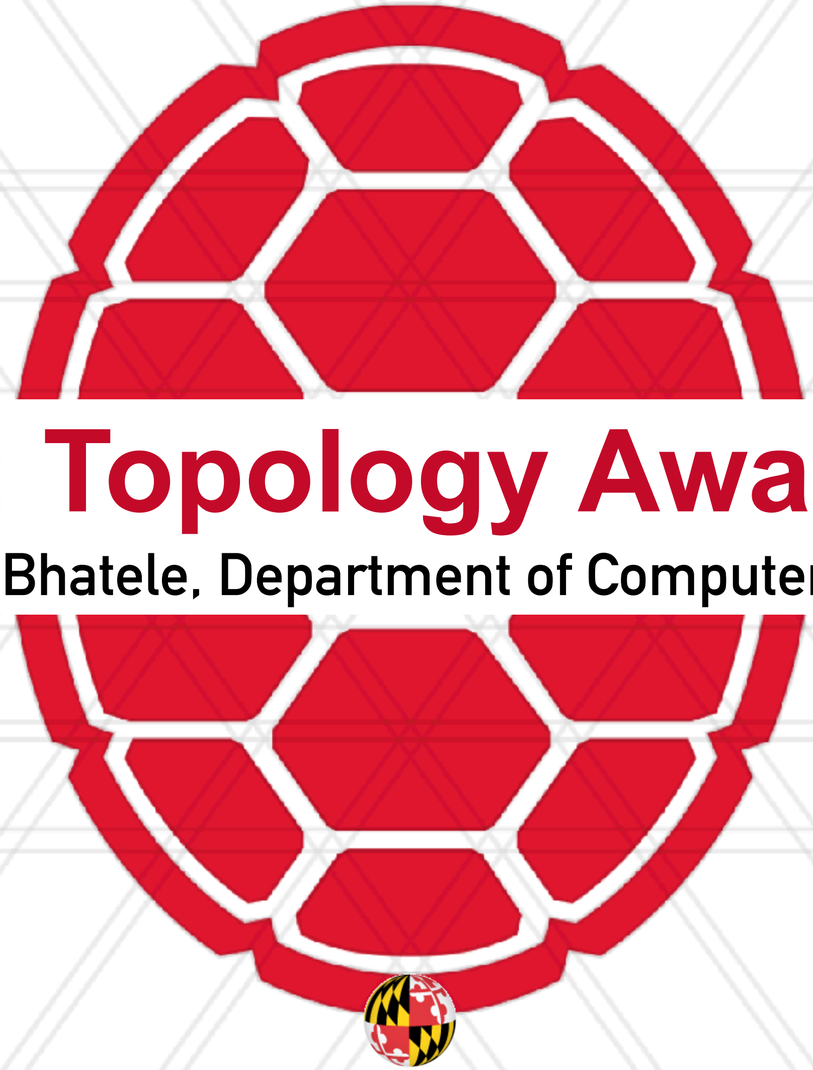
High Performance Computing Systems (CMSC714)



Lecture 19: Topology Aware Mapping

Abhinav Bhatele, Department of Computer Science

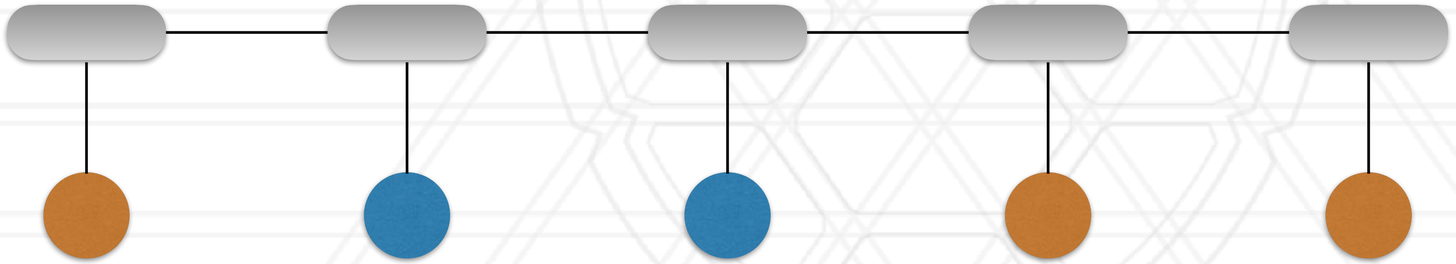
UNIVERSITY OF MARYLAND

Summary of last lecture

- Most HPC systems use a job/batch scheduler
- Scheduler decides what jobs to run next and what resources to allocate
  - Backfilling to use idle nodes and improve utilization
- Different quality of service metrics to evaluate schedulers

