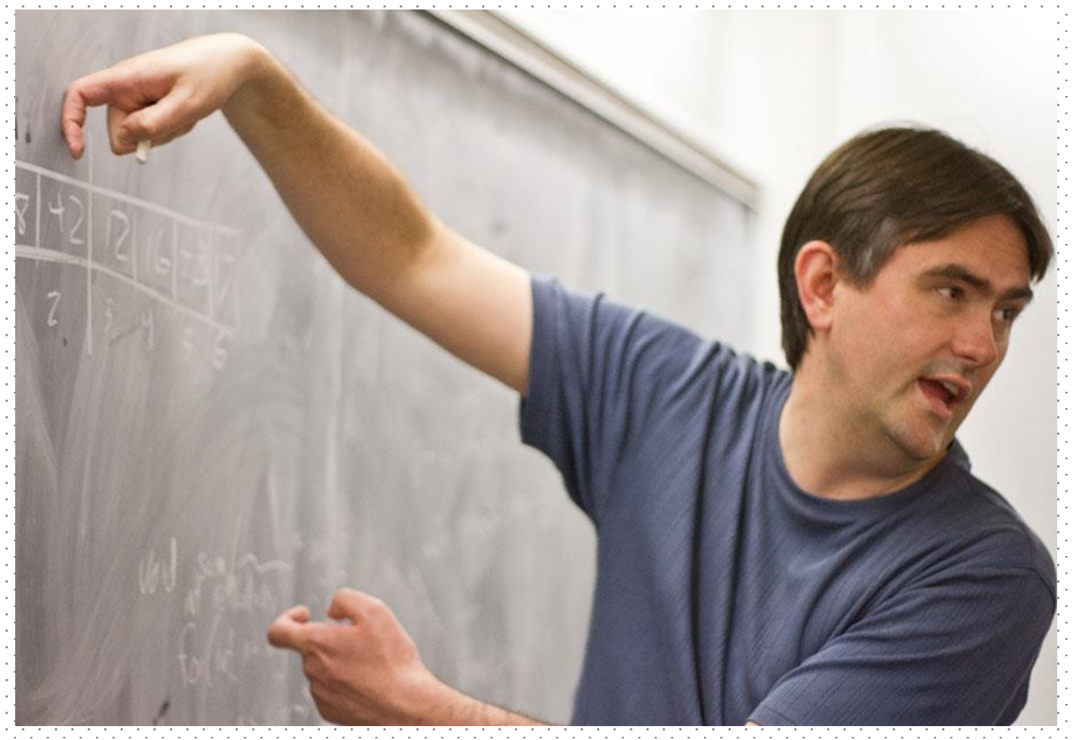
Dragonfly Task Mapping For Non-Square Jobs

