FluxInfer: Automatic Diagnosis of Performance Anomaly for Online Database System

Ping Liu[1], Shenglin Zhang[2], Yongqian Sun[2], Yuan Meng[1], Jiahai Yang[1,3], Dan Pei[1]

1 清華大學 Tsinghua University

2 南開大學 Nankai University

3 鹏城实验室 Peng Cheng Laboratory

Background

Design

Evaluation

Case Study

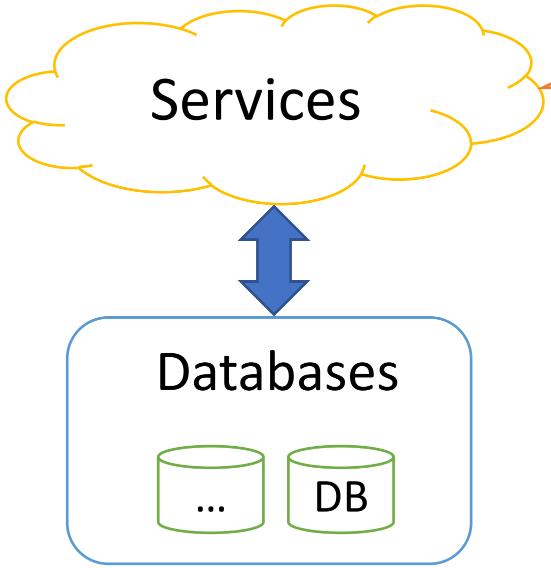
Background

Design

Evaluation

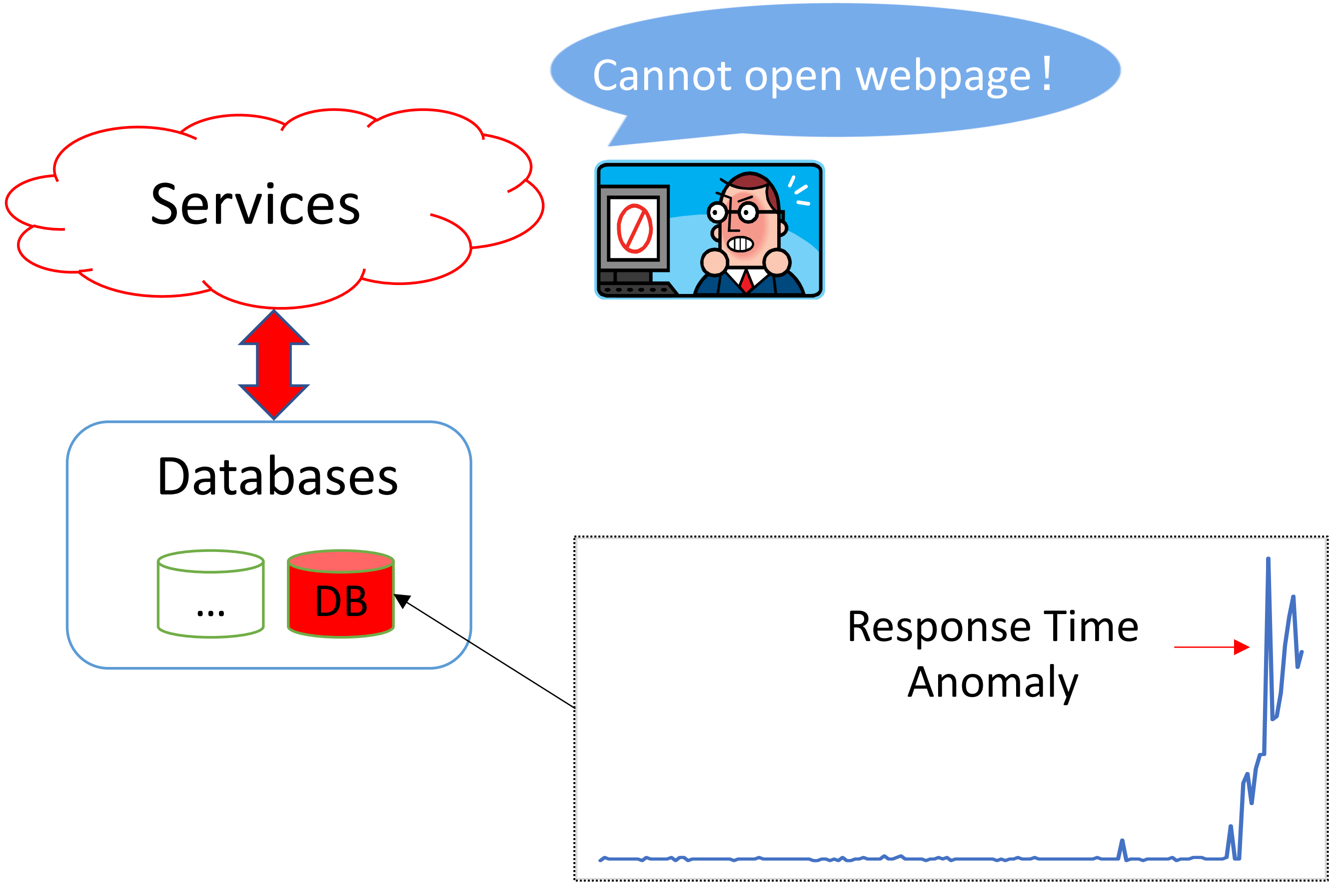
Case Study

Database anomaly



The services need databases to support their mission-critical and real-time applications

Database anomaly

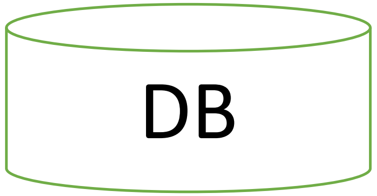


Database anomaly

How to diagnose the **anomalies**?

Anomaly diagnosis

- To rapidly diagnose anomalies and trigger mitigation, database operators monitor hundreds of KPIs (Key Performance Indicator) of the database.



Anomaly diagnosis

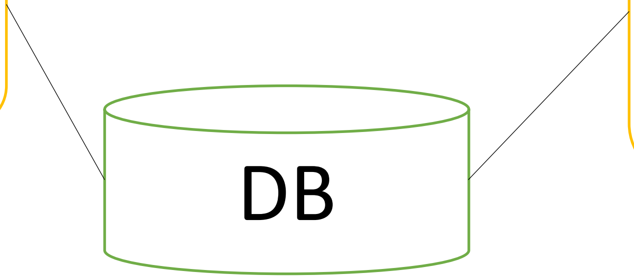
- To rapidly diagnose anomalies and trigger mitigation, database operators monitor hundreds of KPIs (Key Performance Indicator) of the database.

Anomaly diagnosis

- To rapidly diagnose anomalies and trigger mitigation, database operators monitor hundreds of KPIs (Key Performance Indicator) of the database.

Workload related KPIs

- Query requests: INSERT, UPDTE, DETETE, …



Storage-Engine-IO related KPIs

- **Physical IO**: The actual number of physical bytes involved in IO related operations.
- **Logic IO**: The actual number of virtual pages involved in IO related operations.

Anomaly diagnosis

- To rapidly diagnose anomalies and trigger mitigation, database operators monitor hundreds of KPIs (Key Performance Indicator) of the database.

Workload related KPIs

- Query requests: INSERT, UPDTE, DETETE, …

Resource related KPIs

- CPU, Memory, Network, …

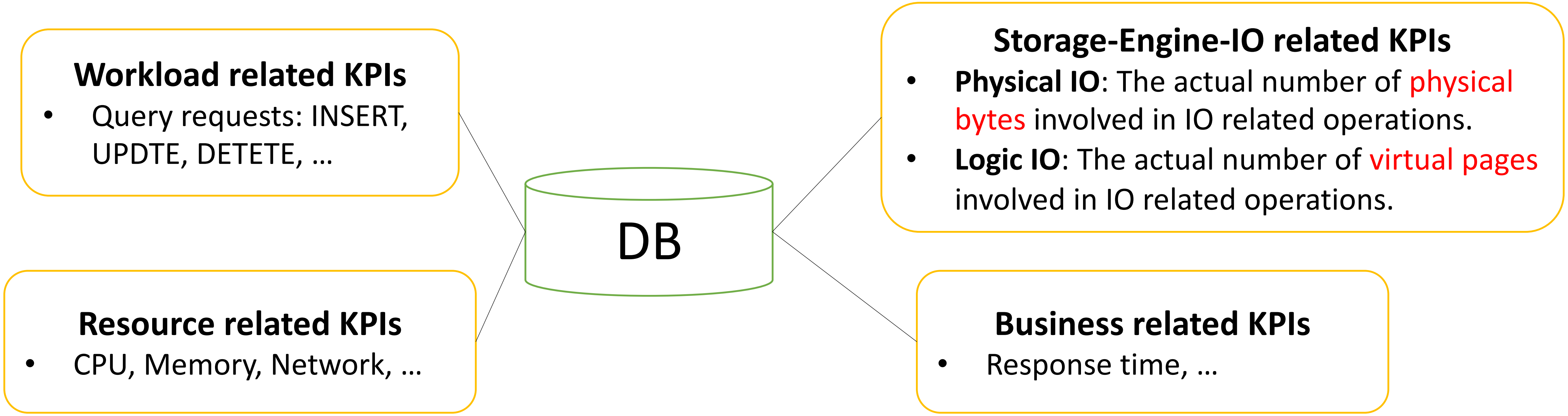
DB

Storage-Engine-IO related KPIs

- **Physical IO**: The actual number of physical bytes involved in IO related operations.
- **Logic IO**: The actual number of virtual pages involved in IO related operations.

