Illuminating the Gray Zone: Non-intrusive Gray Failure Localization in Server Operating Systems

Shenglin Zhang[1], Yongxin Zhao[1], Xiao Xiong[1], Yongqian Sun[1], Xiaohui Nie[2], Jiacheng Zhang[1], Fenglai Wang[3], Xian Zheng[3], Yuzhi Zhang[1], Dan Pei[4]

[1]Nankai University, [2]Chinese Academy of Science
[3]Huawei Technologies, [4]Tsinghua University

01

Background

02

Design

03

Evaluation

04

Conclusion

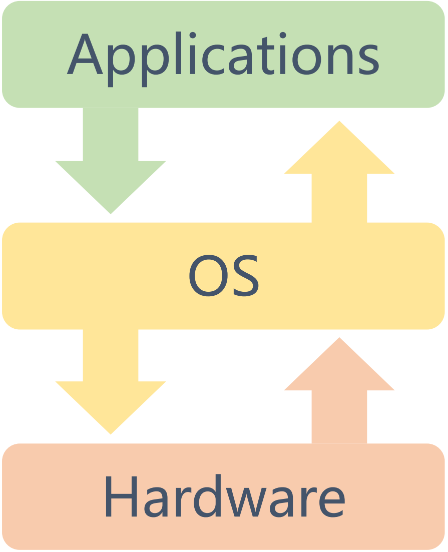
Server OS

Servers support countless applications and services

- Servers serve as the core of large-scale data management and a key component in providing network services.

Server operating system (OS) acts as an intermediary between applications and the server hardware

- Server OS enables applications to run efficiently and securely on hardware.

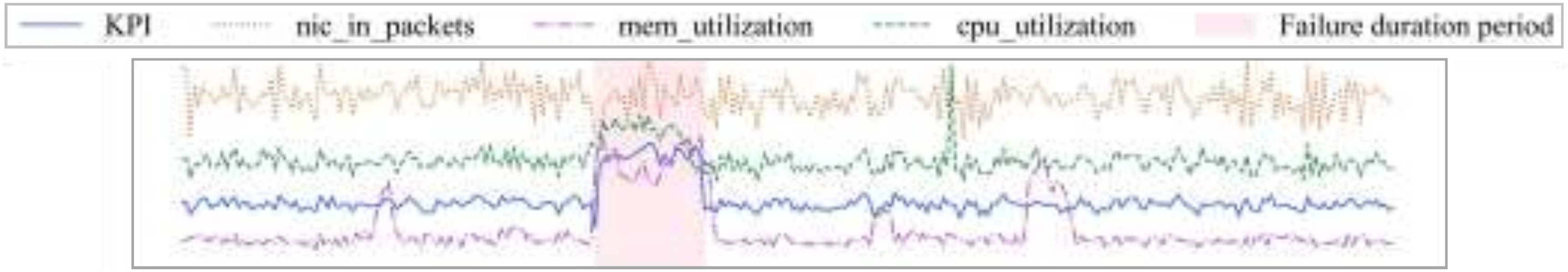


Gray failures occur frequently but are difficult to localize

- Gray failures are the root cause of many catastrophic failures in the real world.
- When one component becomes unhealthy, it will likely impact the performance
-

Timely and accurate localization and mitigation of gray failures in server OSes are crucial for ensuring their high availability

Anomalies on KPIs often signal potential gray failures, and root cause metrics exhibit anomalies and correlate with the KPI



Expert knowledge is essential for accurate causality learning

Method	Disk Failure_1	Disk Failure_2	Delay Failure_1	Delay Failure_2	Packet Loss Failure_1	Packet Loss Failure_2	CPU Failure_1	CPU Failure_2
Granger causality tests [10] w knowledge	76 (✓)	92 (✓)	88 (✓)	81 (✓)	42 (✓)	142 (✓)	63 (✓)	54 (✓)
Granger causality tests [10] w/o knowledge	297 (×)	345 (×)	152 (✓)	153 (✓)	155 (×)	395 (×)	210 (✓)	217 (×)
PC algorithm [39] w knowledge	12 (×)	42 (✓)	7 (×)	6 (×)	16 (✓)	15 (×)	31 (✓)	3 (×)
PC algorithm [39] w/o knowledge	59 (×)	95 (×)	40 (×)	43 (×)	54 (✓)	64 (×)	60 (×)	53 (×)
PCTS algorithm [30] w knowledge	32 (✓)	47 (✓)	52 (✓)	50 (×)	48 (✓)	45 (✓)	64 (✓)	43 (×)
PCTS algorithm [30] w/o knowledge	40 (✓)	51 (×)	69 (✓)	63 (×)	73 (✓)	48 (✓)	64 (✓)	89 (×)

Related Work

Research on root cause localization for gray failures is scarce

- Some intrusive methods rely on modifying the source code of applications, limiting their practical deployment due to high modification costs and long localization cycles.