David Bunde
dbunde@knox.edu

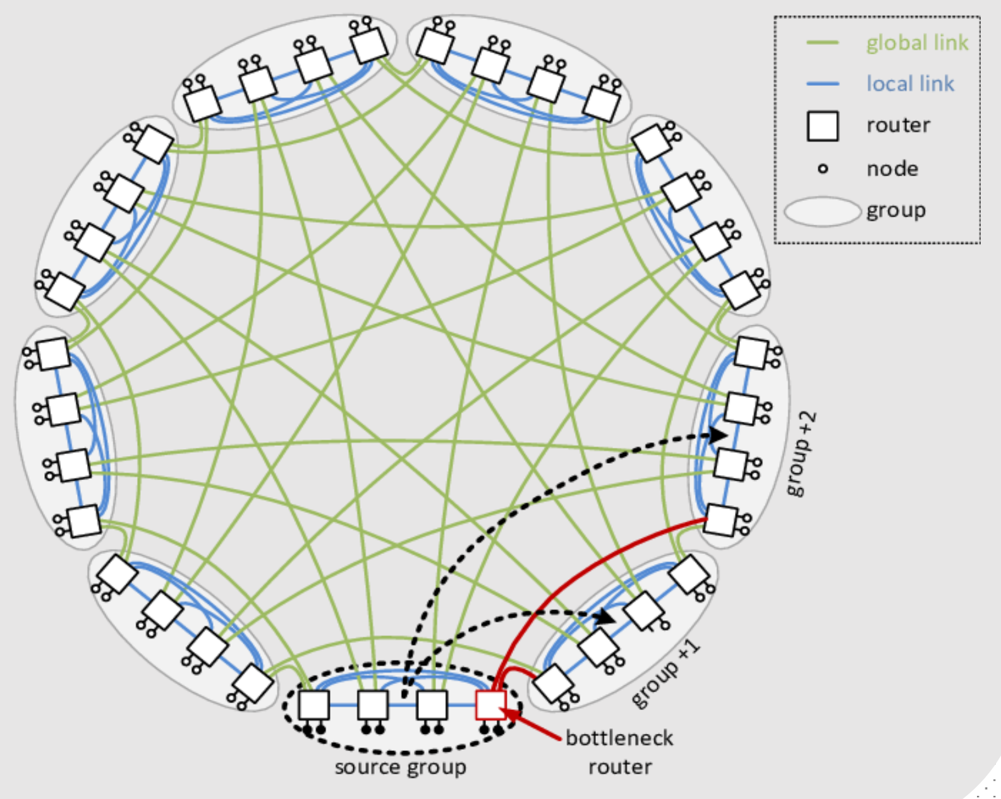
Musaddiq Javed
mujaved@knox.edu

Arsalan Bin Najeeb
abnajeeb@knox.edu



What Is Dragonfly?

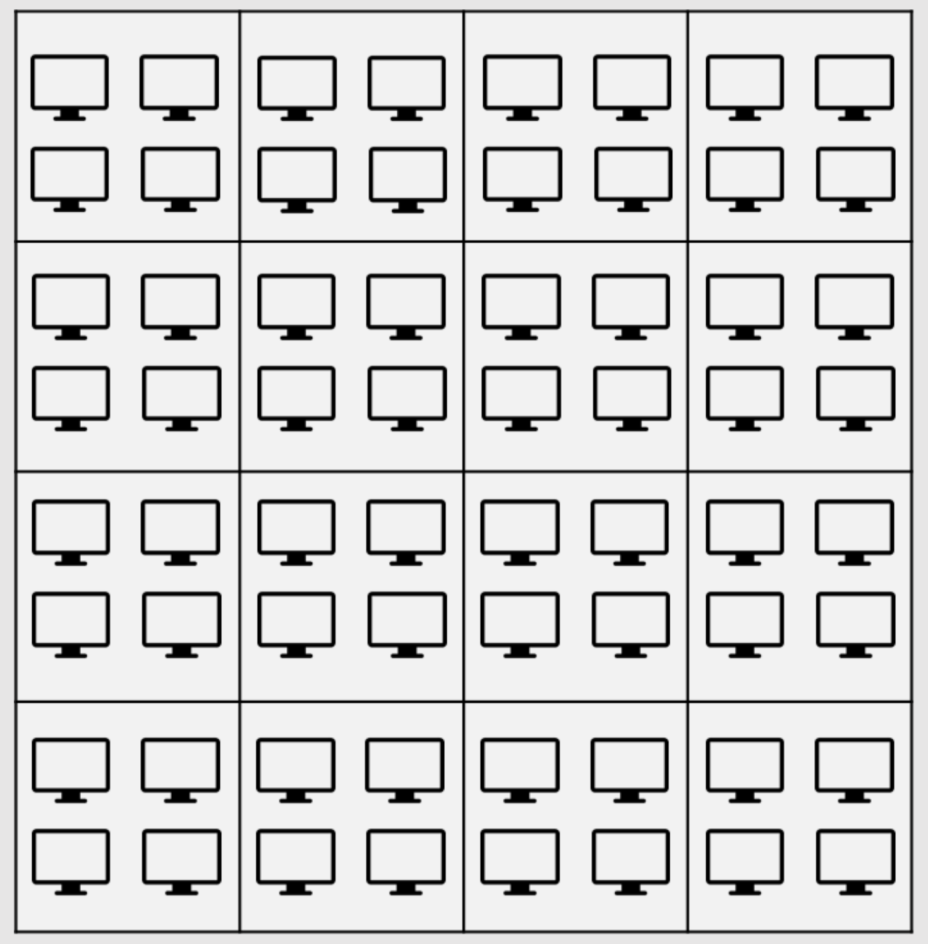
Dragonfly Topology is a scheme for high-performance computers (HPC) that works to balance bandwidth present on the system and reduce latency. By doing this we can balance the bandwidth of a Dragonfly system and assure the flow of information in a dragonfly system is not disrupted.



