RICE

Big Data and Optical Lightpaths Driven Lab

Exploiting Inter-Flow Relationship for Coflow Placement in Data Centers

Xin Sunny Huang, T. S. Eugene Ng
Rice University

This Work

- **Optimizing Coflow performance** has many benefits such as avoiding application straggles[1,2] and improving resource utilization[3,4].

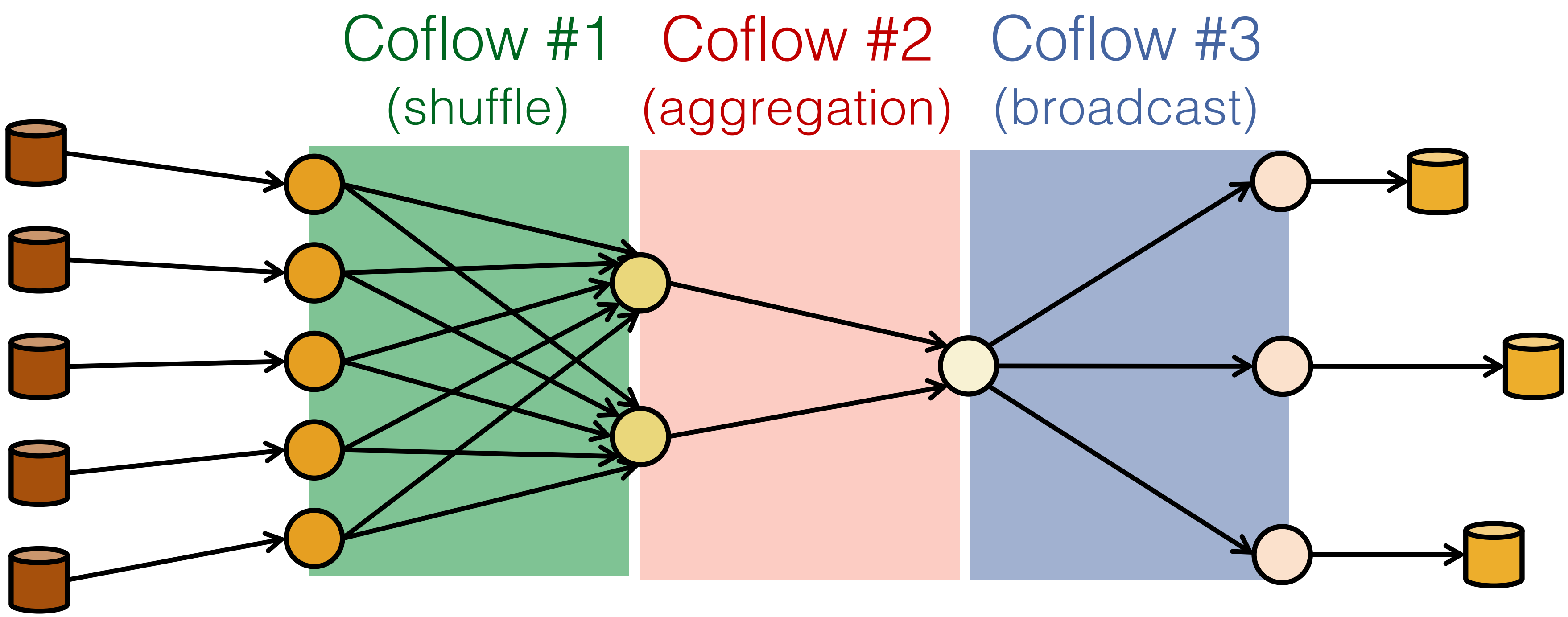
- **Coflow placement** is an unexplored, important factor to determine Coflow performance.

- **2D-Placement** leverages inter-flow relationship to find good placement for Coflows.

[1] **Orchestra** (SIGCOMM ’11). [2] **Varys** (SIGCOMM ’14).
[3] **CARBYNE** (OSDI ‘16). [4] **YARN-ME** (memory elasticity, in ATC ’17)

Coflow

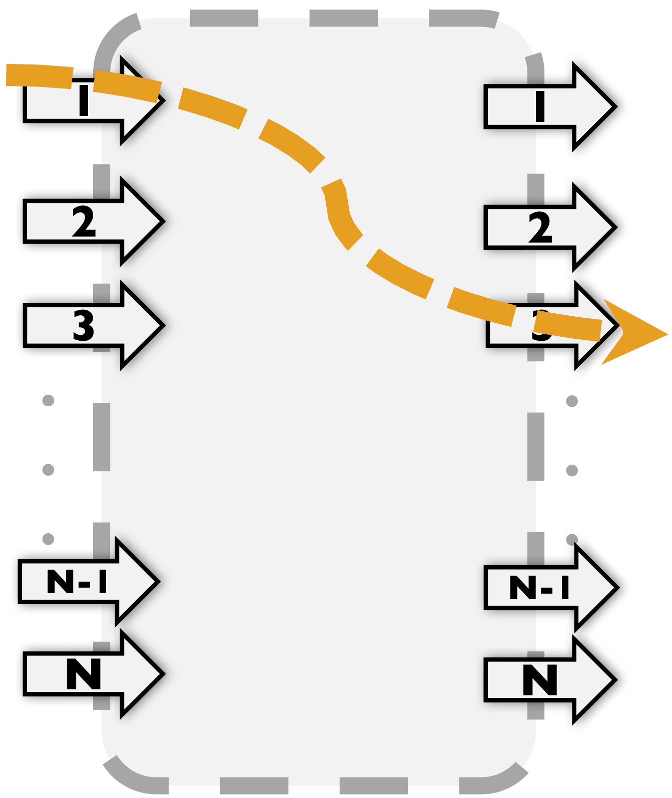
- Coflow [1] : A set of parallel flows.
- Produced by distributed applications (e.g. Hadoop & Spark).
- Performance is measured by Coflow Completion Time (CCT), i.e. the slowest flow's completion time.

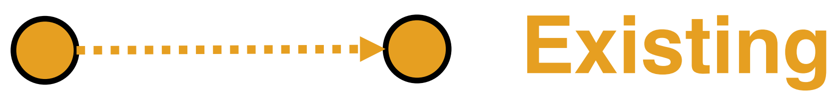


[1] Chowdhury, M. et al. Coflow: An application layer abstraction for cluster networking. (HotNets'12)

Coflow Scheduling

- Prior works demonstrate benefits of Coflow scheduling.
- **Limitation**: Assume predetermined placement for Coflows, i.e. predetermined sender/receiver locations.

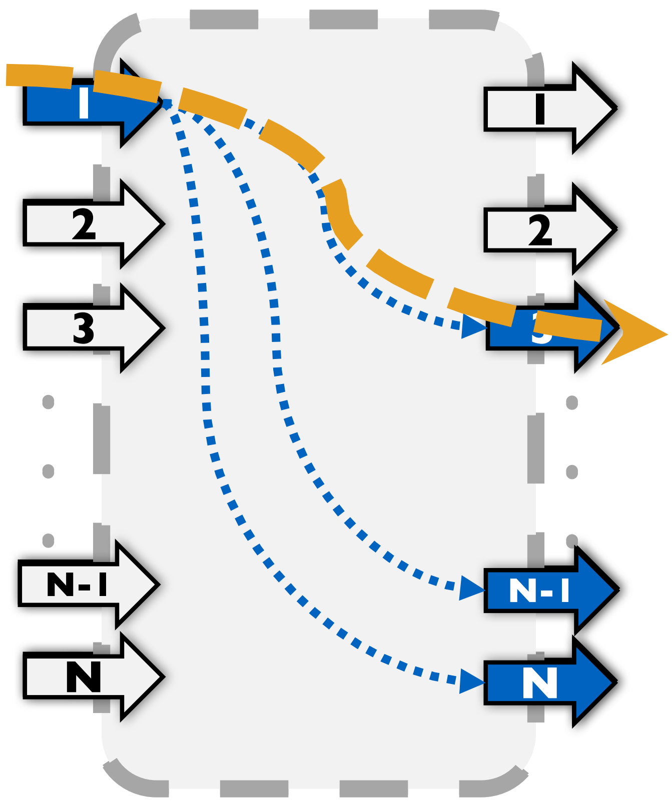


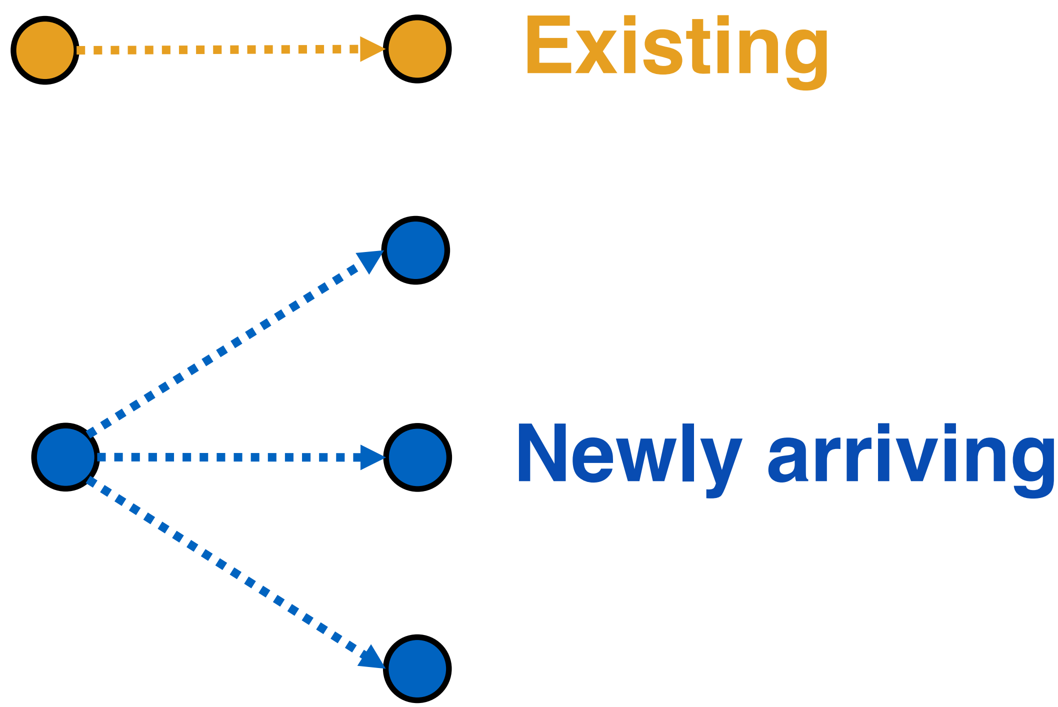


Varys (SIGCOMM '14), **Aalo** (SIGCOMM '15), **CODA** (SIGCOMM '16) and **Sunflow** (CoNEXT '16), etc.

Coflow Scheduling

- Prior works demonstrate benefits of Coflow scheduling.
- **Limitation**: Assume predetermined placement for Coflows, i.e. predetermined sender/receiver locations.

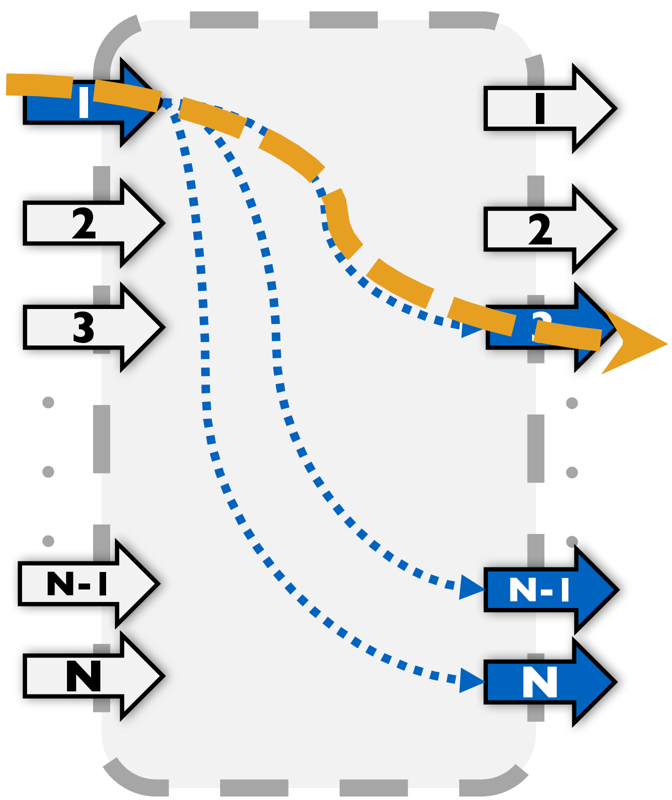




Varys (SIGCOMM '14), **Aalo** (SIGCOMM '15), **CODA** (SIGCOMM '16) and **Sunflow** (CoNEXT '16), etc.