A collection of metric-based root cause localization methods has been proposed for distributed systems

- Statistical methods are easily affected by data noise.
- Feature learning methods often rely on many high-quality labeled cases.
- Causality graph-based methods are promising for non-intrusive metric-based gray failure localization in server OS.

Challenges

Complex causal relationships between metrics

- Server OSes feature hundreds of dynamically changing metrics, with evolving relationships between them.

Underutilization of the correlations

- The correlation between metrics and the gray failure can guide the root cause inference method to localize the metrics causing the gray failure.

Interpretability

- A lack of information about the propagation paths of gray failures can affect the efficiency of operators in mitigating failures.

Contributions

Complex causal relationships between metrics

Integrates expert knowledge with causal learning techniques

Contributions

Underutilization of the correlations

Combines partial correlation with anomaly degree

Contributions

Interpretability

Infers the gray failure propagation paths between metrics

01

Background

02

Design

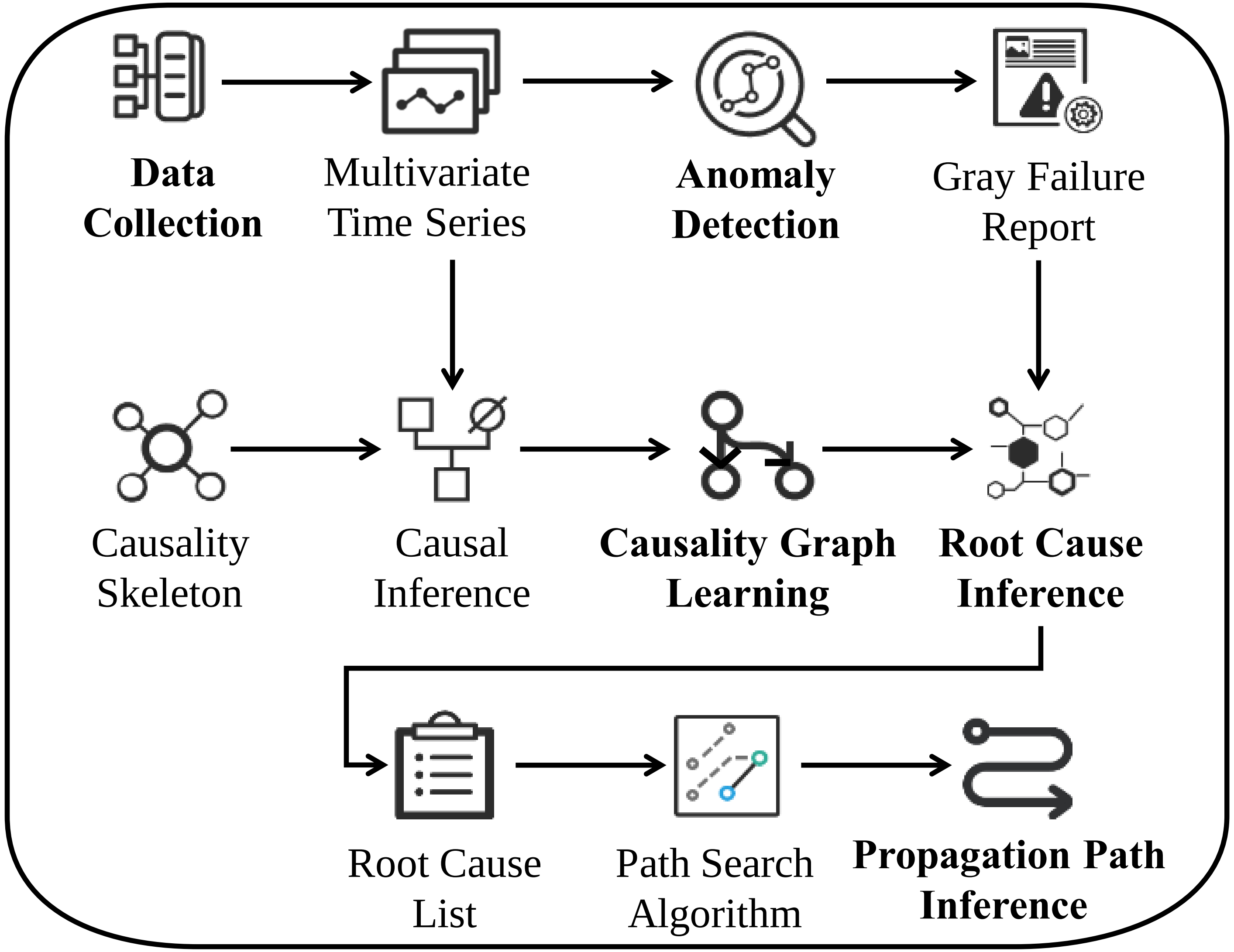
03

Evaluation

04

Conclusion

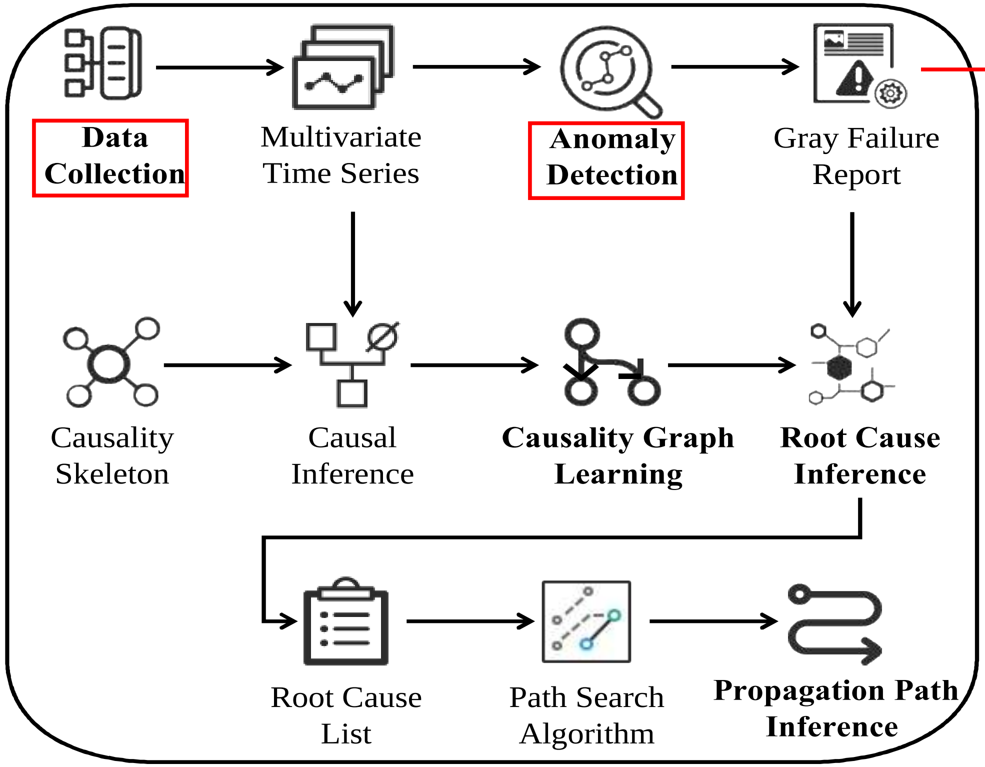
GrayScope Overview



Four key modules:

- Data Collection and Anomaly Detection
- Causality Graph Learning
- Root Cause Inference
- Propagation Path Inference

