

# A Case For Cross-Domain Observability to Debug Performance Issues in Microservices

Ranjitha K, Praveen Tammana,

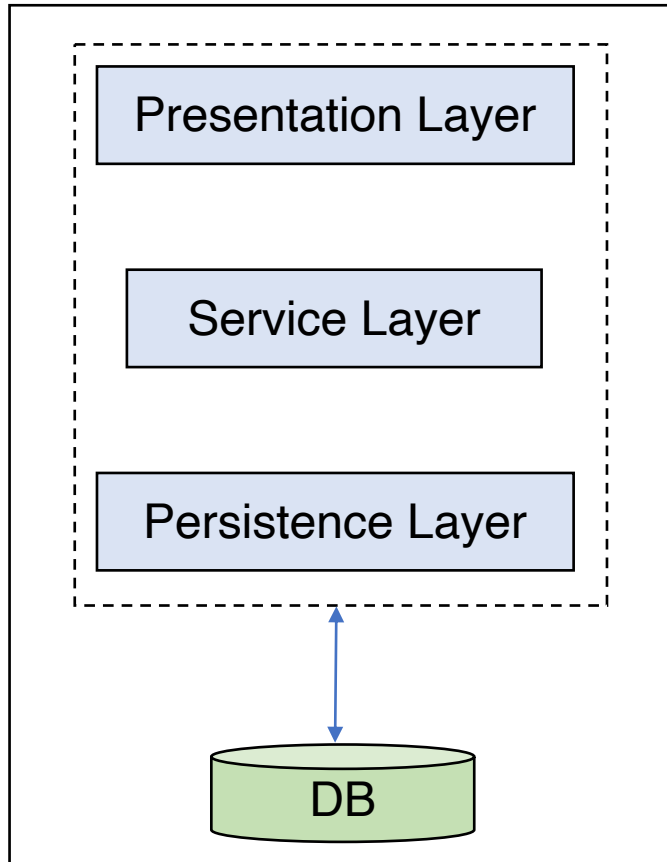
**Pravein Govindan Kannan**, Priyanka Naik



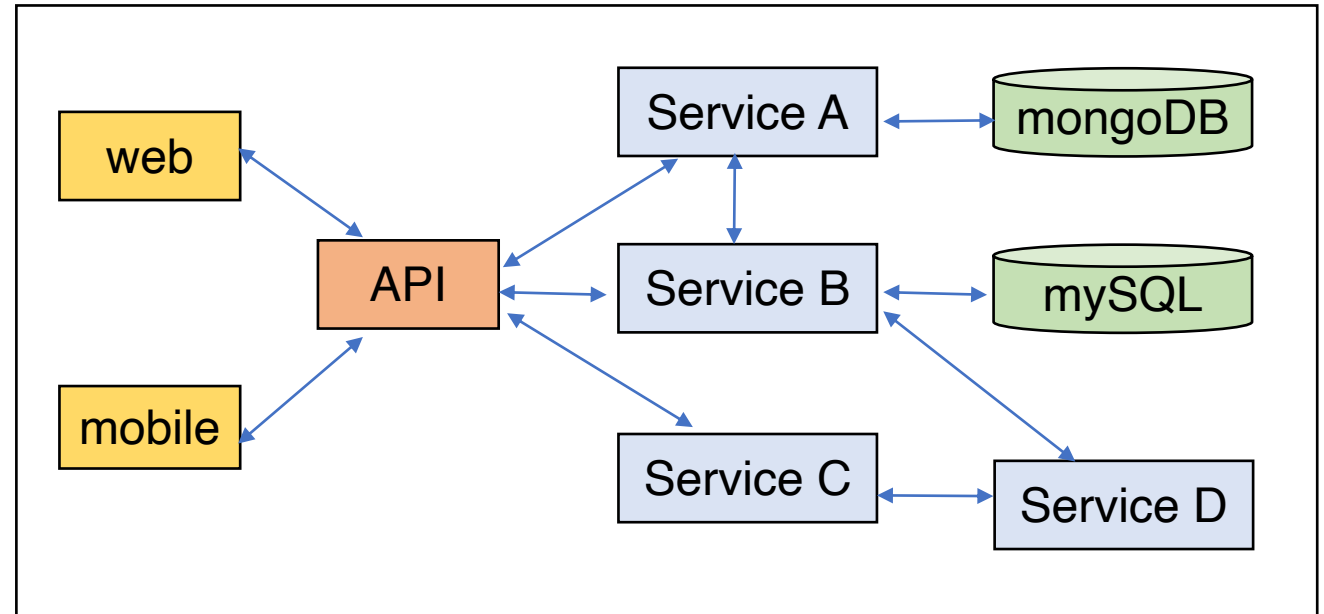
भारतीय प्रौद्योगिकी संस्थान हैदराबाद  
Indian Institute of Technology Hyderabad