Why Task Mapping?

The goal for our task mapping is to decrease the latency in communication time, for this we assign a job task to a specific computer resource. These computer resources are called groups which consists of routers, each router is also connected to nodes.

This is especially useful for applications of HPC, such as weather simulations for predicting weather patterns. A group performs specific computations which are then sent to neighboring groups to help in completing their computations.

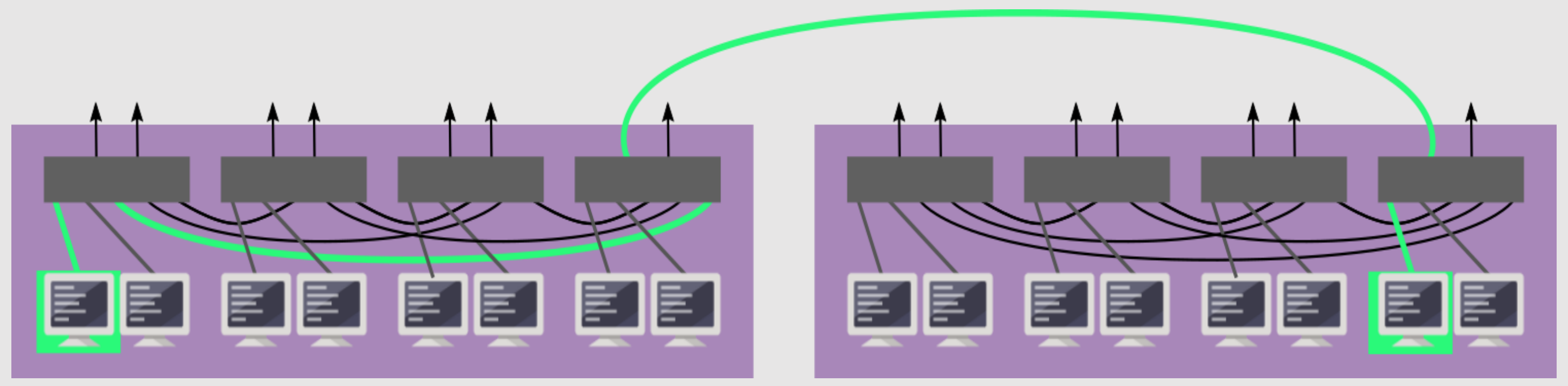


Balanced Adjacency Coloring (BAC)

Colors are assigned to a group to specify that each group is different from one another. The number of times each pairs of colors are made should be equal, or at least within one of each other.

This is because groups only have one link between each other in comparison routers inside a group have multiple connections and are connected to each other router.

0	1	2	3
2	0	3	1
1	3	0	2
3	2	1	0



4 x K With K Colors

Our Balanced Adjacency scheme works by adding two lines at a time, by doing so we can continue creating a Non-square BAC that still holds true to the definition of a BAC. Our scheme works on both even and odd k's. The example below shows how to create a Non-square BAC (4 x k) on a 2-D stencil grid, in this case K starts with 4 but by the end of our scheme becomes 8. The steps for our scheme have been outlined below.

0	1	2	3
2	0	3	1
1	3	0	2
3	2	1	0
0	3	2	1

For the first row:

- Push forward a color by one step after every fourth row

0	1	2	3
2	0	3	1
1	3	0	2
3	2	1	0
0 A	3	2 B	1
2 B	1	0 A	3

To add the next row:

- Switch Group A with Group B
- This now completes the 2 lines that we add

0	1	2	3
2	0	3	1
1	3	0	2
3	2	1	0
0 A	3	2 B	1
2 B	1	0 A	3
3 A^R	0	2 B	1
2 B	1	3 A^R	0

- Change Group A to A^R
- Now switch Group A^R with B
- Switch Group A^R with Group B again for the last line to be added

Comparison of Number of Pairs Formed

Old 4x8	New 4x8
0 - 1 -------> 9	0 - 1 -------> 9
0 - 2 -------> 8	0 - 2 -------> 8
0 - 3 -------> 9	0 - 3 -------> 8
1 - 2 -------> 9	1 - 2 -------> 8
1 - 3 -------> 8	1 - 3 -------> 9
2 - 3 -------> 9	2 - 3 -------> 9

This work was partially supported by the National Science Foundation under grant CNS-1423413

KNOX COLLEGE