Data Collection and Anomaly Detection



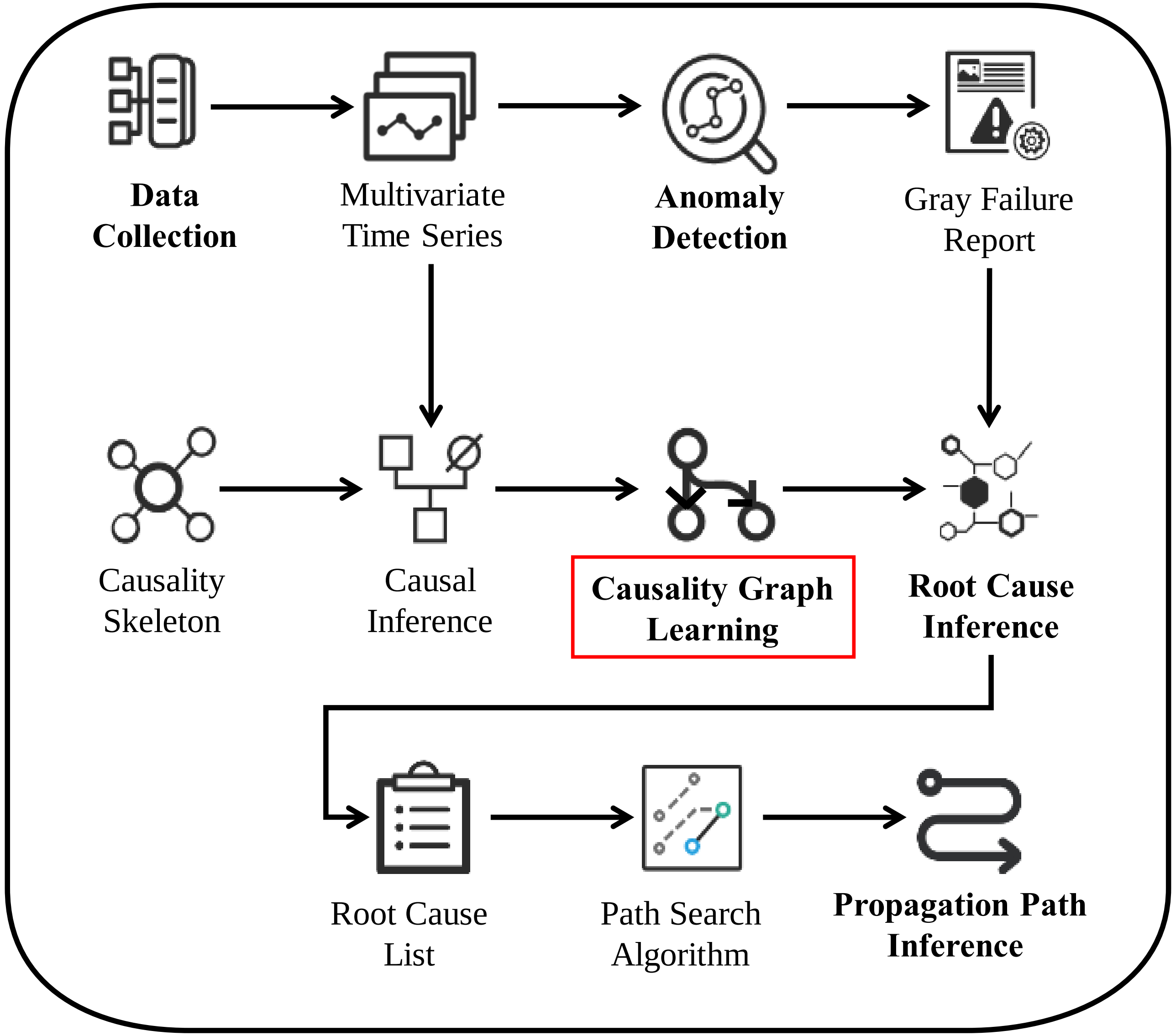
Gray Failure Report

Timestamp: 1698030369485,
Server OS: server_os_1
Resource: {
    Anomalous KPI: redis_sli_rtt_nsec,
    Anomaly Score: 9.42558,
    Cause Metrics: [
        {
            Metric: tcp_link_retran_packets,
            Anomaly Score: 5.03055
        },
        {
            Metric: tcp_link_avl_snd_wnd,
            Anomaly Score: 4.79047
        },
        {
            Metric: nic_tc_sent_drop,
            Anomaly Score: 4.57826
        },
        ...
    ]
}

- The Data Collection module gathers multiple runtime information from the server OS across various data sources, including system calls, applications, and communications.

