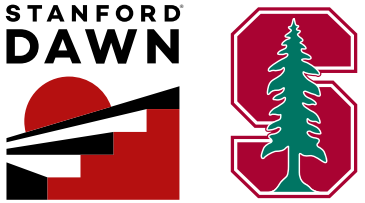
Scalable Interconnects for Reconfigurable Spatial Architectures

Yaqi Zhang, *Alexander Rucker, Matthew Vilim, Raghu Prabhakar, William Hwang, Kunle Olukotun*

Electrical Engineering
Stanford University



ISCA '19: The 46th International Symposium on Computer Architecture, Phoenix, AZ

Spatial Accelerators

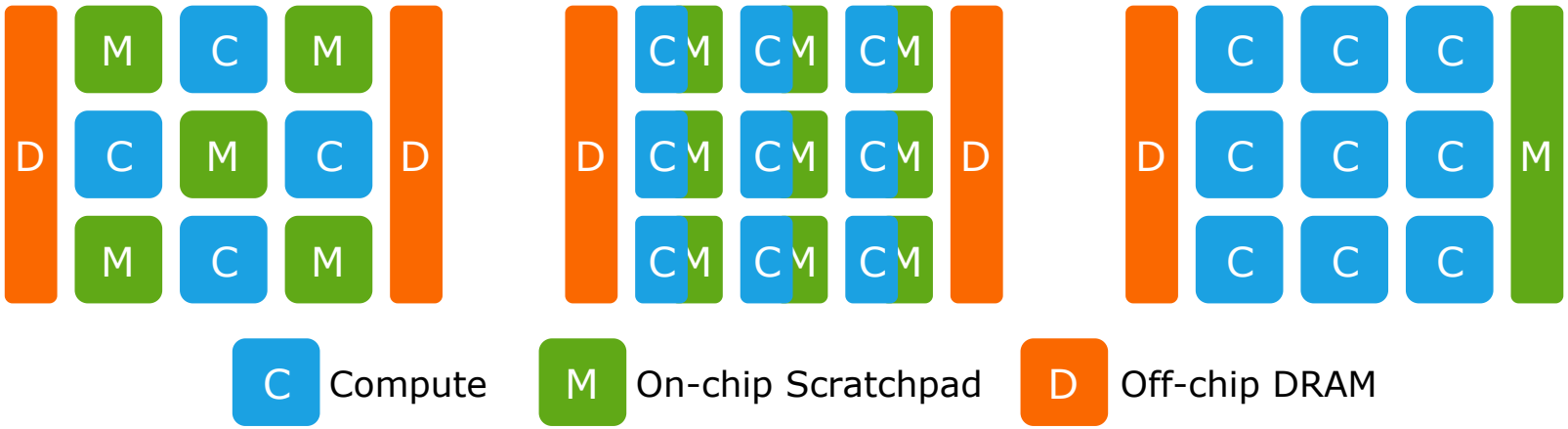
- Energy efficient
- High-throughput
- Low-latency

Examples:

- Plasticine (ISCA ‘17)
- Compressed-sparse CNN accelerator (ISCA ‘17)
- Stream-dataflow accelerator (ISCA ‘17)

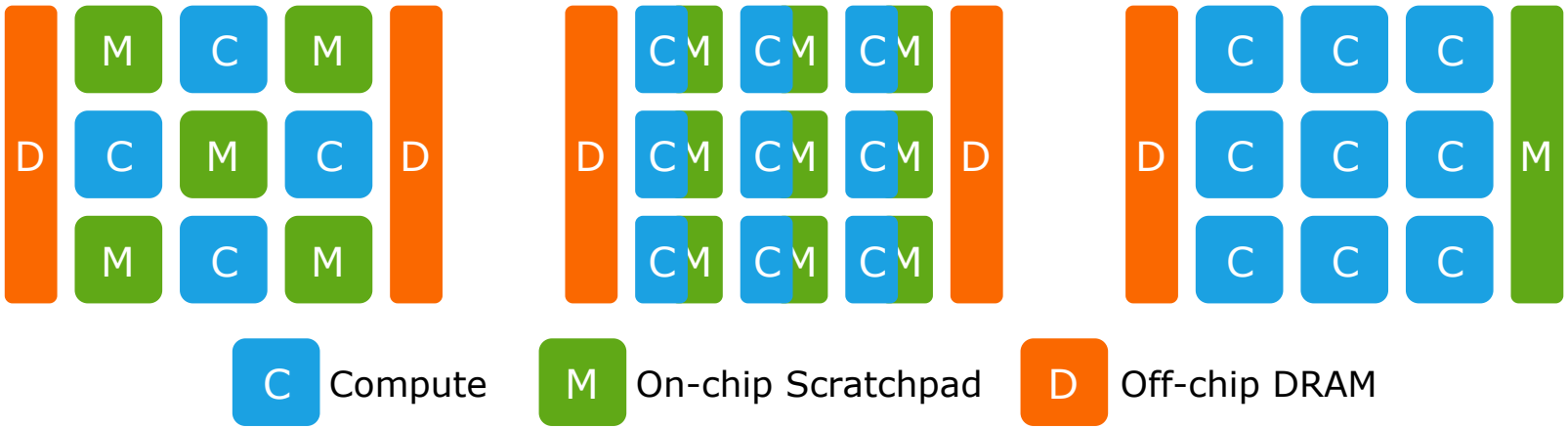
Accelerator Characteristics

- High compute density
- High on-chip memory bandwidth



Accelerator Characteristics

- High compute density
- High on-chip memory bandwidth
- Distributed compute and memory resources
- Streaming interface between compute and memory
- Statically mapped and scheduled compute graph



Key Challenges

On-chip networks play a critical role in:

Key Challenges

On-chip networks play a critical role in:

- Energy efficiency (↓ data movement)

Key Challenges

On-chip networks play a critical role in:

- Energy efficiency (↓ data movement)
- Flexibility

Key Challenges

On-chip networks play a critical role in:

- Energy efficiency (↓ data movement)
- Flexibility
- Scalability

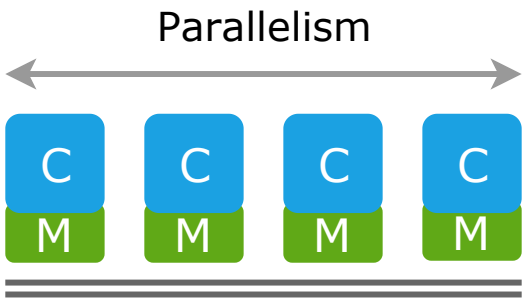
Key Challenges

On-chip networks play a critical role in:

- Energy efficiency (↓ data movement)
- Flexibility
- Scalability
- **Compute utilization**

Communication Patterns

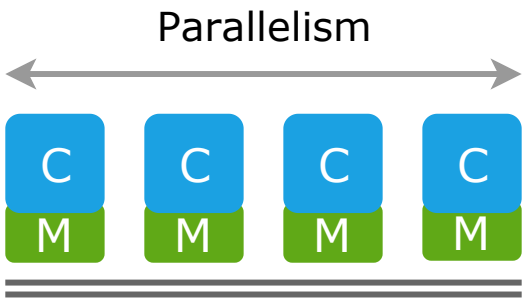
Architecture	Communication		Limited by
	Frequency	Granularity	
Processor	Infrequent	Packet	Latency



Multi-Processor

Communication Patterns

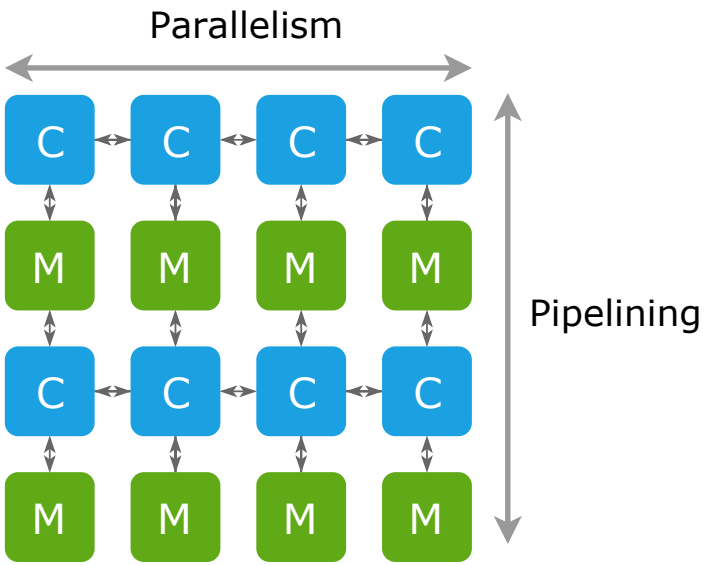
Architecture	Communication		Limited by
	Frequency	Granularity	
Processor	Infrequent	Packet	Latency



Multi-Processor

Communication Patterns

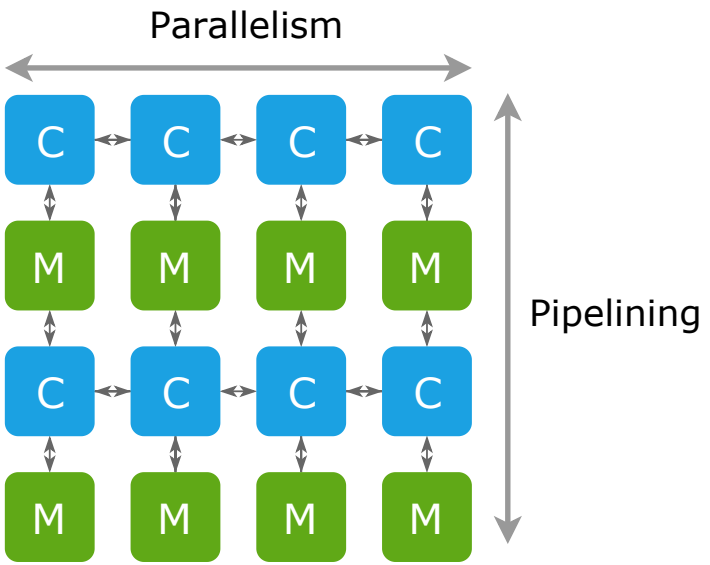
Architecture	Communication		Limited by
	Frequency	Granularity	
Processor	Infrequent	Packet	Latency
Spatial Accelerator	Frequent	Fine-grained	



Spatial Accelerator

Communication Patterns

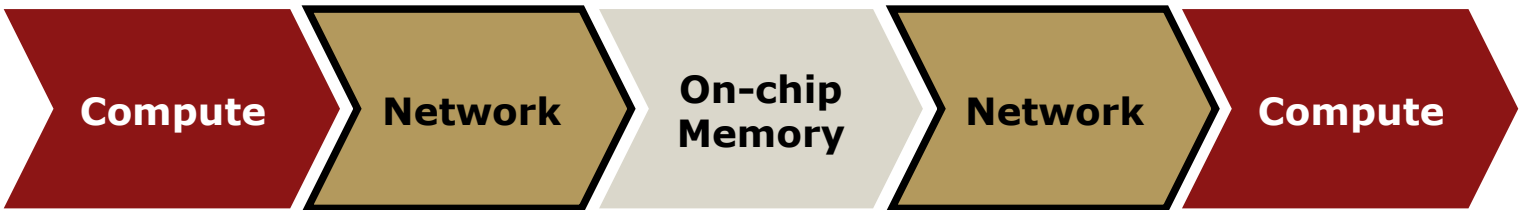
Architecture	Communication		Limited by
	Frequency	Granularity	
Processor	Infrequent	Packet	Latency
Spatial Accelerator	Frequent	Fine-grained	