**IBM Research**

# Cloud Deployments - Microservices

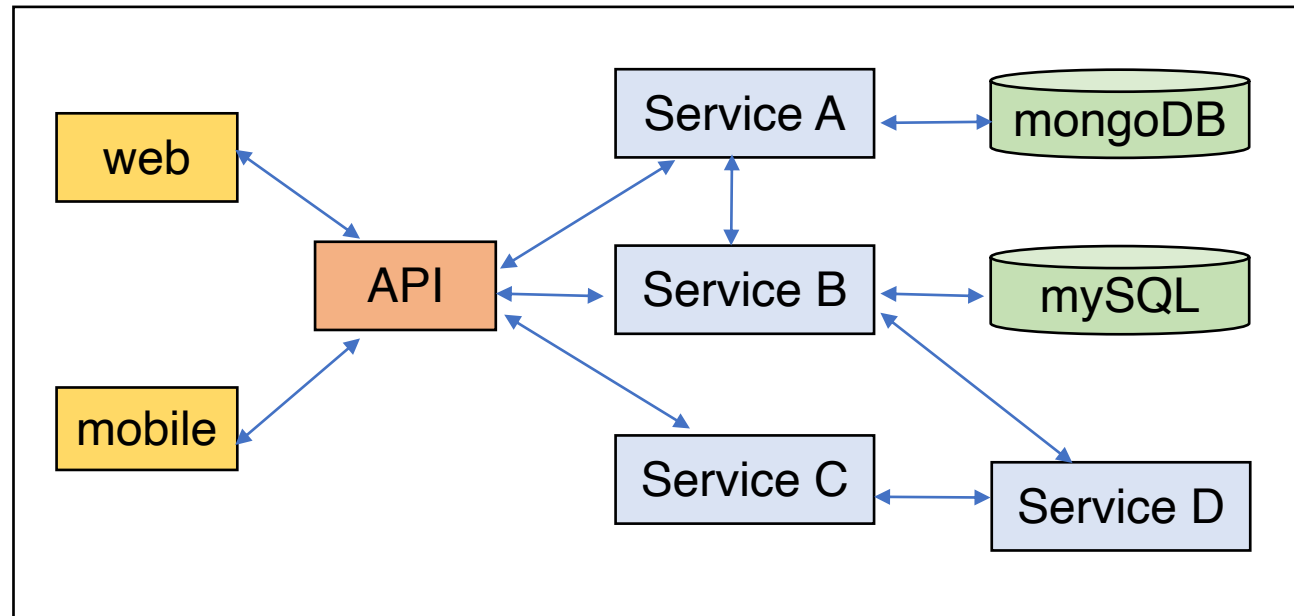


Monolithic Architecture



Microservices Architecture

# Cloud Deployments – SLA Violations!



Kubernetes

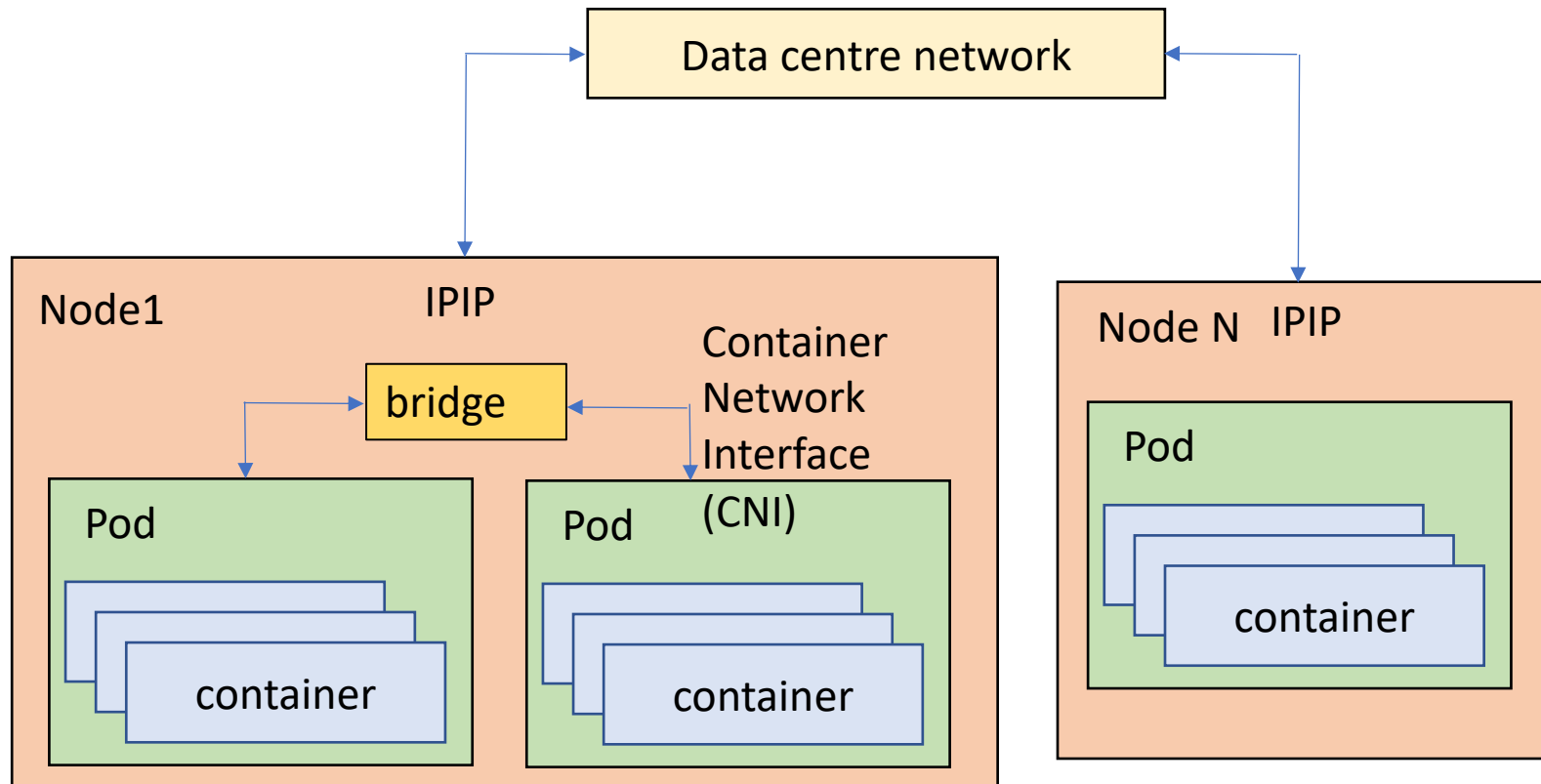


OPENSIFT



openstack.

# Network Connectivity in Microservices



Kubernetes



OPENSIFT



openstack.

