Arena: A Kubernetes-based Testbed for Evaluating Application Deployment across the Computing Continuum

Chih-Kai Huang[123], Konstantinos Krouti[4], Stella Markopoulou[4], Konstantinos Tserpes[5], and Georgios Bouloukakis[62]

1 Telecom Paris, France
2 Telecom SudParis, France
3 Institut Polytechnique de Paris, France
4 AEGIS IT Research GmbH, Germany
5 National Technical University of Athens, Greece
6 University of Patras, Greece

chih-kai.huang@telecom-paris.fr

26 May 2026, IEEE International Conference on Communications 2026 (ICC)

Introduction and Background

Internet of Things (IoT)

Growing number of IoT devices

Forecast: **40.6 billion** devices by 2034 [1]

Data Processing

IoT generates massive data that need to be processed

⚠ Centralized cloud processing faces **latency**, **bandwidth**, and **responsiveness** challenges

Edge Computing

Processes data closer to the source

Reduces **latency** and saves **bandwidth**

Improves **real-time responsiveness**

[1] statista. (2025) Number of Internet of Things (IoT) connections worldwide. [Online]. Available: https://reurl.cc/Om0deg

Introduction and Background

Limitations of Edge

Limited compute and storage capacity

⚠ Cannot handle **large and compute-intensive tasks** alone

Edge and Cloud Collaboration

Combine **Edge responsiveness** with **Cloud's computing power and storage**

Forms the concept of the **Computing Continuum**

Computing Continuum

Seamless integration across **IoT**, **Edge**, and **Cloud**

Dynamically distributes workloads based on **needs and context**

Motivation and Related Work

Computing Continuum is promising, but realizing its potential requires evaluating how applications behave across IoT, Edge, and Cloud.

Approaches

Direct Deployment on Real Infrastructure

- ✓ **High realism using real devices and networks**
- ✗ **Expensive and time-consuming to deploy**
- ✗ **Difficult to control the environment**

Example : FIT IoT-LAB [2] (Edge) + Grid5000 [3] (Cloud)
Example : Homemade Raspberry Pi + Google Cloud

[2] C. Adjih et al., “Fit IoT-Lab: A Large Scale Open Experimental IoT Testbed,” in IEEE World Forum on Internet of Things (WF-IoT), 2015.
[3] R. Bolze et al., “Grid’5000: A Large Scale and Highly Reconfigurable Experimental Grid Testbed,” The International Journal of High Performance Computing Applications, vol. 20, no. 4, 2006.

Motivation and Related Work

Computing Continuum is promising, but realizing its potential requires evaluating how applications behave across IoT, Edge, and Cloud.

Approaches

Simulation-Based Evaluation

- ✓ **Fast, scalable, and easy to run**
- ✗ **Lacks system-level realism**
- ✗ **Cannot test real applications or runtime behaviour**

Example : iFogSim2 [4]
Example : YAFS [5]

[4] R. Mahmud et al, "IFogSim2: An Extended iFogSim Simulator for Mobility, Clustering, and Microservice Management in Edge and Fog Computing environments," Journal of Systems and Software, vol. 190, 2022.
[5] I. Lera et al, "YAFS: A Simulator for IoT Scenarios in Fog Computing," IEEE Access, vol. 7, 2019.

Motivation and Related Work

Computing Continuum is promising, but realizing its potential requires evaluating how applications behave across IoT, Edge, and Cloud.

Approaches

Emulation-Based Approach

- ✓ **Runs real applications under different conditions/failures**
- ✓ **High controllability**
- ✗ **Slightly less scalable than pure simulation**

In this work, we adopt an emulation-based approach to build Arena

Limitations of Existing Emulation Platforms

Fogbed relies on the Docker and MiniNet [6]

↳ Does not support dynamic application behaviors or orchestration

EmuFog relies on the Docker and MaxiNet [7]

↳ Lack of ecosystem support

iContinuum relies on Kubernetes, ONOS, and Mininet [8]

↳ Higher setup complexity (SDN) and resource overhead (VM-based)

Limitations

Limited orchestration support	**Limited ecosystem maturity**	**High deployment overhead**

[6] A. Coutinho et al,"Fogbed: A Rapid-prototyping Emulation Environment for Fog Computing," in IEEE International Conference on Communications (ICC), 2018.
[7] R. Mayer et al,"EmuFog: Extensible and Scalable Emulation of Large-scale Fog Computing Infrastructures," in IEEE Fog World Congress (FWC), 2017.
[8] N. Akbari et al, "iContinuum: An Emulation Toolkit for Intent-based Computing across the Edge-to-Cloud Continuum," in IEEEInternational Conference on Cloud Computing (CLOUD), 2024.

Proposed Solution

Arena: A Kubernetes-based Testbed for Evaluating Application Deployment across the Computing Continuum

Arena addresses these gaps through:

Dynamic Orchestration

Real applications are deployed and managed through Kubernetes.

Ecosystem Compatibility

Experiments can use standard Kubernetes-native tools and workflows.

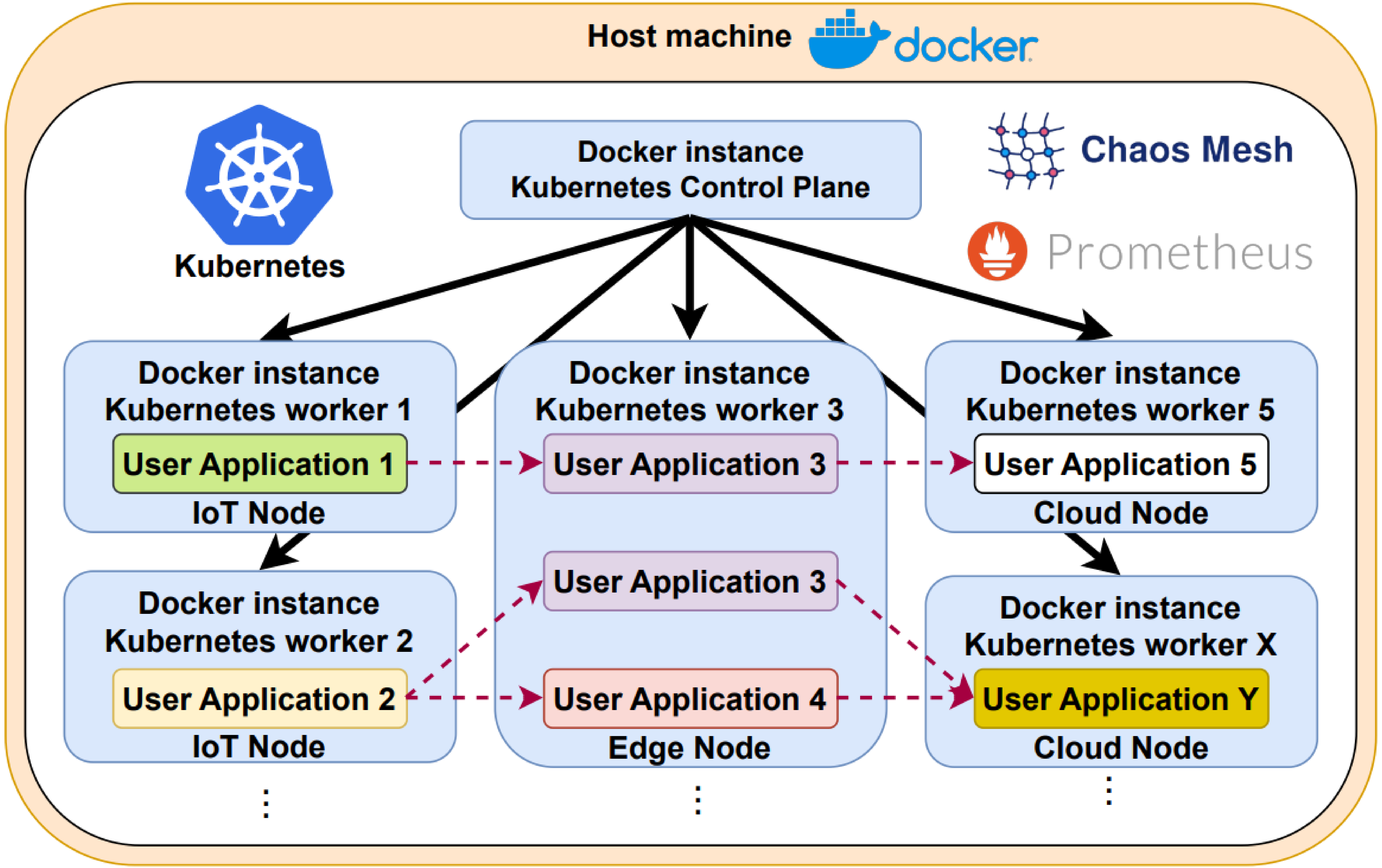
Container-as-Node Emulation

Each computing node is emulated as a lightweight container.

Kubernetes-native Fault Injection

Chaos Mesh injects delay and failures without relying on an external SDN stack.

System Overview



Key Components

Docker

Kubernetes

Chaos Mesh

Prometheus

Key Components – Docker and Kubernetes

Docker

In Arena, Docker is used to:

- Lightweight container-based computing nodes
- Configurable heterogeneous node resources
- Enable portable deployment without VM-based infrastructure

Kubernetes

In Arena, Kubernetes is used to:

- Applications are deployed as Kubernetes workloads
- Kubernetes orchestrates services across emulated computing nodes
- Standard Kubernetes workflows enable reproducible deployment experiments

Key Components – Docker and Kubernetes

Chaos Mesh

Prometheus

In Arena, Chaos Mesh is used to:

- Injects network delay, loss, and bandwidth constraints
- Introduces workload failures under unstable conditions
- Applies resource stress to evaluate application resilience

In Arena, Prometheus is used to:

- Kubernetes-native metric collection
- Runtime observation of application and resource behavior
- Time-series analysis for deployment evaluation

Arena in Action

1. **Define Continuum Topology**

Configure IoT, Edge, and Cloud nodes with heterogeneous resource profiles.

2. **Deploy Real Applications**

Deploy application components as Kubernetes workloads across emulated nodes.

3. Inject Runtime Conditions

Use Chaos Mesh to introduce network instability, workload failures, or resource stress.

4. Collect and Analyze Metrics

Use Prometheus to monitor application performance and resource behavior.

Evaluation of Arena Testbed

Environment Setup

Platform: Grid'5000 large-scale testbed

Host server: Dual Intel Xeon E5-2630L v4, 20 cores total, 128 GB RAM

OS: Debian 11.11, Linux kernel 5.10.0-35-amd64

Core stack: Kubernetes v1.33.2, Docker v28.0.4, containerd v2.1.3, Cilium v1.17.6

Arena components: kube-prometheus-stack v75.18.1, Chaos Mesh v2.28.0, QEMU v5.2.0

Repetition: Each experiment repeated 5 times; average values reported

Emulation Fidelity: Container-based vs. VM-based Nodes

Experimental Setup

- **Compare Arena container-based nodes with VM nodes**
- **Use the same Kubernetes configuration and Online Boutique application**
- **Measure CPU and memory usage under different user loads**
- **Validate whether containers preserve VM-like resource-usage behavior**

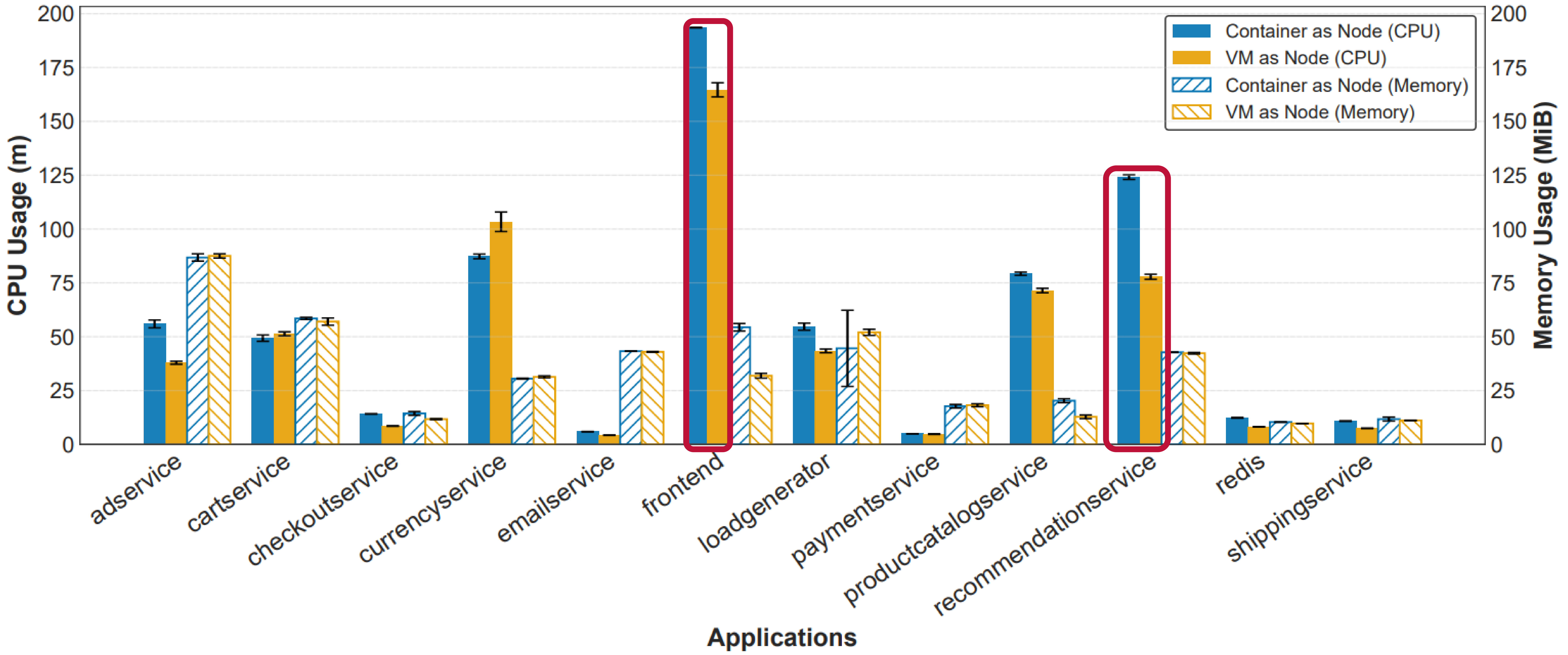
TABLE I
MICROSERVICES AND THEIR PROGRAMMING LANGUAGES

Service	Language
adservice	Java
cartservice	C#
checkoutservice	Go
currencyservice	Node.js
emailservice	Python
frontend	Go
loadgenerator	Python/Locust
paymentservice	Node.js
productcatalogservice	Go
recommendationservice	Python
redis	C
shippingservice	Go

Emulation Fidelity: Container-based vs. VM-based Nodes

Results

59.5% higher - 46.3m (0.0463 core)



CPU and memory utilization of twelve microservices running on container-based and VM-based nodes under a workload of 200 virtual users.

Validation of Network Chaos Injection

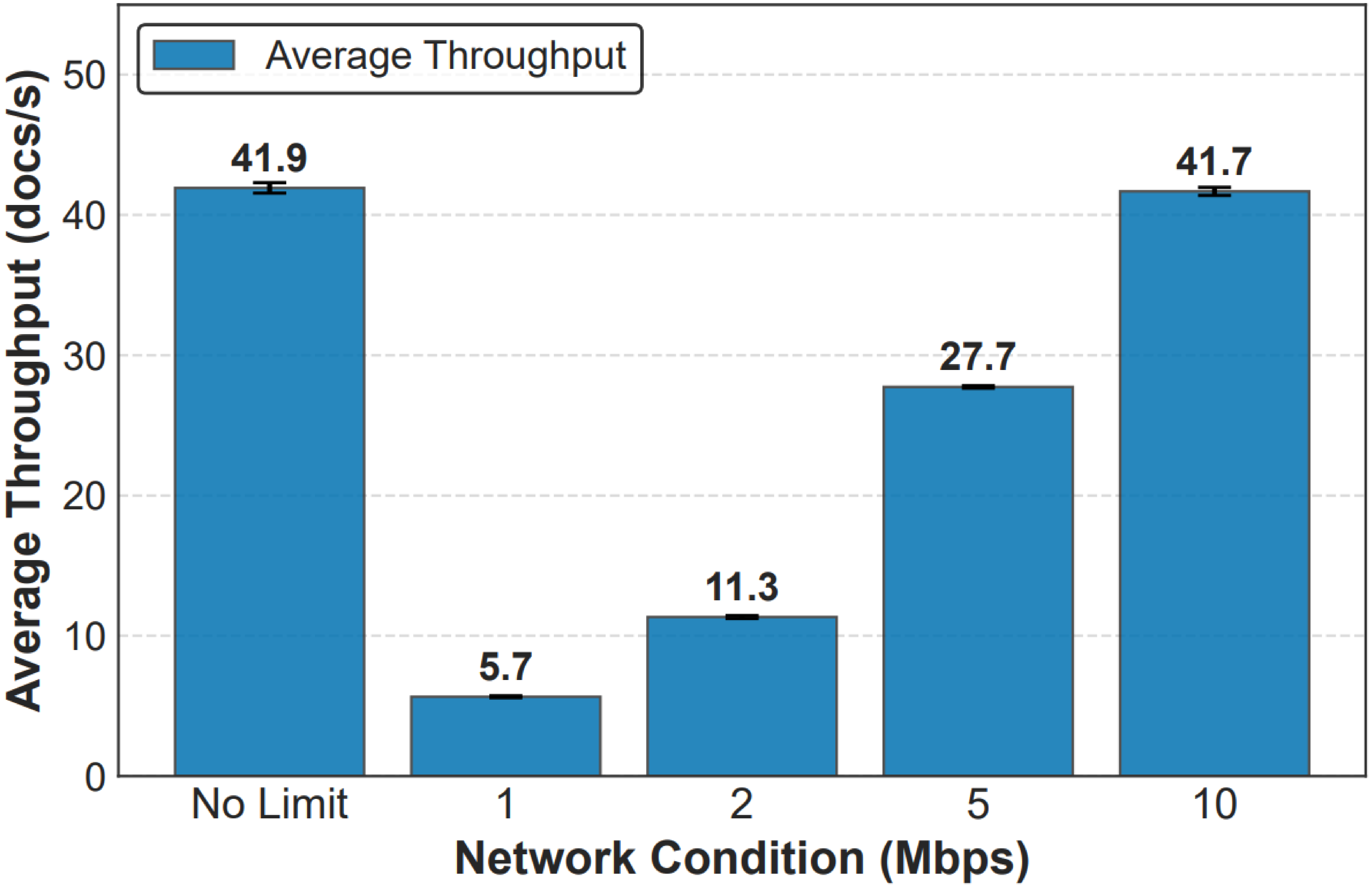
Experimental Setup

- **Deploy data pipeline applications**
- **Inject bandwidth limits with Chaos Mesh**
- **Test controlled rates: 1, 2, 5, and 10 Mbit/s**
- **Measure ingestion throughput over 10 minutes**
- **Validate Arena’s network-condition control**

IoT data producers → **Kafka (v3.7.0)** → **Elasticsearch (v8.17.3)**

Validation of Network Chaos Injection

Results



Average throughput of the data pipeline under different outbound bandwidth limitations applied to Logstash.

Conclusions and Future Work

Conclusions

- Arena enables Kubernetes-native experimentation across the computing continuum.
- Container-as-node emulation provides VM-like resource-usage trends.
- Integrated orchestration, fault injection, and monitoring support deployment evaluation.

Future Work

- Scale Arena to multi-host deployments.
- AI-agent-assisted experiment automation
- Extend hardware diversity, e.g., ARM and GPU nodes.

Chih-Kai HUANG
Assistant/Associate Professor at Telecom Paris / IP Paris
chih-kai.huang@telecom-paris.fr

Supported by

PANDORA

MEDIATE