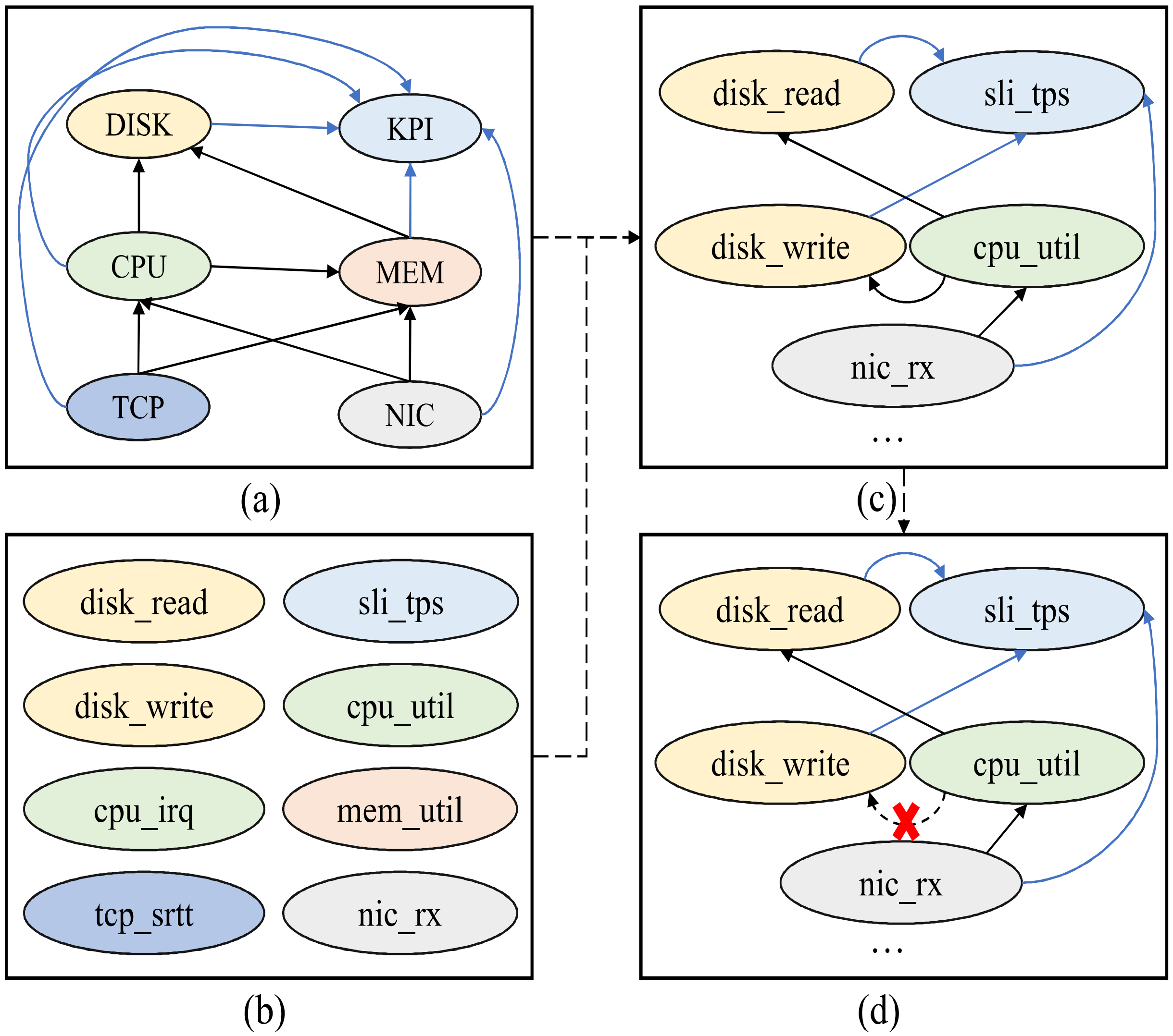
- The Anomaly Detection module identifies anomalies in KPI and reports the gray failure occurring in the system.

Causality Graph Learning



- Learning effective causality graphs is crucial for failure root cause localization.
- Granger causality tests, a method of time series analysis used to test for causality between two time series, to learn causality graphs between metrics.
- We propose a causality graph learning model that combines expert knowledge with Granger causality tests.