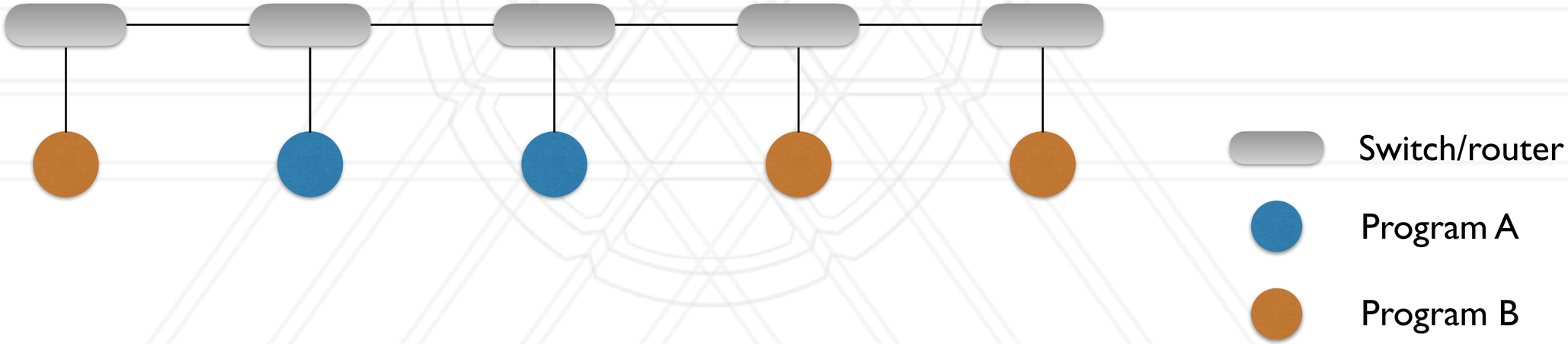
Congestion due to network sharing

- Sharing refers to network flows of different programs using the same hardware resources: links, switches
- When multiple programs communicate on the network, they all suffer from congestion on shared links



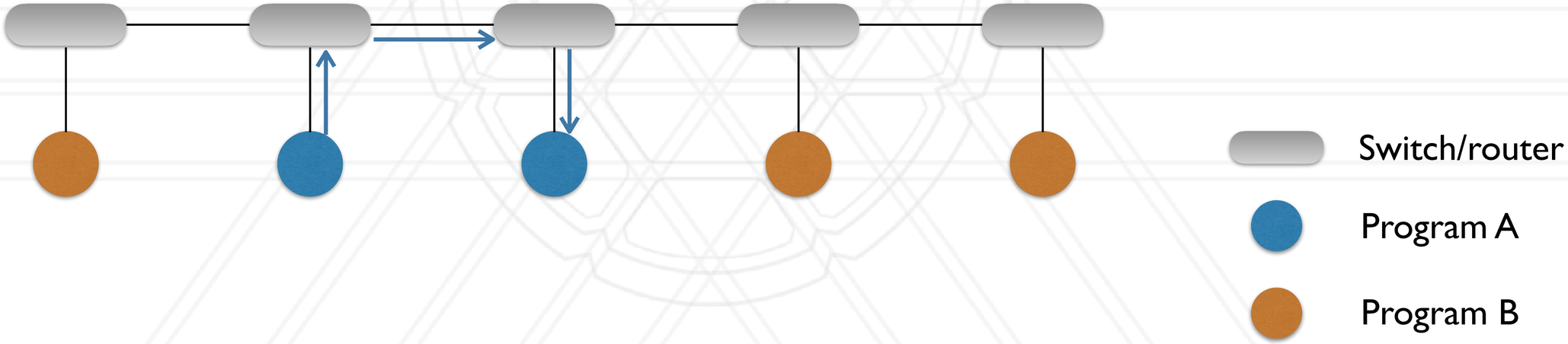
Congestion due to network sharing

- Sharing refers to network flows of different programs using the same hardware resources: links, switches
- When multiple programs communicate on the network, they all suffer from congestion on shared links



