Pilot-Edge: Distributed Resource Management Along the Edge-to-Cloud Continuum

Andre Luckow[1], Kartik Rattan[2] and Shantenu Jha[2,3]

[1]Ludwig-Maximilian University, Munich, Germany
[2]RADICAL, ECE, Rutgers University, Piscataway, NJ 08854, USA
[3]Brookhaven National Laboratory, Upton, NY, USA

https://arxiv.org/abs/2104.03374

Motivation

The **Internet of Things (IoT)** is becoming an essential part of many scientific and industry applications, e.g., light source science, farming, and manufacturing.*

These applications require high performance compute and data capabilities across multiple layers of infrastructures from the **edge-to-cloud**.

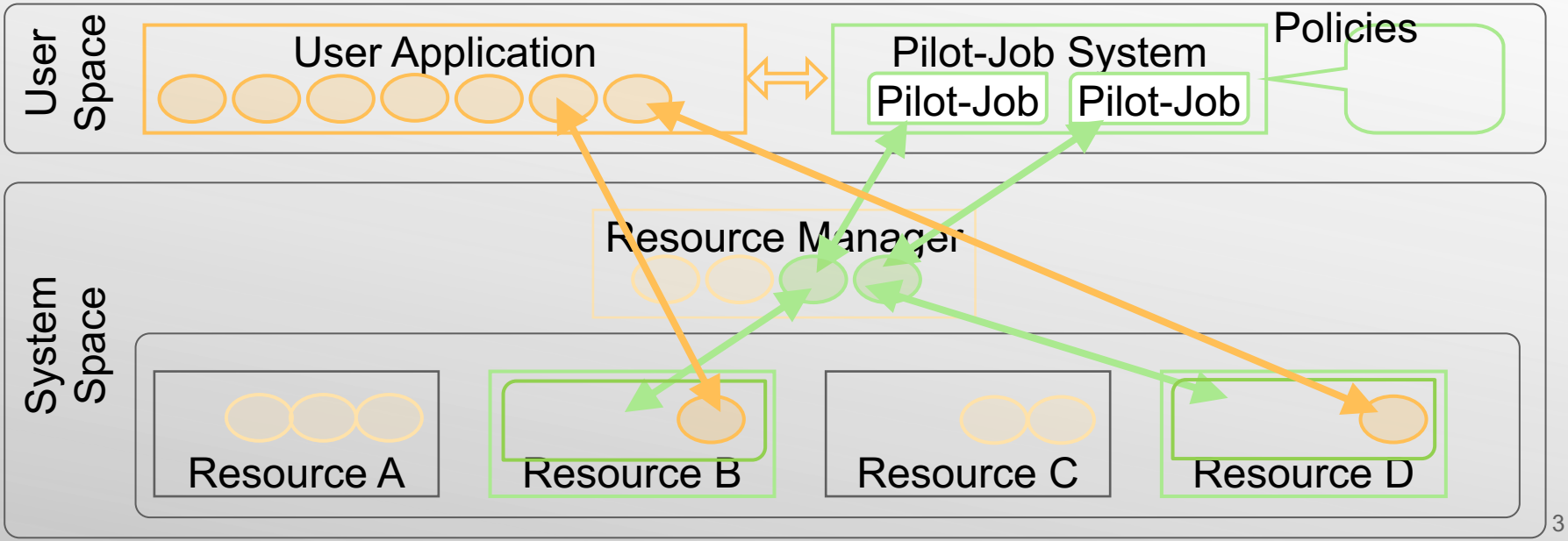
Main Challenges:

- **Heterogeneity:**
  - **Infrastructures and devices:** instruments, edge, fog, HPC, cloud, serverless, accelerators need to be integrated
  - **Different programming models:** Scripting, Serverless (FaaS), HPC (MPI, OpenMP, CUDA), Data (MapReduce, Streaming)
- **Dynamic and distributed environment:** Data sources (IoT devices) geographically distributed, environment constantly changing
- **Provisioning and management of resources and tasks:** right amount of resources at right time, optimal execution strategy for applications