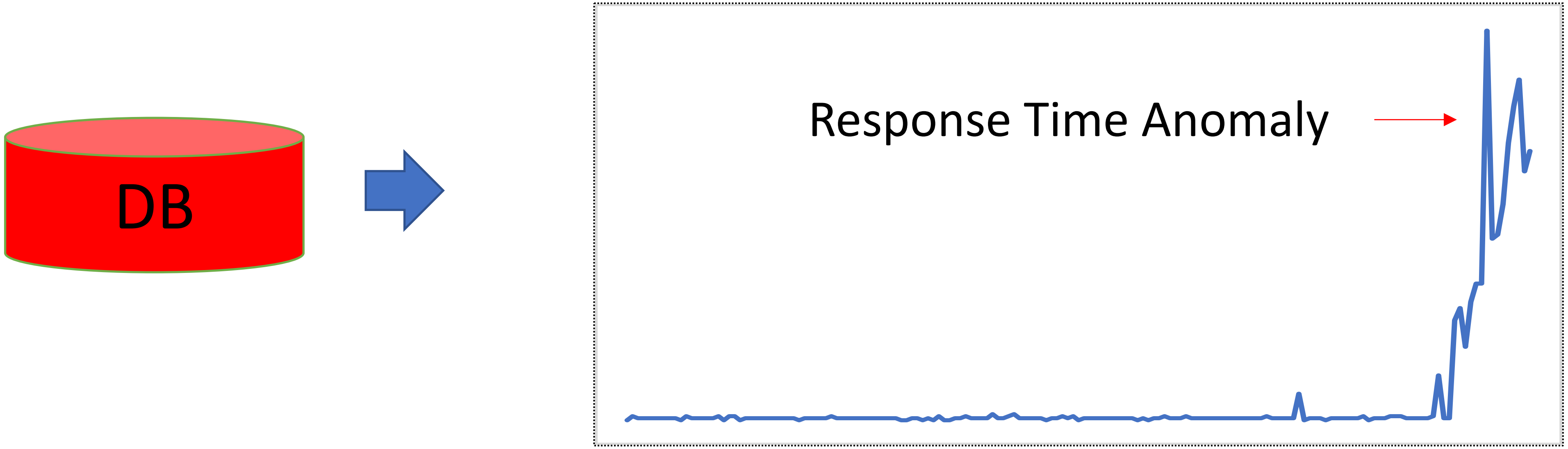
Anomaly diagnosis

- To rapidly diagnose anomalies and trigger mitigation, database operators monitor hundreds of KPIs (Key Performance Indicator) of the database.

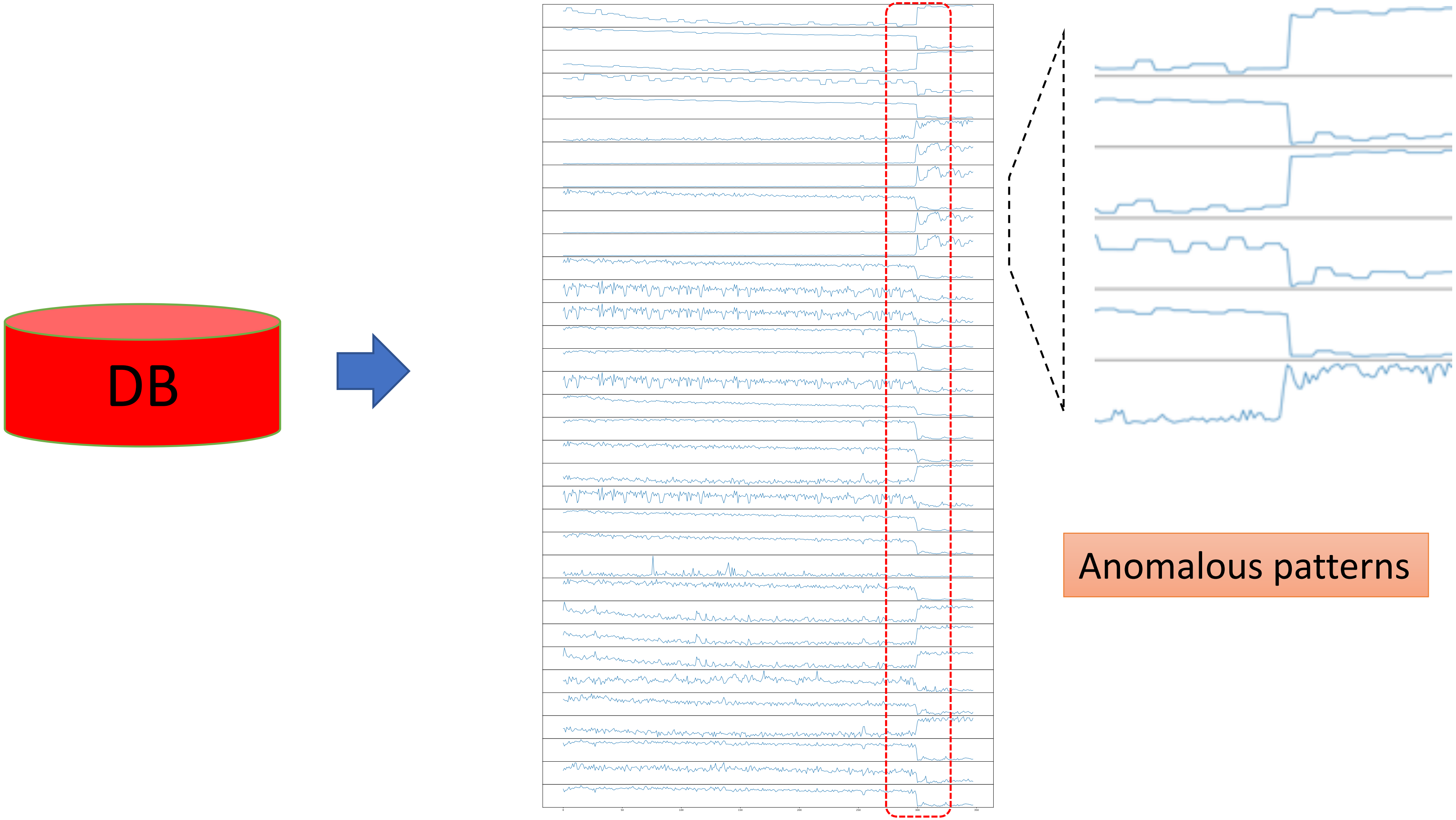


Anomaly diagnosis

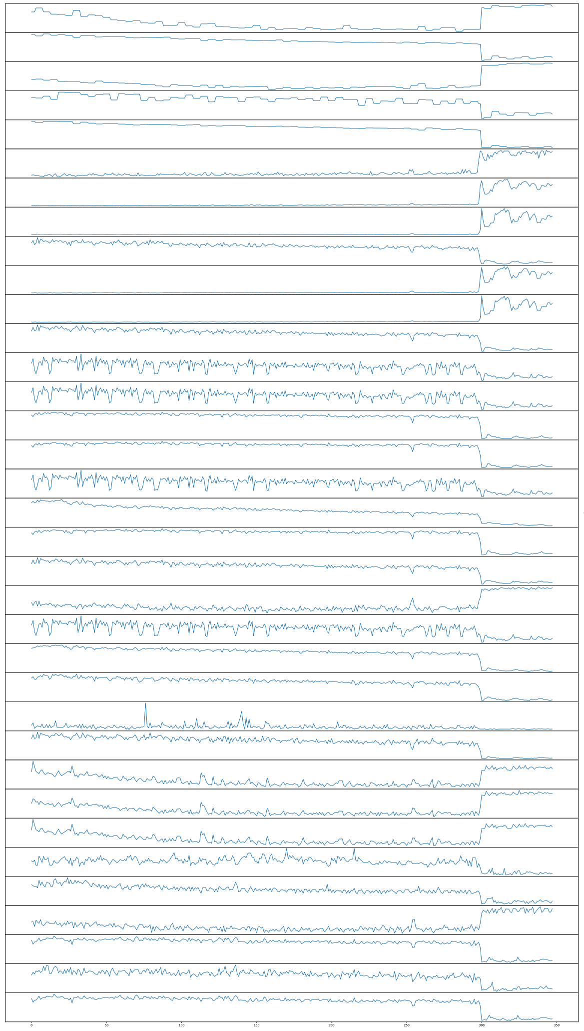
- When a database anomaly arises (response times are too slowly), some KPIs manifest anomalous patterns.



