Towards An Application Objective-Aware Network Interface

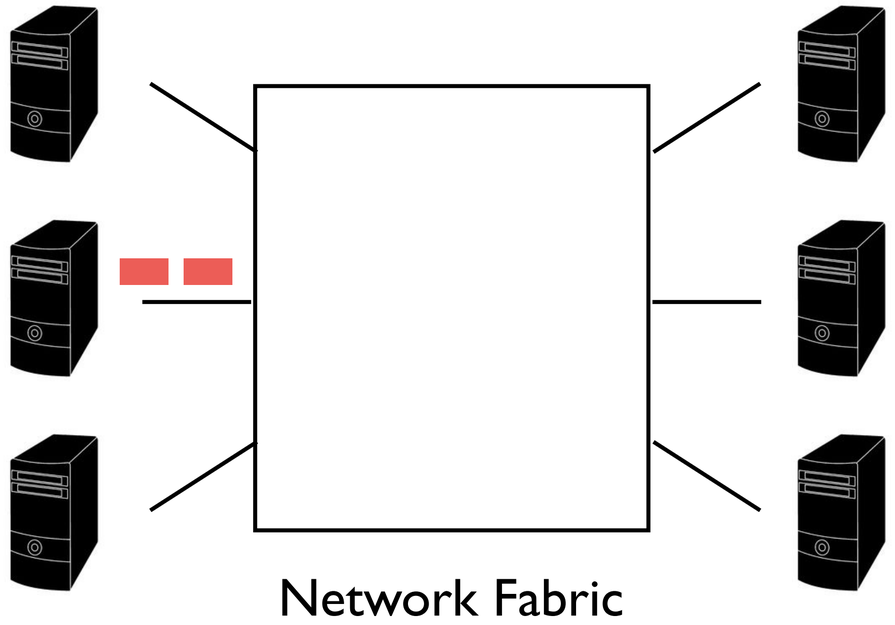
Sangeetha Abdu Jyothi Sayed Hadi Hashemi Roy Campbell Brighten Godfrey

UC Irvine ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN vmware®

HotCloud'20

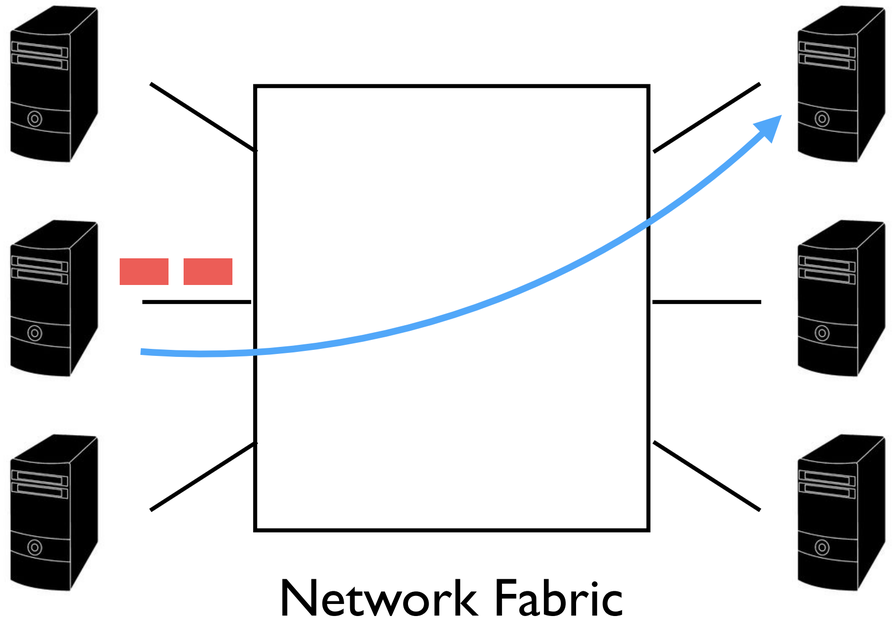
Evolution of Application Network Interface (ANI)

ANI	Metrics
Packet	Delay, jitter



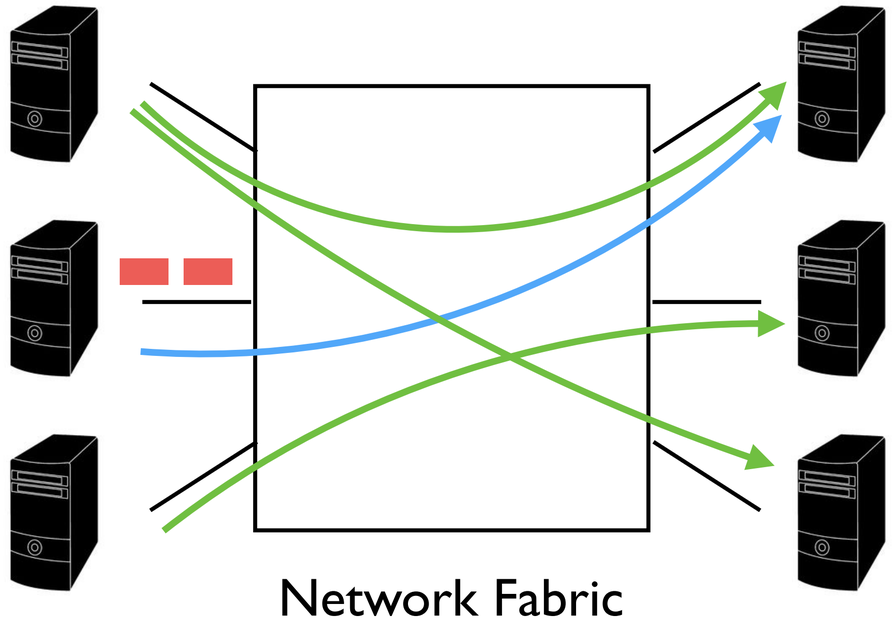
Evolution of Application Network Interface (ANI)

ANI	Metrics
Packet	Delay, jitter
↓	
Flow	Flow Completion Time



Evolution of Application Network Interface (ANI)

ANI	Metrics
Packet	Delay, jitter
Flow	Flow Completion Time
Coflow	Coflow Completion Time



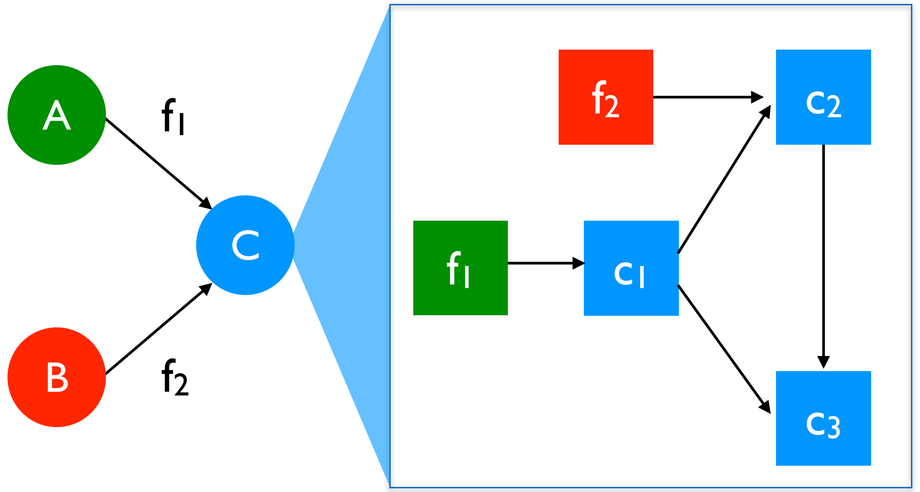
What is the ultimate goal of an ANI?

Translating
application requirements to
actionable network requirements

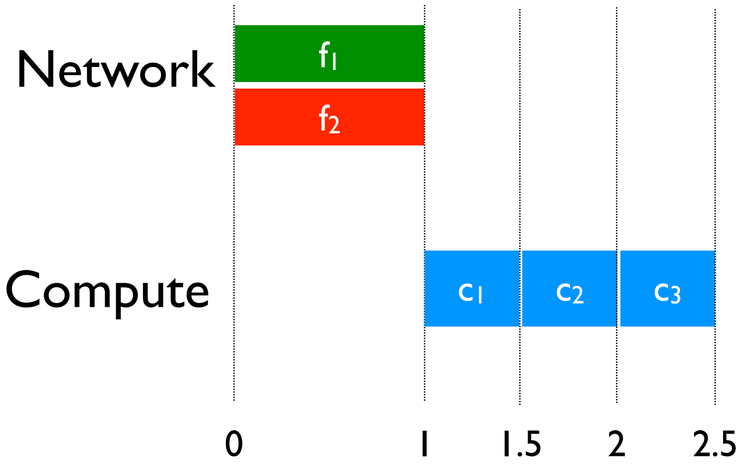
Are current ANIs sufficient?

Understanding an Application's Objective

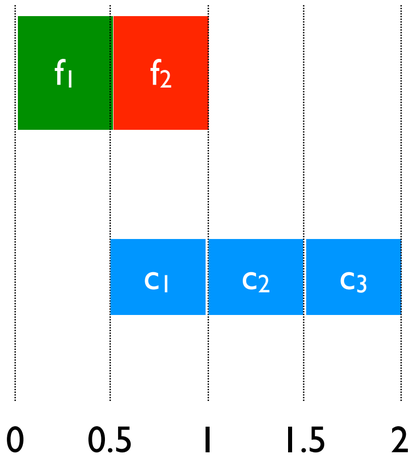
- Applications have complex interdependencies between computation and communication
- Prioritizing flows based on computations in succeeding stage is critical



