Paper Discussion
Topology and Interference

Lecture 22

Apr 10, 2024

Network Performance Aware MPI Collective Communication Operations in the Cloud
Gong et al., IEEE TPDS 2015

Differences/Challenges

- Topology information is not available
- Certain privileges are not available
- Topology is not static
- Network asymmetry
    - Performance varies for different VM pairs
- Schedule communication for machine pairs with low network performance in an optimal manner
- How do we infer/discover the topology

Goals

- Model to capture the network performance among different machine pairs in the cloud environment
- Network performance aware algorithms for collective operations with different message sizes
  - Broadcast
  - Reduce

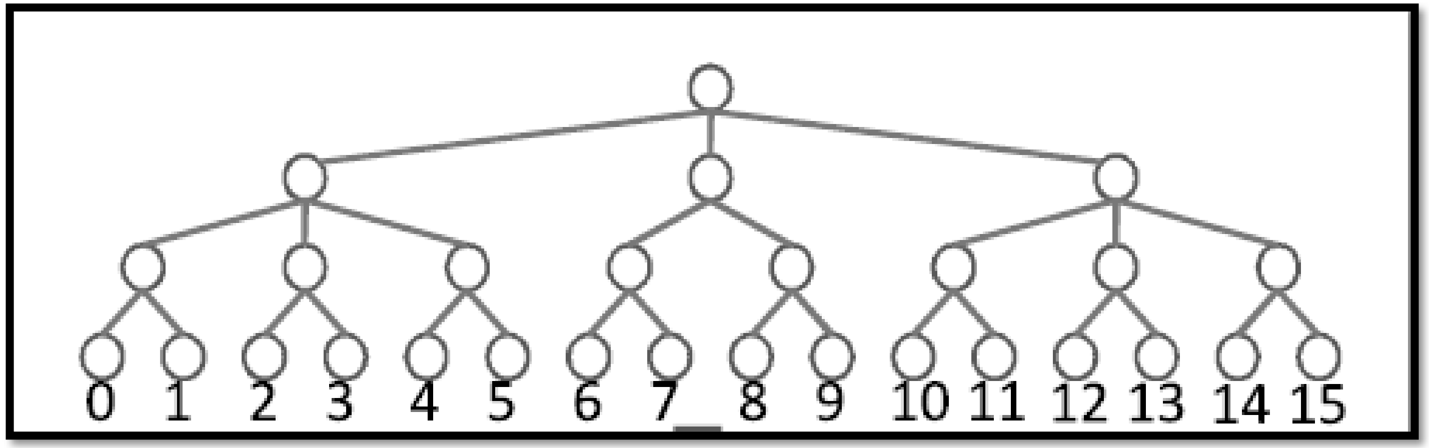
Topology Discovery

- Communication time between N/2 pairs are measured (repeated)
- Latency matrix [N x N]
  - L (i, j) – average latency from node i to node j
  - L (i, i) = 0
- Bandwidth matrix [N x N]
  - B (i, j) – average bandwidth from node i to node j
  - B (i, i) = ?
- Network performance matrix for a message size m [N x N]
  - M (i,j) = L (i,j) + m/B (i,j)

Algorithm

- Network performance among a set of VMs
  - Latency matrix
  - Bandwidth matrix
  - Network performance matrix (Hockney model)
- Group the machines
  - Form a number of non-overlapping groups
  - Each group has machines that have similar network performance (use a distance parameter and threshold)
  - Build hierarchy among the groups by increasing the closeness parameter
  - Adjust span (difference in #machines between largest and smallest group)