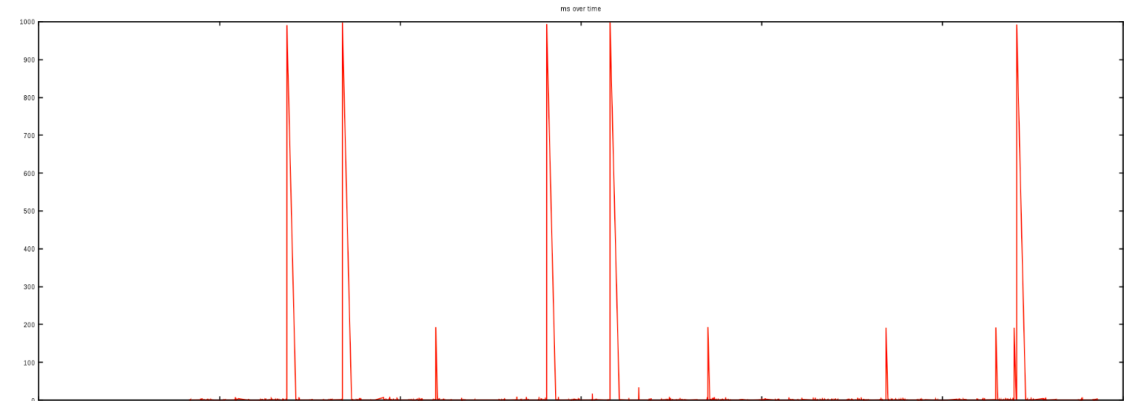
# Performance Issues

- Sporadic increase in latencies
- 36% of performance anomalies are Transient [Bufscope, NSDI '22]
- Reasons could be :
  - On any of the nodes involved :
    - NAT, load-balancer, sender, receiver, etc.
      - IPTables configuration
      - CPU scheduling
      - NIC Queueing
    - Network links
      - Congestion
      - Microbursts
      - Link Failures
      - Packet corruption

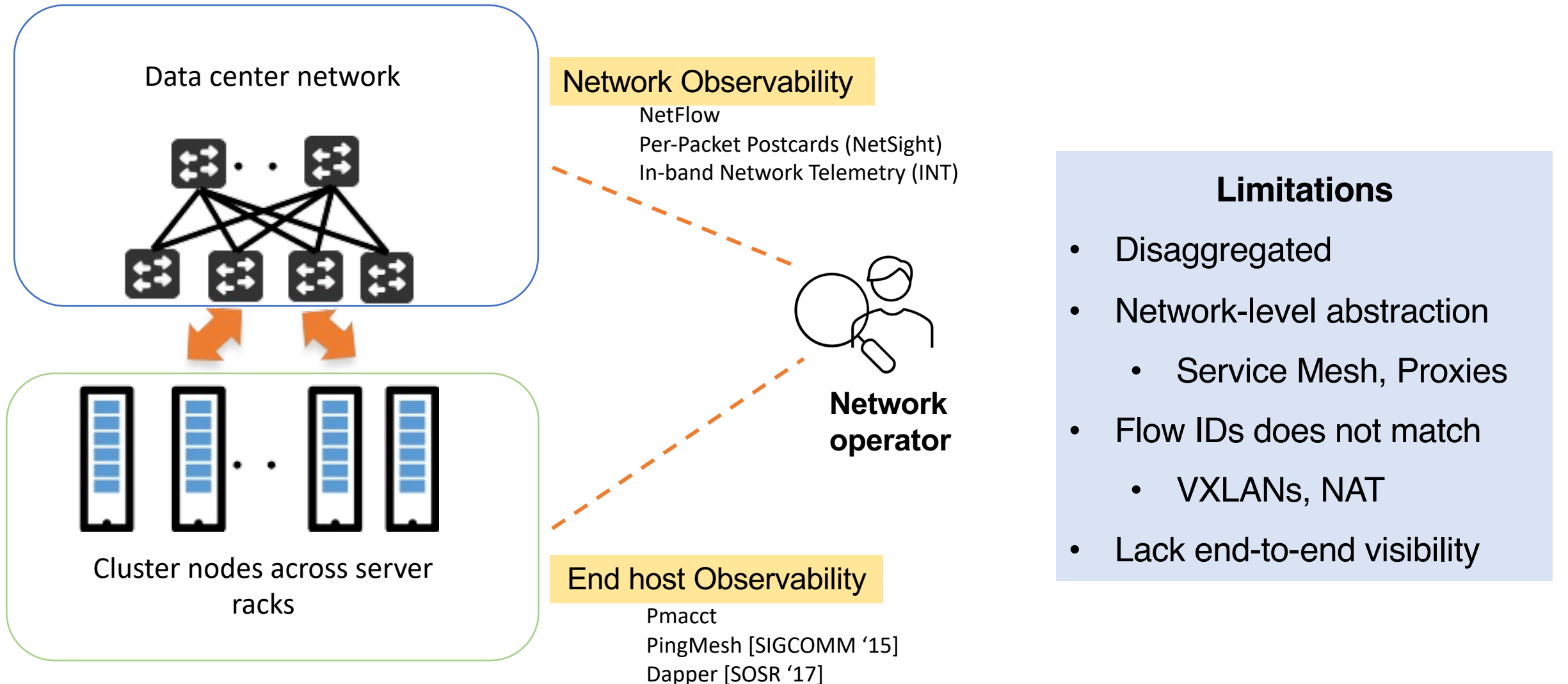
Facebook Microbursts [IMC'17]