*see: https://arxiv.org/abs/2104.03368

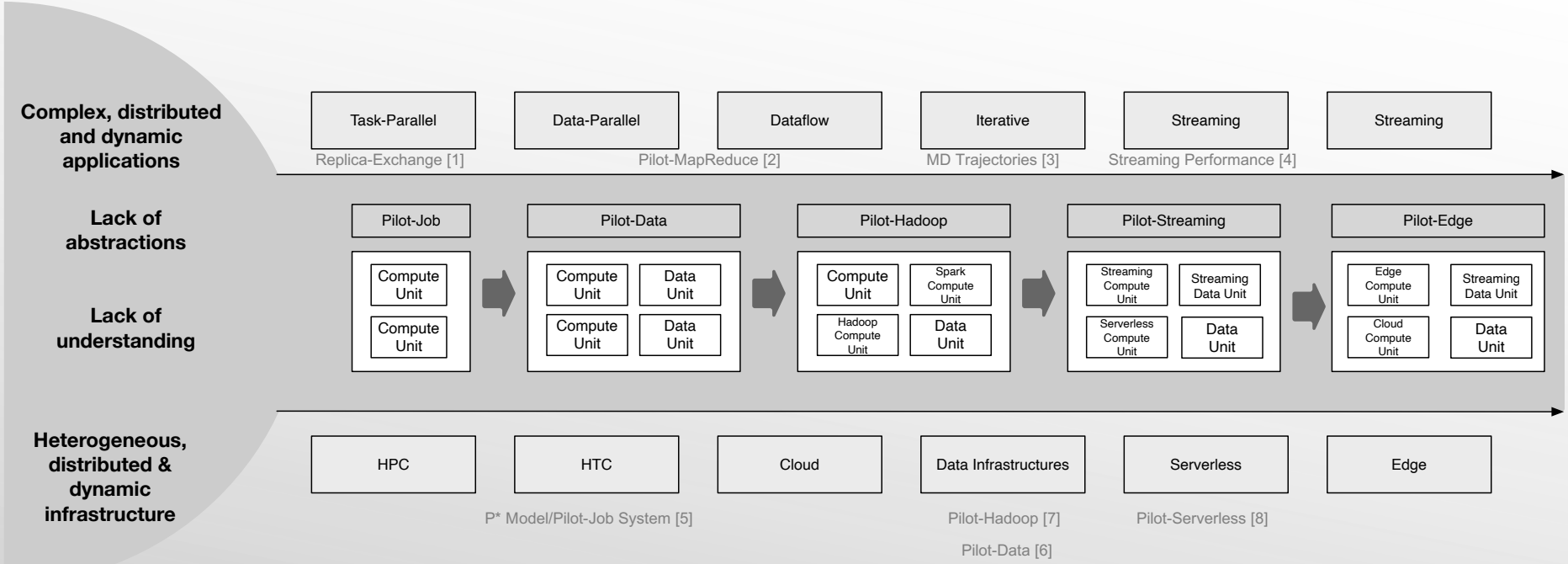
Pilot-Abstraction: A common resource management abstraction for data and compute

Working Definition: A system that generalizes a placeholder job to provide multi-level scheduling to allow application-level control over the system scheduler via a scheduling overlay.



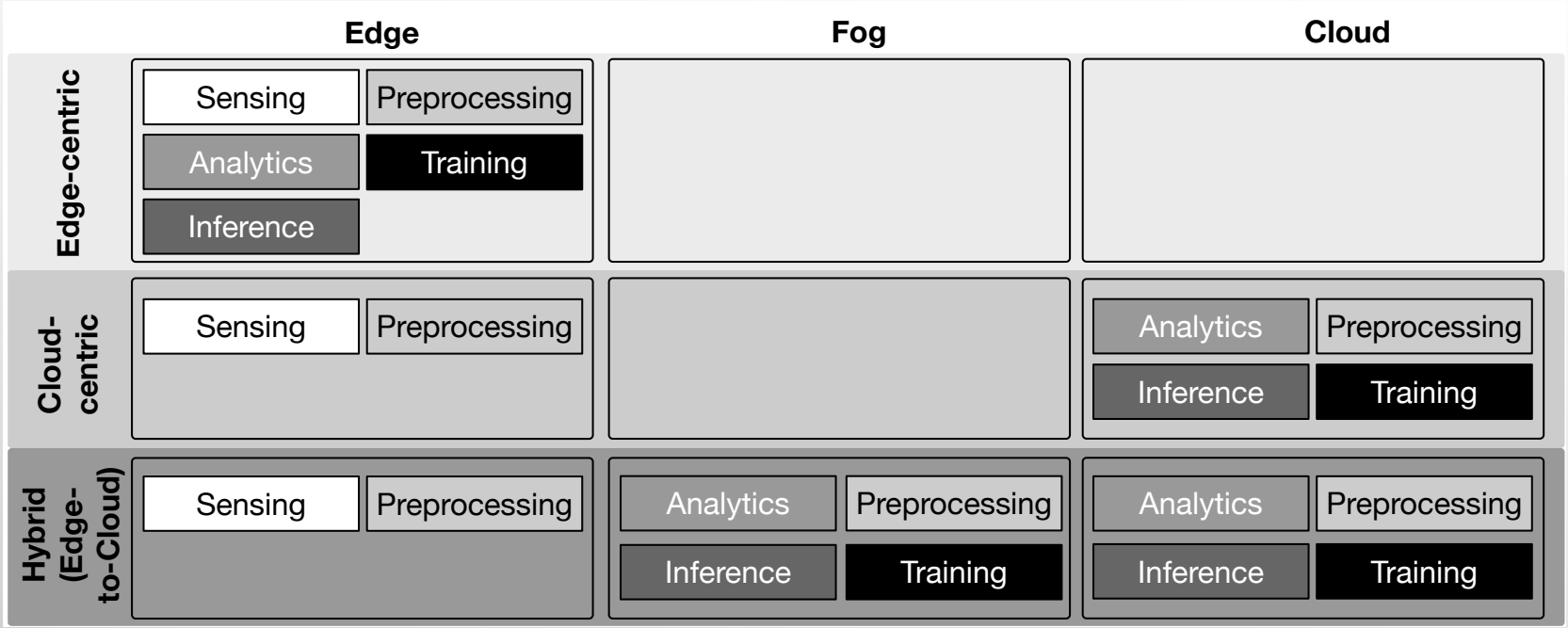
Luckow et al.: P* A Model of Pilot-Abstractions, IEEE eScience 2012, http://arxiv.org/abs/1207.6644

Pilot-Abstraction: Evolution of the Abstraction

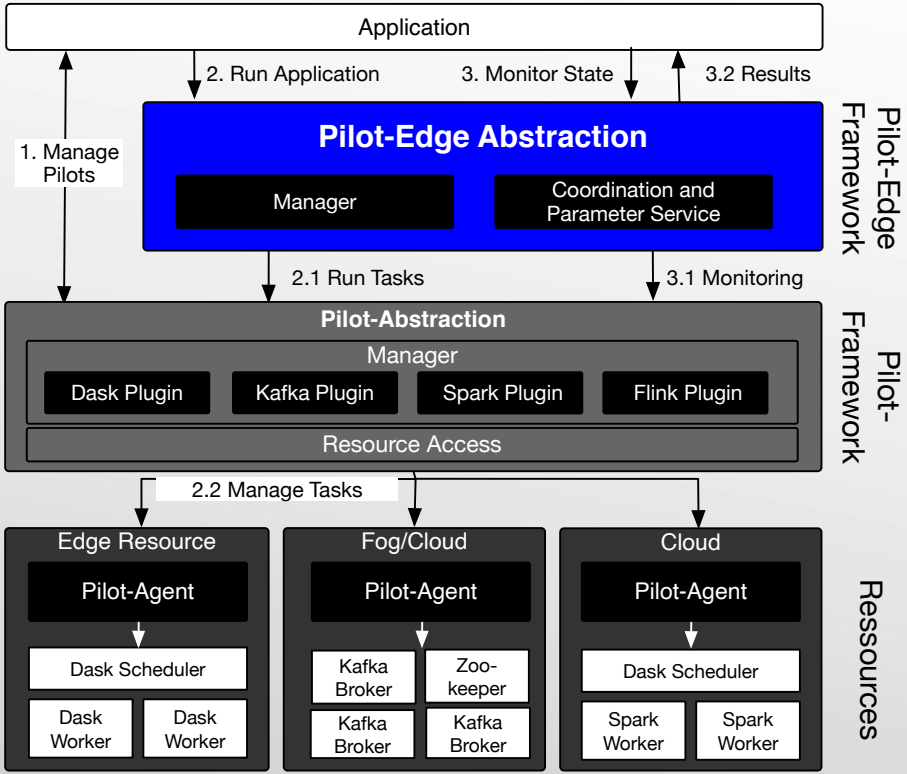


[1] Andre Luckow, Shantenu Jha, Joohyun Kim, Andre Merzky, and Bettina Schnor. Adaptive Replica-Exchange Simulations. Royal Society Philosophical Transactions A, 2009.
[2] Pradeep Kumar Mantha, Andre Luckow, and Shantenu Jha. Pilot- MapReduce: An Extensible and Flexible MapReduce Implementation for Distributed Data. In Proceedings of third international workshop on MapReduce and its Applications, MapReduce '12, pages 17–24, New York, NY, USA, 2012. ACM.
[3] Ioannis Paraskevakos, Andre Luckow, Mahzad Khoshlessan, George Chantzialexiou, Thomas E. Cheatham, Oliver Beckstein, Geoffrey C. Fox, and Shantenu Jha. Task-parallel analysis of molecular dynamics trajectories. In Proceedings of the 47th International Conference on Parallel Processing, ICPP 2018, New York, NY, USA, 2018. ACM.
[4] Andre Luckow and Shantenu Jha. Performance characterization and modeling of serverless and hpc streaming applications. In Proceedings of StreamML Workshop at IEEE International Conference on Big Data (IEEE BigData 2019), 2019.
[5] Andre Luckow, Mark Santcroos, Andre Merzky, Ole Weidner, Pradeep Mantha, and Shantenu Jha. P*: A model of pilot-abstractions. IEEE 8th International Conference on e-Science, pages 1–10, 2012. http://dx.doi.org/10.1109/eScience.2012.6404423.
[6] Andre Luckow, Mark Santcroos, Ashley Zebrowski, and Shantenu Jha. Pilot-data: An abstraction for distributed data. Journal of Parallel and Distributed Computing, 2014.
[7] Andre Luckow, Pradeep Kumar Mantha, and Shantenu Jha. Pilot- abstraction: A valid abstraction for data-intensive applications on hpc, hadoop and cloud infrastructures? CoRR, abs/1501.05041, 2015.
[8] Andre Luckow, George Chantzialexiou, and Shantenu Jha. Pilot-streaming: A stream processing framework for high-performance computing. IEEE eScience, 2018.

Edge-to-Cloud Applications: Common Deployment Modalities



Pilot-Edge Architecture and Interactions



Pilot-Edge API and Abstraction

Listing 1: Pilot-Edge FaaS API

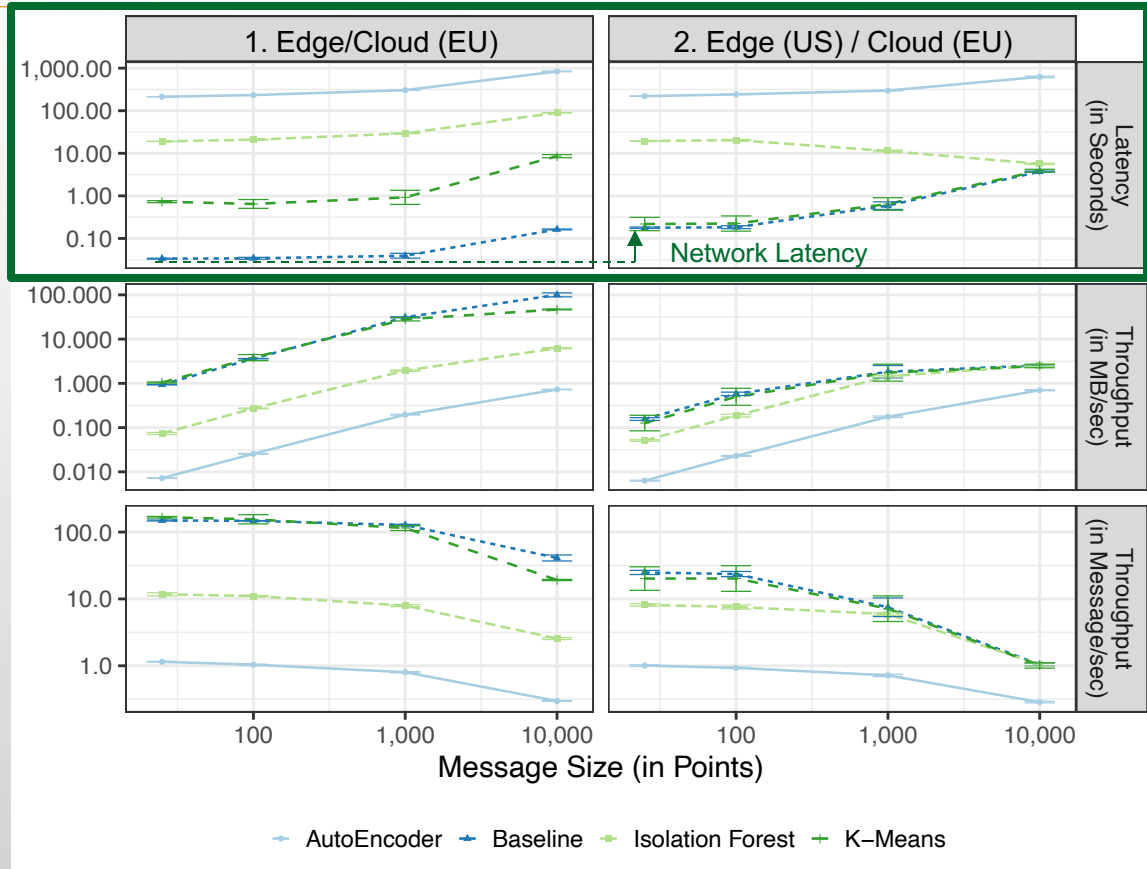
```
def produce_edge(context)

def process_edge(context: dict = None, data=None)

def process_cloud(context: dict = None, data=None)
```

Listing 2: Pilot-Edge API: Instantiation of an Application

```
pilot.EdgeToCloudPipeline  (
  pilot_cloud_processing=pilot_job_cloud_processing,
  pilot_cloud_broker=pilot_job_cloud_broker,
  pilot_edge=pilot_job_edge,
  produce_function_handler=produce_block_edge,
  process_edge_function_handler=process_block_edge,
  process_cloud_function_handler=process_block_cloud,
  function_context=context,
  ...
  ).run()
```

Evaluation of Pilot-Edge using Edge-to-Cloud Machine Learning Application

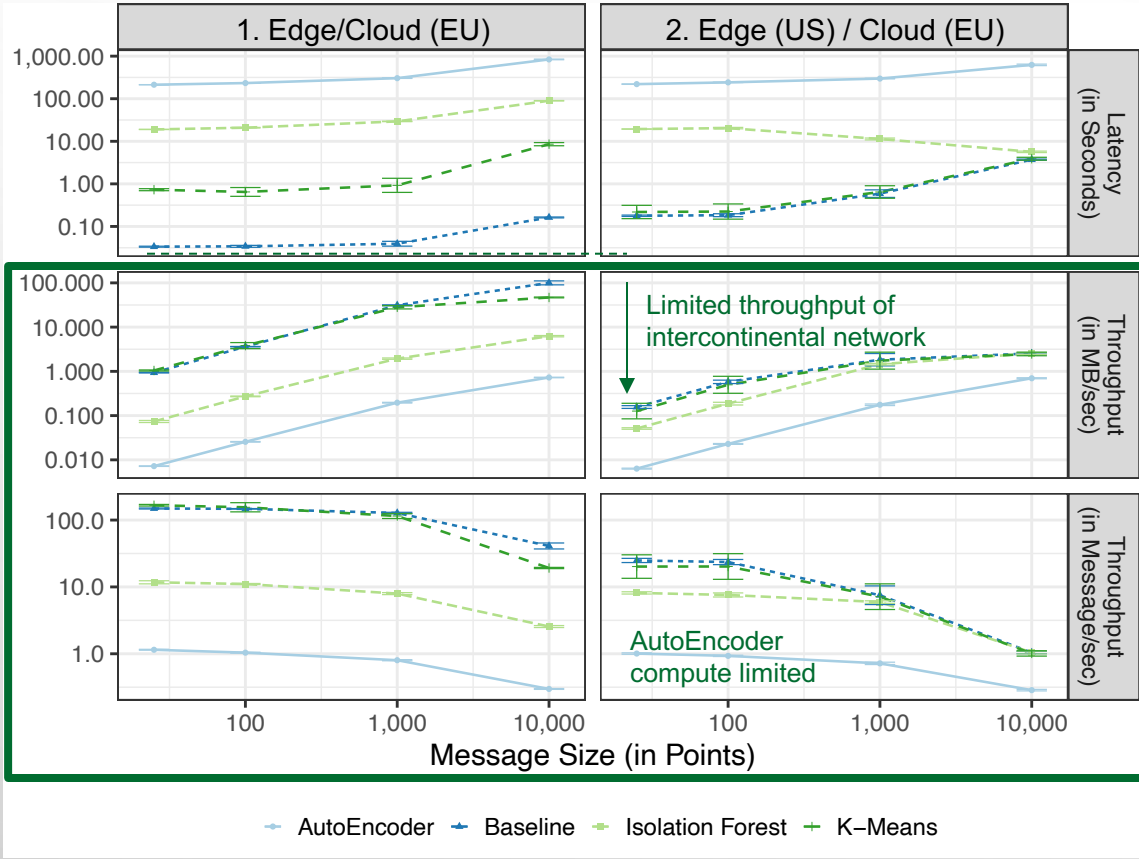
- Evaluation latency and throughput and latencies on **Leibniz Supercomputing Center (EU)** and **XSEDE Jetstream Cloud (US)** cloud
- Edge tasks are emulated using 1 core / ~2 GB (Raspberry Pi)
- Cloud processing and broker node 10 cores/44 GB memory
- Mini-App Framework for data generation*
- Three different machine learning models: K-Means, Isolation Forest and AutoEncoder
- Baseline: No compute



*https://ieeexplore.ieee.org/document/8588652

Evaluation of Pilot-Edge using Edge-to-Cloud Machine Learning Application

- Evaluation latency and throughput and latencies on **Leibniz Supercomputing Center (EU)** and **XSEDE Jetstream Cloud (US)** cloud
- Edge tasks are emulated using 1 core / ~2 GB (Raspberry Pi)
- Cloud processing and broker node 10 cores/44 GB memory
- Mini-App Framework for data generation*
- Three different machine learning models: K-Means, Isolation Forest and AutoEncoders
- Baseline: No compute



*https://ieeexplore.ieee.org/document/8588652

Evaluation: Result Summary

- Utilized Pilot-Edge abstraction and implementation for an **edge-to-cloud machine learning application** on geographically distributed edge and cloud resources
- Impact of **latencies** and **bandwidths** (EU only vs. US/EU) on the performance of the application
- Model **complexity:**
  - For K-Means throughput is limited by bandwidth
  - For AutoEncoders throughput is the limited by compute

Conclusion and Future Work

Pilot-Edge addresses the following challenges:

- **Heterogeneity:** Single abstraction and programming model from the edge to the cloud
- **Dynamic and distributed environment:** Handle distributed data flows. Ability to respond to changes in application and environment
- **Provisioning and management of resources and tasks:** common abstraction for resource management. Ability to support dynamic task placement

Future Work:

- Extension of Pilot-Edge abstraction to arbitrary infrastructure topologies
- Resource management and scheduling: Explore advanced task placement and execution strategies, e.g., energy-aware scheduling
- New workload types, e.g., federated learning