Causality Graph Learning

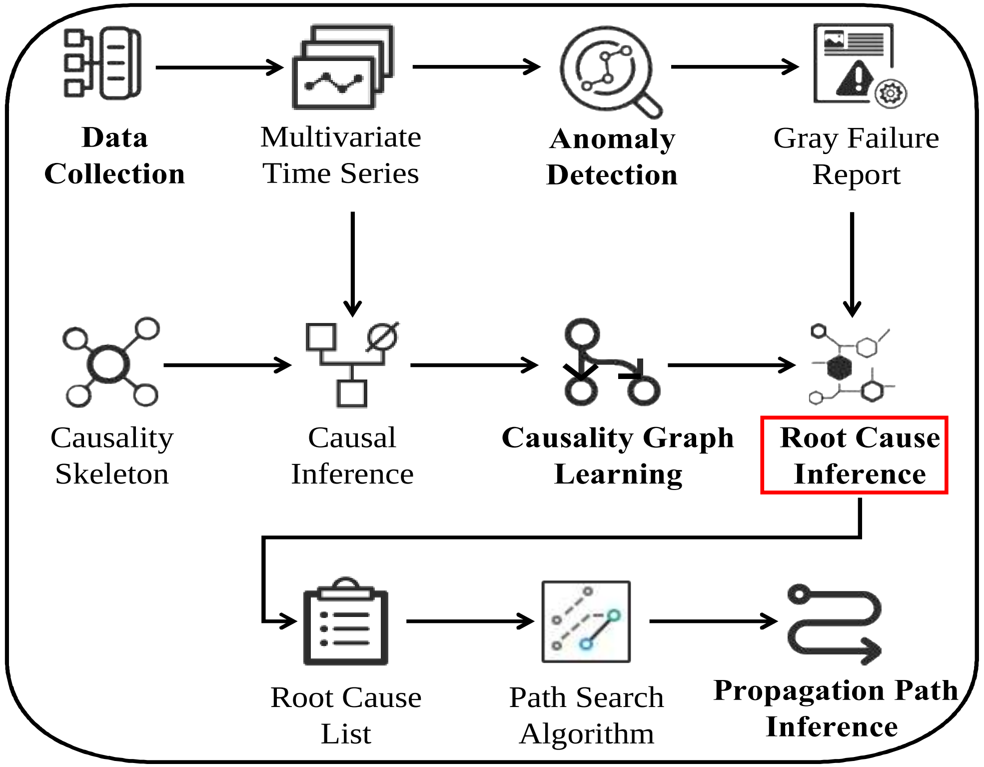


(a) We construct a causality skeleton graph of meta metrics for server OS gray failures by leveraging expert knowledge.

(b) We insert the top m related metrics for each category of meta-metrics.

(c) We connect related metrics fully and construct the metric causality structure graph.

(d) We perform the Granger causality test for all related metrics and preserve the anomalous KPI subgraph, resulting in the learned metric causality graph.



$$anomaly\_degree(v_j) = \frac{anomaly\_score(v_j)}{anomaly\_score(v_i) + anomaly\_score(v_j)} \quad (1)$$

- Forward step (walk from result metric to cause metric):

$$H'_{i,j} = \lambda \cdot correlation(v_j) + (1-\lambda) \cdot anomaly\_degree(v_j) \quad (2)$$

- Backward step (walk from cause metric to result metric):

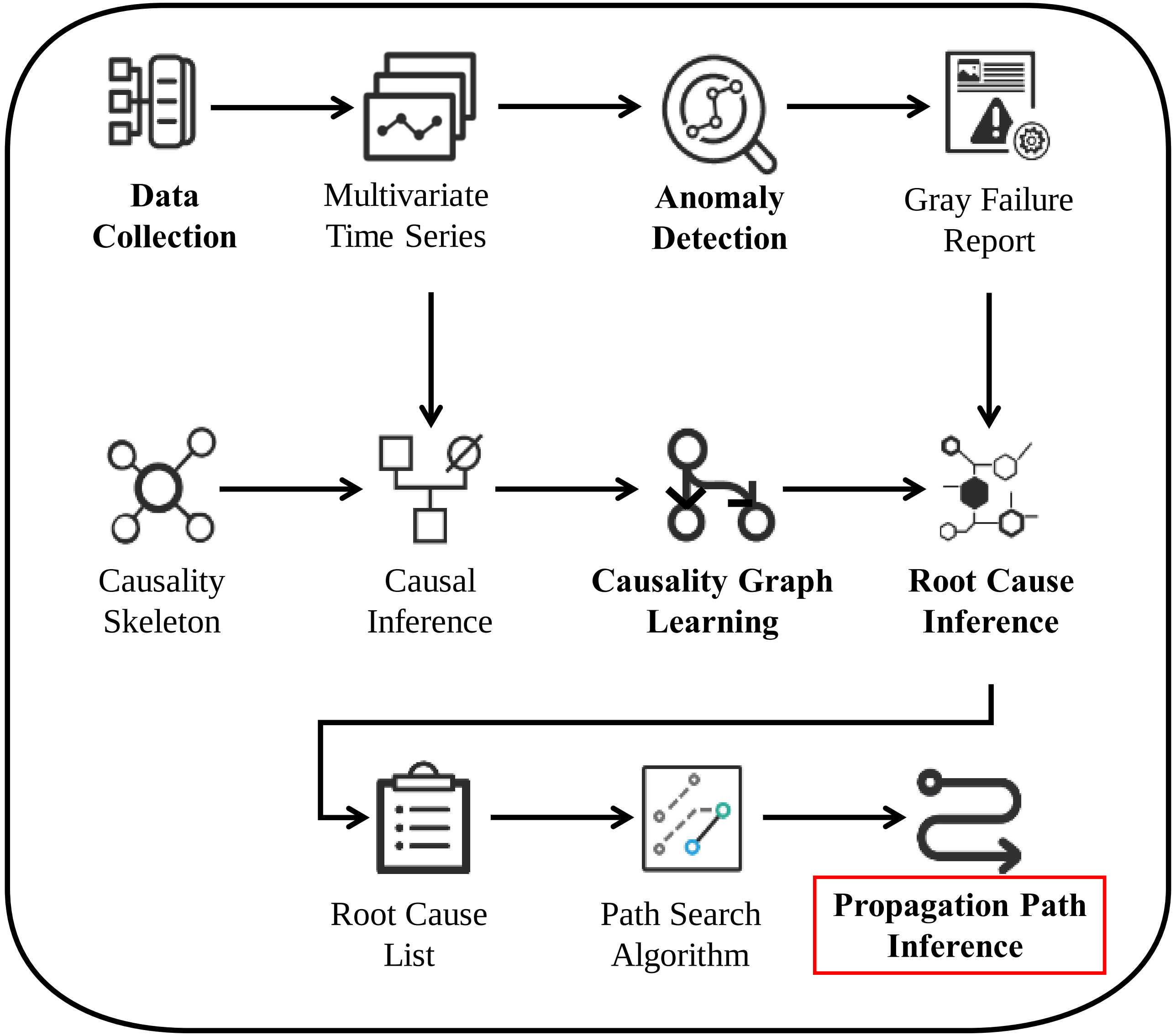
$$H'_{j,i} = \rho \cdot (\lambda \cdot correlation(v_i) + (1-\lambda) \cdot anomaly\_degree(v_i)) \quad (3)$$