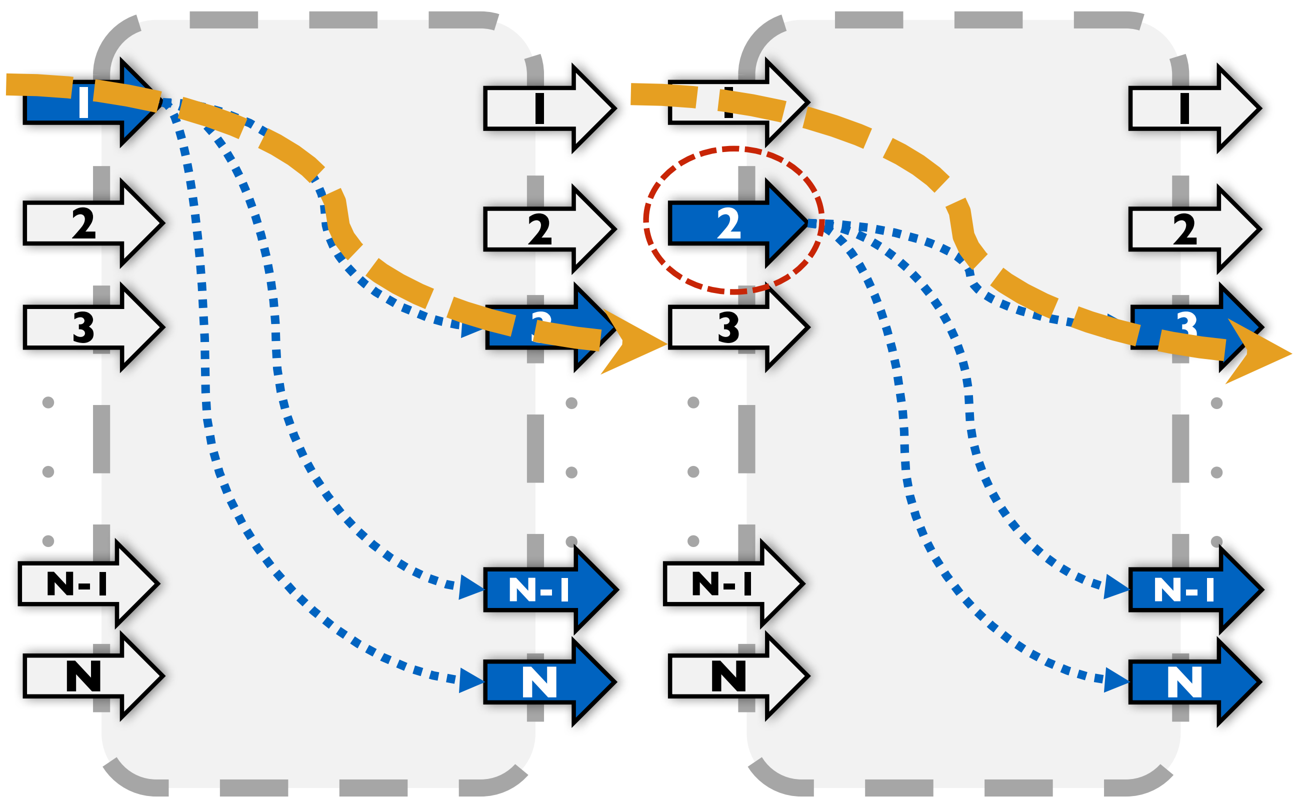
Coflow Placement

- Coflow placement can be flexible (e.g. cluster scheduler to choose machines for tasks in a stage).



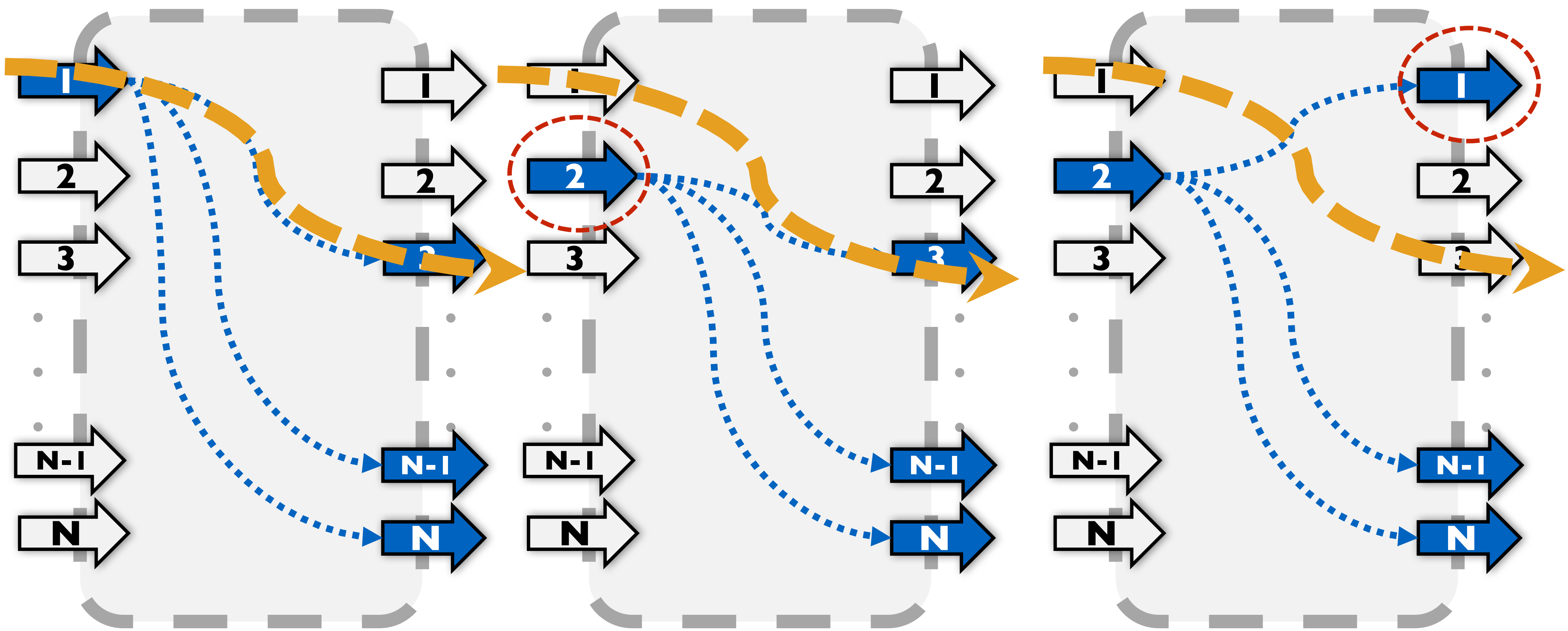
Coflow Placement

- Coflow placement can be flexible (e.g. cluster scheduler to choose machines for tasks in a stage).



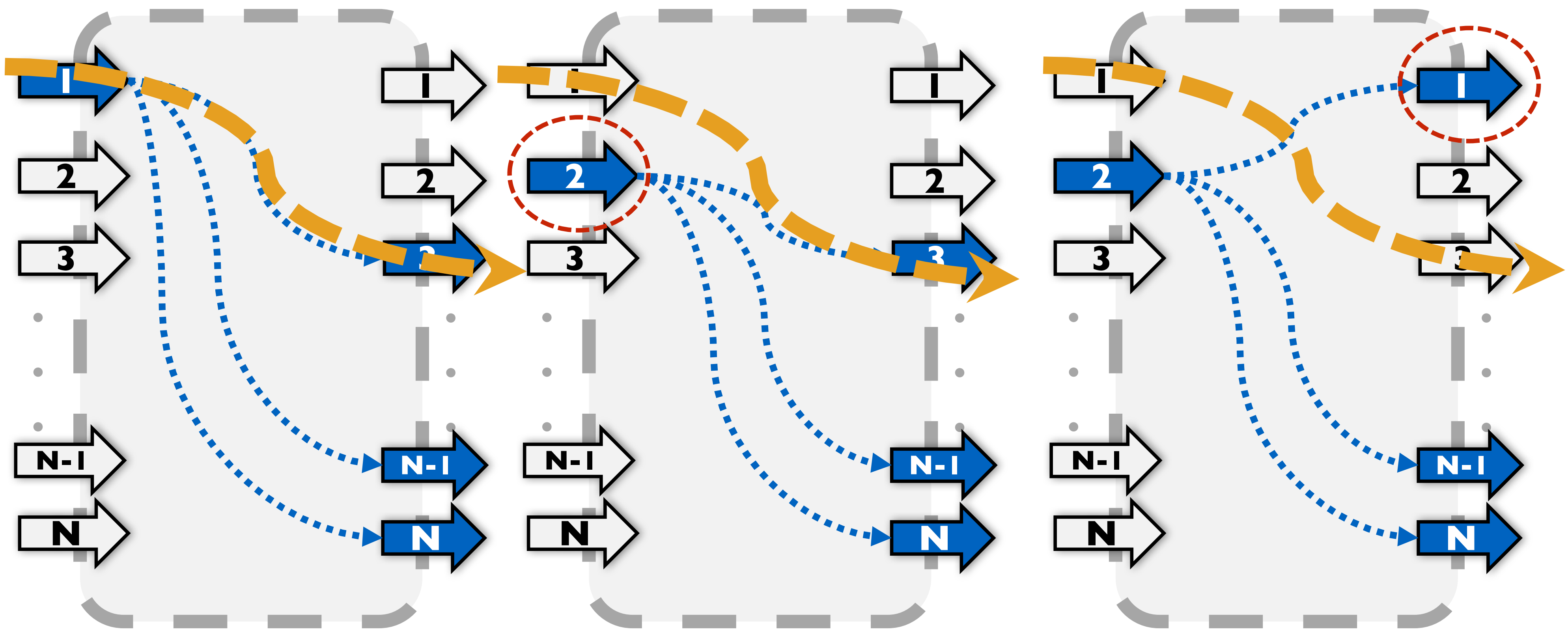
Coflow Placement

- Coflow placement can be flexible (e.g. cluster scheduler to choose machines for tasks in a stage).



Coflow Placement

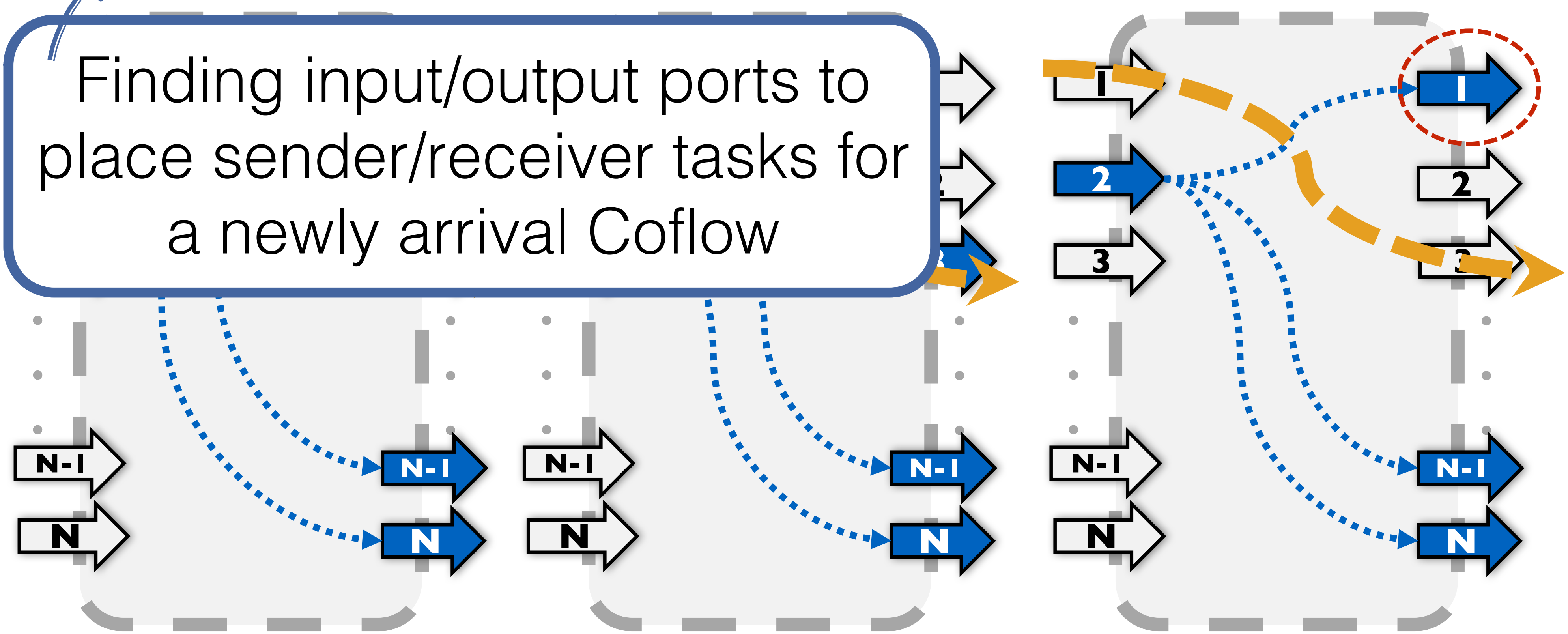
- Coflow placement can be flexible (e.g. cluster scheduler to choose machines for tasks in a stage).
- Placement and scheduling decide Coflow performance.



Coflow Placement

- Coflow placement can be flexible (e.g. cluster scheduler to choose machines for tasks in a stage).
- Placement and scheduling decide Coflow performance.

Finding input/output ports to place sender/receiver tasks for a newly arrival Coflow

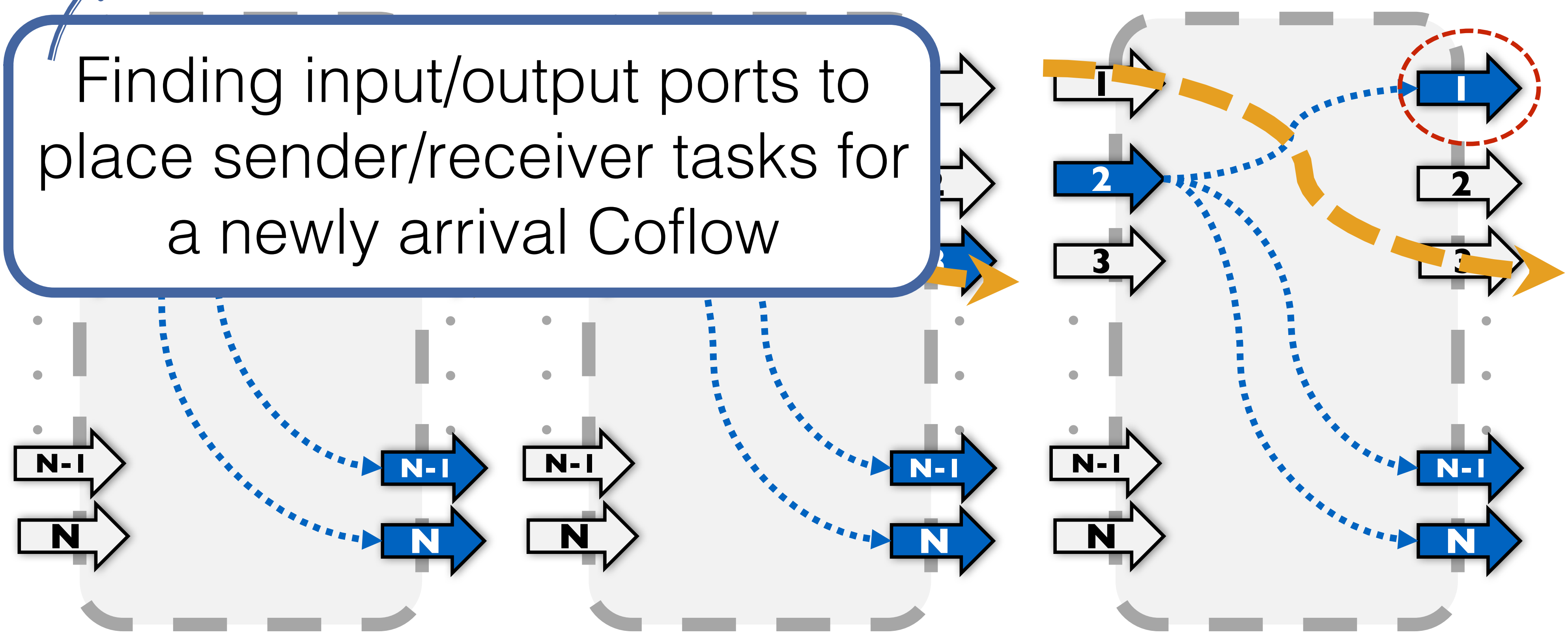


Coflow Placement

This work: good placement under optimal scheduling

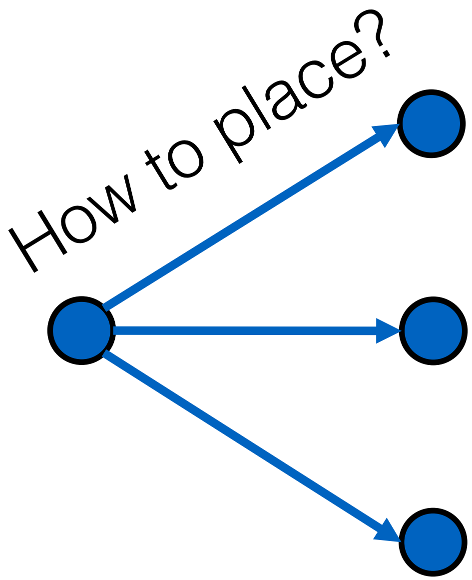
- Coflow place
  to choose ma… for tasks in a stage).
- Placement and scheduling decide Coflow performance.

Finding input/output ports to place sender/receiver tasks for a newly arrival Coflow



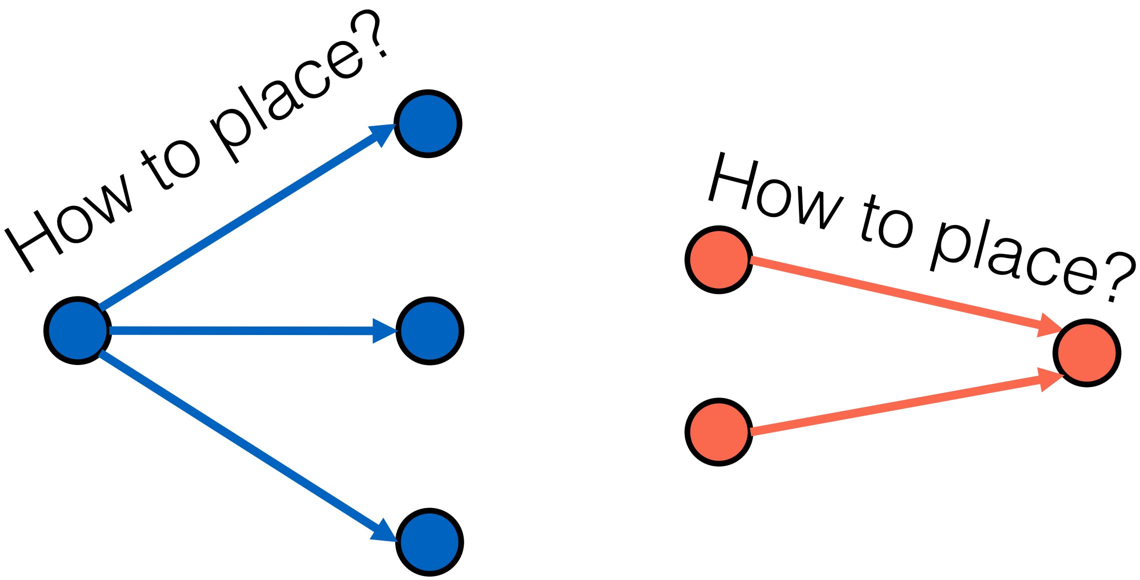
Coflow Placement Constrained by Inter-Flow Relationship

- Within a Coflow, flows' placement are dependent.



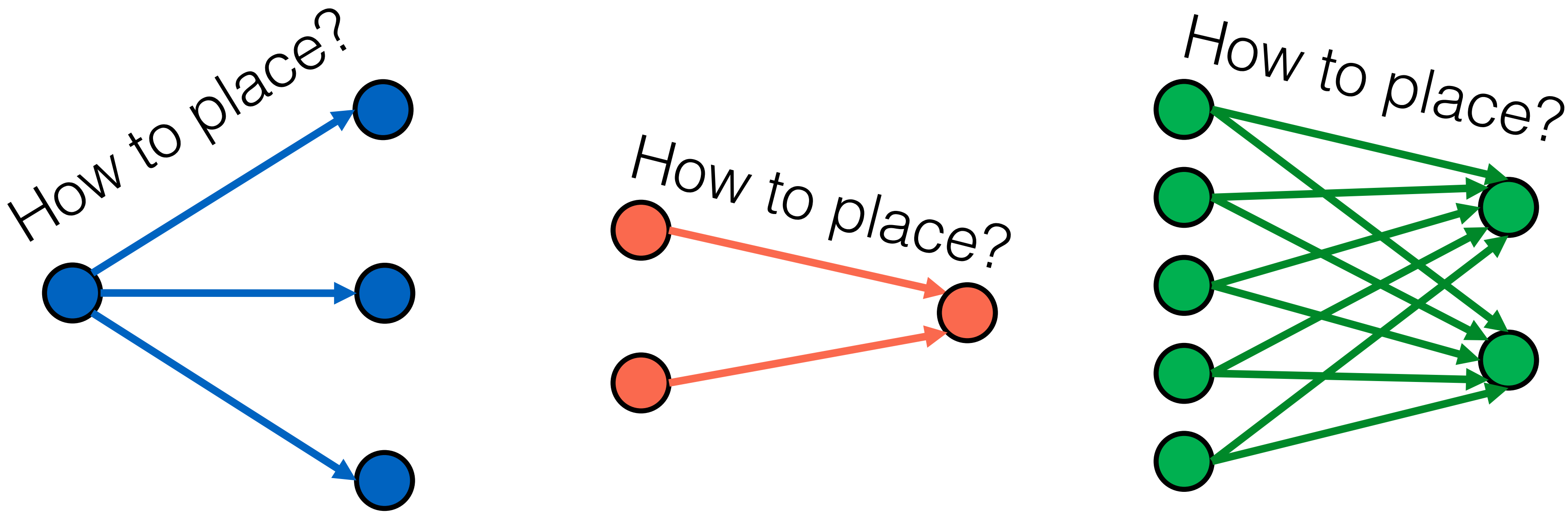
Coflow Placement Constrained by Inter-Flow Relationship

- Within a Coflow, flows' placement are dependent.



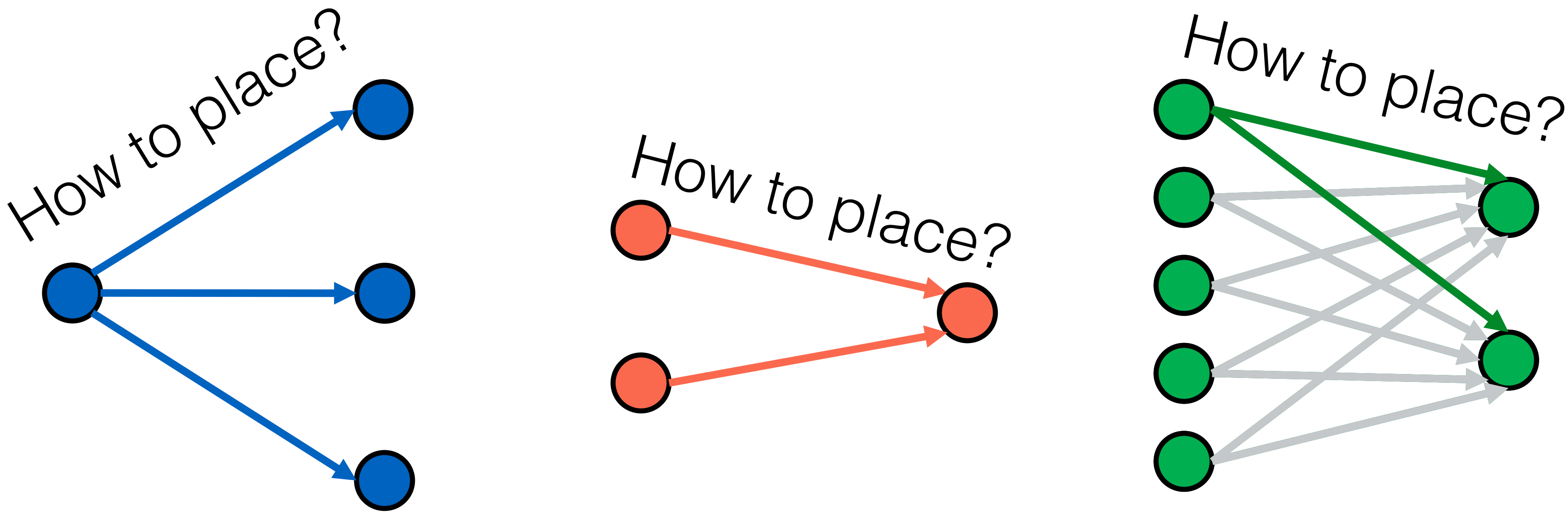
Coflow Placement Constrained by Inter-Flow Relationship

- Within a Coflow, flows' placement are dependent.



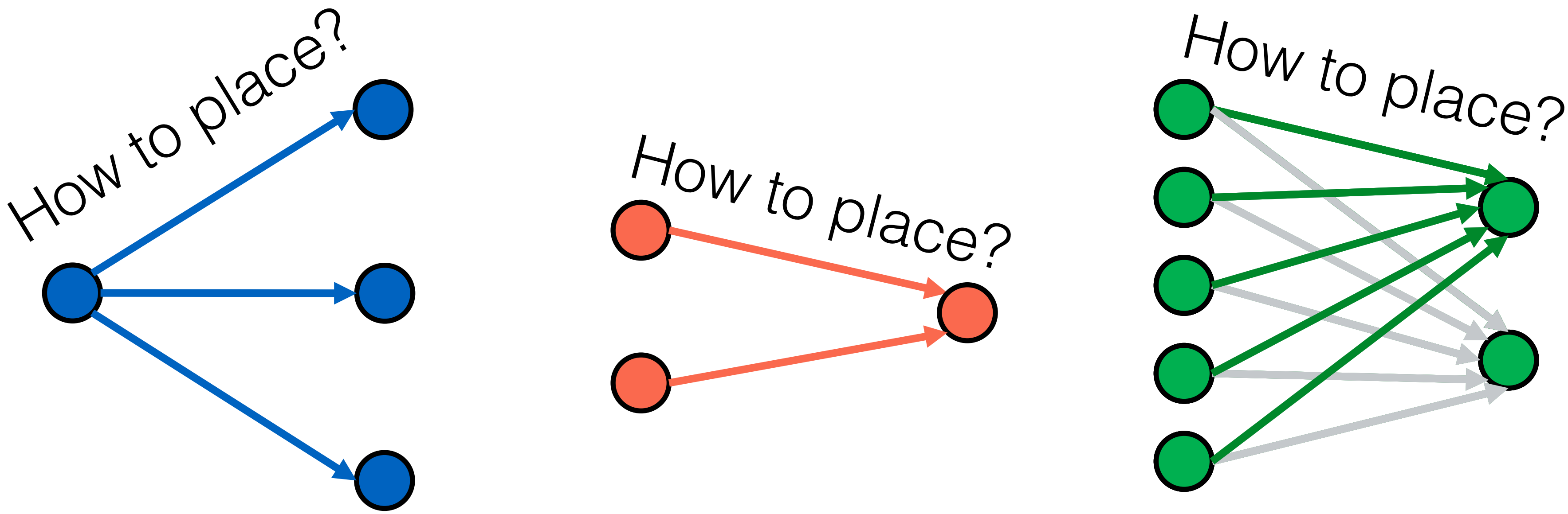
Coflow Placement Constrained by Inter-Flow Relationship

- Within a Coflow, flows' placement are dependent.



Coflow Placement Constrained by Inter-Flow Relationship

- Within a Coflow, flows' placement are dependent.



Challenge #1: Intra-Coflow Bottleneck Delay

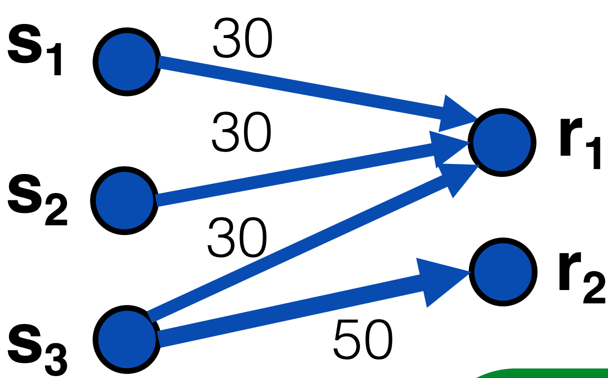
C_2

Network with C_1

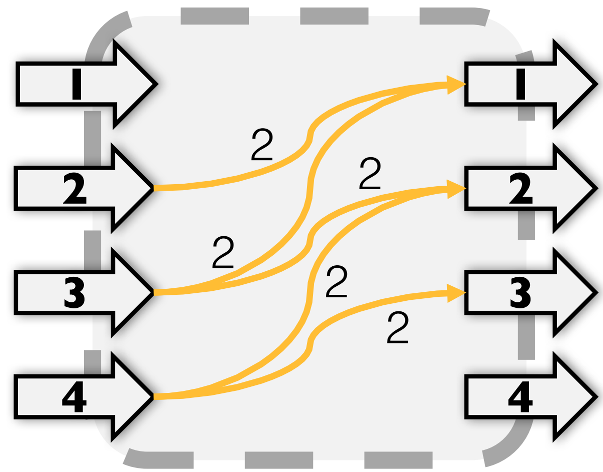
	r_1	r_2
s_1	**30**	
s_2	**30**	
s_3	**30**	**50**

	out.1	2	3	4
in.1				
2	**2**			
3	**2**	**2**		
4		**2**	**2**	

Challenge #1: Intra-Coflow Bottleneck Delay



How to place?



C_2

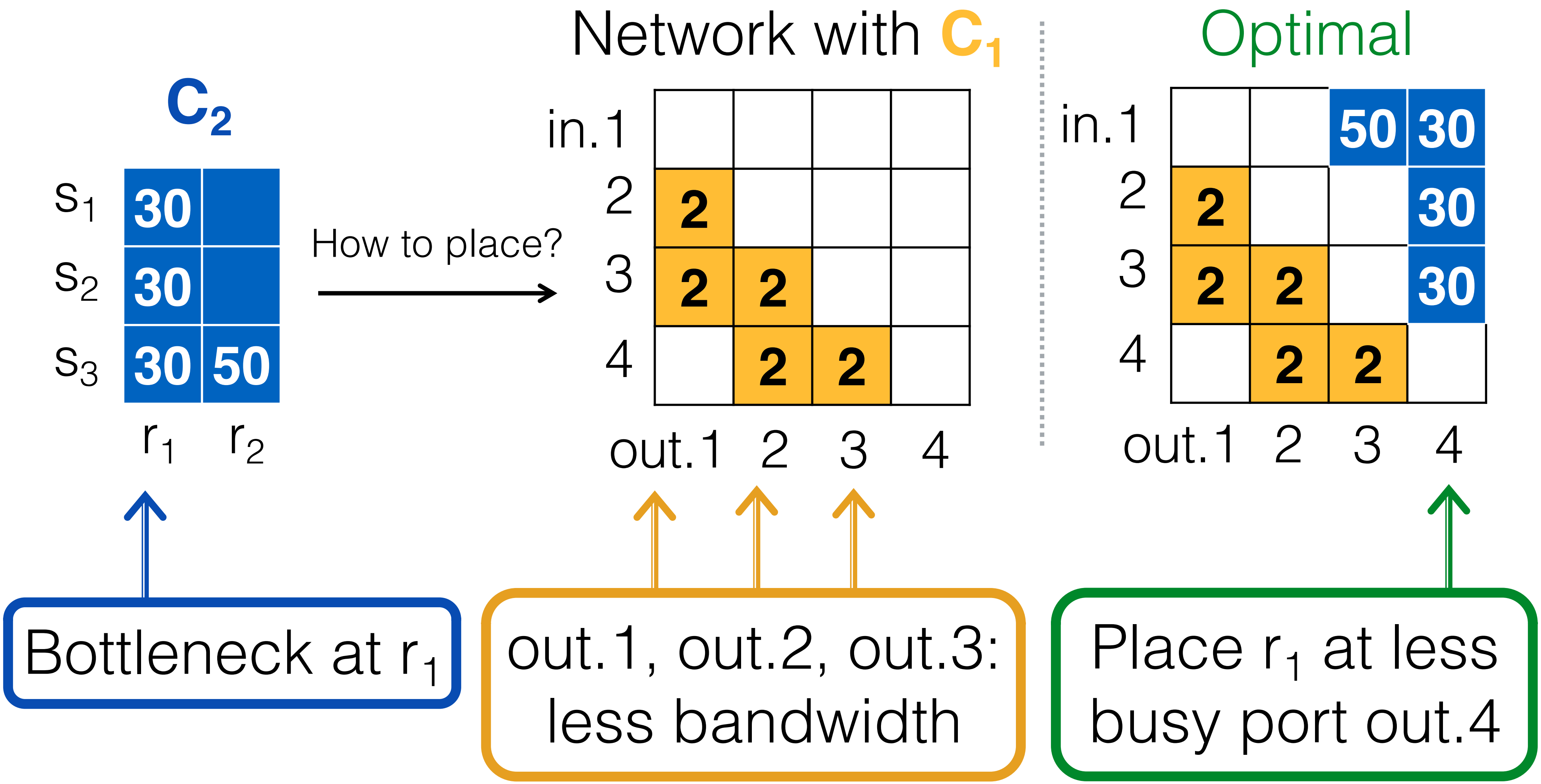
	r_1	r_2
s_1	30	
s_2	30	
s_3	30	50

Only consider C_2 : C_1 is prioritized under optimal scheduling, and thus C_1 is not sensitive to C_2.

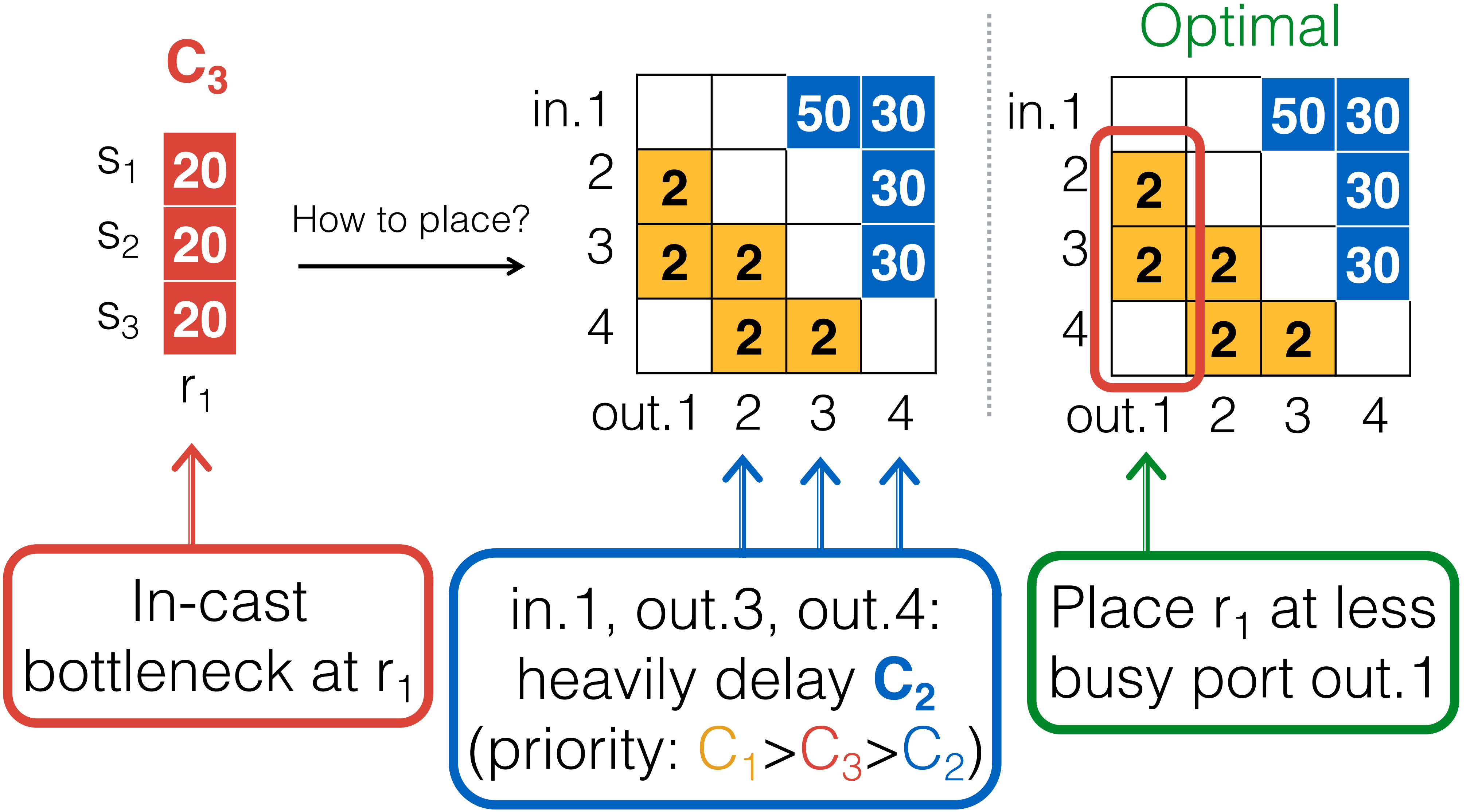
Network with C_1

in. \ out.	1	2	3	4
1				
2	2			
3	2	2		
4		2	2	

Challenge #1: Intra-Coflow Bottleneck Delay



Challenge #2: Inter-Coflow Bottleneck Contentions



Summary: Keys to Coflow Placement

Intra-Coflow	Inter-Coflow
Avoid delaying critical endpoints (bottleneck)	Avoid contentions among critical endpoints.

2D-Placement

Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	
Identify critical endpoints that require better placement.	

2D-Placement

Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	Step 2: Calculate load on ports

Identify critical endpoints that require better placement.

Find ports with less contentions.

2D-Placement

Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	Step 2: Calculate load on ports

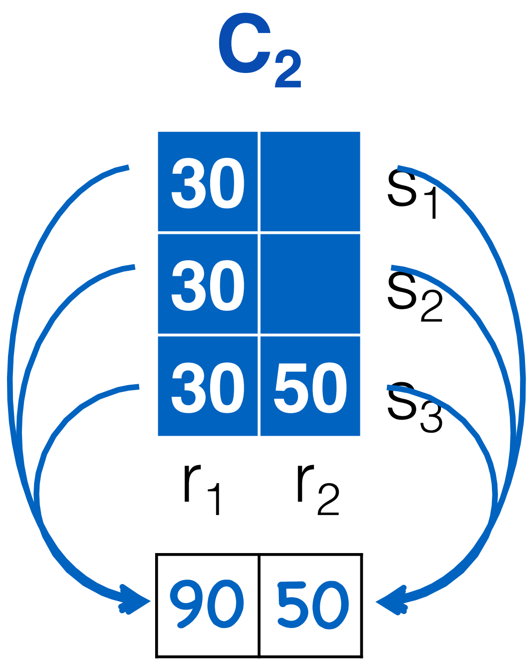
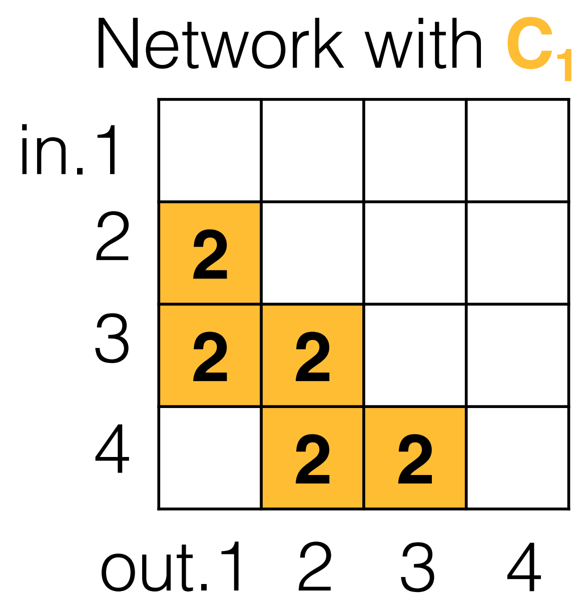
Identify critical endpoints that require better placement.

Find ports with less contentions.

Avoid contentions on critical endpoints.

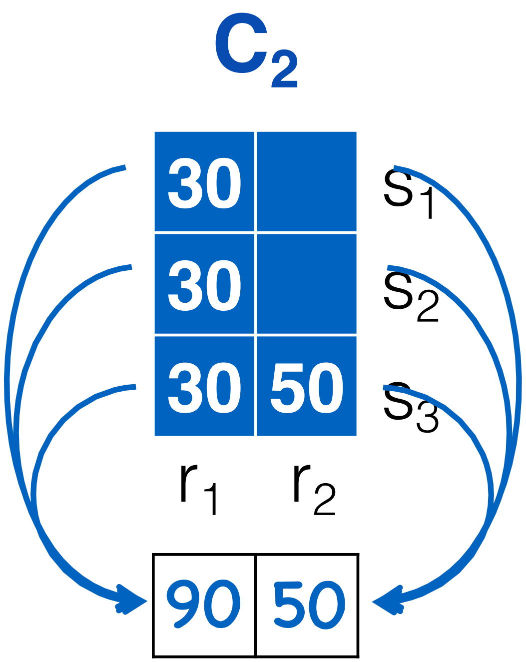
Step 3: Place heavily loaded endpoints on less loaded ports!

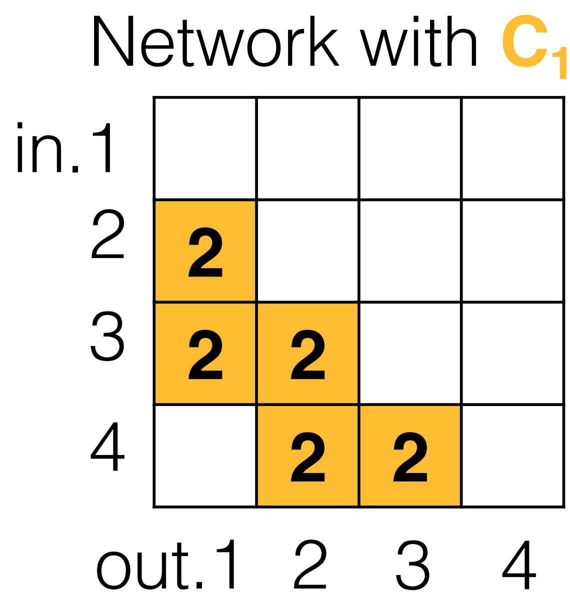
2D-Placement

Intra-Coflow	Inter-Coflow
	

2D-Placement

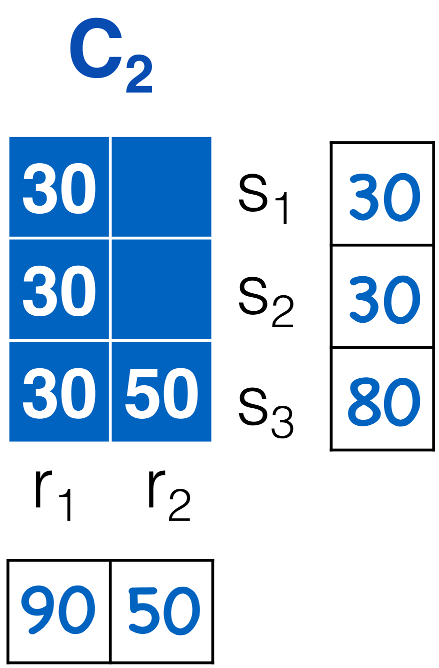
Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	

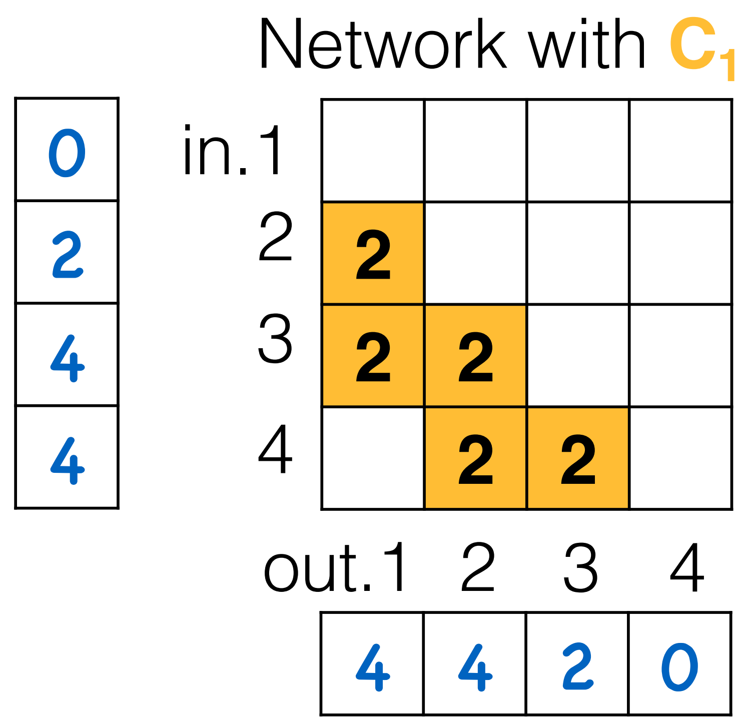




2D-Placement

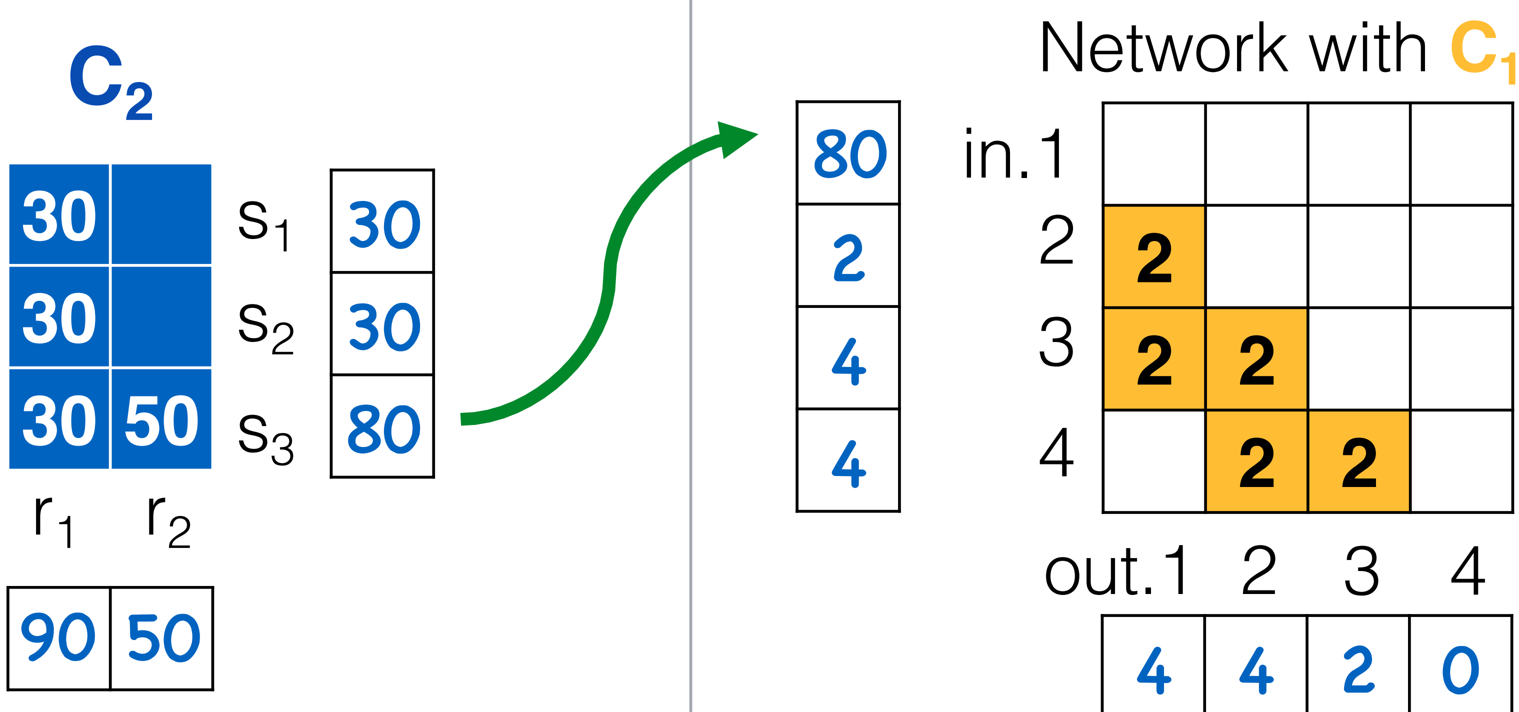
Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	Step 2: Calculate load on ports





2D-Placement

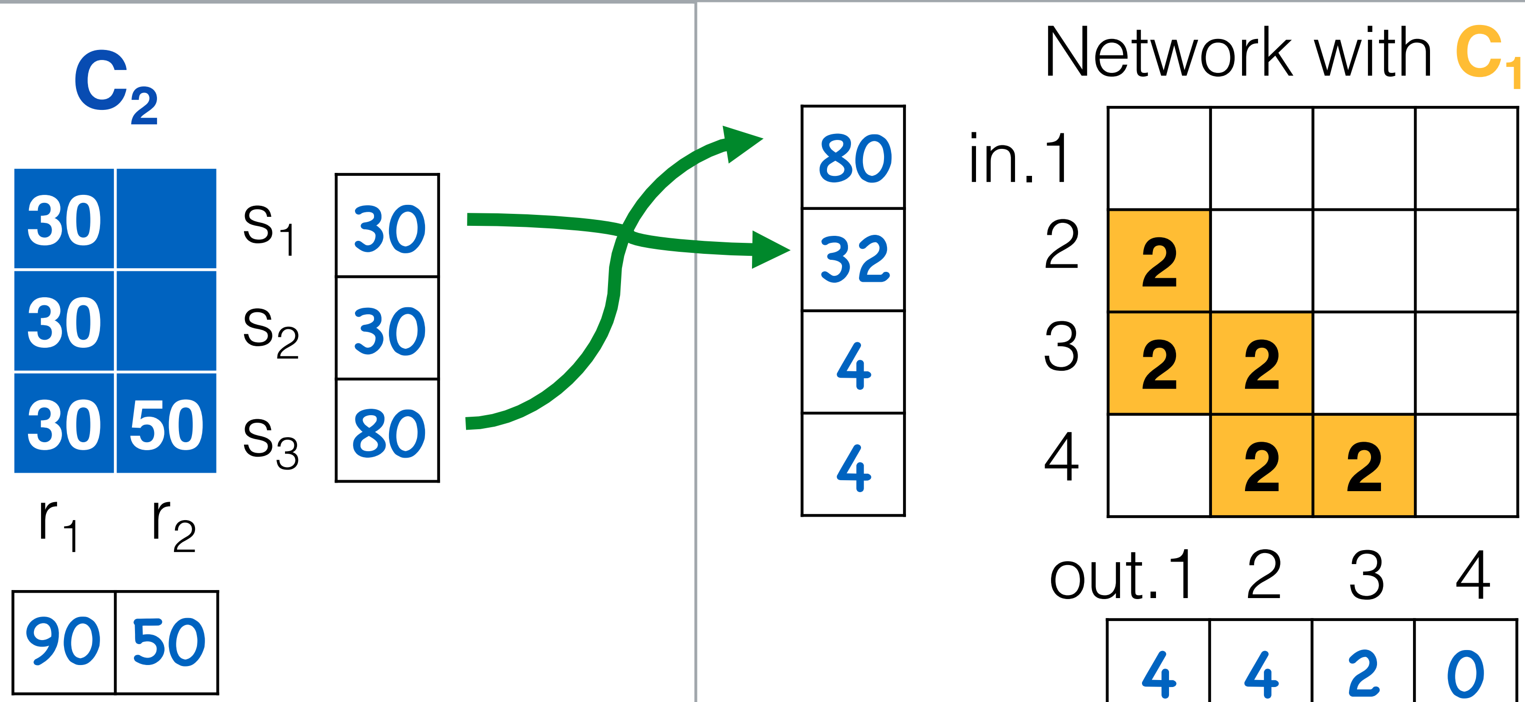
Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	Step 2: Calculate load on ports



Step 3: Place heavily loaded endpoints
on less loaded ports!

2D-Placement

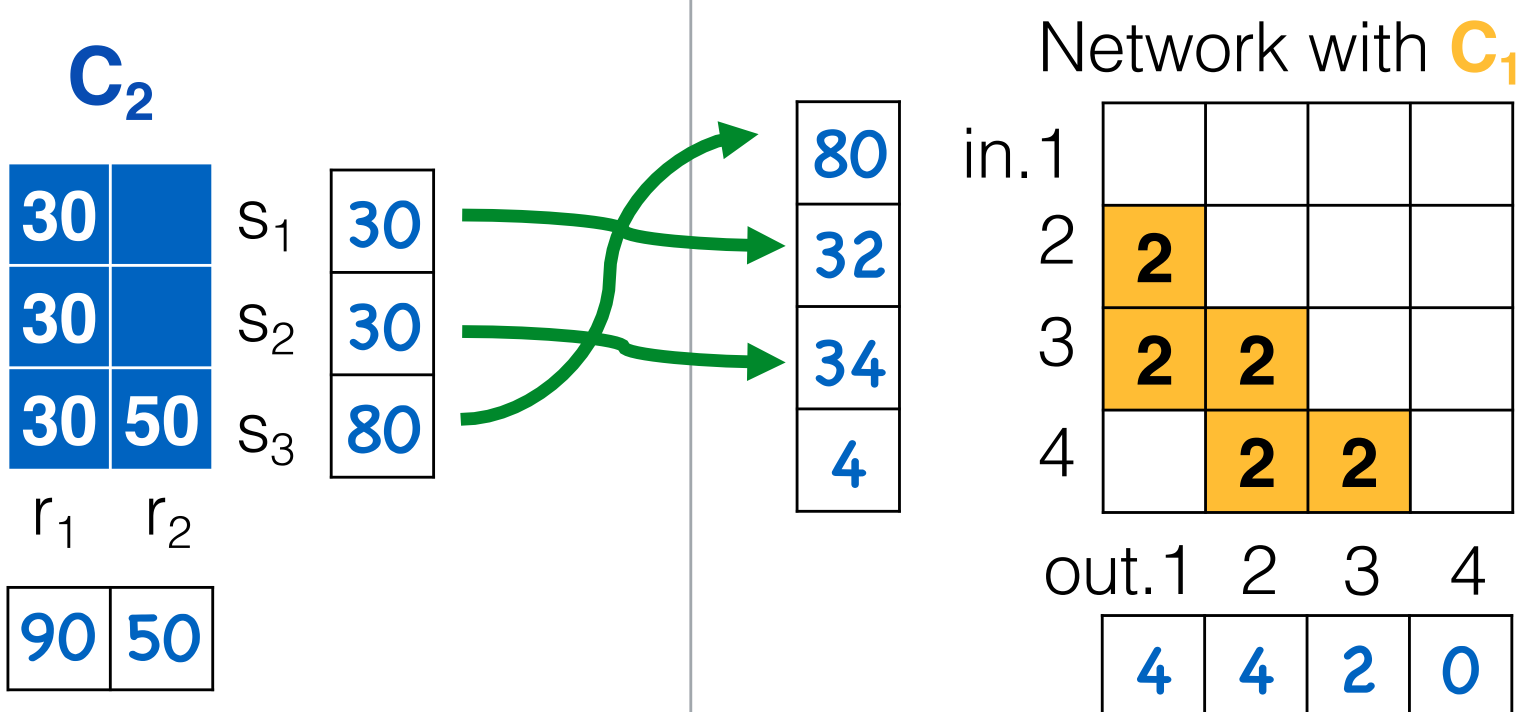
Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	Step 2: Calculate load on ports



Step 3: Place heavily loaded endpoints
on less loaded ports!

2D-Placement

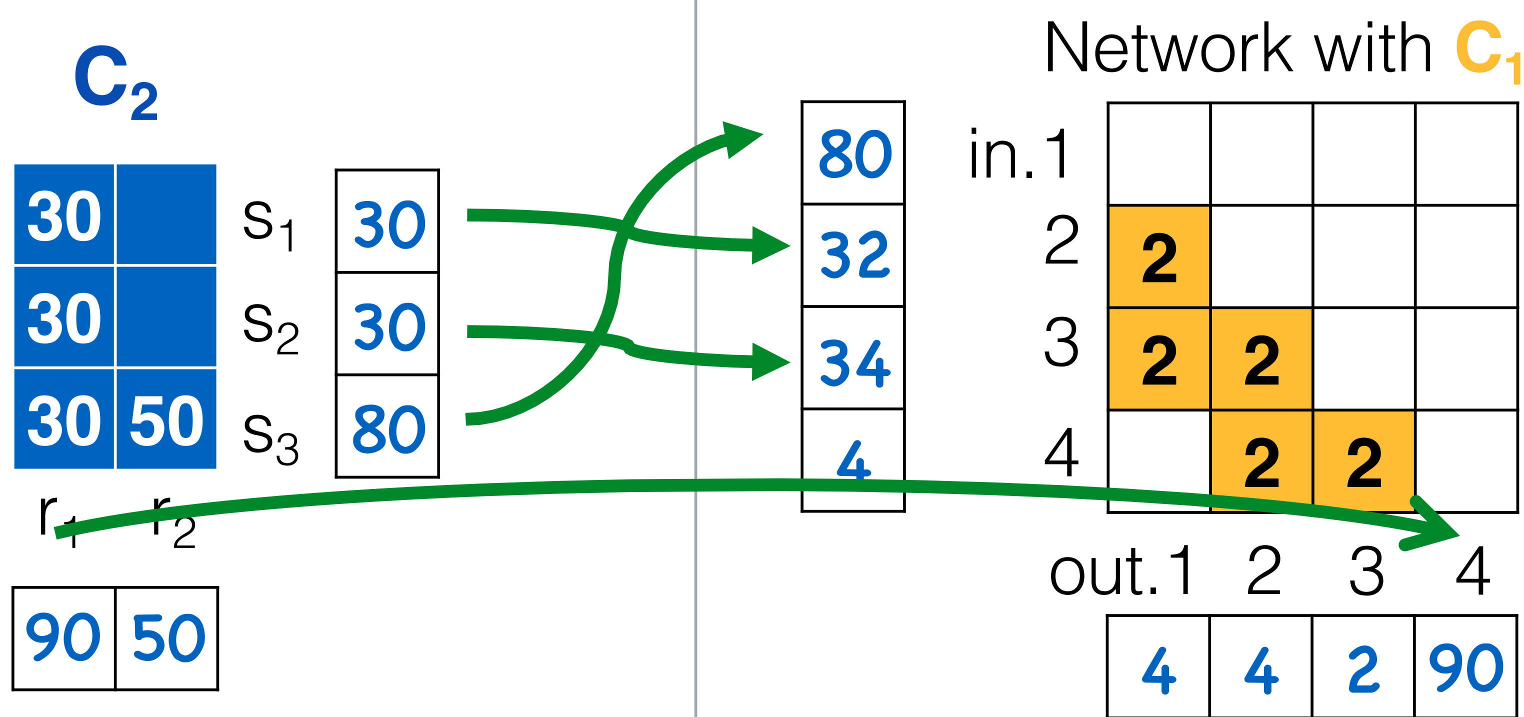
Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	Step 2: Calculate load on ports



Step 3: Place heavily loaded endpoints
on less loaded ports!

2D-Placement

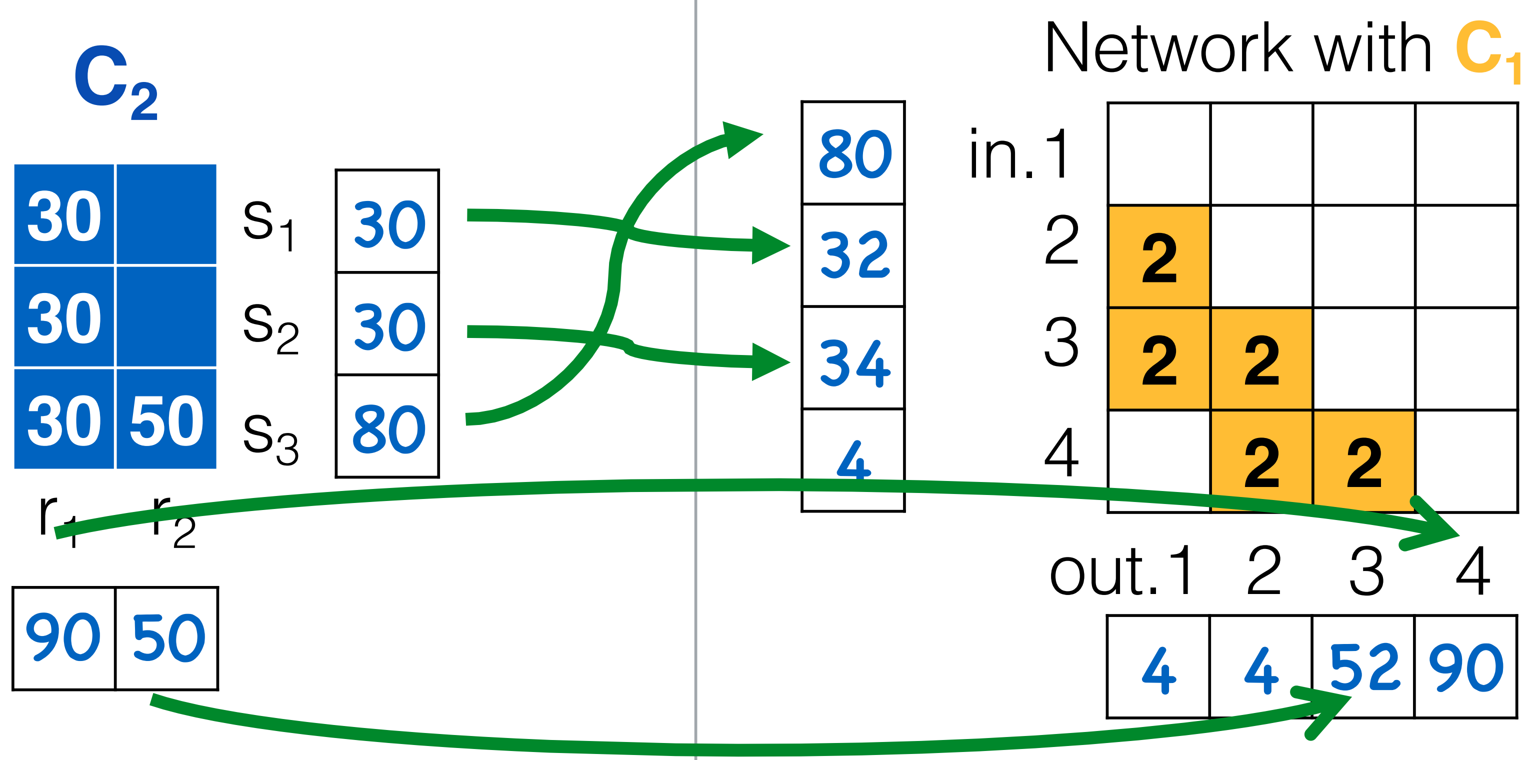
Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	Step 2: Calculate load on ports



Step 3: Place heavily loaded endpoints
on less loaded ports!

2D-Placement

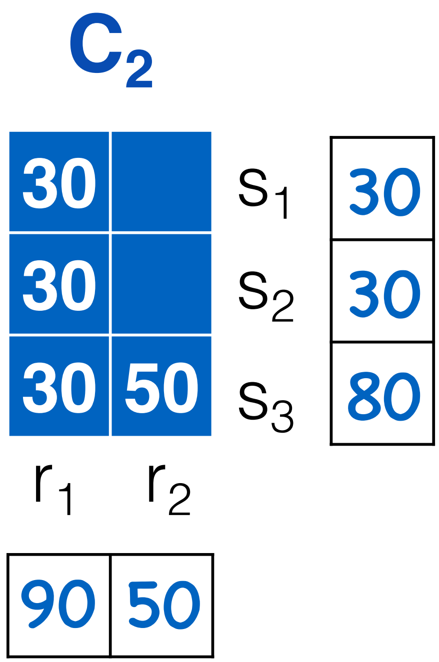
Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	Step 2: Calculate load on ports

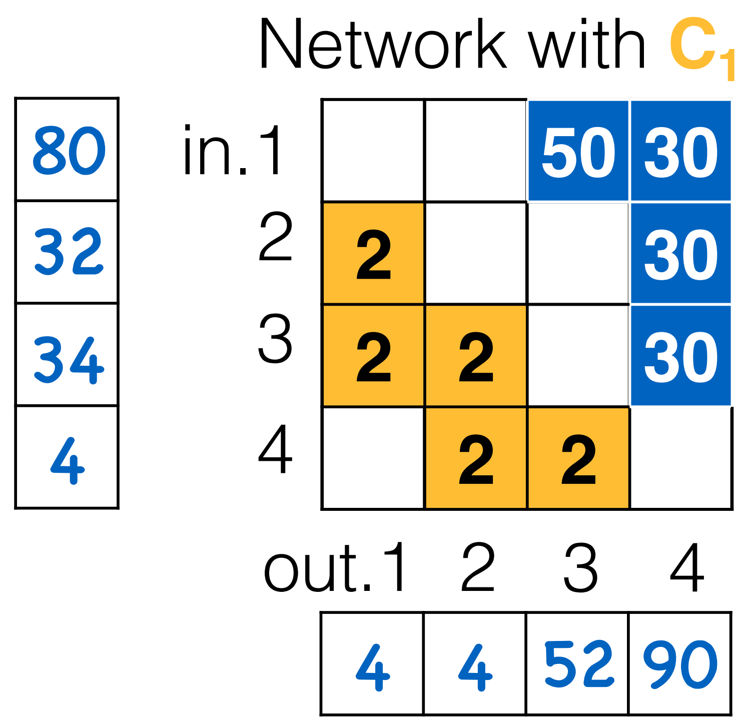


Step 3: Place heavily loaded endpoints on less loaded ports!

2D-Placement

Intra-Coflow	Inter-Coflow
Step 1: Calculate endpoint demand	Step 2: Calculate load on ports



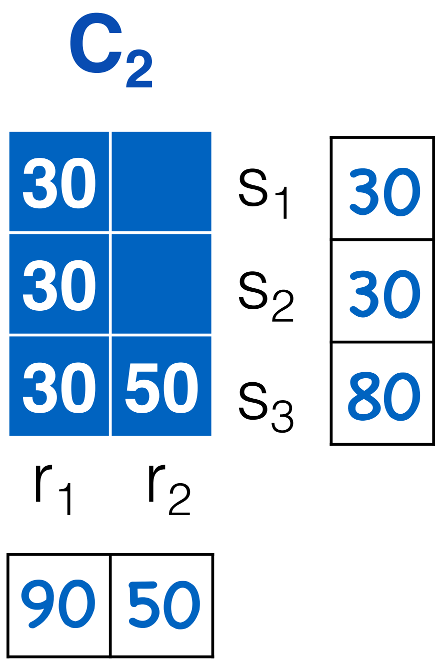


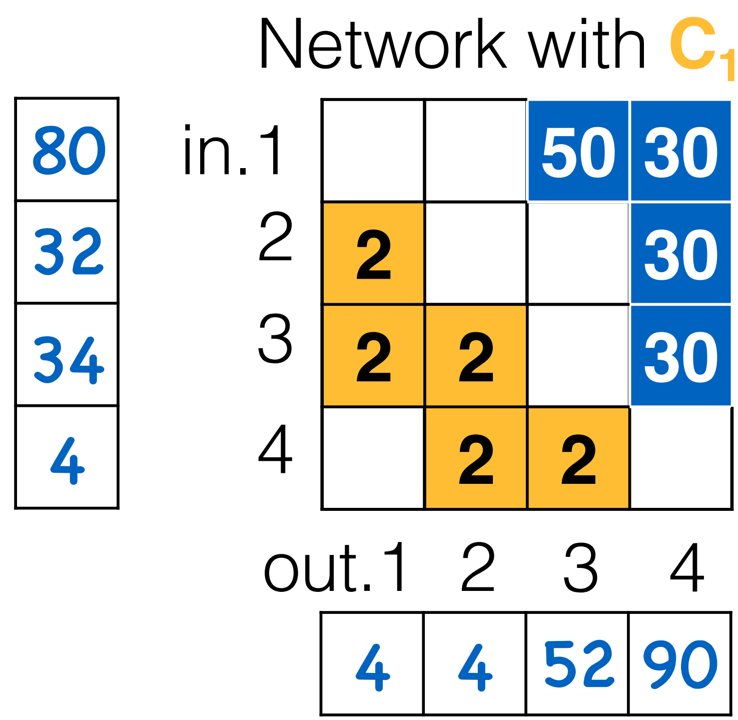
Step 3: Place heavily loaded endpoints
on less loaded ports!

2D-Placement

Greedy heuristic

Intra-Coflo	-Coflow
Step 1: Calculate endp	late load on ports





Step 3: Place heavily loaded endpoints
on less loaded ports!

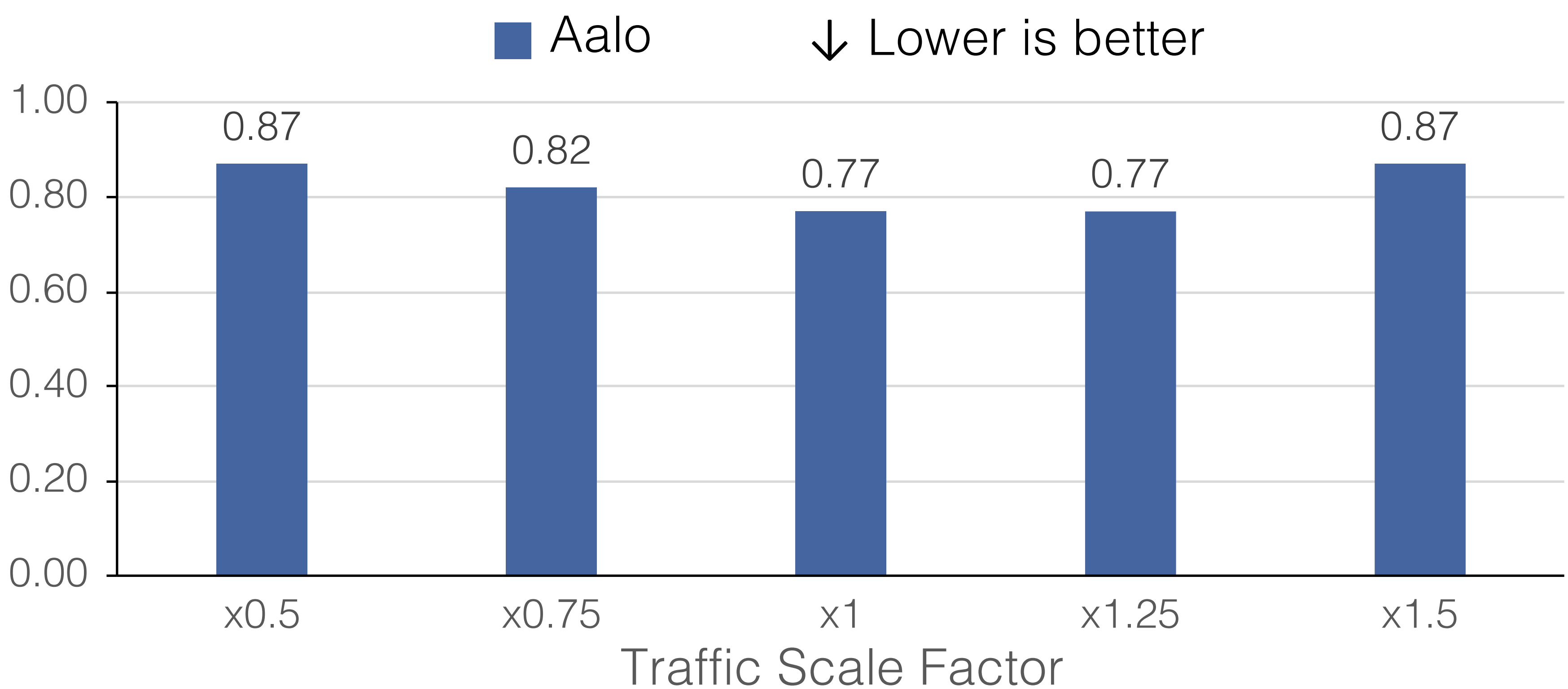
Simulation setup

- Implemented a flow-level, discrete-event simulator
- Workload[1] : realistic trace derived from Facebook cluster
  - 1hr traffic trace, > 500 Coflows, > 700,000 flows
- Baseline: flow-by-flow placement for Coflows (Neat [3])
- Coflow schedulers: Aalo [2] (this talk) and Varys [1] (paper), both designed to minimize average CCT by prioritizing small Coflows to avoid HOL blocking.

