中山大學
SUN YAT-SEN UNIVERSITY

TraStrainer: Adaptive Sampling for Distributed Traces with System Runtime State

Haiyu Huang*, Xiaoyu Zhang†, Pengfei Chen*,
Zilong He*, Zhiming Chen*, Guangba Yu*, Honyang Chen* and Chen Sun†

*Sun Yat-sen University

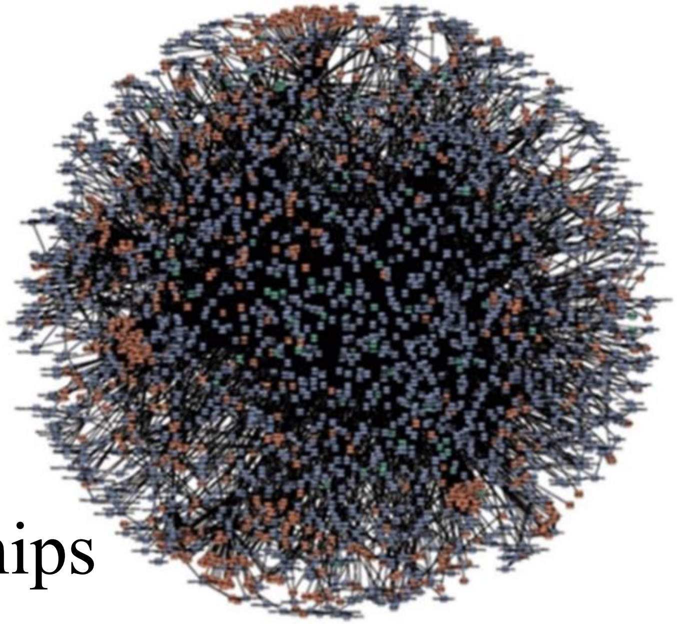
†Huawei

Introduction

➢ Microservice Systems

- Many single-concerned, loosely-coupled services

- Many complex call relationships (RPCs or RESTful APIs)



amazon.com

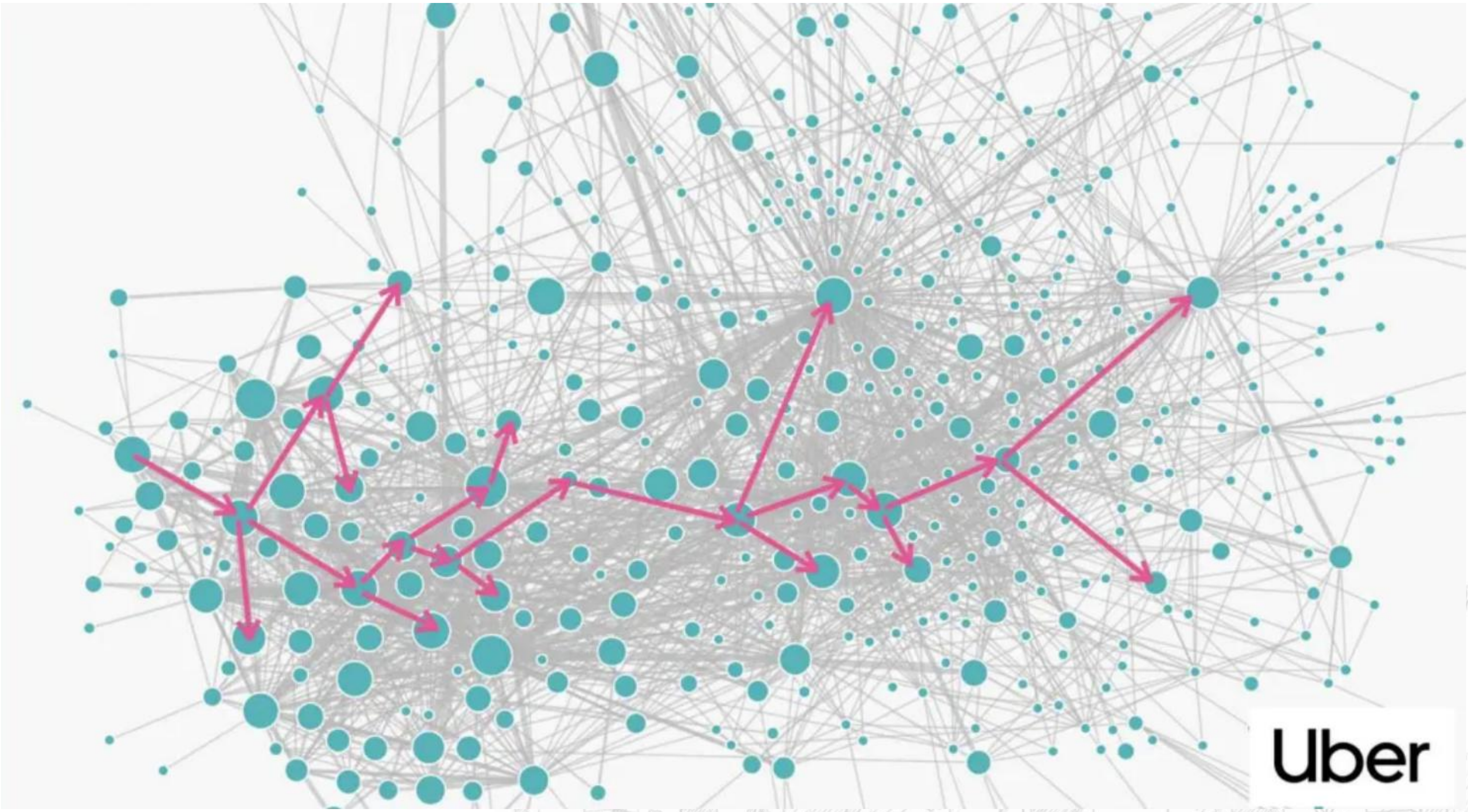
NETFLIX

Introduction

➢ Distributed Tracing

- Visualizes the end-to-end paths of requests through services

- Tracing Frameworks: Jaeger, OpenTelemetry, Zipkin, etc.



Introduction

➢ Traces are helpful for analysis

- Traces play a crucial role in analysis the systems.

Trace-Based Analysis

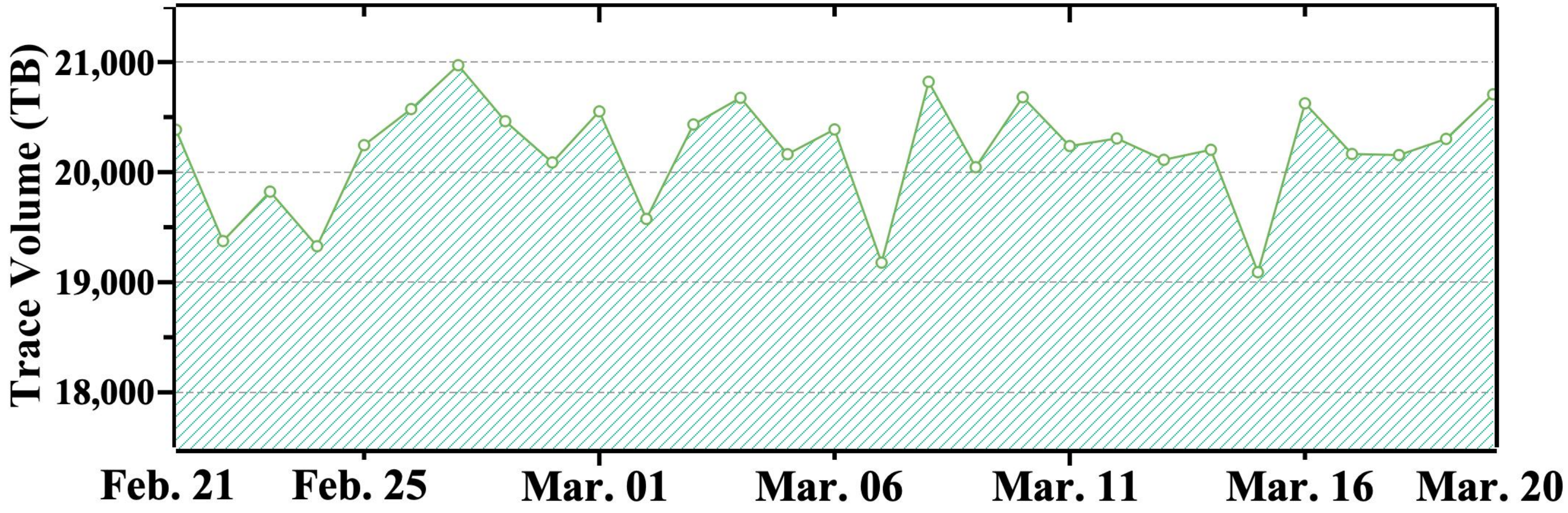
- System Profiling
- Anomaly Detection
- Root Cause Analysis

Introduction

➢ Too many traces

The quantities and storage costs of traces are often very high.

