PURDUE UNIVERSITY

STyGIANet

When Light Bends to the Collective Will
A Theory and Vision for Adaptive Photonic Scale-UP Domains

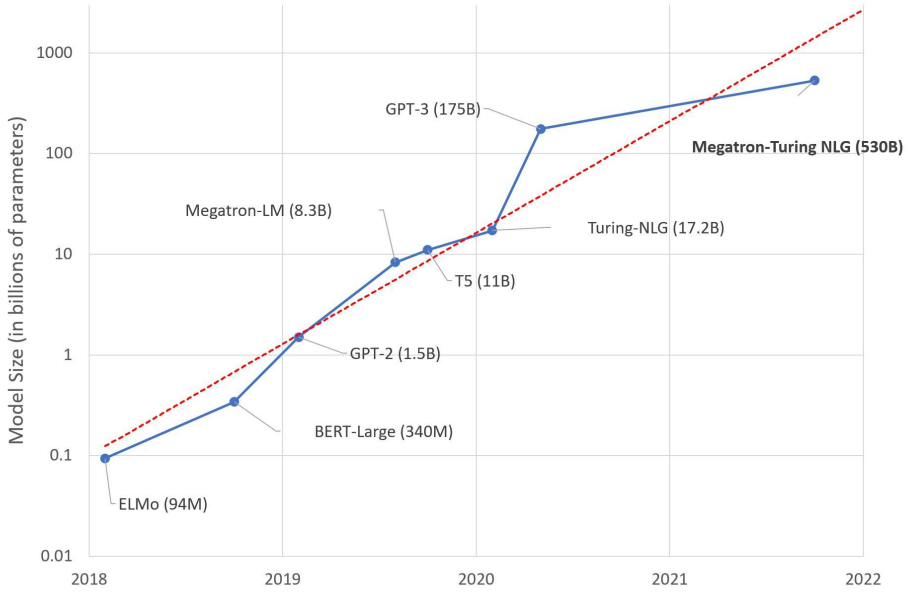
Vamsi Addanki

https://stygianet.cs.purdue.edu

ACM HotNets
18 November 2025

Exponential Growth in Model Sizes

-



PURDUE UNIVERSITY

STyGIANet

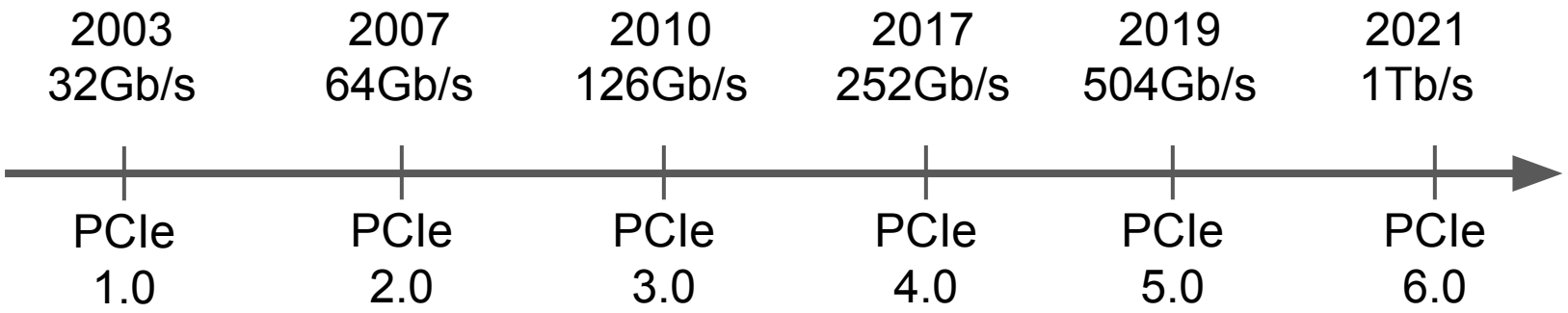
Chip-to-Chip Communication is Essential

e.g., NVIDIA DGX server

PURDUE UNIVERSITY

STyGIANet

Chip-to-Chip Communication is a Bottleneck!



PURDUE UNIVERSITY

STyGIANet

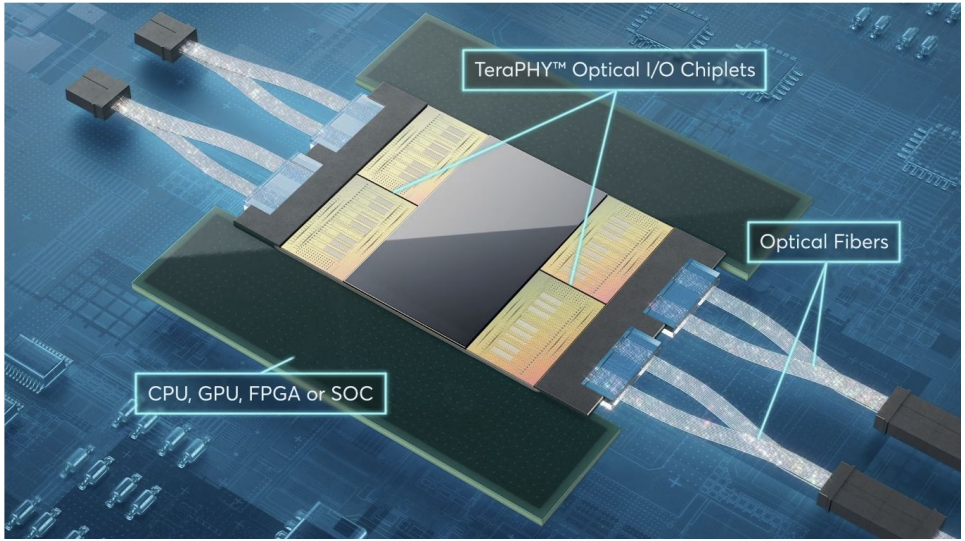
Chip-to-Chip Communication is a Bottleneck!

Memory bandwidth ~doubles every two years

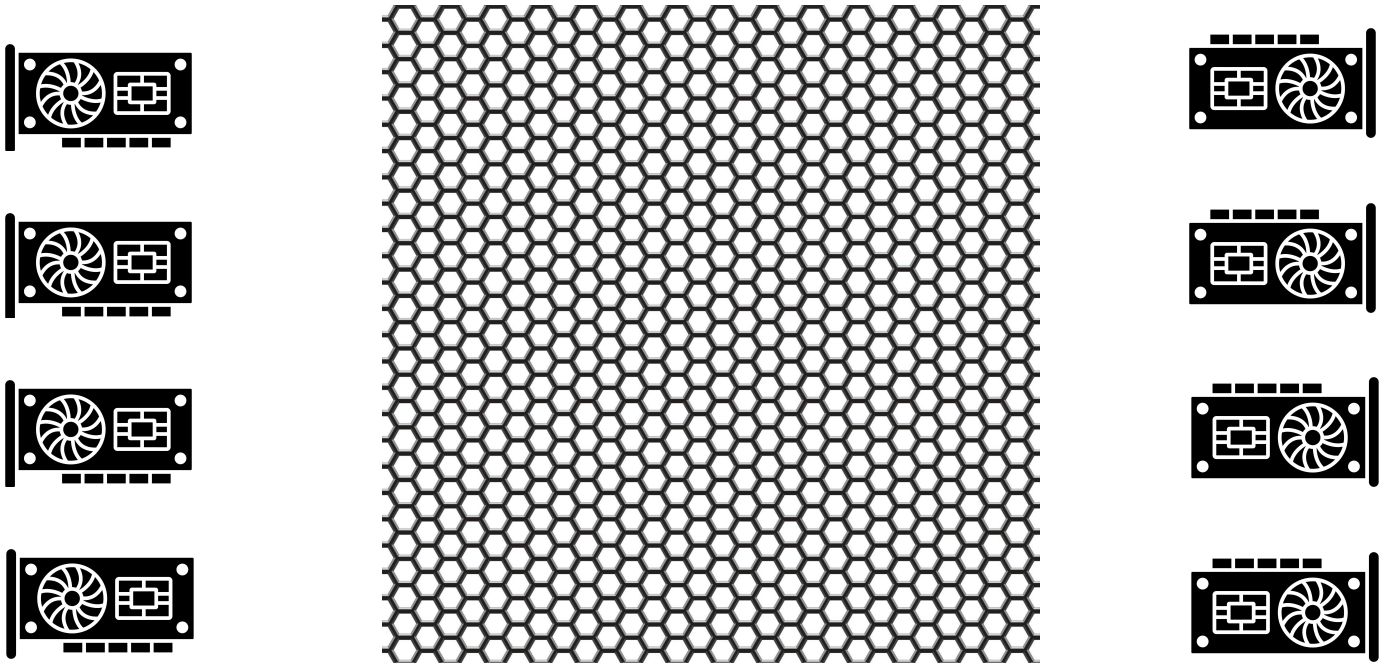
But bandwidth *requirement* is growing much faster

Chip-to-Chip Photonic Interconnects are on the Rise

- Ayar Labs: “Moving Data with Light!”



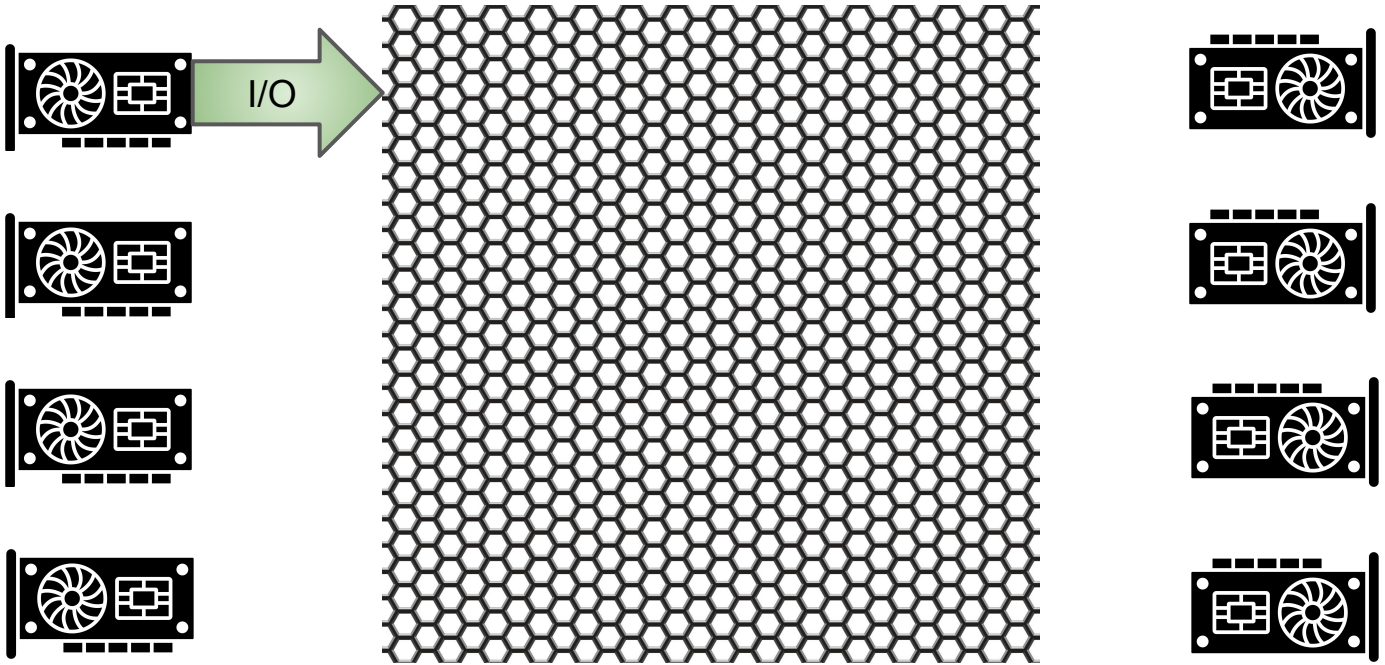
Chip-to-Chip Programmable Photonics



PURDUE UNIVERSITY

STyGIANet

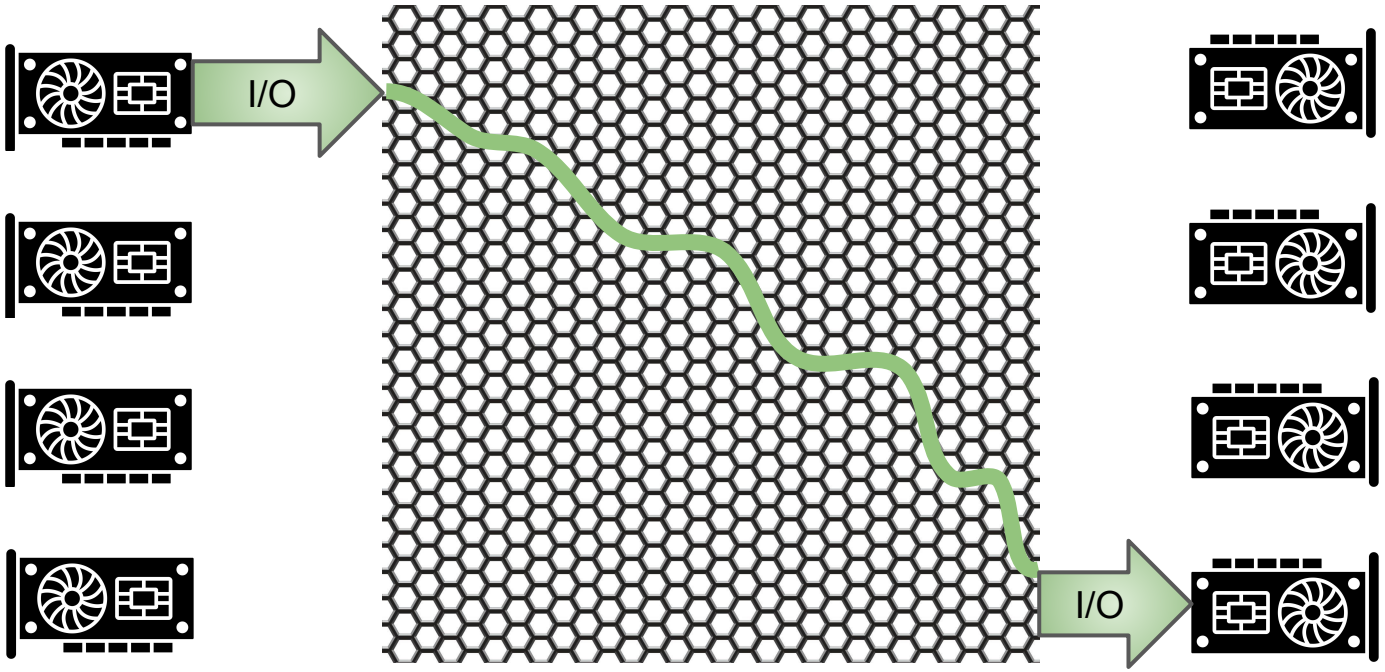
Chip-to-Chip Programmable Photonics



PURDUE UNIVERSITY

STyGIANet

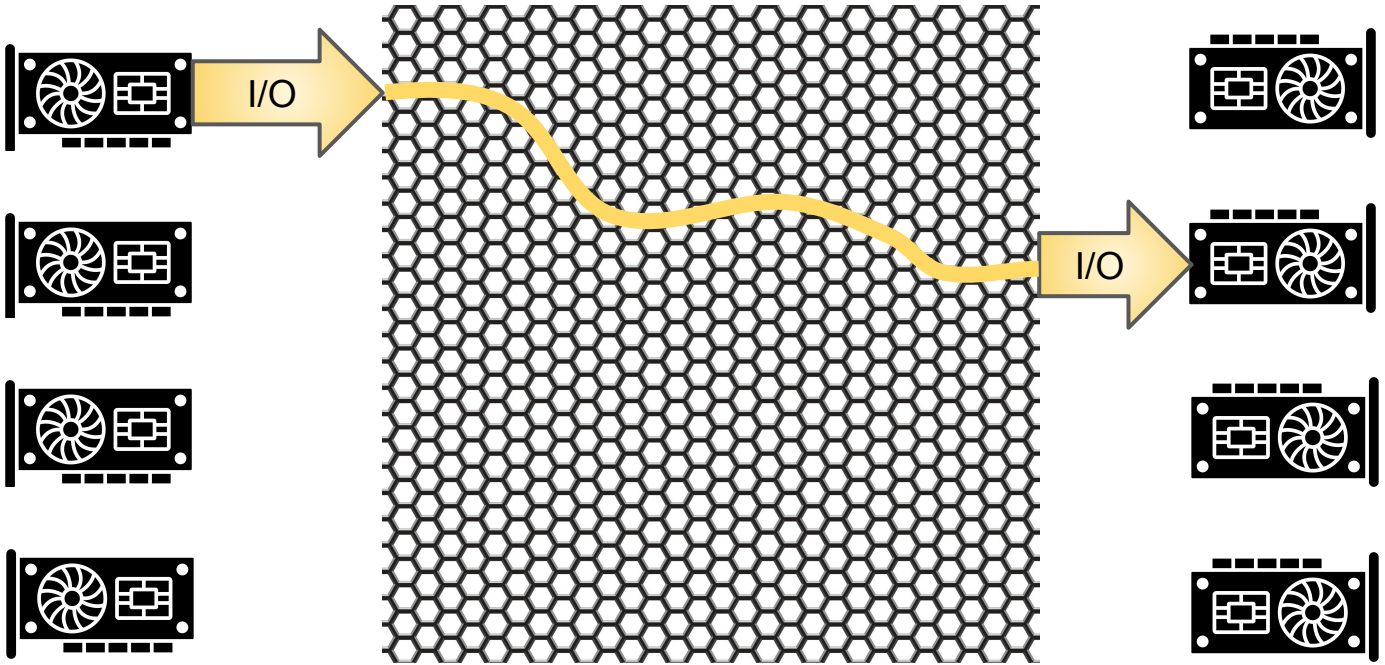
Chip-to-Chip Programmable Photonics



PURDUE UNIVERSITY

STyGIANet

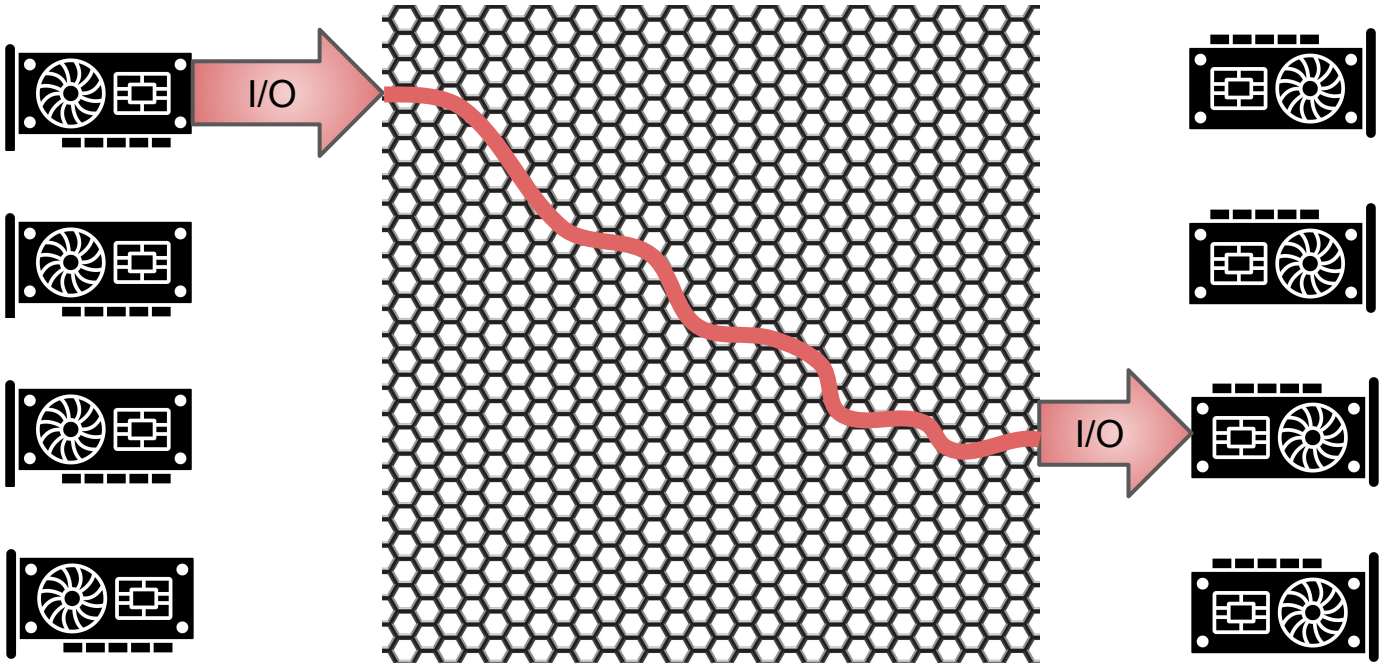
Chip-to-Chip Programmable Photonics



PURDUE UNIVERSITY

STyGIANet

Chip-to-Chip Programmable Photonics



PURDUE UNIVERSITY

STyGIANet

Reconfigurable Network Design

- Circuit-switched network
- Bufferless
- Reconfiguration delays

Reconfigurable Network Design

- Reconfigurable networks are well studied in the recent past
  - Demand-aware reconfigurable networks
    - BvN decompositions
    - Greedy matchings
  - Oblivious reconfigurable networks
    - Periodic circuit switching

Reconfigurable Network Design

- Reconfigurable networks are well studied in the recent past
  - Demand-aware reconfigurable networks
    - BvN decompositions
    - Greedy matchings
  - Oblivious reconfigurable networks
    - Periodic circuit switching

Assumption:
Reconfiguration delay is negligible

Reconfigurable Network Design

- Reconfigurable networks are well studied in the recent past
  - Demand-aware reconfigurable networks
    - BvN decompositions
    - Greedy matchings
  - Oblivious reconfigurable networks
    - Periodic circuit switching
  - One-shot optimization
  - Failure mitigation

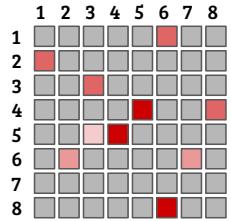
Assumption:
Reconfiguration delay is negligible

Assumption:
Reconfiguration delay is too high

PURDUE UNIVERSITY
STyGIANet

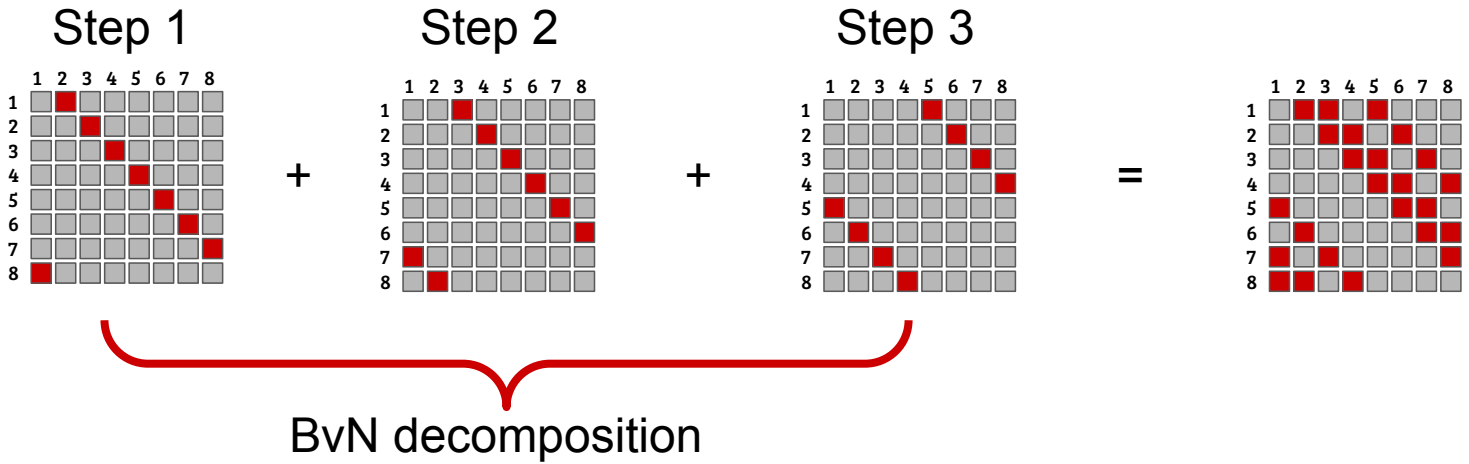
Reconfigurable Network Design

- Reality is much different!
  - Different range of reconfiguration delays across technologies, vendors…
    - Anywhere between 10s of nanoseconds to 100s of milliseconds
  - GPU communication has data dependencies
    - Aggregate demand matrix is not a good abstraction!



Key Observation (1)

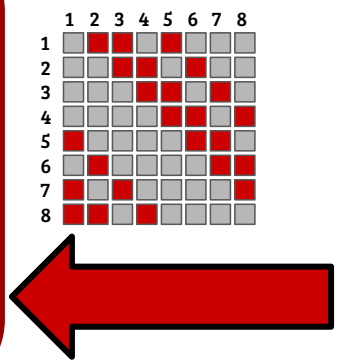
- Each step of collective communication is a matching!
  - The sequence of matchings form *a* BvN decomposition of the aggregate demand matrix



Key Observation (1)

- Each step of collective communication is a matching!
  - The sequence of matchings form *a* BvN decomposition of the aggregate demand matrix

Decomposing the aggregate matrix may not align with the collective



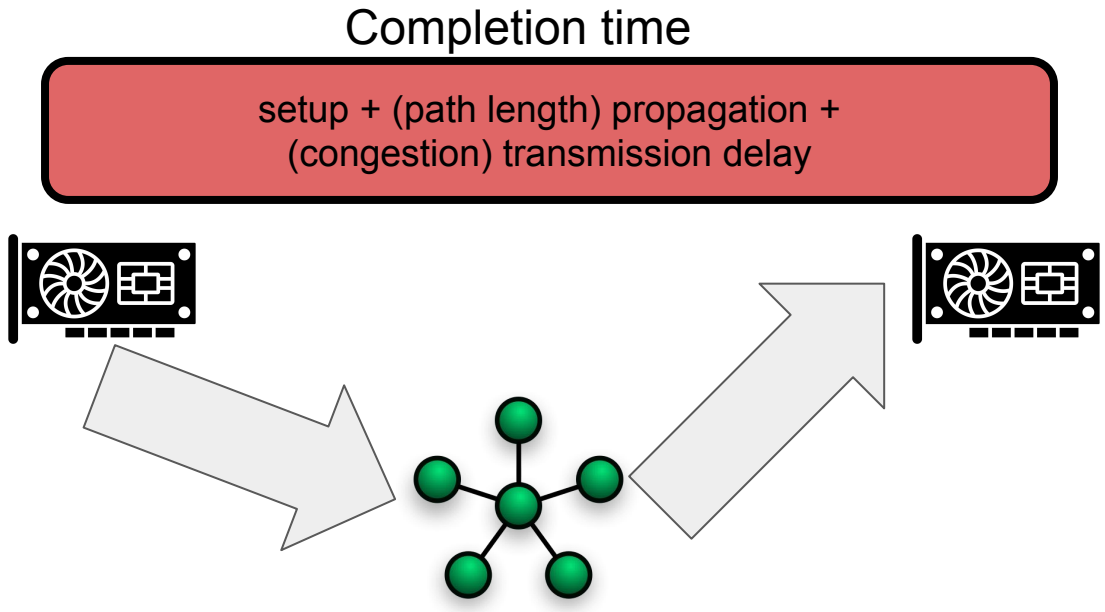
Key Observation (2)

a: Time to prepare the chunk, a setup latency

b: Link propagation delay

c: Time to transmit one bit

d: Congestion factor



PURDUE UNIVERSITY

STyGIANet

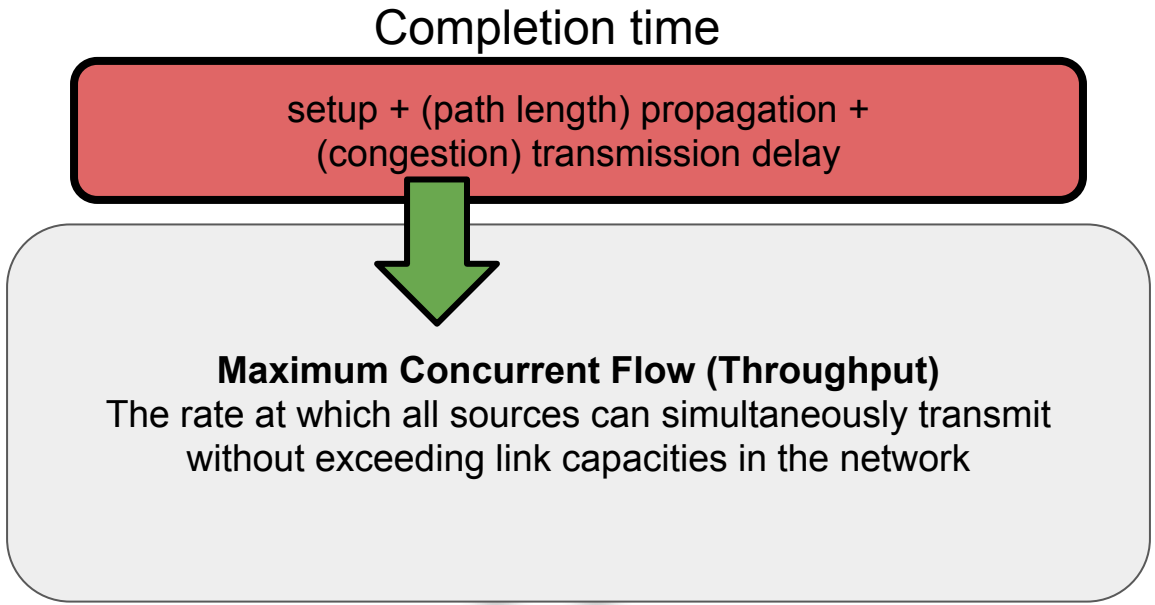
Key Observation (2)

a: Time to prepare the chunk, a setup latency

b: Link propagation delay

c: Time to transmit one bit

d: Congestion factor



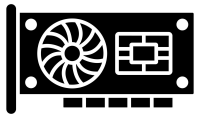
Key Observation (2)

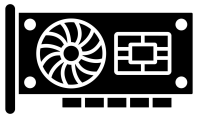
α: Time to prepare the chunk, a setup latency

δ: Link propagation delay

β: Time to transmit one bit

θ: Congestion factor





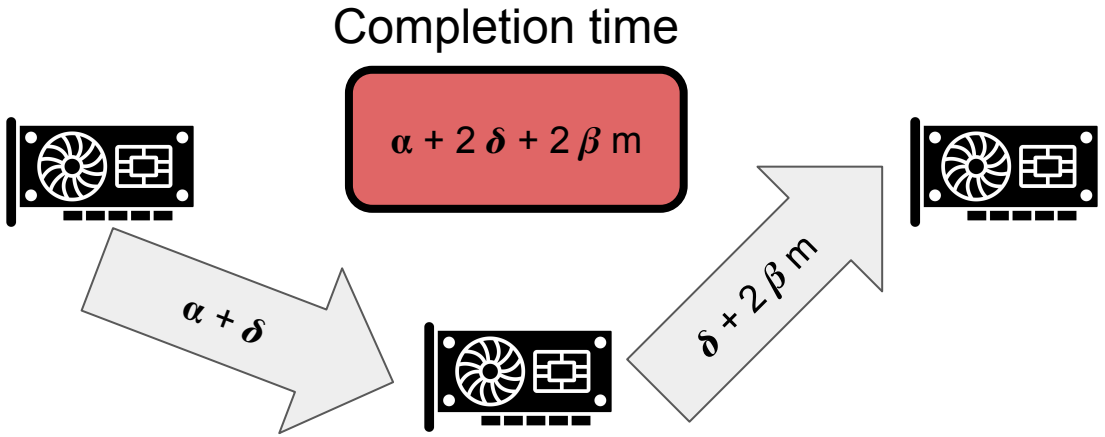
Key Observation (2)

α: Time to prepare the chunk, a setup latency

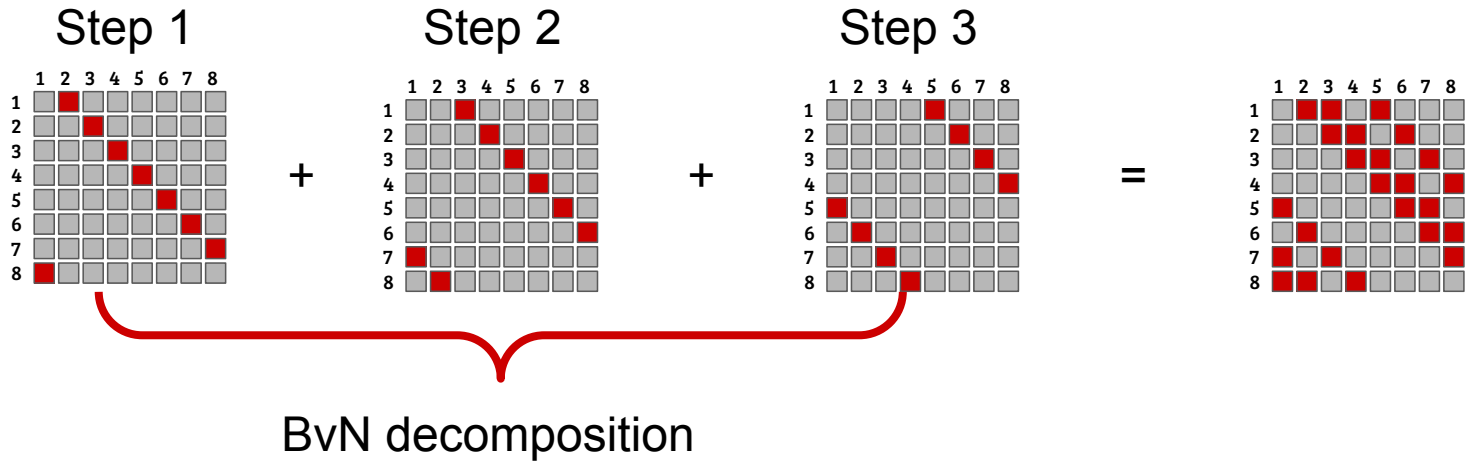
δ: Link propagation delay

β: Time to transmit one bit

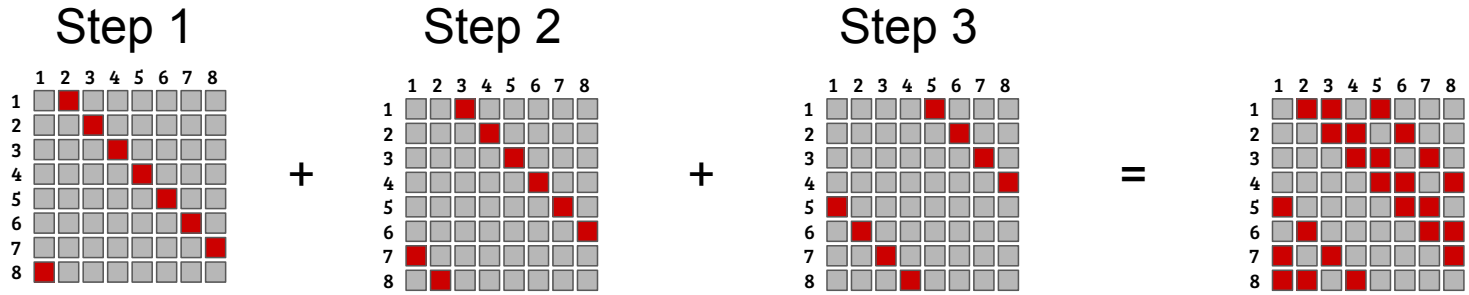
θ: Congestion factor



BvN, Concurrent Flow, and α-β Cost Model



BvN, Concurrent Flow, and α-β Cost Model



Completion time of each step of the collective

setup + (path length) propagation + (congestion) transmission delay

$\alpha + l\,\delta + \beta\, m/\theta$

α-β is not merely a cost model.

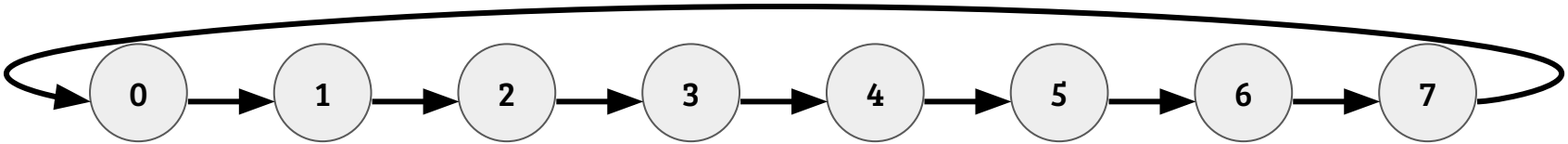
It should be seen as a *completion time* model.

AllGather Operation

- Every GPU transmits *distinct* chunks of its data to all other GPUs

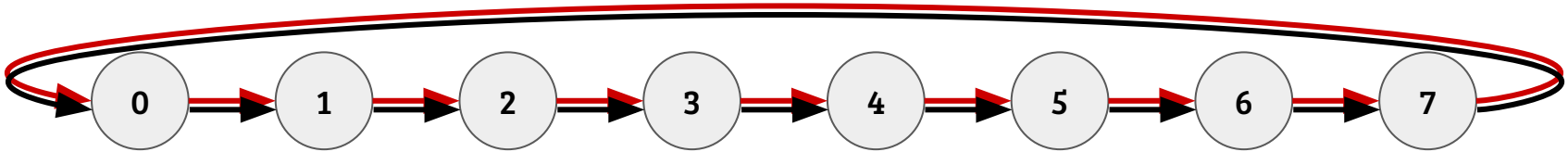
AllGather: Recursive Doubling

- 8 GPUs connected in a ring topology



AllGather: Recursive Doubling

- Step 3

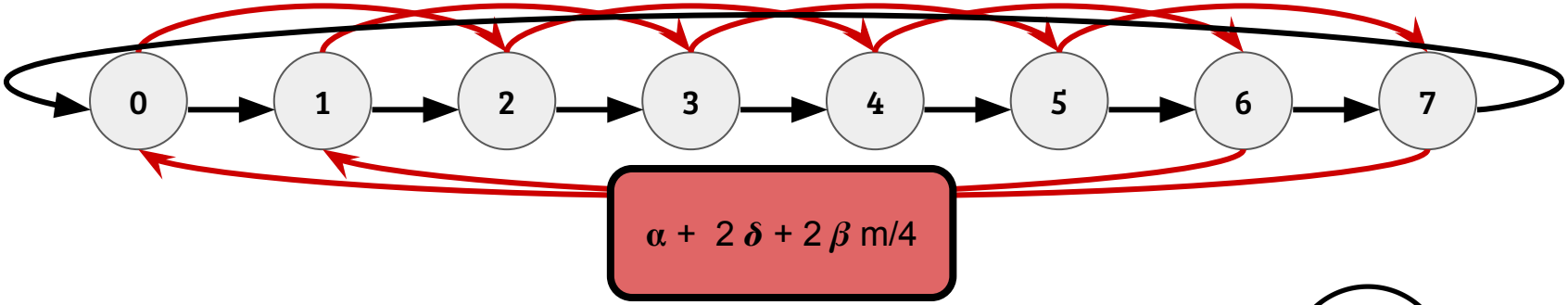


$\alpha + \delta + \beta$ m/2

No congestion

AllGather: Recursive Doubling

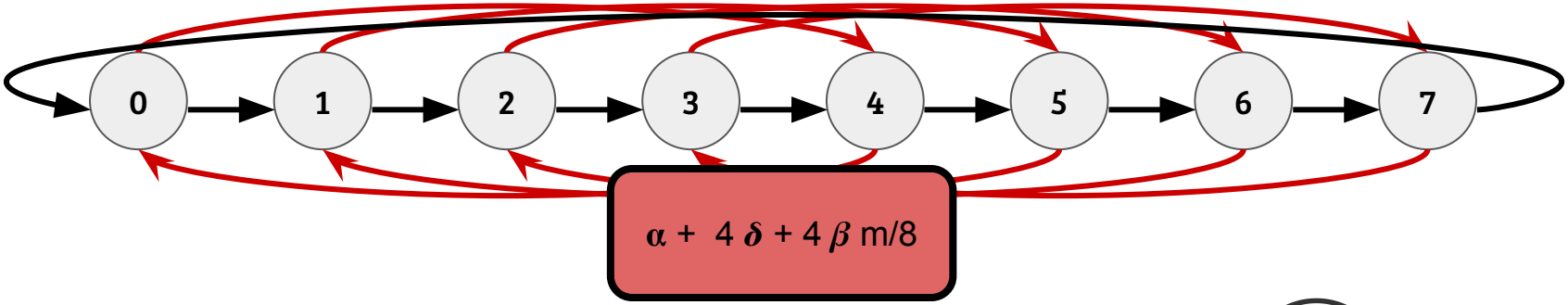
- Step 2



Moderate congestion→2 overlapping transfers

AllGather: Recursive Doubling

- Step 1

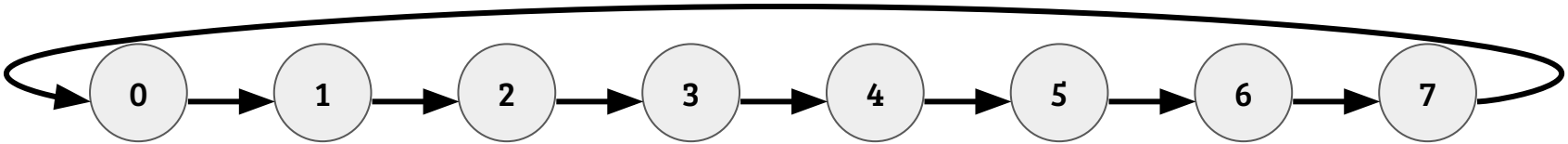


Severe congestion→4 overlapping transfers

What if we can change the topology?

AllGather: Recursive Doubling

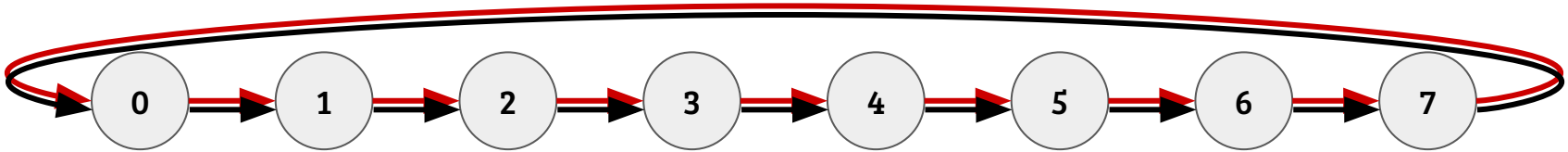
- 8 GPUs connected via *reconfigurable* photonic interconnect



Topology can be changed

AllGather: Recursive Doubling

- Step 3

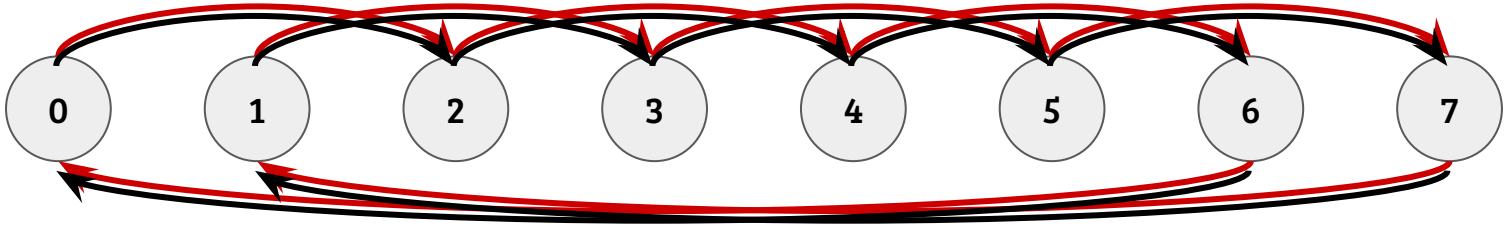


No congestion

$\alpha + \delta + \beta$ m/2

AllGather: Recursive Doubling

- Step 2



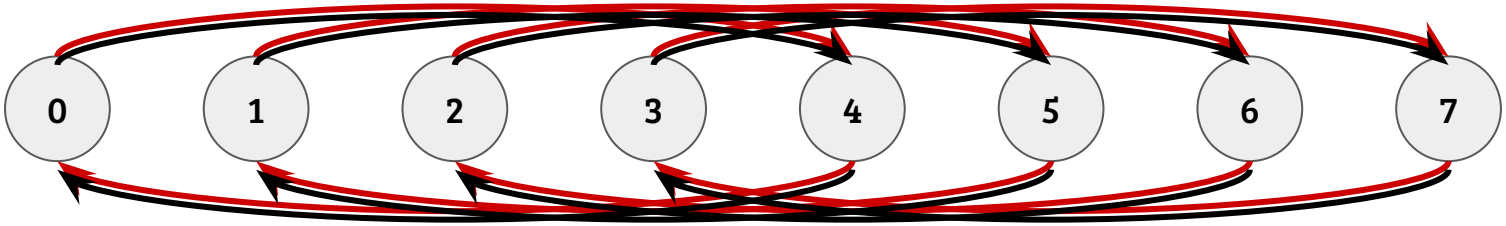
Reconfigure!

No congestion

$\alpha + \delta + \beta$ m/4

AllGather: Recursive Doubling

- Step 1

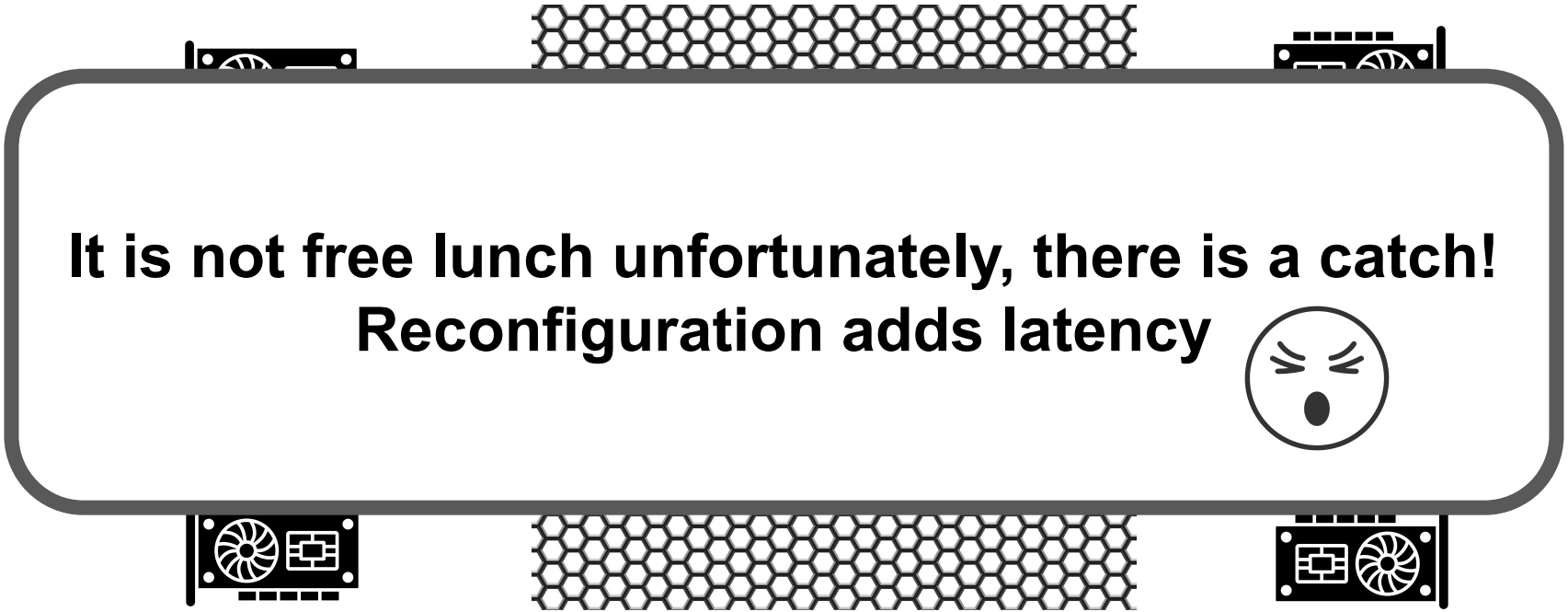


Reconfigure!

No congestion

$\alpha + \delta + \beta$ m/8

Reconfigurable Photonic Interconnect



It is not free lunch unfortunately, there is a catch!
Reconfiguration adds latency

PURDUE UNIVERSITY

STyGIANet

An Optimization Opportunity

- Tradeoff between congestion, propagation delay and reconfiguration delay
- **Central question:** When and how to reconfigure?

An Optimization Opportunity

- Decision variable: Reconfigure or not

α + α_r + □ + m β : Reconfigure and align topology with communication

α + **l** □ + m β/**θ** : Mismatch

An Optimization Opportunity

$$\min \quad s\cdot\alpha+\delta\cdot\sum_{i=1}^{s}\Bigg(\overbrace{x_i\cdot\ell_i(G)}^{\substack{\text{propagation delay}\\ \text{w/o reconf.}}} \overbrace{+(1-x_i)\cdot 1}^{\substack{\text{direct}\\ \text{with reconf.}}}\Bigg)+\sum_{i=1}^{s}\overbrace{(1-z_i)\cdot\alpha_r}^{\text{reconf. delay}}$$

$$+\quad \beta\cdot\sum_{i=1}^{s} m_i\cdot\Big(\underbrace{x_i\cdot\frac{1}{\theta(G,\mathcal{M}_i)}}_{\substack{\text{congestion}\\ \text{w/o reconf.}}}+\underbrace{(1-x_i)\cdot 1}_{\substack{\text{no congestion}\\ \text{with reconf.}}}\Big)$$

PURDUE UNIVERSITY

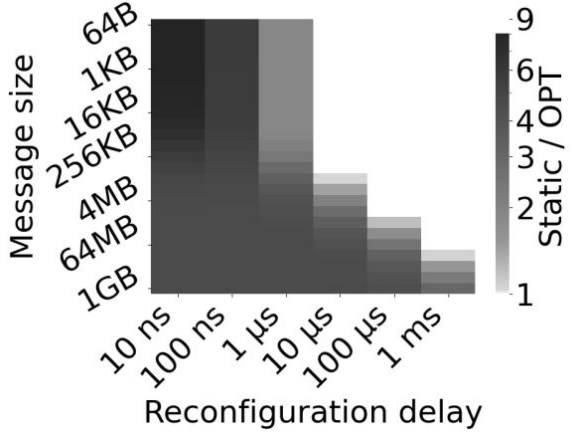
STyGIANet

An Optimization Opportunity

- Turns out the problem admits efficient dynamic programming solution for *some* collective communication algorithms, while retaining optimality

An Optimization Opportunity

- Turns out the problem admits efficient dynamic programming solution for *some* collective communication algorithms, while retaining optimality
- The payoff is significant!



PURDUE UNIVERSITY

STyGIANet

Open Questions

- What is the optimal schedule for All-to-All, Tree algorithms..?
- How should routing algorithms adapt to topology changes?
- How to implement custom routing algorithms in hardware?
- Can reconfigurations be overlapped with computations?
  - Potentially zero reconfiguration overhead!
- The reconfiguration delay is not just optical rise/fall e.g., 10ns-10us
  - GPU cuda kernels, PCIe latency, error correction, negotiation, thermal, electrical…. all contribute to the latency profile!
- Many more….

Thank You

Vamsi Addanki
vaddank@purdue.edu
@Vamsi_DT

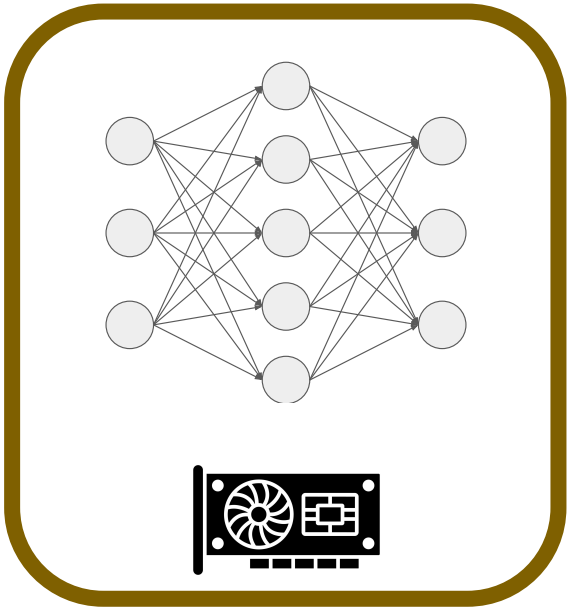
PURDUE UNIVERSITY

STyGIANet

Backup Slides

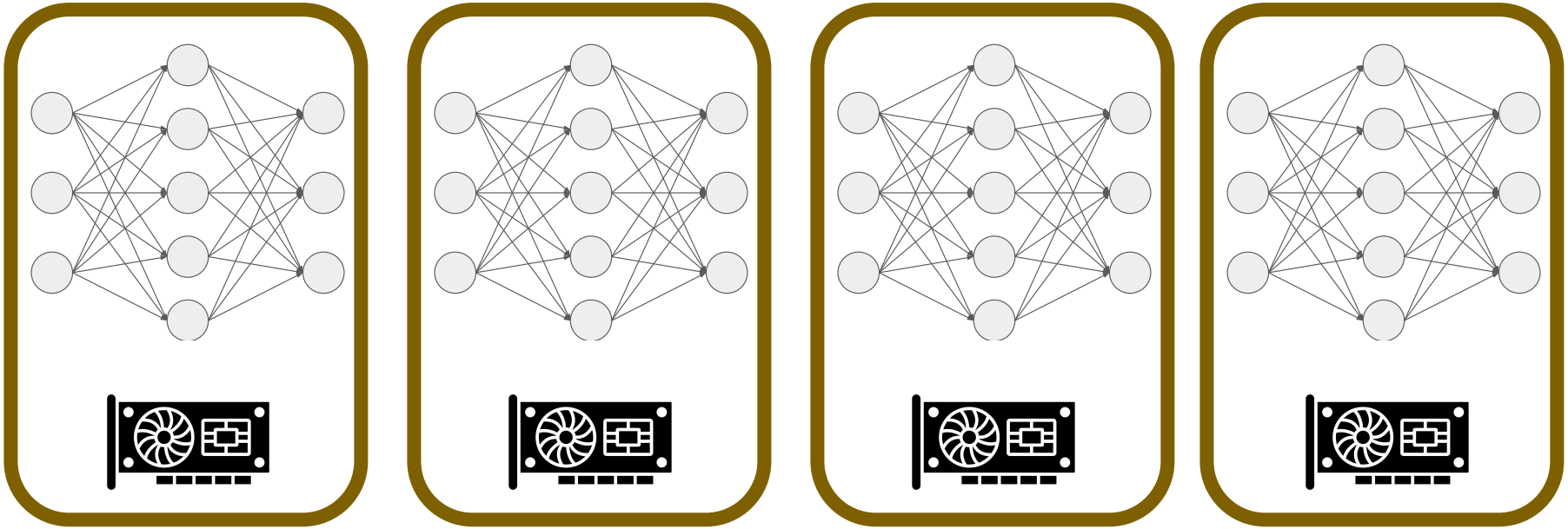
Parallelize Compute and Split Memory

- Single GPU



Parallelize Compute and Split Memory

- Data Parallelism

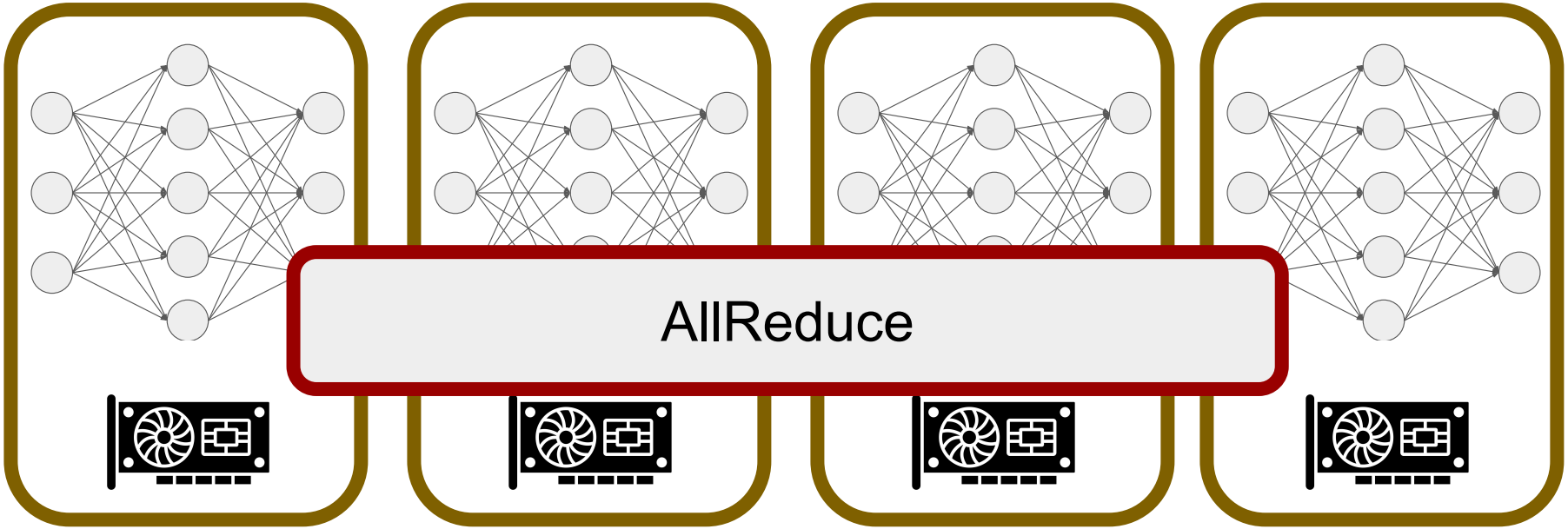


PURDUE UNIVERSITY

STyGIANet

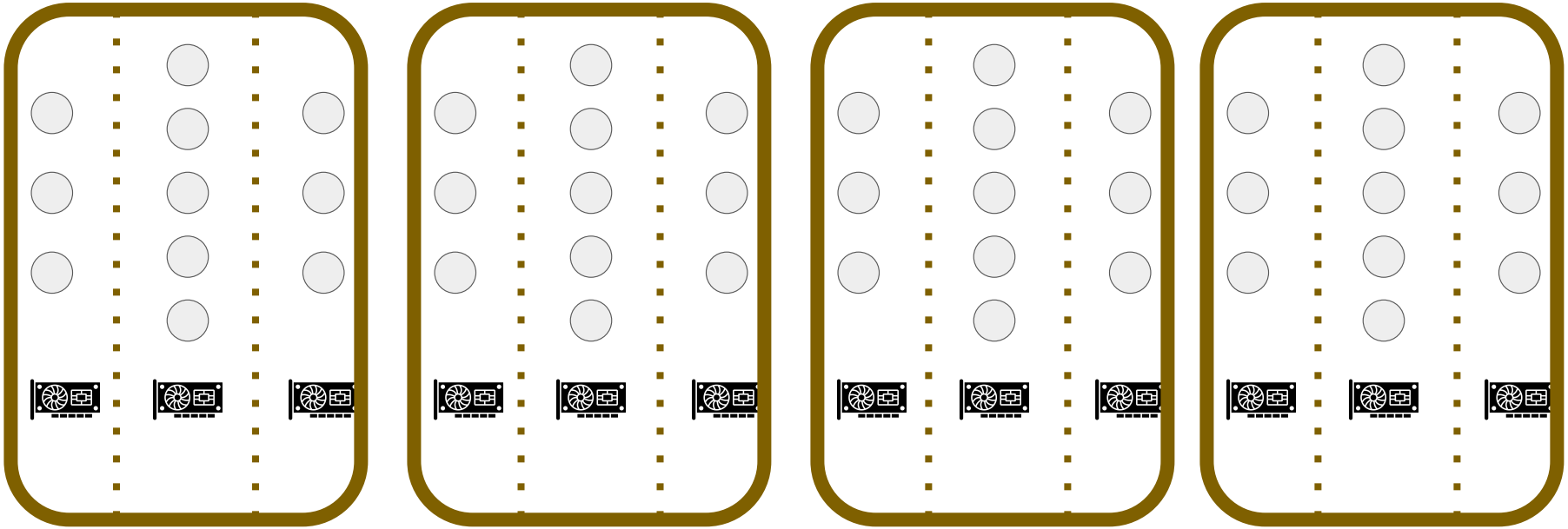
Parallelize Compute and Split Memory

- Data Parallelism



Parallelize Compute and Split Memory

- Data, Pipeline Parallelism

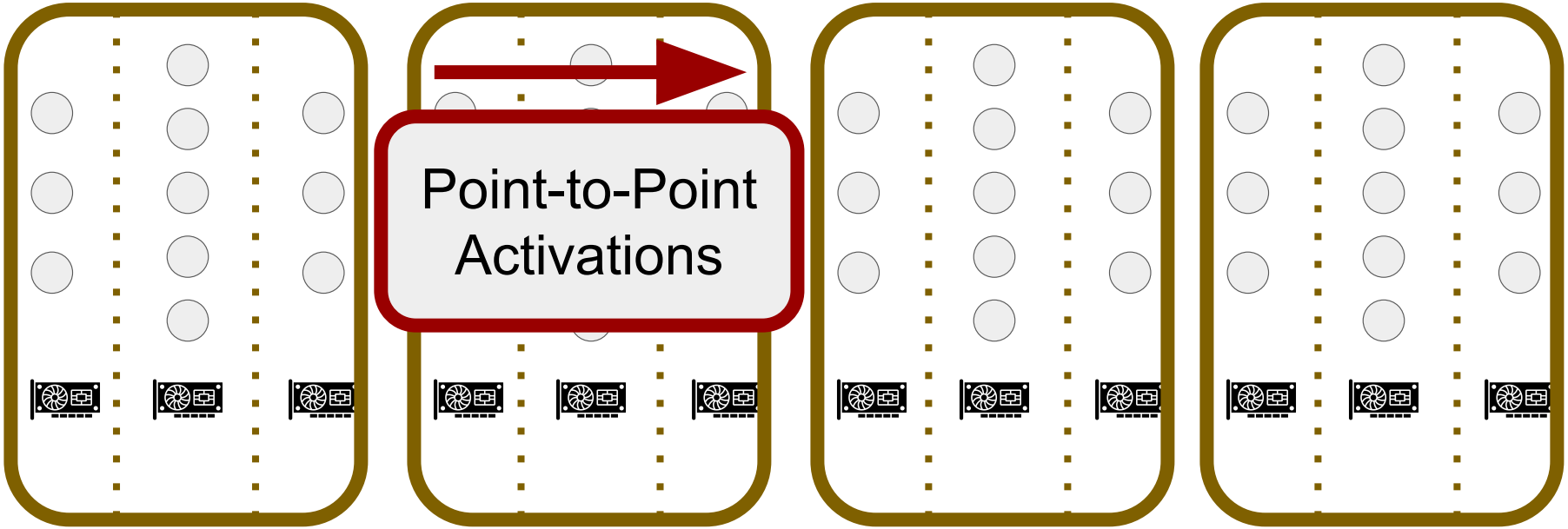


PURDUE UNIVERSITY

STyGIANet

Parallelize Compute and Split Memory

- Data, Pipeline Parallelism

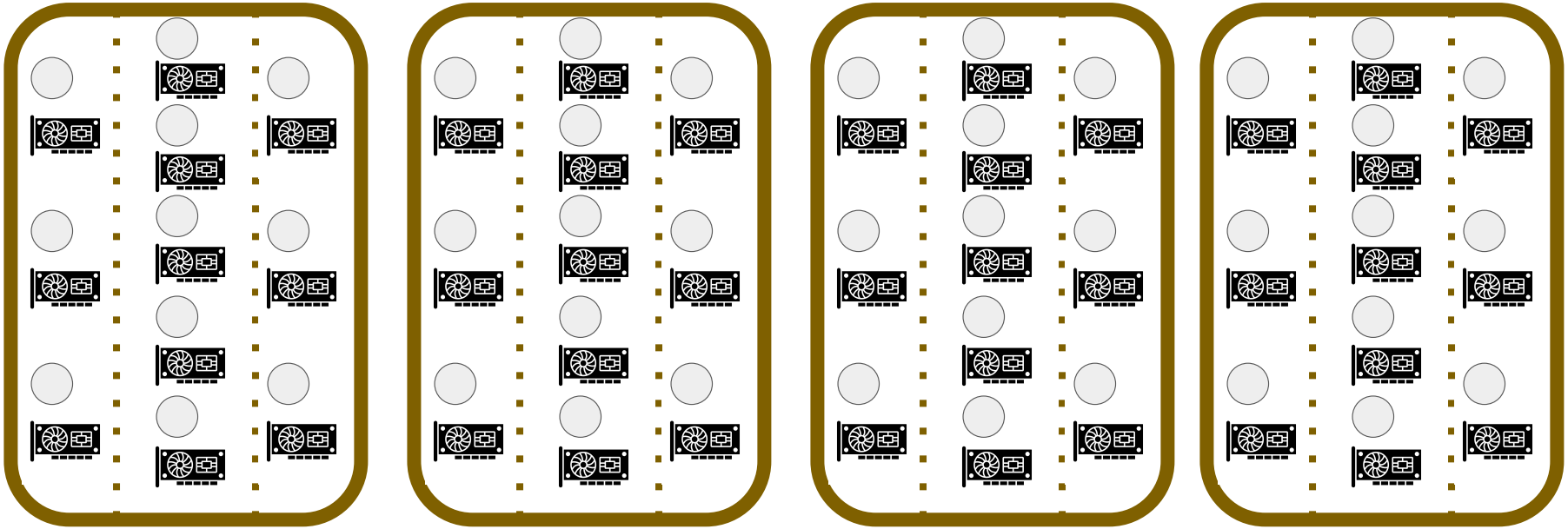


PURDUE UNIVERSITY

STyGIANet

Parallelize Compute and Split Memory

- Data, Pipeline, Tensor Parallelism

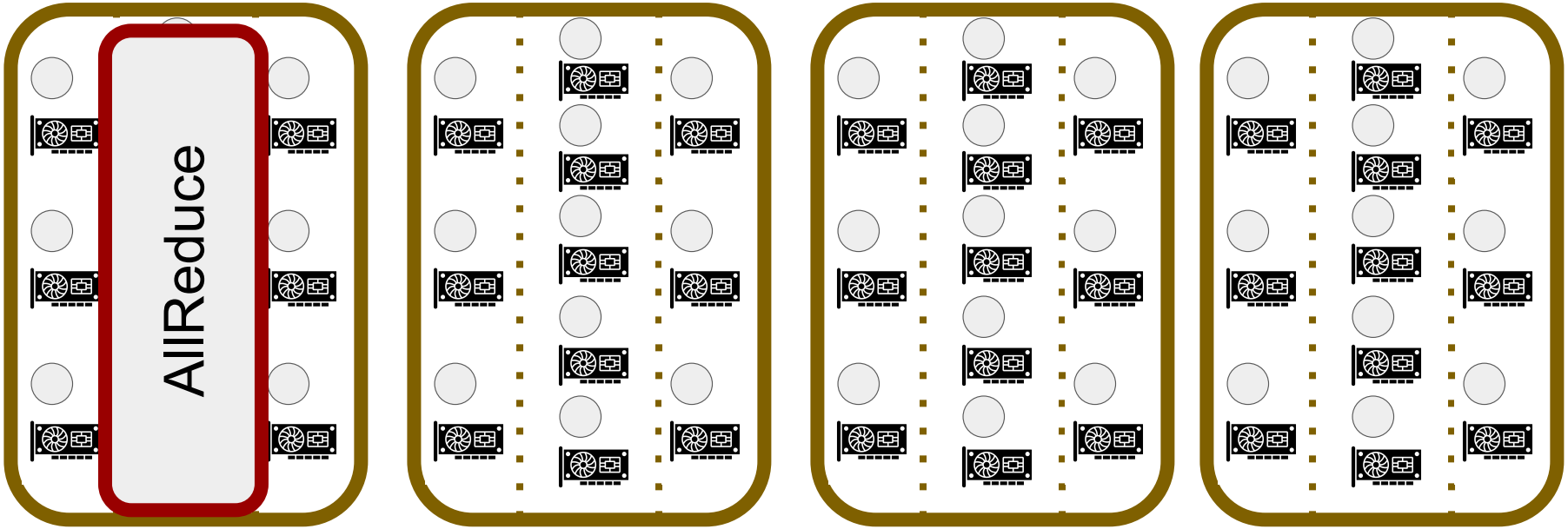


PURDUE UNIVERSITY

STyGIANet

Parallelize Compute and Split Memory

- Data, Pipeline, Tensor Parallelism



Parallelize Compute and Split Memory

- Data, Pipeline, Tensor Parallelism