[1] **Varys** (SIGCOMM '14). [2] **Aalo** (SIGCOMM '15). [3] **Neat** (CoNEXT '16)

Improvement in Average CCT

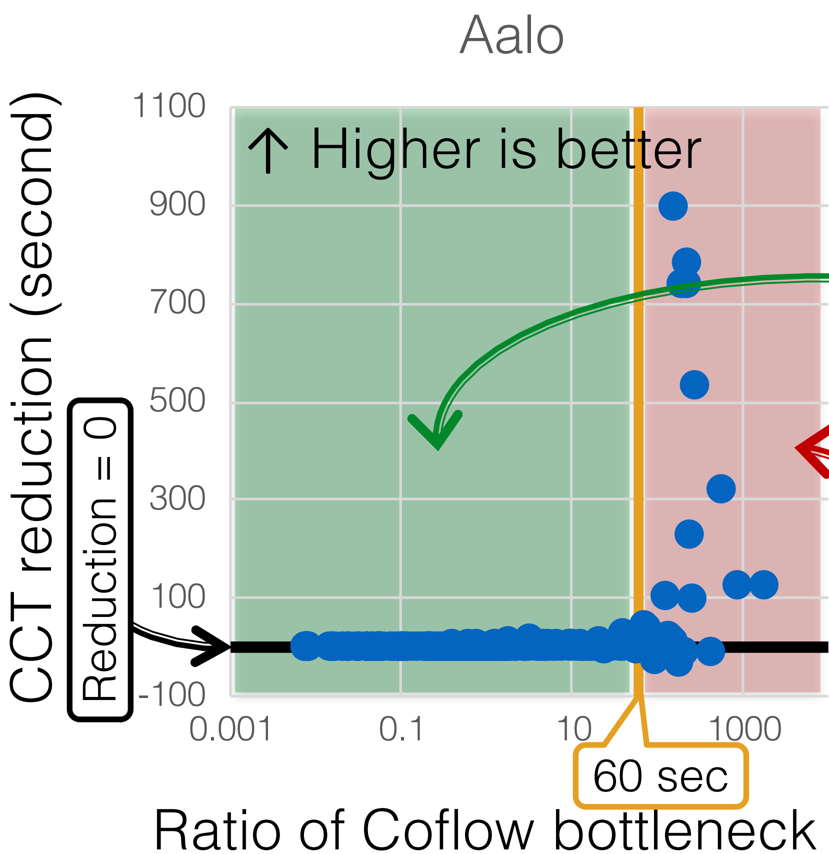
2D-Placement's average-CCT over Neat's average-CCT



2D-Placement improves over Neat by up to 23% under Aalo Scheduling.

Improvement in Individual CCT

Individual CCT Reduction by 2D-Placement from Neat



Small Coflows are prioritized and less sensitive to placement.

Large Coflows are harder to place and more sensitive to placement.

For large Coflows, 2D-Placement is only 0.85× of Neat under Aalo scheduling.

More in paper:

Results under Varys scheduling,
Sensitivity to Schedulers, …

Conclusions

- First study on **Coflow placement**, which has decisive impact on Coflow performance.
- Coflow placement is more challenging due to **inter-flow dependency**.
- **2D-Placement** leverages inter-flow relationship to find good placement for Coflows.

Thank You!

Thank You!

Xin Sunny Huang, T. S. Eugene Ng
Rice University

RICE

BOLD
Big Data and Optical Lightpaths Driven Lab

Backup slides

Sensitivity to Schedulers

- 2D-Placement's improvement over Neat is usually larger under Aalo scheduling.

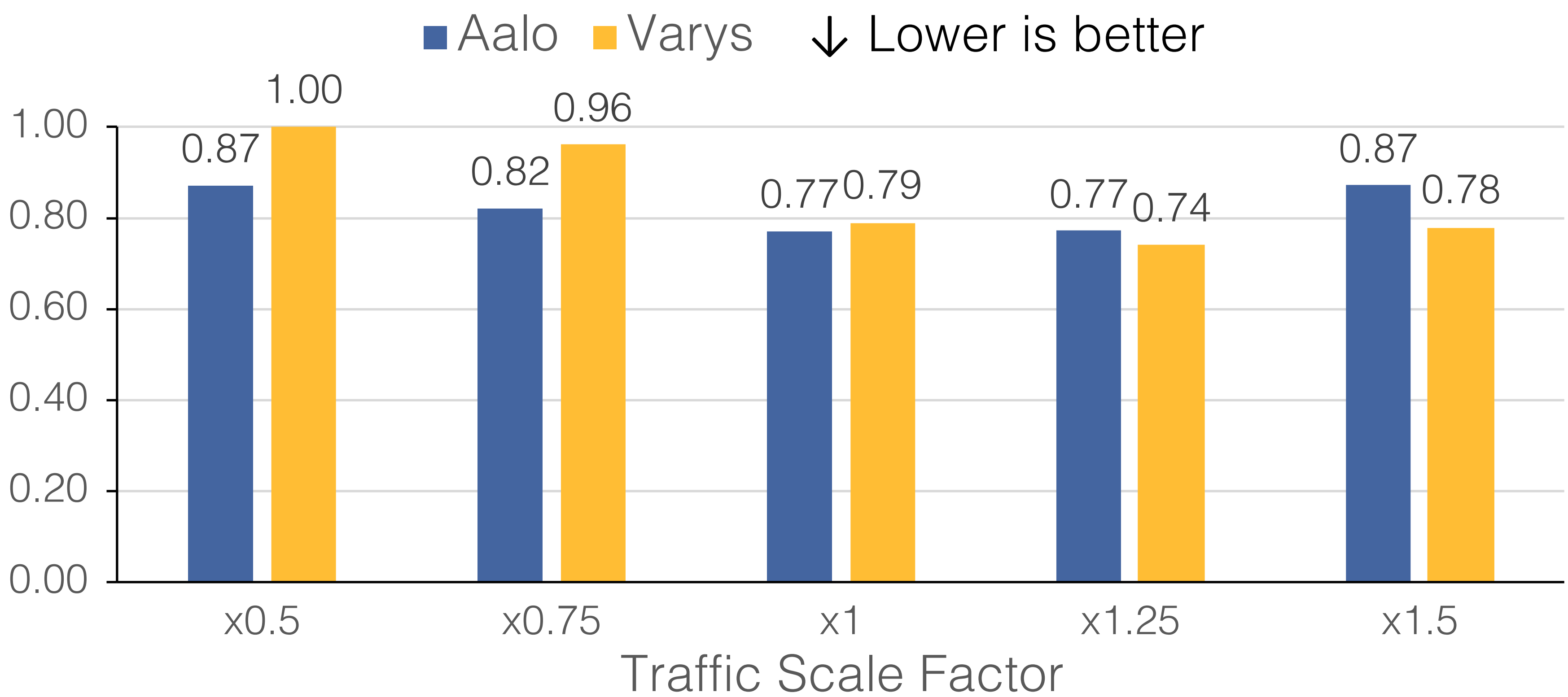
1. **Aalo**, due to lack of precise information of Coflow size, may allow temporary violation of the smallest-Coflow-first priority.

2. **Neat** optimizes placement based on a specific traffic priority used for scheduling. Thus it is prone to error in scheduling dynamics during runtime.

3. **2D-Placement** optimizes placement in a more general case independent of the scheduling.

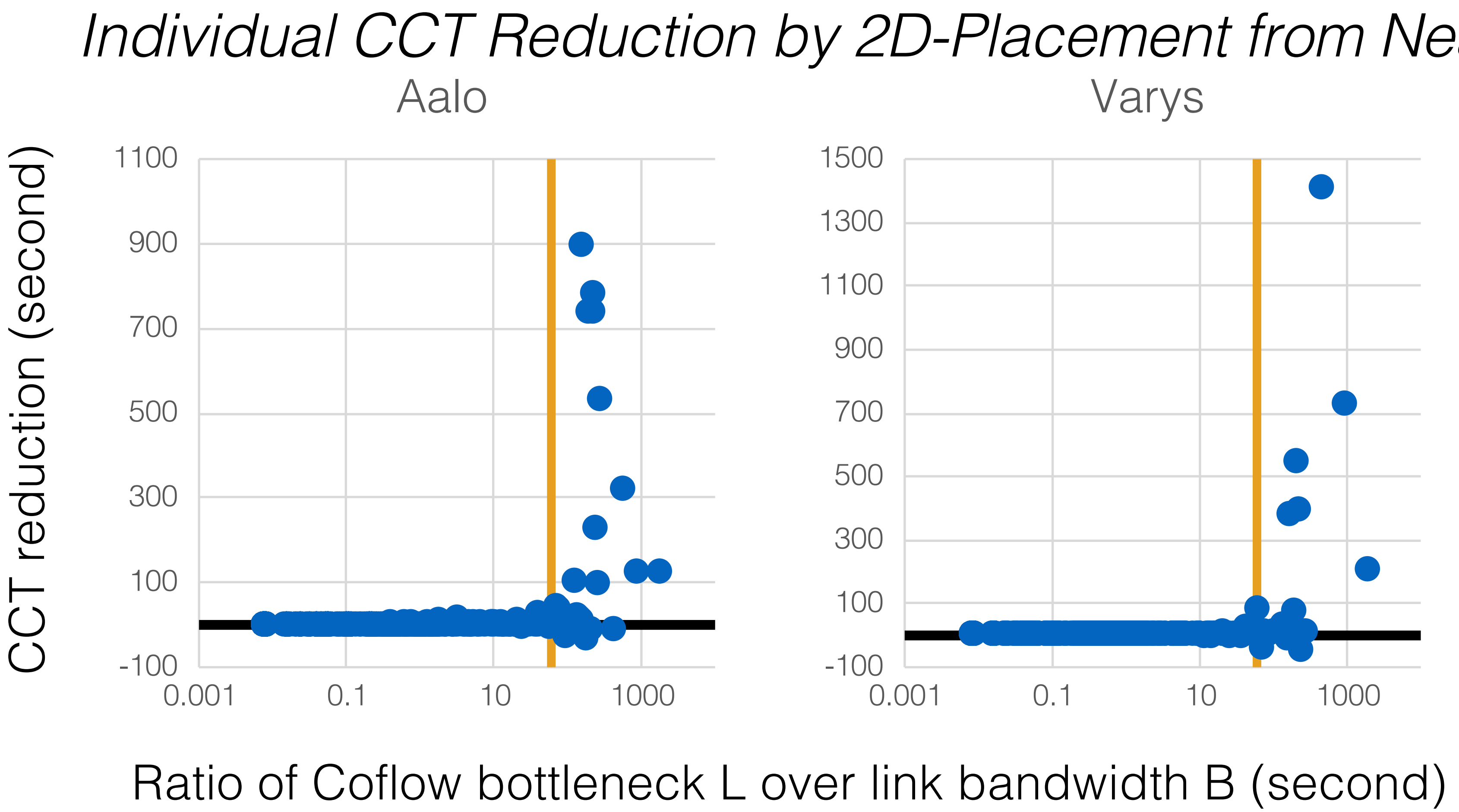
Improvement in Average CCT

2D-Placement's average-CCT over Neat's average-CCT



2D-Placement improves over Neat by up to 26%.

Improvement in Individual CCT



For large Coflows, 2D-Placement is only 0.85× (0.92×) of Neat under Aalo (Varys) scheduling.

Thank You!

Xin Sunny Huang, T. S. Eugene Ng
Rice University

Big Data and Optical Lightpaths Driven Lab