- Self step (stay in the present metric):

$$H'_{j,j} = \max[0, H'_{j,j} - H'^{max}_{j,k}] \quad (4)$$

- Identifying root causes should prioritize metrics highly correlated with KPI.
- Root cause metrics usually exhibit anomalies during a gray failure.
- The random walk should consider the correlation between each metric and the anomalous KPI and each metric’s anomaly degree.

Propagation Path Inference



- Studying gray failure paths boosts operator confidence about results, reduces mitigation time, and improves system availability.
- Our goal is to deduce the gray failure propagation path from v_{root} to v_{KPI} .
- We aim to find the shortest path with the metrics' highest cumulative anomaly score as the propagation path.

01

Background

02

Design

03

Evaluation

04

Conclusion

Dataset

Dataset	#CPU Exhaustion	#Disk IO High Load	#Network Latency	#Network Packet Loss
GaussDB	0	78	62	83
Redis	0	196	46	32
Kafka	20	0	94	187
Tomcat	192	0	134	117

- We establish a cluster environment in Huawei, comprising five physical host machines and 11 virtual machines, and deploy four popular applications (GaussDB, Redis, Kafka, and Tomcat) across these server OSes. EulerOS is installed on each of these 16 machines.
- We use Chaosblade for gray failure simulation to simulate network latency, packet loss, disk IO high load, and CPU exhaustion.
- We inject 1241 gray failures, including 212 gray failures caused by CPU exhaustion, 274 caused by disk IO high load, 336 caused by network latency, and 419 caused by network packet loss.