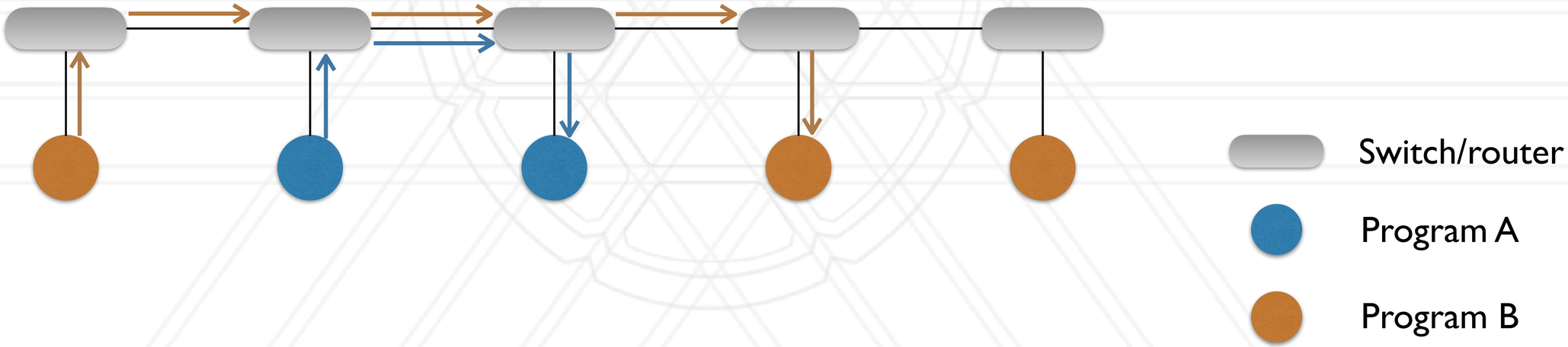
Congestion due to network sharing

- Sharing refers to network flows of different programs using the same hardware resources: links, switches
- When multiple programs communicate on the network, they all suffer from congestion on shared links



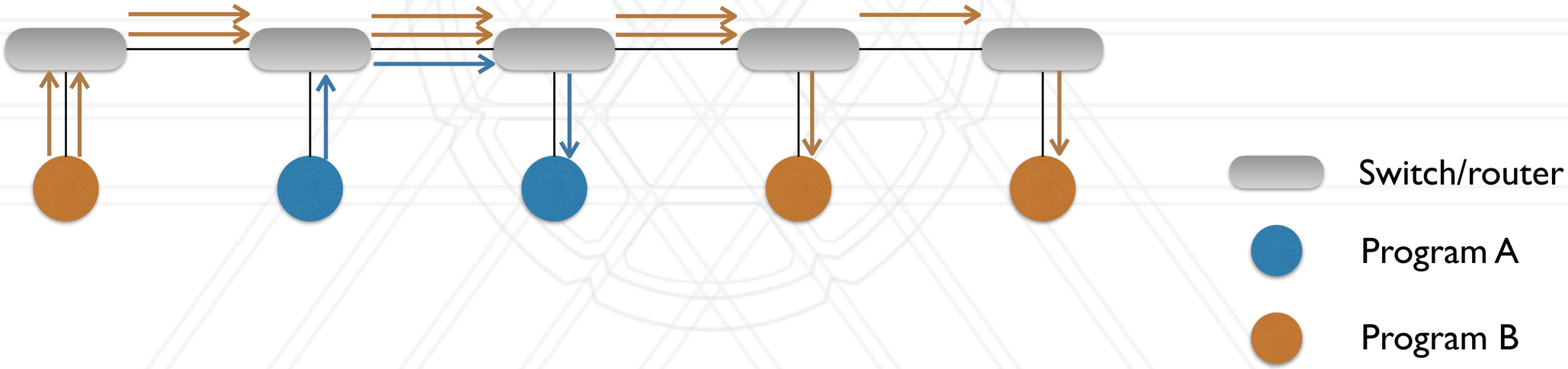
Congestion due to network sharing

- Sharing refers to network flows of different programs using the same hardware resources: links, switches
- When multiple programs communicate on the network, they all suffer from congestion on shared links