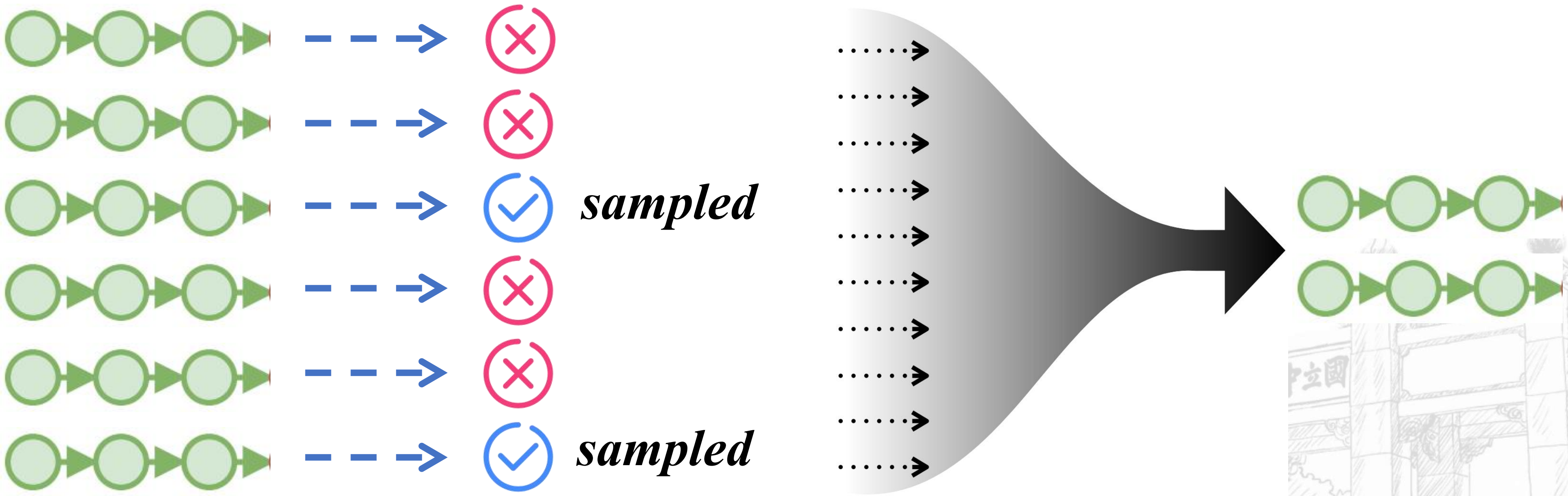
■ Impractical to retain and analyze all traces



Introduction

➢Trace Sampling

Trace sampling aims to retain only a portion of traces.



Introduction

➢ Head Sampling vs. Biased Sampling

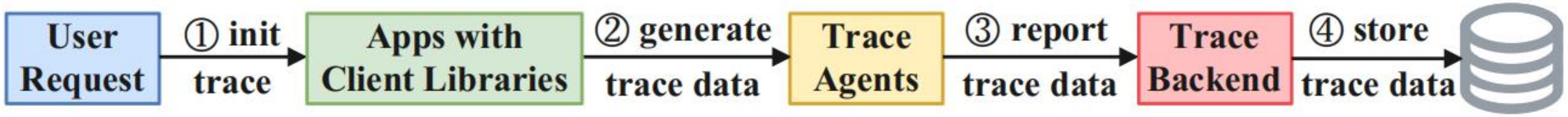


Figure The lifecycle of a trace.

Head Sampling:
- Randomly sampling
- Sample at the beginning

Biased Sampling:
- Sample based on certain policies
- Sample at the end

Introduction

➢ Head Sampling vs. Biased Sampling

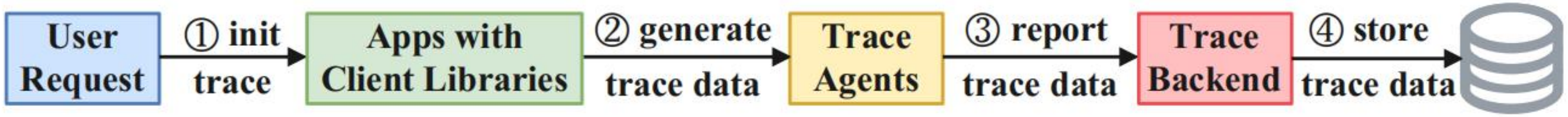


Figure The lifecycle of a trace.

Head Sampling:

- Randomly sampling
- Sample at the beginning

- Unable to retian high-quality trace information

Introduction

➢ Head Sampling vs. Biased Sampling

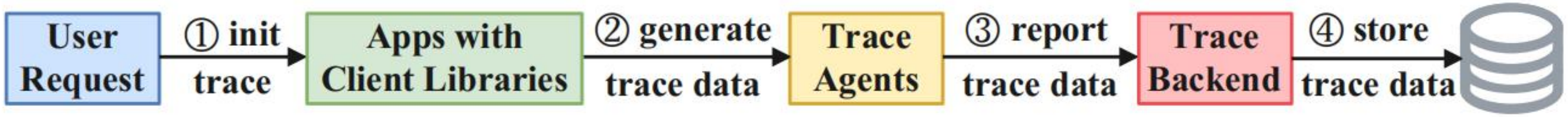
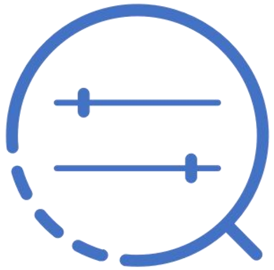


Figure The lifecycle of a trace.

- Decide whether to sample based on trace characteristics.



Biased Sampling:

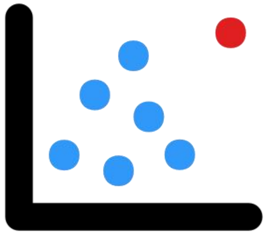
- Sample based on certain policies
- Sample at the end

Motivations

➢ Limitation of Previous Work

Previous biased sampling policies only focus on ‘edge-cases’.

- Edge-cases: rare or symptomatic traces

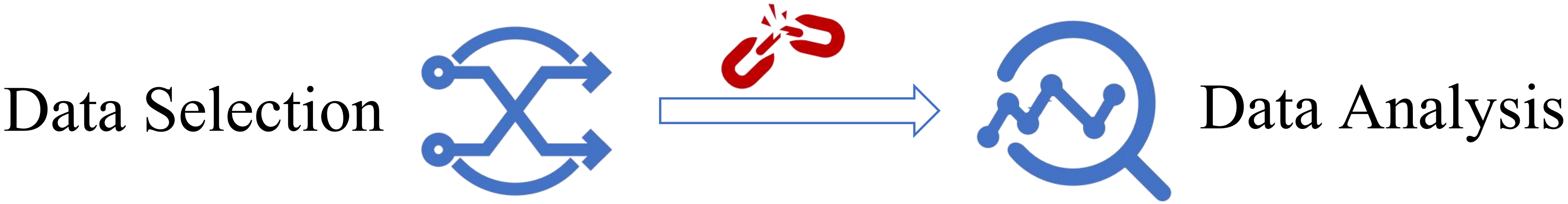


- Core: keep more edge-case traces and fewer common-case traces.

Motivations

➢ Limitation of Previous Work

Previous methods did not generate truly high-quality traces for downstream trace-based analysis.



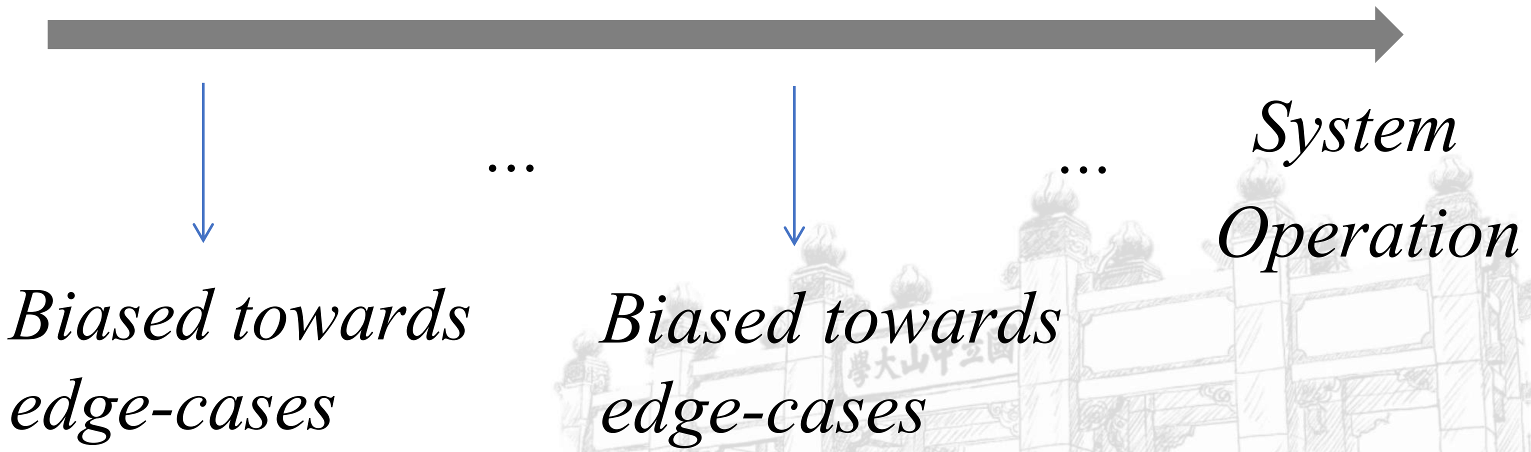
Sampling Approach	A@1			A@3			MRR		
	0.1%	**1.0%**	**2.5%**	**0.1%**	**1.0%**	**2.5%**	**0.1%**	**1.0%**	**2.5%**
Random	10.71	9.26	16.67	44.73	57.41	57.41	0.2820	0.3503	0.3997
HC	9.26	12.96	18.52	37.04	42.59	55.56	0.2590	0.3664	0.3747
Sifter	11.67	24.07	16.67	37.04	57.41	62.96	0.2753	0.4025	0.4145
Sieve	8.81	18.52	29.63	44.44	53.70	57.41	0.2620	0.3762	0.4383

Motivations

➢ Limitation of Previous Work

Ignoring the crucial context of the system's runtime state.