Anomaly diagnosis



Anomaly diagnosis



Root cause related KPI

- These KPIs indicate the root cause of the database anomaly and thus is closely related to the root cause

Symptom KPI

- These KPIs are not directly related to the root cause

Anomaly diagnosis

Root cause related KPI

- These KPIs indicate the root cause of the database anomaly and thus is closely related to the root cause

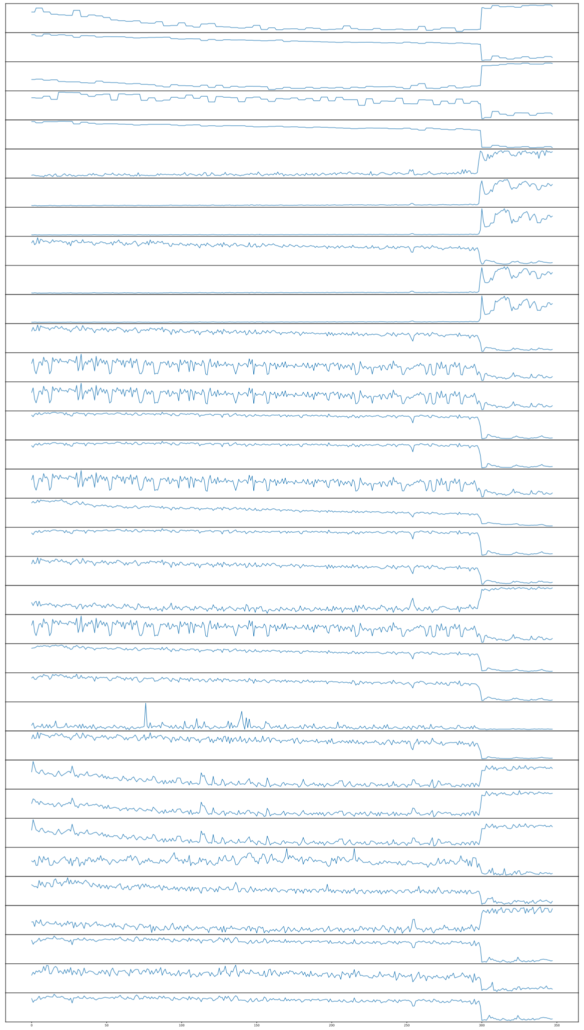
Symptom KPI

- These KPIs are not directly related to the root cause

Example

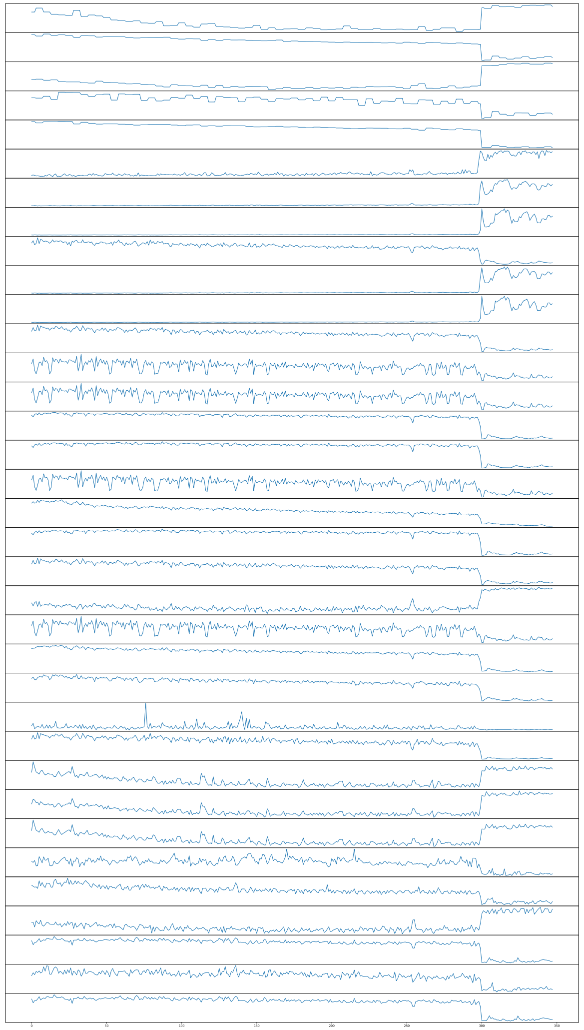
- A sudden increase in workload leads to a database anomaly.
- The QPS KPI, which represents the workload of the database, is a **root cause related KPI**.
- On the other hand, the increase in workload also leads to a sudden increase in CPU utilization, which is represented by the CPU_USAGE KPI. Consequently, CPU_USAGE KPI is a **symptom KPI**.

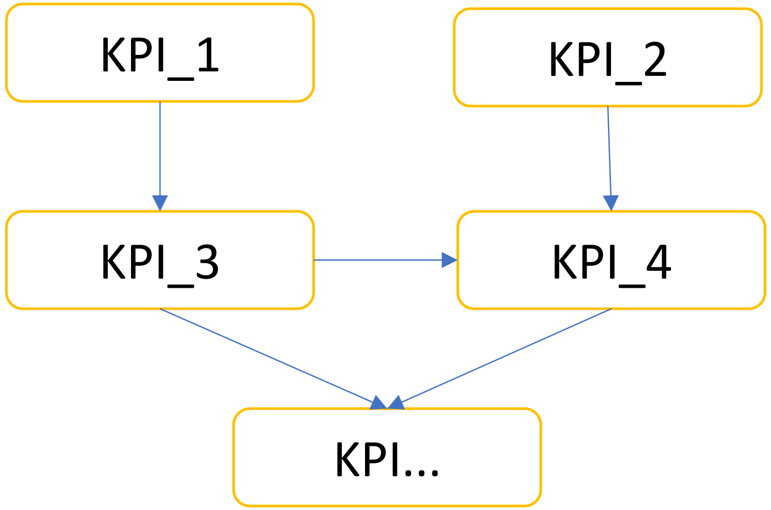
Anomaly diagnosis



Which are the root cause related KPIs

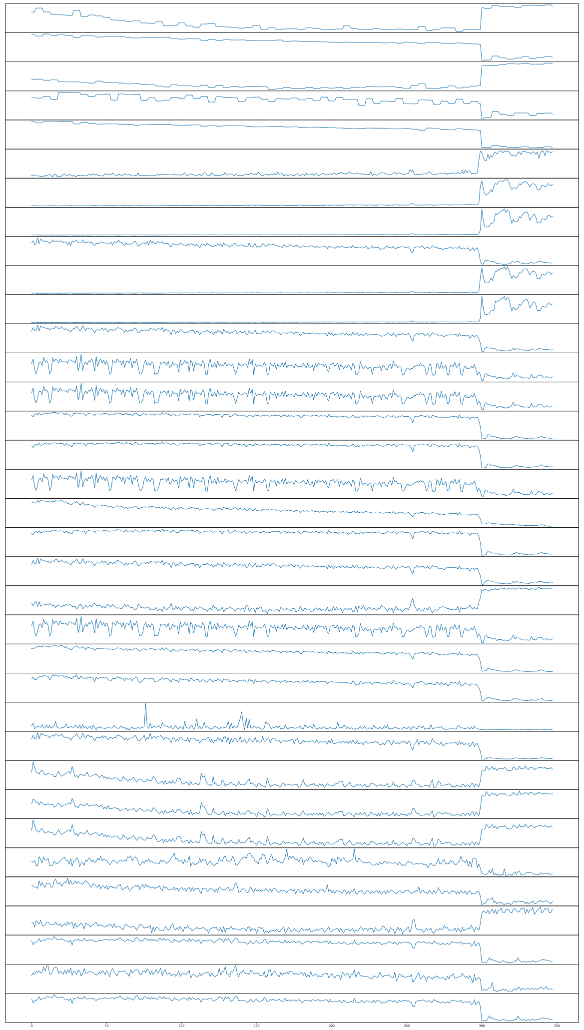
Anomaly diagnosis

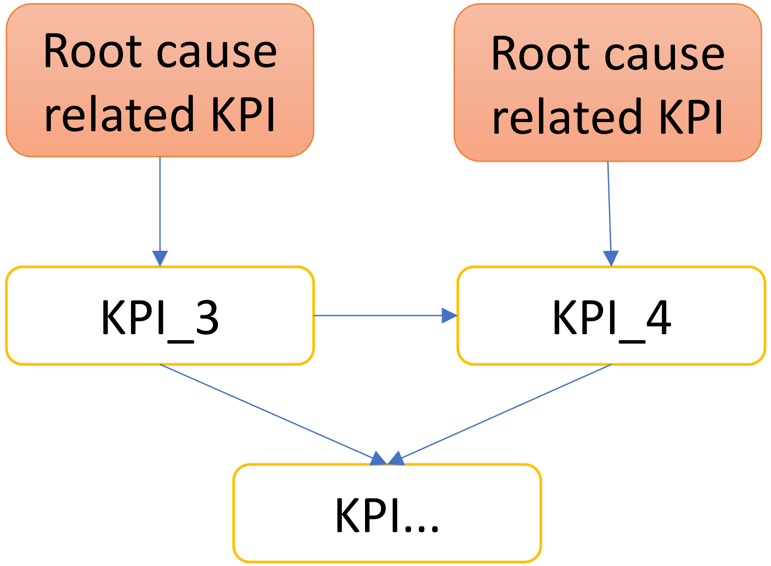




Operators **manually** infer the dependency relationships of these KPIs, which is highly dependent on the domain knowledge and experience