Coflow-Optimized



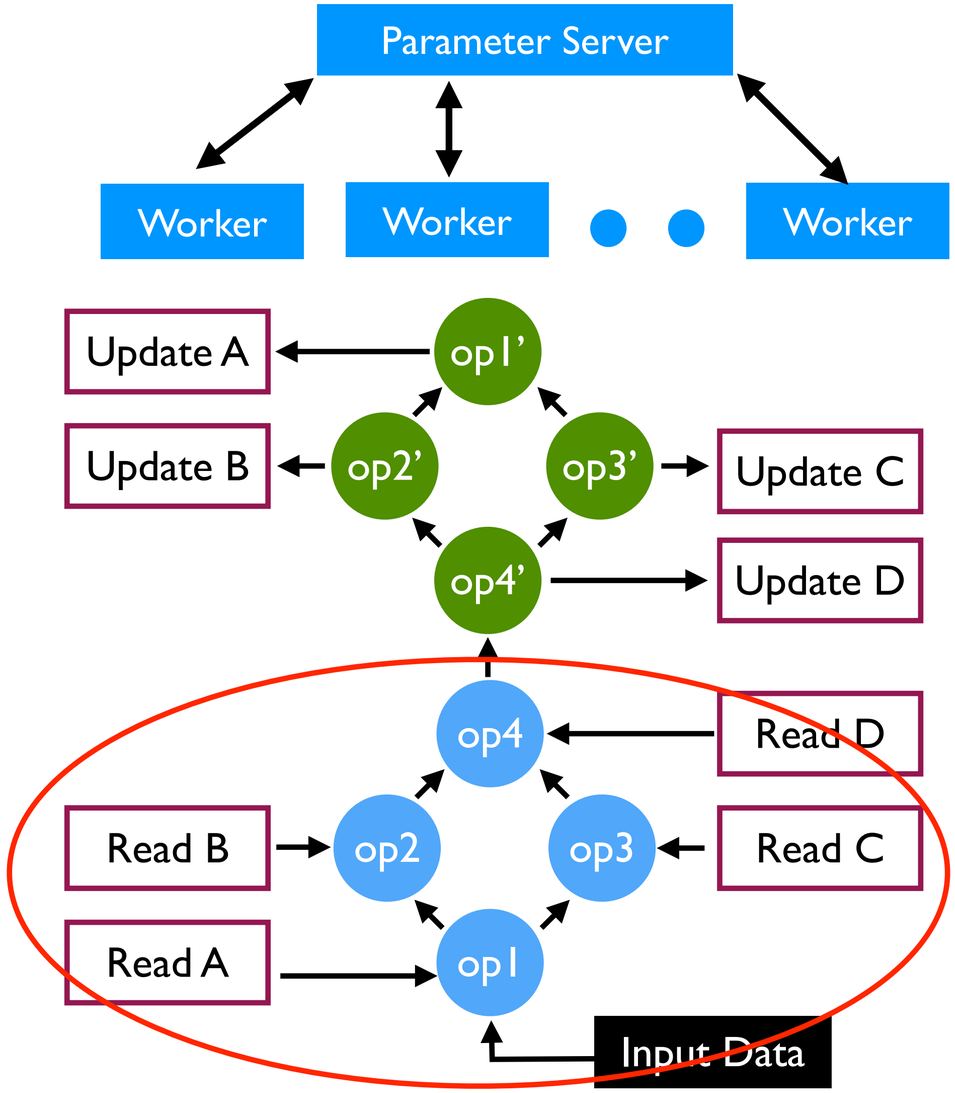
Performance-Optimized



Current abstractions fail to capture application objective effectively

An Example Application: Distributed Deep Learning

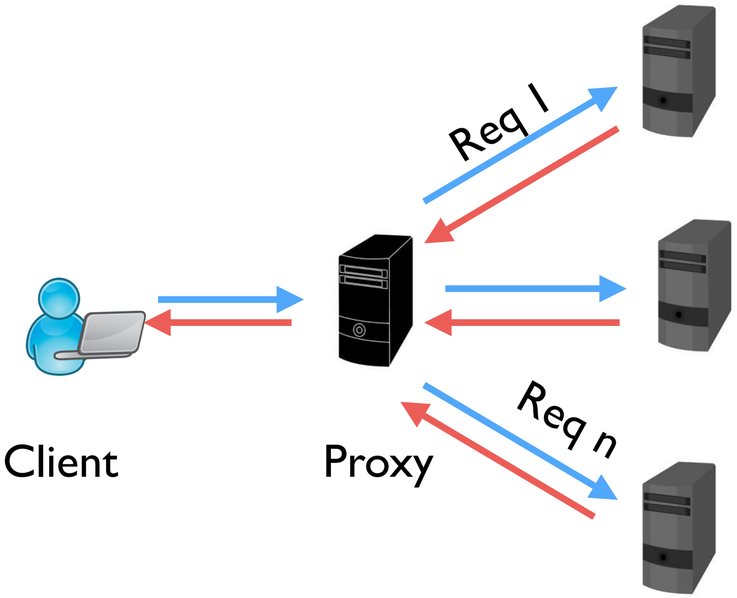
- Gigabytes of data transferred in each iteration which lasts milliseconds
  (e.g., VGG-16 send ~1GB data every 200ms)
- Parameters consumed in a particular order
- Parameter updates from PS to workers send in the best order can accelerate training



Sample TensorFlow Model: One Iteration

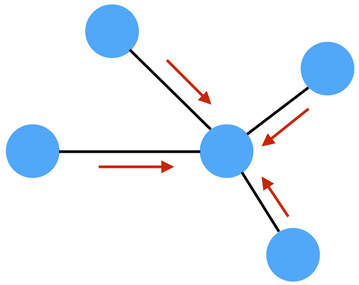
Other Applications

- User-facing partition-aggregation workloads
  (remote dependency resolution at a Web proxy)

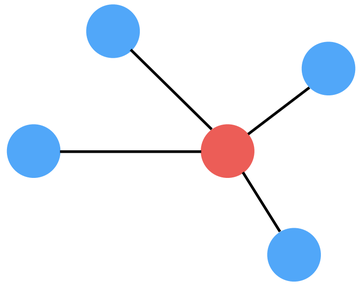


- Graph processing systems

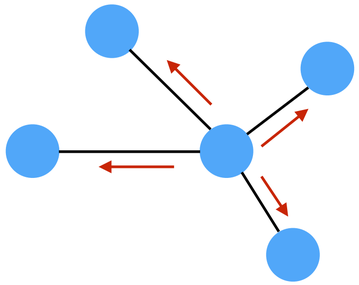
- Iterative analytics with deadlines (eg: Naiad) and so on …



Gather



Update



Scatter

Towards A Novel Application Network Interface

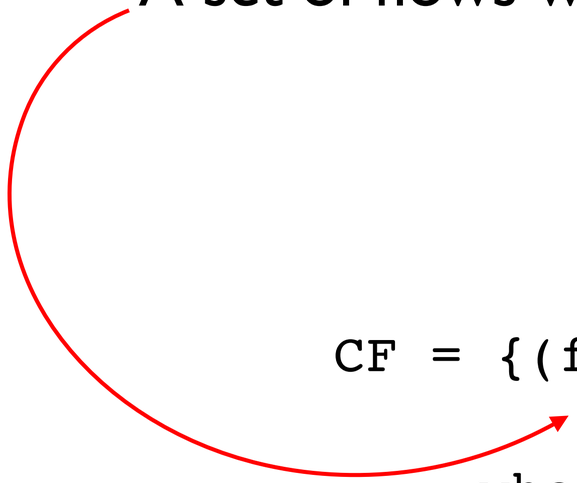
- Computation completely represented by a DAG. What is the network equivalent?
- The goal is to capture an application's network objective
- CadentFlow:

$$CF = \{(f_1, T_1), (f_2, T_2), \ldots, (f_n, T_n), \Gamma\}$$

$$\text{where} \quad T_i = (t_{i1}, m_{i1}), (t_{i2}, m_{i2}) \ldots$$

Towards A Novel Application Network Interface

- Computation completely represented by a DAG. What is the network equivalent?
- The goal is to capture an application's network objective
- CadentFlow:
  - A set of flows with metrics AND



$$CF = \{(f_1, T_1), (f_2, T_2), \ldots, (f_n, T_n), \Gamma\}$$

$$\text{where} \quad T_i = (t_{i1}, m_{i1}), (t_{i2}, m_{i2}) \ldots$$

Towards A Novel Application Network Interface

- Computation completely represented by a DAG. What is the network equivalent?
- The goal is to capture an application's network objective
- CadentFlow:
  - A set of flows with metrics AND
  - An application-level objective

$$CF = \{(f_1, T_1), (f_2, T_2), \ldots, (f_n, T_n), \Gamma\}$$

$$\text{where} \quad T_i = (t_{i1}, m_{i1}), (t_{i2}, m_{i2}) \ldots$$

Towards A Novel Application Network Interface

- Computation completely represented by a DAG. What is the network equivalent?
- The goal is to capture an application's network objective
- CadentFlow:
  - A set of flows with metrics AND
  - An application-level objective
  - Metrics may be priority, deadline, weight, etc.

$$CF = \{(f_1, T_1), (f_2, T_2), \ldots, (f_n, T_n), \Gamma\}$$

$$\text{where} \quad T_i = (t_{i1}, m_{i1}), (t_{i2}, m_{i2}) \ldots$$

Towards A Novel Application Network Interface

- Computation completely represented by a DAG. What is the network equivalent?
- The goal is to capture an application's network objective
- CadentFlow:
  - A set of flows with metrics AND
  - An application-level objective
  - Metrics may be priority, deadline, weight, etc.

$$CF = \{(f_1, T_1), (f_2, T_2), \ldots, (f_n, T_n), \Gamma\}$$

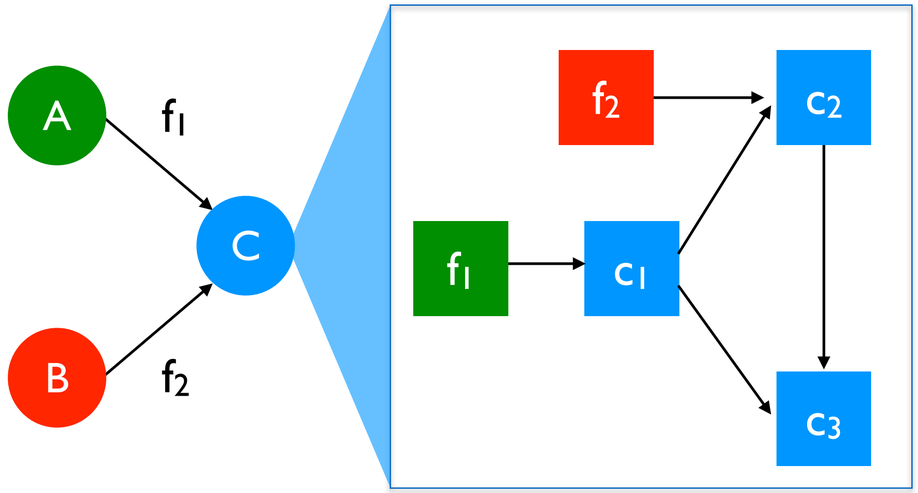
$$\text{where} \quad T_i = (t_{i1}, m_{i1}), (t_{i2}, m_{i2}) \ldots$$

Defining CCT flexibility ratio

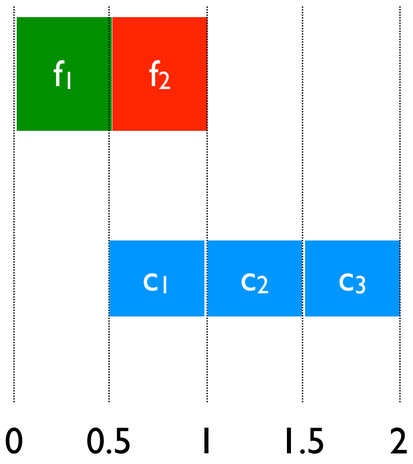
- When computation is the bottleneck, CadentFlow with deadlines provide flexibility for *delaying* some flows without affecting application performance

- In the example, best Coflow Completion Time (CCT) is 1s, but upto 1.5s is tolerable without any impact

- CCT flexibility ratio = $\frac{\text{Max tolerable CCT}}{\text{Min CCT}}$

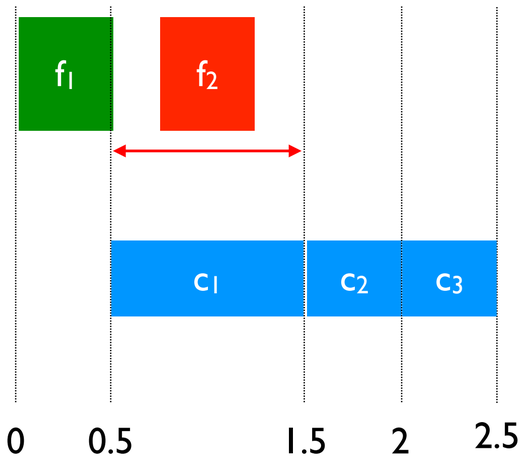


Performance-Optimized



c1 takes 0.5s

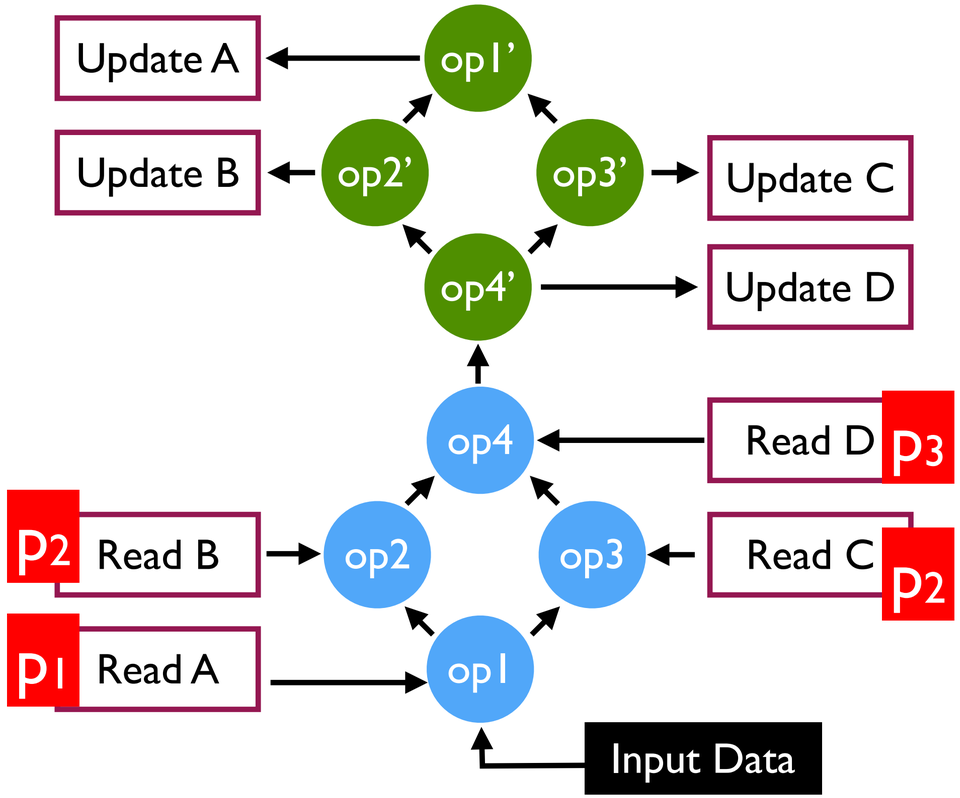
Performance-Optimized



c1 takes 1s

Distributed DNN Training CadentFlow

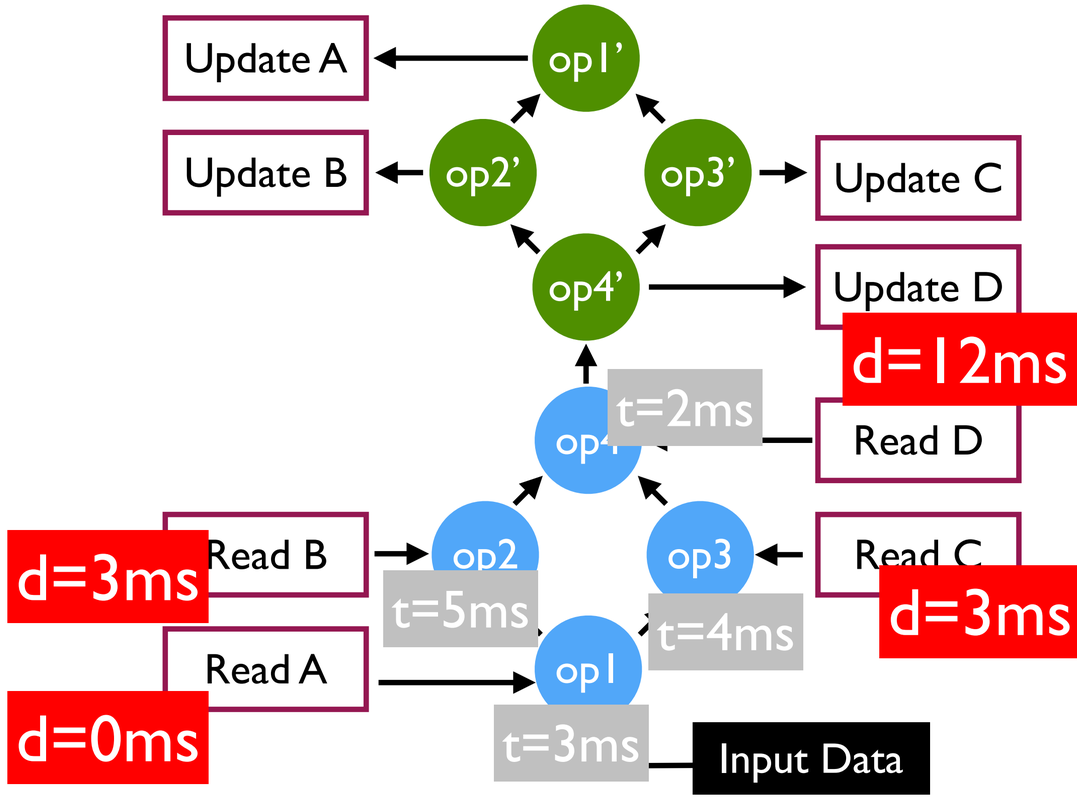
- Priority-based
  - Assign priorities based on DAG structure
  - Objective: Minimize completion time subject to priorities



Sample TensorFlow Model: One Iteration

Distributed DNN Training CadentFlow

- Priority-based
  - Assign priorities based on DAG structure
  - Objective: Minimize completion time subject to priorities
- Deadline-based
  - Assign deadlines based on per-op computation time
  - Objective: Minimize $\max_i(\text{endTime}_i - \text{deadline}_i)$

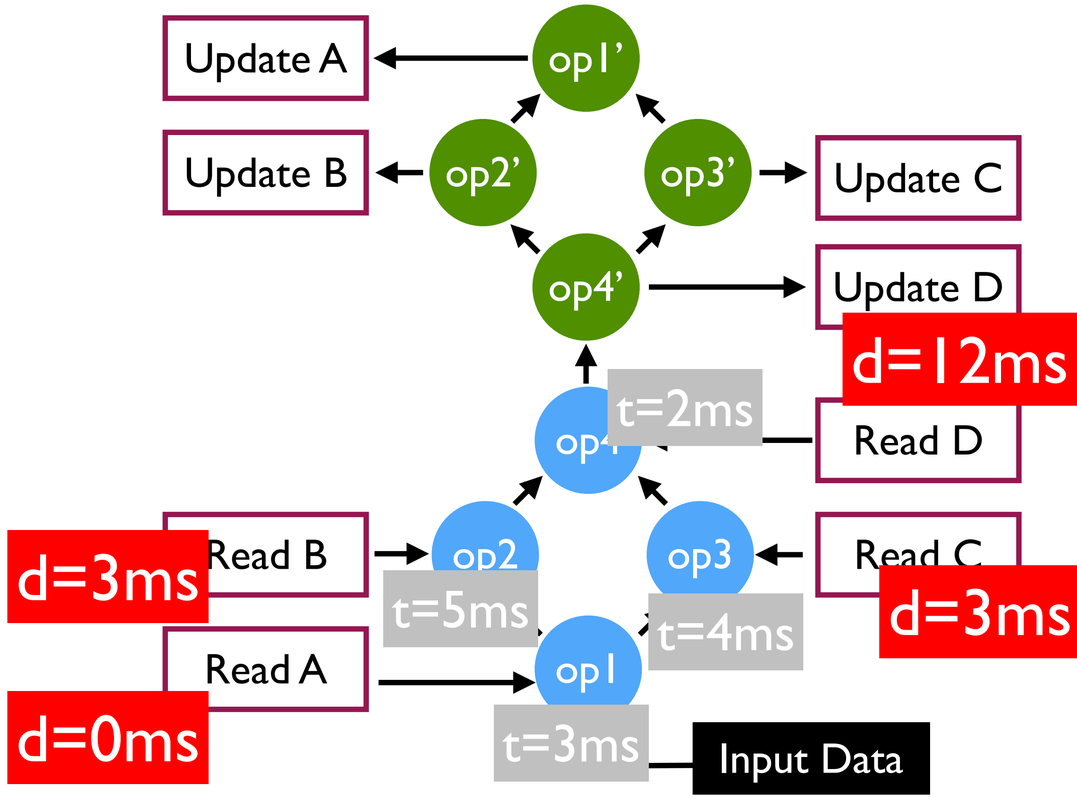


Sample TensorFlow Model: One Iteration

Distributed DNN Training CadentFlow

- Priority-based
  - Assign priorities based on DAG structure
  - Objective: Minimize completion time subject to priorities
- Deadline-based
  - Assign deadlines based on per-op computation time
  - Objective: Minimize $\max_i(\text{endTime}_i - \text{deadline}_i)$

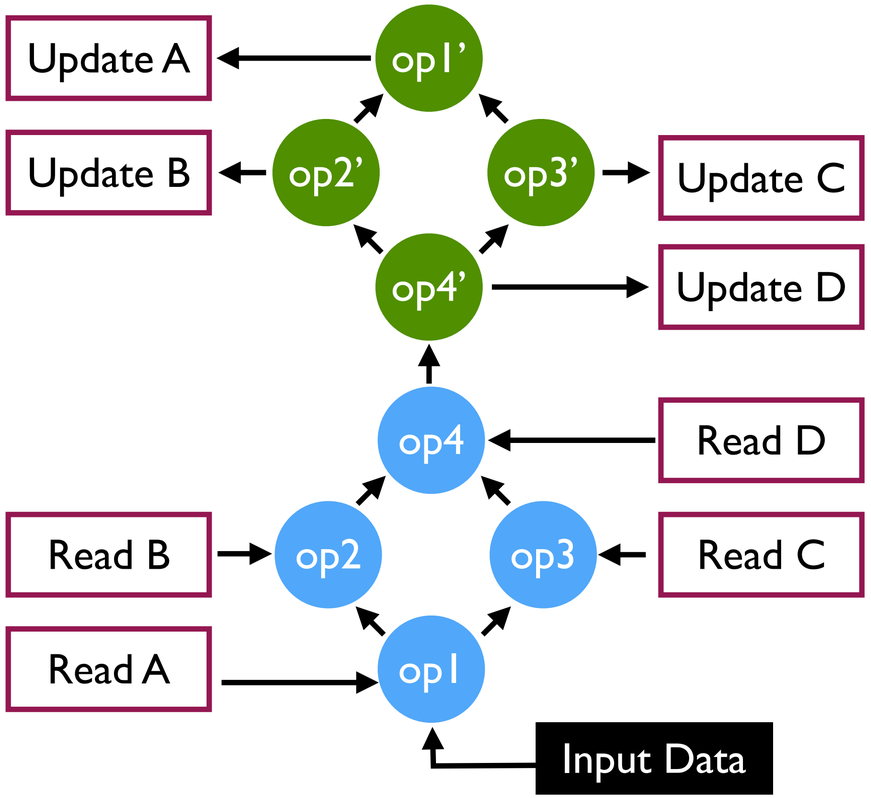
delay of flow i



Sample TensorFlow Model: One Iteration

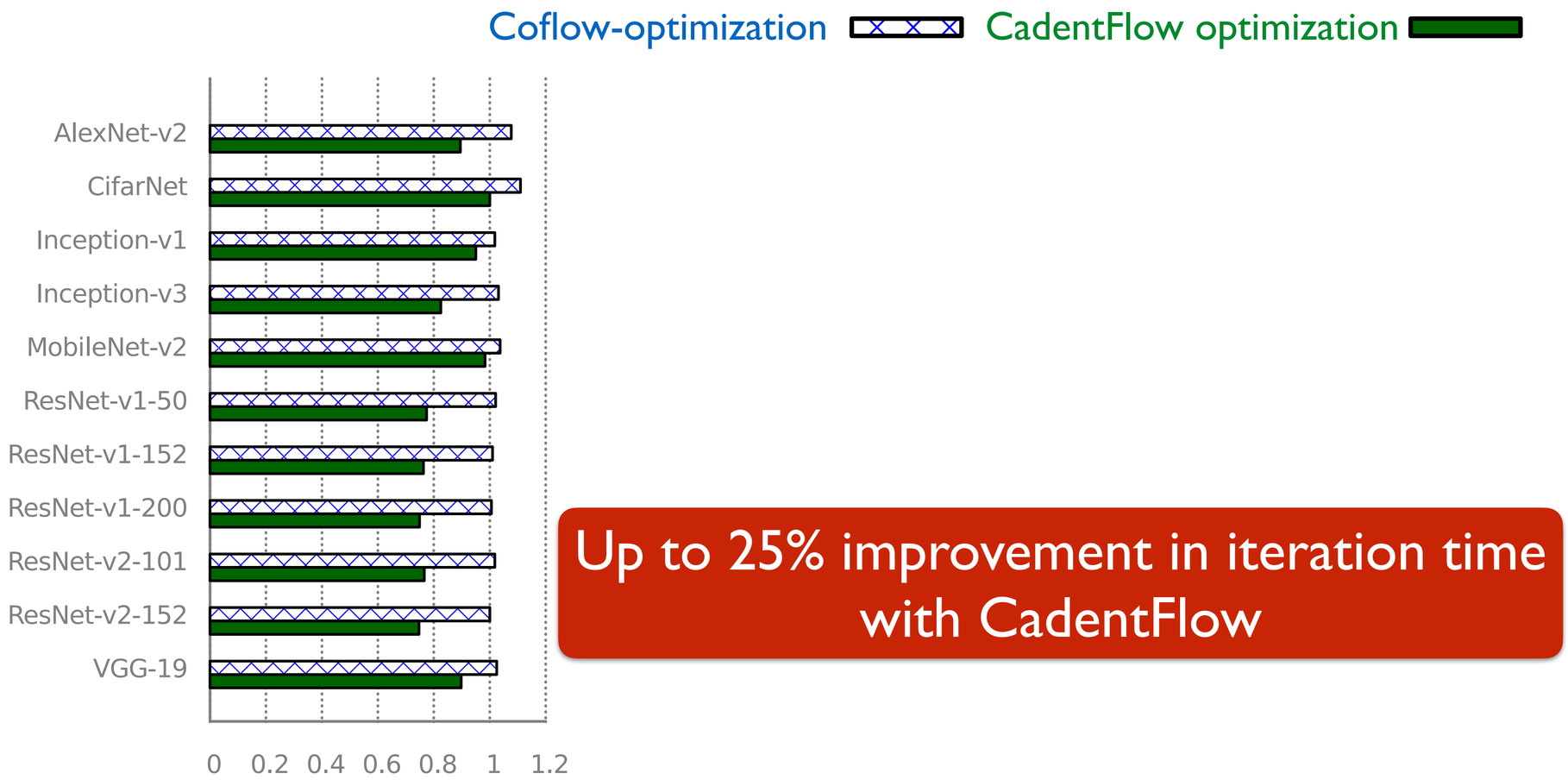
Quantifying benefits achievable with a better network abstraction

- Representative application: distributed deep learning
- Methodology
  - Tracing distributed deep learning workloads to obtain dependencies and computation/communication times
  - Simulate various network control schemes
    1. TCP (max-min fairness across flows sharing a link)
    2. Minimum Allocation for Desired Duration (MADD) [Coflow control in Varys]
    3. CadentFlow-optimized scheme



Sample TensorFlow Model: One Iteration

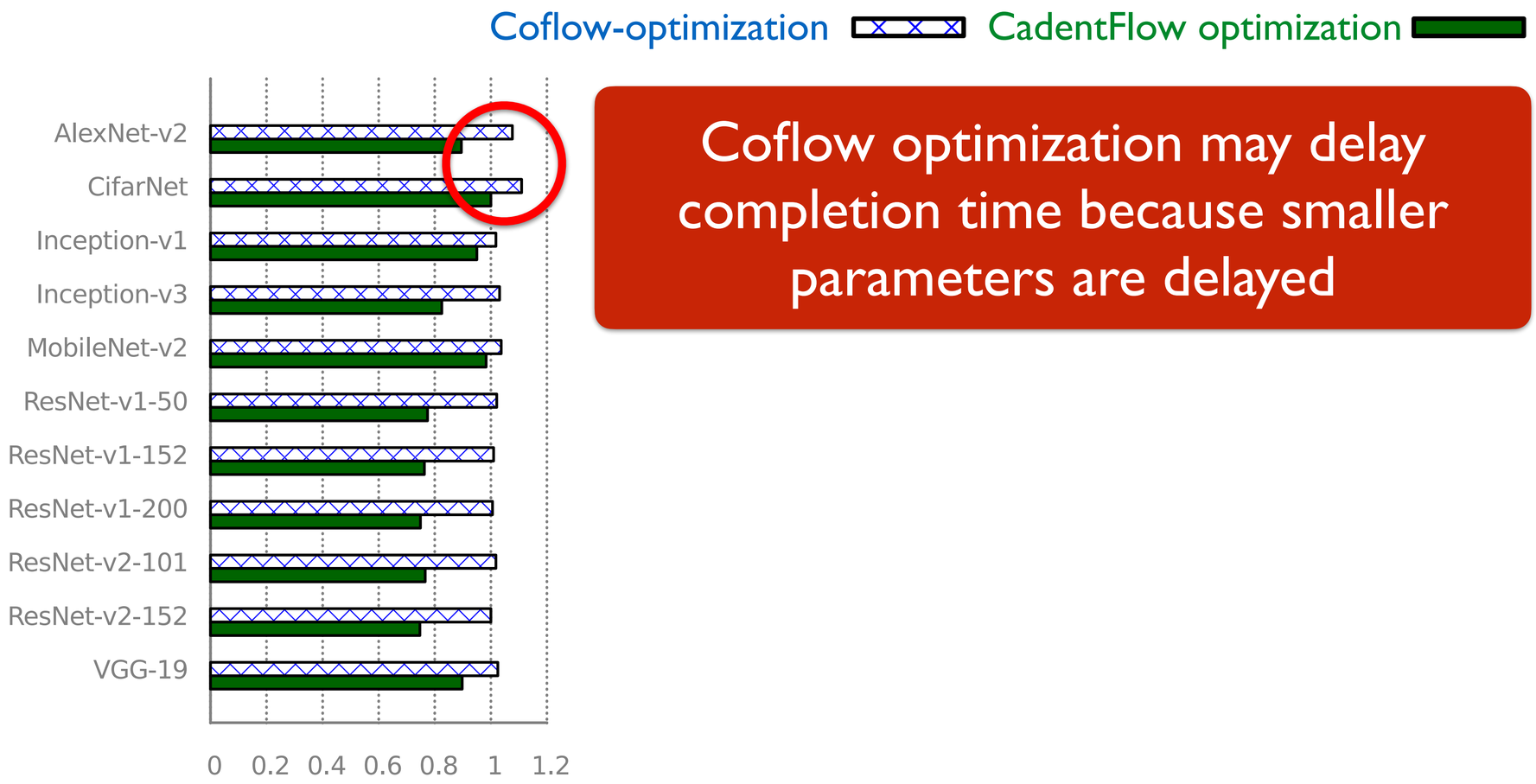
Performance Improvement



Up to 25% improvement in iteration time with CadentFlow

8 workers, 8 PS

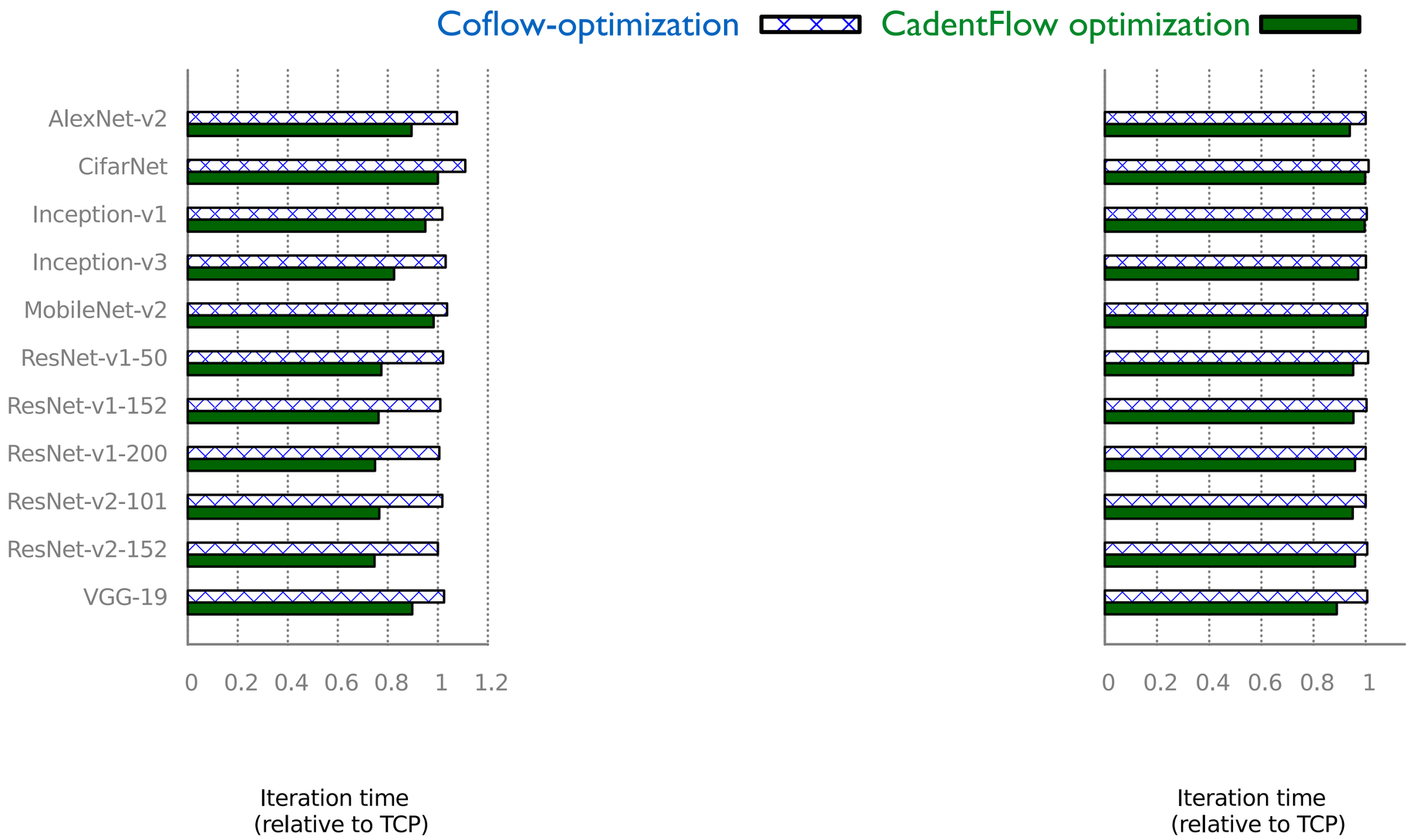
Performance Improvement



Coflow optimization may delay completion time because smaller parameters are delayed

8 workers, 8 PS

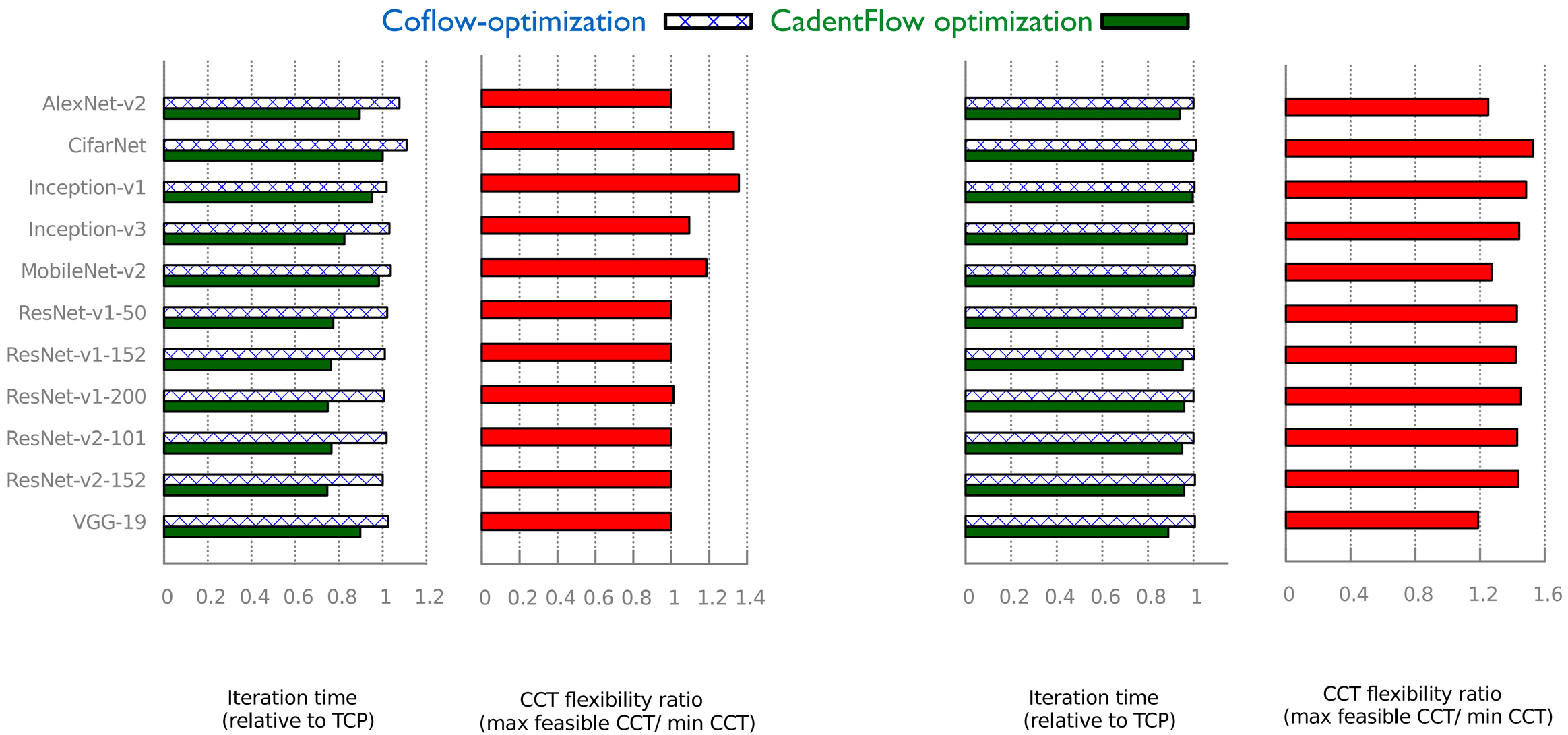
Performance Improvement



8 workers, 8 PS

16 workers, 16 PS

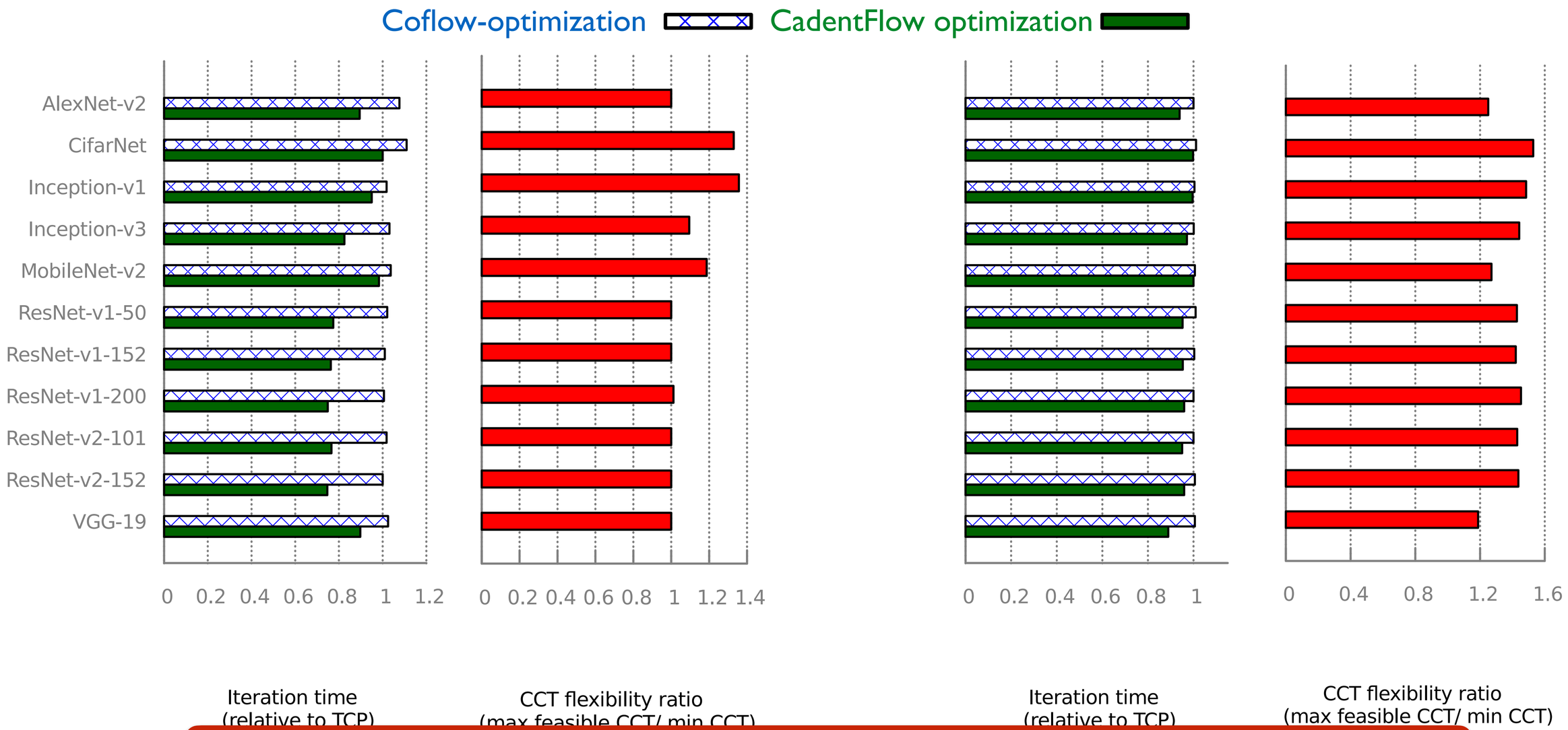
Performance Improvement



8 workers, 8 PS

16 workers, 16 PS

Performance Improvement



When gain in iteration time is lower, CCT flexibility ratio is higher

Open Challenges

- Extracting the application objective
- Designing network controllers that can handle multiple application objectives
    - How to handle conflicting objectives?
- Implementation challenges
    - Real-time decision making
    - End host vs. in-network implementation

Thank You

Email: sangeetha.aj@uci.edu