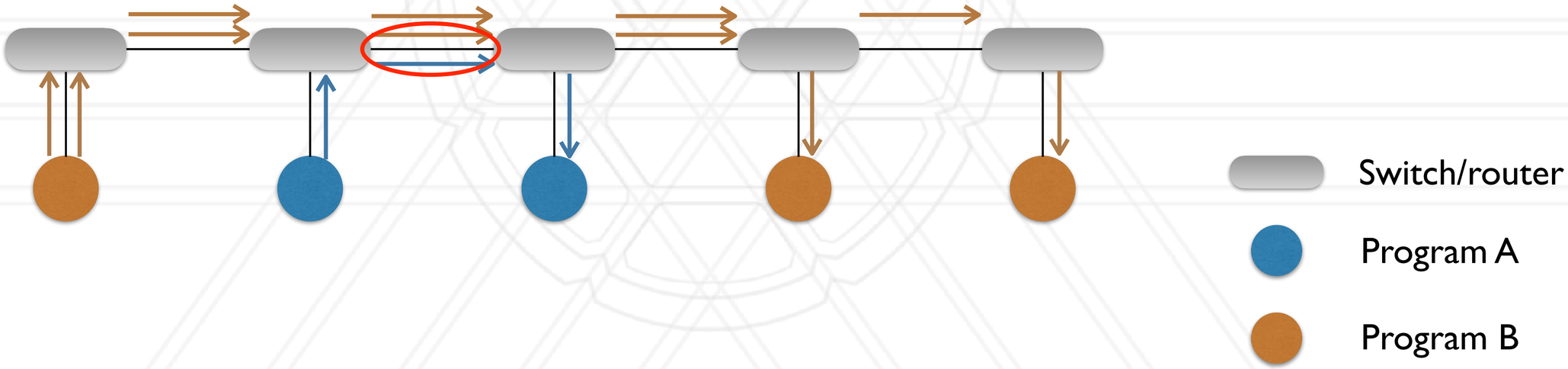
Congestion due to network sharing

- Sharing refers to network flows of different programs using the same hardware resources: links, switches
- When multiple programs communicate on the network, they all suffer from congestion on shared links



Congestion due to network sharing

- Sharing refers to network flows of different programs using the same hardware resources: links, switches
- When multiple programs communicate on the network, they all suffer from congestion on shared links



Communication is a bottleneck at scale

- GPU-based platforms have a large number of flop/s per node
  - Network bandwidths do not increase proportionally
- More energy is spent on sending data across the network

	Time (ns)	Energy spent (pJ)
Floating point operation	< 0.25	30-45
Time to access DRAM	50	128
Get data from another node	> 1000	128-576

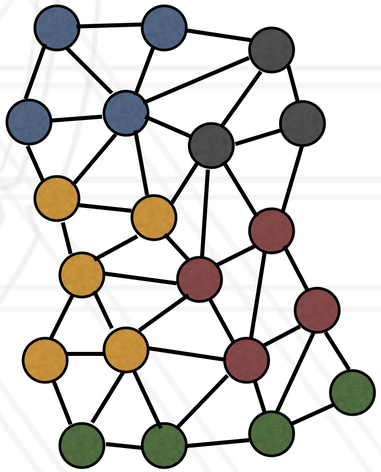
P. Kogge et al., Exascale computing study: Technology challenges in achieving exascale systems, *Technical Report*, 2008.

Different approaches to mitigate congestion

- At the system level
  - Network topology aware job scheduler — attempts to assign compact allocation to jobs
  - Congestion-mitigating routing algorithms
- At the individual job level
  - Users can try to optimize the mapping of MPI processes to allocated nodes

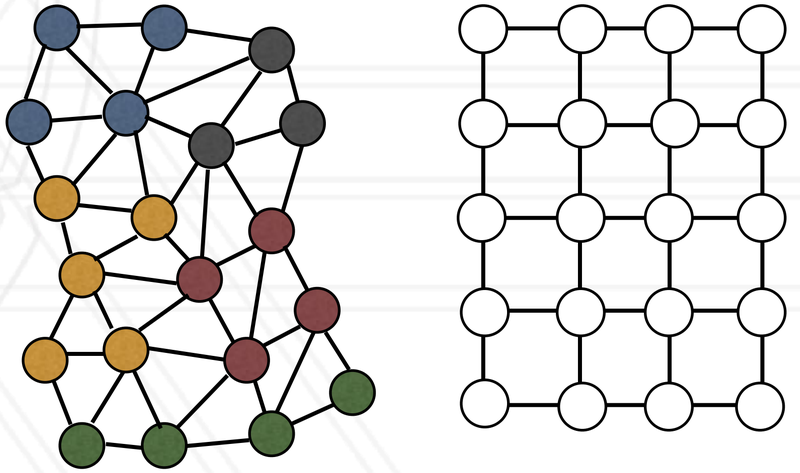
Topology aware task mapping

- Also referred to as task placement or node mapping
- Given an allocation, decide which MPI processes are placed on which physical nodes/ cores
  - In case of task-based models, map finer-grained tasks to cores
- Goal:
  - Minimize communication volume on the network
  - Optimize "unavoidable" communication on the network



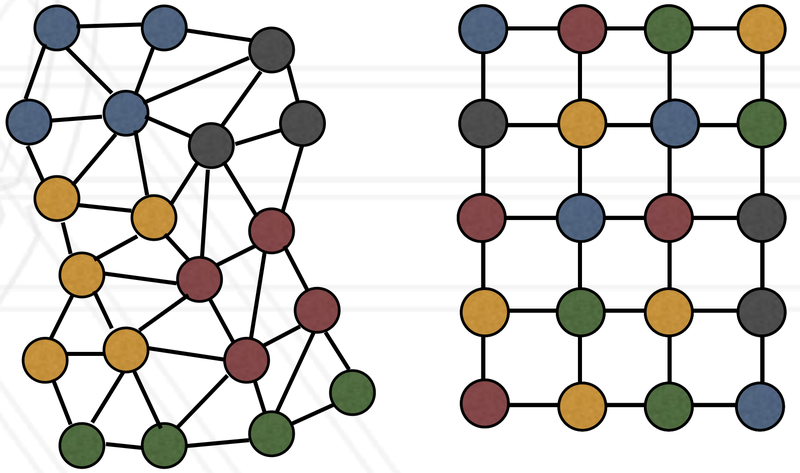
Topology aware task mapping

- Also referred to as task placement or node mapping
- Given an allocation, decide which MPI processes are placed on which physical nodes/ cores
  - In case of task-based models, map finer-grained tasks to cores
- Goal:
  - Minimize communication volume on the network
  - Optimize "unavoidable" communication on the network