<https://github.blog/2019-11-21-debugging-network-stalls-on-kubernetes/>

<https://blog.cloudflare.com/the-story-of-one-latency-spike/>

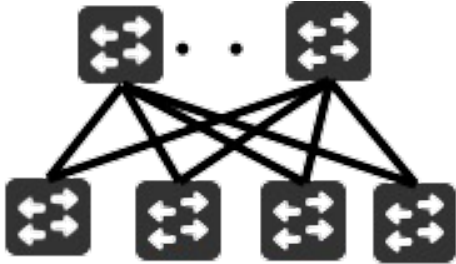


# Need for end-to-end observability

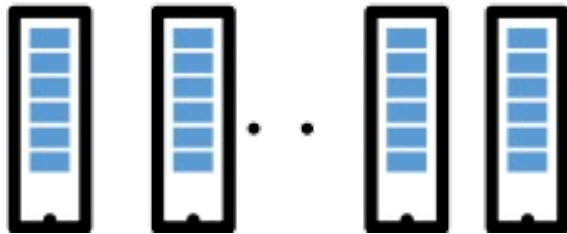


Aggregating information and performing root cause analysis can be slow, inaccurate and misleading.

Data center network



Network Observability



Cluster nodes across server racks

End host Observability

Is it possible to design and efficient performance monitoring framework that can achieve **end-to-end (cross-domain) Observability?**