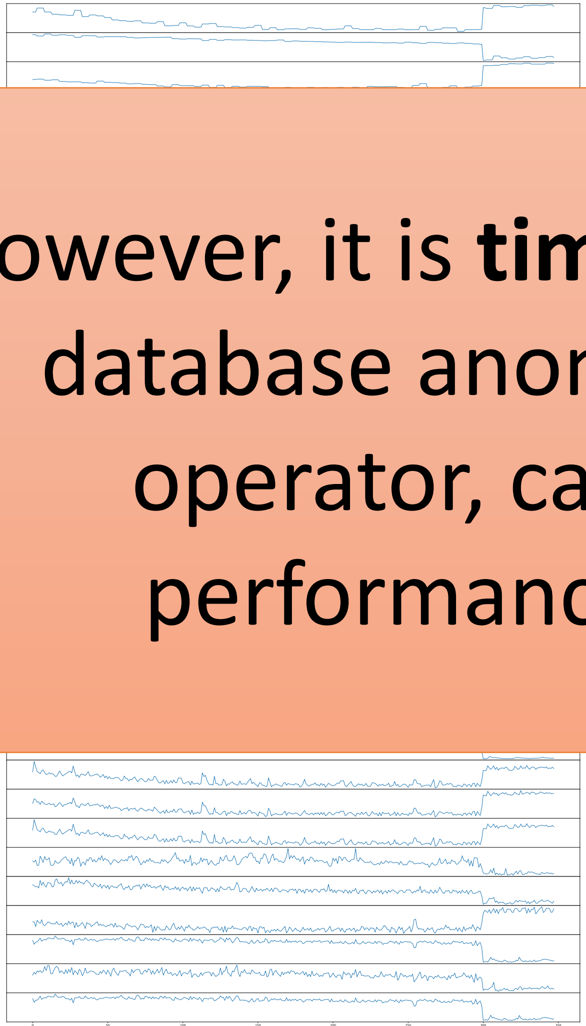
Anomaly diagnosis





Then localize the root cause related KPIs based on the dependency relationships

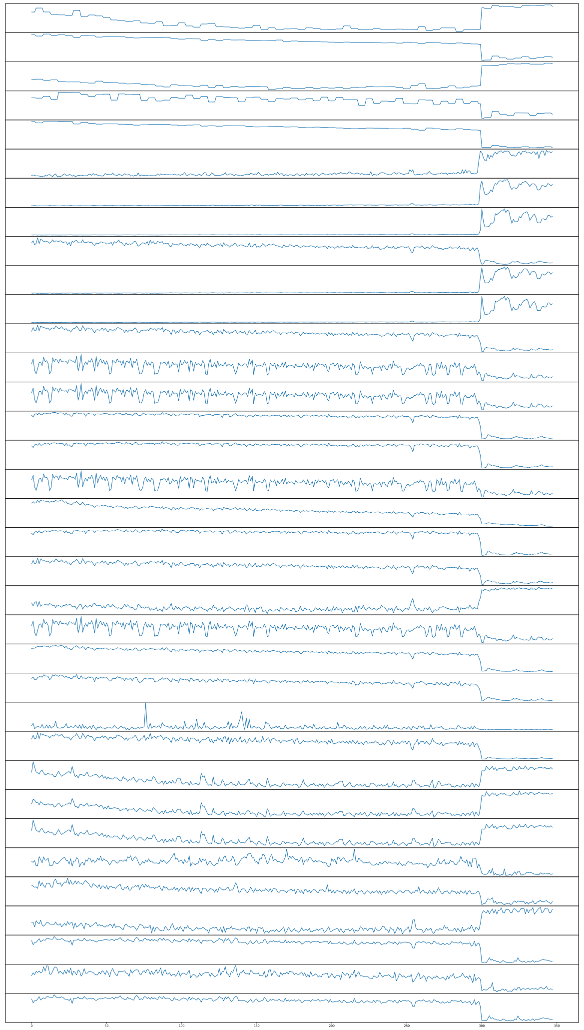
Anomaly diagnosis



However, it is **time-consuming** to manually diagnose a database anomaly even for a **highly experienced** operator, causing the services to suffer from performance degradation for a long period.

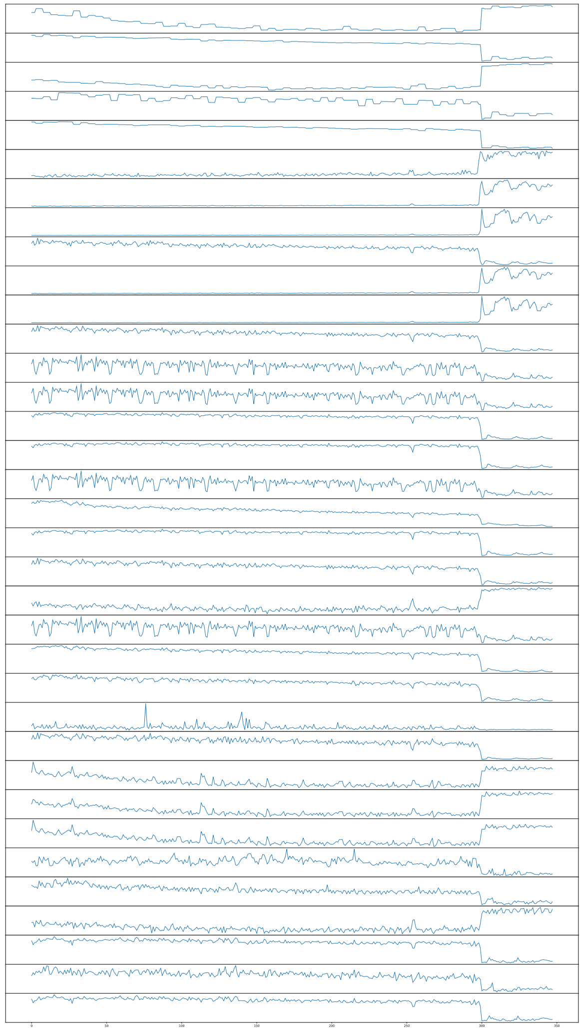
Then localize the root cause related KPIs based on the dependency relationships

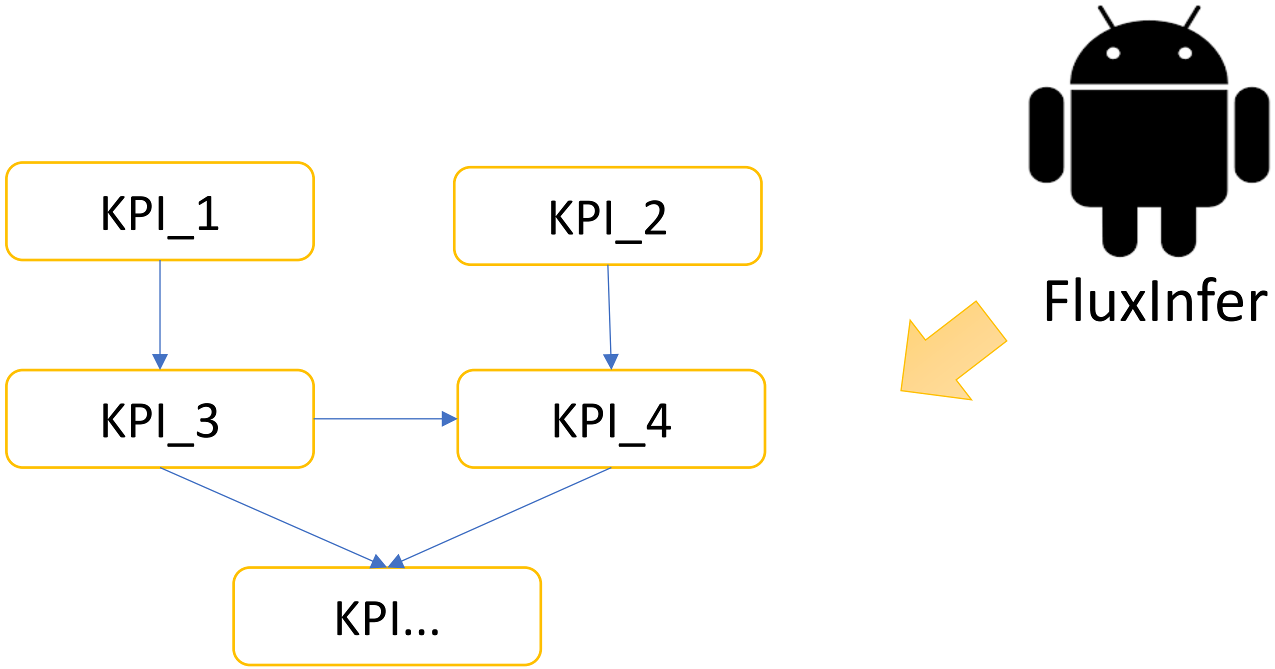
Core idea



FluxInfer

Core idea





FluxInfer can automatically construct the dependency relationships of anomalous KPIs accurately and localize root cause related KPIs