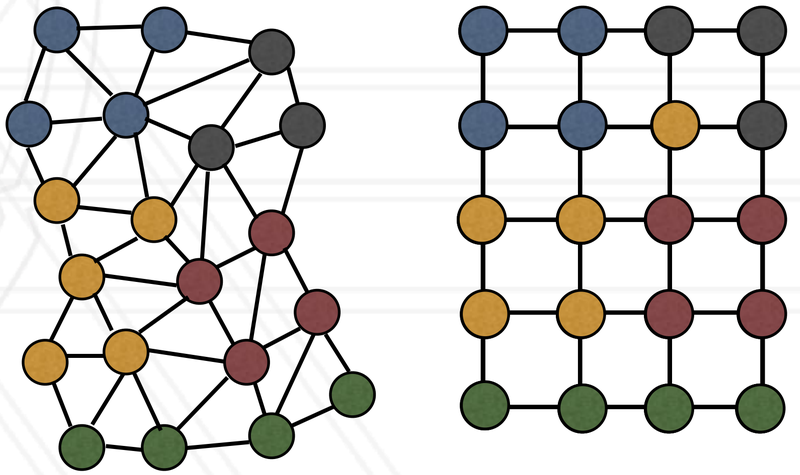
Topology aware task mapping

- Also referred to as task placement or node mapping
- Given an allocation, decide which MPI processes are placed on which physical nodes/ cores
  - In case of task-based models, map finer-grained tasks to cores
- Goal:
  - Minimize communication volume on the network
  - Optimize "unavoidable" communication on the network



Topology aware task mapping

- Also referred to as task placement or node mapping
- Given an allocation, decide which MPI processes are placed on which physical nodes/ cores
  - In case of task-based models, map finer-grained tasks to cores
- Goal:
  - Minimize communication volume on the network
  - Optimize "unavoidable" communication on the network



Graph embedding problem

- Inputs: Application communication graph, network topology graph (of one's job allocation)
- Output: Process-to-node/core mapping
- Most mapping algorithms do not consider that communication patterns might evolve over time

Metrics to evaluate mapping

- Hop-count

$$\sum_{(i,j)} H(i,j)$$

- Hop-bytes

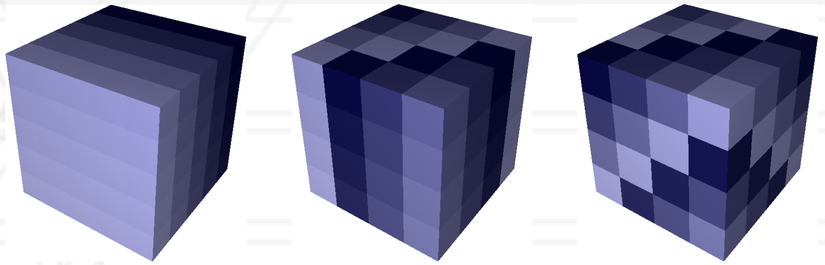
$$\sum_{(i,j)} C(i,j) \times H(i,j)$$

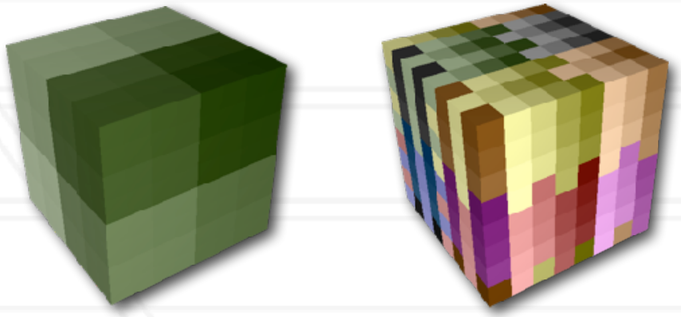
Different techniques

- Heuristics-based
  - Recursive bi-partitioning
  - Random pairwise swaps
- Physical optimization problems
  - Simulated annealing
  - Genetic algorithms

Rubik: Python tool for mapping

- Define various operations on prisms
  - Partitioning or blocking
  - Permuting operations