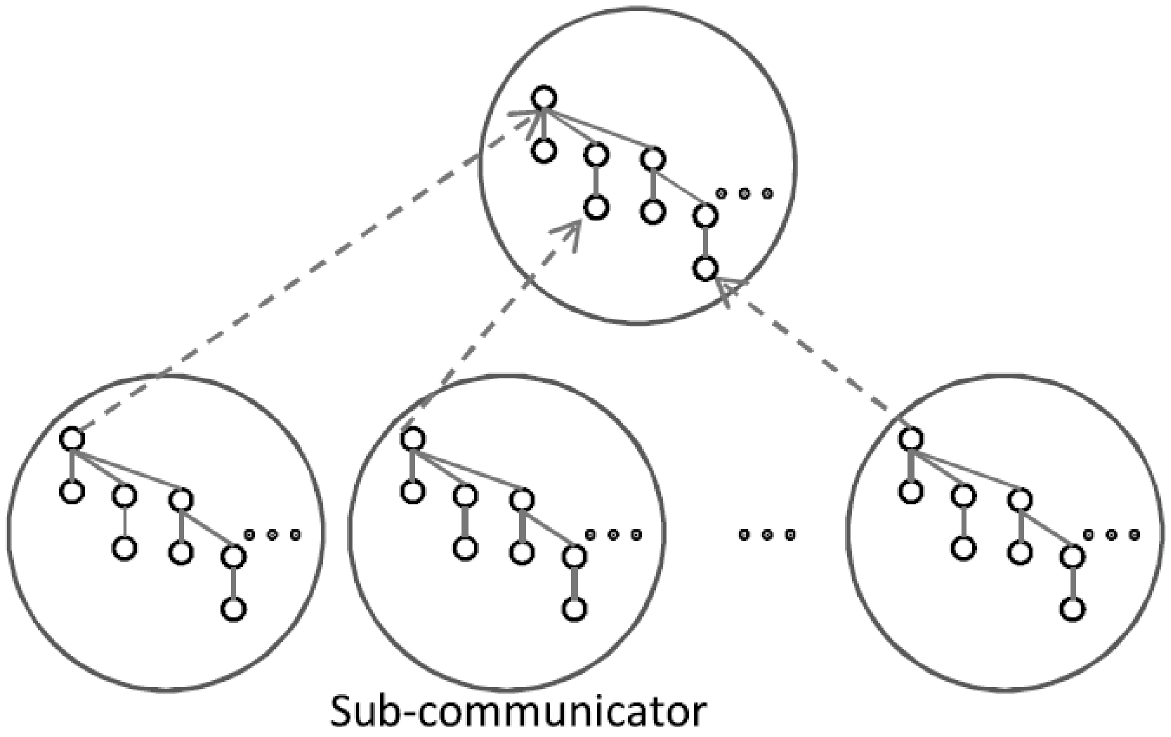
Network-aware Hierarchy



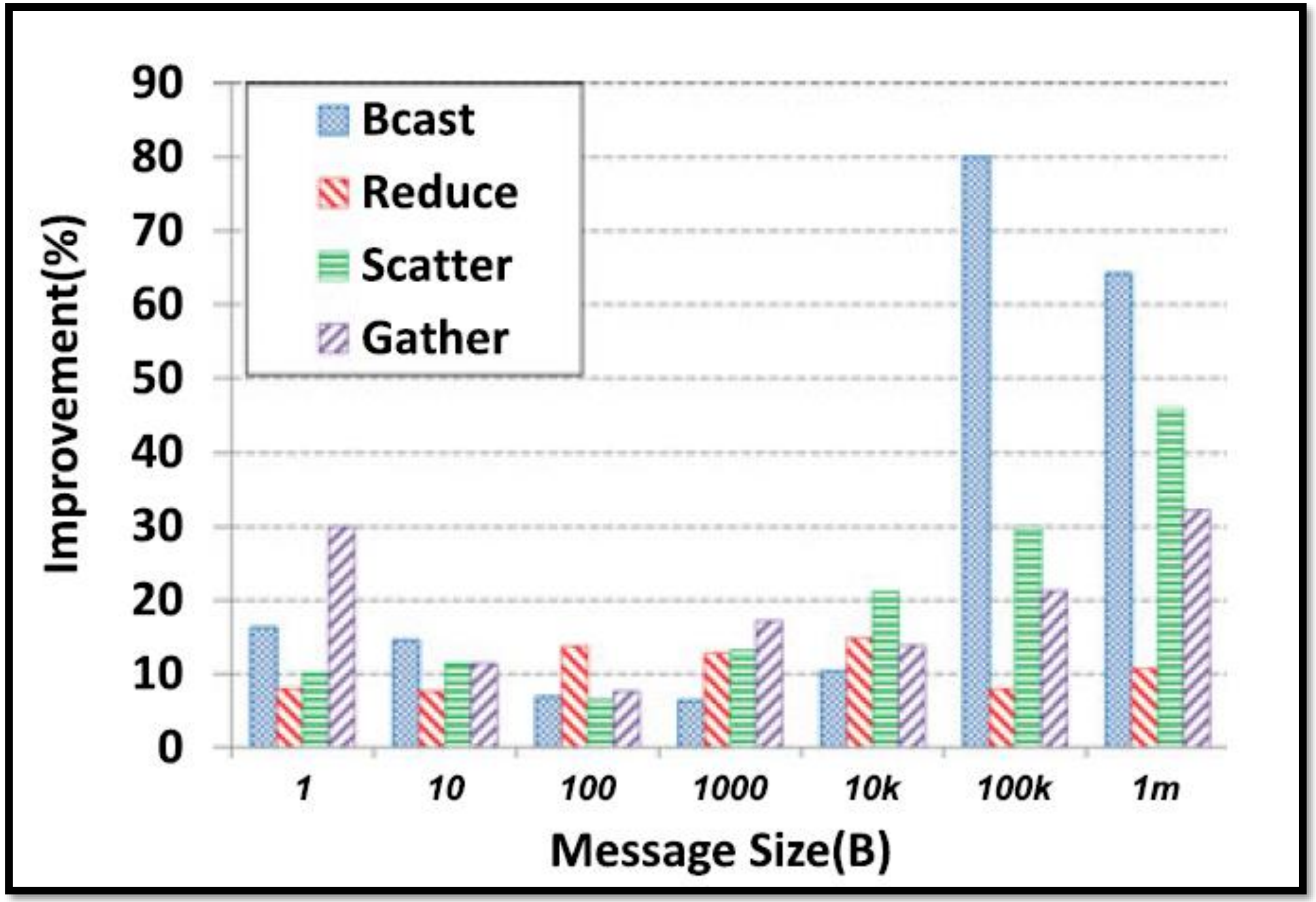
Network Performance Aware Reduce

- Build network performance hierarchy based on L, B, m (message size)
- Select leader process in each group
- Create sub-communicators for each group
- Each group performs reduce independently
- Hierarchically reduce

Network-aware Reduce



Results


Bcast
Reduce
Scatter
Gather
Improvement(%)
90
80
70
60
50
40
30
20
10
0
1
10
100
1000
10k
100k
1m
Message Size(B)

Analyzing Network Health and Congestion in Dragonfly-based Supercomputers
Bhatele et al., IPDPS 2016

Overview

- Job-interference is an important factor for communication-intensive applications