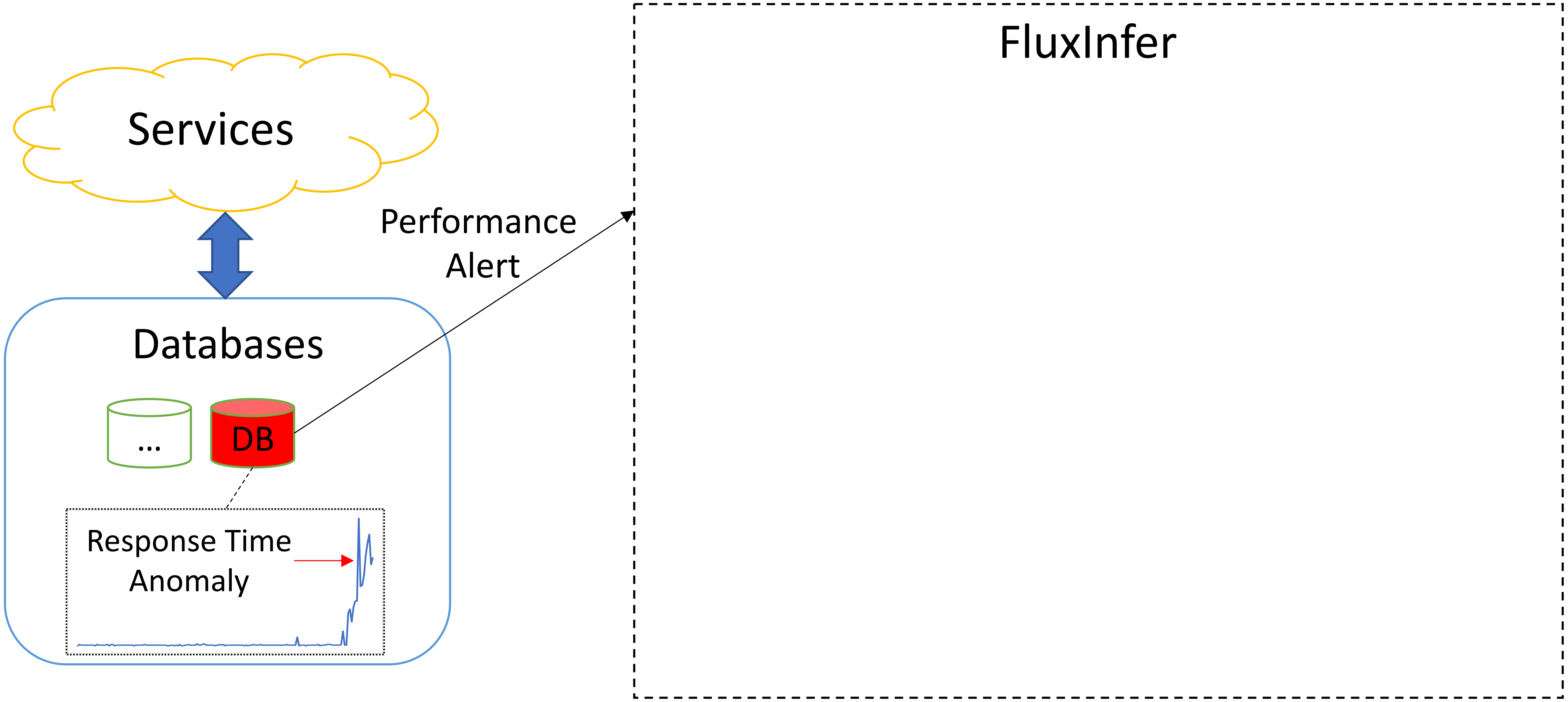
Background

Design

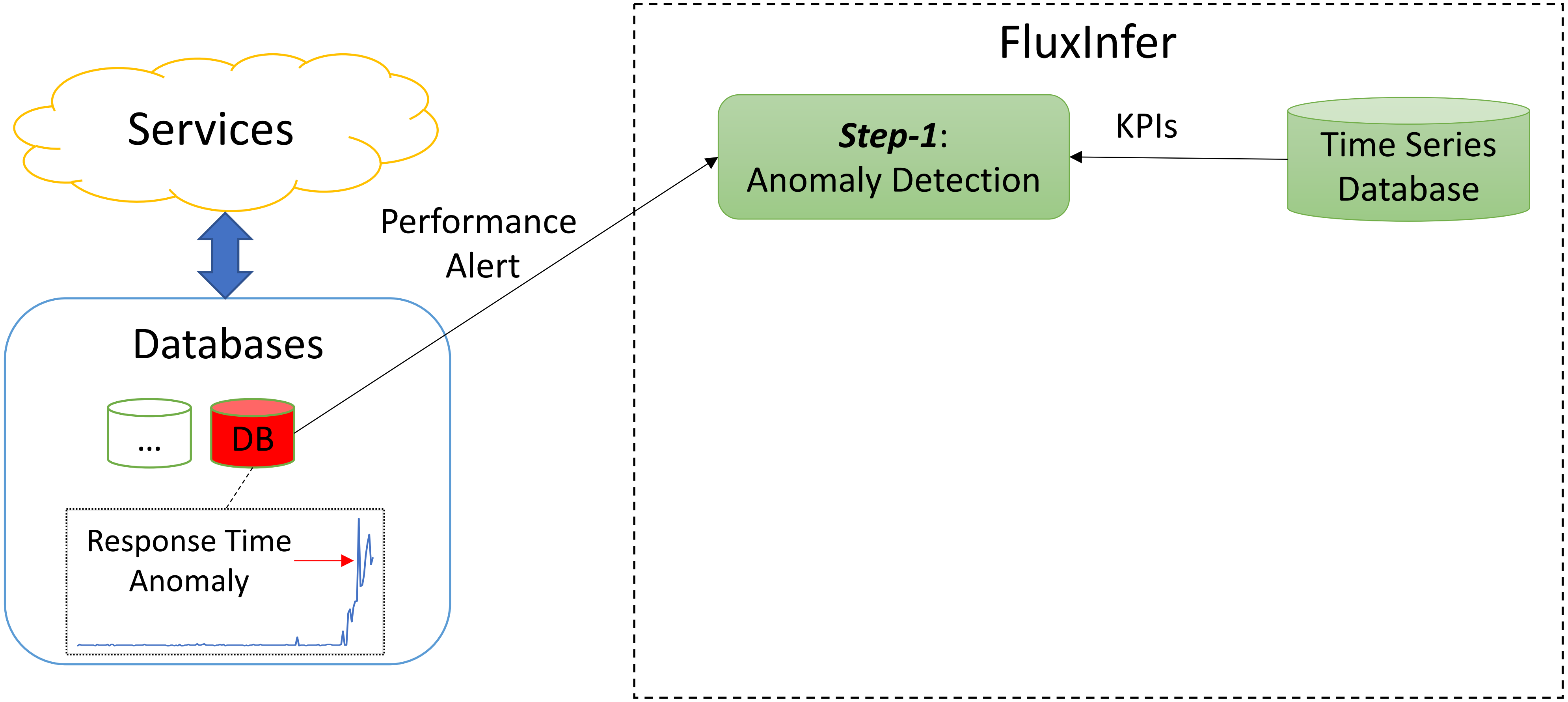
Evaluation

Case Study

Design

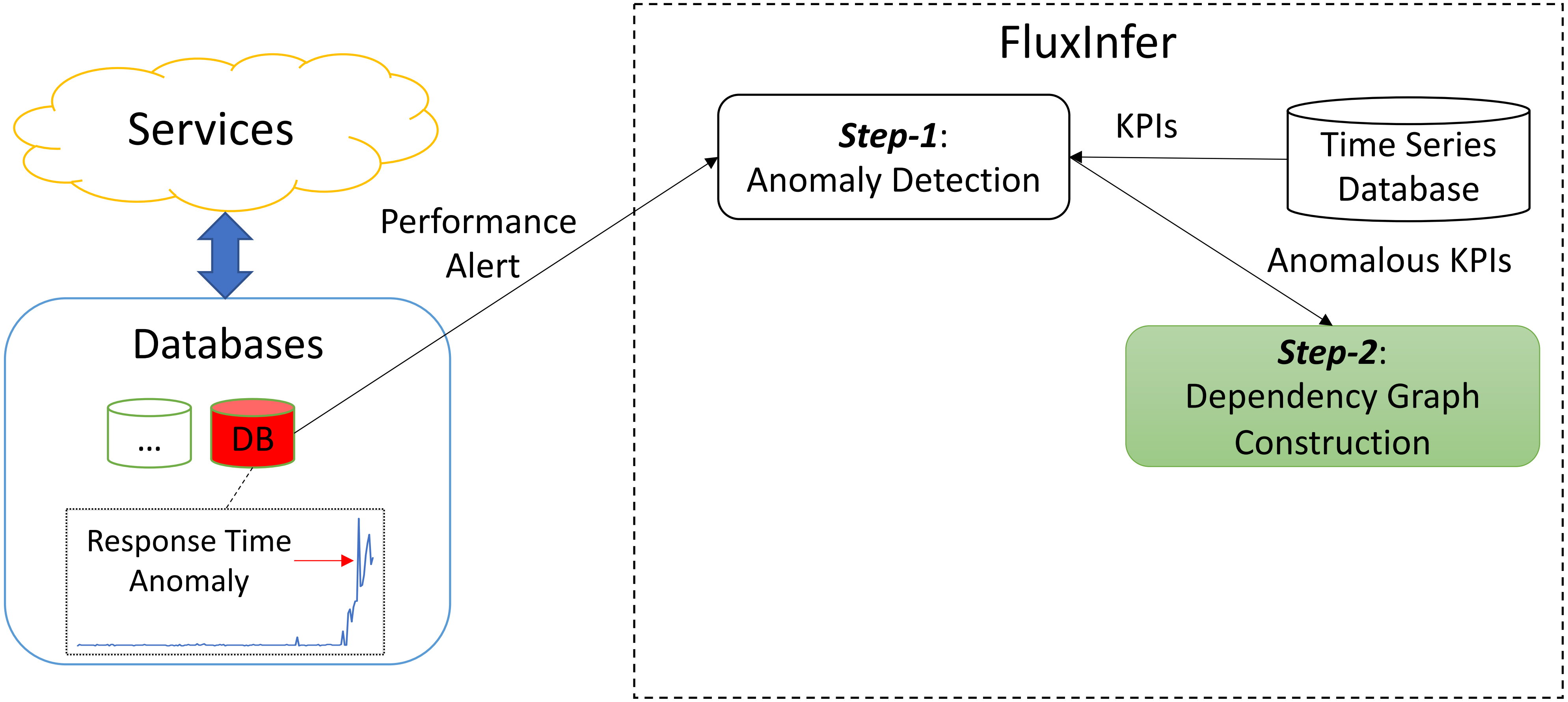

Services
Performance
Alert
Databases
...
DB
Response Time
Anomaly
FluxInfer

Design



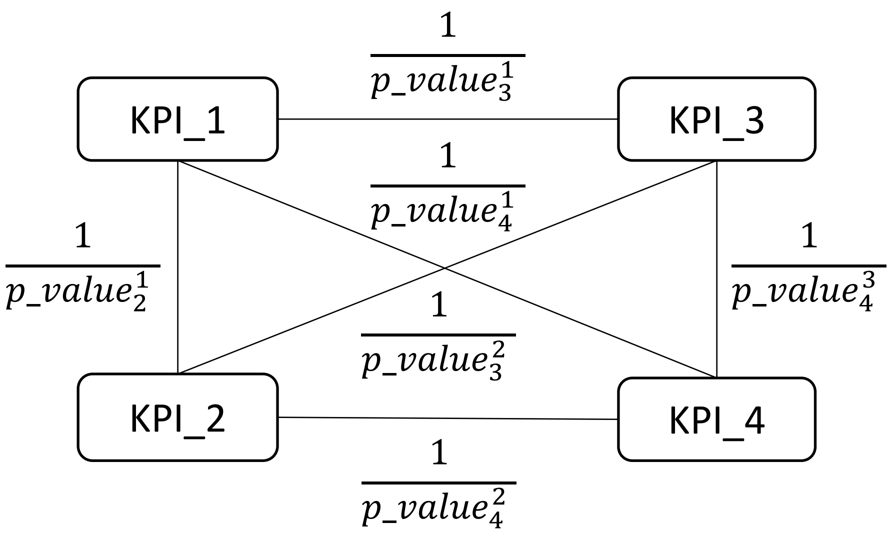
FluxInfer
Services
Performance Alert
Databases
...
DB
Response Time Anomaly
Step-1: Anomaly Detection
KPIs
Time Series Database

Design