Spatial Accelerator

Communication Patterns

Architecture	Communication		Limited by
	Frequency	Granularity	
Processor	Infrequent	Packet	Latency
Spatial Accelerator	Frequent	Fine-grained	**Throughput**



Outline

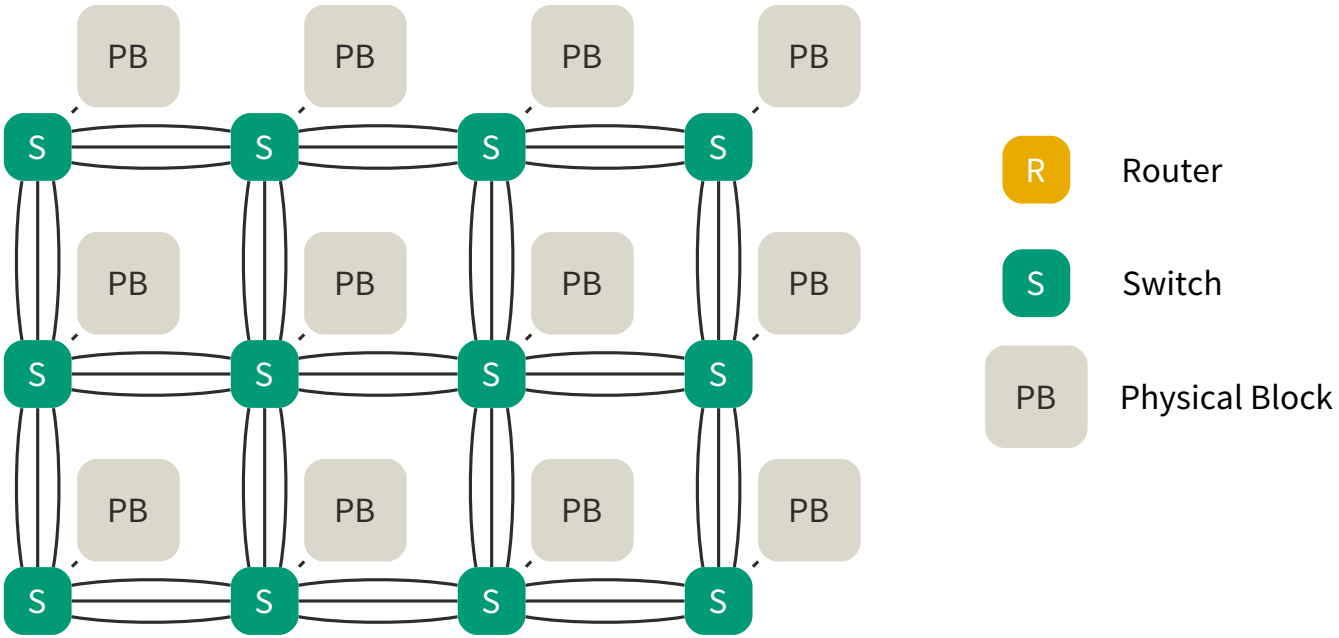
Motivation

Network Design Space

Compilation Flow

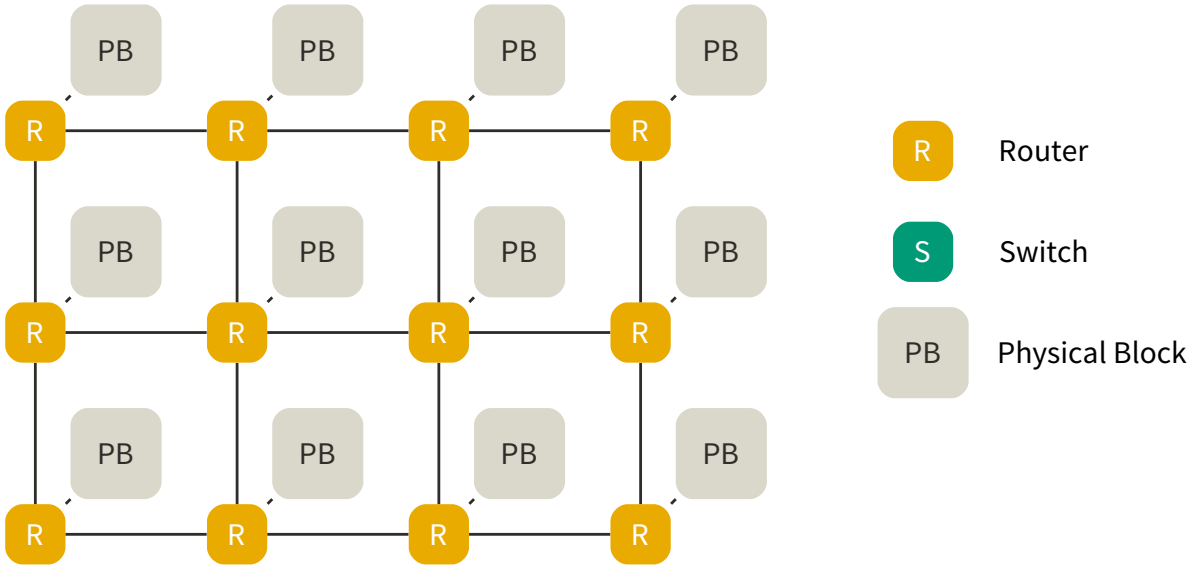
Evaluation

Static Network



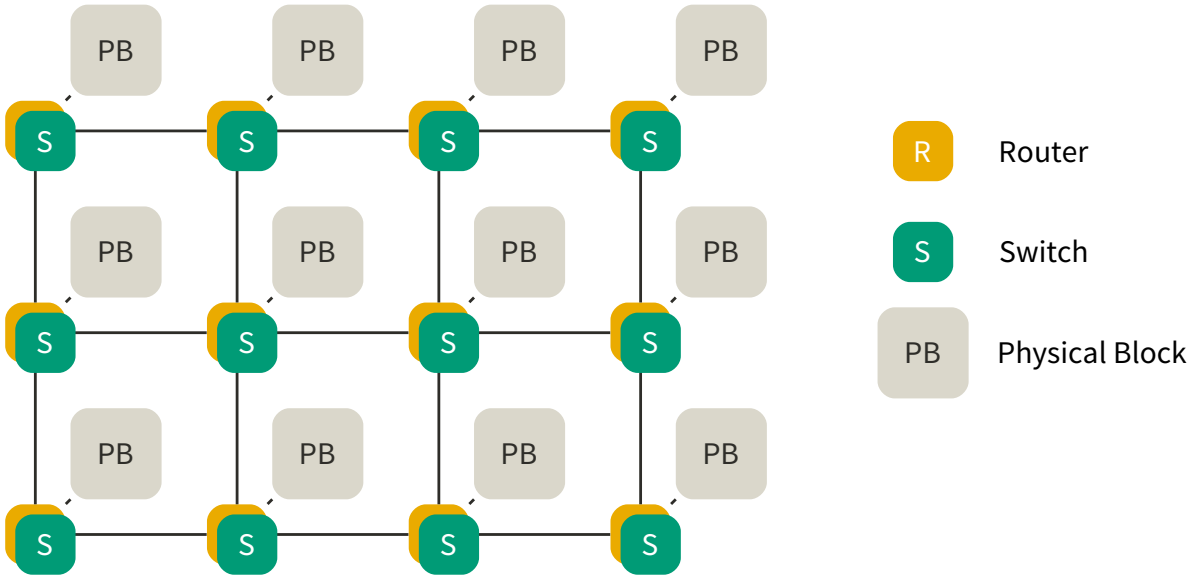
Pros	Cons
Guaranteed bandwidth	Low link utilization P&R failures

Dynamic Network



Pros	Cons
Link sharing	Limited bandwidth Deadlock

Hybrid Network: Static and Dynamic



Pros	Cons
Link sharing More bandwidth Guaranteed P&R	More area More static power

Outline

Spatial

A DSL for Reconfigurable Accelerators

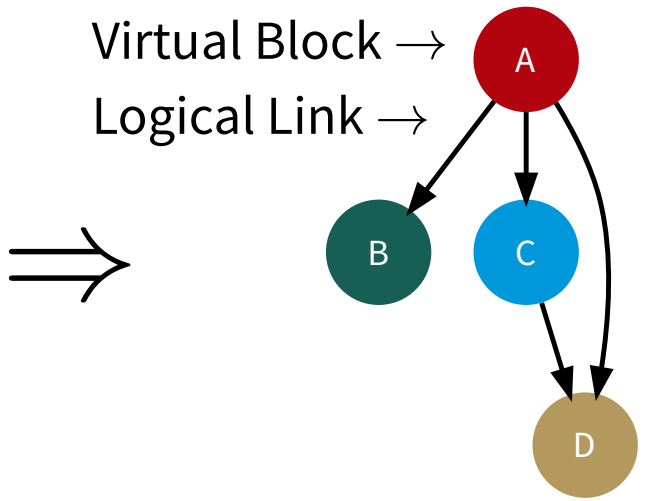
```
1  // Tiled inner product
2  val vecA,vecB = DRAM[T](N)
3  Reduce(Reg[T])( N ){ i =>
4    val tileA,tileB = SRAM[T](tilesize)
5    tileA load vecA(i::i+tilesize)
6    tileB load vecB(i::i+tilesize)
7    Reduce(Reg[T])(tilesize) { j =>
8      tileA(j) * tileB(j)
9    } { _ + _ }
10 }
```

- Annotate data size N
- Calculate loop iterations

Accelerator Compiler

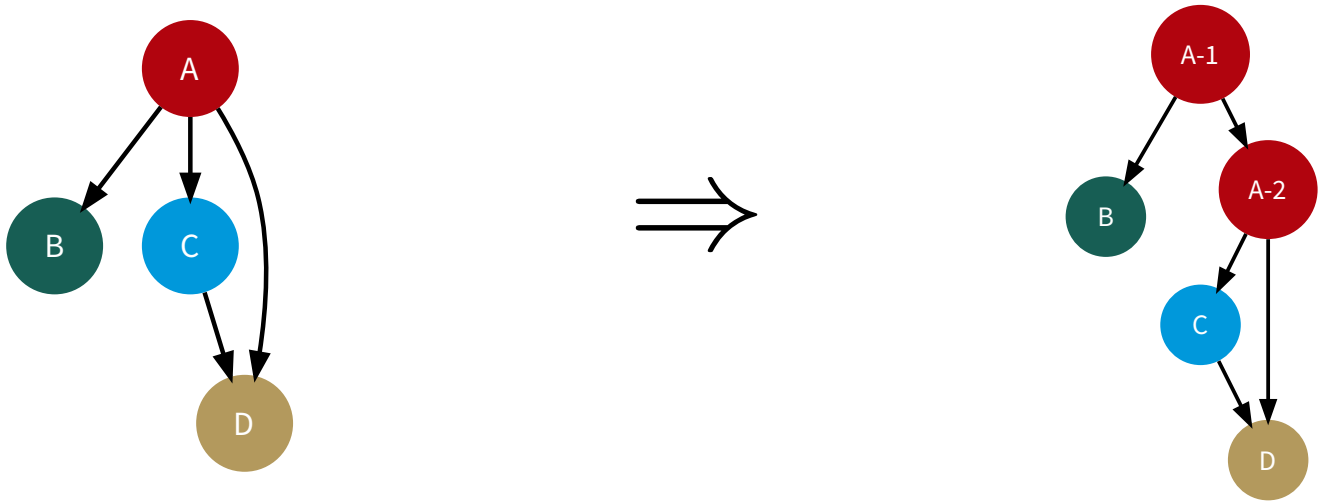
- Allocate compute and memory Virtual Blocks (VBs)
- Infer activation counts for logical links

```
1   // Tiled inner product
2   val vecA,vecB = DRAM[T](N)
3   Reduce(Reg[T])( N ){ i =>
4     val tileA,tileB = SRAM[T](tilesize)
5     tileA load vecA(i::i+tilesize)
6     tileB load vecB(i::i+tilesize)
7     Reduce(Reg[T])(tilesize) { j =>
8       tileA(j) * tileB(j)
9     } { _ + _ }
10  }
```