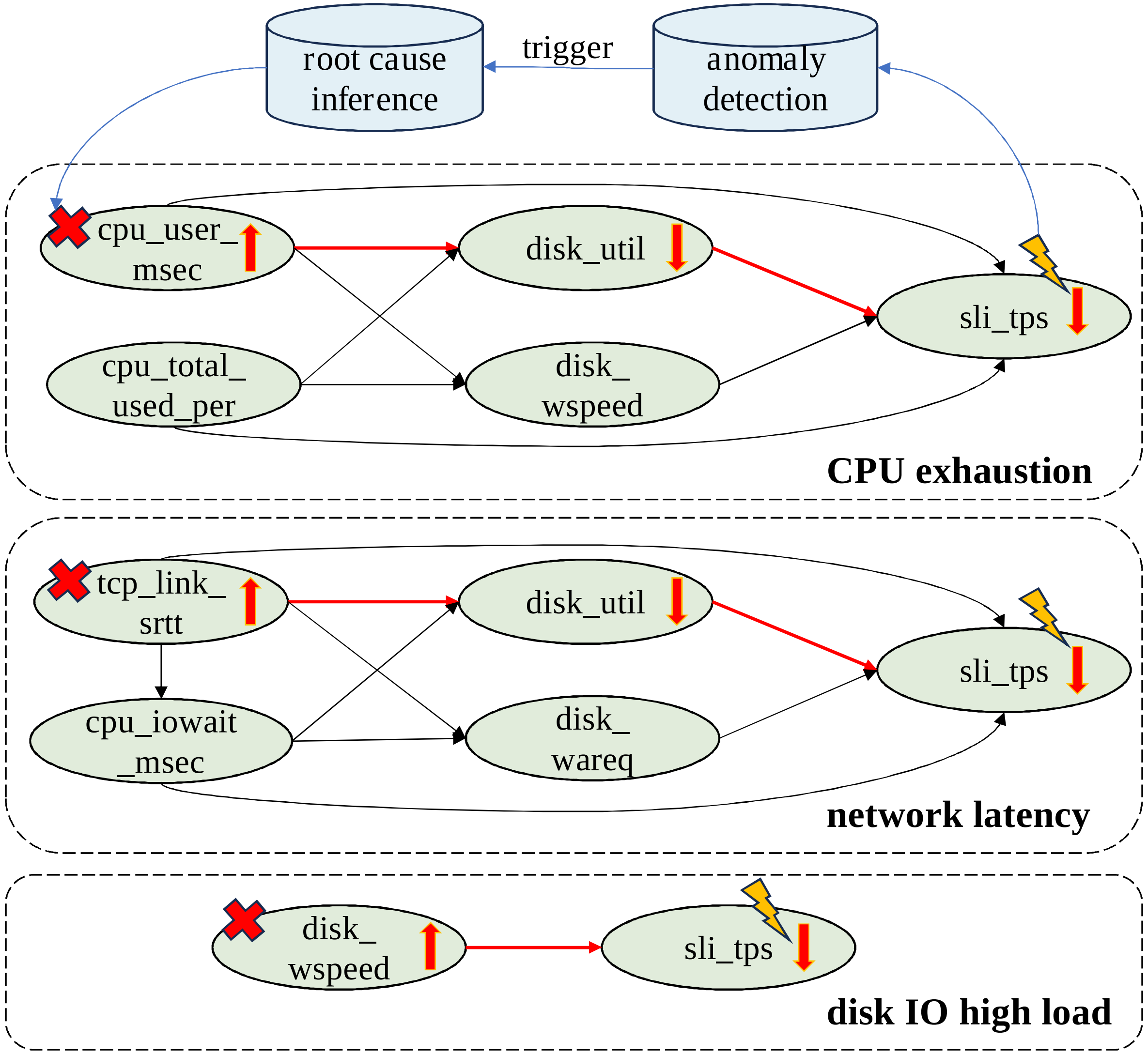
Effectiveness

Method	All			GaussDB			Redis			Kafka			Tomcat		
	AC@3	*AC@5*	*Avg@5*	*AC@3*	*AC@5*	*Avg@5*	*AC@3*	*AC@5*	*Avg@5*	*AC@3*	*AC@5*	*Avg@5*	*AC@3*	*AC@5*	*Avg@5*
GrayScope	**0.86**	**0.90**	**0.82**	**0.96**	**0.97**	**0.95**	**0.97**	**0.97**	**0.91**	**0.81**	**0.85**	**0.80**	**0.77**	**0.86**	**0.70**
CauseInfer [4]	0.23	0.25	0.21	0.39	0.41	0.37	0.42	0.49	0.40	0.14	0.15	0.12	0.09	0.10	0.08
MicroCause [30]	0.68	0.75	0.64	0.69	0.73	0.67	0.75	0.84	0.69	0.57	0.63	0.55	0.71	0.79	0.65
TS-InvarNet [13]	0.68	0.80	0.63	0.87	0.93	0.81	0.86	0.93	0.81	0.49	0.66	0.46	0.60	0.74	0.55
CIRCA [19]	0.51	0.64	0.50	0.74	0.83	0.73	0.92	0.95	0.88	0.39	0.57	0.38	0.21	0.39	0.22

outstanding performance on all scenarios

Compared with baseline methods, the results show that GrayScope is indeed effective in root cause localization.

Effectiveness



- We further evaluate the performance of GrayScope based on a dataset collected from the industrial environment of Huawei Cloud, denoted as $\mathcal{C}$.
- In 48 network latency cases, GrayScope's AC@3 reached 0.83; in 50 disk IO high load cases, the AC@3 achieved 0.98; in 37 high memory utilization cases, the AC@3 attained 0.94.
- It took GrayScope 6.97s to localize the root cause of each gray failure on average.

01

Background

02

Design

03

Evaluation

04

Conclusion

Conclusion

GrayScope: A Framework for Localizing Root Causes of Gray Failures

- Integrates expert knowledge with causal learning =>Learns reliable metric causal graphs
- Combines partial correlation with anomaly degree => Enhances the accuracy
- Recommends propagation paths => Enhances the interpretability
- Effectively and efficiently localize the root causes of gray failures in server OS

Opensource GrayScope

- https://gitee.com/ milohaha/grayscope

Thanks

Q&A