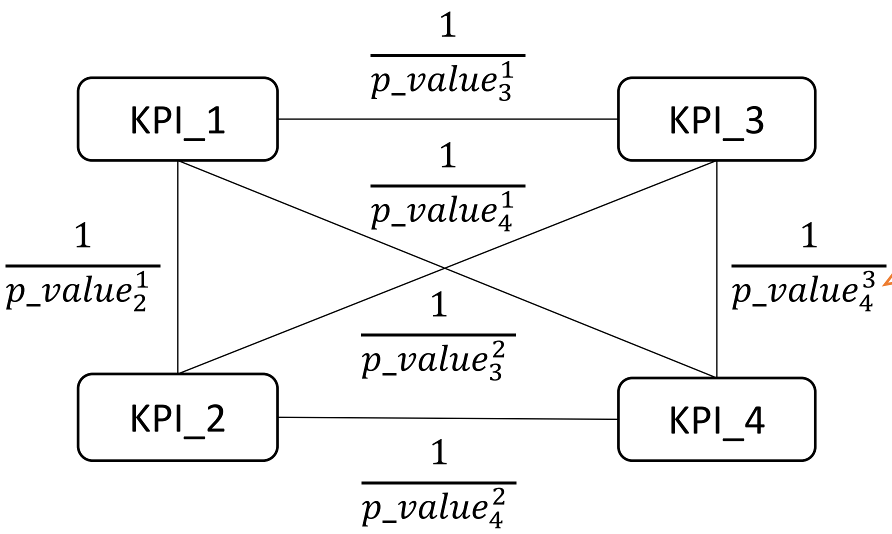
FluxInfer
Services
Performance
Alert
Databases
...
DB
Response Time
Anomaly
Step-1:
Anomaly Detection
KPIs
Time Series
Database
Anomalous KPIs
Step-2:
Dependency Graph
Construction

Dependency Graph



Weighted undirected dependency graph (WUDG)

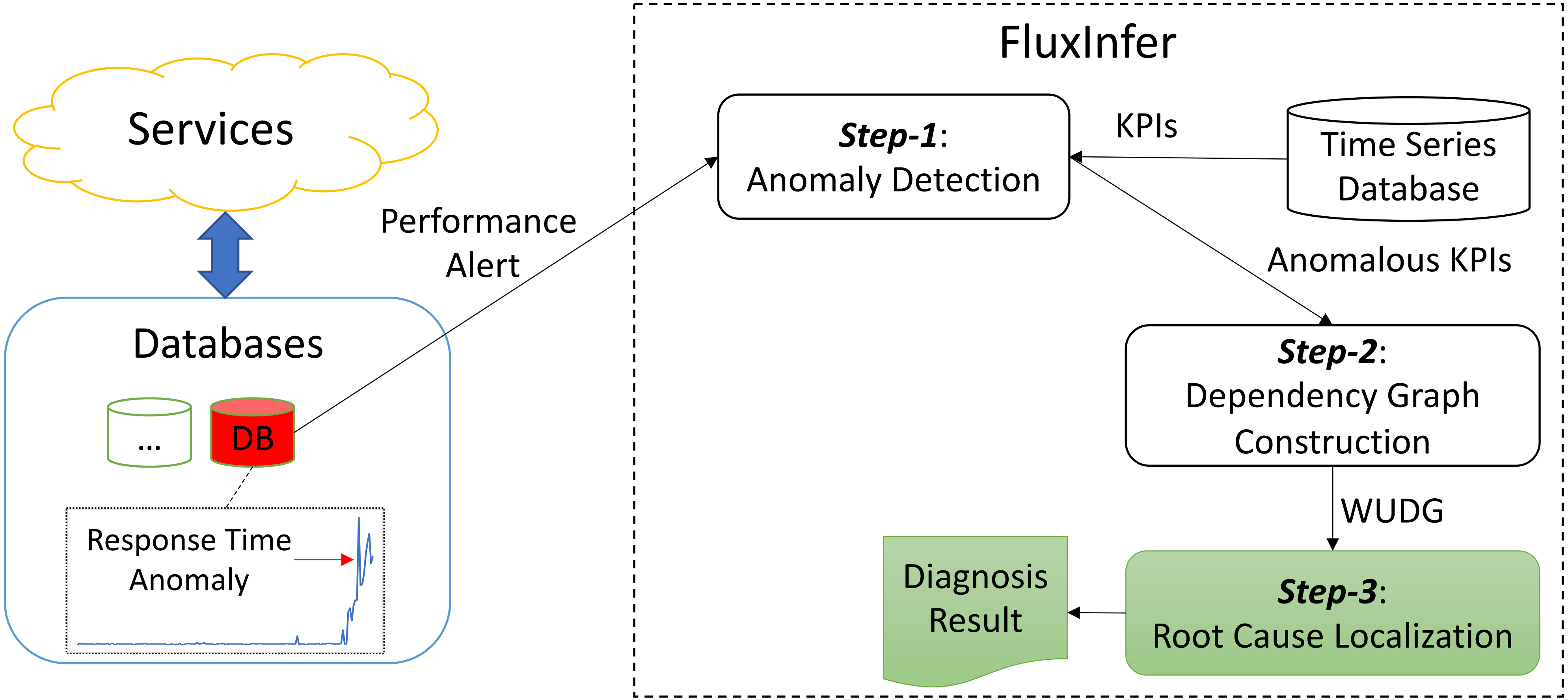
Dependency Graph



The strength of the dependency between two KPIs can be measured by: $\frac{1}{p\_value_4^3}$

