Open Programmable Networks: Microservices, Infrastructure and Coordination

Martin Swany, Intelligent Systems Engineering

INDIANA UNIVERSITY

OPeN: Open Programmable Networks (OPeN MIC)

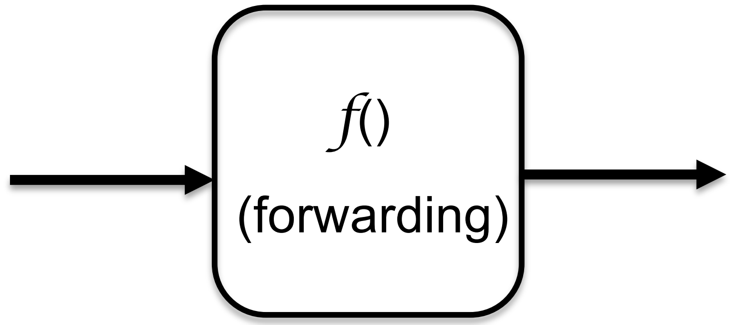
- Complete network programming and processing ecosystem developed by the OPeN Lab at IU
- **M**icroservices
  - Network processing architecture - InLocus
- **I**nfrastructure (/ **I**nstrumentation)
  - Network topology and performance monitoring – UNIS, Periscope, perfSONAR
- **C**oordination (/ **C**onfiguration)
  - Network (Data) Flow oriented programming language - Flange

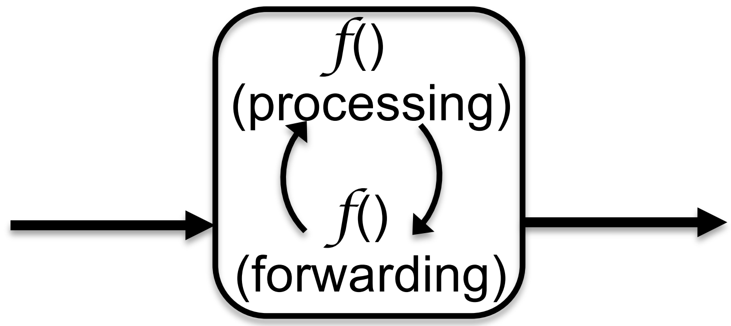
Microservices in the Network

- Network performance impacts all distributed memory computing environments, from High Performance Computing to Cloud environments
  - Bandwidth is relatively easy, but latency is more difficult; this motivates processing at the edge
- Microservices are the next step in service decomposition for scaling
  - Modern design pattern that is the logical extension of service-oriented, tiered or staged architectures
  - Function as a service, serverless computing, streaming operators (“Bolts” in Storm or Heron)
  - Over-decomposition in asynchronous many-tasking systems – Dataflow!

Network Microservices

- Network Microservices extend this processing model to the network fabric and hardware acceleration creates new opportunities
- Network function virtualization is the same design pattern
- Extend network devices to include functions other than forwarding

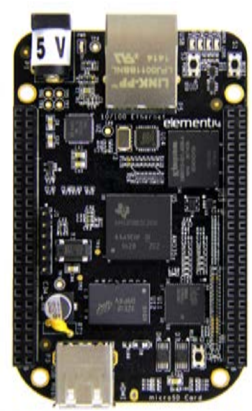




Computing for the Edge - InLocus

- Targeting microservices for network-oriented Edge computing led us to explore a simple execution model and runtime we call InLocus
- The goal is to support stream processing of sensor data in the network across a variety of simple, small platforms
- A lightweight alternative to “Docker on Linux”

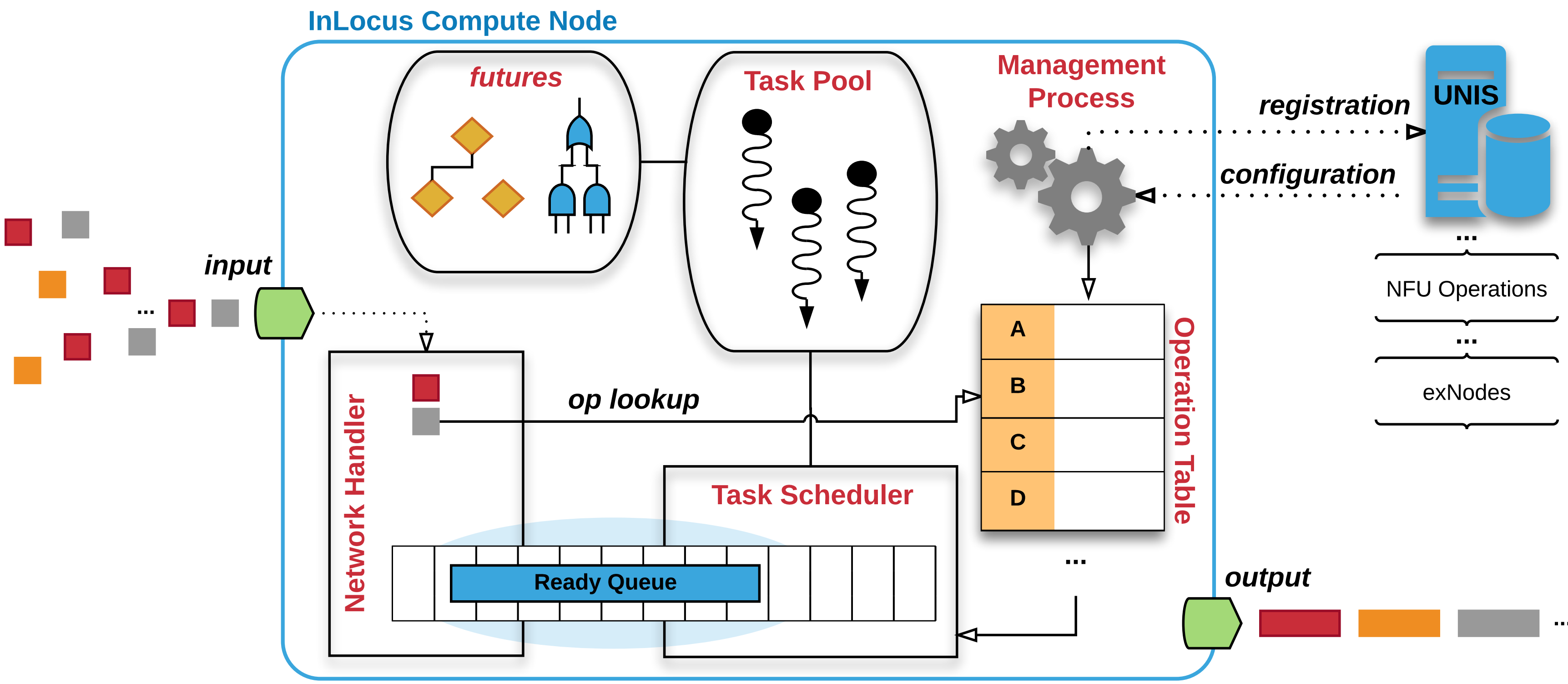
NodeMCU
ESP8266



BeagleBone Black

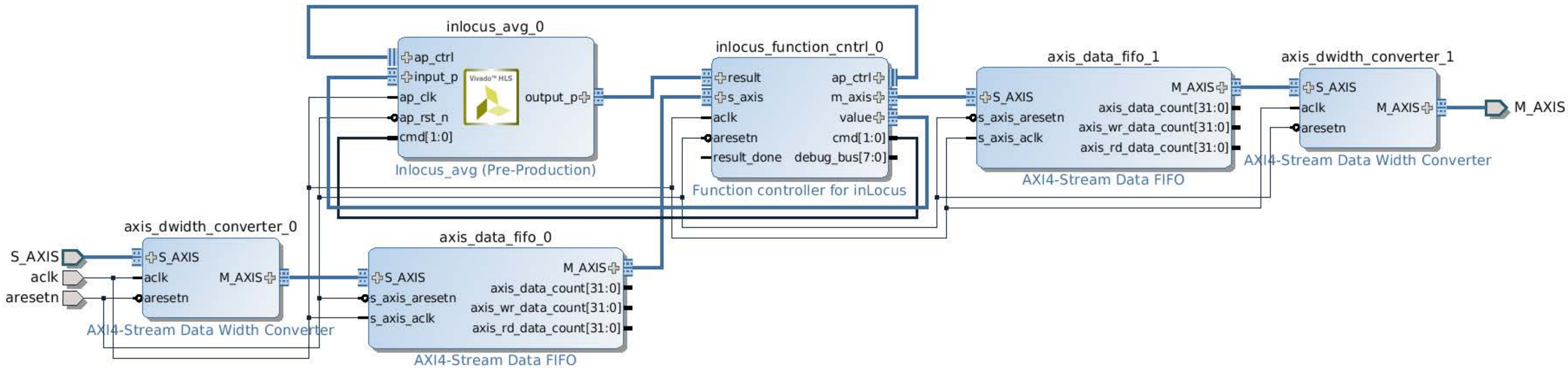
ZedBoard with Xilinx Zynq 7000

The InLocus Model



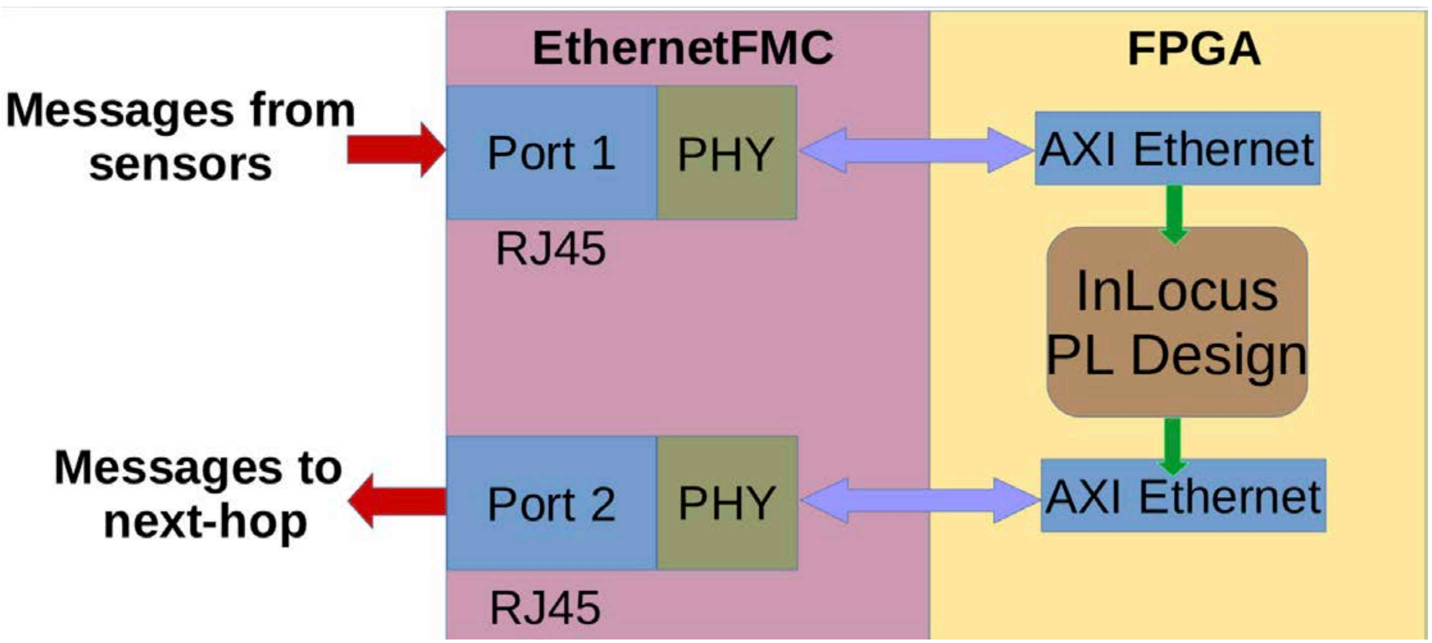
Implementation

- Streaming sensor data in timestamp, value tuples
  - CoAP/CBOR messages
- C and Node.js reference implementations
- Programmable logic infrastructure with HLS summarization function

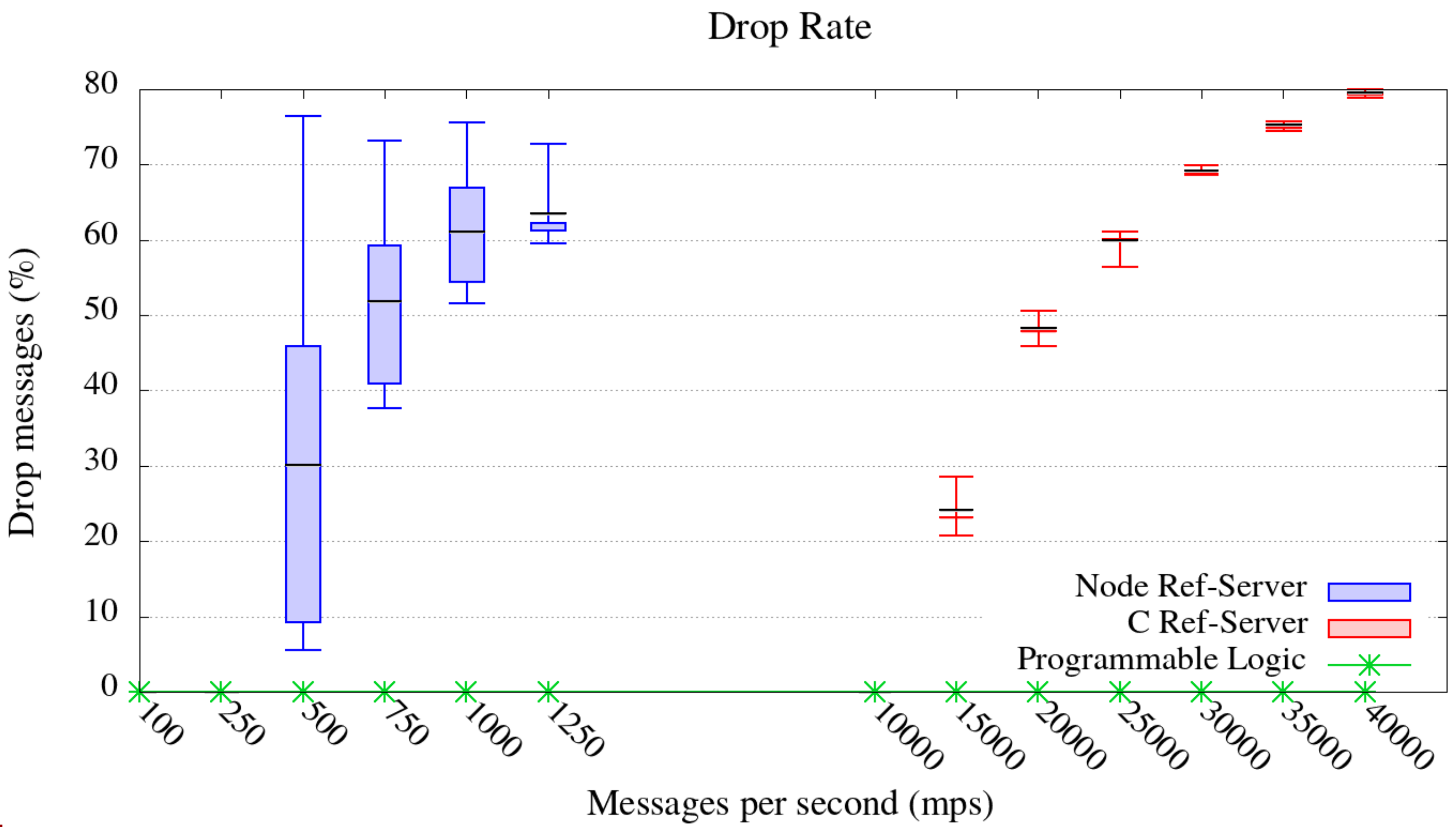


Evaluation

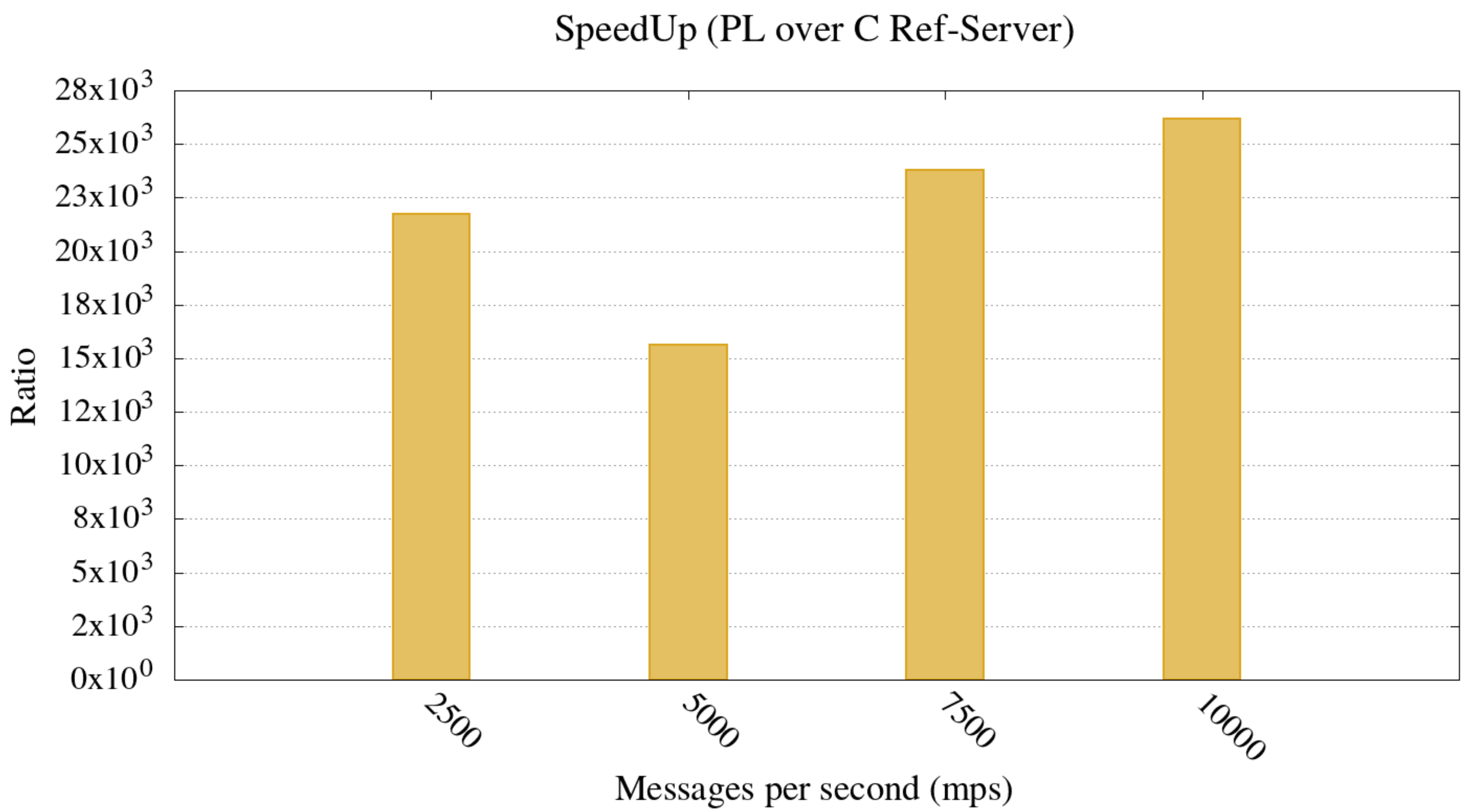
- Send streams through PL and C/Node.js designs



Results – Drop Rates



Results - Speedup



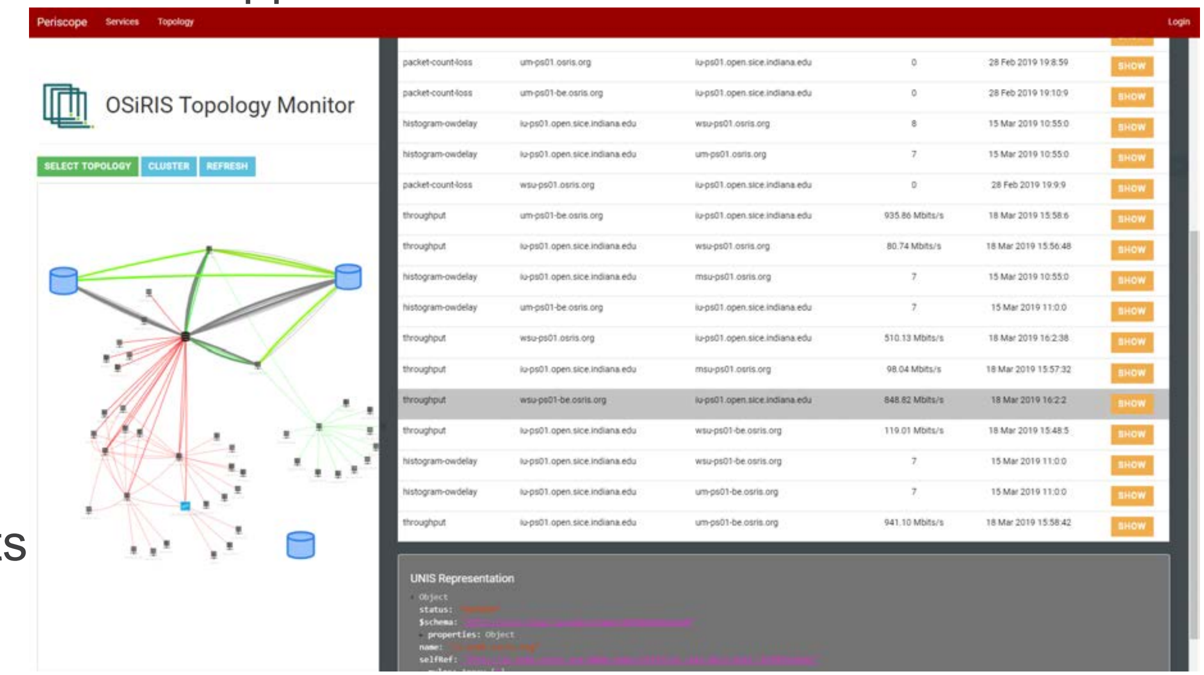
Results – Where does the time go?

Component	Mean (%)	Std Dev (%)	Time (μs)
Kernel	51.62	0.22	101.84
GlibC	18.68	0.18	36.85
libCoAP	13.43	0.15	26.50
libCBOR	13.37	0.08	26.38
Arith_avg	0.06	0.01	0.12

L. R. B. Brasilino, A. Shroyer, N. Marri, S. Agrawal, C. Pilachowski, E. Kissel, and M. Swany. Data Distillation at the Network's Edge: Exposing Programmable Logic with InLocus. In *IEEE International Conference on Edge Computing*, July 2018.

Infrastructure and Instrumentation

- The IoT needs an Internet of stuff to support it!
- Topology database (UNIS)
- Topology discovery
- Service discovery
- Performance monitoring
- User-network interface
- Network configuration agents
  - OpenFlow, OVS, etc.



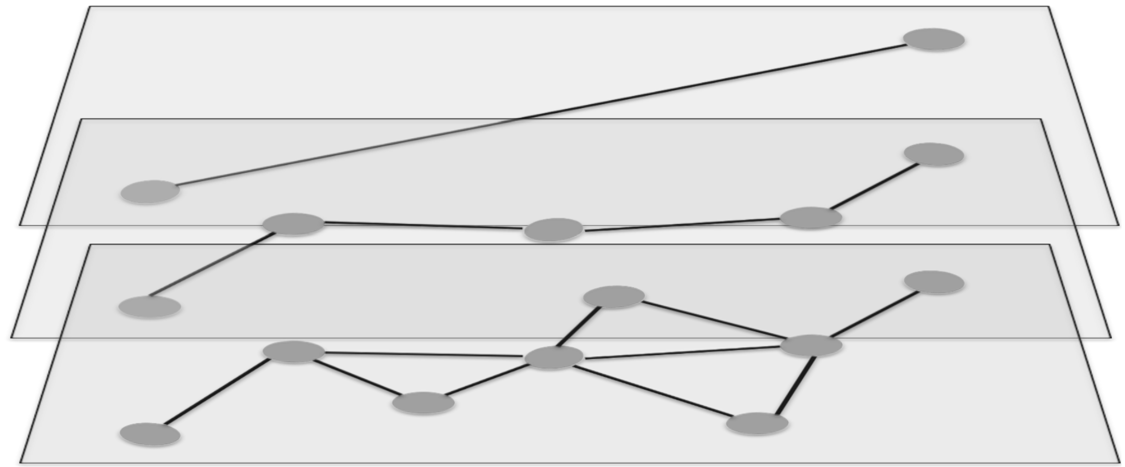
A. El-Hassany, E. Kissel, D. Gunter, and M. Swany. Design and implementation of a Unified Network Information Service. In *10th IEEE International Conference on Services Computing*, 2013.

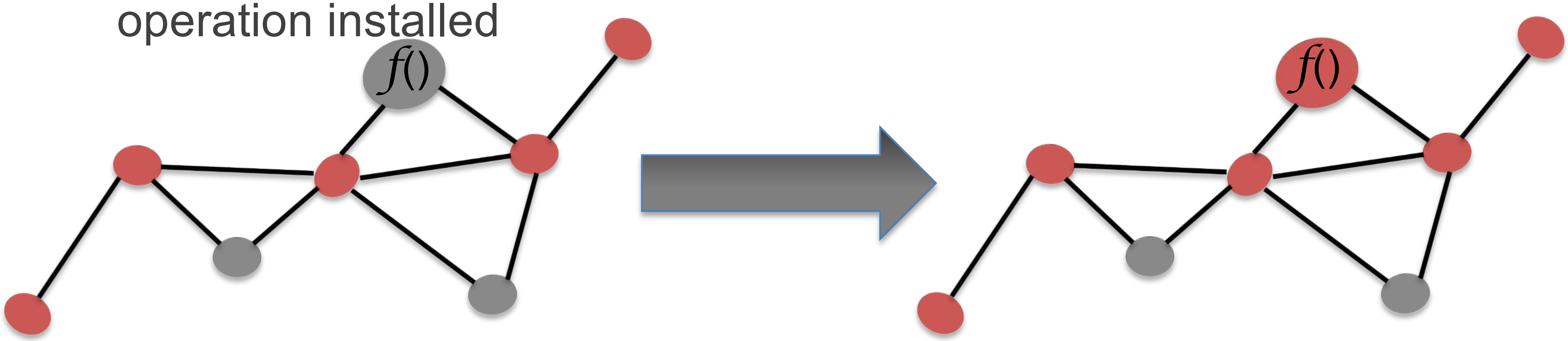
Coordination (/ Configuration*)

- We have developed a flow-oriented, domain-specific language for network programming called $\mathcal{F}$lange
- The intent of $\mathcal{F}$lange is to express the desired behavior of network flows
- Expressing constraints for a desired flow graph allows is to express
  - Traffic engineering
  - Security / firewall policies
  - Content distribution
  - Other network appliances like load balancers
  - Network / stream function placement

Flows and Functions

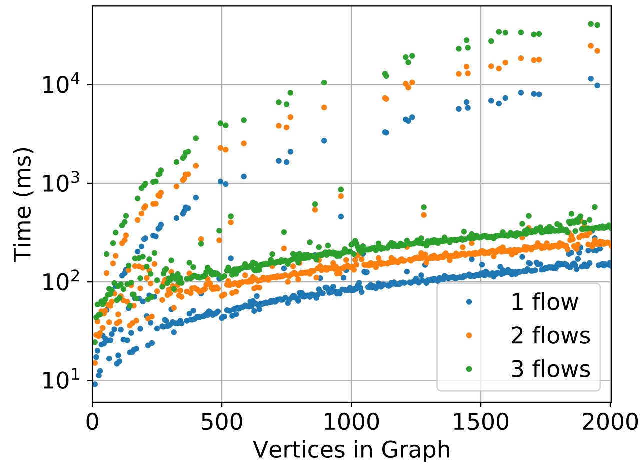
- $\mathcal{F}$lange (and UNIS) consider multi-layer graphs
- Applying operations to IoT data streams ⇒ routing through an InLocus node with the appropriate operation installed

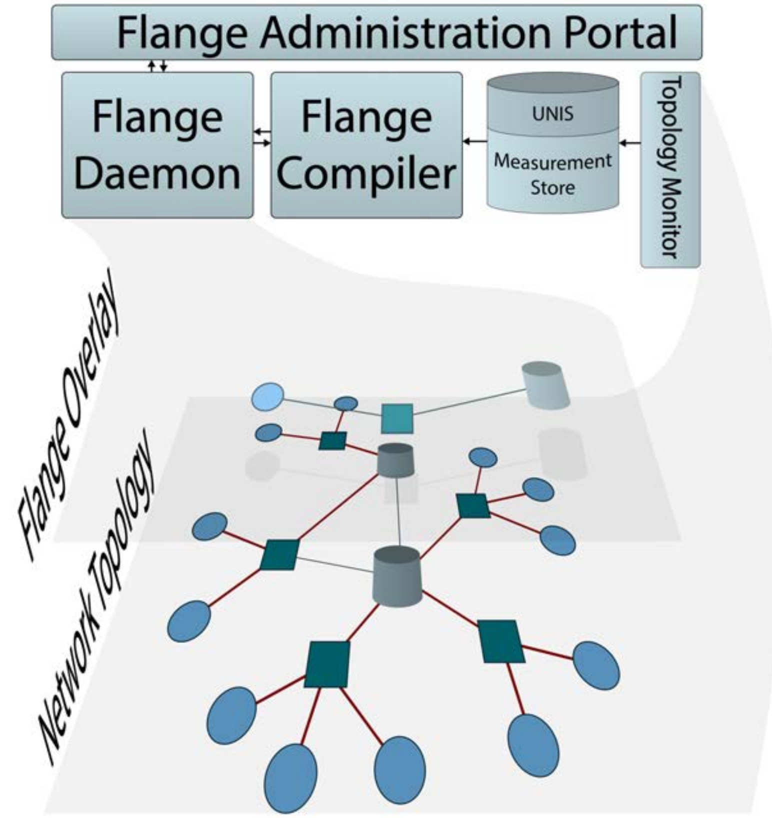




Results

- $\mathcal{F}$lange uses well-understood graph algorithms and performs as expected on common algorithms





J. Musser, E. Kissel, G. Skipper, and M. Swany. Multi-layer stream orchestration with Flange. In *IEEE International Conference on Fog Computing*, June 2019 (to appear).

OPeN Team

- Dr. Ezra Kissel
- Adam Barker
- Lucas Brasilino
- Adam Duncan
- Dominik Kovacs
- Jeremy Musser
- Daniel Osei
- Sabra Ossen
- Grant Skipper
- Juliette Zerick

Programming the network!

- The OPeN Programmable Networks ecosystem provides a complete solution for network programming for smart, adaptive networks and the Internet of Things
- We look forward to experiments using and extending this in the Pragma ENT

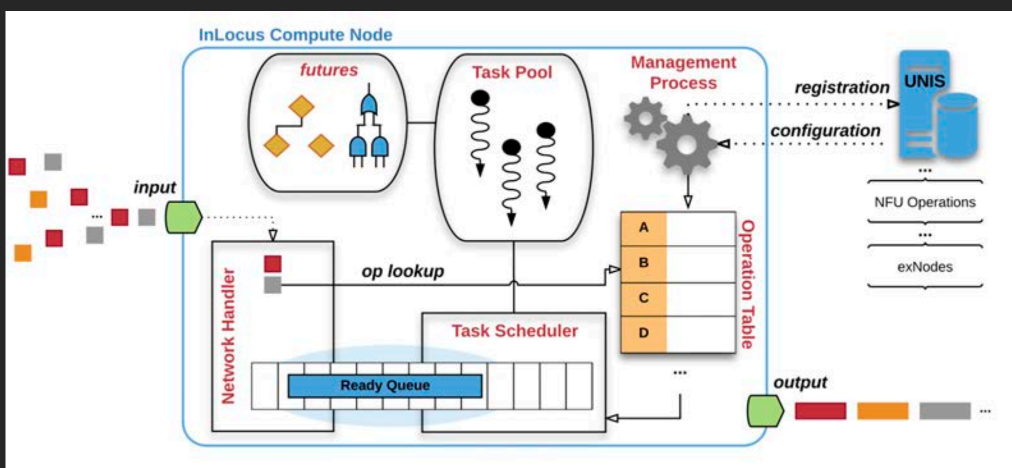
InLocus

In-Network Processing at the Edge

InLocus is an architecture that allows streaming data to be processed in the network. InLocus targets microcontrollers, microprocessors, network processors and Field Programmable Gate Arrays (FPGAs) that can be embedded in the network for highly efficient data processing at the edge.

Edge Devices

- Microcontroller
- SoC Computer
- FPGA



Motivation • Smart Cities and the Internet of Things generate massive data. Moving processing to the edge improves latency and reduces network usage.

Implementation • Network Functional Units form a distributed, heterogeneous compute fabric of MCUs, FPGAs, and resource-constrained devices, programmed and dynamically routed by a central server (Unified Network Information Service or UNIS).

Future Work • Benchmarking against VM-based applications like Heron to compare performance of C and FPGA implementation with traditional cloud architectures.