	2.40
Memory Utilzation of System	No	Not Exist	Not Exist
...	...	...	...

Top N Root Cause List

Metrics	Anomaly Time	Anomaly Degree	Level	Visit Time	Rank
QPS of Web	17:17	10.62	Level 1	270	1
YGC Time of JVM	17:17	18.35	Level 2	75	2
RT of Middleware1	17:18	2.40	Level 3	180	3
...	...	...	...	..	...

Outline

Background

Algorithm

Evaluation

Case Studies

Dataset & Evaluation Metrics

- **Dataset**
  - 86 online failure tickets in an online shopping platform
  - monitoring more than 400 microservice status.
  - Sep. 2019 to Jan. 2020
  - 4 KPIs:
    - RT of Web
    - provider RT of middleware1
    - receive RT of middleware2
    - receive RT of middleware3.
  - Metrics
- **Evaluation Metrics**

$$AC@k = \frac{1}{|A|} \sum_{a \in A} \frac{\sum_{i<k} R^a[i] \in V_{rc}^a}{min(k, |V_{rc}^a|)} \qquad Avg@k = \frac{1}{k} \sum_{1 \leq j \leq k} AC@j$$

MicroCause VS baseline methods

Method	AC@1	AC@2	AC@5	Avg@5
MicroCause	**46.7%**	**62.7%**	**98.7%**	**69.7%**
TON18, MonitorRank[SIGMETRICS13]	34.7%(-12.0%)	48.0%(-14.7%)	65.3%(-33.4%)	48.2%(-21.5%)
CloudRanger[CCGGRID18]	19.0%	32.9%	69.6%	46.8%
Microscope[SOC18]	12.2%	21.9%	29.3%	23.9%
Anomaly Time Order	11.4%	21.5%	43.0%	28.4%

Analysis about MicroCause

- **Evaluation of PCTS**

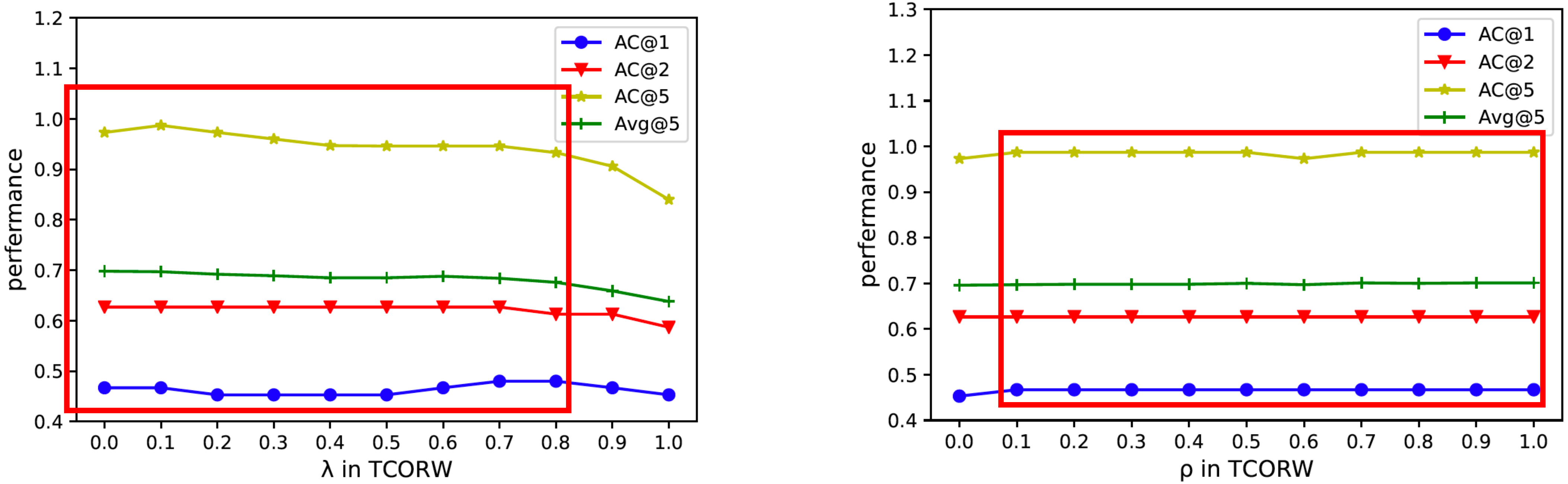
Method	AC@1	AC@2	AC@5	Avg@5
MicroCause	**46.7%**	**62.7%**	**98.7%**	**69.7%**
MicroCause w/PC	44.9%(-1.8%)	59.0%(-3.7%)	93.6%(-5.1%)	67.4%(-2.3%)