Weighted undirected dependency graph (WUDG)

Design



FluxInfer
Services
Performance Alert
Databases
...
DB
Response Time Anomaly
Step-1: Anomaly Detection
KPIs
Time Series Database
Anomalous KPIs
Step-2: Dependency Graph Construction
WUDG
Step-3: Root Cause Localization
Diagnosis Result

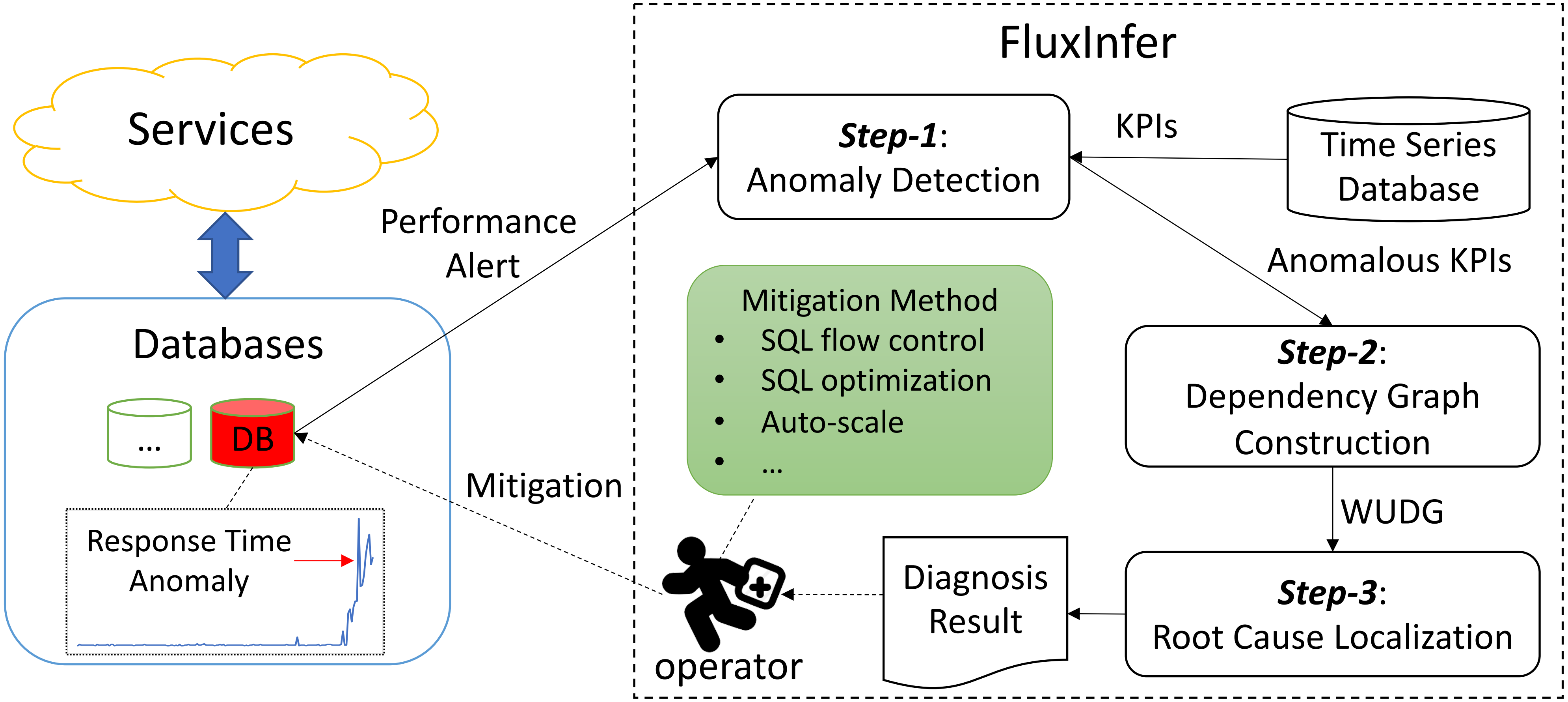
Root Cause Localization

- Due to that the root cause of a database system can quickly spread and lead to more and more anomalous KPIs, the root cause has the largest influence on the dependency graph among KPIs.

- Therefore, we suppose that **the root cause related KPI is the KPI who has the largest influence on the dependency graph**.

- The weighted PageRank algorithm are used to measure the influences of nodes in a weighted undirected graph, the possible root cause related KPIs are the KPIs ranked at the top.

Design



FluxInfer
Services
Databases
...
DB
Response Time
Anomaly
Performance
Alert
Mitigation
Step-1:
Anomaly Detection
KPIs
Time Series
Database
Anomalous KPIs
Mitigation Method
SQL flow control
SQL optimization
Auto-scale
...
Step-2:
Dependency Graph
Construction
WUDG
Step-3:
Root Cause Localization
Diagnosis
Result
operator

Background

Design

Evaluation

Case Study

Dataset

- We constructed a testbed to generate accurately labeled anomalies of database performance for evaluation.
- We injected five different types of anomalies: CPU saturation, network congestion, IO saturation, memory saturation, anomalous workload.

TABLE II: Details of injected 150 database anomalies for evaluation.

Type of anomaly	Number	Description
CPU Saturation	30	Invoke stress-ng, which starts N workers that perform various matrix operations on floating point values.
Network Congestion	30	Simulate network congestion by adding an artificial 500-milliseconds delay to every traffic over the network via Linux's tc (Traffic Control) command.
IO Saturation	30	Invoke stress-ng, which starts N workers that perform a mix of sequential, random and memory mapped read/write operations as well as forced sync'ing and cache dropping.
Memory Saturation	30	Invoke stress-ng, which starts N workers that grow their heaps by reallocating memory.
Anomalous workload	30	Greatly increase the rate of transactions and the number of clients simulated by OLTPBenchmark (150 additional terminals with transaction rate of 50,000).

Evaluation metric

- **AC@k** represents the accuracy that top K results include the root causes related KPIs for all anomaly cases.
- **Avg@k** represents the overall performance.

$$Avg@k = \frac{1}{k} \sum_{1 \leq j \leq k} AC@j.$$

Evaluation result

TABLE III: The evaluation results of different algorithms

Algorithm	Relationships Learning	Root Cause Inference	AC@1	AC@2	AC@3	AC@5	Avg@5
PAL [20]	N/A	Anomaly Time Order	0.09	0.12	0.14	0.20	0.14
CauseInfer [3]	PC Algorithm	Deep First Search	0.12	0.20	0.22	0.28	0.21
CloudRanger [4], MS-Rank [7]	PC Algorithm	Second-order Random Walk	0.08	0.19	0.27	0.36	0.24
Microscope [6]	PC Algorithm	Traversing+Pearson Correlation	0.06	0.11	0.16	0.24	0.15
MicroRCA [21]	PC Algorithm	Personalized PageRank	0.08	0.17	0.30	0.38	0.25
MonitorRank [18], TON18 [19]	PC Algorithm	Random Walk	0.08	0.16	0.28	0.39	0.24
MicroCause [5]	PCTS	TCORW	0.23	0.38	0.47	0.60	0.44
FluxInfer	**WUDG**	**Weighted PageRank**	**0.43**	**0.69**	**0.90**	**0.95**	**0.77**
FluxInfer-with-CUSUM	WUDG	Weighted PageRank	0.23	0.40	0.62	0.73	0.53
FluxInfer-without-AD	WUDG	Weighted PageRank	0.13	0.20	0.32	0.50	0.38

