Distributed Machine Learning and the Network

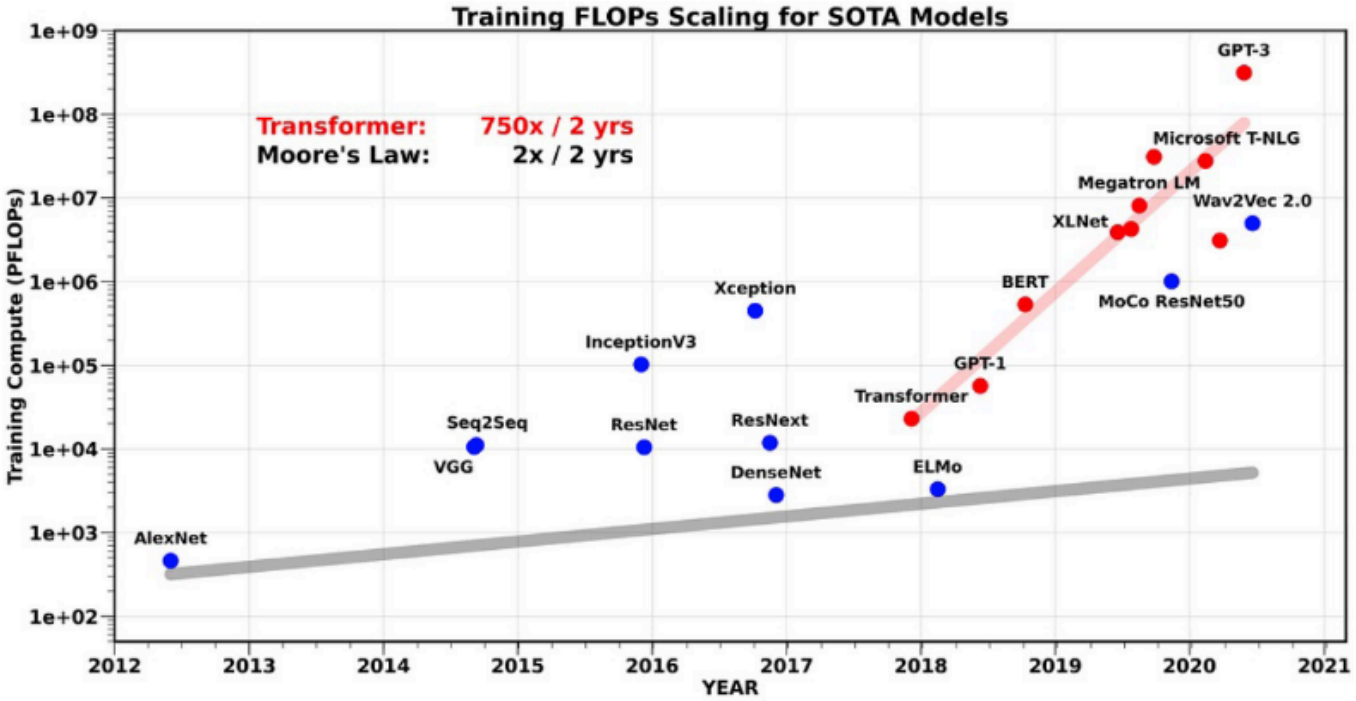
Dirk KUTSCHER

2024-03-15

Hong Kong Internet Research Workshop 2024

香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

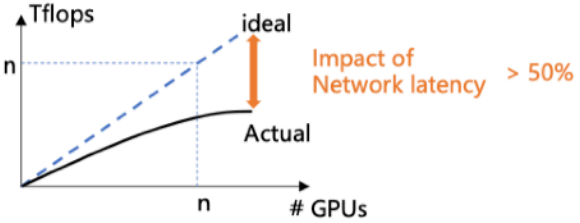
Moore's Law (Nvidia)



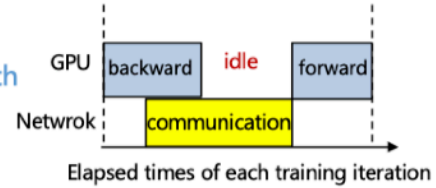
Distributed Machine Learning and the Network

Large Cluster Size ≠ Effective Flops

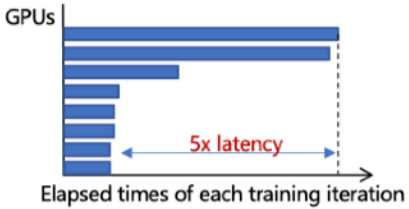
Network Performance constrains effective flops of GPU clusters



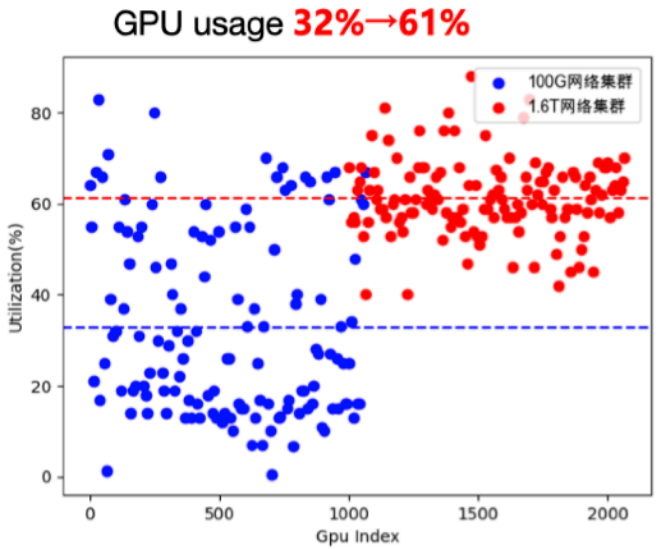
Impacts of Bandwidth (TB magnitude)



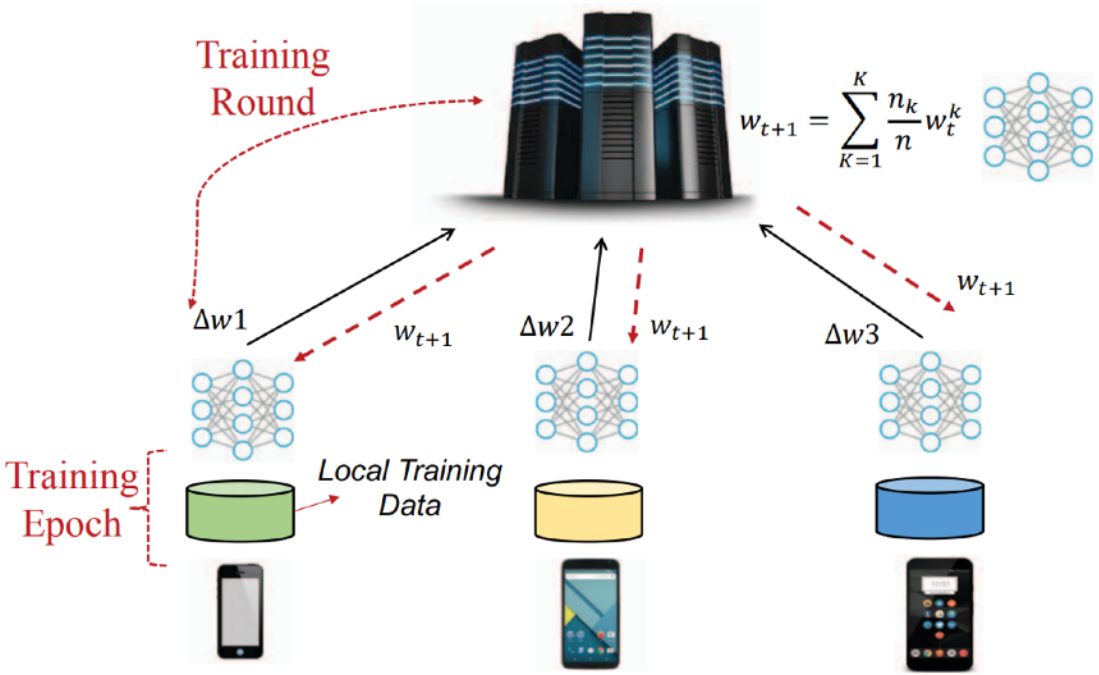
Impacts of Latency (5x)



Impacts of packet loss (0.1% packet loss incur 50%)



Traditional Centralized Federated Learning

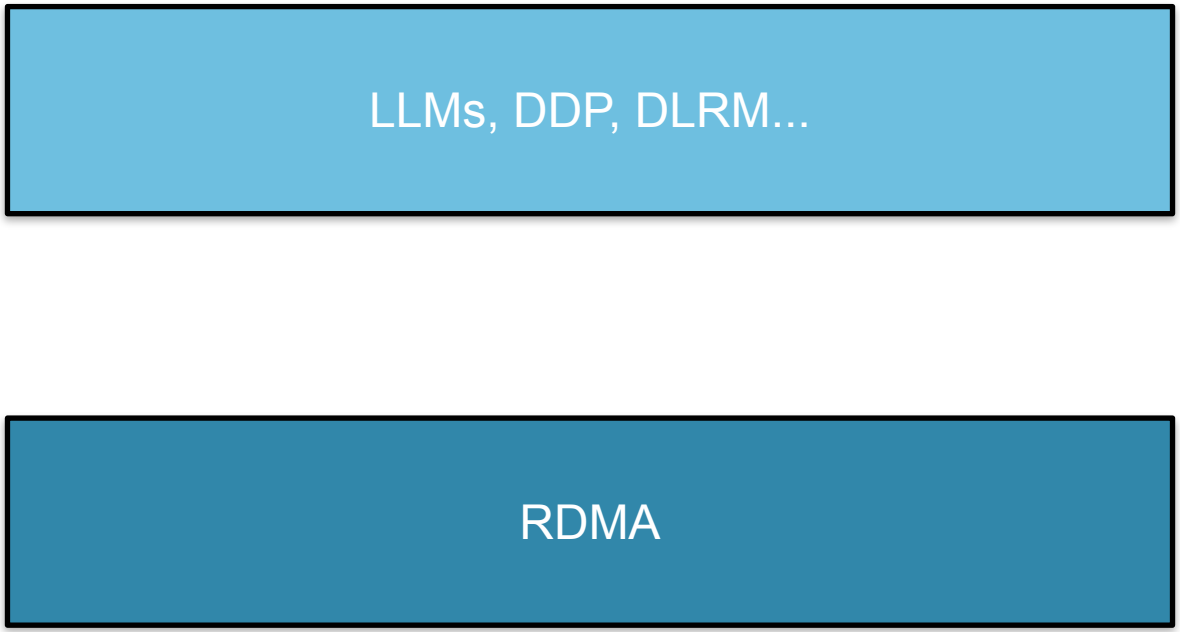


SmartPC: Hierarchical Pace Control in Real-Time Federated Learning System.
2019 IEEE Real-Time Systems Symposium (RTSS)

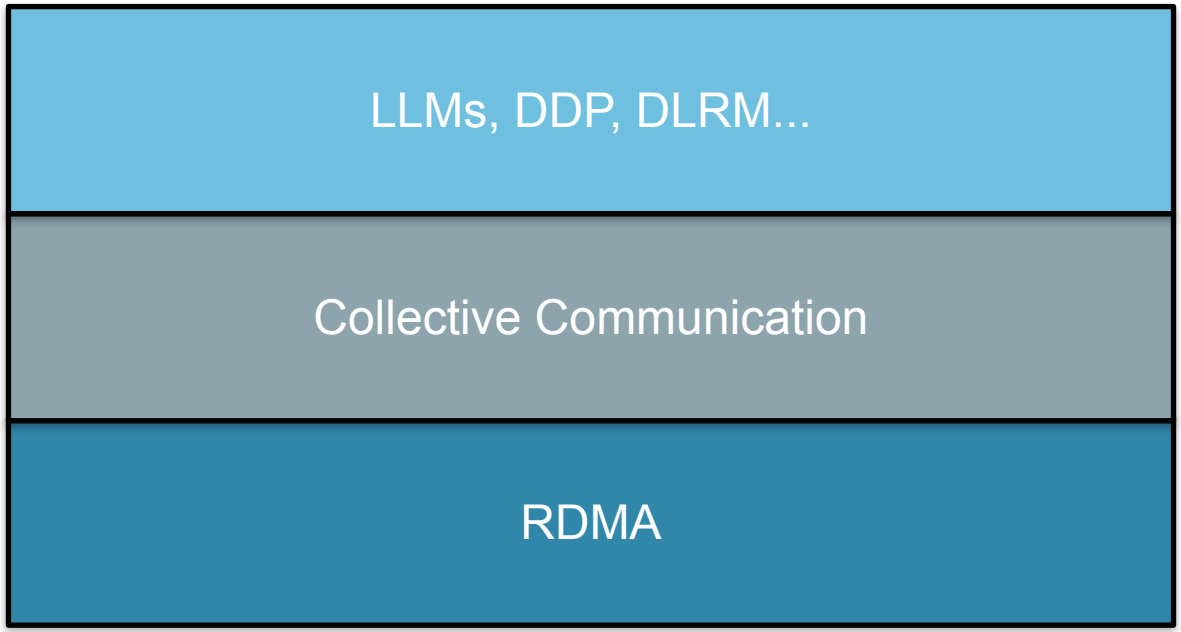
Networking for Machine Learning

LLMs, DDP, DLRM...

Networking for Machine Learning



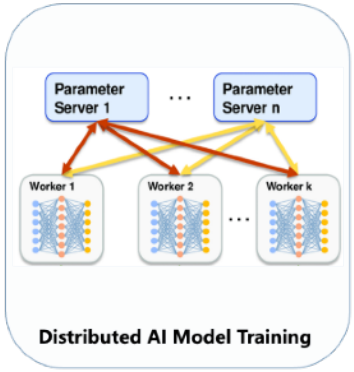
Networking for Machine Learning

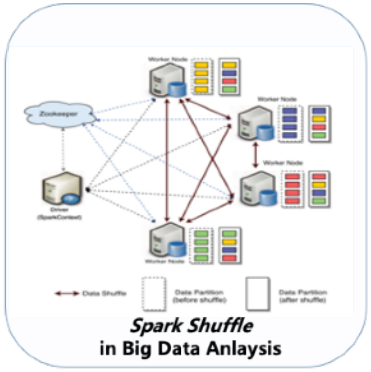


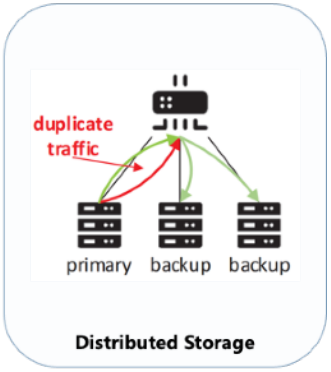
Collective Communications

- **Inter-process communication model**
  - High-performance computing and distributed machine learning model training
  - Data-oriented communication between a group of nodes
- **Processing and communication models, e.g.,**
  - **Broadcasting**, e.g., for distributing configuration data or common ML models
  - **Scattering**: single process involves a single process sending distinct pieces of data to each process
  - **Gathering**: one process collecting and combining data pieces from other processes
  - **All-to-all** communications: every process sends data to every other processes
  - **Reduction**: collect data from all processes, aggregate and send result

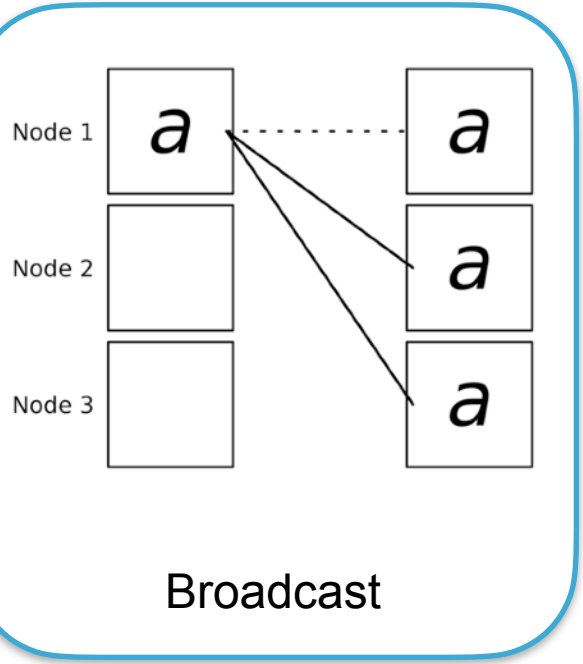
Use cases:





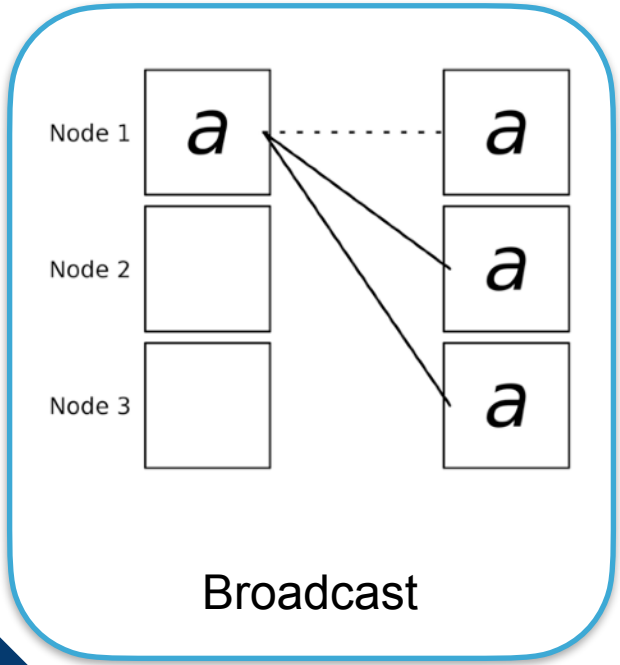


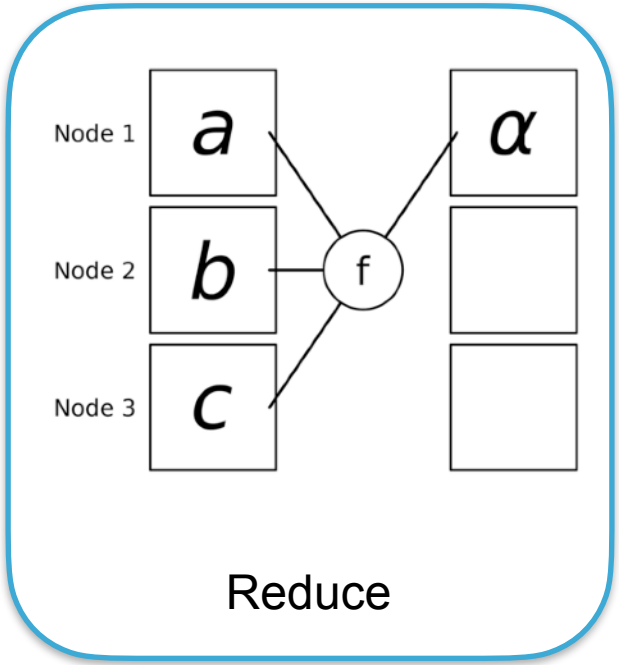
Examples



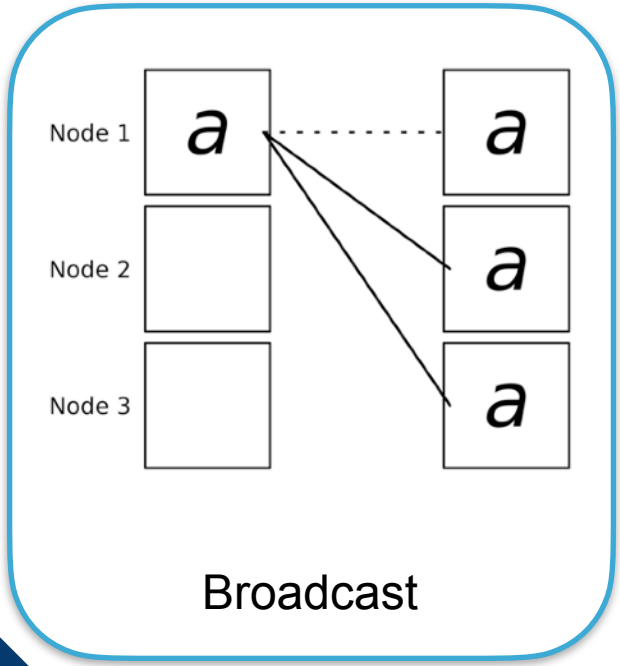
Broadcast

Examples

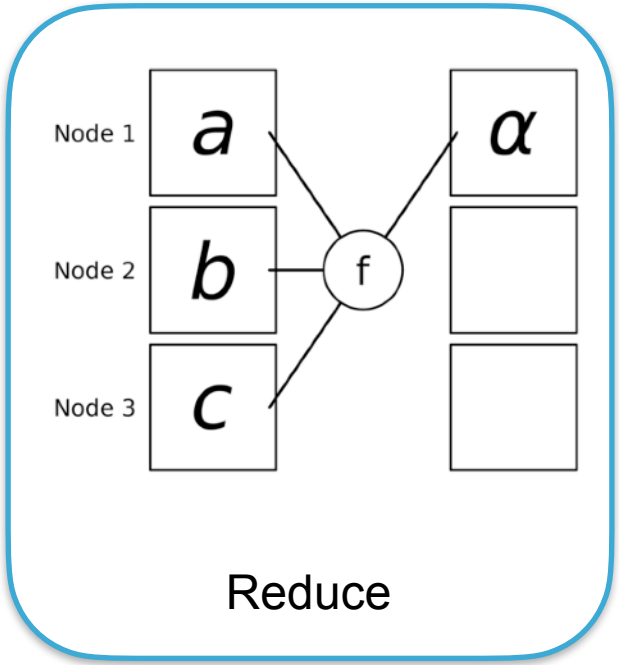




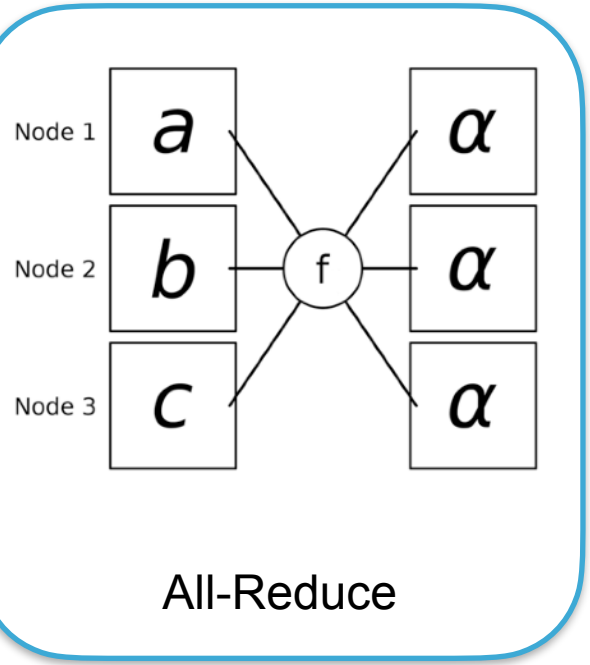
Examples



Broadcast



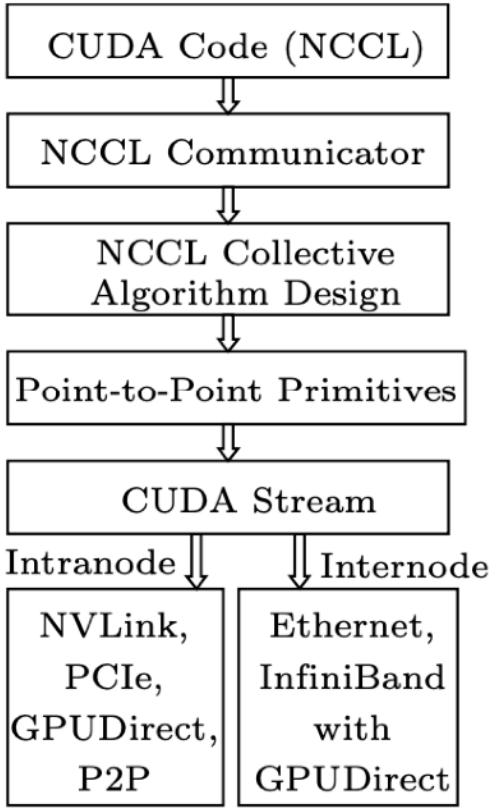
Reduce



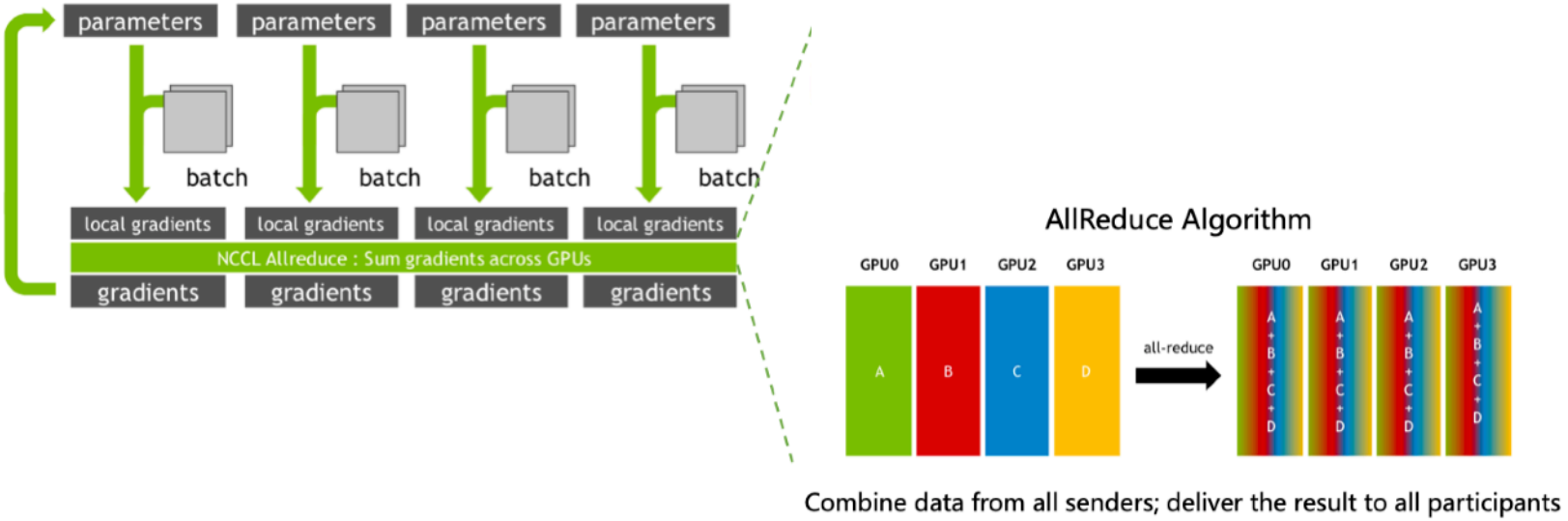
All-Reduce

Collective Communication Implementations

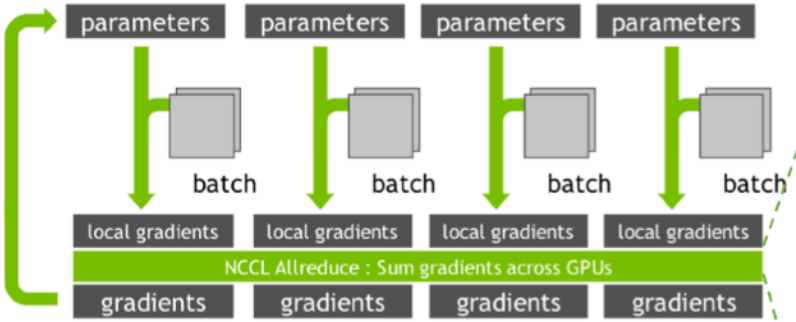
- **Example: NCCL (NVIDIA Collective Communication Library)**
  - popular GPU-accelerated collective communication library
- **Communication among GPUs using CUDA**
  - use rings to move data across all GPUs
  - chunking concept for large data objects
  - different communication backends

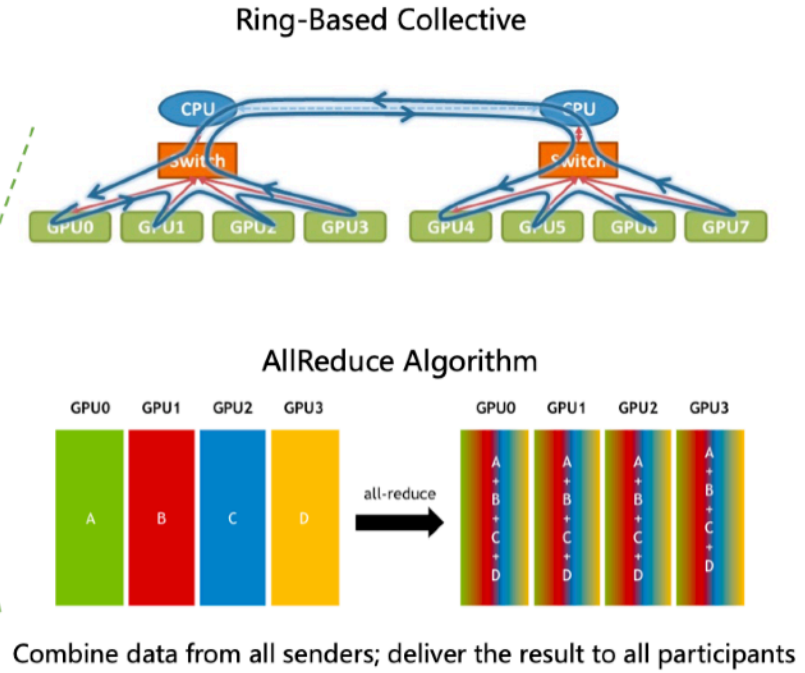


All-Reduce


parameters
parameters
parameters
parameters
batch
batch
batch
batch
local gradients
local gradients
local gradients
local gradients
NCCL Allreduce : Sum gradients across GPUs
gradients
gradients
gradients
gradients
AllReduce Algorithm
GPU0
GPU1
GPU2
GPU3
A
B
C
D
all-reduce
GPU0
GPU1
GPU2
GPU3
A+B+C+D
Combine data from all senders; deliver the result to all participants

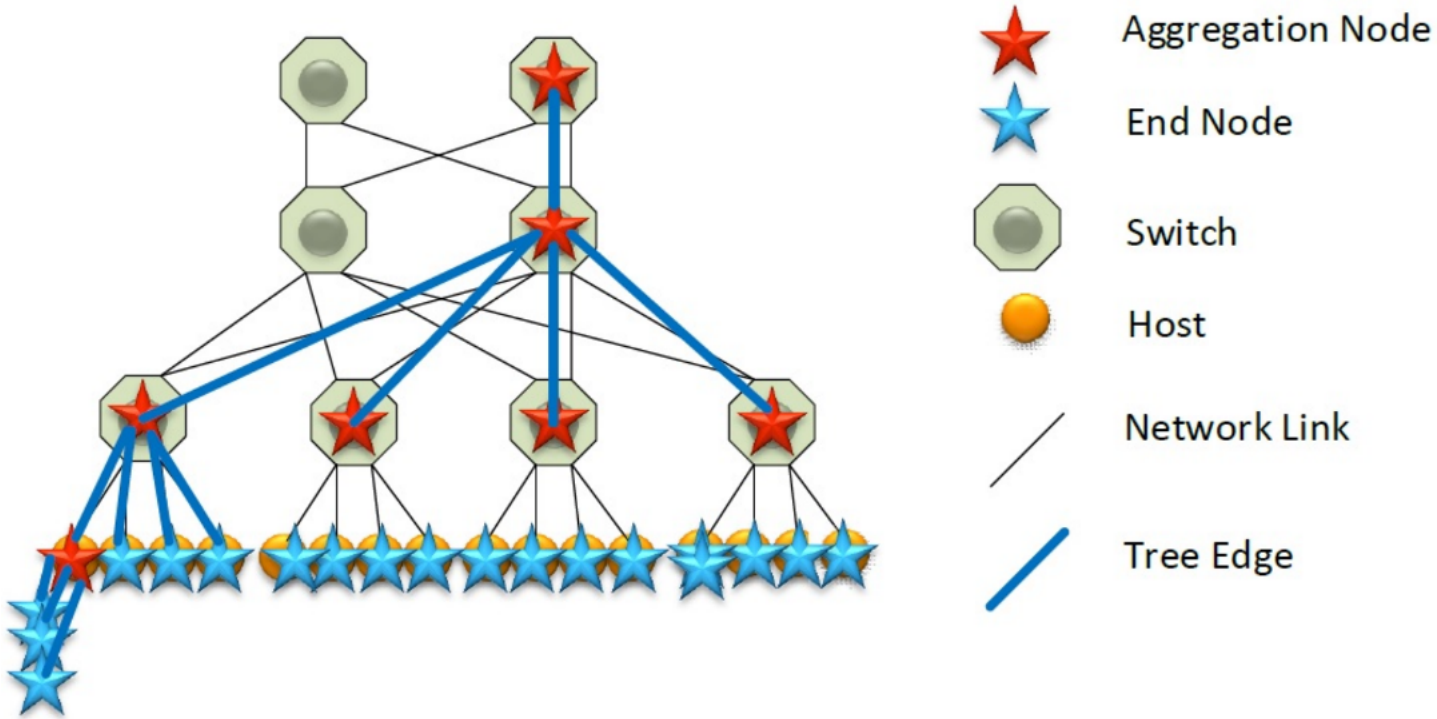
All-Reduce


parameters
parameters
parameters
parameters
batch
batch
batch
batch
local gradients
local gradients
local gradients
local gradients
NCCL Allreduce : Sum gradients across GPUs
gradients
gradients
gradients
gradients


Ring-Based Collective
CPU
CPU
GPU0
GPU3
GPU4
GPU7
AllReduce Algorithm
GPU0
GPU1
GPU2
GPU3
A
B
C
D
all-reduce
GPU0
GPU1
GPU2
GPU3
A+B+C+D
Combine data from all senders; deliver the result to all participants

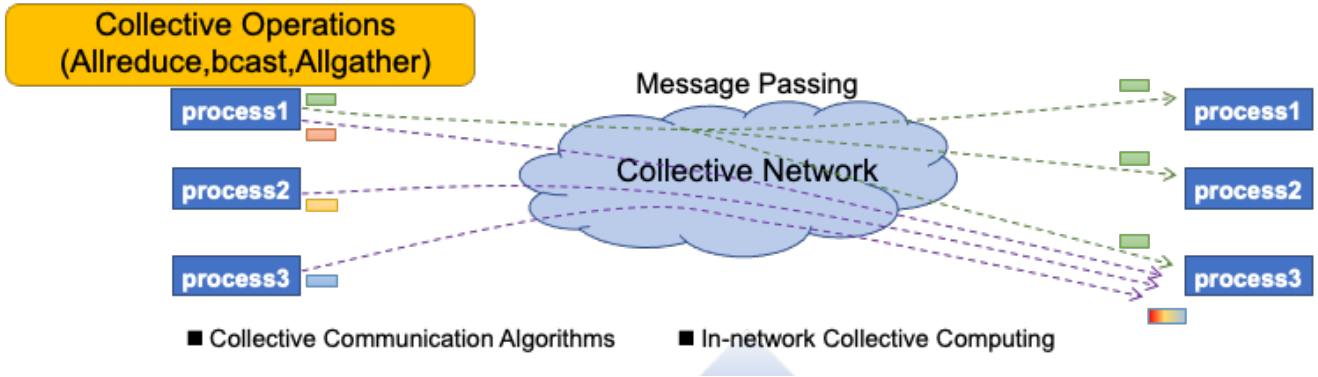
Distributed Machine Learning

Hierarchical Aggregation

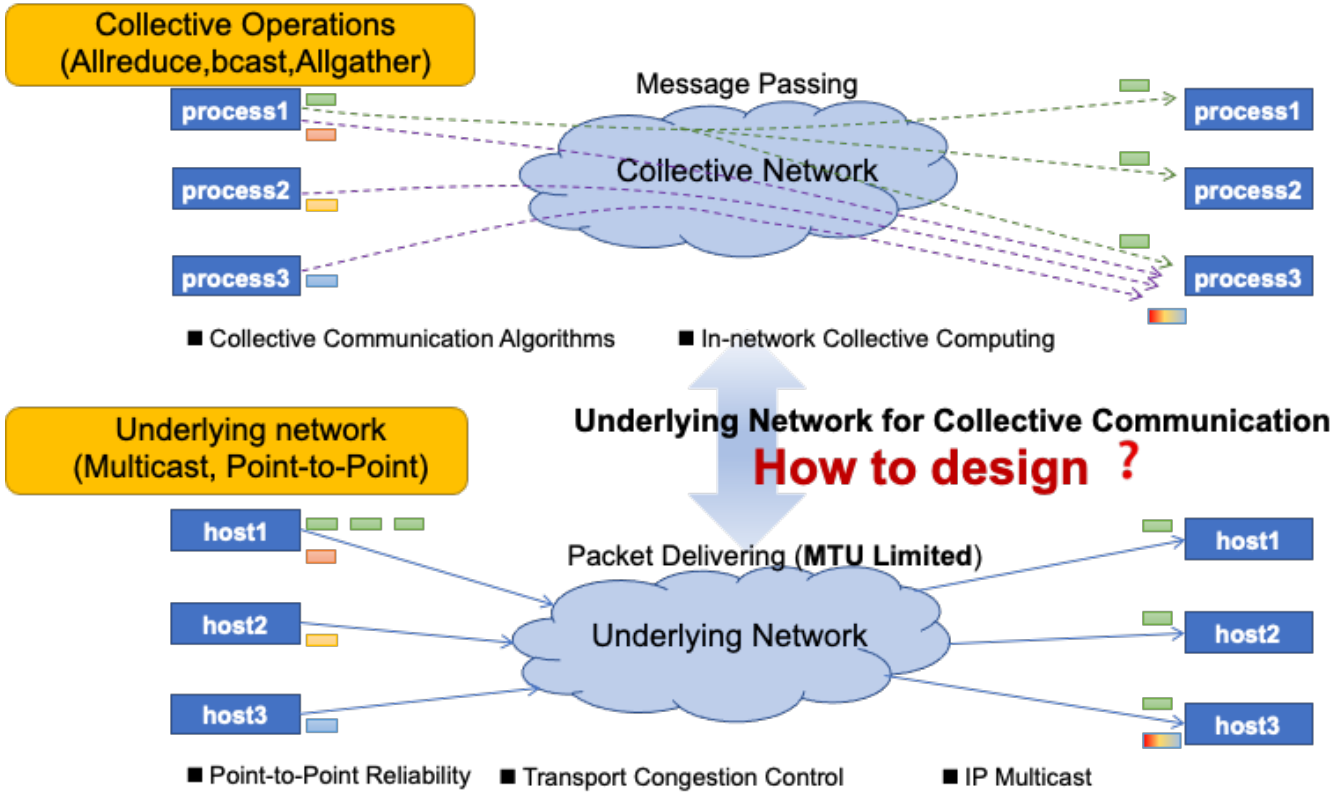


Scalable Hierarchical Aggregation Protocol (SHArP): A Hardware Architecture for Efficient Data Reduction.
2016 First Workshop on Optimization of Communication in HPC Runtime System

Mapping Collective Communications to the Network



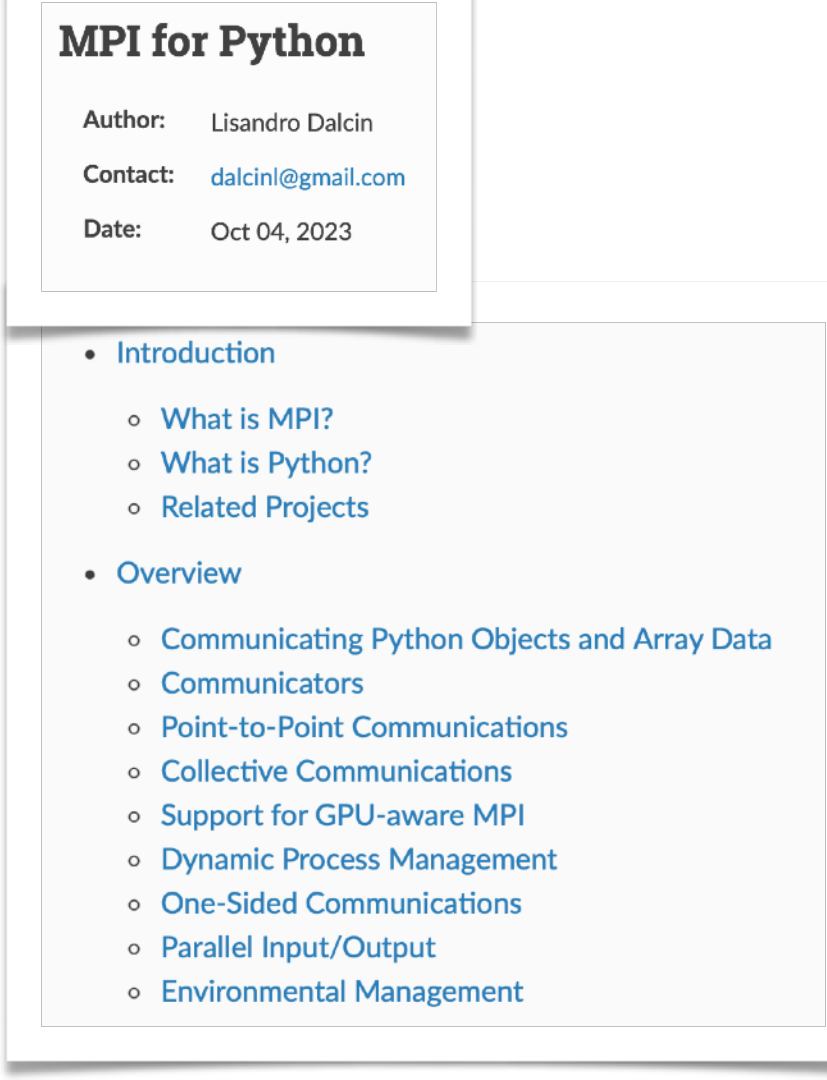
Mapping Collective Communications to the Network



Challenges

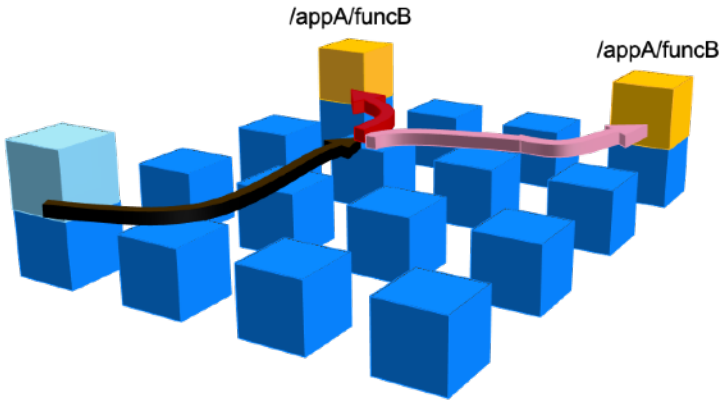
Current CC Libraries are mostly P2P-based

- Significant data duplication
- Sub-optimal forwarding
- All operations happen on endpoints
- Significant load at endpoints



Collective Communication in the Network

- **Perform generic computations**
  - reduce, gather etc.
  - on optimal nodes in the network
- **Leverage network-level multicast for distribution**
  - replication in the network
- **Enable optimization for scheduling and link utilization**



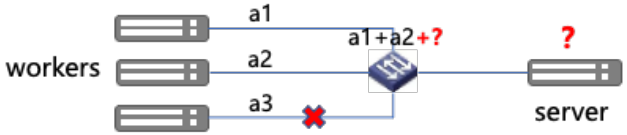
Design Challenges

- **Transport**
  - Reliability: underlying network lacks communication reliability
  - Application data units instead of packets
  - Blocking & non-blocking communication modes
  - Security?
- **Multi-destination delivery**
  - IP-Multicast possibly not the best fit
- **Computing in the Network Framework**
  - Generic operations as primitives (at least per application domain)
  - Stringent performance requirement
- **Control, Optimizations, Management**
  - Topology and utilization awareness
  - Scheduling communication and computation for optimal performance

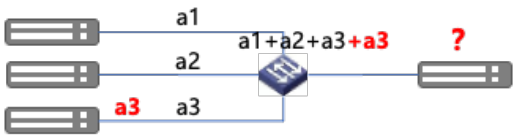
Collective Communication Transport

Reliability

- Cannot exclude packet loss
- E2E reliability conflicts with in-network computing
- Computing in the Network functions need to be promoted to communication endpoints – exceeding switch capabilities?



Sender side pkt loss

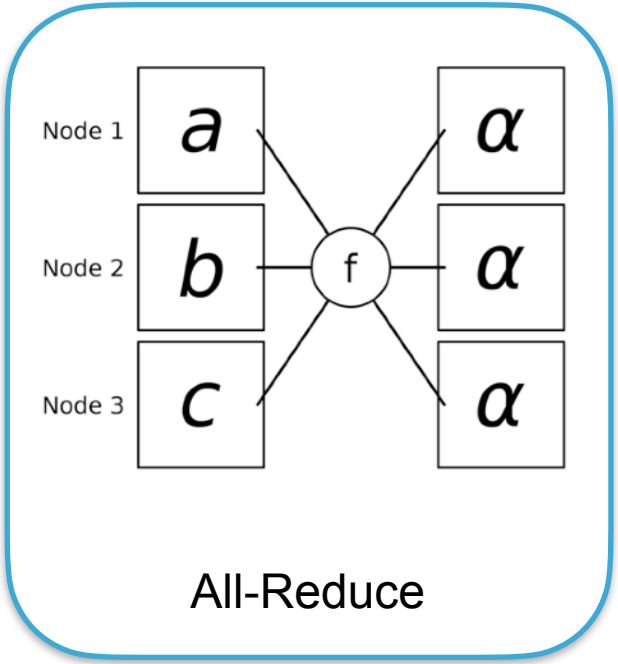


Duplicate pkts for aggregation

Collective Communication Transport

Communication Semantics

- Inter-process messaging – unbounded message size
- Packet-level abstraction does not fit
- Framing concept – or actual message-oriented communication needed



In-Network Computing

Networked Computing

- use networking to connect compute instances
- VMs, microservice instances
- interaction types such as RPC, REST
- applications such as CDN
- not really "computing in the network" – just connected computing

In-Network Computing

Networked Computing

- use networking to connect compute instances
- VMs, microservice instances
- interaction types such as RPC, REST
- applications such as CDN
- not really "computing in the network" – just connected computing

Packet Processing

- Transparent middleboxes applying processing functions on packets
- typically not very programmable

In-Network Computing

Networked Computing

- use networking to connect compute instances
- VMs, microservice instances
- interaction types such as RPC, REST
- applications such as CDN
- not really "computing in the network" – just connected computing

Active Networking

- offering abstractions for programming packet processing from an endpoint perspective

Packet Processing

- Transparent middleboxes applying processing functions on packets
- typically not very programmable

In-Network Computing

Networked Computing

- use networking to connect compute instances
- VMs, microservice instances
- interaction types such as RPC, REST
- applications such as CDN
- not really "computing in the network" – just connected computing

Packet Processing

- Transparent middleboxes applying processing functions on packets
- typically not very programmable

Active Networking

- offering abstractions for programming packet processing from an endpoint perspective

Programmable Data Plane

- abstractions of different types of network switch hardware (NPUs, CPUs, FPGA, PISA)
- programs are constrained by target platform capabilities (instruction set, memory)
- typically operate on packets/flow abstractions (e.g., *match-action* style)

In-Network Computing

Networked Computing

- use networking to connect compute instances
- VMs, microservice instances
- interaction types such as RPC, REST
- applications such as CDN
- not really "computing in the network" – just connected computing

Packet Processing

- Transparent middleboxes applying processing functions on packets
- typically not very programmable

Active Networking

- offering abstractions for programming packet processing from an endpoint perspective

Programmable Data Plane

- abstractions of different types of network switch hardware (NPUs, CPUs, FPGA, PISA)
- programs are constrained by target platform capabilities (instruction set, memory)
- typically operate on packets/flow abstractions (e.g., *match-action* style)

Network Functions Virtualization

- networked computing applied to telco functions
- some VNFs happen to process/ forward packets
- packet steering could be programmed through SDN

In-Network Computing

Networked Computing

- use networking to connect compute instances
- VMs, microservice instances
- interaction types such as RPC, REST
- applications such as CDN
- not really "computing in the network" – just connected computing

Packet Processing

- Transparent middleboxes applying processing functions on packets
- typically not very programmable

Active Networking

- offering abstractions for programming packet processing from an endpoint perspective

Programmable Data Plane

- abstractions of different types of network switch hardware (NPUs, CPUs, FPGA, PISA)
- programs are constrained by target platform capabilities (instruction set, memory)
- typically operate on packets/flow abstractions (e.g., *match-action* style)

Network Functions Virtualization

- networked computing applied to telco functions
- some VNFs happen to process/ forward packets
- packet steering could be programmed through SDN

Service Function Chaining

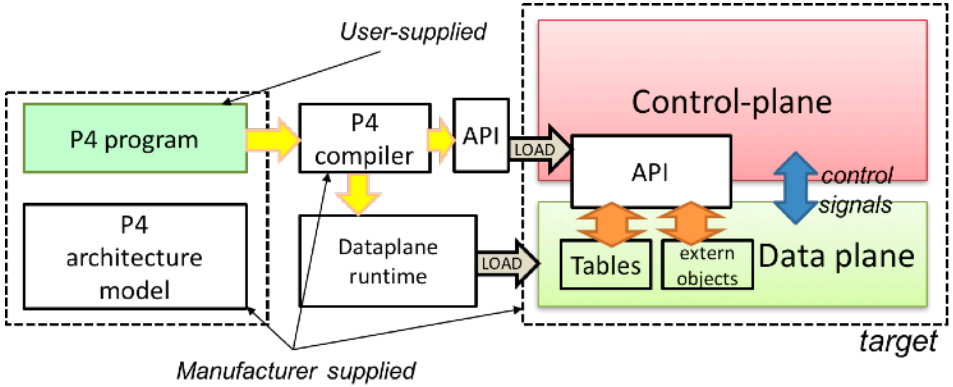
- more dynamic way for traffic steering
- dynamic chain of IP-addressable packet processors
- implemented by encapsulation

In-Network Computing

- **Sometimes, networked computing and packet processing go well together**
  - for example, when network virtualization is achieved through data-plane programming (SDN-style) to provide connectivity for VMs
  - MEC and network slicing could be an example
- **Not really computing *in* the network though**

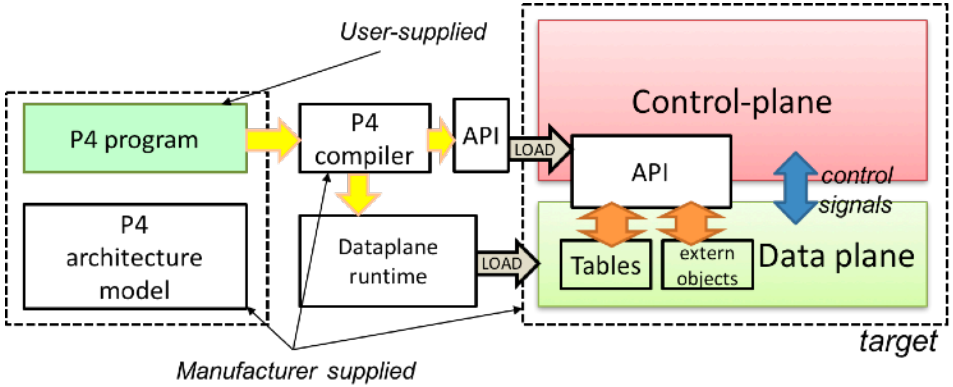
Dataplane Programmability (P4)

P4 Concept

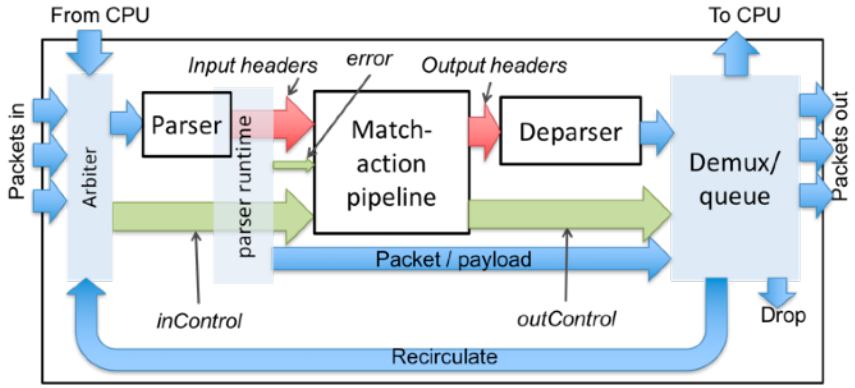


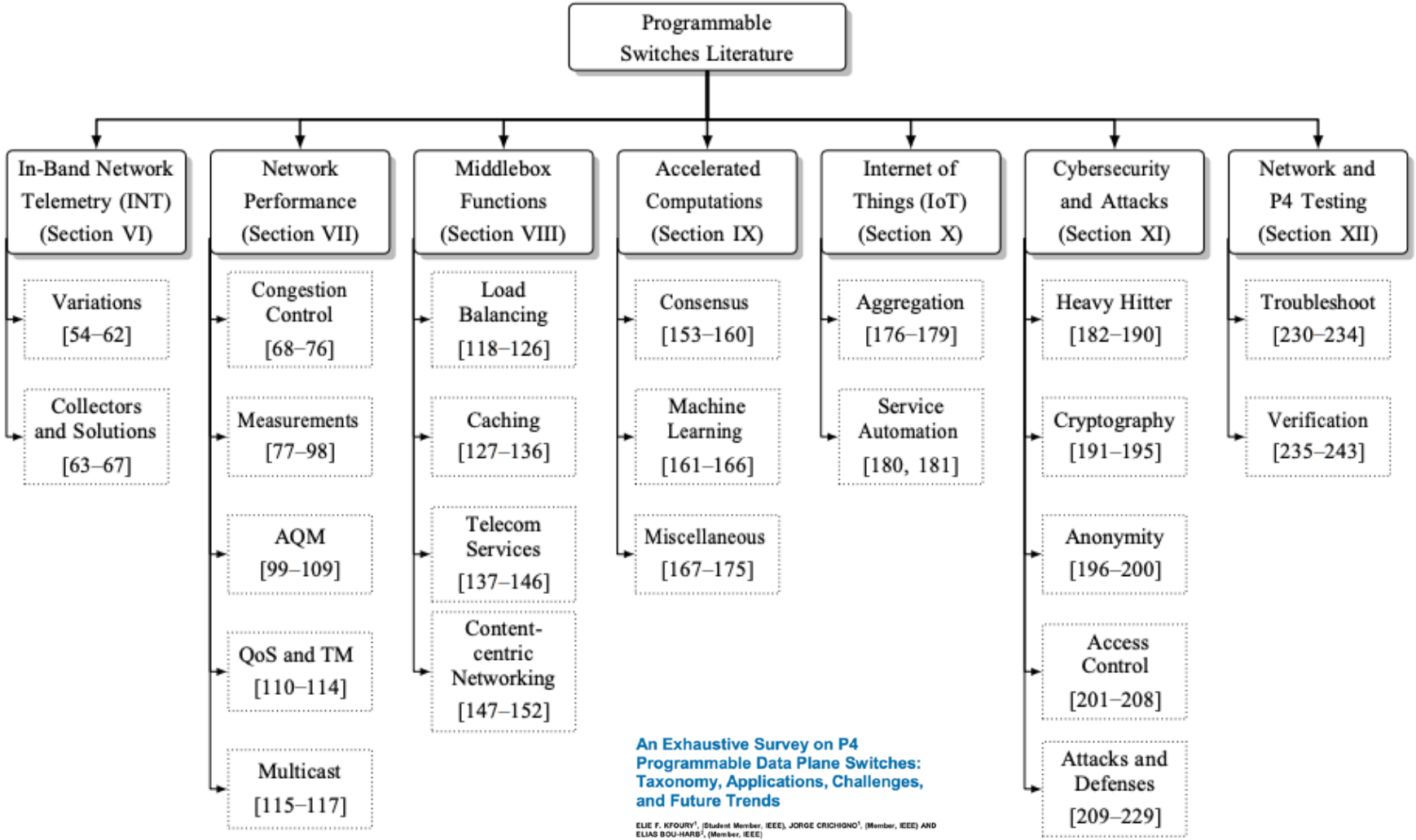
Dataplane Programmability (P4)

P4 Concept



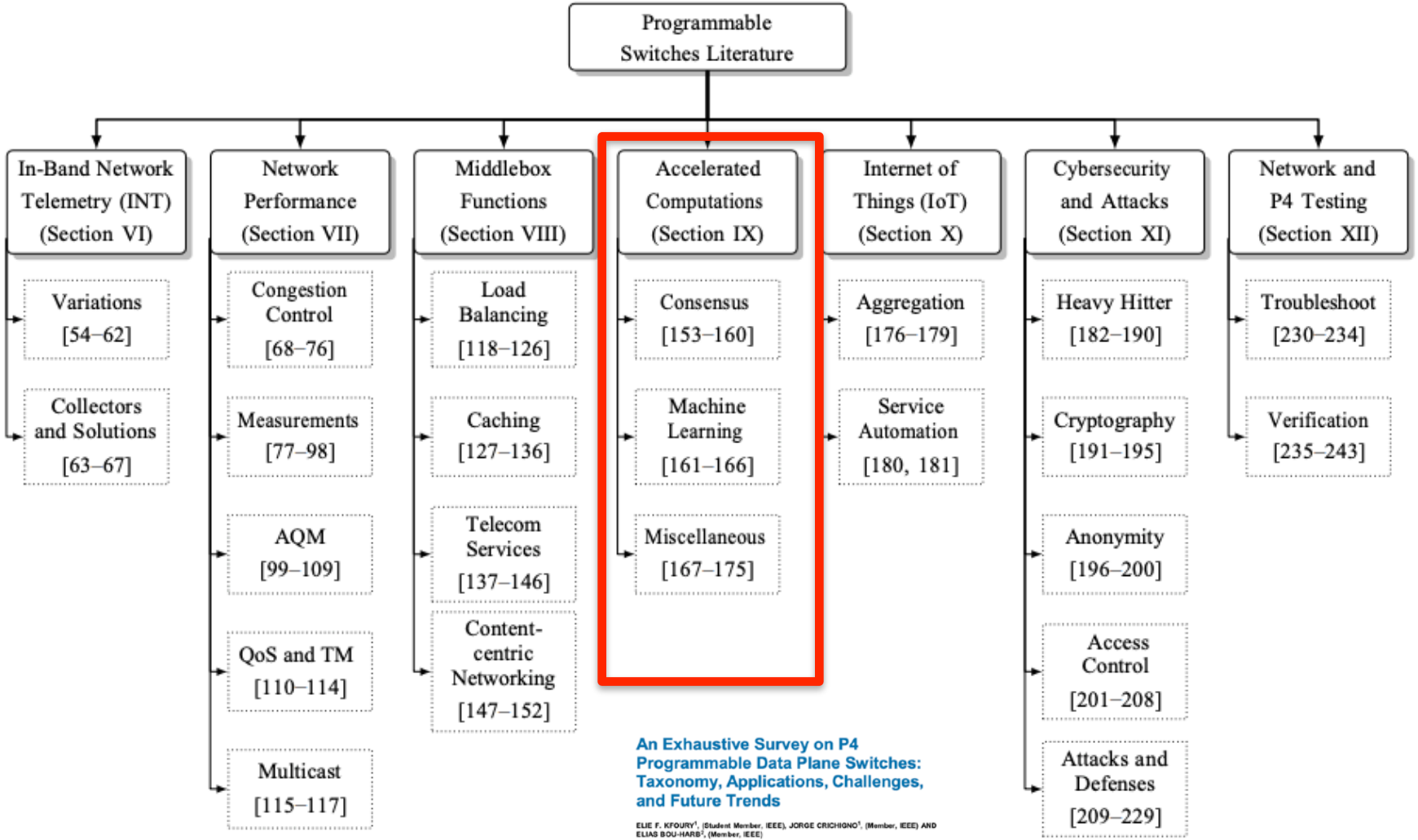
Runtime example





An Exhaustive Survey on P4 Programmable Data Plane Switches: Taxonomy, Applications, Challenges, and Future Trends

ELIE F. KFOURY[1], (Student Member, IEEE), JORGE CRICHIGNO[1], (Member, IEEE) AND ELIAS BOU-HARB[2], (Member, IEEE)


Programmable Switches Literature
In-Band Network Telemetry (INT) (Section VI)
Variations [54–62]
Collectors and Solutions [63–67]
Network Performance (Section VII)
Congestion Control [68–76]
Measurements [77–98]
AQM [99–109]
QoS and TM [110–114]
Multicast [115–117]
Middlebox Functions (Section VIII)
Load Balancing [118–126]
Caching [127–136]
Telecom Services [137–146]
Content-centric Networking [147–152]
Accelerated Computations (Section IX)
Consensus [153–160]
Machine Learning [161–166]
Miscellaneous [167–175]
Internet of Things (IoT) (Section X)
Aggregation [176–179]
Service Automation [180, 181]
Cybersecurity and Attacks (Section XI)
Heavy Hitter [182–190]
Cryptography [191–195]
Anonymity [196–200]
Access Control [201–208]
Attacks and Defenses [209–229]
Network and P4 Testing (Section XII)
Troubleshoot [230–234]
Verification [235–243]
An Exhaustive Survey on P4 Programmable Data Plane Switches: Taxonomy, Applications, Challenges, and Future Trends
ELIE F. KFOURY[1], (Student Member, IEEE), JORGE CRICHIGNO[1], (Member, IEEE) AND ELIAS BOU-HARB[2], (Member, IEEE)

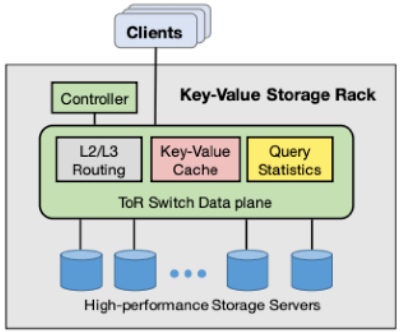
Supporting Application Logic in the Network

- **Interesting work on support for**
  - consensus protocols
  - aggregation for machine learning
  - key/value stores etc.
- **Mostly point solutions**
  - special solution for special use case
  - proof-of-concepts with limited functionality
  - dependent on specific hardware platforms
- **Typical problems**
  - limited programmability and expressiveness
  - limited storage
  - end-to-end semantics (and security)...
  - multi-tenancy

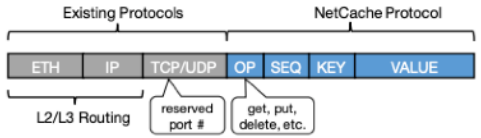
NetCache: Balancing Key-Value Stores with Fast In-Network Caching

Xin Jin[1], Xiaozhou Li[2], Haoyu Zhang[3], Robert Soulé[2,4], Jeongkeun Lee[2], Nate Foster[2,5], Changhoon Kim[2], Ion Stoica[6]

[1]Johns Hopkins University, [2]Barefoot Networks, [3]Princeton University, [4]Università della Svizzera italiana, [5]Cornell University, [6] UC Berkeley



(a) NetCache architecture.



Promising: Protocol Support

NetClone

- request cloning and result filtering
- leverage programmable switch capabilities
- without requiring application logic processing

NetClone: Fast, Scalable, and Dynamic Request Cloning for Microsecond-Scale RPCs

Gyuyeong Kim
Sungshin Women's University
South Korea
gykim@sungshin.ac.kr

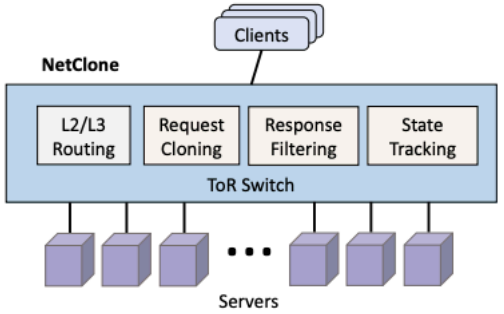
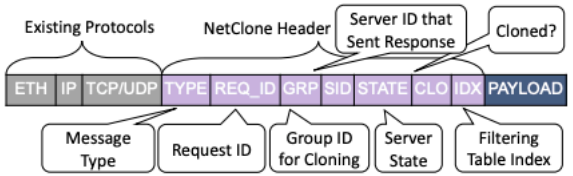
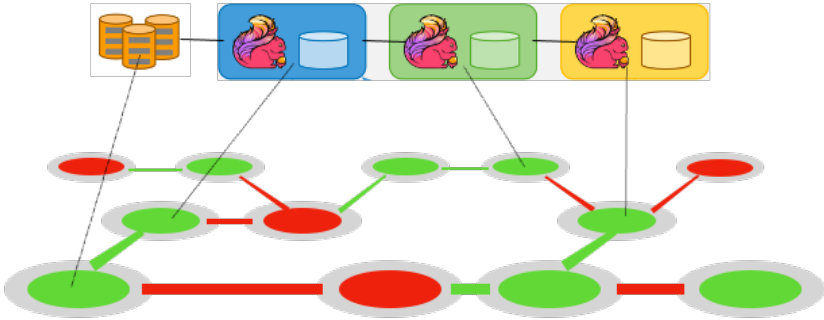


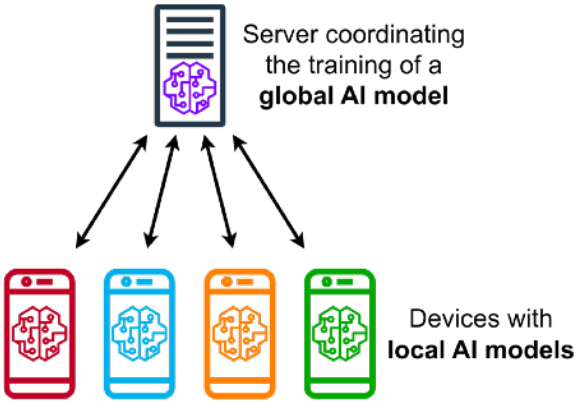
Figure 2: NetClone system architecture.



How to Support Application-Layer Distributed Computing?

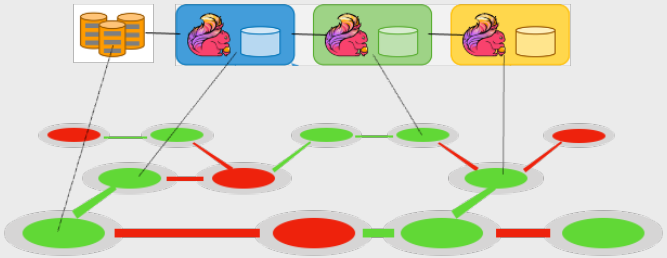
- **Distributed Stream Processing**
  - Dataflow
  - Pub-Sub
- **Distributed Machine Learning**
  - Large-scale training networks
  - Federated & de-centralized learning





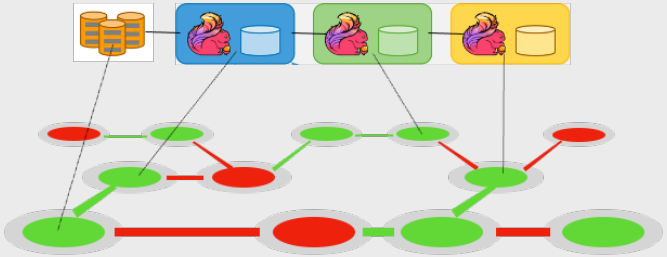
These are typically just applications – computing "on" the network...

Confluence?

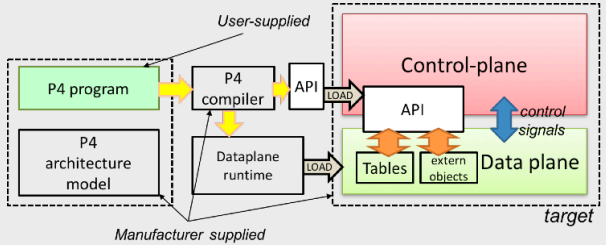


Application semantics and protocols for distributed computing

Confluence?



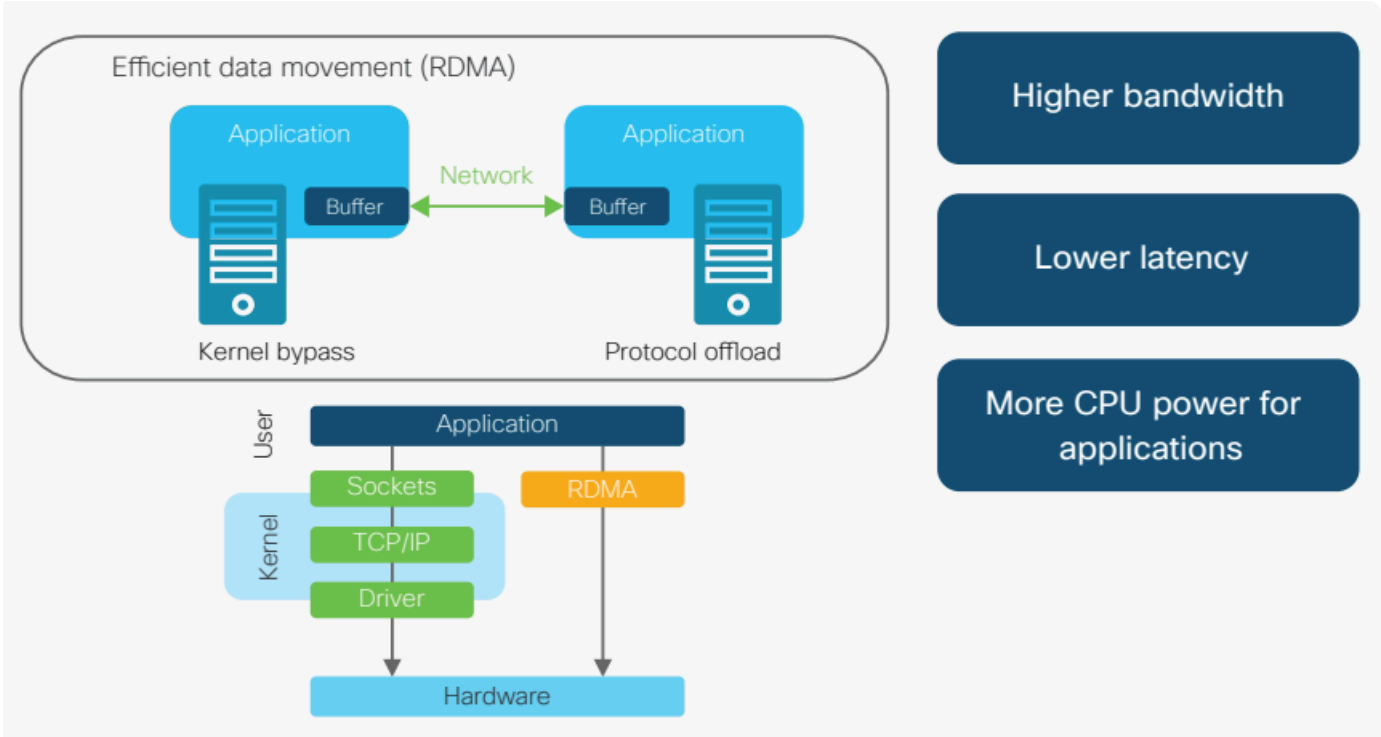
Application semantics and protocols for distributed computing



Capabilities of modern hardware (programmable switches, smartNICs etc.)

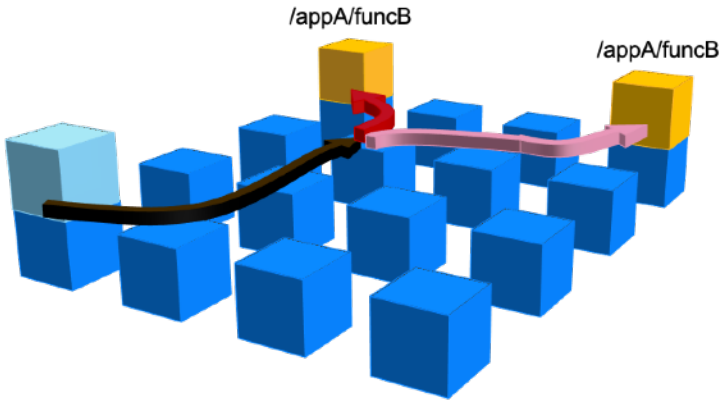
Computing in the Network

Demanding Performance Requirements



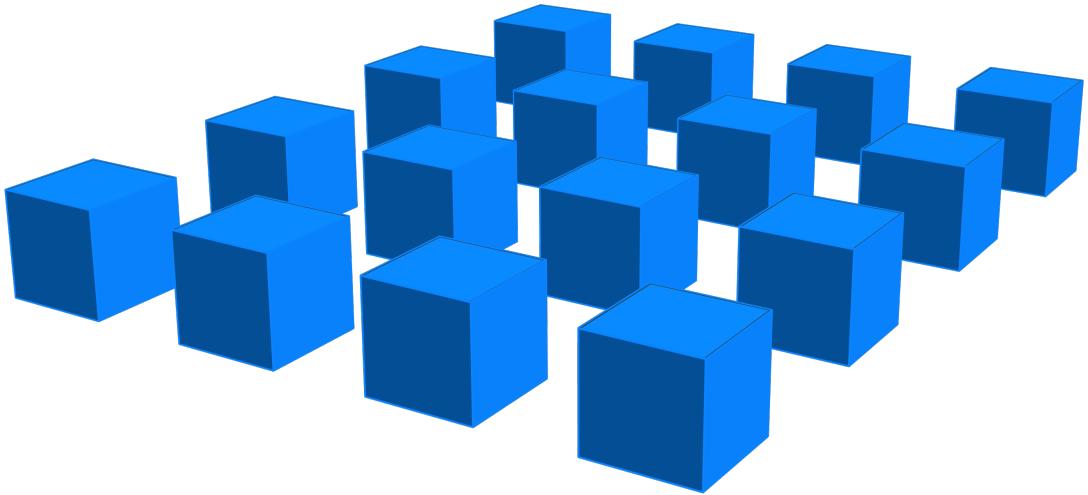
Collective Communication Needs

- **Data-oriented communication**
  - computing in the network can access and produce new application data units
- **In-network support for**
  - replication
  - retransmission
  - forwarding strategies



ICN and Distributed Computing

Network Perspective



ICN and Distributed Computing

Network Perspective

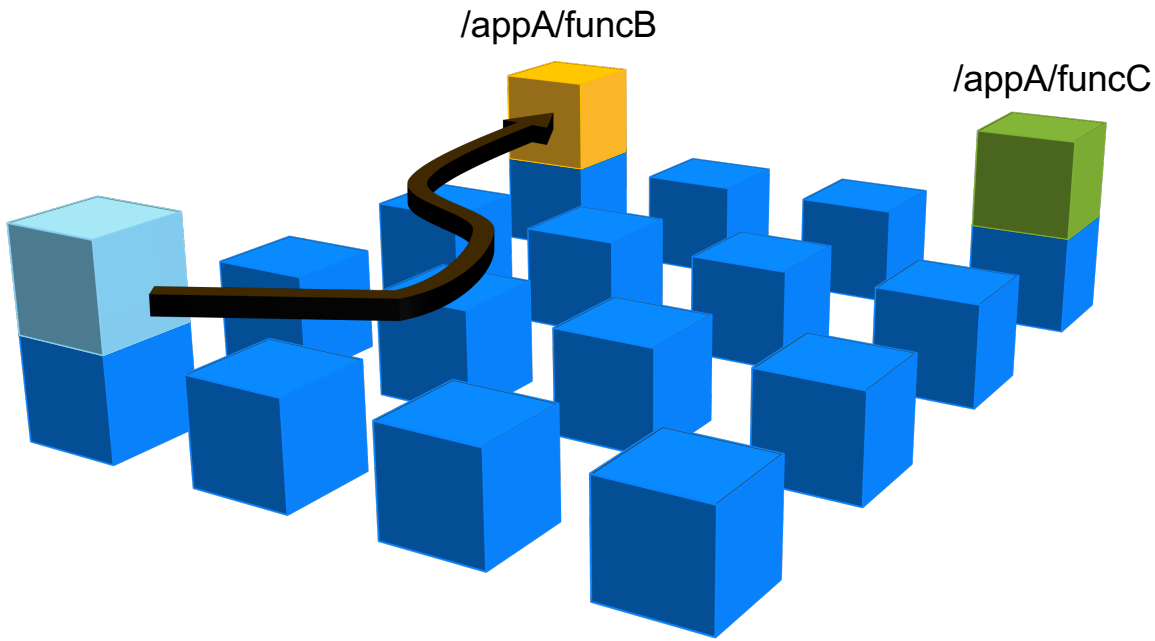
香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Request forwarding

- Self-learning
- Routing protocols
- Direct name-based forwarding

Locator-less operation

- No mapping to network topology namespace
- Support for resource mobility



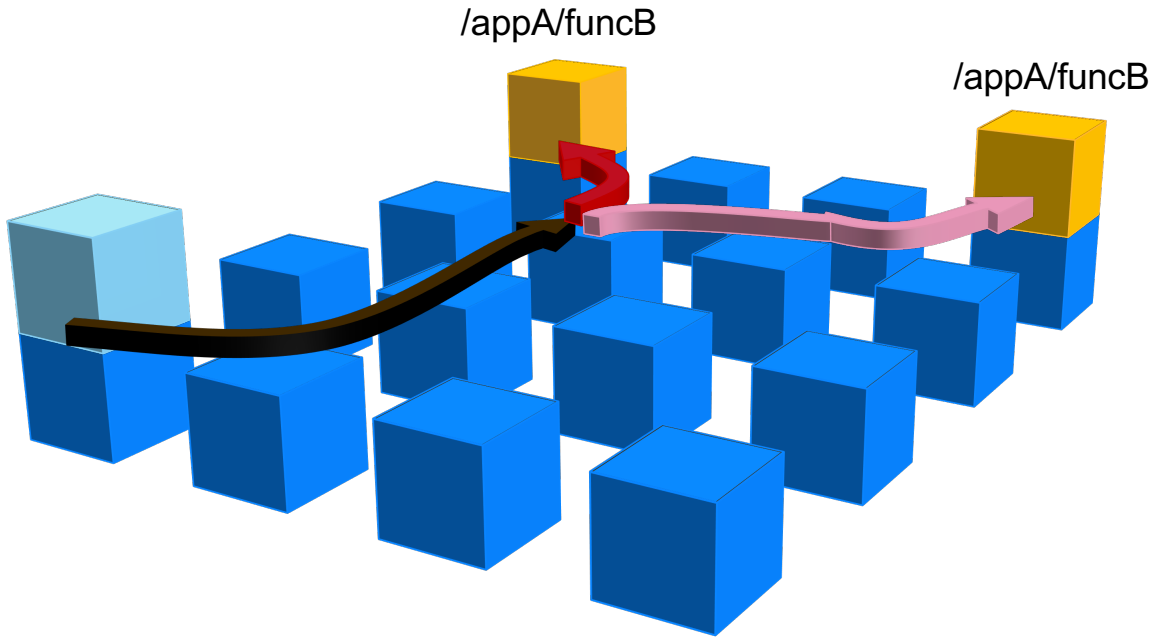
ICN and Distributed Computing

Network Perspective

香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Forwarding Strategies

- Multicasting
- Load balancing



ICN and Distributed Computing

Network Perspective

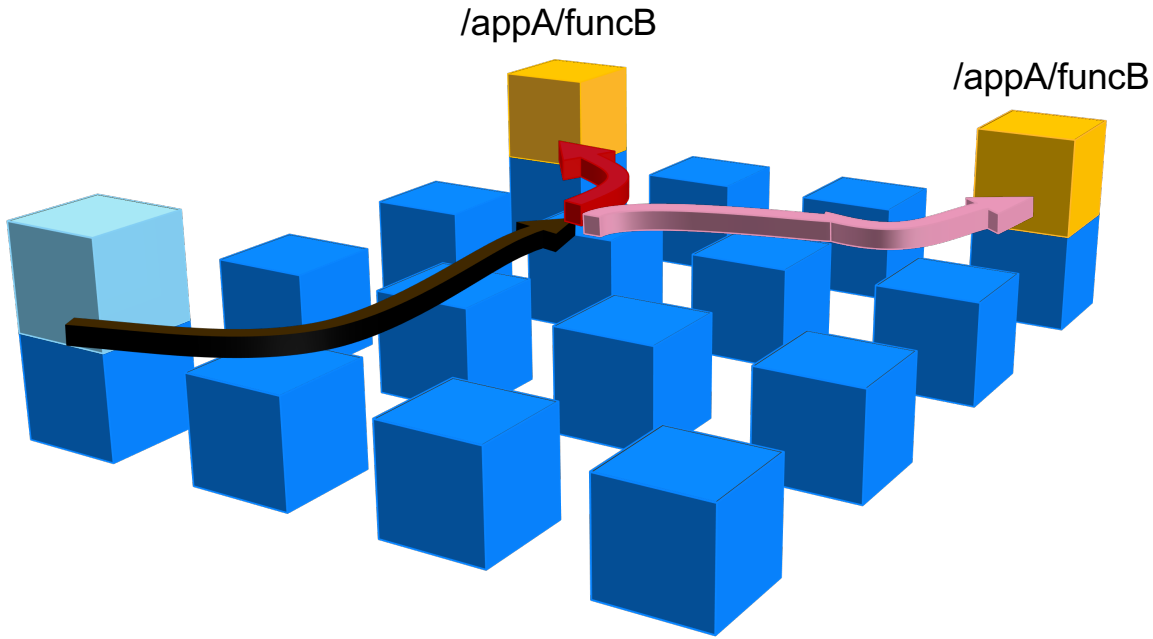
香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Forwarding Strategies

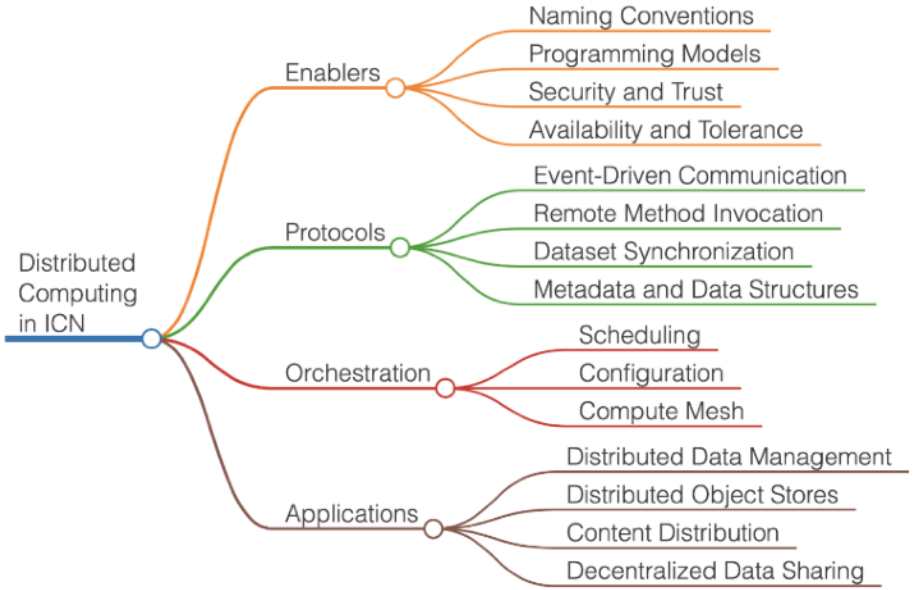
- Multicasting
- Load balancing

General ICN functions

- In-network retransmission
- Caching
- Multi-destination delivery for shared data



Distributed Computing in ICN



ACM ICN 2023

SoK: Distributed Computing in ICN

Wei Geng, HKUST(GZ)†, Guangzhou, China
Yulong Zhang, HKUST(GZ)†, Guangzhou, China
Dirk Kutscher*, HKUST(GZ)†, Guangzhou, China
Abhishek Kumar, University of Oulu, Oulu, Finland
Sasu Tarkoma, University of Helsinki and University of Oulu, Helsinki, Finland
Pan Hui, HKUST(GZ)†and University of Helsinki, Guangzhou, China

ABSTRACT

Information-Centric Networking (ICN), with its data-oriented operation and generally more powerful forwarding layer, provides an attractive platform for distributed computing. This paper provides a systematic overview and categorization of different distributed computing approaches in ICN encompassing fundamental design principles, frameworks and orchestration, protocols, enablers, and applications. We discuss current pain points in legacy distributed computing, attractive ICN features, and how different systems use them. This paper also provides a discussion of potential future work for distributed computing in ICN.

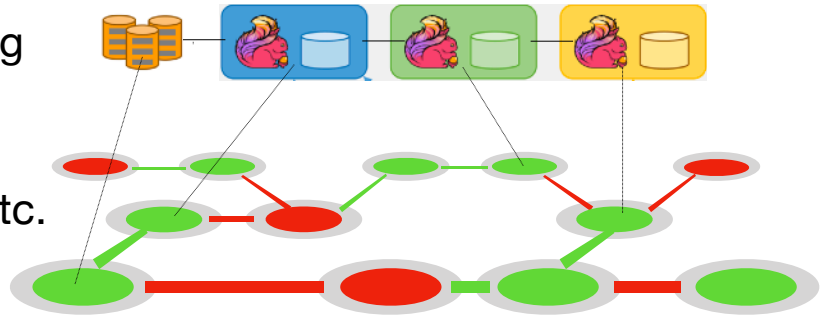
frameworks such as TLS and the web PKI [55] constrain the use of public-key cryptography for different security functions.

With currently available Internet technologies, we can observe a relatively succinct layering of networking and distributed computing, i.e., distributed computing is typically implemented in overlays with Content Distribution Networks (CDNs) being prominent and ubiquitous example. Recently, there has been growing interest in revisiting this relationship, for example by the IRTF Computing in the Network Research Group (COINRG)[1] – motivated by advances in network and server platforms, e.g., through the development of programmable data plane platforms [66] and the development of different types of distributed computing frameworks, e.g., stream

Compute-First Networking (CFN)

Goals

- **Ubiquitous distributed computing in the network**
  - Compose functions/actors/services flexibly
  - Secure communication and secure data sharing
  - Leveraging network optimally
  - Adapting to service load, network conditions etc.
- **Enable joint optimization considering**
  - Application and network operational requirements
  - Holistic view of communication, computation, caching resources



Example

Packet Interception vs. Dataflow

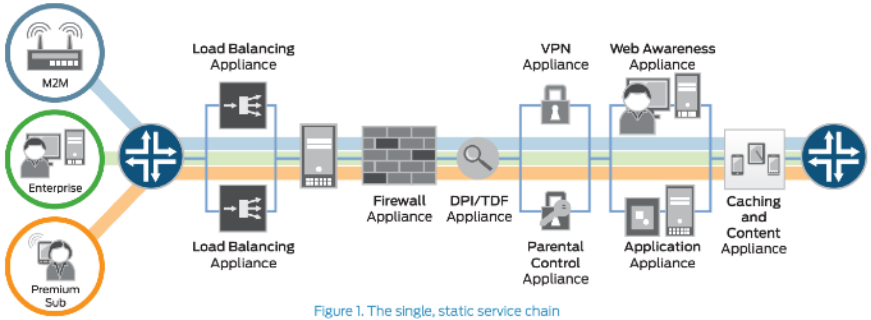
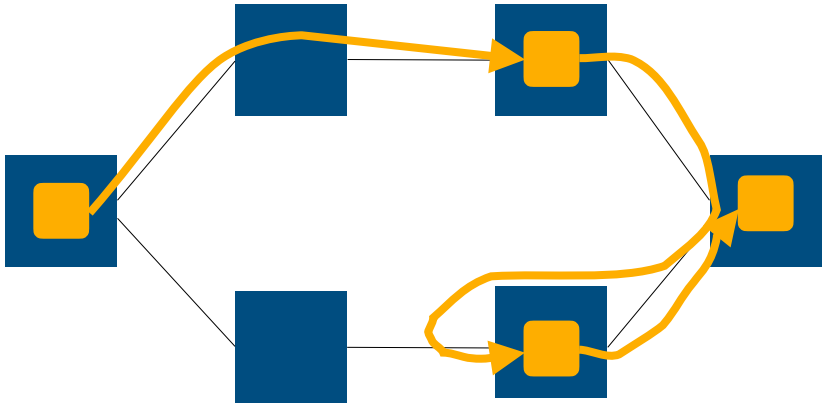


Figure 1. The single, static service chain



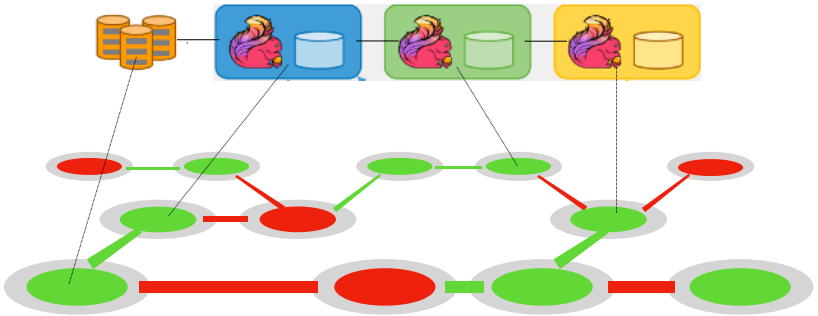
- Transparent middleboxes
- Transformations on packet flow
- Side effects

- Explicit transformations on input data
- Can have side effects
- But principally, focus is on result data

CFN Thesis

Formulated as Questions...

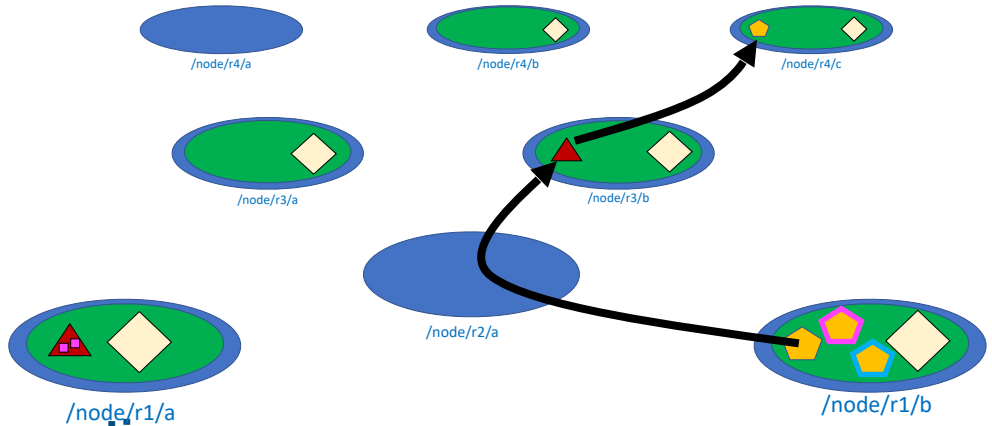
- **Can we re-imagine computing in the network as a principled approach?**
  - Not as an overlay
  - Not as packet manipulation
  - But as ubiquitous service?
- **Can we build simple, light-weight abstractions that leverage programmable network hardware and host platforms?**
  - P4 switches, Smart NICS
  - Lightweight execution environments
- **Can we layer different versions of computing networks**
  - From primitive services to distributed application middleware?
- **Can we leverage joint optimization potential?**
  - By not treating the network as an opaque virtual broadcast domain



Distributed Computing

Many Different Types of Interactions

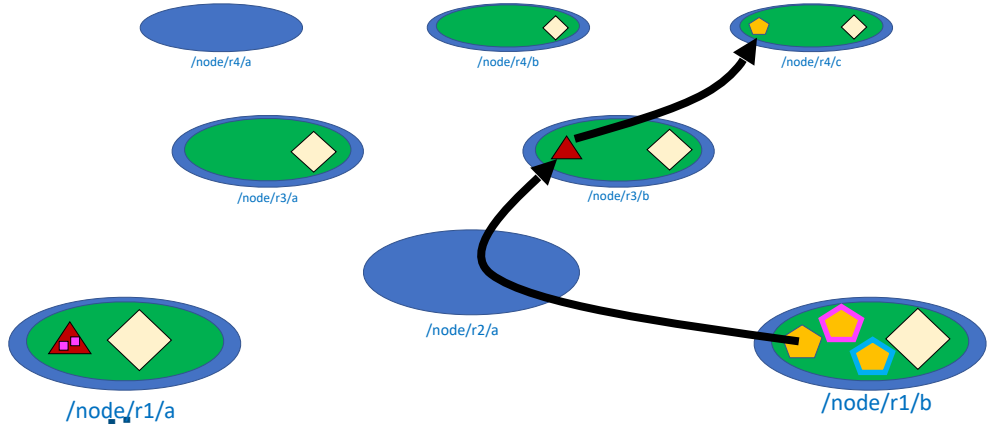
- Message passing
- Remote Method Invocation
- Dataset synchronization
- Key-value store
- Turing-complete distributed computing
- Dataflow
- Collective Communication



Distributed Computing

Many Different Types of Interactions

- Message passing
- **Remote Method Invocation**
- Dataset synchronization
- Key-value store
- **Turing-complete distributed computing**
- **Dataflow**
- Collective Communication



RICE: Remote Method Invocation in ICN

Michał Król
UCL
m.krol@ucl.ac.uk

Karim Habak
Georgia Tech
karim.habak@gatech.edu

David Oran
Network Systems Research & Design
daveoran@orandom.net

Dirk Kutscher
Huawei
dirk.kutscher@huawei.com

Ioannis Psaras
UCL
i.psaras@ucl.ac.uk

Best paper award at ACM ICN-2018

Efficient Remote Method Invocation

- Information-Centric Networking
- Locator-less invocation
- Result caching
- Robust against computational overload attacks

Best paper award at ACM ICN-2018

RICE: Remote Method Invocation in ICN

Michał Król
UCL
m.krol@ucl.ac.uk

Karim Habak
Georgia Tech
karim.habak@gatech.edu

David Oran
Network Systems Research & Design
daveoran@orandom.net

Dirk Kutscher
Huawei
dirk.kutscher@huawei.com

Ioannis Psaras
UCL
i.psaras@ucl.ac.uk

Compute First Networking: Distributed Computing meets ICN

Michał Król[1], Spyridon Mastorakis[2], Dave Oran[3], Dirk Kutscher[4]

[1]University College London/UCLouvain
[2]University of Nebraska, Omaha
[3]Network Systems Research & Design
[4]University of Applied Sciences Emden/Leer

Efficient Remote Method Invocation

- Information-Centric Networking
- Locator-less invocation
- Result caching
- Robust against computational overload attacks

Turing-complete distributed computing

- On-demand function offload considering result data availability
- Considering resource availability and observed performance
- Joint optimization of network and computation resources

Best paper award at ACM ICN-2018

RICE: Remote Method Invocation in ICN

Michał Król
UCL
m.krol@ucl.ac.uk

Karim Habak
Georgia Tech
karim.habak@gatech.edu

David Oran
Network Systems Research & Design
daveoran@orandom.net

Dirk Kutscher
Huawei
dirk.kutscher@huawei.com

Ioannis Psaras
UCL
i.psaras@ucl.ac.uk

Compute First Networking: Distributed Computing meets ICN

Michał Król[1], Spyridon Mastorakis[2], Dave Oran[3], Dirk Kutscher[4]

[1]University College London/UCLouvain
[2]University of Nebraska, Omaha
[3]Network Systems Research & Design
[4]University of Applied Sciences Emden/Leer

Vision: Information-Centric Dataflow – Re-Imagining Reactive Distributed Computing

Dirk Kutscher
University of Applied Sciences Emden/Leer
Emden, Germany
Dirk.Kutscher@hs-emden-leer.de

Laura Al Wardani
University of Applied Sciences Emden/Leer
Emden, Germany
laura.al.wardani@hs-emden-leer.de

T M Rayhan Gias
University of Applied Sciences Emden/Leer
Emden, Germany
rayhan.gias@hs-emden-leer.de

Decentralized ICN-based Dataflow System Implementation

Laura Al Wardani
University of Applied Sciences Emden/Leer
Emden, Germany
laura.al.wardani@hs-emden-leer.de

T M Rayhan Gias
University of Applied Sciences Emden/Leer
Emden, Germany
rayhan.gias@hs-emden-leer.de

Dirk Kutscher
University of Applied Sciences Emden/Leer
Emden, Germany
Dirk.Kutscher@hs-emden-leer.de

Efficient Remote Method Invocation

- Information-Centric Networking
- Locator-less invocation
- Result caching
- Robust against computational overload attacks

Turing-complete distributed computing

- On-demand function offload considering result data availability
- Considering resource availability and observed performance
- Joint optimization of network and computation resources

Information-Centric Dataflow

- Fully decentralized (no orchestrator)
- Efficient function output sharing (multicast)
- Receiver-driven operation with joint control loop considering computing and networking resources
- Automatic scaling based on observed performance

ICN Applications

Distributed Vision Processing

Vision: Information-Centric Dataflow – Re-Imagining Reactive Distributed Computing

Dirk Kutscher
University of Applied Sciences
Emden/Leer
Emden, Germany
Dirk.Kutscher@hs-emden-leer.de

Laura Al Wardani
University of Applied Sciences
Emden/Leer
Emden, Germany
laura.al.wardani@hs-emden-leer.de

T M Rayhan Gias
University of Applied Sciences
Emden/Leer
Emden, Germany
rayhan.gias@hs-emden-leer.de

ABSTRACT

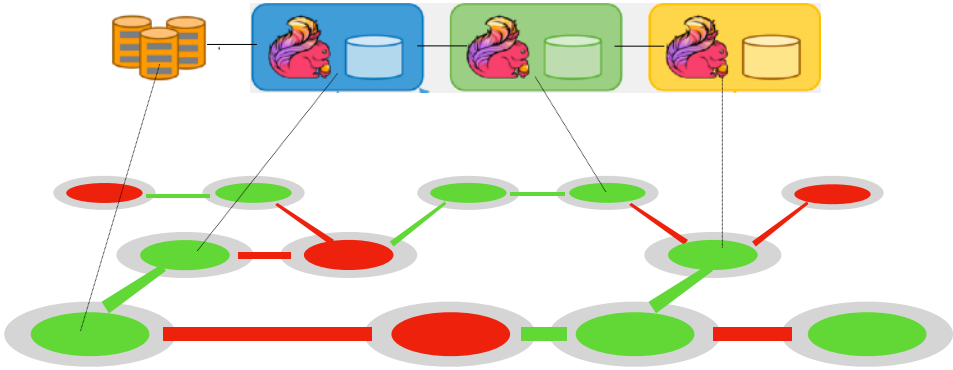
This paper describes an Information-Centric Dataflow system that is based on name-based access to computation results, NDN PSync dataset synchronization for enabling consuming compute functions to learn about updates and for coordinating the set of compute functions in a distributed Dataflow pipeline. We describe how relevant Dataflow concepts can be mapped to ICN and how data-sharing, data availability and scalability can be improved compared to state-of-the-art systems. We also provide a specification of an application-independent namespace design and report on our experience with a first prototype implementation.

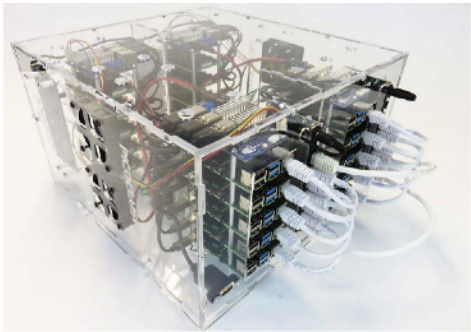
KEYWORDS

Information-Centric Networking, Distributed Computing, Dataflow

relationships with upstream producers and downstream consumers. For example, when parallelizing a computation step, it typically implies that each instance is consuming a partition of the inputs instead of all the inputs. **An indirection- and connection-based approach makes it harder to configure (and especially to dynamically re-configure) such dataflow graphs.**

In some variants of Dataflow, for example stream processing, it can be attractive if one computation output can be consumed by multiple downstream functions. Connection-based overlays typically require **duplicating the data for each such connection**, incurring significant overheads. In large-scale scenarios, the computation functions may be distributed to multiple hosts that are inter-connected in a network. Orchestrators may have visibility into compute resource availability but typically have to treat the TCP/IP network as a blackbox. As a result, the actual **data flow is**

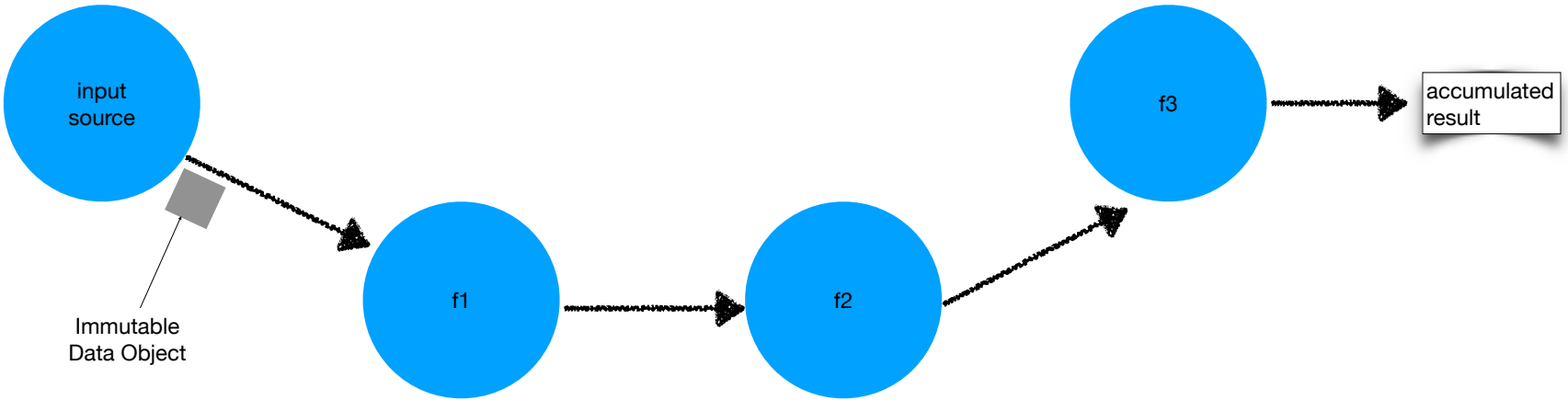




- **Highly scalable and self-organized Dataflow Processing**
- **Works without edge orchestration (Kubernetes etc.)**
- **Facilitates application development and deployment (compute graph specification)**

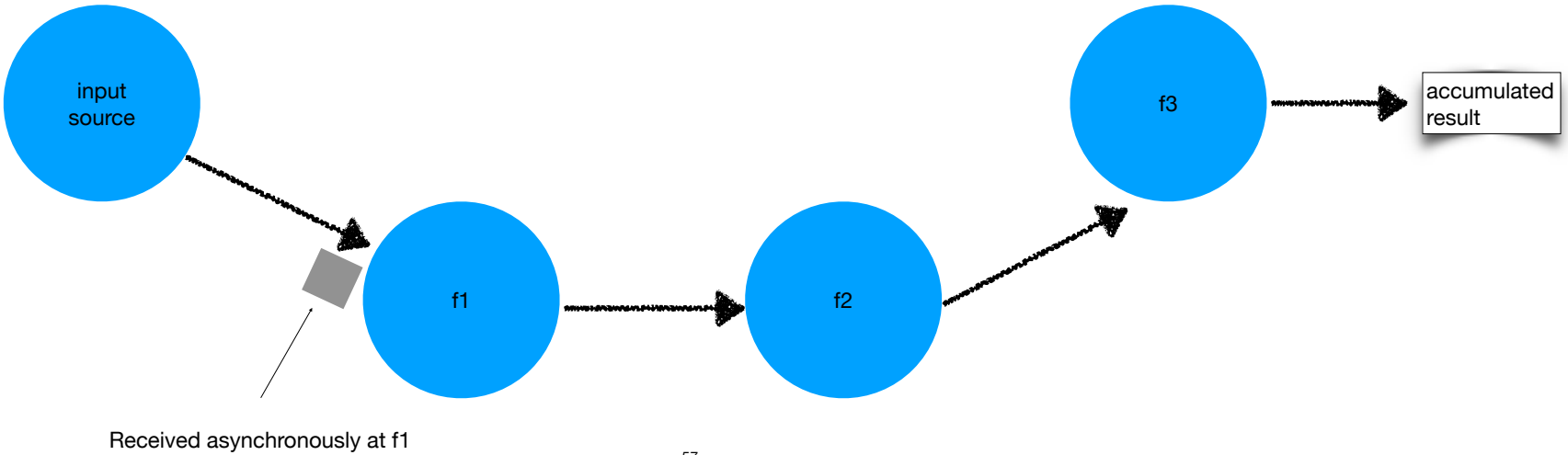
Dataflow

Structured Distributed Data Processing



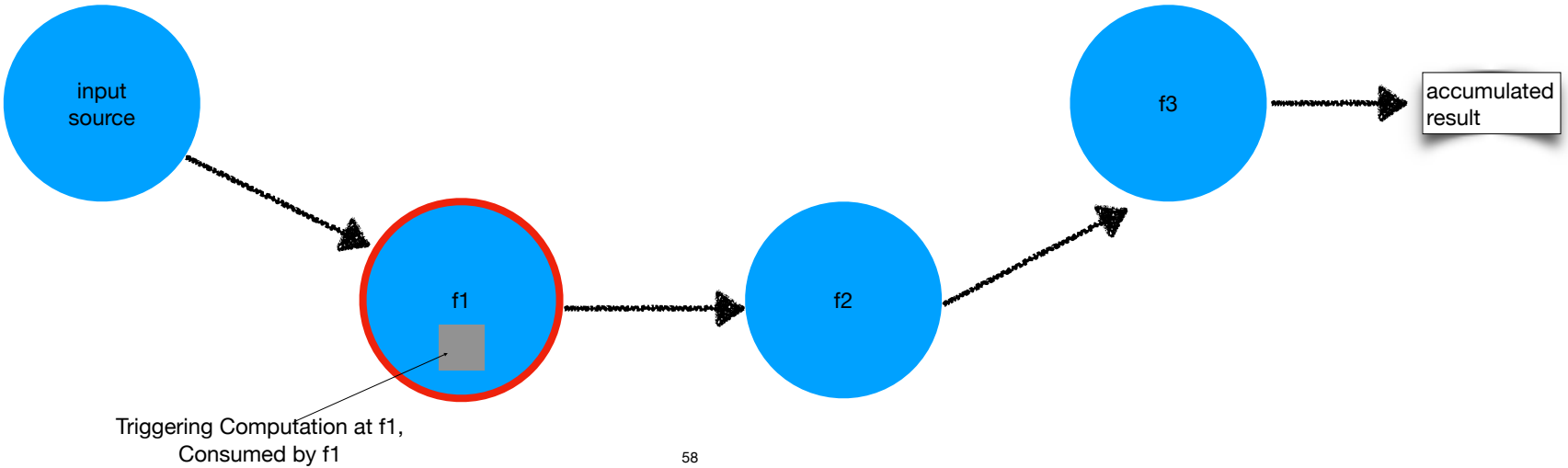
Dataflow

Structured Distributed Data Processing



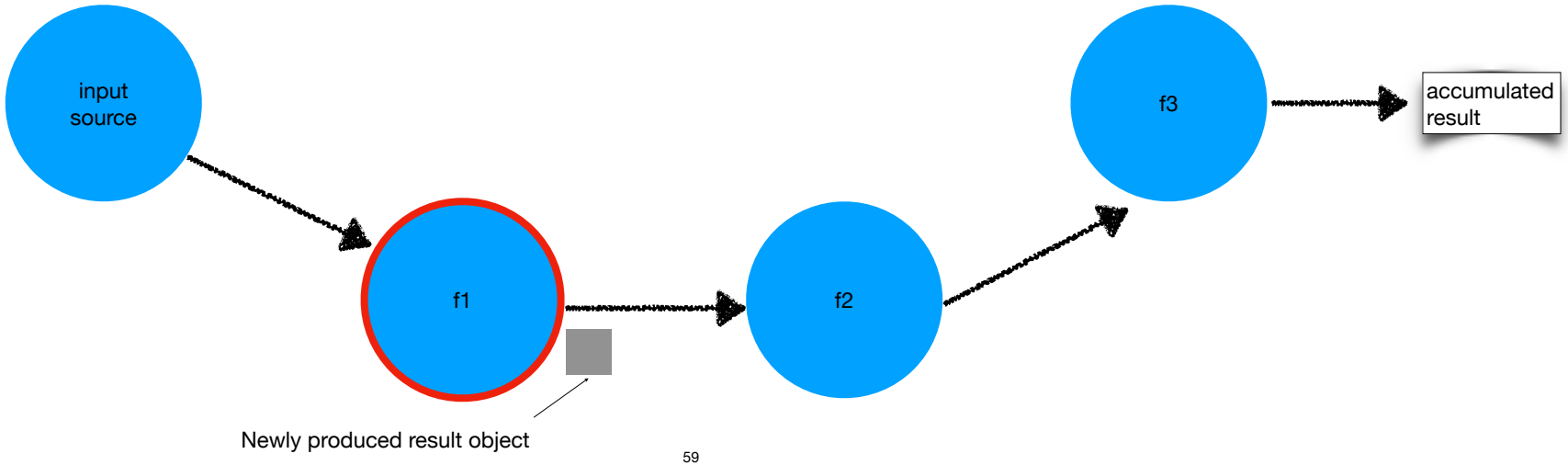
Dataflow

Structured Distributed Data Processing



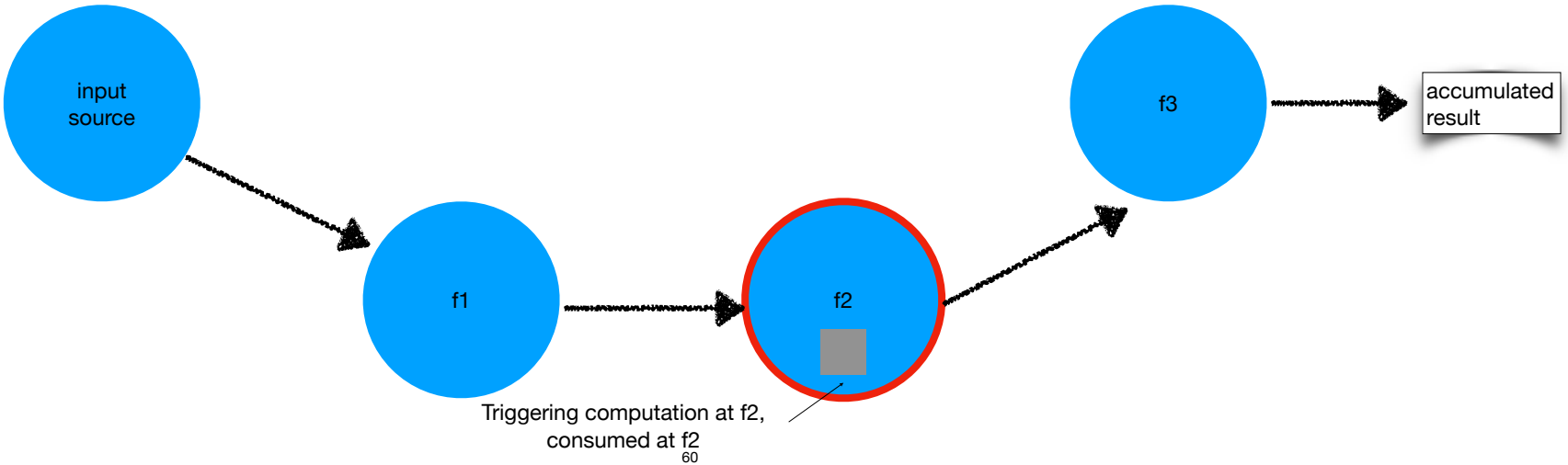
Dataflow

Structured Distributed Data Processing



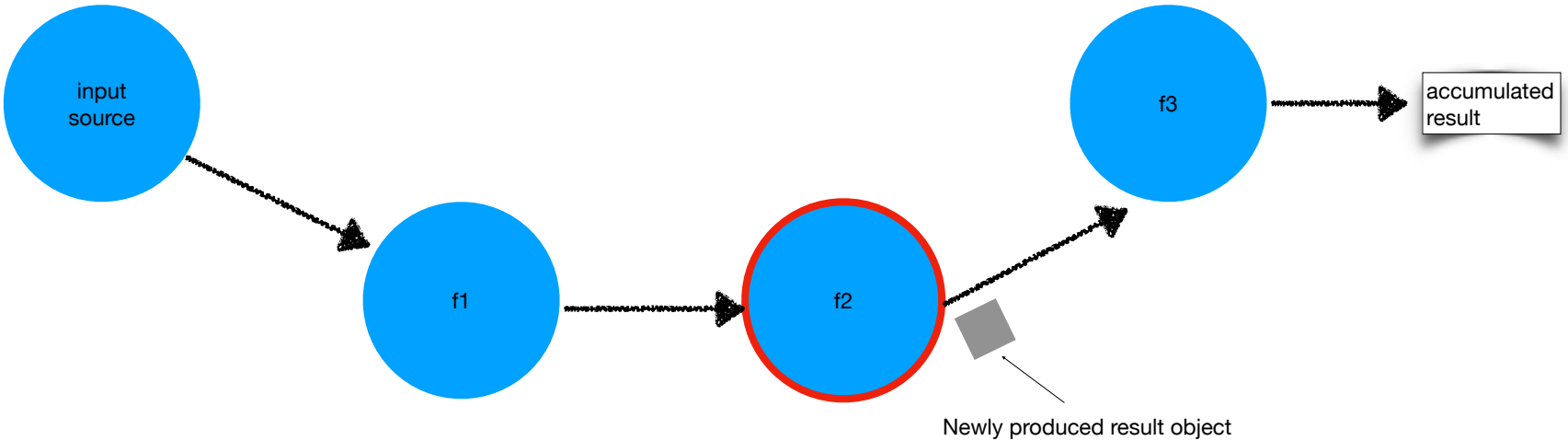
Dataflow

Structured Distributed Data Processing



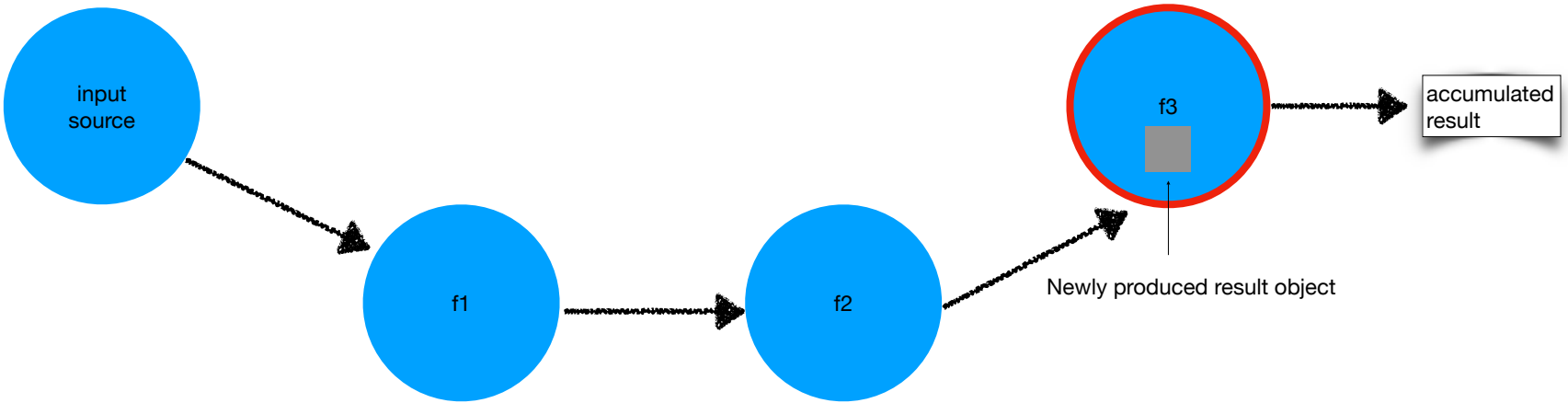
Dataflow

Structured Distributed Data Processing



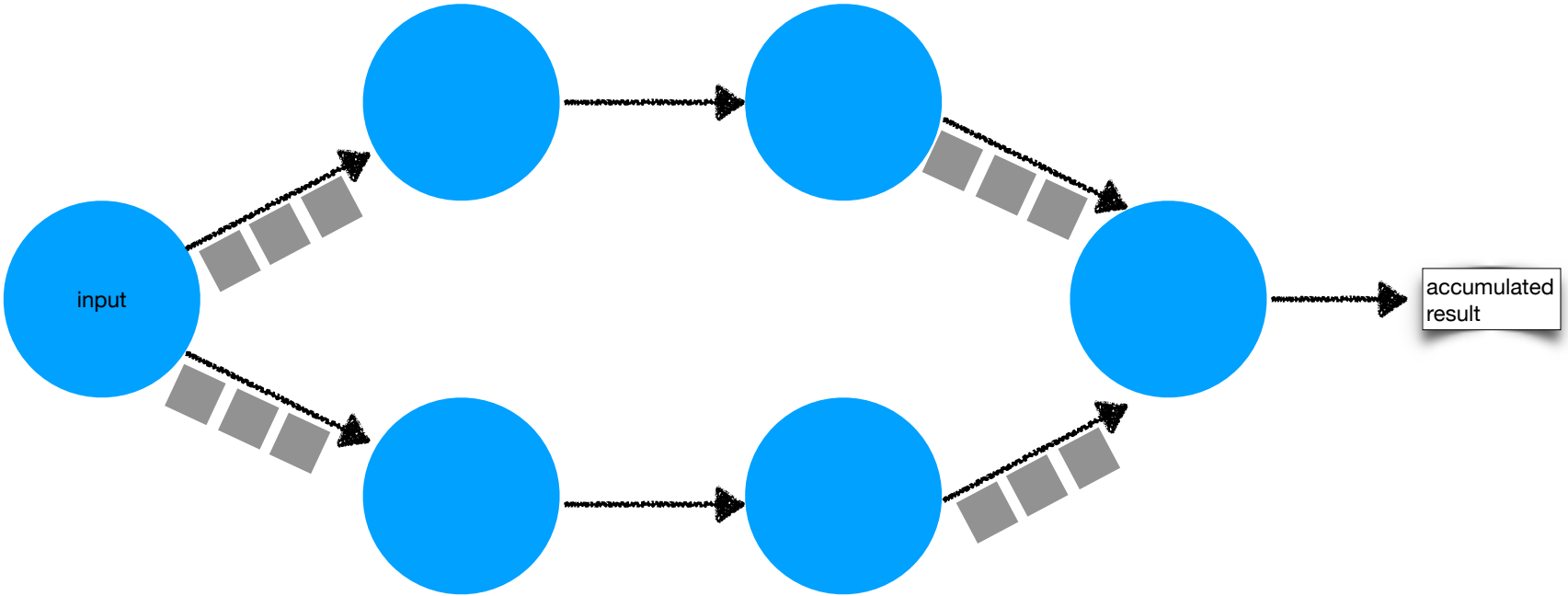
Dataflow

Structured Distributed Data Processing



Dataflow

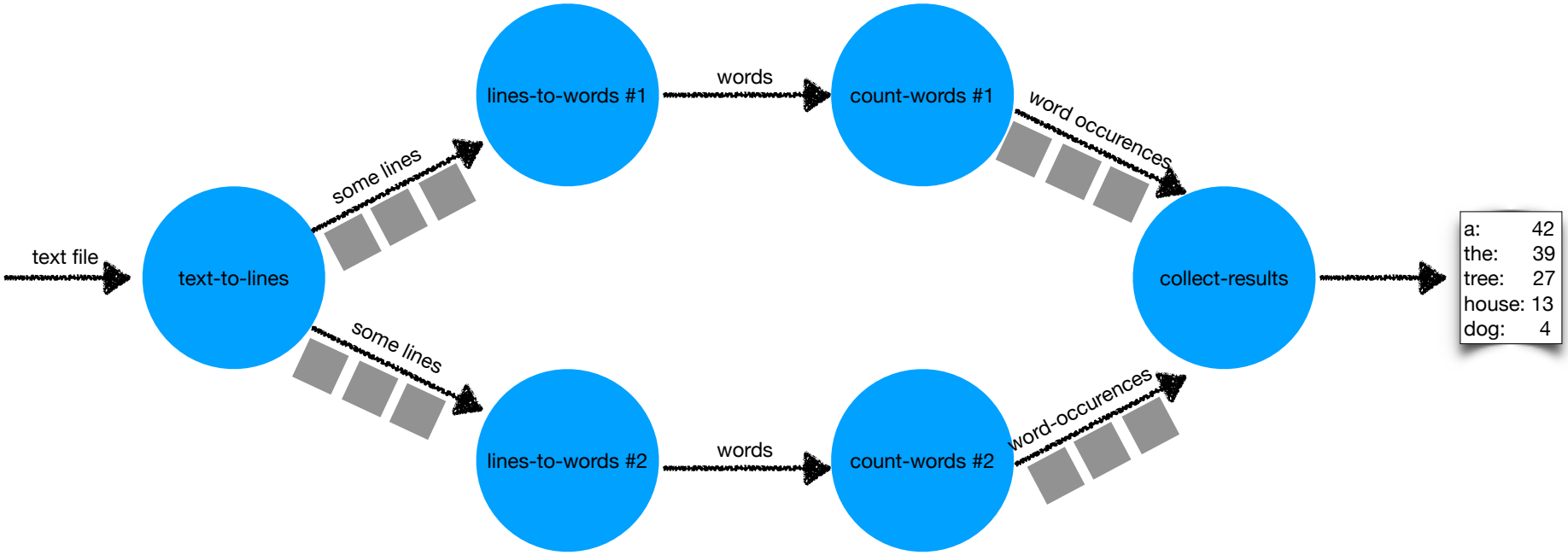
Structured Distributed Data Processing



Dataflow

Poster Child Example: word-count

```
text-to-lines -> lines-to-words -> word-count ->
    ↪ collect-results
```