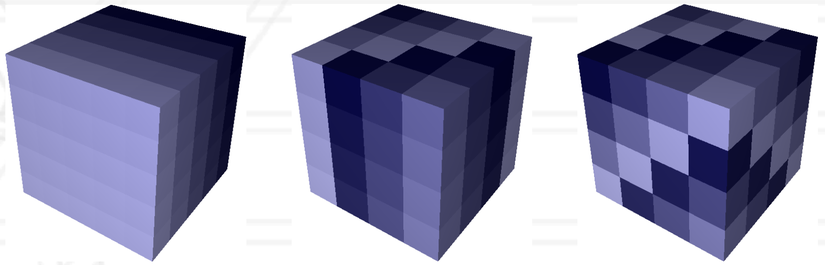


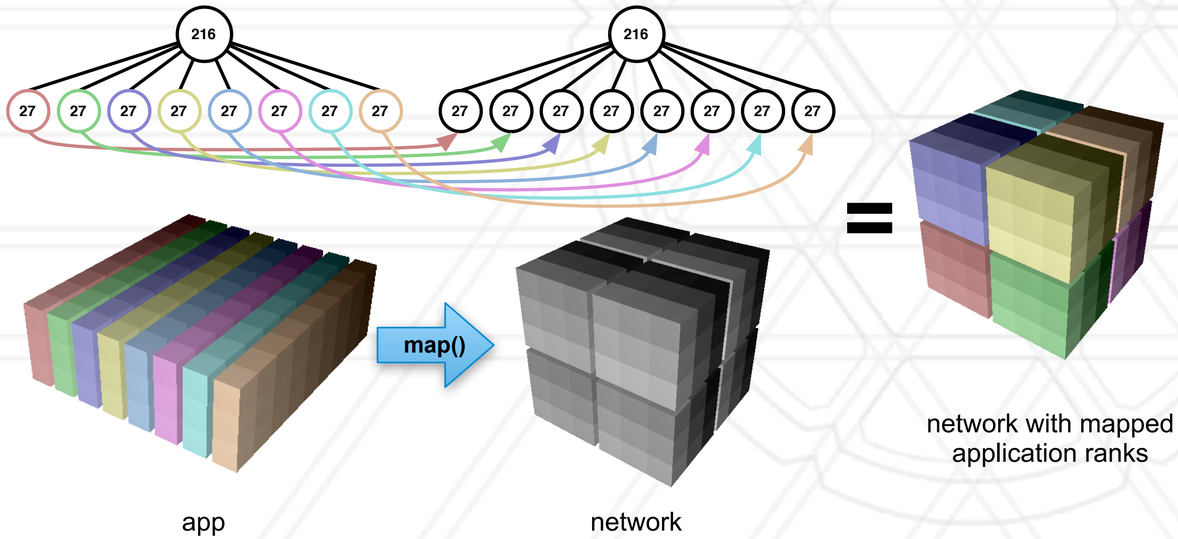


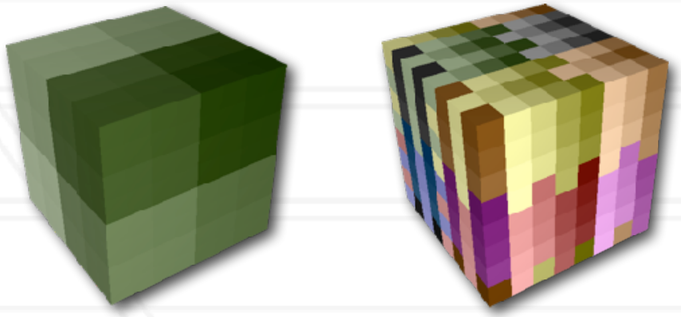
https://github.com/LLNL/rubik

Rubik: Python tool for mapping

- Define various operations on prisms
  - Partitioning or blocking
  - Permuting operations