# Design

## Enhance Host-observability:

- **Monitoring Primitive**

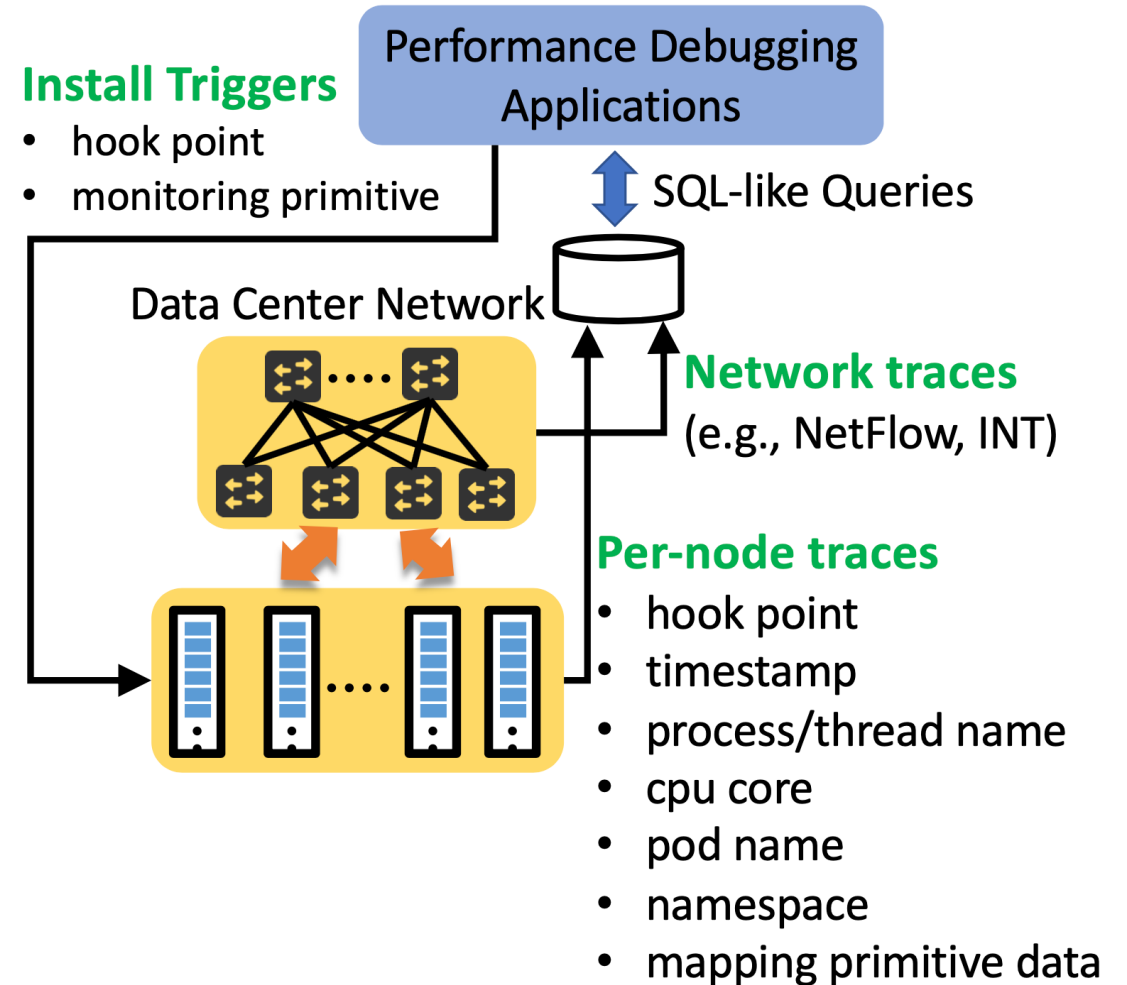
- RTT increase
- Packet Drops

- **Tracer**

- Collect Host-metrics (TPs, Socket, TC, etc)
- Maintain recent history

- **Mapping Primitive**

- Container flow-IDs to Node flow-IDs





# Prototype Implementation

- **Monitoring Primitive [eBPF<sup>1</sup>-based]**

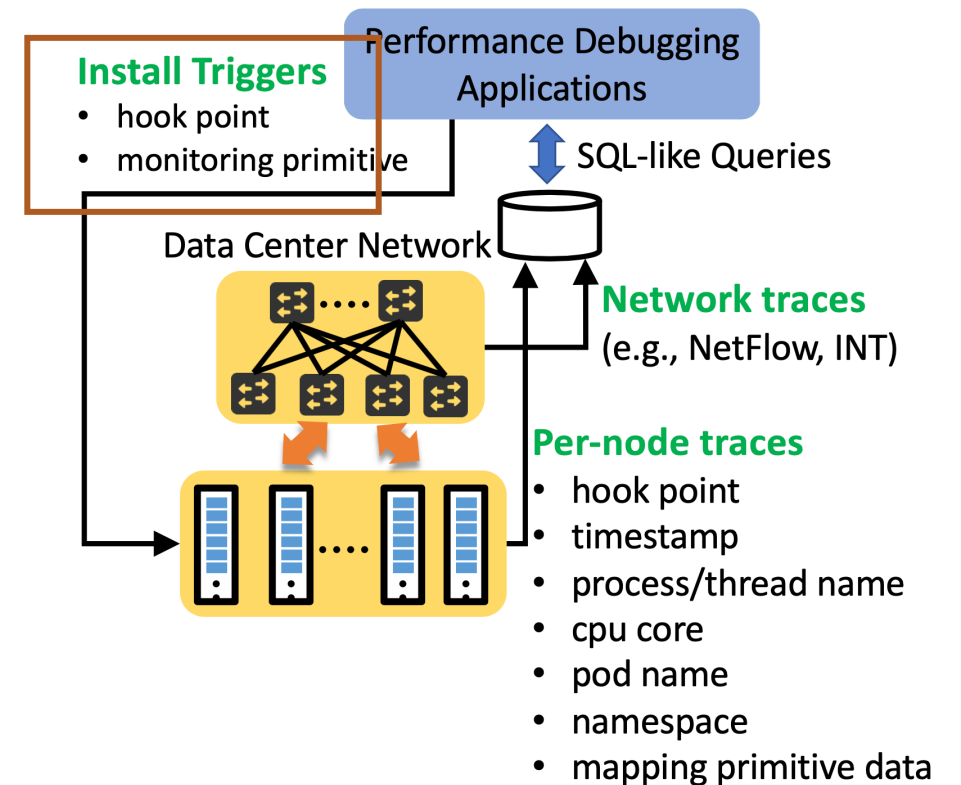
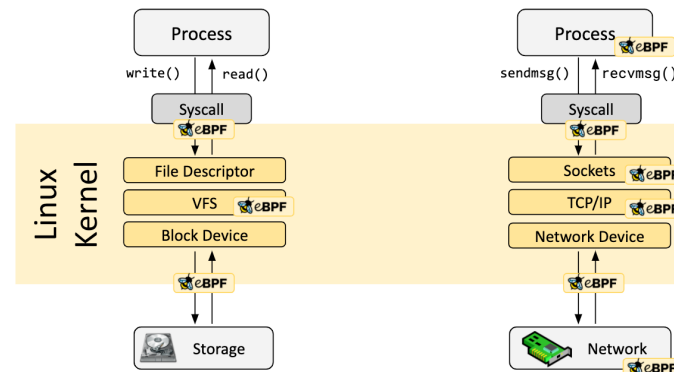
- RTT monitoring for TCP Flows
- Stateful monitoring of Seq/ack-seq
- Per-CPU LRU Hash to maintain