- **Evaluation of TCORW**

Method	AC@1	AC@2	AC@5	Avg@5
MicroCause	**46.7%**	**62.7%**	**98.7%**	**69.7%**
MicroCause w/RW-1	34.7%(-12.0%)	48.0%(-14.7%)	65.3%(-33.4%)	48.2%(-21.5%)
MicroCause w/RW-2	29.3%	46.7%	62.7%	46.3%

Analysis about MicroCause

- **Parameters in MicroCause**



Outline

Background

Algorithm

Evaluation

Case Studies

Causal graph of time series

- **Isolated subgraphs via PC[INFOCOM14]**

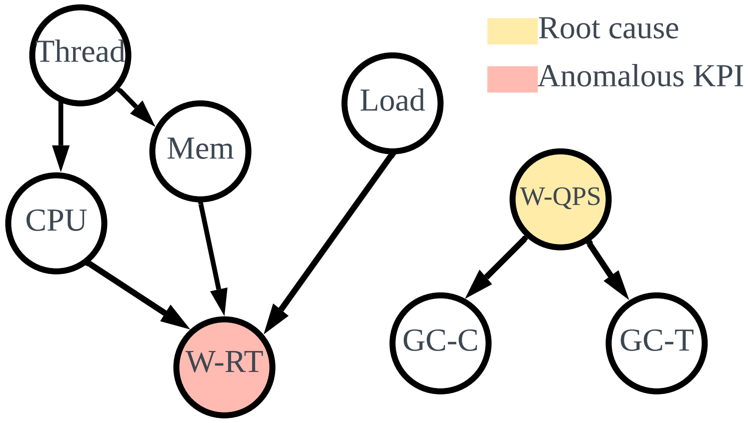


Fig1: Failure causal graph via PC algorithm of failure ticket X {Microservice A, RT of Web, 17:18}

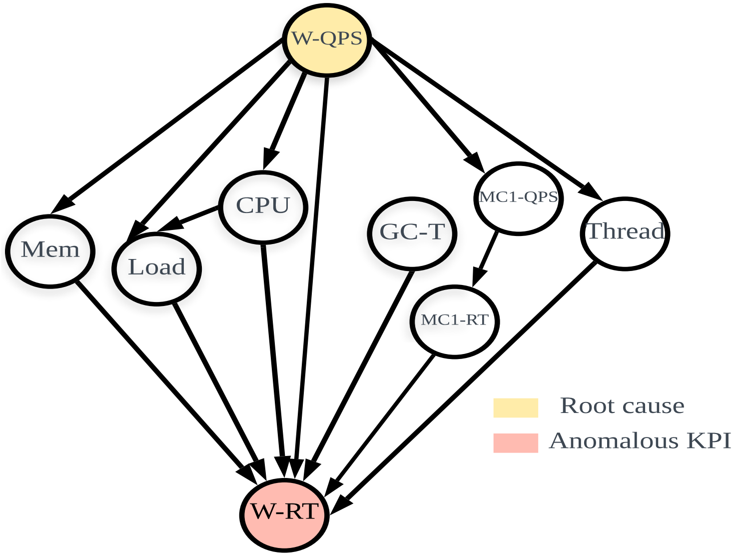


Fig2: Failure causal graph via PCTS algorithm of failure ticket X {Microservice A, RT of Web, 17:18}

Conclusion

- To the best of our knowledge, this paper is the first attempt to investigate the failure root cause in a microservice
- We design a framework, **MicroCause**, to localize the failure root cause in a microservice, which achieves high performance in the experiments based on 86 the online failure tickets.
- In MicroCause, we design PCTS, which can learn the causal graph of monitoring indicators. We believe it can be used in other time series related root cause localization problems.

Thank you!
Q & A

meng-y16@mails.tsinghua.edu.cn

IWQOS 2020