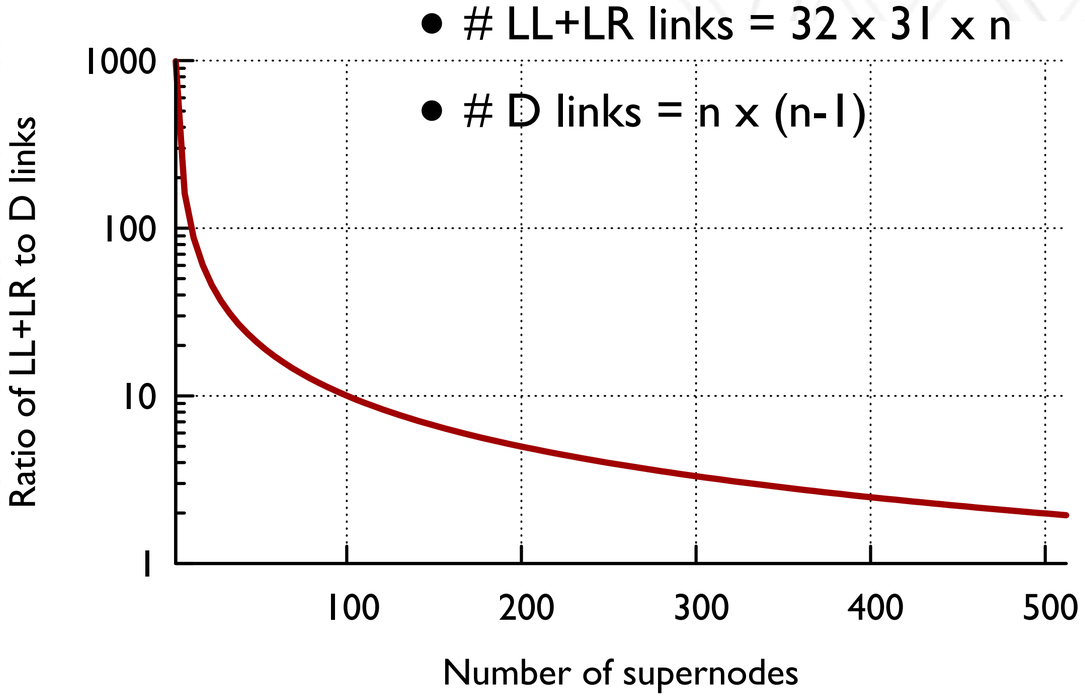


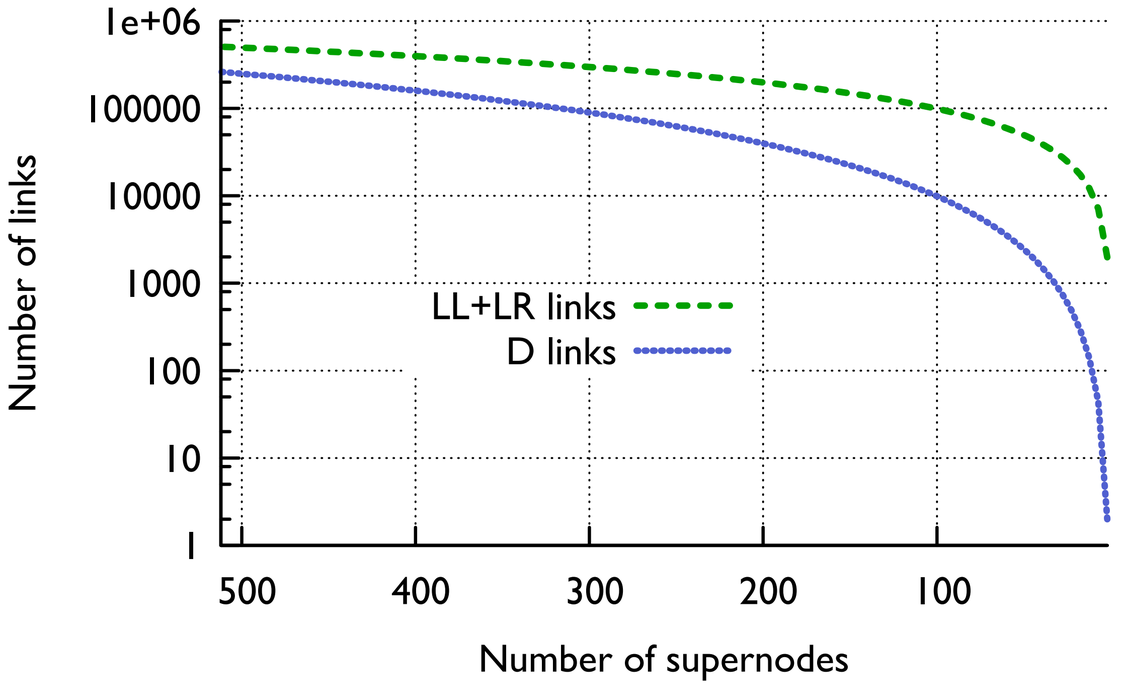


https://github.com/LLNL/rubik

Global link bottleneck in dragonfly systems

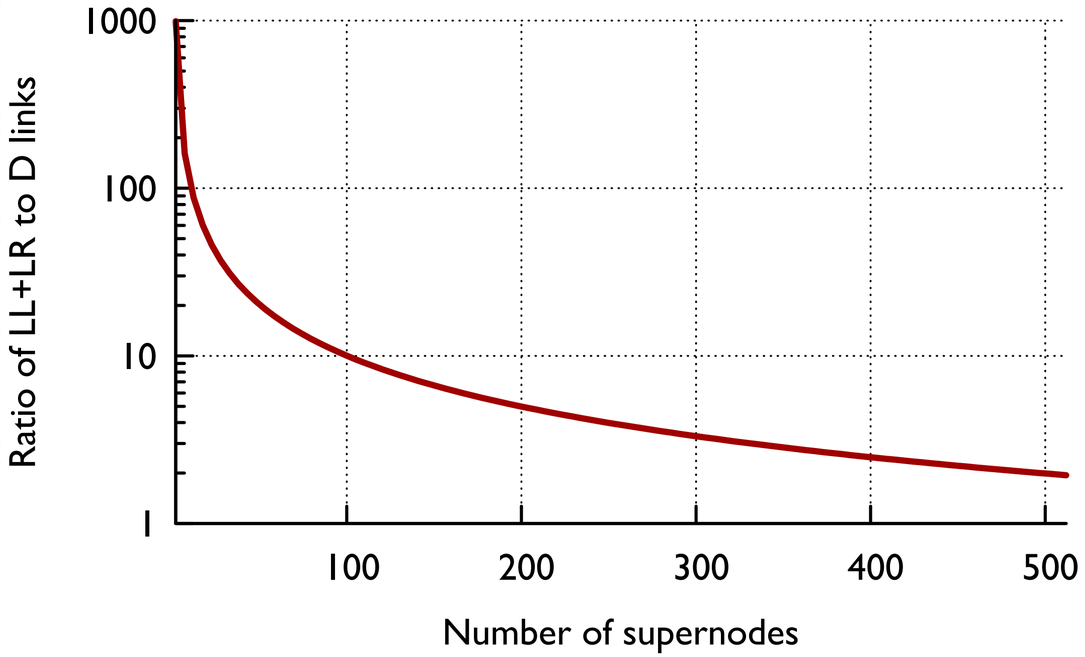
- Few global links when building a smaller than full-sized system