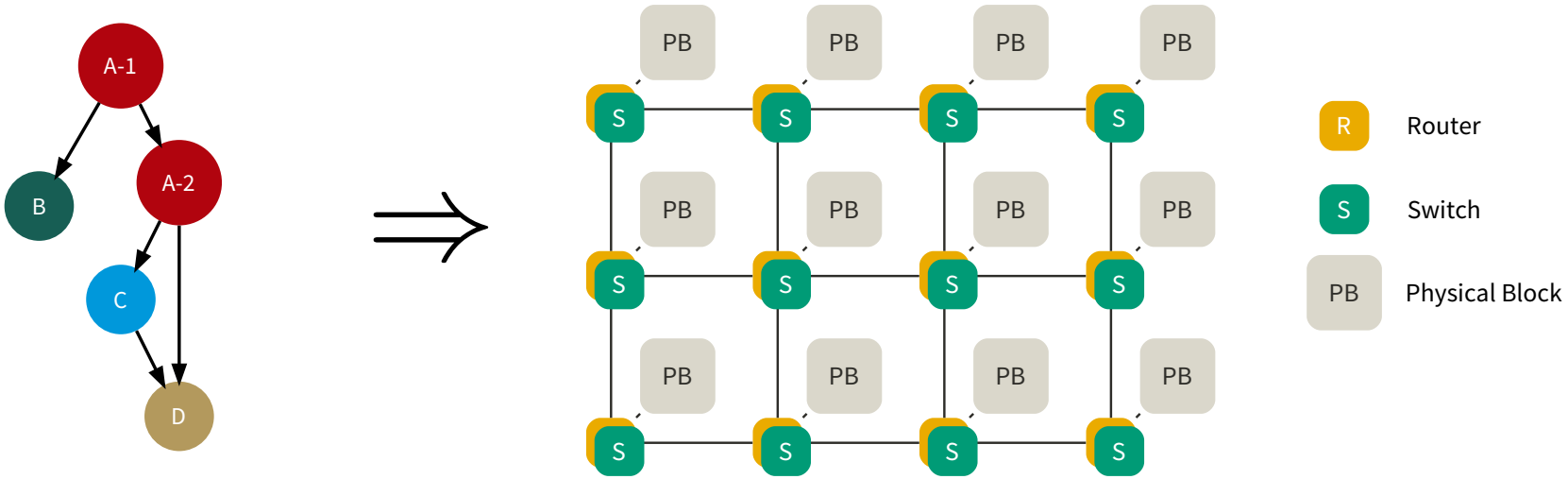
Mapping

- Partition VB graph to meet hardware constraints



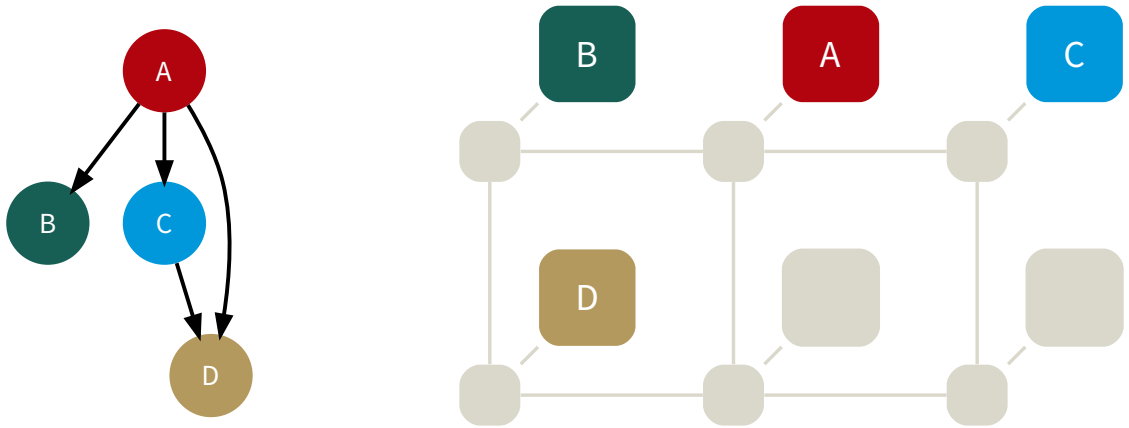
Mapping

- Partition VB graph to meet hardware constraints
- Place and route the VB graph onto the network
- Allocate VCs for the dynamic network



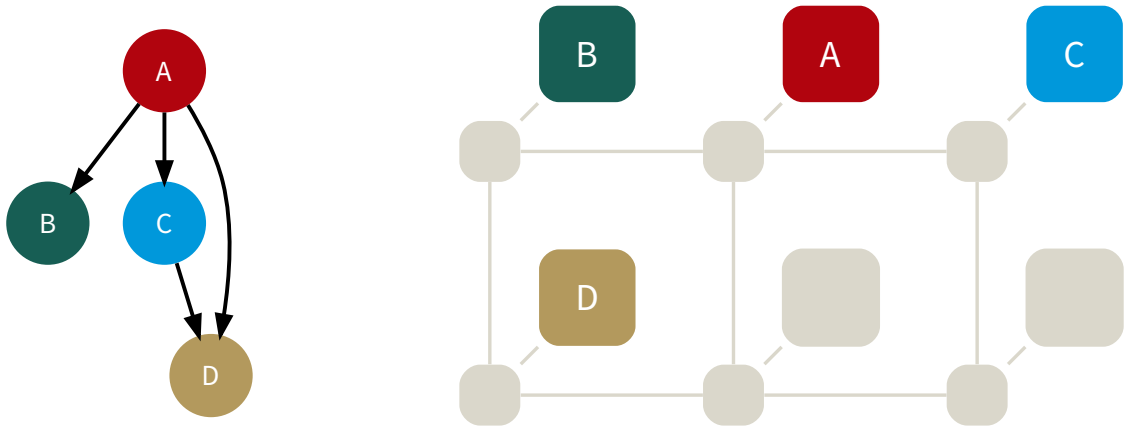
Placement and Routing

- Start with random placement



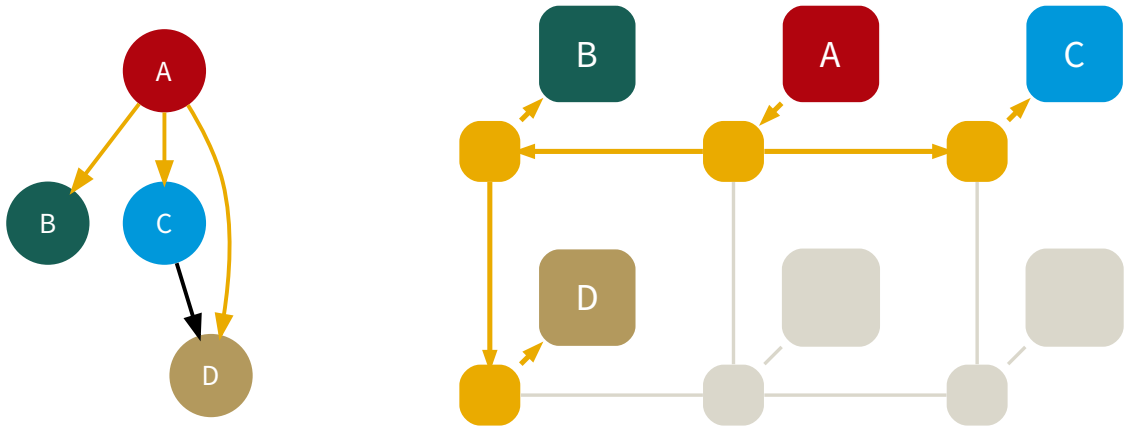
Placement and Routing

- Start with random placement
- Route all links, in order of activation count



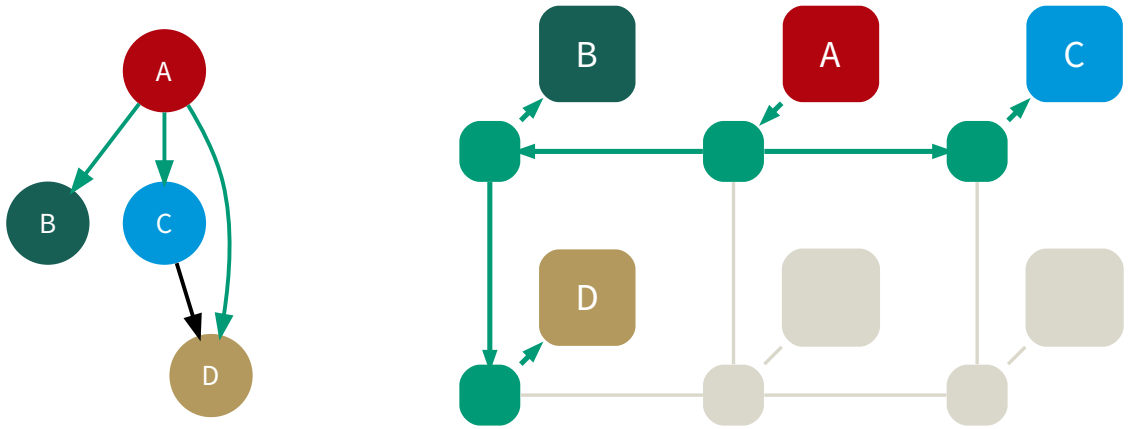
Placement and Routing

- Start with random placement
- Route all links, in order of activation count
  - Build most efficient broadcast tree
  - Guarantee static network placement, if possible



Placement and Routing

- Start with random placement
- Route all links, in order of activation count
  - Build most efficient broadcast tree
  - Guarantee static network placement, if possible
  - Else, map the link to the dynamic network

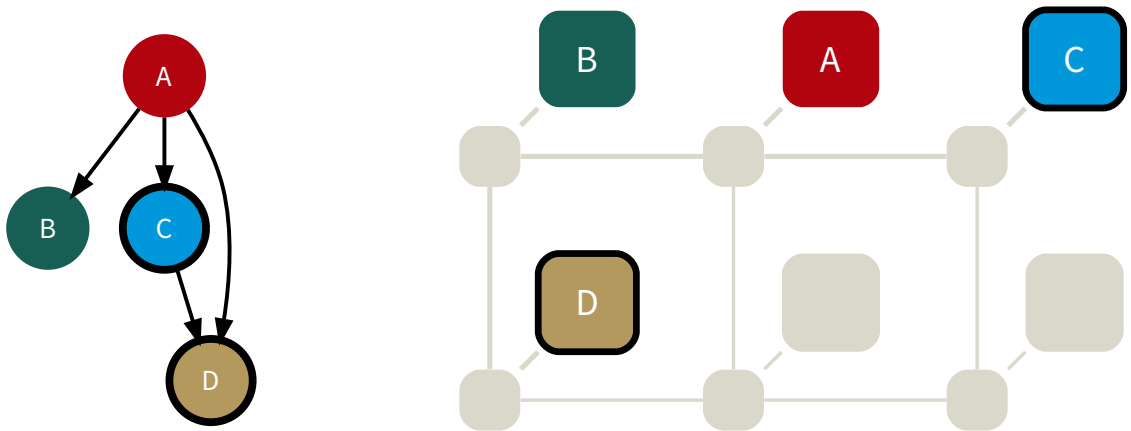


Placement and Routing

- Start with random placement
- Route all links, in order of activation count
- Re-place VBs with the highest routing cost
  - Dynamic network congestion
  - Average route length
  - Maximum route length

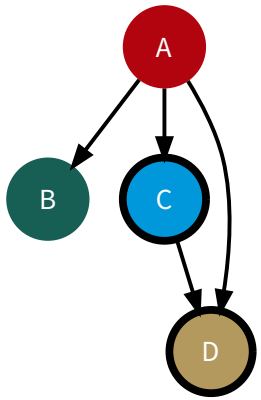
Placement and Routing

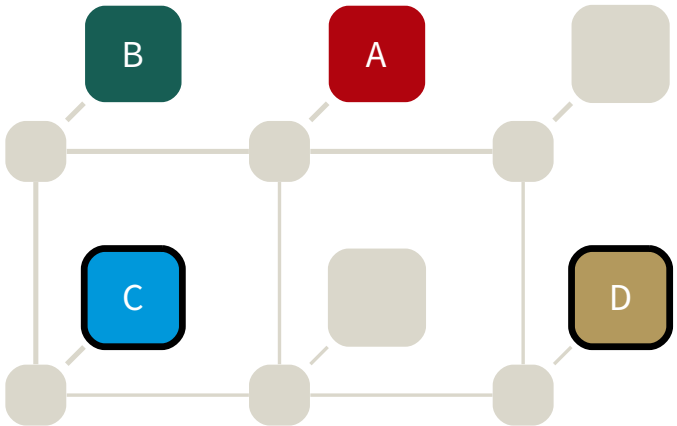
- Start with random placement
- Route all links, in order of activation count
- Re-place VBs with the highest routing cost
  - Dynamic network congestion
  - Average route length
  - Maximum route length



Placement and Routing

- Start with random placement
- Route all links, in order of activation count
- Re-place VBs with the highest routing cost
  - Dynamic network congestion
  - Average route length
  - Maximum route length





Placement and Routing

- Start with random placement
- Route all links, in order of activation count
- Re-place VBs with the highest routing cost
- Repeat routing

Placement and Routing

- Start with random placement
- Route all links, in order of activation count
- Re-place VBs with the highest routing cost
- Repeat routing

Summary

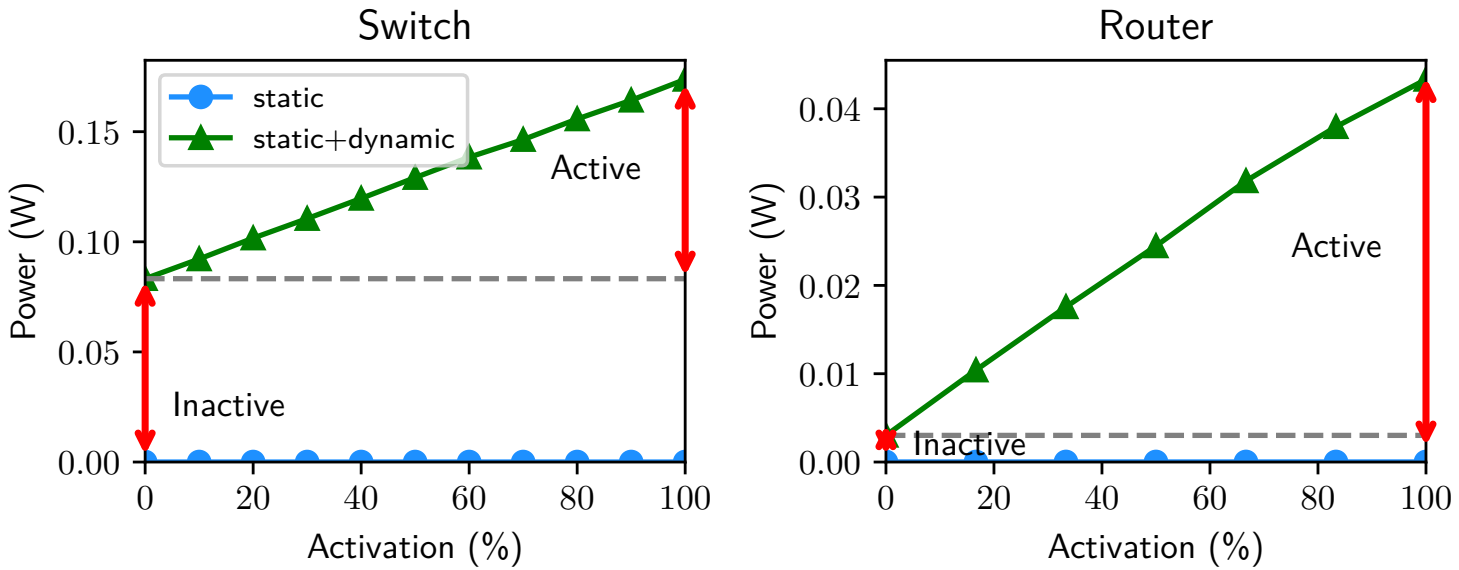
Iteratively reduce routing cost
Map bandwidth-critical links onto the static network

Area and Energy Characterization

- Synthesize switch and router RTL at 28 nm, 1GHz
- Power simulation with Primetime

Area and Energy Characterization

- Synthesize switch and router RTL at 28 nm, 1GHz
- Power simulation with Primetime
- Decompose power into:
  - Inactive (per-cycle)
  - Active (per-bit)



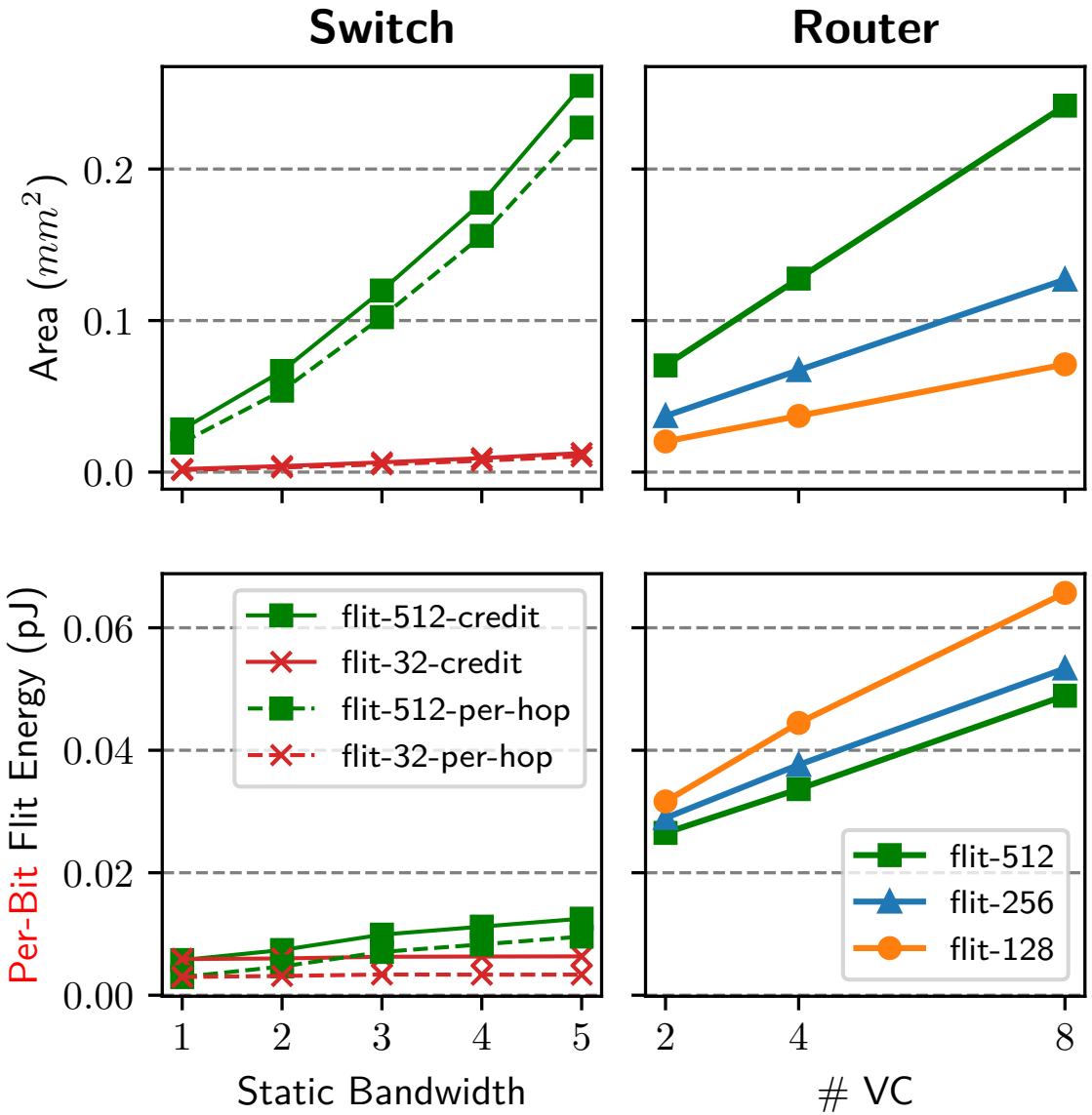
Simulation

- Integrate simulator with DRAMSim and BookSim
- Track transmitted data in switches and routers
- Estimate per-app power with activity traces:

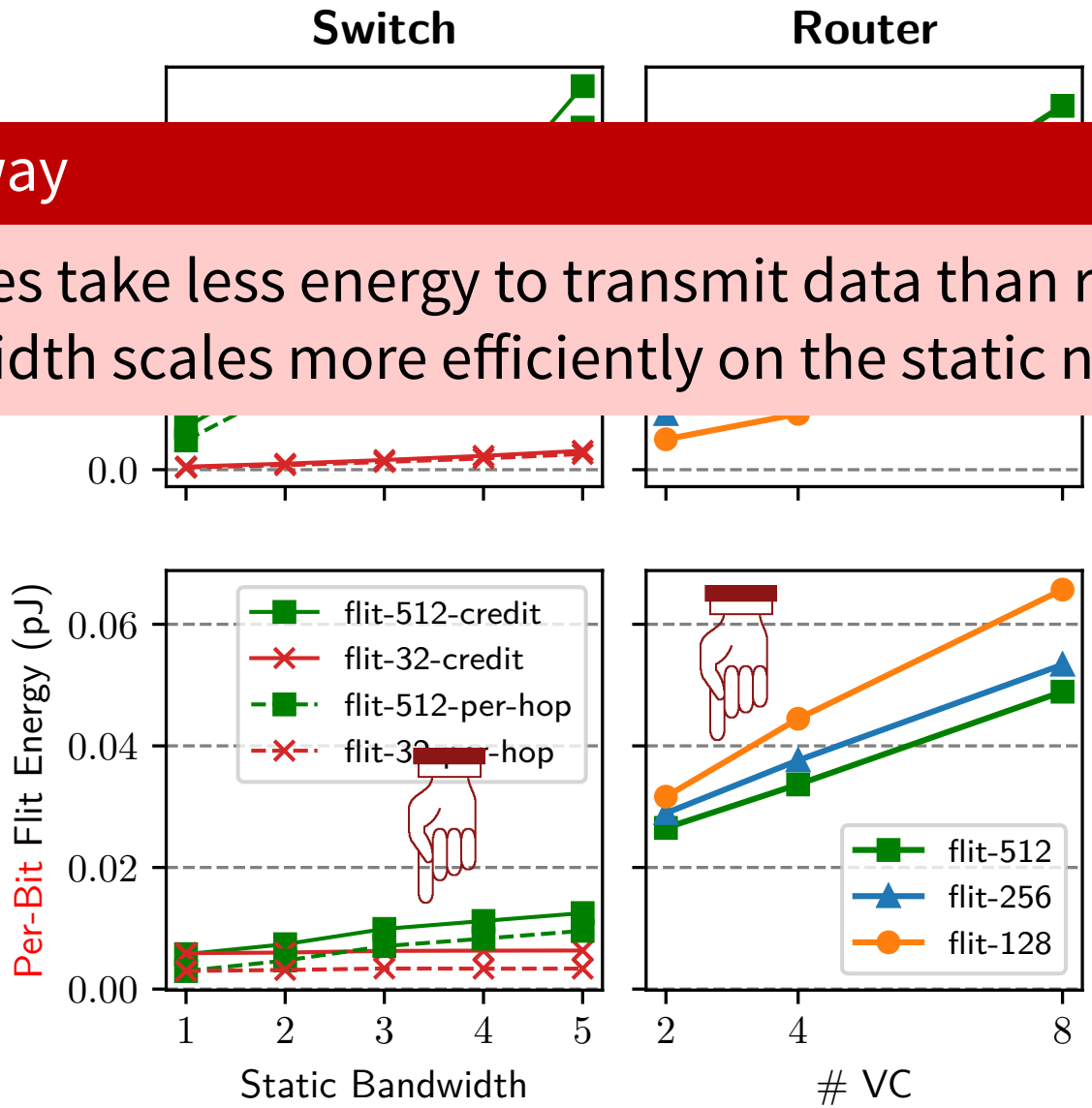
$$E_{\text{net}} = \sum_{\text{allocated}} P_{\text{inactive}} T_{\text{sim}} + E_{\text{flit}} \#\text{flit}$$

Outline

Area and Energy Characterization



Area and Energy Characterization



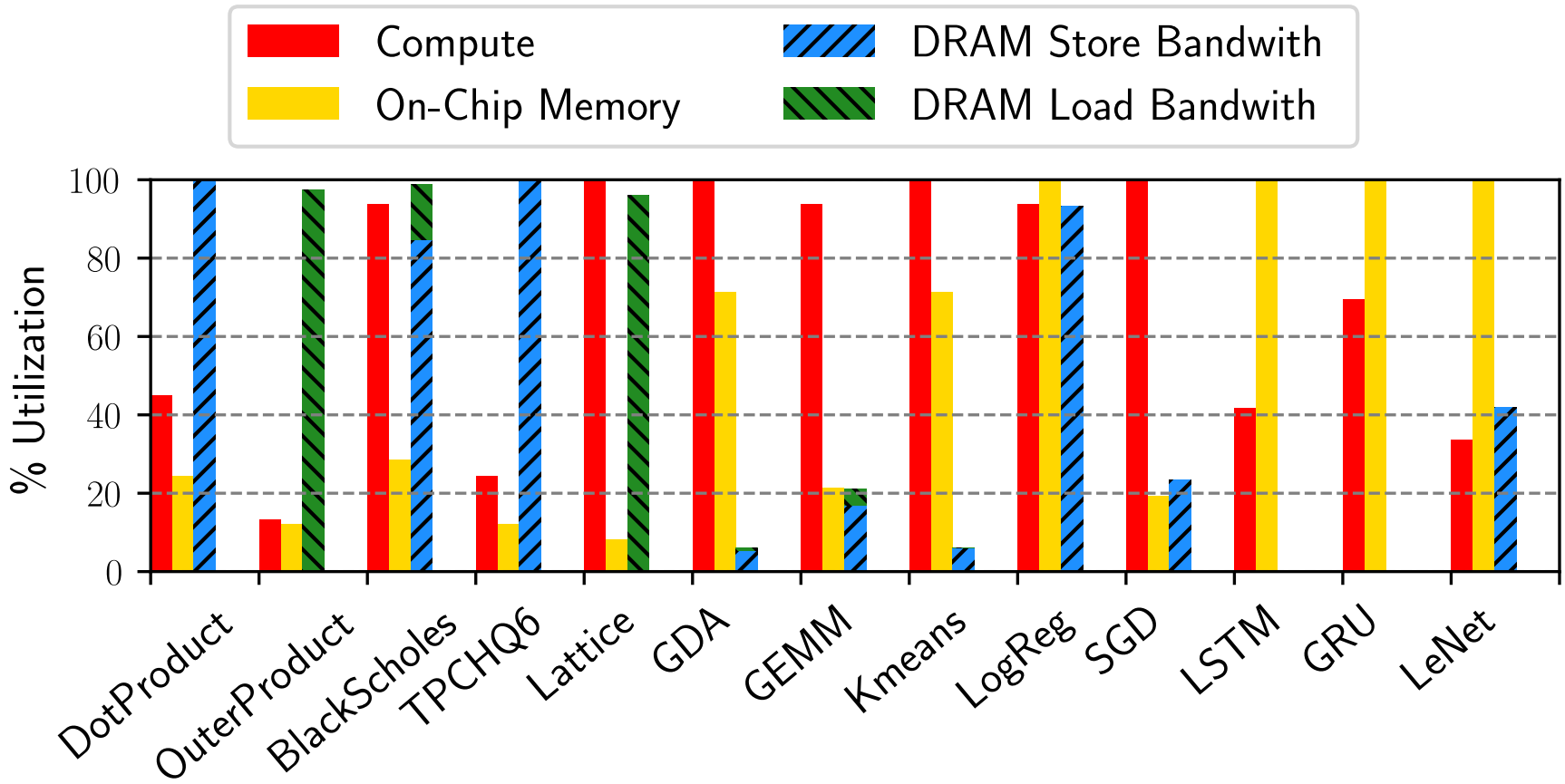
Takeaway

Switches take less energy to transmit data than routers
Bandwidth scales more efficiently on the static network

Benchmarks

Category	Application
Linear Algebra	Dot Product Outer Product Black Scholes GEMM
Database	TPC-H Query 6
Clustering	k-Means Clustering
Inference	Lattice Regression LSTM (RNN) GRU (RNN) LeNet (CNN)
Training	Gaussian Discriminant Analysis Logistic Regression Stochastic Gradient Descent

Benchmark Resource Usage



Evaluated Design Space

- Different network configurations
  - Static: flow control, bandwidth
  - Dynamic: VC count, flit width
  - Hybrid
- Different applications
- Different architectures
  - Pipelined (high throughput)
  - Scheduled (low throughput)

Evaluated Metrics

- Performance **(Perf)**
- Area efficiency **(1/Area)**
- Performance per area **(Perf/Area)**
- Power efficiency **(1/Power)**
- Energy efficiency **(Perf/Watt)**

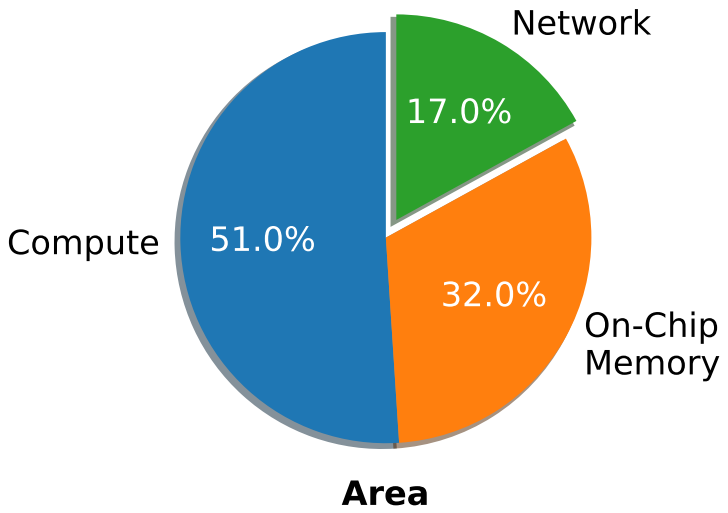
Evaluated Metrics

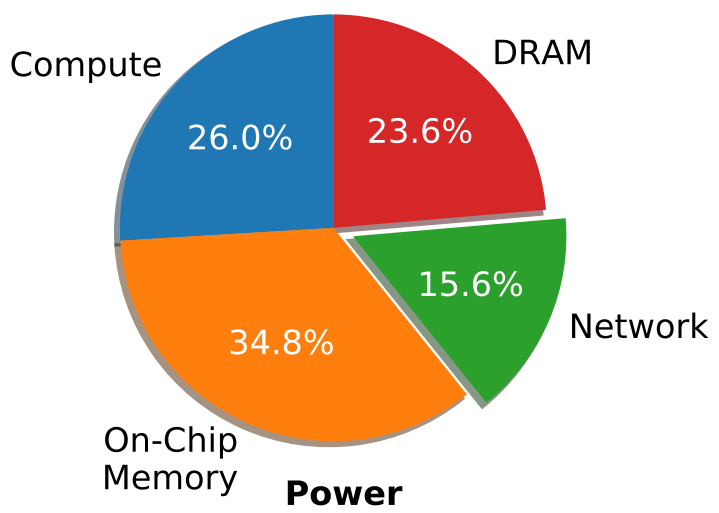
- Performance **(Perf)**
- Area efficiency **(1/Area)**
- Performance per area **(Perf/Area)**
- Power efficiency **(1/Power)**
- Energy efficiency **(Perf/Watt)**

Reported values are the geomean across all applications, normalized to the worst network configuration.

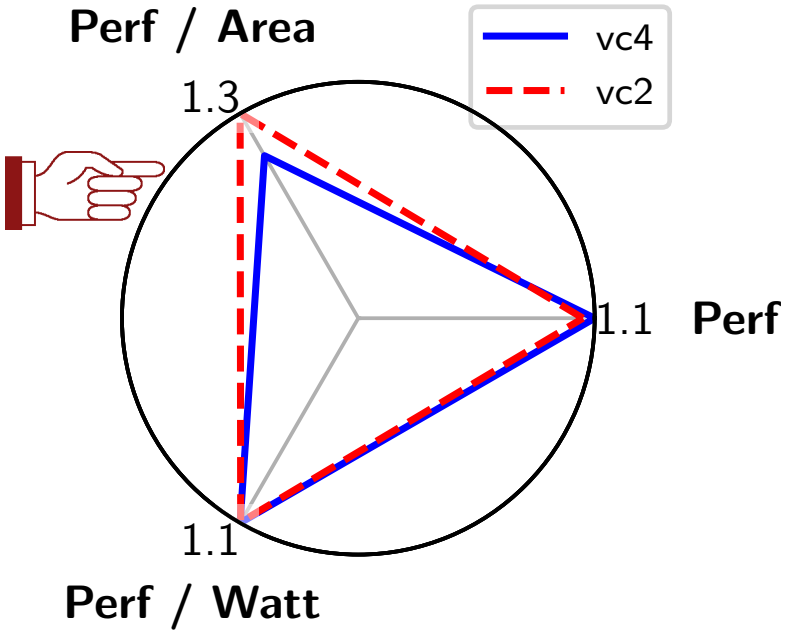
Evaluated Metrics

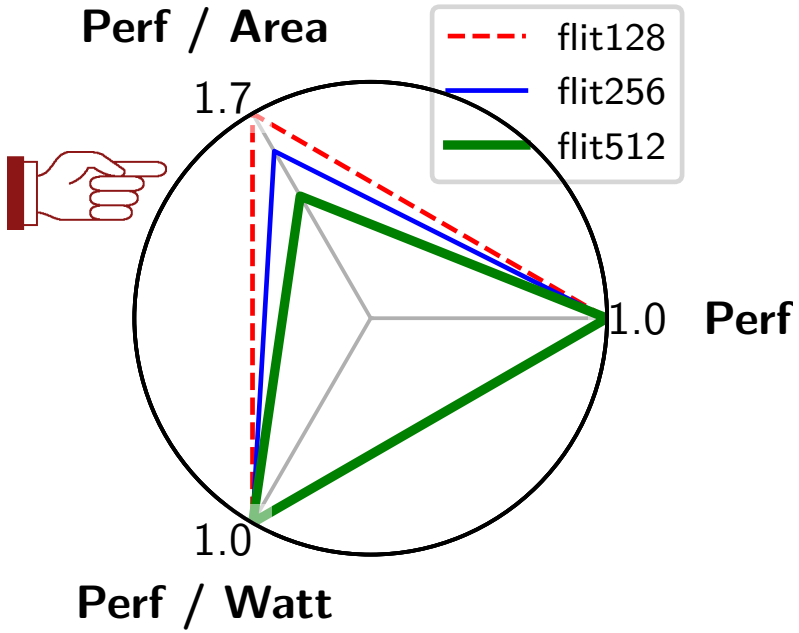
- Performance **(Perf)** ☜
- Area efficiency **(1/Area)**
- Performance per area **(Perf/Area)**
- Power efficiency **(1/Power)**
- Energy efficiency **(Perf/Watt)**





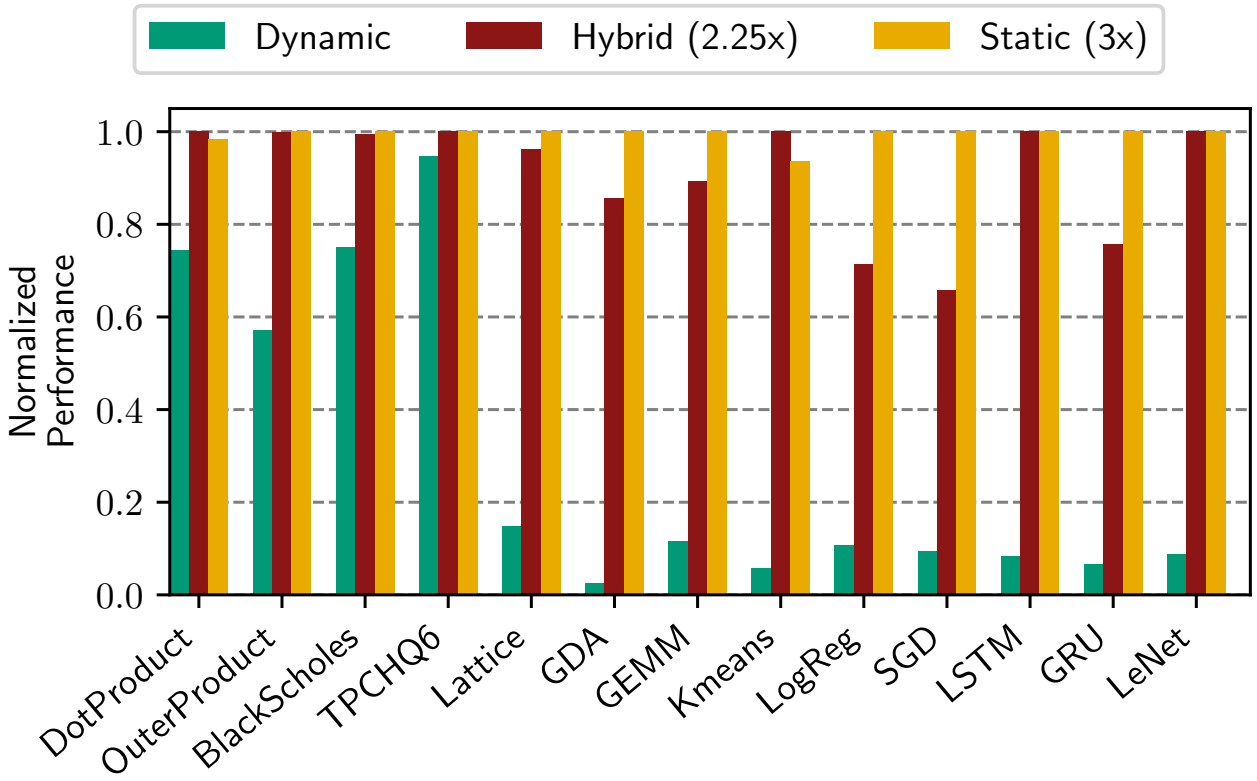
Hybrid Network VCs and Flit Width





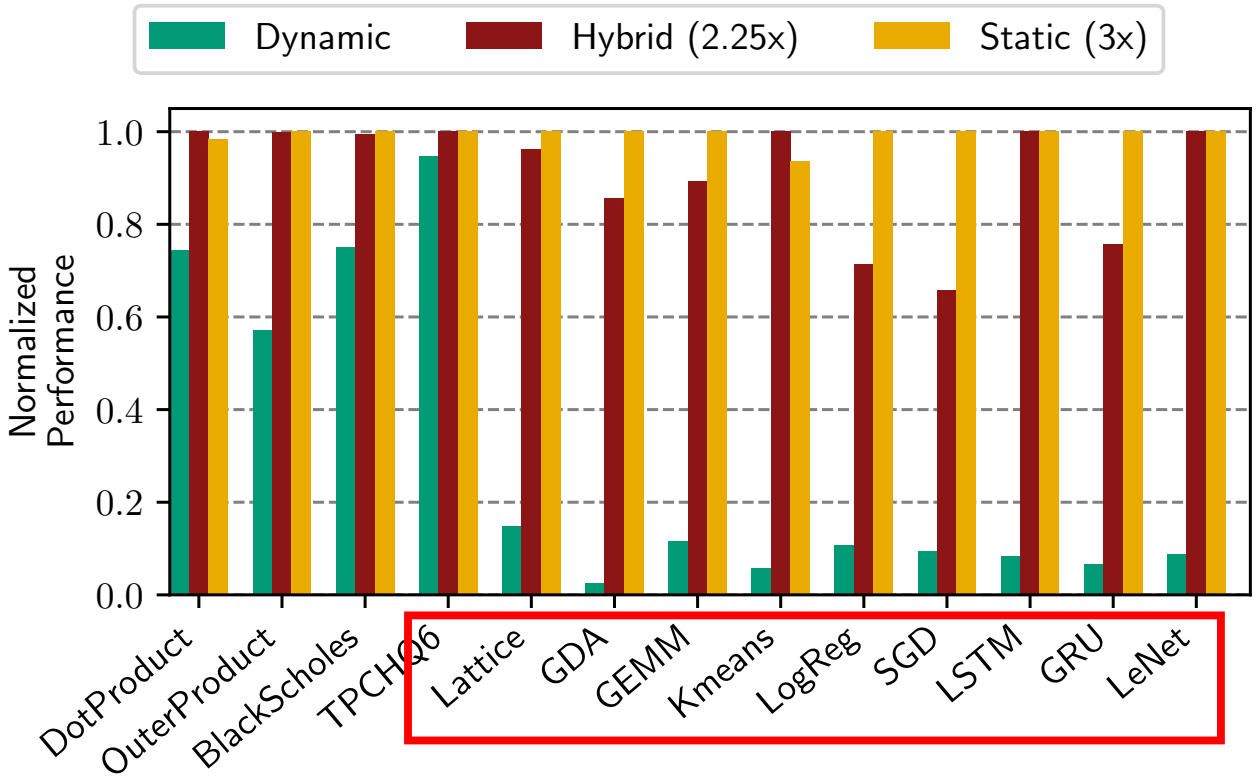
Dynamic network flit width and VC count can be decreased with no performance loss.

Static vs. Dynamic vs. Hybrid



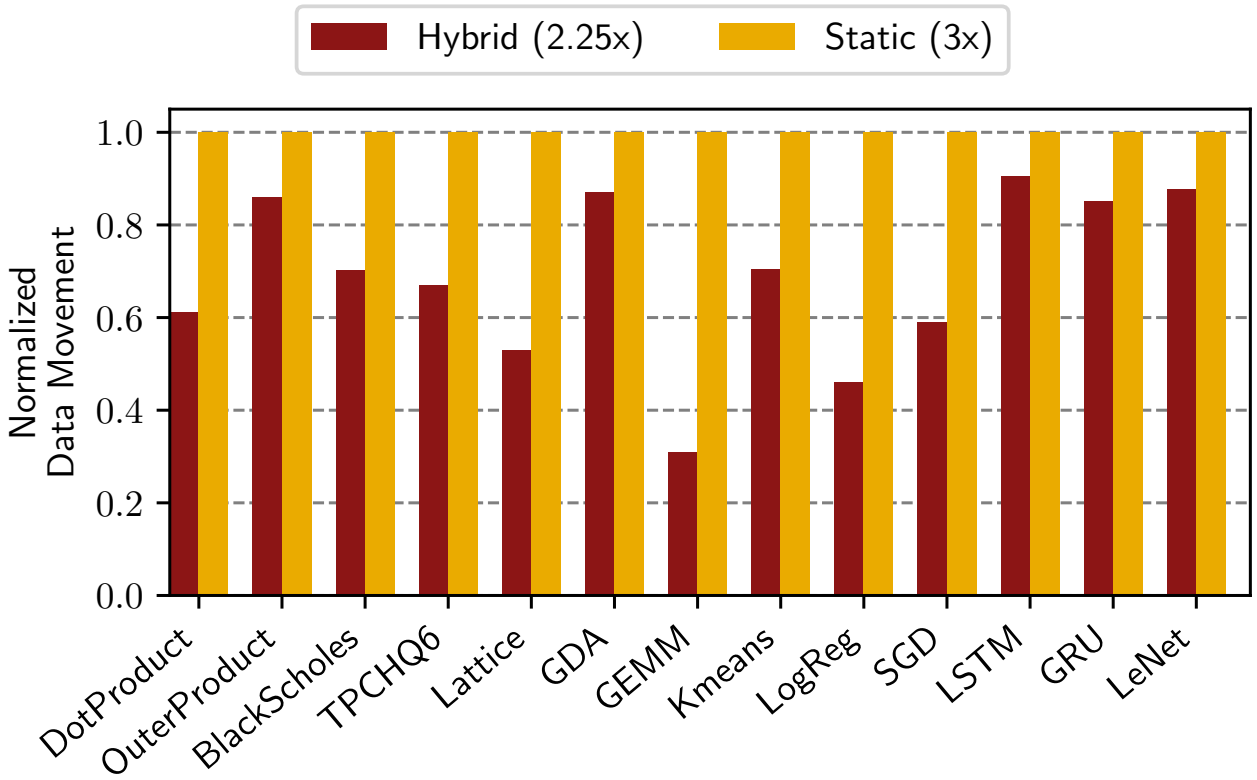
The dynamic network performs poorly on compute-bound applications due to insufficient bandwidth.

Static vs. Dynamic vs. Hybrid



The dynamic network performs poorly on compute-bound applications due to insufficient bandwidth.

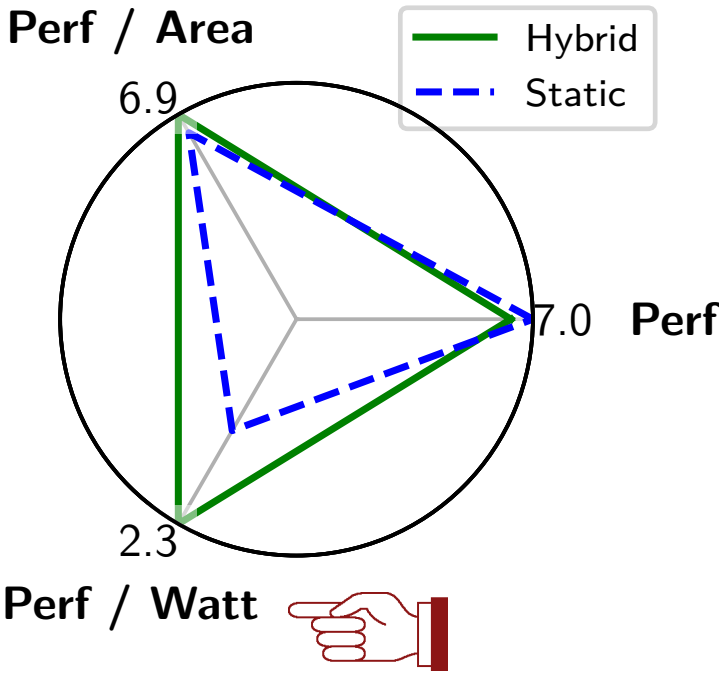
Most Efficient Network Configurations



The hybrid network reduces data movement by using a dynamic network as an escape path.

Most Efficient Network Configurations

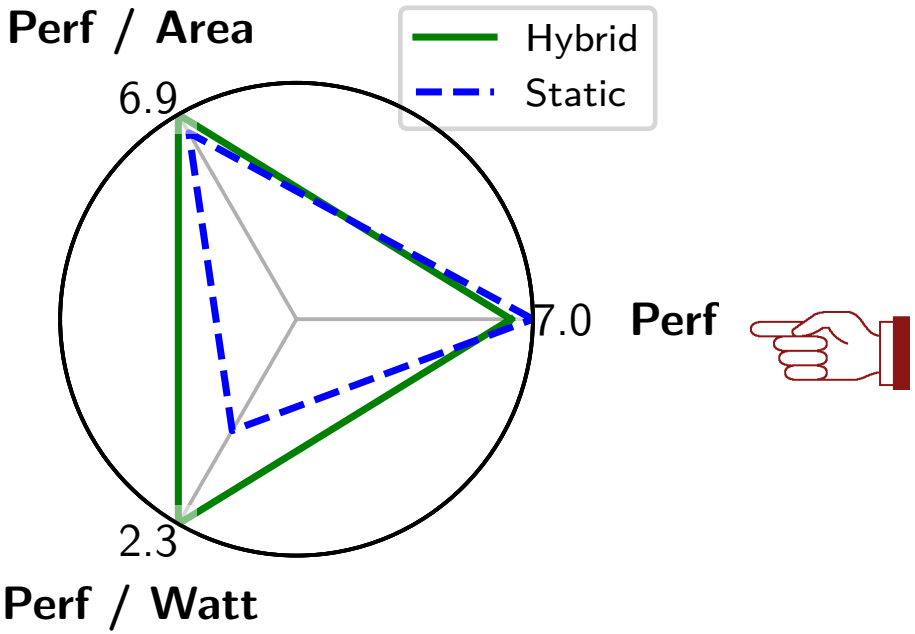
Pipelined Architecture



A hybrid network improves energy efficiency by ***1.8x*** with performance similar to a static network.

Most Efficient Network Configurations

Pipelined Architecture



A hybrid network improves energy efficiency by ***1.8x*** with performance similar to a static network.
Performance varies up to ***7x*** between the best and worst network configurations.

Conclusion

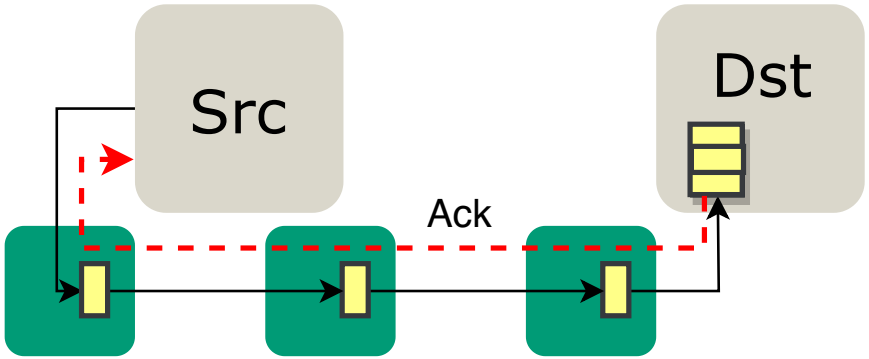
- Network performance correlates strongly with ***bandwidth*** for spatial accelerators
- Bandwidth scales more efficiently on a static network
- A hybrid (large static, small dynamic) network:
  - Eliminates place and route failure
  - Improves ***perf/watt***

Conclusion

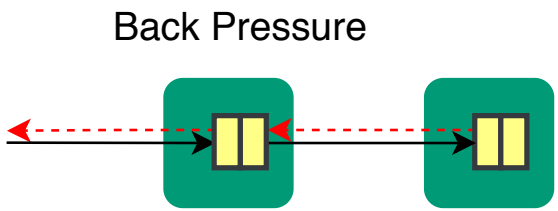
- Network performance correlates strongly with ***bandwidth*** for spatial accelerators
- Bandwidth scales more efficiently on a static network
- A hybrid (large static, small dynamic) network:
  - Eliminates place and route failure
  - Improves ***perf/watt***

Thank You!

Static Network: Flow Control

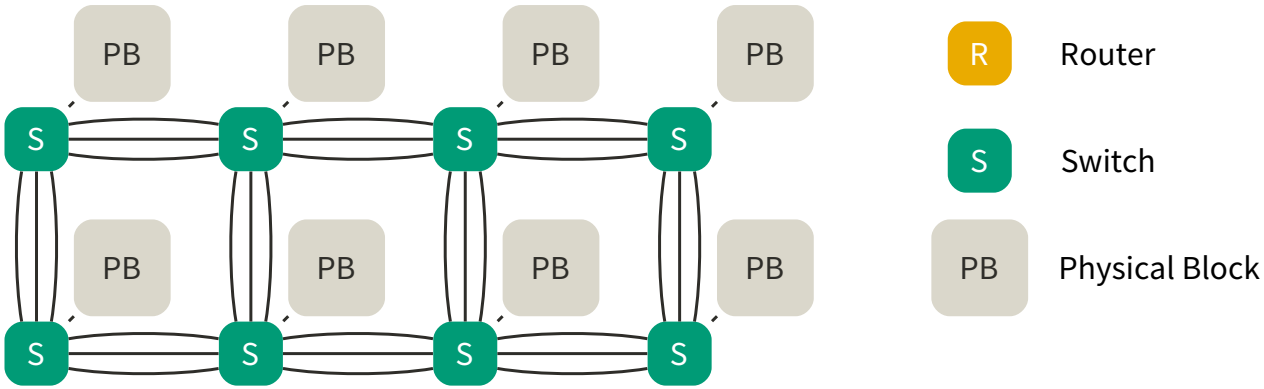


End-to-end Flow Control



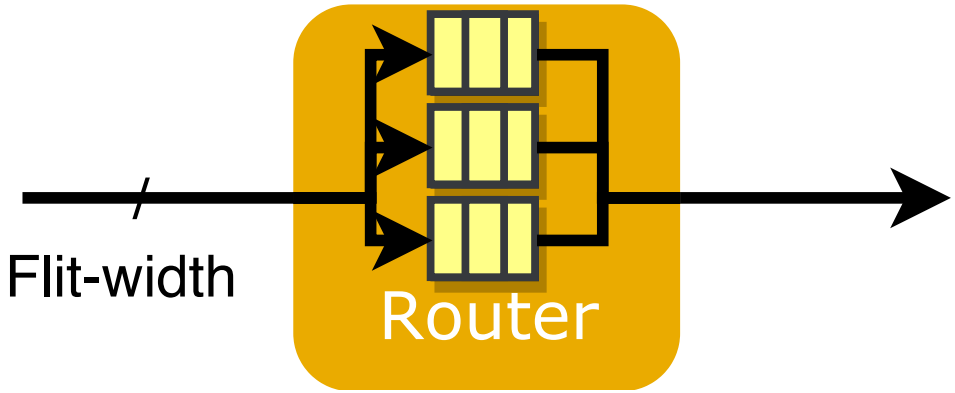
Per-hop Flow Control

Static Network: Bandwidth



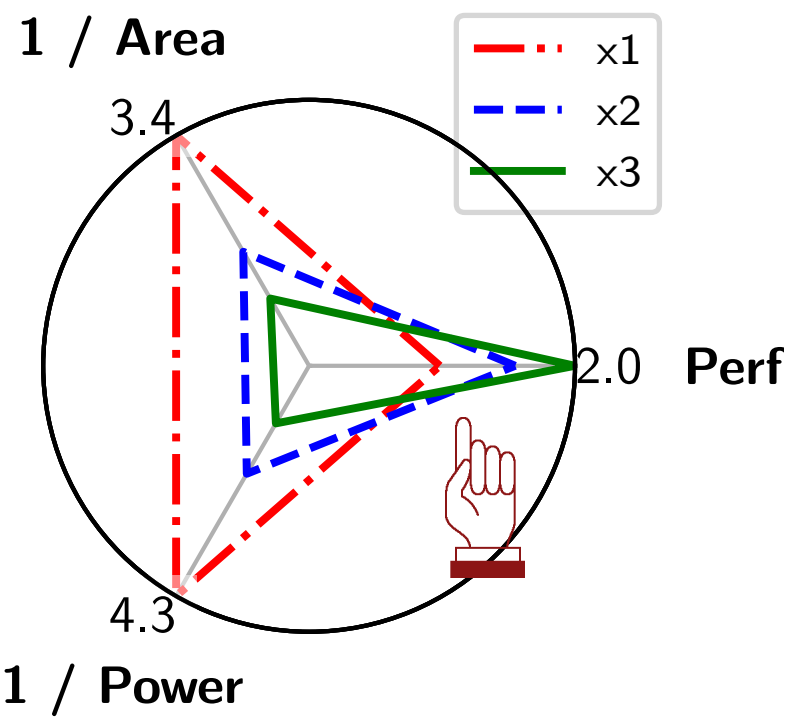
We vary the number of links between switches.

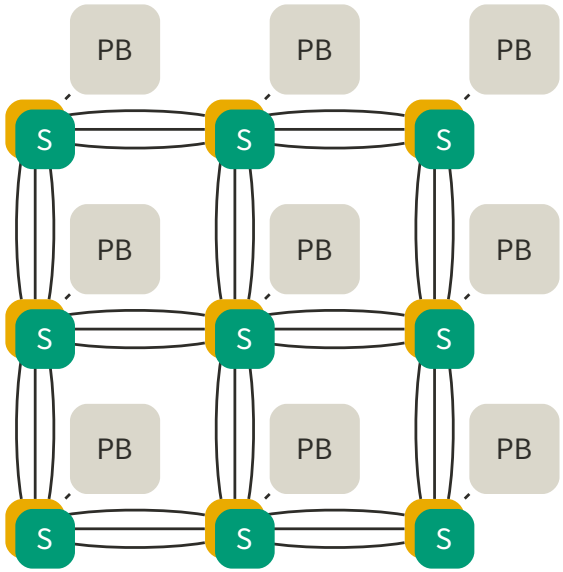
Dynamic Network



We vary the number of Virtual Channels (VCs) and flit width.

Static Network Bandwidth



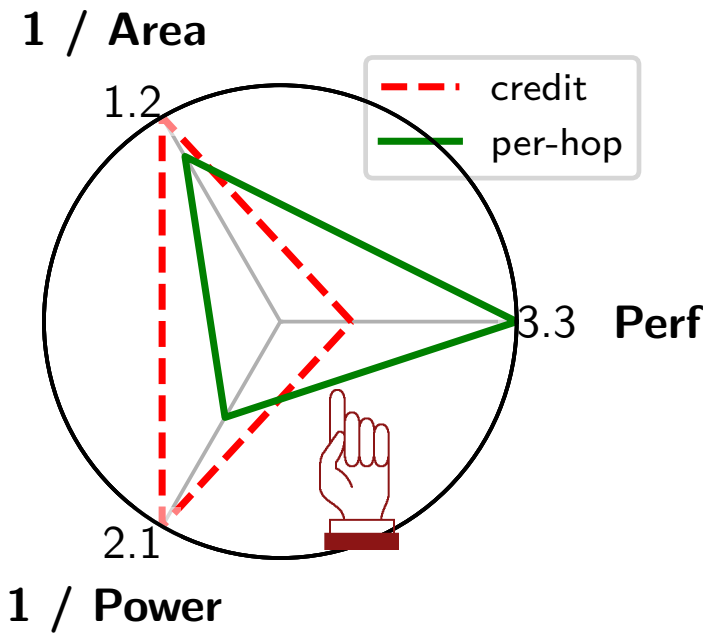


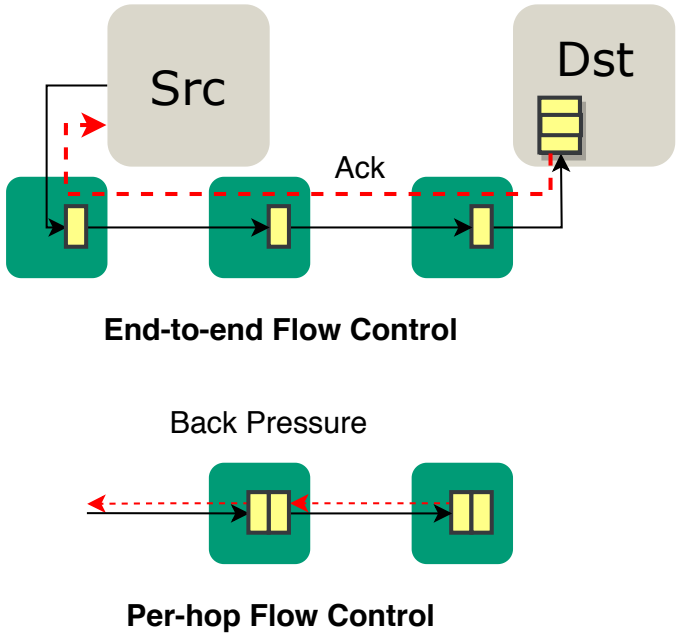
3x static network bandwidth

Bandwidth strongly impacts accelerator performance.

Static Network Flow Control

Credit-Based vs. Per-Hop

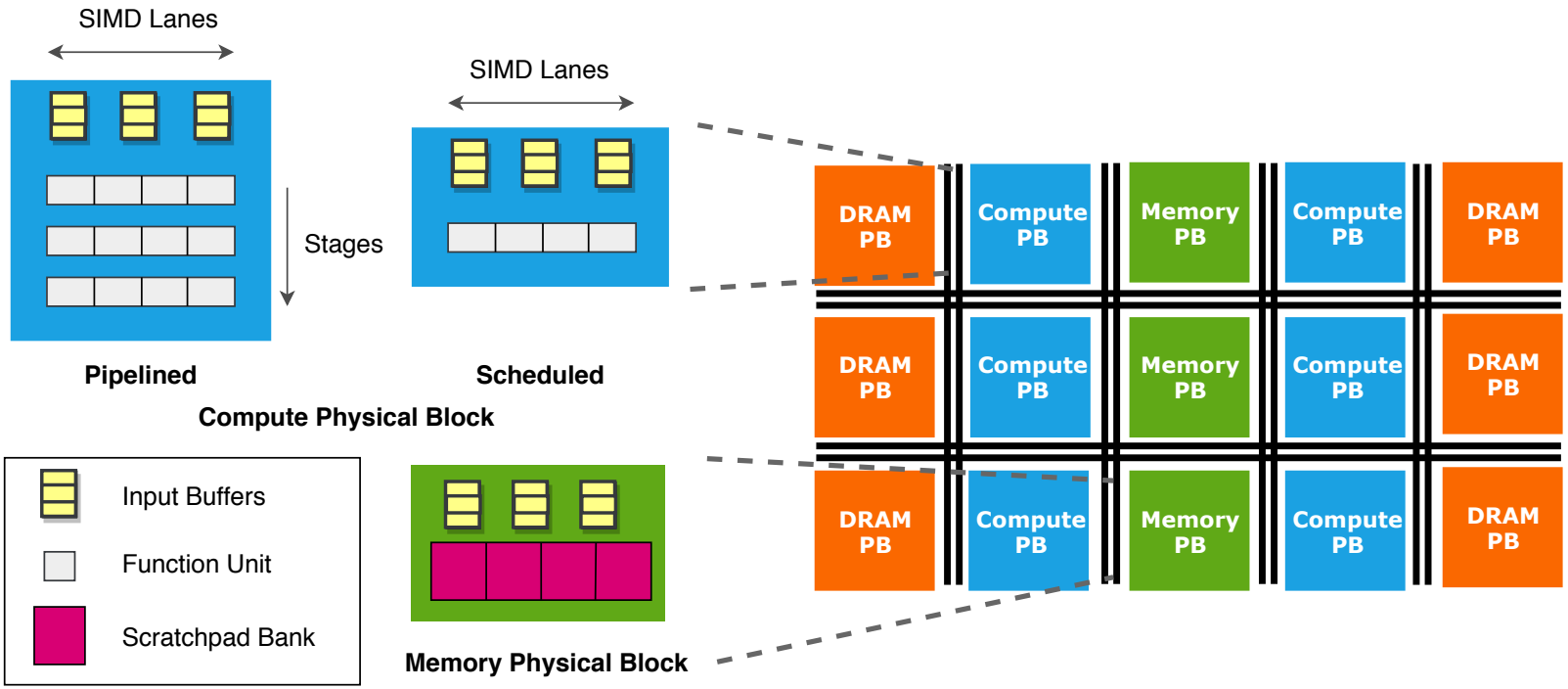




Credit-based flow control has **3x** lower performance.

Accelerator Model

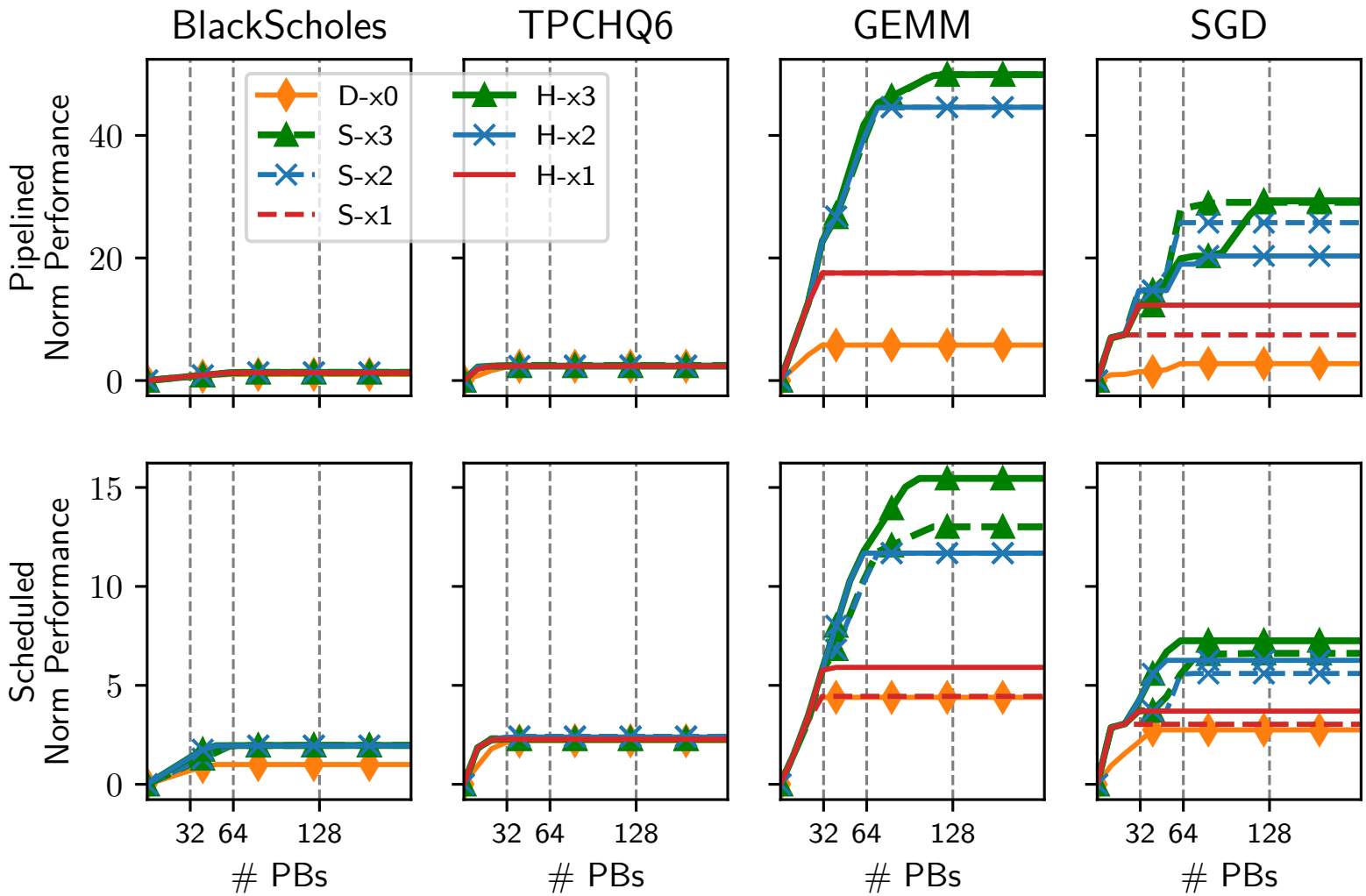
- Pool of compute and memory resource
- Compute:
  - SIMD pipeline, or
  - Vector processor with a small instruction window



Statically Routed Dynamic Network

- Streaming protocol requires in-order transmission
  - Can’t use adaptive or oblivious routing
  - Can’t drop packets
- Routes are looked up in a table at runtime
  - Route to multiple outputs for efficient broadcast links

Performance Scaling



Key Design Challenges