IceFlow

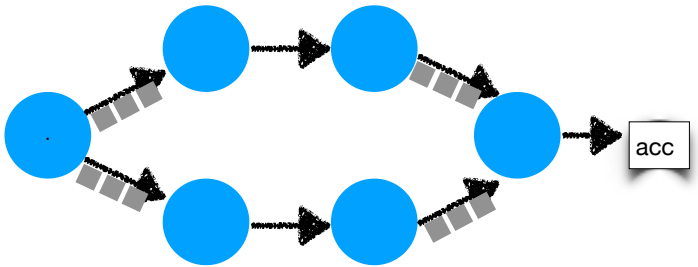
Concepts

- **Just Names**
  - For infrastructure
  - And for functions
- **Computation results as Named Data Objects**
  - Usual ICN properties:
  - Implicit multi-destination distribution
  - Opportunistic caching
  - In-network re-transmissions
  - Forwarding strategies
- **Flow control**
  - Some coupling between consumers and producers
  - Considering compute resources
  - Back-pressure propagation

```
/[app]/[actor]/[instance]/data/[partition]/[object]
```

app	the name of the application
actor	the name of a Dataflow actor
instance	actor instance number
partition	monotonically increasing partition number to structure data objects on the producer's side
object	monotonically increasing sequence number

```
/word-count/text-to-lines/1/data/1/1
/word-count/lines-to-words/2/data/3/27
```



Collective Communication and ICN

Promising

- Data-oriented communication model
- Locator-less model conducive to data production and consumption at different places in the network (computing)
- Multi-destination delivery included
- In-network retransmission and caching could help with reliability and performance

Collective Communication and ICN

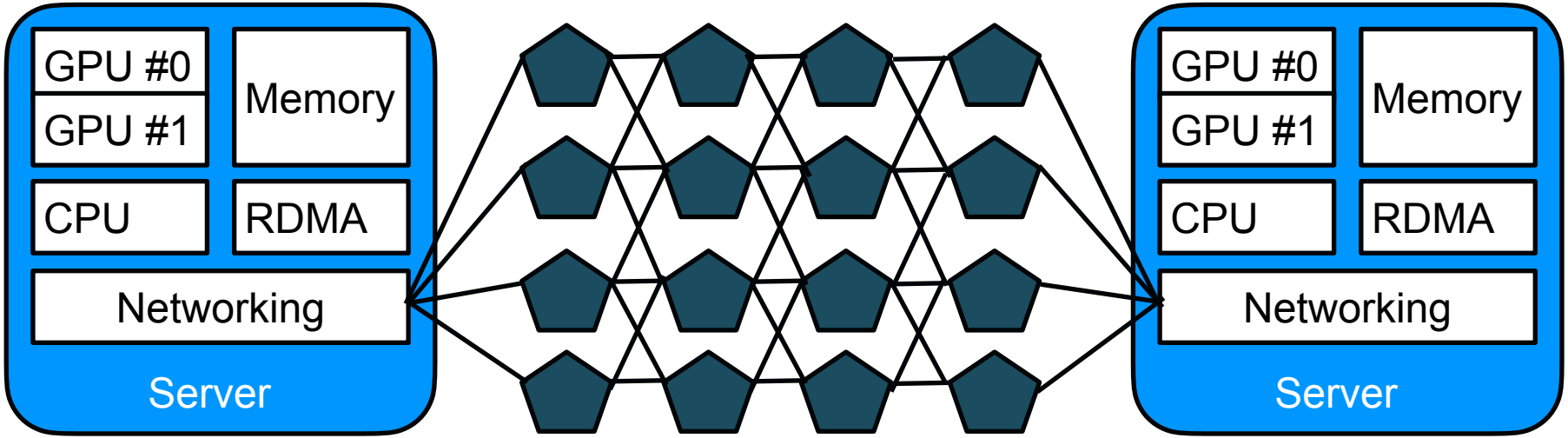
Promising

- Data-oriented communication model
- Locator-less model conducive to data production and consumption at different places in the network (computing)
- Multi-destination delivery included
- In-network retransmission and caching could help with reliability and performance

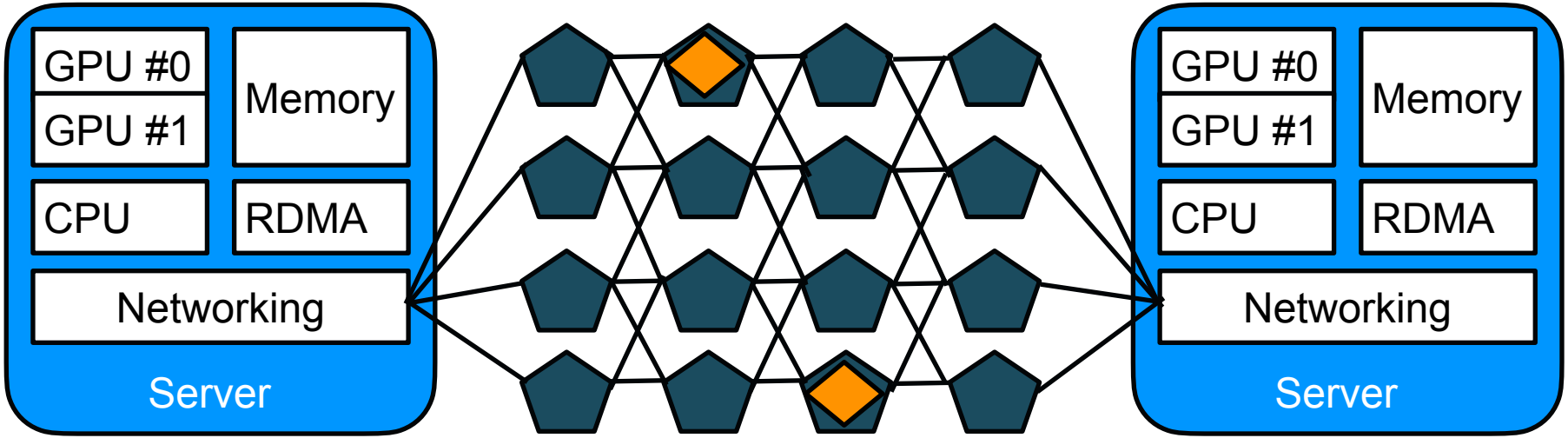
Challenges

- Receiver-driven transport results in polling – efficient enough?
- RDMA-like communication unexplored
- Security concept: data-oriented security good – unclear whether it can be afforded
- Exact scheduling may be at odds with current ICN system design – more work needed

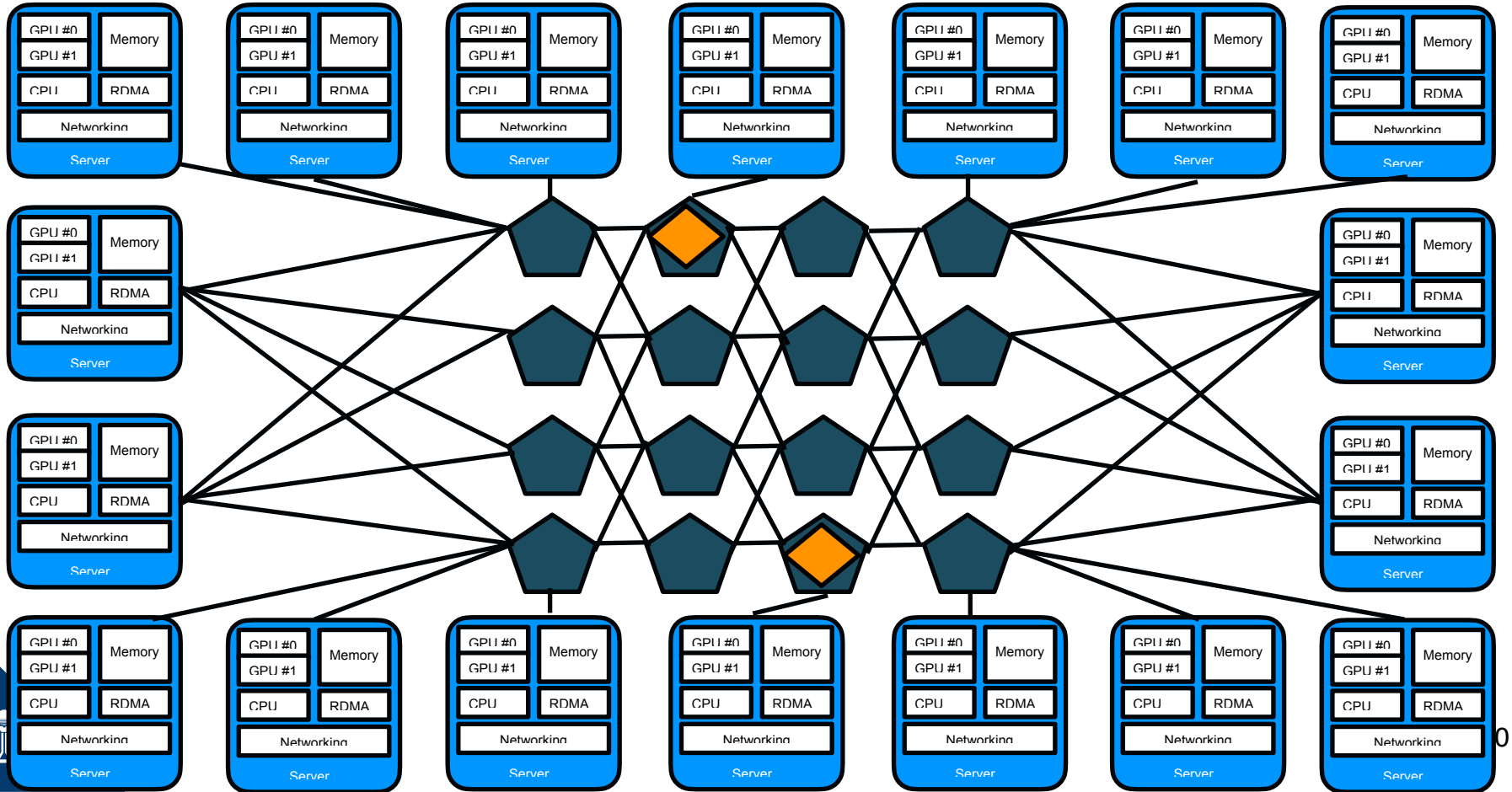
Elements of Data-Oriented Collective Communication



Elements of Data-Oriented Collective Communication

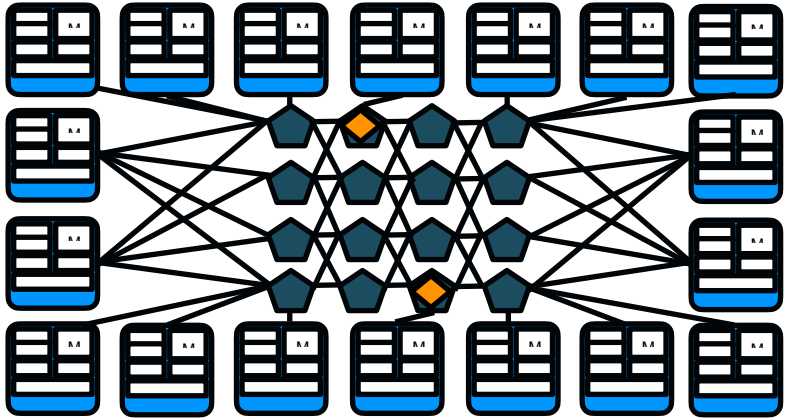


Elements of Data-Oriented Collective Communication



Elements of Data-Oriented Collective Communication

- **Static graph generation**
  - workload specification
  - topology knowledge
  - placement of collective communication functions
- **Forwarding information**
  - names for workers and functions
  - link-state routing
- **Naming**
  - name workers and functions
  - functions are first-class citizens
- **Data transport**
  - Receiver-driven Interest-Data
  - Implicit multi-destination transport through Interest aggregation
  - Soft-state pub/sub?
  - Explicit Interest retransmission for repair, potentially in-network
- **Optimization**
  - Path selection and scheduling: achieve optimality with respect to level of parallelism and path utilization



Directions for Computing in the Network

draft-irtf-coinrg-dir

COINRG
Internet-Draft
Intended status: Experimental
Expires: 9 February 2024

D. Kutscher
HKUST(GZ)
T. Kaerkkaeinen
J. Ott
Technical University Muenchen
8 August 2023

Directions for Computing in the Network
draft-irtf-coinrg-dir-00

Abstract

In-network computing can be conceived in many different ways -- from active networking, data plane programmability, running virtualized functions, service chaining, to distributed computing.

This memo proposes a particular direction for Computing in the Networking (COIN) research and lists suggested research challenges.

Collective Communication Transport

draft-yao-tsvwg-cco-problem-statement-and-usecases-00
draft-yao-tsvwg-cco-requirement-and-analysis-00

Workgroup: Transport Area Working Group
Internet-Draft:
draft-yao-tsvwg-cco-problem-statement-and-usecases-00
Published: 23 October 2023
Intended Status: Informational
Expires: 25 April 2024

K. Yao
China Mobile
S. Xu
China Mobile
Y. Li
Huawei Technologies
H. Huang
Huawei Technologies
D. KUTSCHER
HKUST (Guangzhou)

Collective Communication Optimization: Problem Statement and Use cases

Abstract

Collective communication is the basic logical communication model for distributed applications. When distributed systems scales, the communication overhead becomes the bottleneck of the entire system, impeding system performance to increase. This draft describes the performance challenges when the collective communication is employed in a network with more nodes or processes participating in or a larger number of such communication rounds required to complete a single job. And the document presents several use cases where different aspects of collective communication optimization are needed.

Workgroup: Transport Area Working Group
Internet-Draft:
draft-yao-tsvwg-cco-requirement-and-analysis-00
Published: 23 October 2023
Intended Status: Informational
Expires: 25 April 2024

K. Yao
China Mobile
S. Xu
China Mobile
Y. Li
Huawei Technologies
H. Huang
Huawei Technologies
D. KUTSCHER
HKUST (Guangzhou)

Collective Communication Optimization: Requirement and Analysis

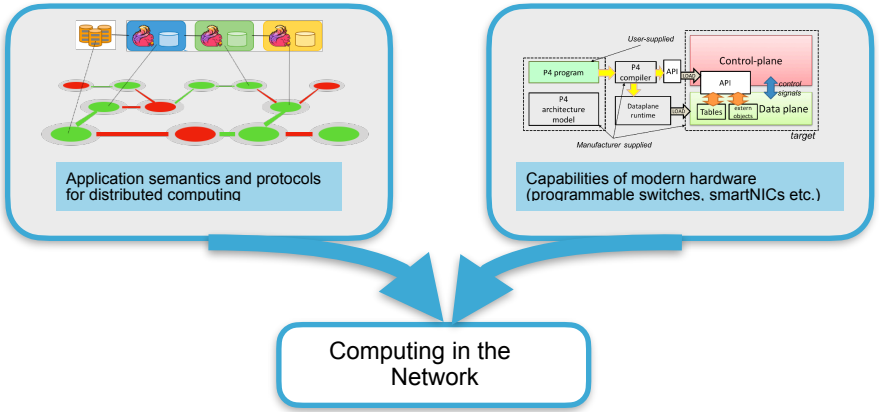
Abstract

As is mentioned in draft [CCO PS & USECASE], the most obvious problem on why existing protocols cannot meet the high-performance requirements of collective communications is that these distributed applications are not co-designed with the underlying networking protocols. There is a semantic gap between inter-process message transportation and packet forwarding, which should be bridged by efficient mapping and optimization.

This draft further presents the technical requirements on how the collective communication optimization should be designed, and makes some discussion and analysis on several related work.

Conclusions

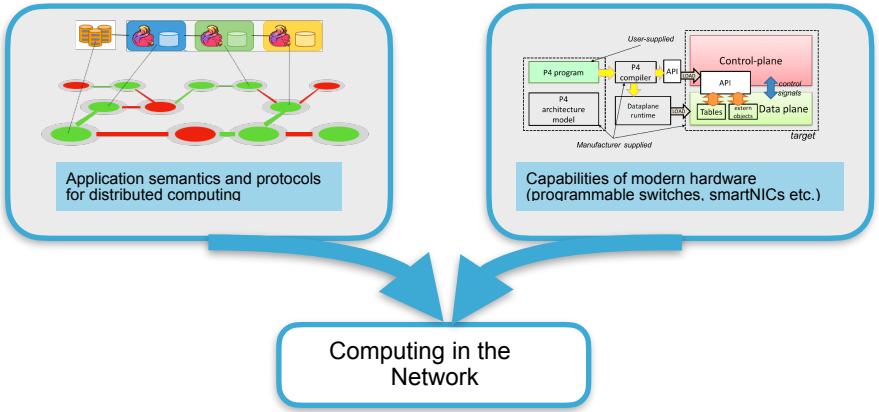
Computing in the Network



- **Distributed machine learning drives the development of new concepts for communication and computing**
  - Efficient multi-destination communication
  - Efficient mapping of MPI-inspired Collective Communication
- **Current abstractions do not fit well**
  - IP packet level communication too limited
  - Connection-oriented transport at odds with communication semantics
- **Data-oriented communication attractive**
  - Locator-less, conducive to data production and consumption
  - Challenging performance requirement call for more research and possibly evolution of current ICN protocols

Conclusions

Computing in the Network



- **Distributed machine learning drives the development of new concepts for communication and computing**
  - Efficient multi-destination communication
  - Efficient mapping of MPI-inspired Collective Communication
- **Current abstractions do not fit well**
  - IP packet level communication too limited
  - Connection-oriented transport at odds with communication semantics
- **Data-oriented communication attractive**
  - Locator-less, conducive to data production and consumption
  - Challenging performance requirement call for more research and possibly evolution of current ICN protocols

Great area for systems research!

Dirk KUTSCHER
德克·库彻

Come work with us at HKUST in Guangzhou!
Now hiring PostDocs & PhD students!

香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

dirk-kutscher.info
dku@hkust-gz.edu.cn