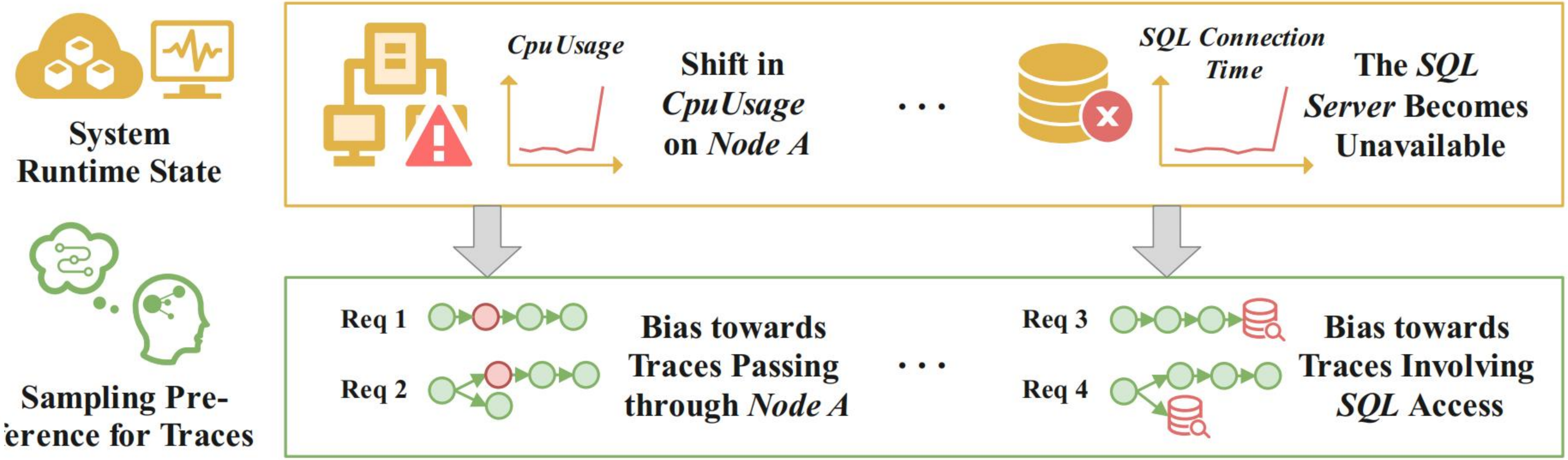
- Solely focus on traces themselves
- Symptoms (edge-cases) and root causes can be far apart



Motivations

➢ Insights

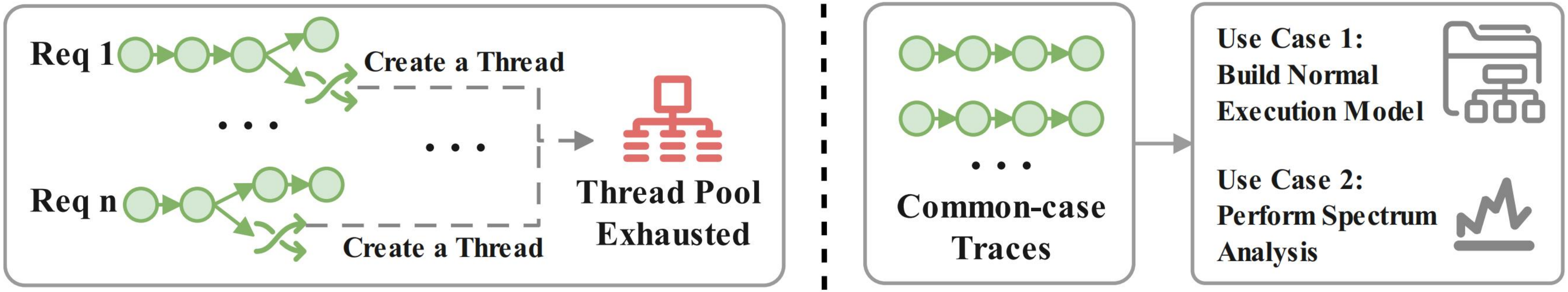
The characteristics of a “valuable trace” also change when the system state changes



Motivations

➢ Insights

Problem-related common-case traces are also helpful in downstream analysis.



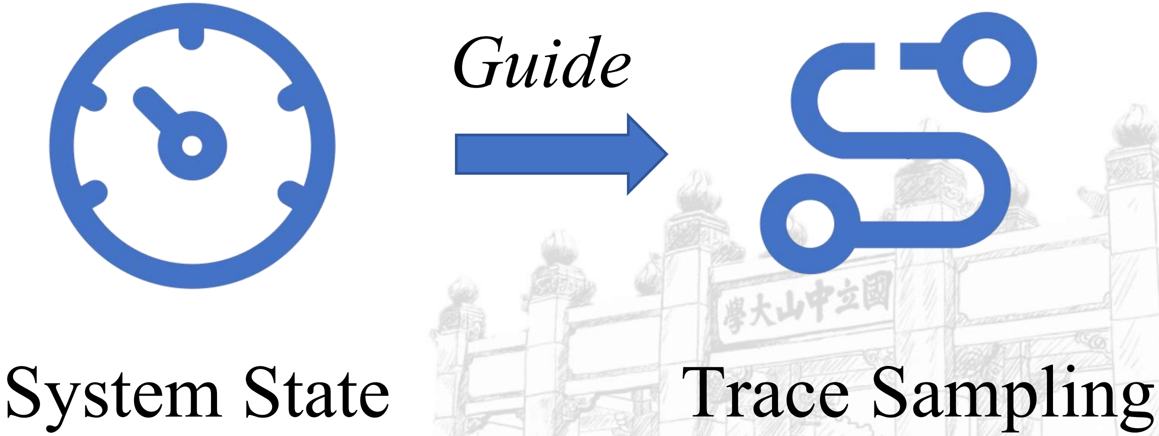
(a) Common-case traces can be related to issues

(b) Common-case traces hold analytical value in downstream analysis algorithms

Motivations

➢ Let's think outside the box

We can consider not just the traces themselves, but also the system's runtime state.



Motivations

➢ A real-world case

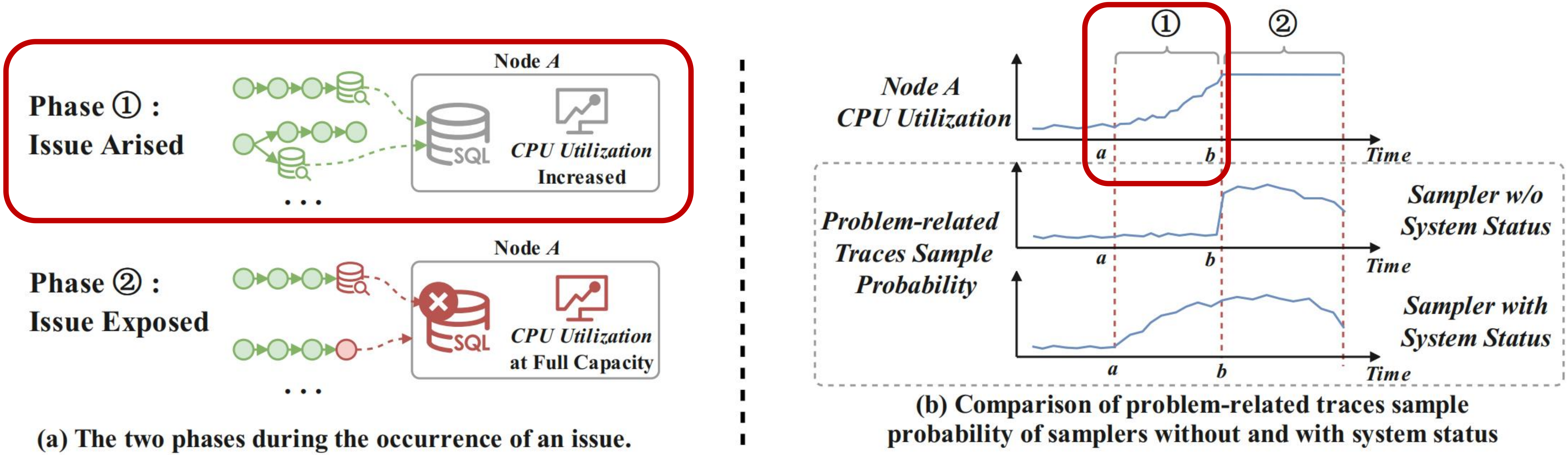


Figure. An example of the importance of system runtime state in trace sampling preferences setting.

Motivations

➢ A real-world case

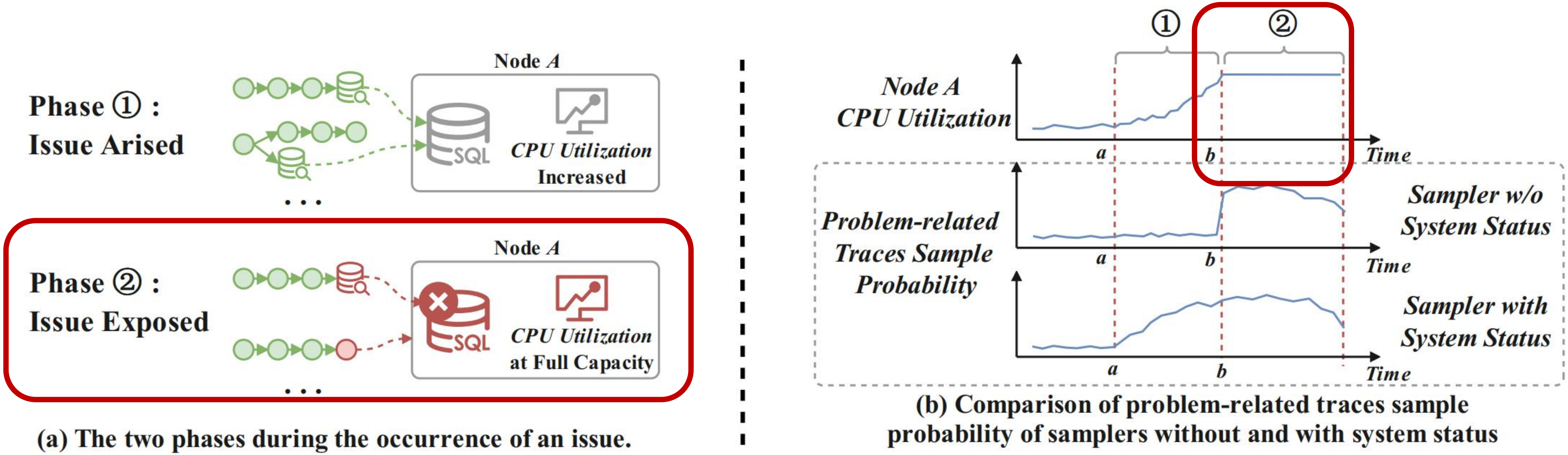


Figure. An example of the importance of system runtime state in trace sampling preferences setting.

Motivations

➢ A real-world case

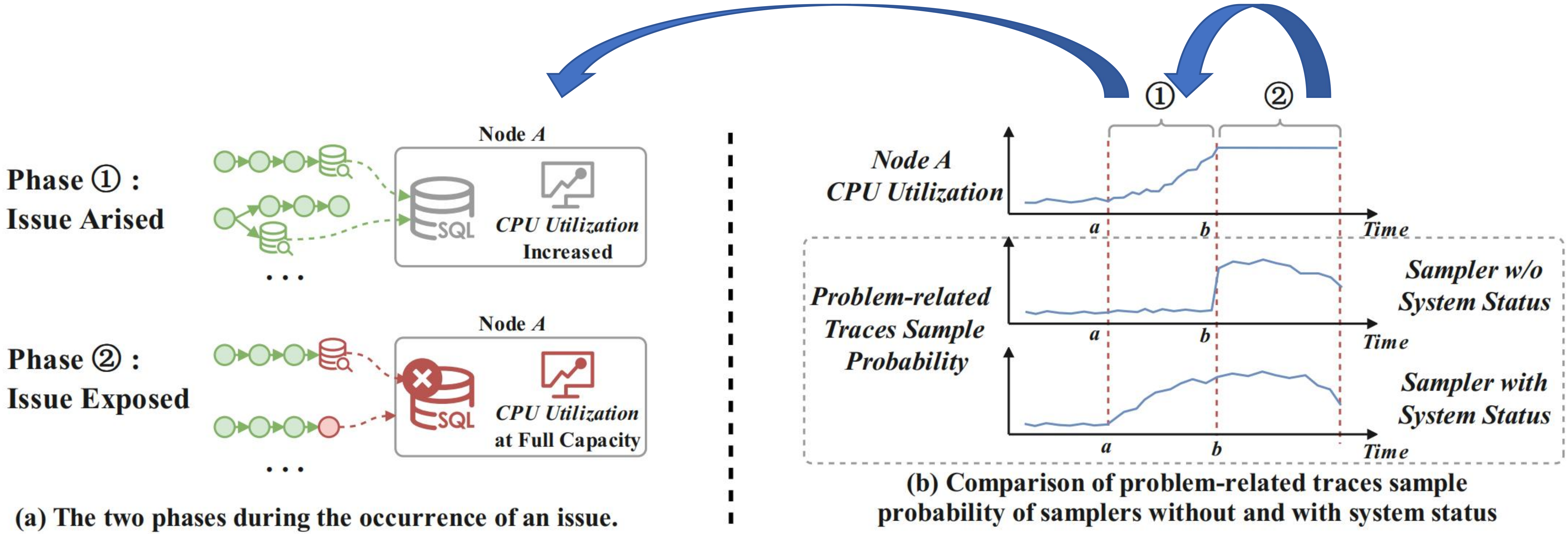


Figure. An example of the importance of system runtime state in trace sampling preferences setting.

Motivations

➢ A real-world case

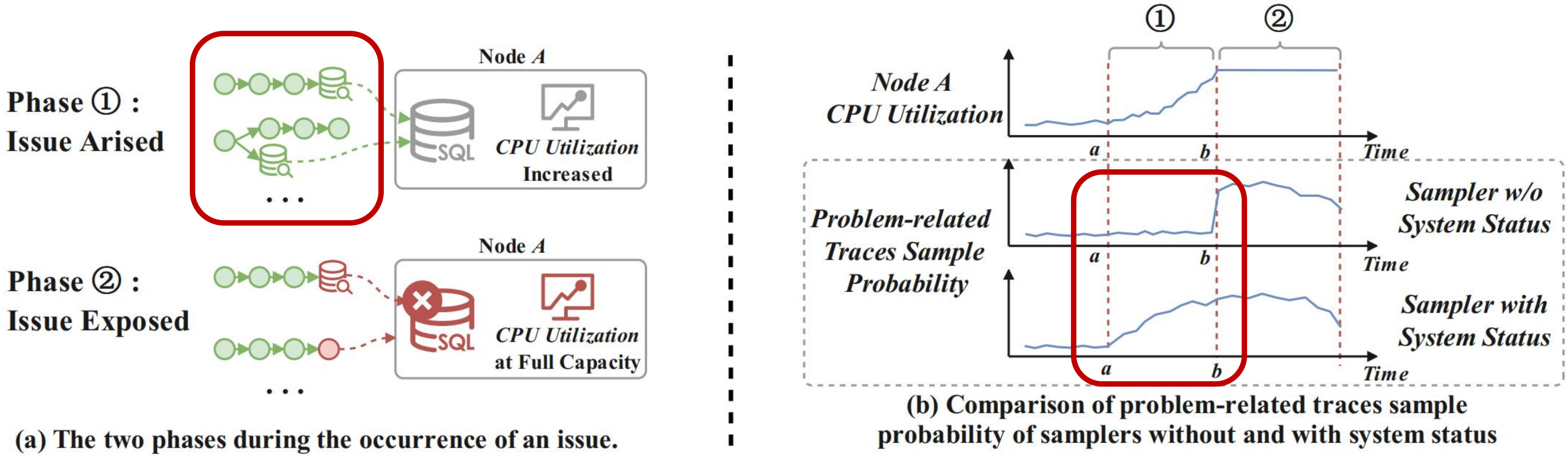
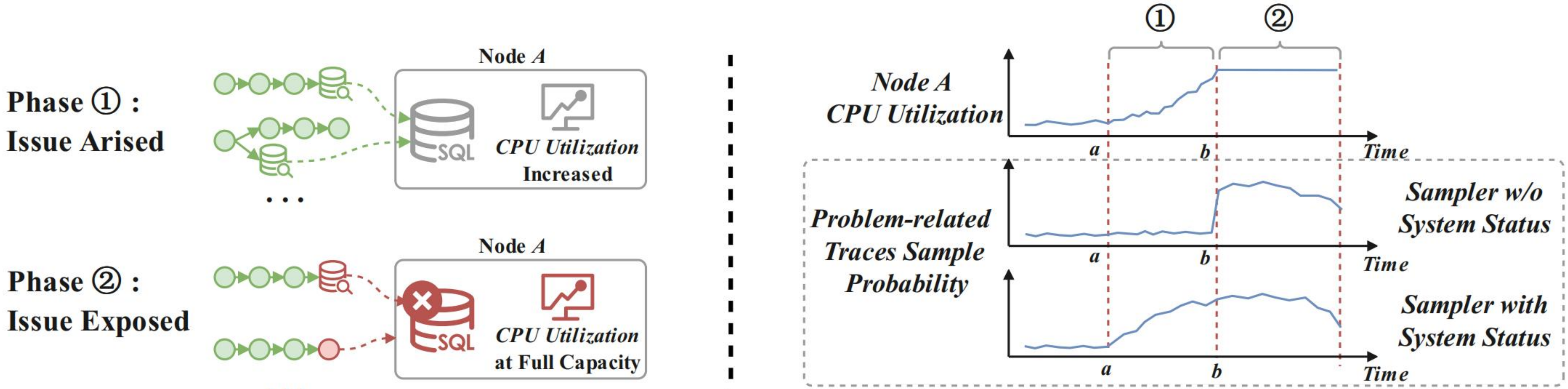


Figure. An example of the importance of system runtime state in trace sampling preferences setting.

Motivations

➢A real-world case

Goal: Achieve a more comprehensive sampling policy.



(a) The two phases during the occurrence of an issue.

(b) Comparison of problem-related traces sample probability of samplers without and with system status

Figure. An example of the importance of system runtime state in trace sampling preferences setting.

Methodology

➢ Overview

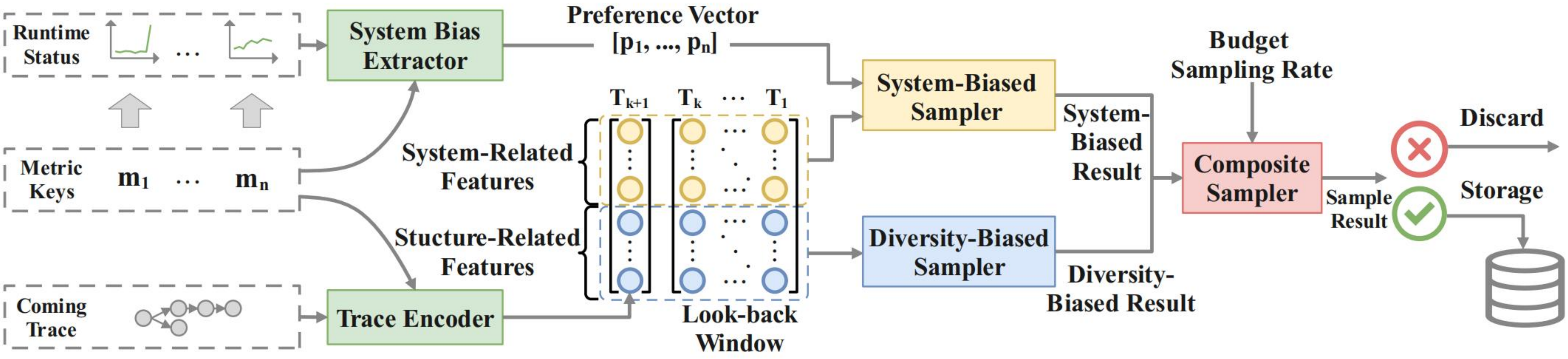


Figure. An overview of *TraStrainer*.

Methodology

➢Trace Encoder

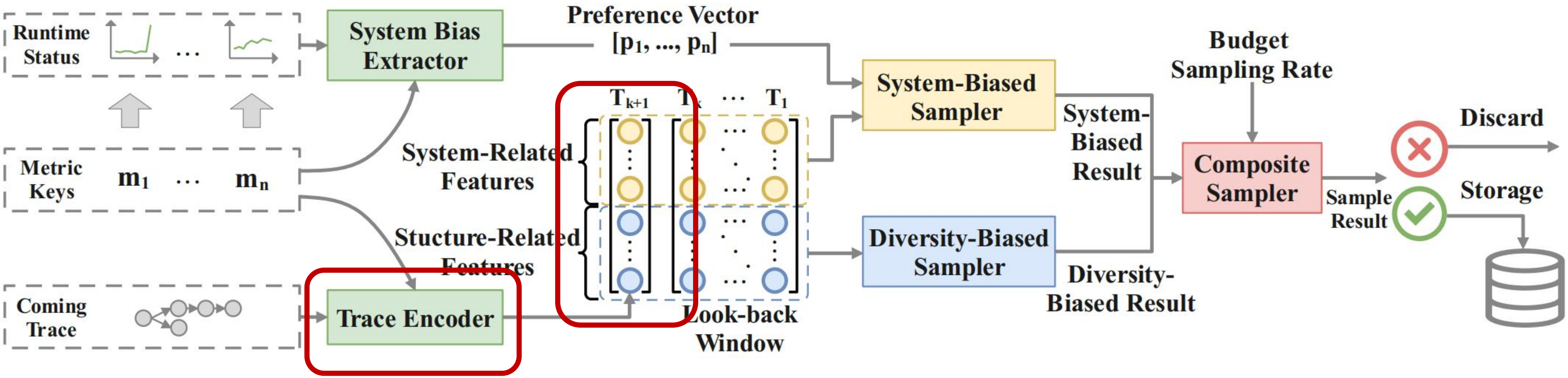
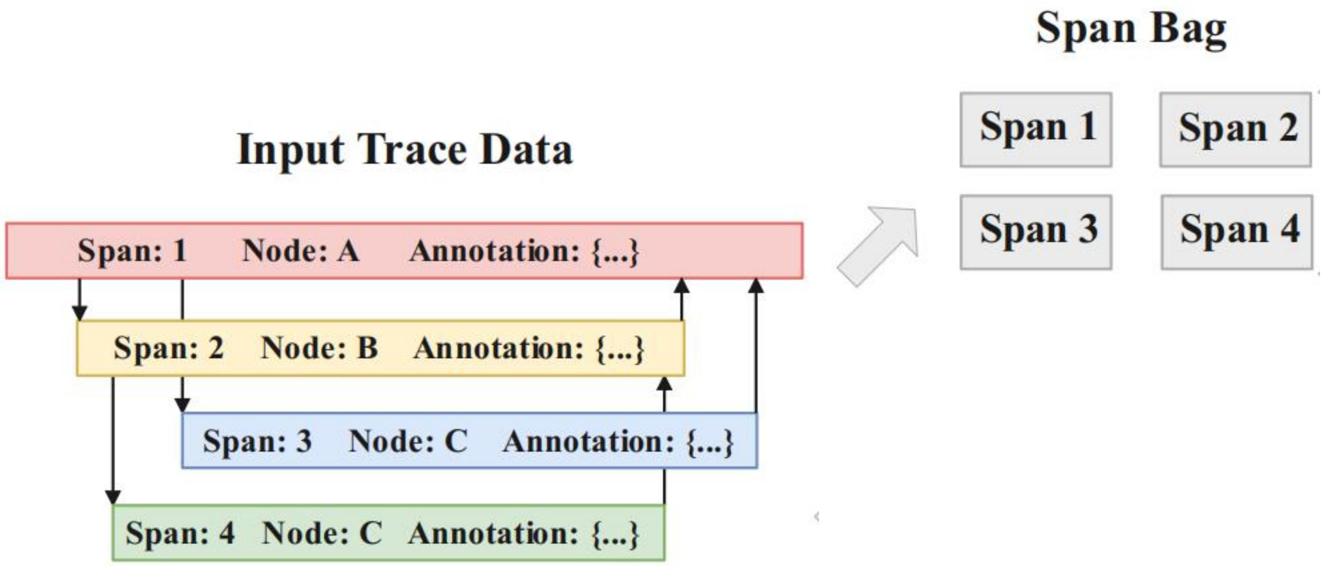


Figure. An overview of *TraStrainer*.

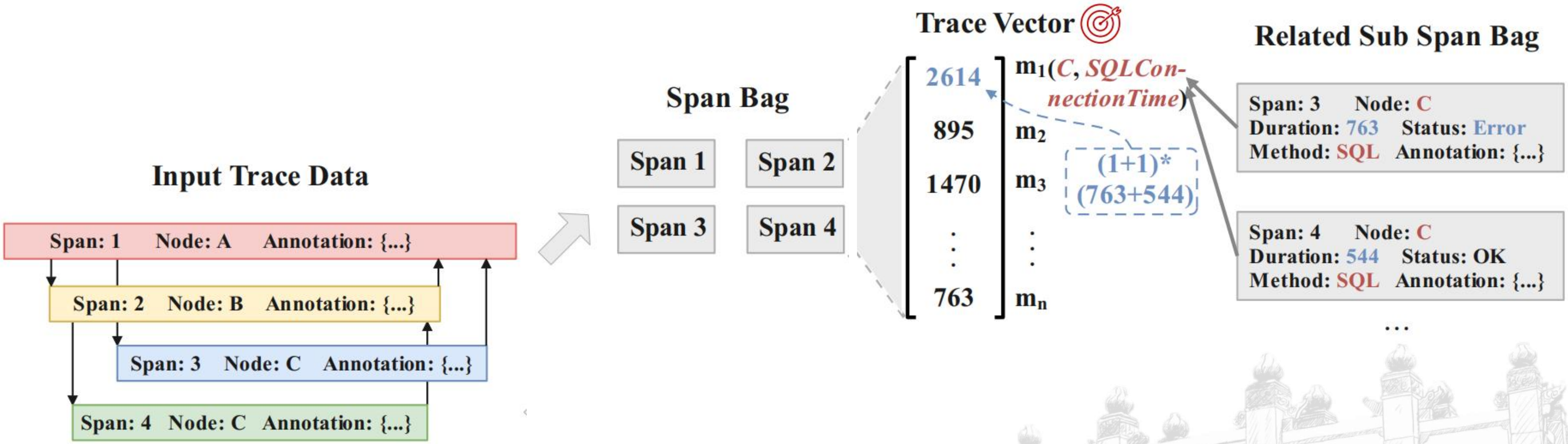
Methodology

➢ Trace Encoder

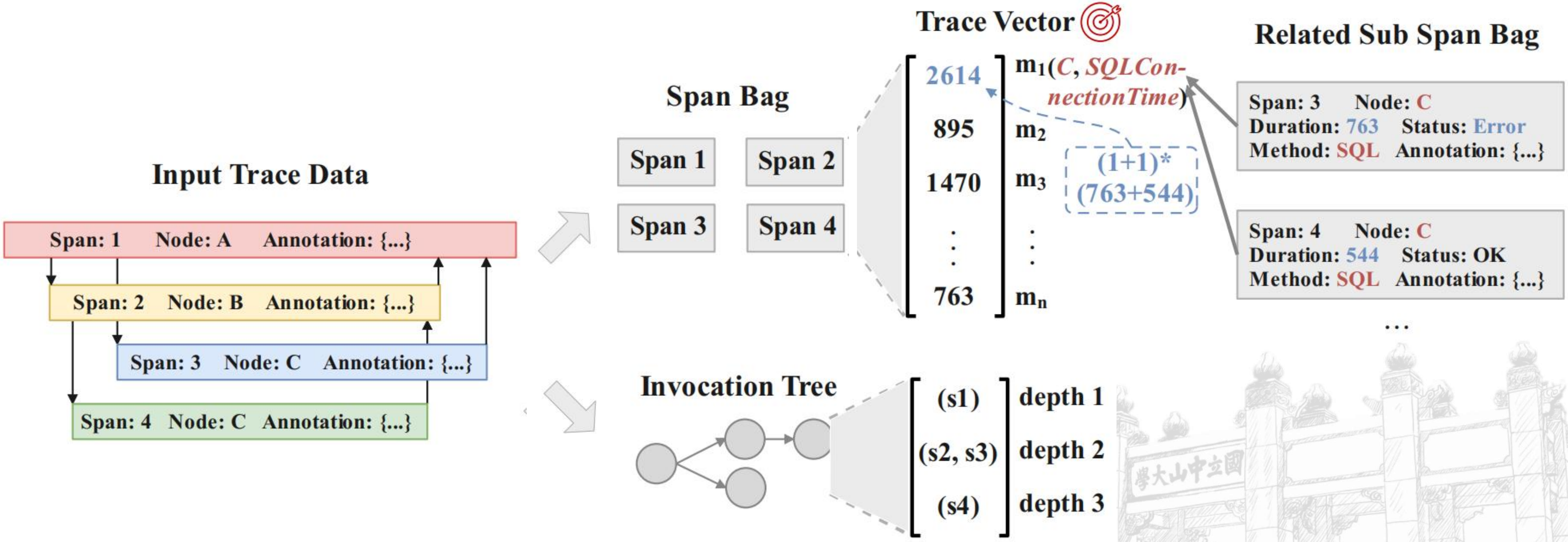


Methodology

➢Trace Encoder

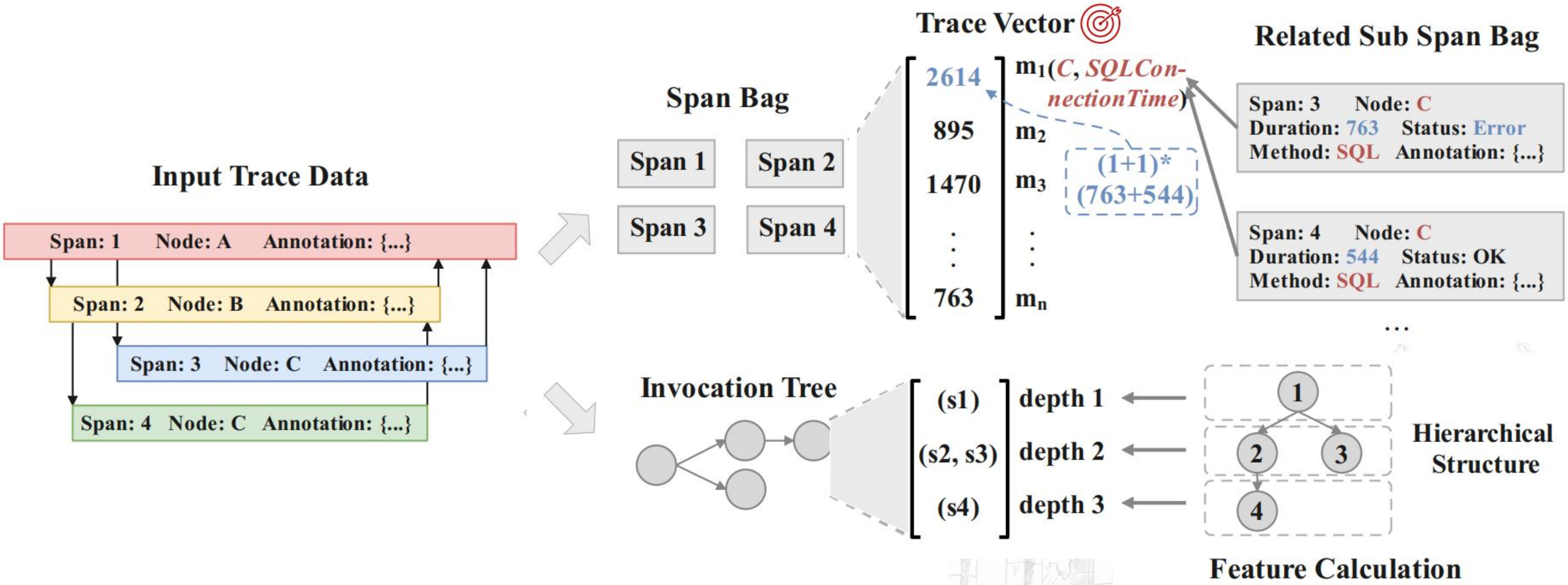


➢Trace Encoder



Methodology

➢Trace Encoder



Methodology

➢System Bias Extractor

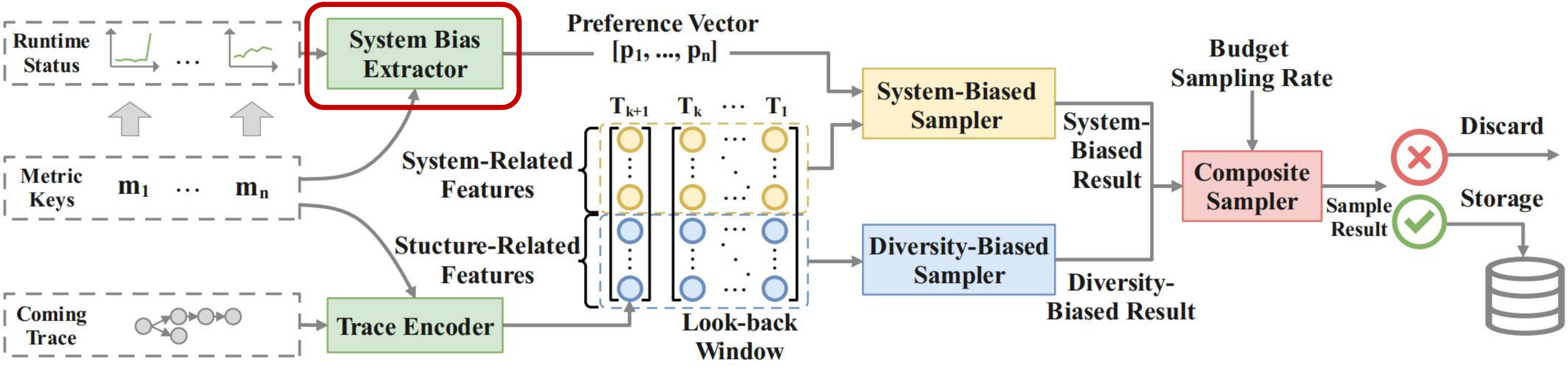
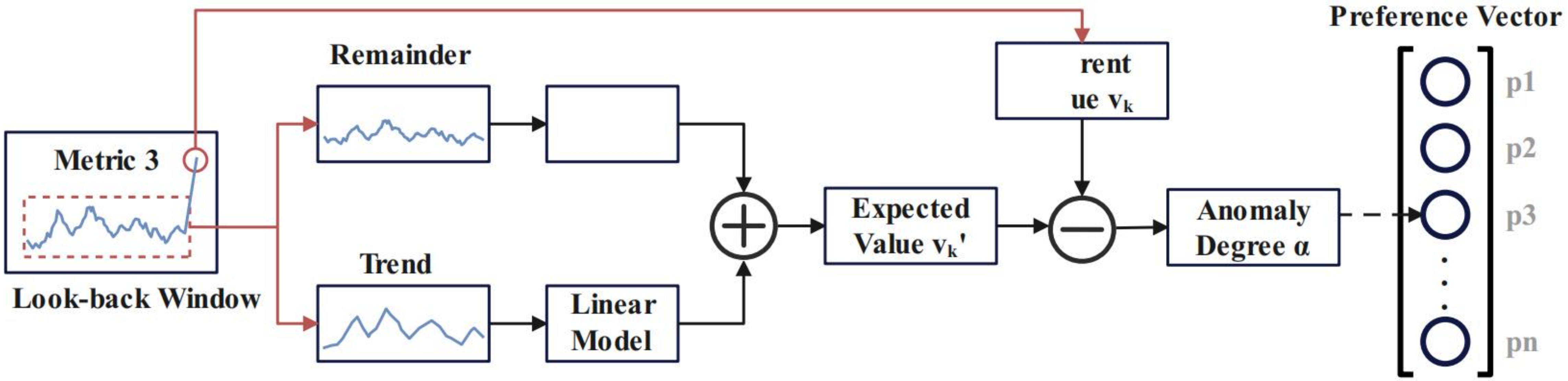


Figure. An overview of *TraStrainer*.

Methodology

➢System Bias Extractor



Methodology

➢ Sampler

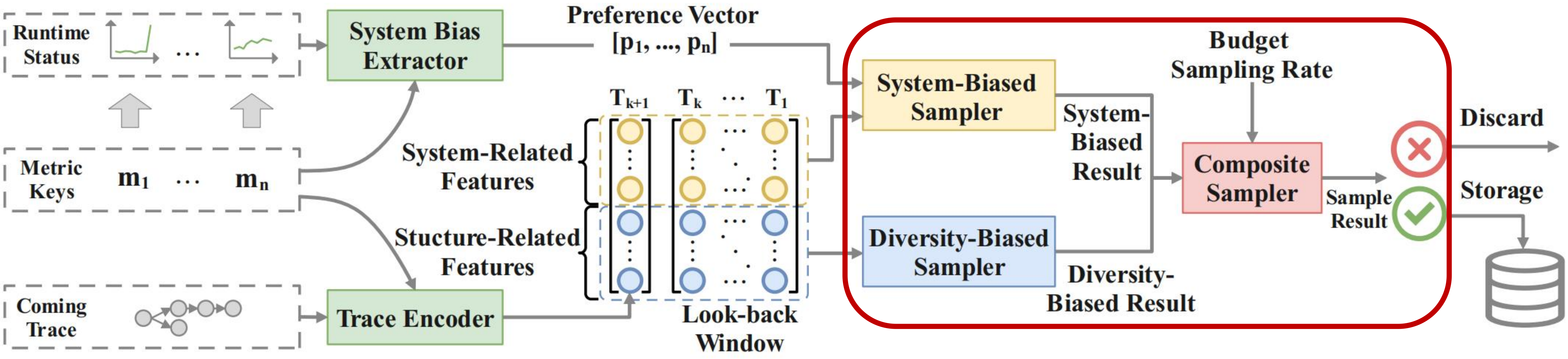
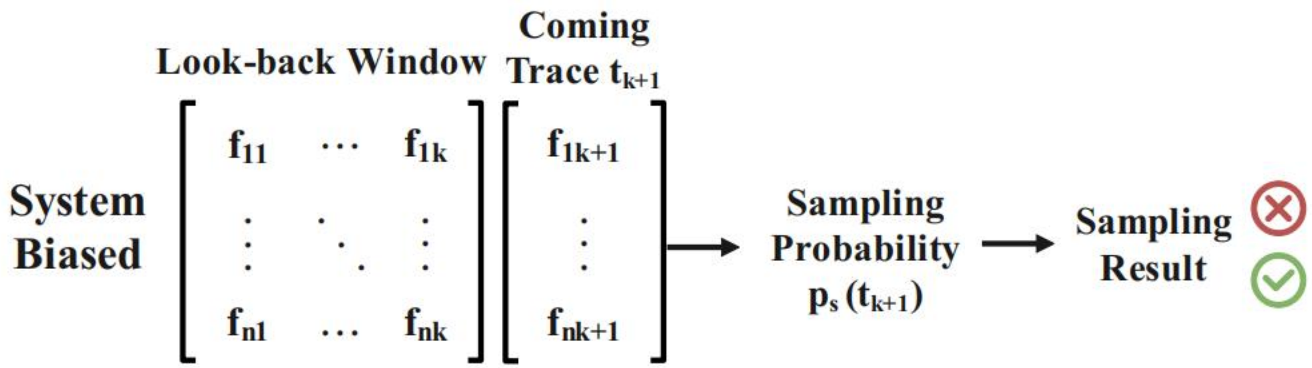


Figure. An overview of *TraStrainer*.

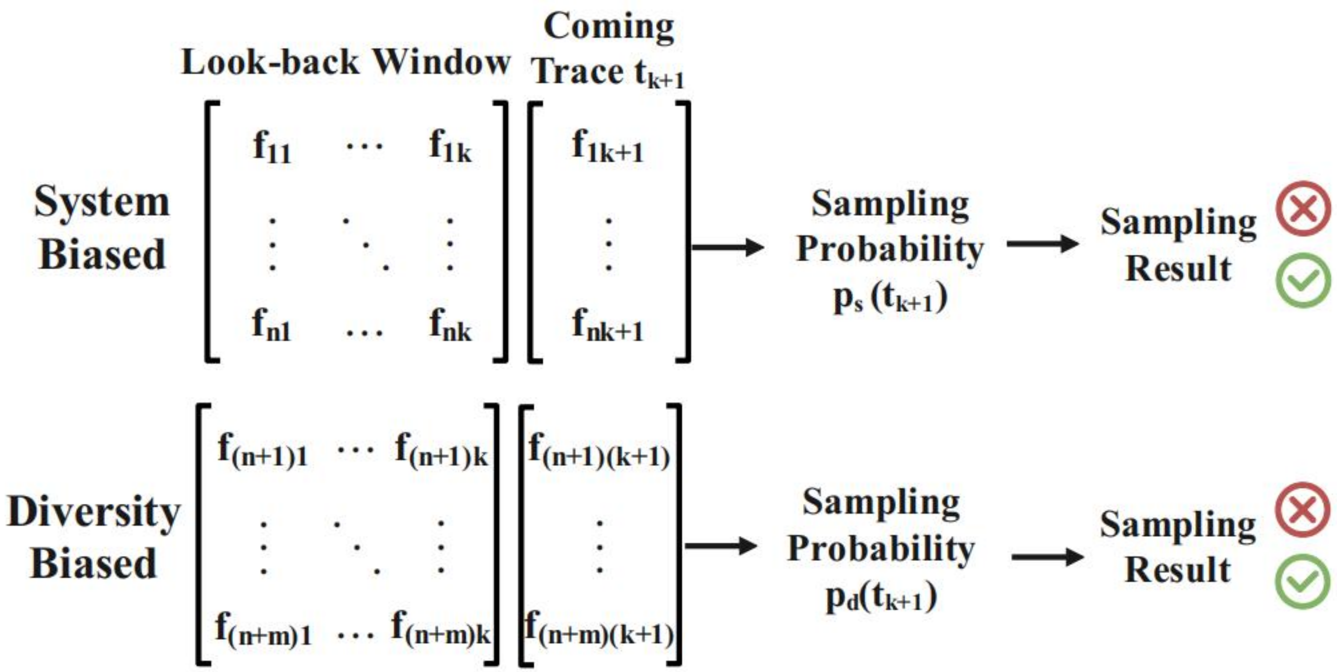
Methodology

➢ System-Biased Sampler



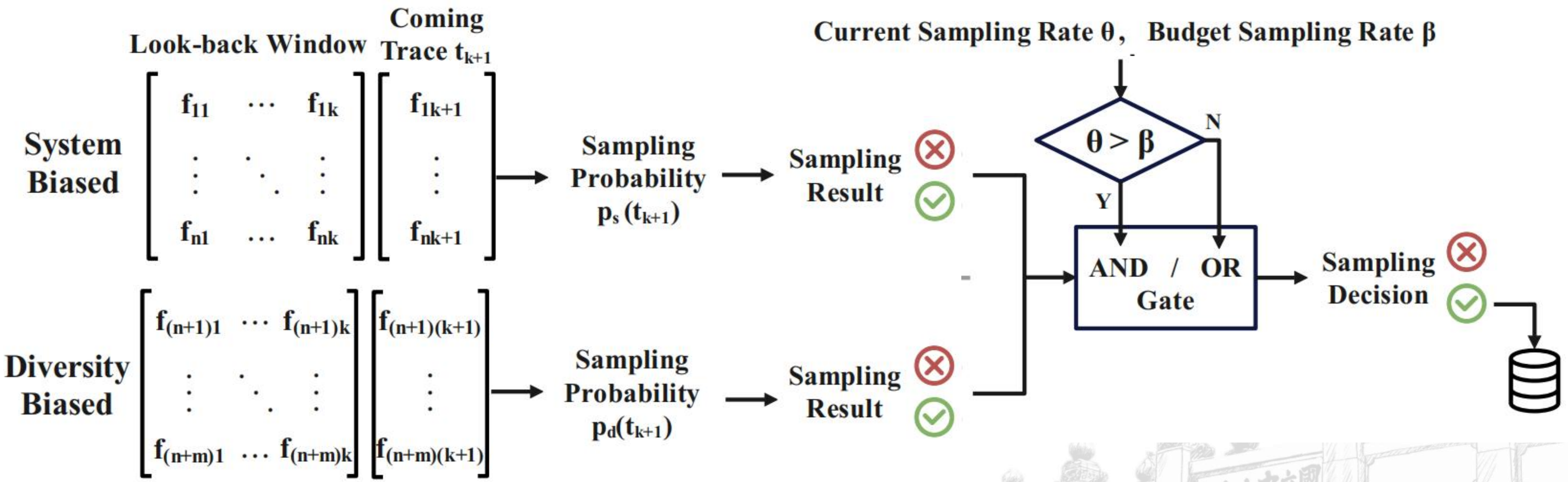
Methodology

➢Diversity-Biased Sampler



Methodology

➢Composite Sampler



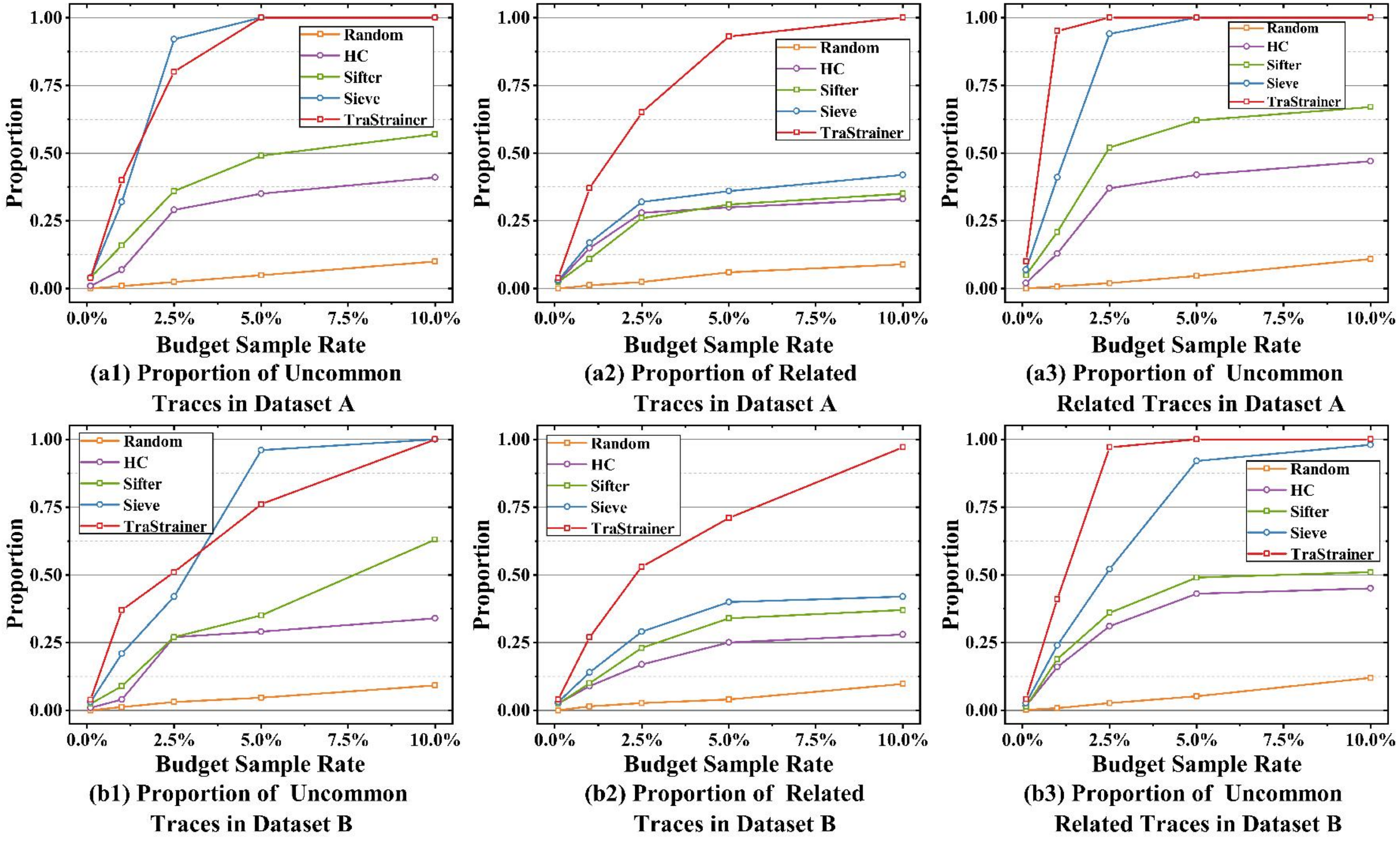
Evaluation

1. How does the quality of the traces sampled by TraStrainer compare to the baseline approaches?
2. How effective is TraStrainer in downstream trace-based root cause analysis compared with baseline approaches?
3. How much does considering both system runtime state and trace diversity contribute to the effectiveness of TraStrainer?
4. How efficient is the sampling of TraStrainer?

Evaluation

➢RQ1: Sampling Quality

TraStrainer achieves a proportion above 90% when catching problem-related traces in both datasets, while the other baselines remain below 55%



(a1) Proportion of Uncommon Traces in Dataset A

(a2) Proportion of Related Traces in Dataset A

(a3) Proportion of Uncommon Related Traces in Dataset A

(b1) Proportion of Uncommon Traces in Dataset B

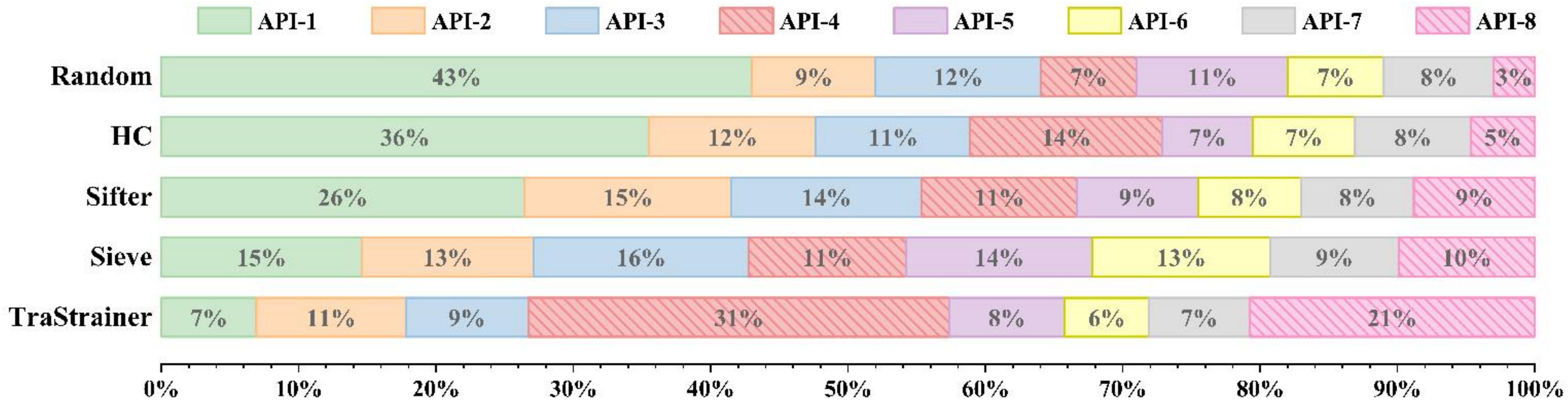
(b2) Proportion of Related Traces in Dataset B

(b3) Proportion of Uncommon Related Traces in Dataset B

Evaluation

➢RQ1: Sampling Quality

TraStrainer can better prioritize underrepresented and abnormal request types.



(a) 8 APIs from a batch in Dataset A (API-4 and API-8 are anomalies)

Evaluation

➢RQ2: Effectiveness for Downstream Root Cause Analysis.

TraStrainer leads to an average increase of 32.63% in Top-1 root cause analysis accuracy compared to four baselines in two datasets.

Table 3. Comparison of the effects of different sampling approaches in downstream root cause analysis.

RCA Approach	Sampling Approach	A@1			A@3			MRR		
		0.1%	1.0%	2.5%	0.1%	1.0%	2.5%	0.1%	1.0%	2.5%
TraceAnomaly	Random	10.71	9.26	16.67	44.73	57.41	57.41	0.2820	0.3503	0.3997
	HC	9.26	12.96	18.52	37.04	42.59	55.56	0.2590	0.3664	0.3747
	Sifter	11.67	24.07	16.67	37.04	57.41	62.96	0.2753	0.4025	0.4145
	Sieve	8.81	18.52	29.63	44.44	53.70	57.41	0.2620	0.3762	0.4383
	TraStrainer w/o $\mathcal{M}$	11.67	20.37	22.22	45.15	51.85	55.56	0.2903	0.3722	0.4068
	TraStrainer w/o $\mathcal{D}$	12.96	38.89	44.44	49.81	77.78	75.93	0.3485	0.5948	0.6247
	TraStrainer	**46.30**	**51.61**	**54.84**	**66.67**	**79.19**	**87.10**	**0.5707**	**0.6438**	**0.7151**
TraceRCA	Random	7.41	20.37	29.63	40.74	61.11	68.52	0.2525	0.4123	0.4991
	HC	9.26	24.07	24.07	46.30	62.96	62.96	0.2546	0.4324	0.4627
	Sifter	8.67	19.63	25.19	37.04	55.56	61.11	0.2449	0.4272	0.4836
	Sieve	18.52	31.48	38.89	42.59	51.85	57.41	0.3008	0.4157	0.4873
	TraStrainer w/o $\mathcal{M}$	18.52	33.33	35.19	44.44	55.56	55.56	0.3191	0.4432	0.4642
	TraStrainer w/o $\mathcal{D}$	24.07	55.56	55.56	38.89	81.48	77.78	0.3650	0.6880	0.6843
	TraStrainer	**55.56**	**55.56**	**58.06**	**70.37**	**85.19**	**89.63**	**0.6265**	**0.7019**	**0.7510**
MicroRank	Random	5.56	16.67	27.78	20.37	50.00	61.11	0.1571	0.3423	0.4352
	HC	7.41	18.52	22.22	27.78	46.30	51.85	0.1954	0.3398	0.3731
	Sifter	5.56	18.52	27.78	23.42	46.30	61.11	0.1605	0.3414	0.4358
	Sieve	9.26	25.83	35.19	20.37	58.15	62.96	0.1657	0.4246	0.4963
	TraStrainer w/o $\mathcal{M}$	12.96	16.67	24.07	42.59	42.59	55.56	0.2994	0.3241	0.4012
	TraStrainer w/o $\mathcal{D}$	29.63	42.59	46.30	74.04	68.52	72.22	0.5228	0.5463	0.5509
	TraStrainer	**42.59**	**45.16**	**50.00**	**77.74**	**78.52**	**82.26**	**0.5509**	**0.5889**	**0.6556**

Evaluation

➢RQ3: Contribution of Each Sampling Factor

Considering both system state and trace diversity achieves better analysis performance across all budgets.

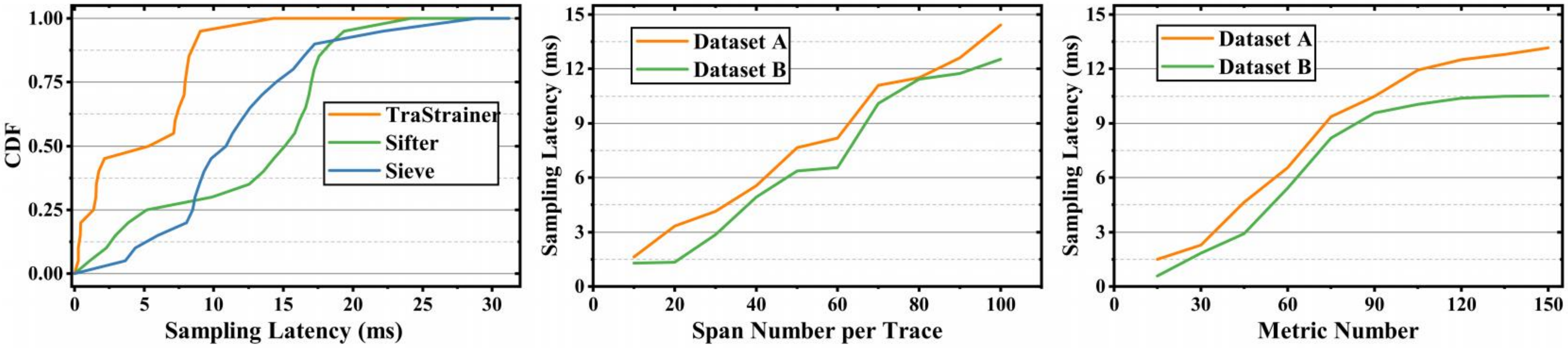
Table 3. Comparison of the effects of different sampling approaches in downstream root cause analysis.

RCA Approach	Sampling Approach	A@1			A@3			MRR		
		0.1%	1.0%	2.5%	0.1%	1.0%	2.5%	0.1%	1.0%	2.5%
TraceAnomaly	Random	10.71	9.26	16.67	44.73	57.41	57.41	0.2820	0.3503	0.3997
	HC	9.26	12.96	18.52	37.04	42.59	55.56	0.2590	0.3664	0.3747
	Sifter	11.67	24.07	16.67	37.04	57.41	62.96	0.2753	0.4025	0.4145
	Sieve	8.81	18.52	29.63	44.44	53.70	57.41	0.2620	0.3762	0.4383
	TraStrainer w/o $\mathcal{M}$	11.67	20.37	22.22	45.15	51.85	55.56	0.2903	0.3722	0.4068
	TraStrainer w/o $\mathcal{D}$	12.96	38.89	44.44	49.81	77.78	75.93	0.3485	0.5948	0.6247
	TraStrainer	**46.30**	**51.61**	**54.84**	**66.67**	**79.19**	**87.10**	**0.5707**	**0.6438**	**0.7151**
TraceRCA	Random	7.41	20.37	29.63	40.74	61.11	68.52	0.2525	0.4123	0.4991
	HC	9.26	24.07	24.07	46.30	62.96	62.96	0.2546	0.4324	0.4627
	Sifter	8.67	19.63	25.19	37.04	55.56	61.11	0.2449	0.4272	0.4836
	Sieve	18.52	31.48	38.89	42.59	51.85	57.41	0.3008	0.4157	0.4873
	TraStrainer w/o $\mathcal{M}$	18.52	33.33	35.19	44.44	55.56	55.56	0.3191	0.4432	0.4642
	TraStrainer w/o $\mathcal{D}$	24.07	55.56	55.56	38.89	81.48	77.78	0.3650	0.6880	0.6843
	TraStrainer	**55.56**	**55.56**	**58.06**	**70.37**	**85.19**	**89.63**	**0.6265**	**0.7019**	**0.7510**
MicroRank	Random	5.56	16.67	27.78	20.37	50.00	61.11	0.1571	0.3423	0.4352
	HC	7.41	18.52	22.22	27.78	46.30	51.85	0.1954	0.3398	0.3731
	Sifter	5.56	18.52	27.78	23.42	46.30	61.11	0.1605	0.3414	0.4358
	Sieve	9.26	25.83	35.19	20.37	58.15	62.96	0.1657	0.4246	0.4963
	TraStrainer w/o $\mathcal{M}$	12.96	16.67	24.07	42.59	42.59	55.56	0.2994	0.3241	0.4012
	TraStrainer w/o $\mathcal{D}$	29.63	42.59	46.30	74.04	68.52	72.22	0.5228	0.5463	0.5509
	TraStrainer	**42.59**	**45.16**	**50.00**	**77.74**	**78.52**	**82.26**	**0.5509**	**0.5889**	**0.6556**

Evaluation

➢RQ4: Efficiency of TraStrainer

TraStrainer's efficiency is comparable to other biased samplers, proving it is a practical tool.



(a) Efficiency Comparison of Different Approaches (b) Efficiency Sensitivity to Span Number (c) Efficiency Sensitivity to Metric Number

Thanks for Listening!

Q & A

huanghy95@mail2.sysu.edu.cn
https://huanghy95.github.io/
https://github.com/IntelligentDDS/TraStrainer