  - Network congestion
- Understanding causes of congestion
  - Routing algorithms
  - Job placement policies
  - Network topology
- 44 workloads, on up to 131K cores, 588 simulations exploring different causes

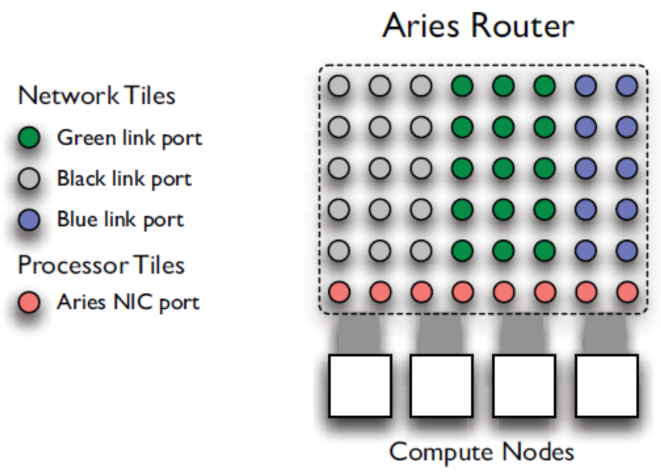
Network Simulation

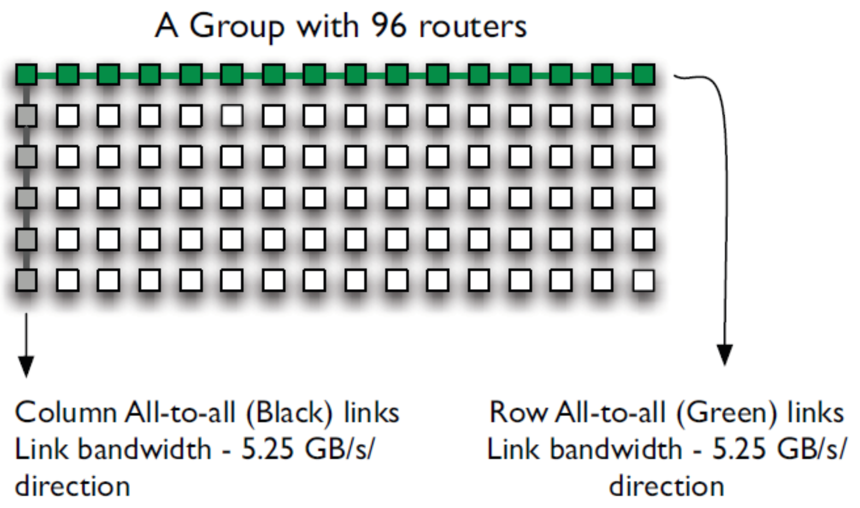
- Parallel network simulator
- Simulate parallel workloads and record network traffic (Damselfly: https://github.com/LLNL/damselfly)
- Input
  - Router connectivity graph
  - Application communication graph
  - Routing policy
- Output
  - Network link traffic (counters) for multi-job simulations
  - Identify job interference
- Enables
  - Exploring job placements, routing, network configurations