<Seq, timestamp>

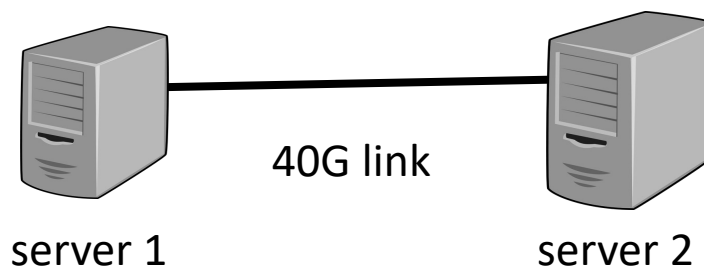
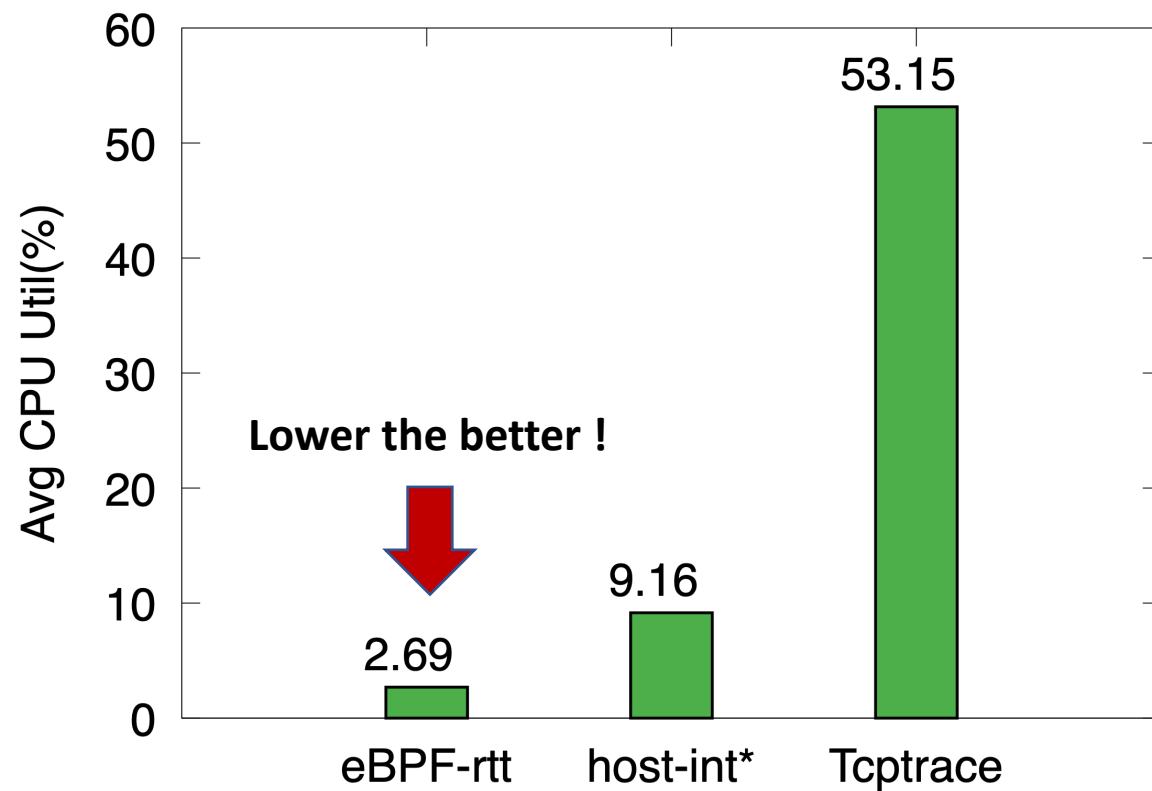
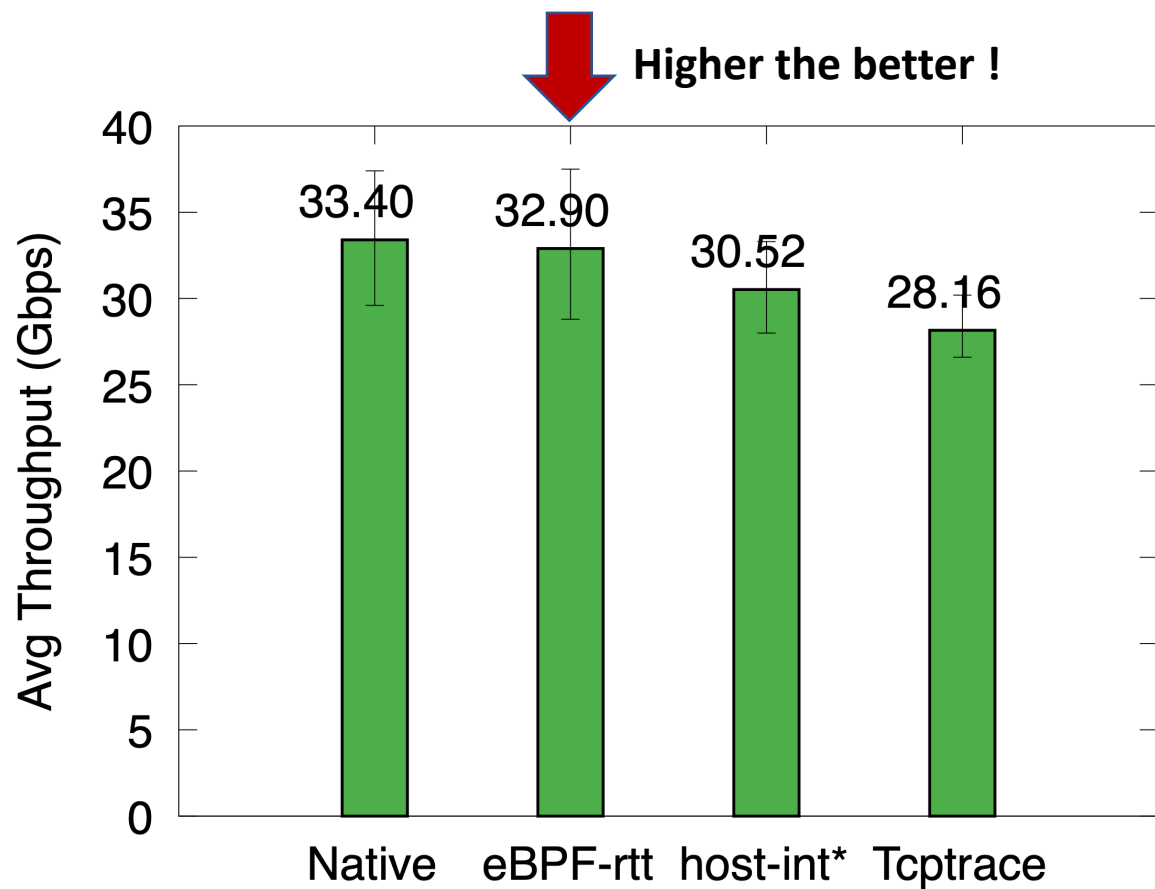
- Per-flow Moving average of RTT

- **Trigger:**

- Upon Increase of avg RTT by x%
- Threshold



# Evaluation



# Ongoing Work

- *Tracer :*
  - *Maintains continuous list of events (syscalls, timestamps)*
  - *Ringbuffer-based recent history*
  - *eBPF/Intel-PT*
- *Mapping Primitive*
  - *eBPF-based flow mapping*
  - *Monitor vETHs and outgoing interfaces*
- *Evaluate on a larger setup*

# Conclusion

- We present a case to build cross-domain observability framework to debug performance issues.
- Feasibility of the system by implementing monitoring primitive.
- eBPF-based RTT monitoring with low overhead.

**Thank You !**

**Contact** : Pravein.Govindan.Kannan@ibm.com