Evaluation result

TABLE III: The evaluation results of different algorithms

Algorithm	Relationships Learning	Root Cause Inference	AC@1	AC@2	AC@3	AC@5	Avg@5
PAL [20]	N/A	Anomaly Time Order	0.09	0.12	0.14	0.20	0.14
CauseInfer [3]	PC Algorithm	Deep First Search	0.12	0.20	0.22	0.28	0.21
CloudRanger [4], MS-Rank [7]	PC Algorithm	Second-order Random Walk	0.08	0.19	0.27	0.36	0.24
Microscope [6]	PC Algorithm	Traversing+Pearson Correlation	0.06	0.11	0.16	0.24	0.15
MicroRCA [21]	PC Algorithm	Personalized PageRank	0.08	0.17	0.30	0.38	0.25
MonitorRank [18], TON18 [19]	PC Algorithm	Random Walk	0.08	0.16	0.28	0.39	0.24
MicroCause [5]	PCTS	TCORW	0.23	0.38	0.47	0.60	0.44
FluxInfer	**WUDG**	**Weighted PageRank**	**0.43**	**0.69**	**0.90**	**0.95**	**0.77**
FluxInfer-with-CUSUM	WUDG	Weighted PageRank	0.23	0.40	0.62	0.73	0.53
FluxInfer-without-AD	WUDG	Weighted PageRank	0.13	0.20	0.32	0.50	0.38

Evaluation result

TABLE III: The evaluation results of different algorithms

Algorithm	Relationships Learning	Root Cause Inference	AC@1	AC@2	AC@3	AC@5	Avg@5
PAL [20]	N/A	Anomaly Time Order	0.09	0.12	0.14	0.20	0.14
CauseInfer [3]	PC Algorithm	Deep First Search	0.12	0.20	0.22	0.28	0.21
CloudRanger [4], MS-Rank [7]	PC Algorithm	Second-order Random Walk	0.08	0.19	0.27	0.36	0.24
Microscope [6]	PC Algorithm	Traversing+Pearson Correlation	0.06	0.11	0.16	0.24	0.15
MicroRCA [21]	PC Algorithm	Personalized PageRank	0.08	0.17	0.30	0.38	0.25
MonitorRank [18], TON18 [19]	PC Algorithm	Random Walk	0.08	0.16	0.28	0.39	0.24
MicroCause [5]	PCTS	TCORW	0.23	0.38	0.47	0.60	0.44
FluxInfer	**WUDG**	**Weighted PageRank**	**0.43**	**0.69**	**0.90**	**0.95**	**0.77**
FluxInfer-with-CUSUM	WUDG	Weighted PageRank	0.23	0.40	0.62	0.73	0.53
FluxInfer-without-AD	WUDG	Weighted PageRank	0.13	0.20	0.32	0.50	0.38

In FluxInfer-with-CUSUM, we replaced the robust anomaly detection algorithm of FluxInfer with the CUSUM algorithm

Evaluation result

TABLE III: The evaluation results of different algorithms

Algorithm	Relationships Learning	Root Cause Inference	AC@1	AC@2	AC@3	AC@5	Avg@5
PAL [20]	N/A	Anomaly Time Order	0.09	0.12	0.14	0.20	0.14
CauseInfer [3]	PC Algorithm	Deep First Search	0.12	0.20	0.22	0.28	0.21
CloudRanger [4], MS-Rank [7]	PC Algorithm	Second-order Random Walk	0.08	0.19	0.27	0.36	0.24
Microscope [6]	PC Algorithm	Traversing+Pearson Correlation	0.06	0.11	0.16	0.24	0.15
MicroRCA [21]	PC Algorithm	Personalized PageRank	0.08	0.17	0.30	0.38	0.25
MonitorRank [18], TON18 [19]	PC Algorithm	Random Walk	0.08	0.16	0.28	0.39	0.24
MicroCause [5]	PCTS	TCORW	0.23	0.38	0.47	0.60	0.44
FluxInfer	**WUDG**	**Weighted PageRank**	**0.43**	**0.69**	**0.90**	**0.95**	**0.77**
FluxInfer-with-CUSUM	WUDG	Weighted PageRank	0.23	0.40	0.62	0.73	0.53
FluxInfer-without-AD	WUDG	Weighted PageRank	0.13	0.20	0.32	0.50	0.38

- In FluxInfer-without-AD, we removed the robust anomaly detection algorithm of FluxInfer.
- All KPIs (normal KPIs and anomalous KPIs) are used to construct the dependency graph and localize the root cause.

Evaluation result

TABLE III: The evaluation results of different algorithms

Algorithm	Relationships Learning	Root Cause Inference	AC@1	AC@2	AC@3	AC@5	Avg@5
PAL [20]	N/A	Anomaly Time Order	0.09	0.12	0.14	0.20	0.14
CauseInfer [3]	PC Algorithm	Deep First Search	0.12	0.20	0.22	0.28	0.21
CloudRanger [4], MS-Rank [7]	PC Algorithm	Second-order Random Walk	0.08	0.19	0.27	0.36	0.24
Microscope [6]	PC Algorithm	Traversing+Pearson Correlation	0.06	0.11	0.16	0.24	0.15
MicroRCA [21]	PC Algorithm	Personalized PageRank	0.08	0.17	0.30	0.38	0.25
MonitorRank [18], TON18 [19]	PC Algorithm	Random Walk	0.08	0.16	0.28	0.39	0.24
MicroCause [5]	PCTS	TCORW	0.23	0.38	0.47	0.60	0.44
FluxInfer	**WUDG**	**Weighted PageRank**	**0.43**	**0.69**	**0.90**	**0.95**	**0.77**
FluxInfer-with-CUSUM	WUDG	Weighted PageRank	0.23	0.40	0.62	0.73	0.53
FluxInfer-without-AD	WUDG	Weighted PageRank	0.13	0.20	0.32	0.50	0.38

- The results of FluxInfer-with-CUSUM and FluxInfer-without-AD demonstrate the effectiveness of our robust anomaly detection design.

Evaluation result

TABLE III: The evaluation results of different algorithms

Algorithm	Relationships Learning	Root Cause Inference	AC@1	AC@2	AC@3	AC@5	Avg@5
PAL [20]	N/A	Anomaly Time Order	0.09	0.12	0.14	0.20	0.14
CauseInfer [3]	PC Algorithm	Deep First Search	0.12	0.20	0.22	0.28	0.21
CloudRanger [4], MS-Rank [7]	PC Algorithm	Second-order Random Walk	0.08	0.19	0.27	0.36	0.24
Microscope [6]	PC Algorithm	Traversing+Pearson Correlation	0.06	0.11	0.16	0.24	0.15
MicroRCA [21]	PC Algorithm	Personalized PageRank	0.08	0.17	0.30	0.38	0.25
MonitorRank [18], TON18 [19]	PC Algorithm	Random Walk	0.08	0.16	0.28	0.39	0.24
MicroCause [5]	PCTS	TCORW	0.23	0.38	0.47	0.60	0.44
FluxInfer	**WUDG**	**Weighted PageRank**	**0.43**	**0.69**	**0.90**	**0.95**	**0.77**
FluxInfer-with-CUSUM	WUDG	Weighted PageRank	0.23	0.40	0.62	0.73	0.53
FluxInfer-without-AD	WUDG	Weighted PageRank	0.13	0.20	0.32	0.50	0.38

- Further, we can see that the performance of FluxInfer-with-CUSUM is better than that of FluxInfer-without-AD, which demonstrates that anomaly detection is necessary for the design of FluxInfer.

Background

Design

Evaluation

Case Study

Case study

- The diagnosis results of FluxInfer and MicroCause baseline for a CPU saturation case. The bold red font represents the root cause related KPI.

FluxInfer

1 cpu.usage
2 cpu.usage_percent
3 mysql.mem_used
4 docker.net_recv
5 docker.net_send
6 docker.io_write
7 mysql.io_bytes_write
8 mem.usage_percent
…

MicroCause

1 mysql.qps
2 mysql.innodb_rows_read
3 mysql.active_session
4 docker.io_write
5 mem.usage_percent
6 mysql.io_bytes_write
7 cpu.usage
8 mysql.bytes_sent
…

Case study

- The diagnosis results of FluxInfer and MicroCause baseline for a CPU saturation case. The bold red font represents the root cause related KPI.

- The diagnosis result of FluxInfer shows that the root cause related KPIs are ranked top 1 and top 2, respectively.
- However, the root cause related KPI is ranked top 7 in the diagnosis result of MicroCause baseline.
- The other eight baselines even cannot rank the root cause related KPIs in top 10.

Conclusion

- This paper presents FluxInfer, which can automatically and accurately localize the root cause related KPIs of online database performance anomalies.
- We propose an algorithm to automatically construct a Weighted Undirected Dependency Graph (WUDG) to accurately represent the dependency relationships of anomalous KPIs.
- We propose to use a weighted PageRank algorithm to traverse WUDG, which can accurately localize root cause related KPIs.
- Detailed evaluation experiments on our testbed show that the AC@3, AC@5, and Avg@5 of FluxInfer are 0.90, 0.95, and 0.77, outperforming nine baselines by 64%, 60%, and 53% on average, respectively.

Thank you!

Q&A

liuping15@mails.tsinghua.edu.cn

IPCCC 2020