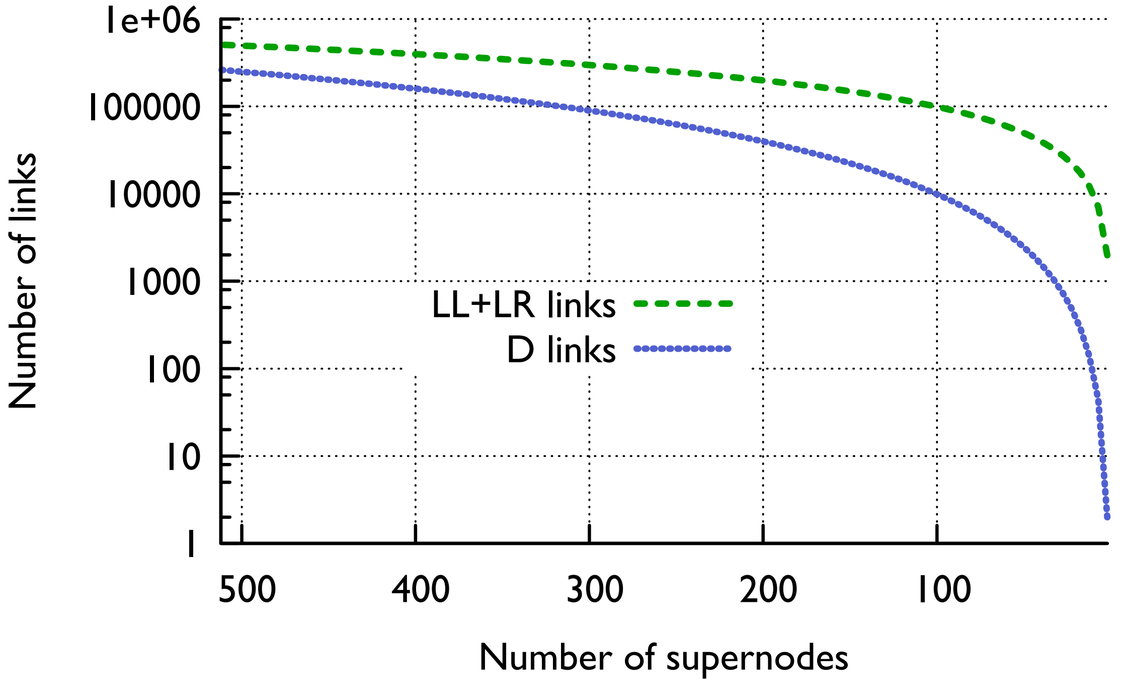




Global link bottleneck in dragonfly systems

- Few global links when building a smaller than full-sized system





Questions?

UNIVERSITY OF
MARYLAND

Abhinav Bhatele

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu