Prism: Revealing Hidden Functional Clusters from Massive Instances in Cloud Systems

Jinyang Liu[1], Zhihan Jiang[1], Jiazhen Gu[1], Junjie Huang[1], Zhuangbin Chen[2], Cong Feng[3], Zengyin Yang[3], Yongqiang Yang[3], Michael R. Lyu[1]

[1]*The Chinese University of Hong Kong,* [2]*Sun Yat-sen University,*
[3]*Computing and Networking Innovation Lab, Huawei Cloud Computing Technology Co., Ltd*

ARISE Automated Reliable Intelligent Software Engineering

香港中文大學 The Chinese University of Hong Kong

中山大學 SUN YAT-SEN UNIVERSITY

HUAWEI

Cloud Computing

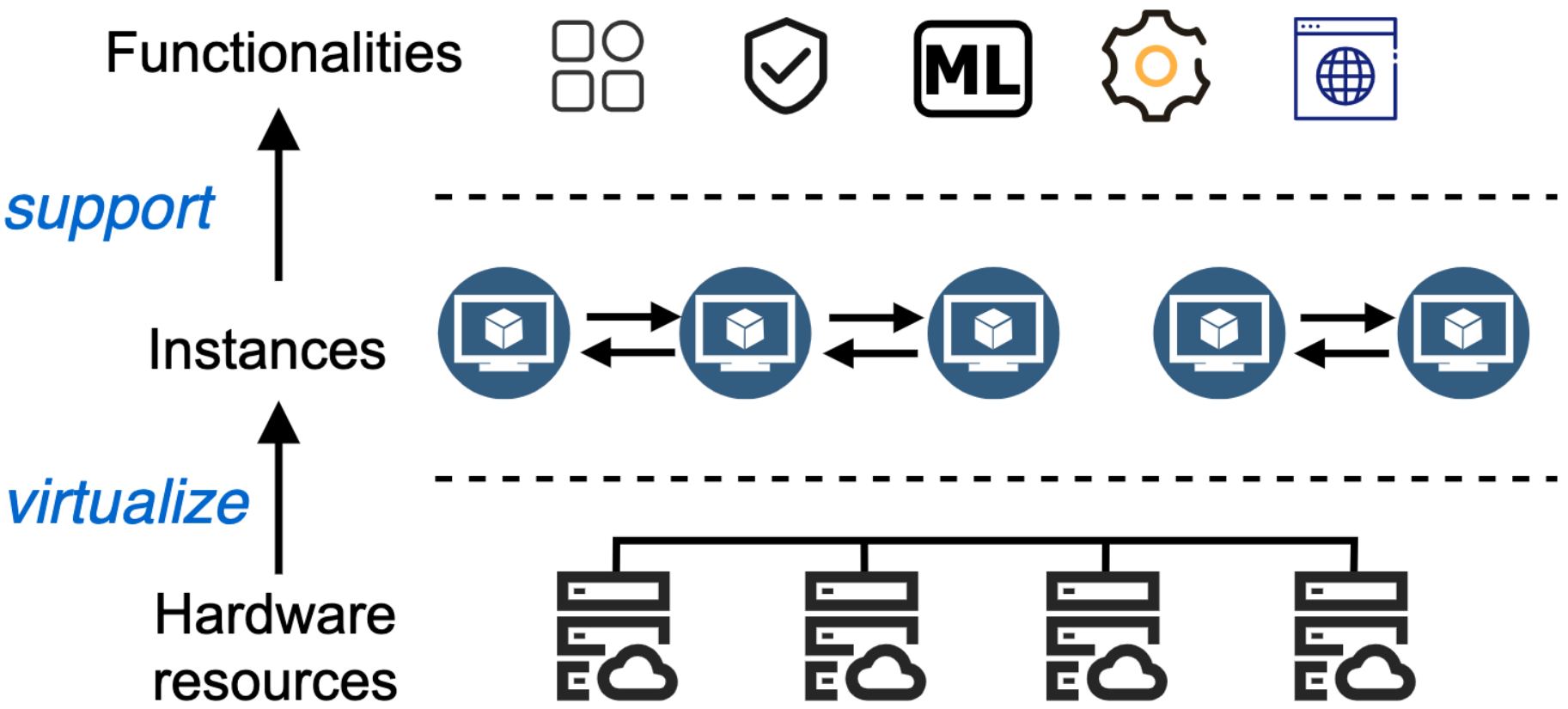
- Many applications have migrated to the cloud.





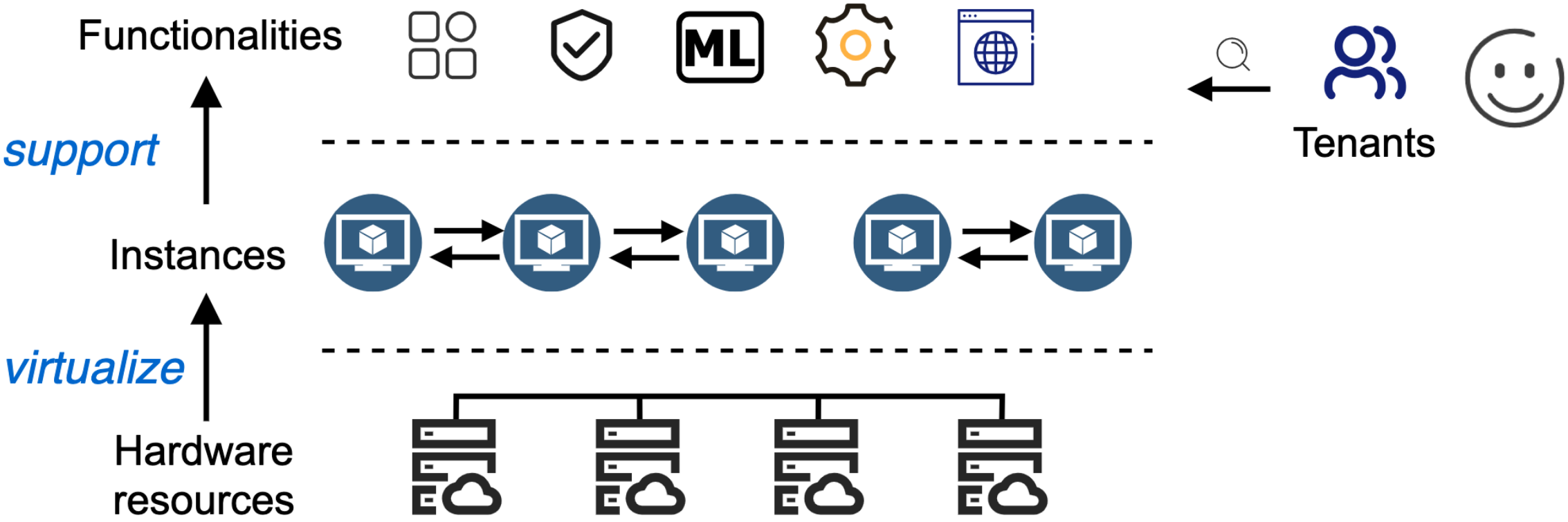

.....

Cloud Infrastructure



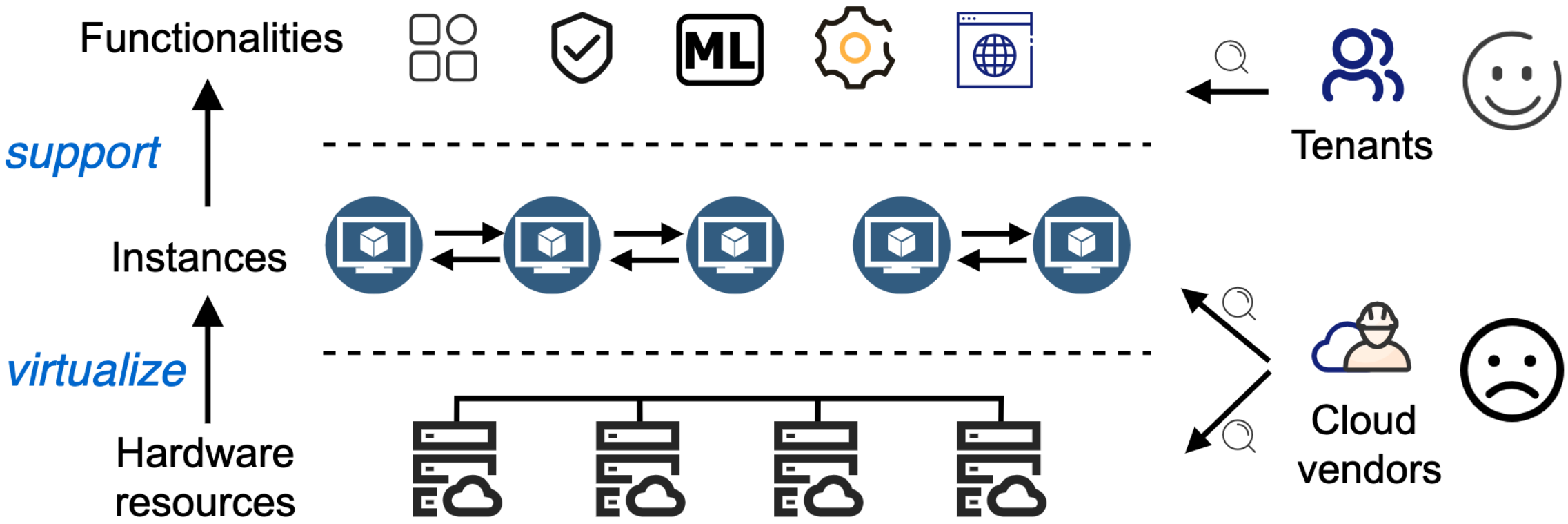
Hierarchical Cloud Infrastructure

Cloud Infrastructure



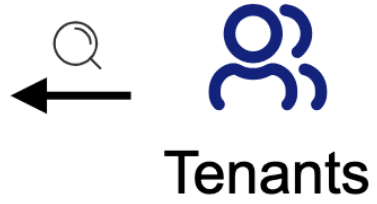
Virtualization from hardware to virtual instances
enhances resource utilization and **simplifies usage for customers**.

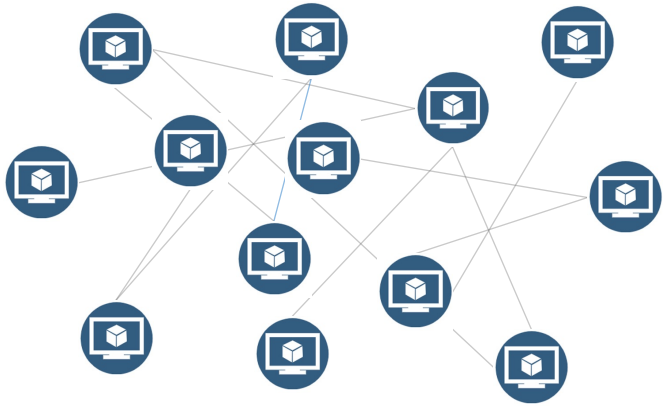
Cloud Infrastructure



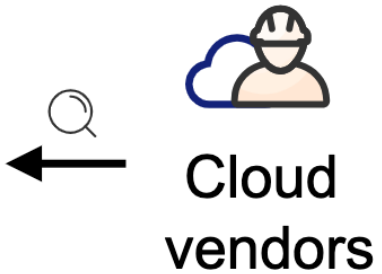
However, it **reduces cloud observability** for cloud vendors during maintenance tasks.

A Motivating Example

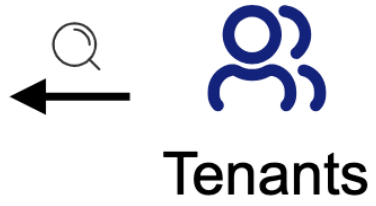




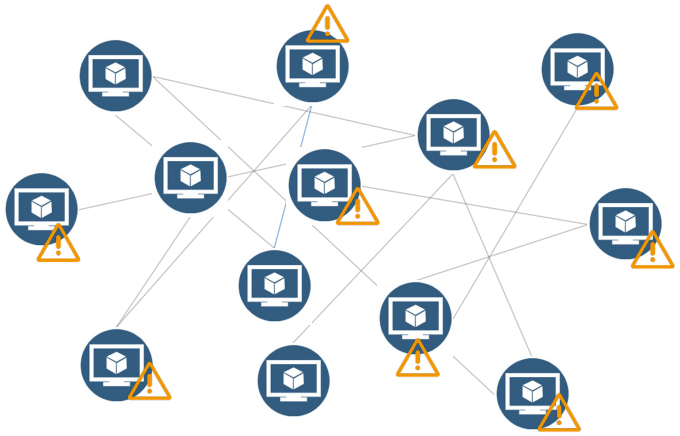
Massive **Black-box** Instances
(typically millions of)



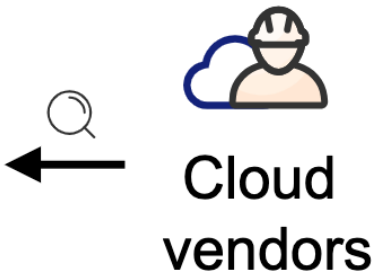
A Motivating Example



Packet loss

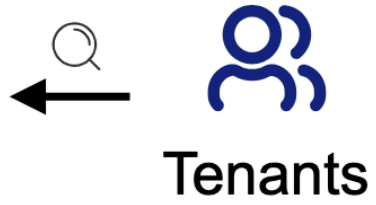


Do they affect customers?
Should I launch diagnosis?

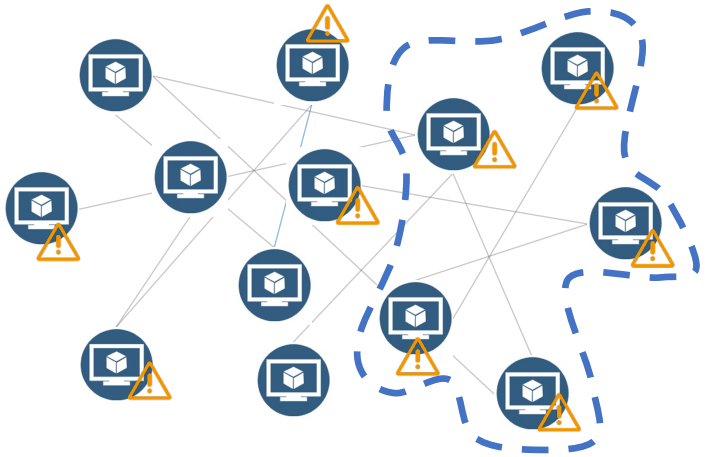


Massive **Black-box** Instances
(typically millions of)

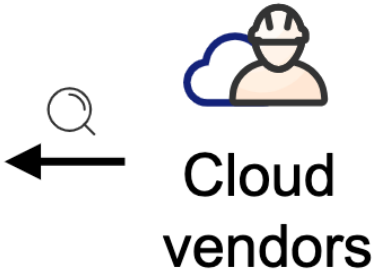
A Motivating Example



Packet loss

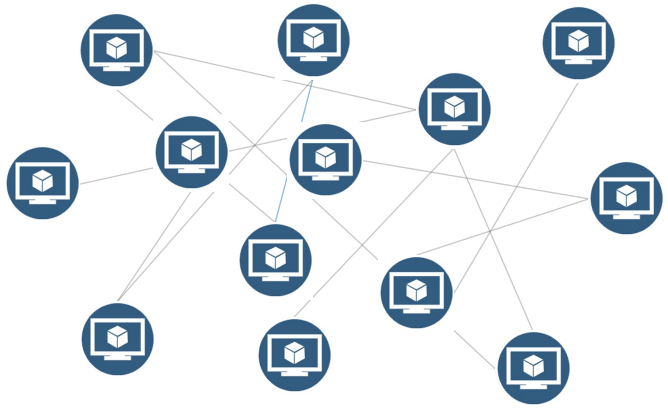


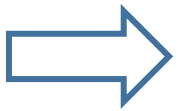
This functionality **is very likely affected**. Launch diagnosis.

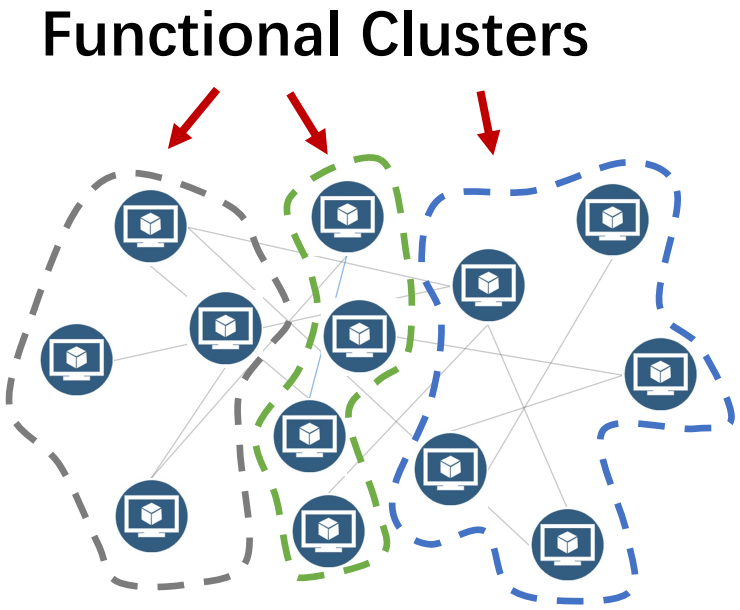


Clustered Instances
(Serving the same functionalities)

Our Problem







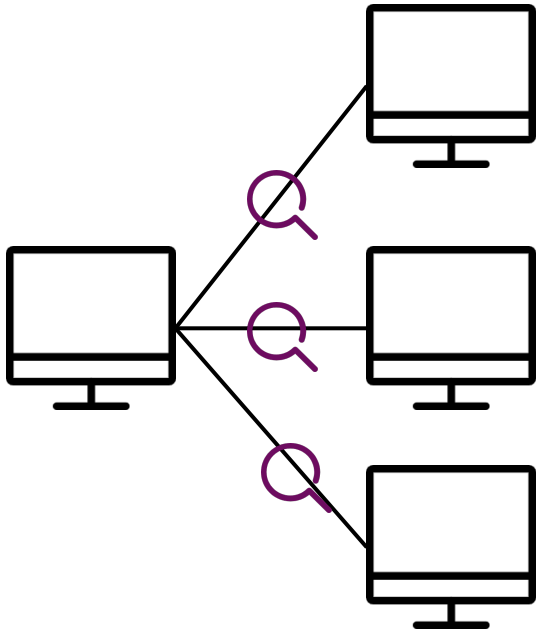
Massive **Black-box** Instances
(typically millions of)

Clustered Instances
(Serving the same functionalities)

Problem: How do we find **functional clusters** in massive instances **with ONLY data visible to cloud vendors** (with customers' consent)?

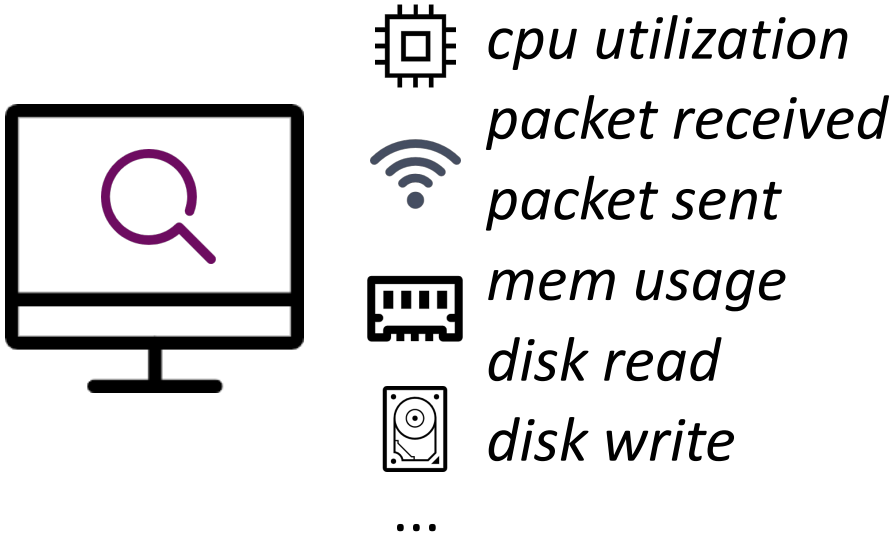
Data visible to cloud vendors

- Two types of typical monitoring data



Trace: (srcIp, dstIp, srcPort, dstPort)

Communication Traces



Monitoring Metrics

A Pilot Study

- 3,062 internal instances covering 397 types of functionalities

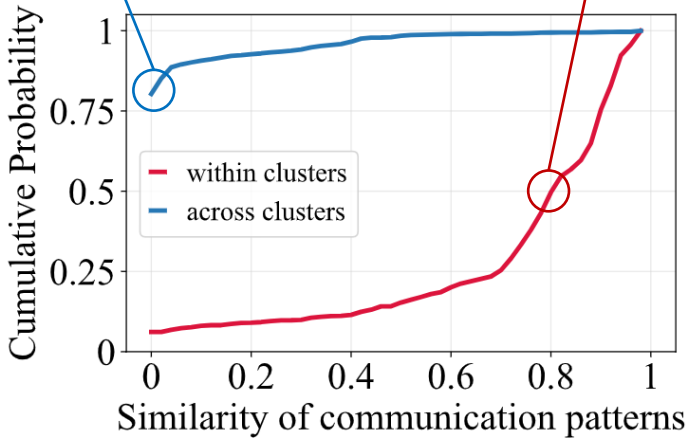
> 75% across-cluster instances have nearly zero similarity.

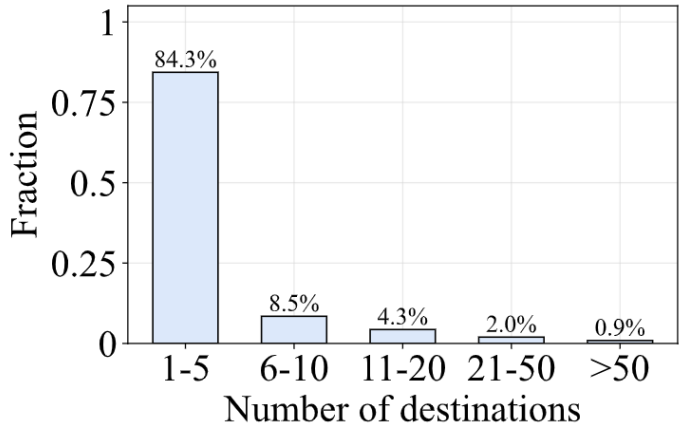
> 50% in-cluster instances have > 0.8 similarity.

Communication Patterns

⇧

Communication Traces

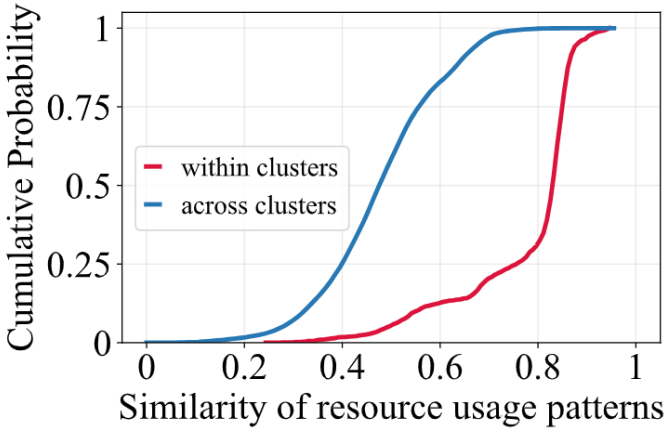




Resource Patterns

⇧

Monitoring Metrics



Findings

- In-cluster instances share **similar communication and resource patterns**.
- Most instances only communicate with a small number of instances (**locality**).
- Both data are **noisy (no guarantee)**.

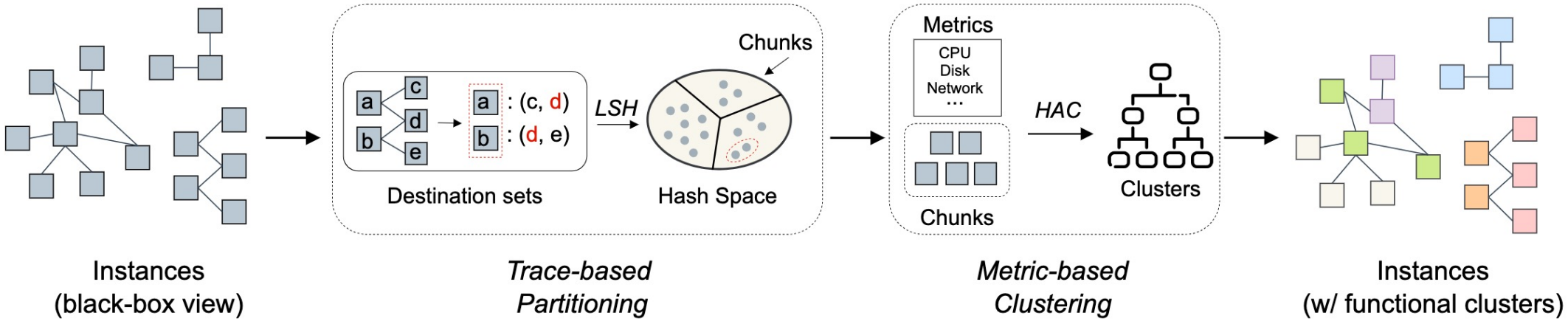
Method

Problem: How do we find **functional clusters** in massive instances **with ONLY data visible to cloud vendors** (with customers' consent)?

Challenges:

- Massive instances (typically millions in cloud systems)
- Limited noisy monitoring data for cloud vendors

Our Solution: **Prism**



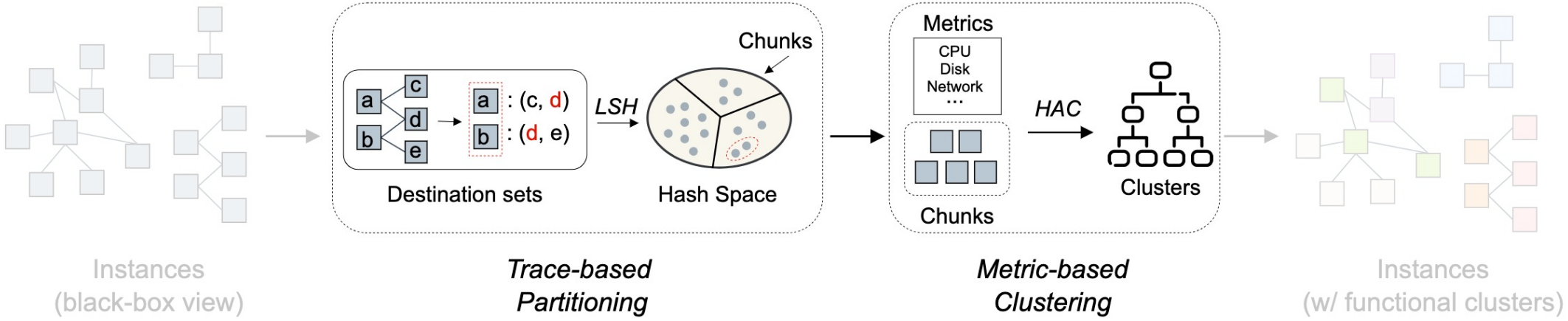
Method

Problem: How do we find **functional clusters** in massive instances **with ONLY data visible to cloud vendors** (with customers' consent)?

Challenges:

- Massive instances (typically millions in cloud systems)
- Limited noisy monitoring data for cloud vendors

Our Solution: **Prism**

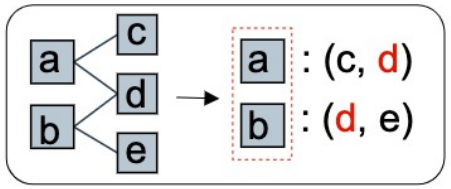


Method

Trace-based Partitioning

Input:
- All instances
- Communication traces

Output:
- Coarse-grained chunks



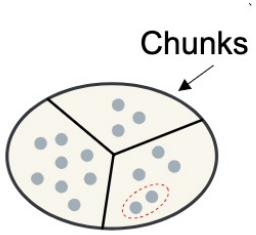
Destination Sets

Jaccard similarity:

$$J(x_i, x_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$$

Pairwise comparison ✗

Efficient Local Sensitive Hashing

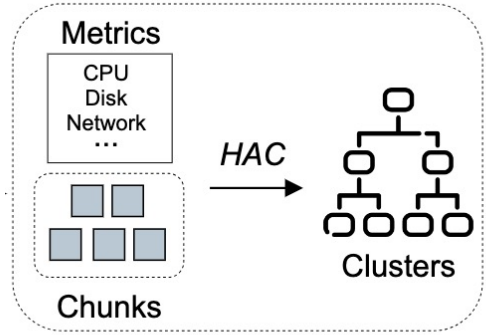


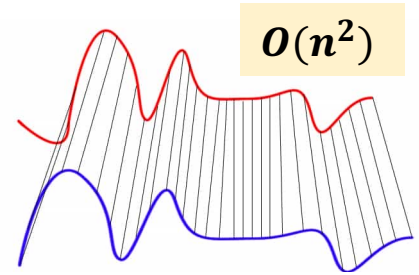
Strong Locality!

Metric-based Clustering

Input:
- Coarse-grained chunks
- Monitoring metrics (cpu, mem, disk, etc.)

Output:
- Functional clusters





Dynamic Time Warping (DTW) Distance

Apply independently for each small chunk (<=50 instances)

Evaluation

- Datasets

Datasets	# Functionalities	# Instances	# Traces	# Metrics
Dataset $\mathcal{A}$	292	2,035	100.2 M	7.25 M
Dataset $\mathcal{B}$	105	1,027	121.6 M	3.71 M
Total	397	3,062	212.6 M	10.96 M

- Real-world data from Huawei Cloud
- Manually labeled internal Instances

- Research Questions
  - RQ1: What is the **effectiveness** of Prism?
  - RQ2: What is the **contribution of each component**?
  - RQ3: What is the **impact of parameter settings**?
  - RQ4: What is the **efficiency** of Prism?
- Metrics
  - Homogeneity: how precise?
  - Completeness: how complete?
  - V-measure: a balanced metric

Evaluation

- RQ1: Effectiveness

Methods	Dataset $\mathcal{A}$			Dataset $\mathcal{B}$		
	Homo.	Comp.	V Meas.	Homo.	Comp.	V Meas.
OSImage	0.238	0.894	0.376	0.258	0.889	0.400
CloudCluster	0.346	0.748	0.473	0.369	0.753	0.495
ROCKA	0.831	0.882	0.856	0.875	0.900	0.887
OmniCluster	0.932	0.862	<u>0.896</u>	0.944	0.877	<u>0.909</u>
Prism	0.976	0.916	**0.945**	0.979	0.922	**0.950**

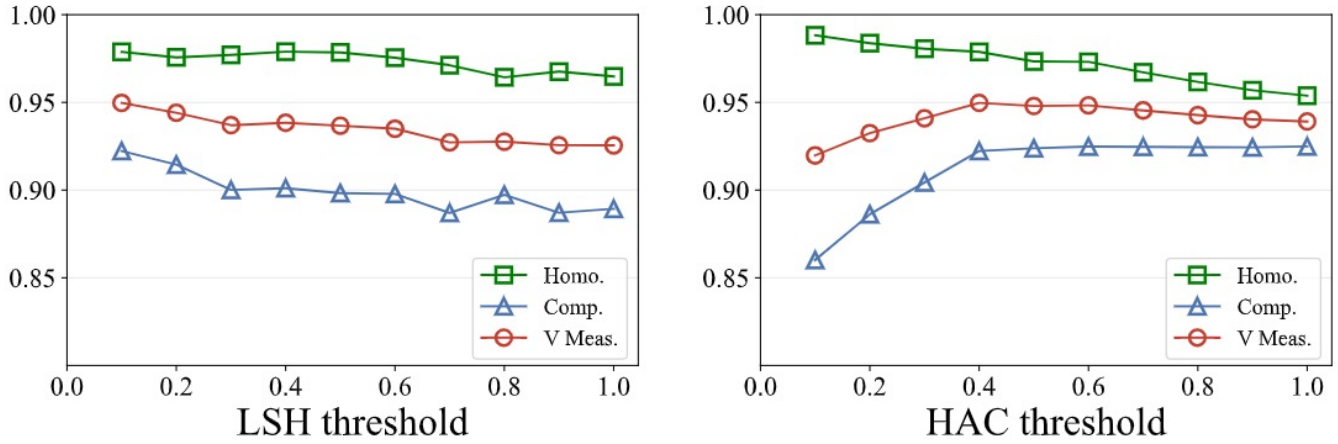
- RQ2: Ablation

Methods	Dataset $\mathcal{A}$			Dataset $\mathcal{B}$		
	Homo.	Comp.	V Meas.	Homo.	Comp.	V Meas.
Prism	0.976	0.916	**0.945**	0.979	0.922	**0.950**
Prism w/o Metrics	0.462	**0.920**	0.615	0.463	**0.949**	0.622
Prism w/o Traces	0.949	0.869	<u>0.907</u>	0.915	0.893	<u>0.904</u>

- Prism **outperforms all state-of-the-art comparative methods**.
- **Both components** contribute to the overall performance.

Evaluation

- RQ3: Parameter Sensitivity



- Prism is robust to threshold settings for both LSH and HAC.

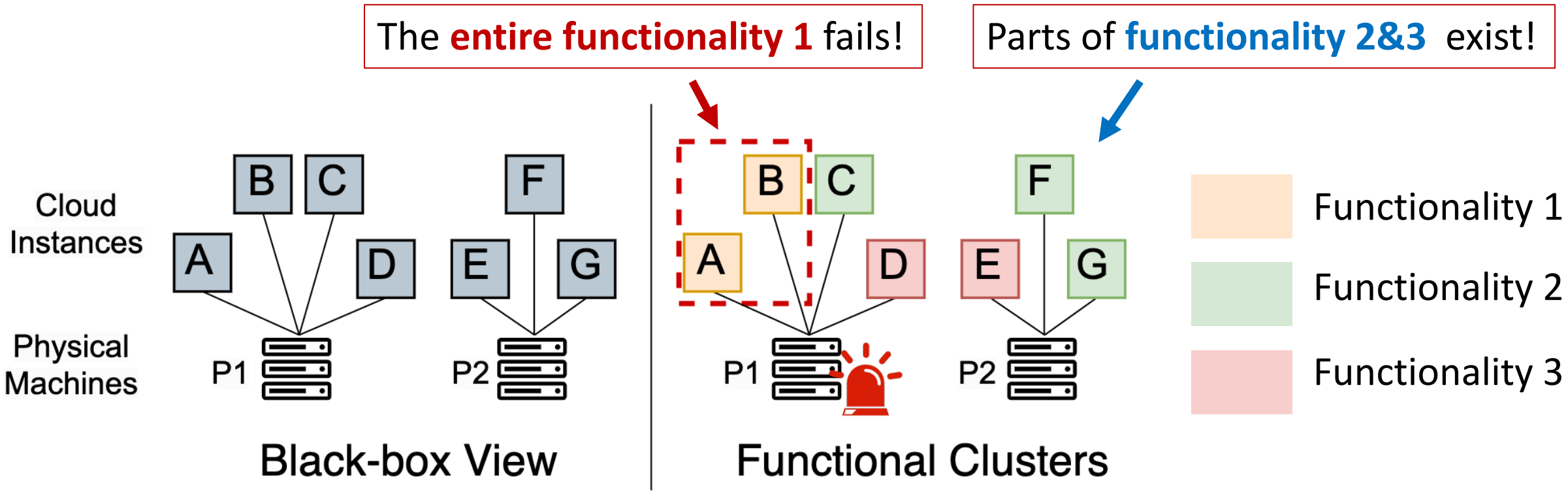
- RQ4: Efficiency

Methods	# Instances				
	1,000	5,000	10,000	50,000	100,000
CloudCluster	0.9	23.87	78.65	1768.7	5585.7
ROCKA	80.7	1981.8	7850.3	-	-
OmniCluster	31.7	264.6	1048.6	26531.8	-
Prism w/o Metrics	3.9	19.1	40.2	195.1	392.4
Prism w/o Traces	80.3	2066.1	8232.3	-	-
Prism	18.2	89.4	183.9	929.2	1912.7

- Prism can efficiently handle massive instances in cloud systems.

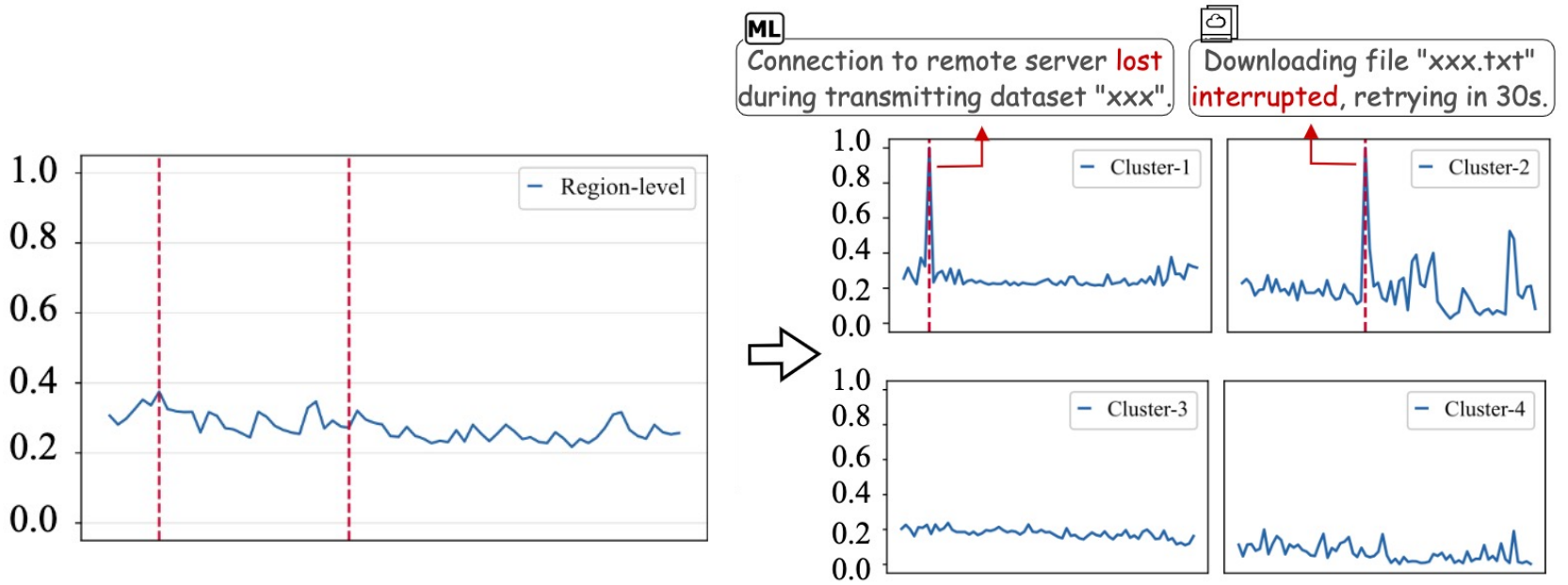
Industrial Experience

- Use case 1: vulnerable deployment identification



Industrial Experience

- Use case 2: latent issue discovery



Aggregated by **region**

Aggregated by **functional clusters (Prism applied)**

Fragmented lost packets of massive instances

Conclusion

- Cloud vendors struggle to ensure the reliability of large virtual instances due to limited observability.
- The proposed **Prism** reveals functional clusters by leveraging communication patterns and resource patterns among instances.
- Prism is effective and efficienct, which provides insights for enhanced cloud monitoring.

Thank you!

ARISE Lab Me

Find code & dataset in **OpsPAI** (IT operations powered by AI).

https://github.com/OpsPAI/

This work!

Observation

Prism

- Functional Cluster Identification of Massive VMs

[ASE'23]

AID

- Dependency Analysis for Micro-services

[ASE'21]

Anomaly Detection

ADSketch

- Adaptive Time-series Anomaly Detection

[ICSE'22]

MTAD

- Tools and Benchmark for Multivariate Time Series Anomaly Detection

Incident Aggregation

iPack

- Incident-aware Duplicate Ticket Aggregation

[ICSE'23]

GRLIA

- Graph-based Incident Aggregation

[ASE'21]

ARISE Automated Reliable Intelligent Software Engineering

香港中文大學 The Chinese University of Hong Kong

中山大學 SUN YAT-SEN UNIVERSITY

HUAWEI