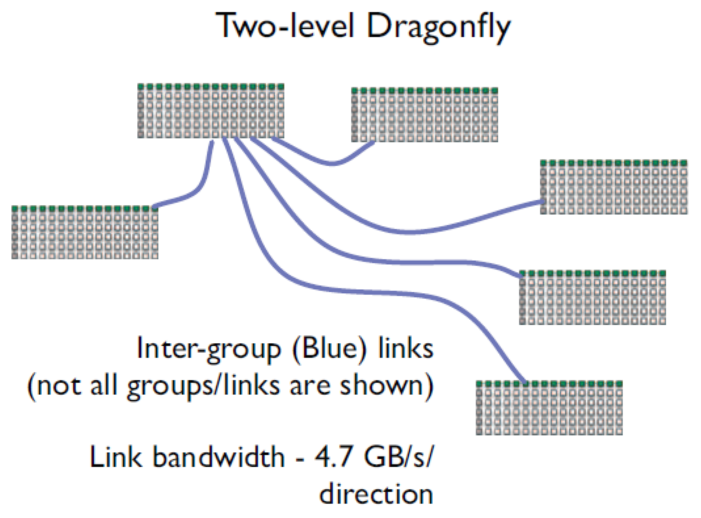
Communication Patterns

- 2D stencil (4 neighbours, 64 KB)
- 4D stencil (8 neighbours, 4 MB)
- Many-to-many (all to all in groups of 16 or 32, 32 KB)
- Spread (Random 6 to 27 neighbours, 512 KB)
- Unstructured mesh (Select 6 to 27 neighbours, 512 KB)

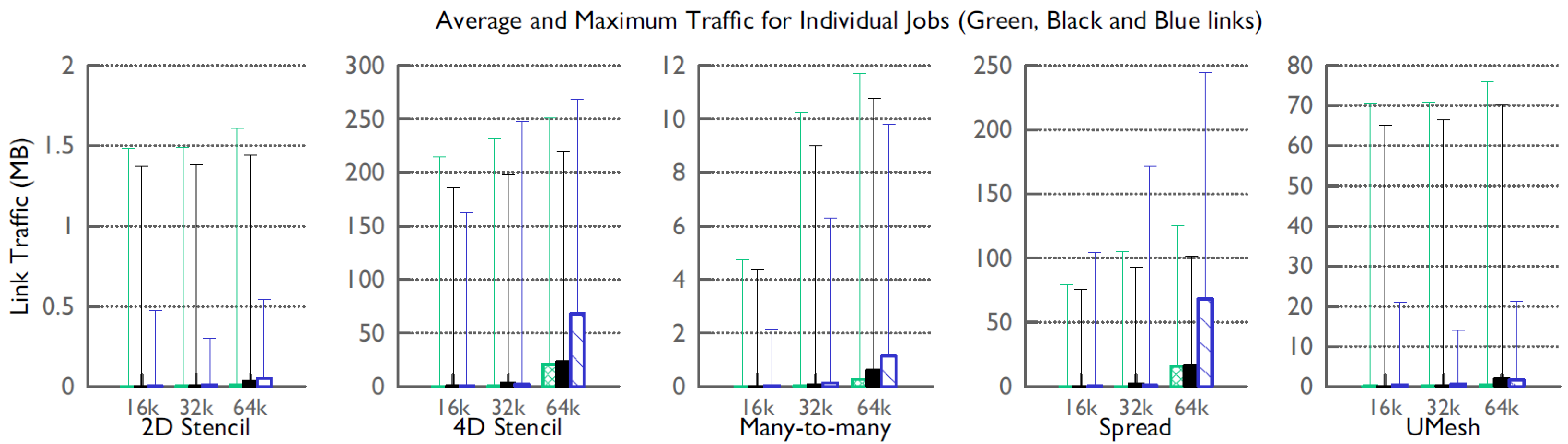
Dragonfly Network Topology




A Group with 96 routers
Column All-to-all (Black) links
Link bandwidth - 5.25 GB/s/ direction
Row All-to-all (Green) links
Link bandwidth - 5.25 GB/s/ direction


Two-level Dragonfly
Inter-group (Blue) links
(not all groups/links are shown)
Link bandwidth - 4.7 GB/s/ direction

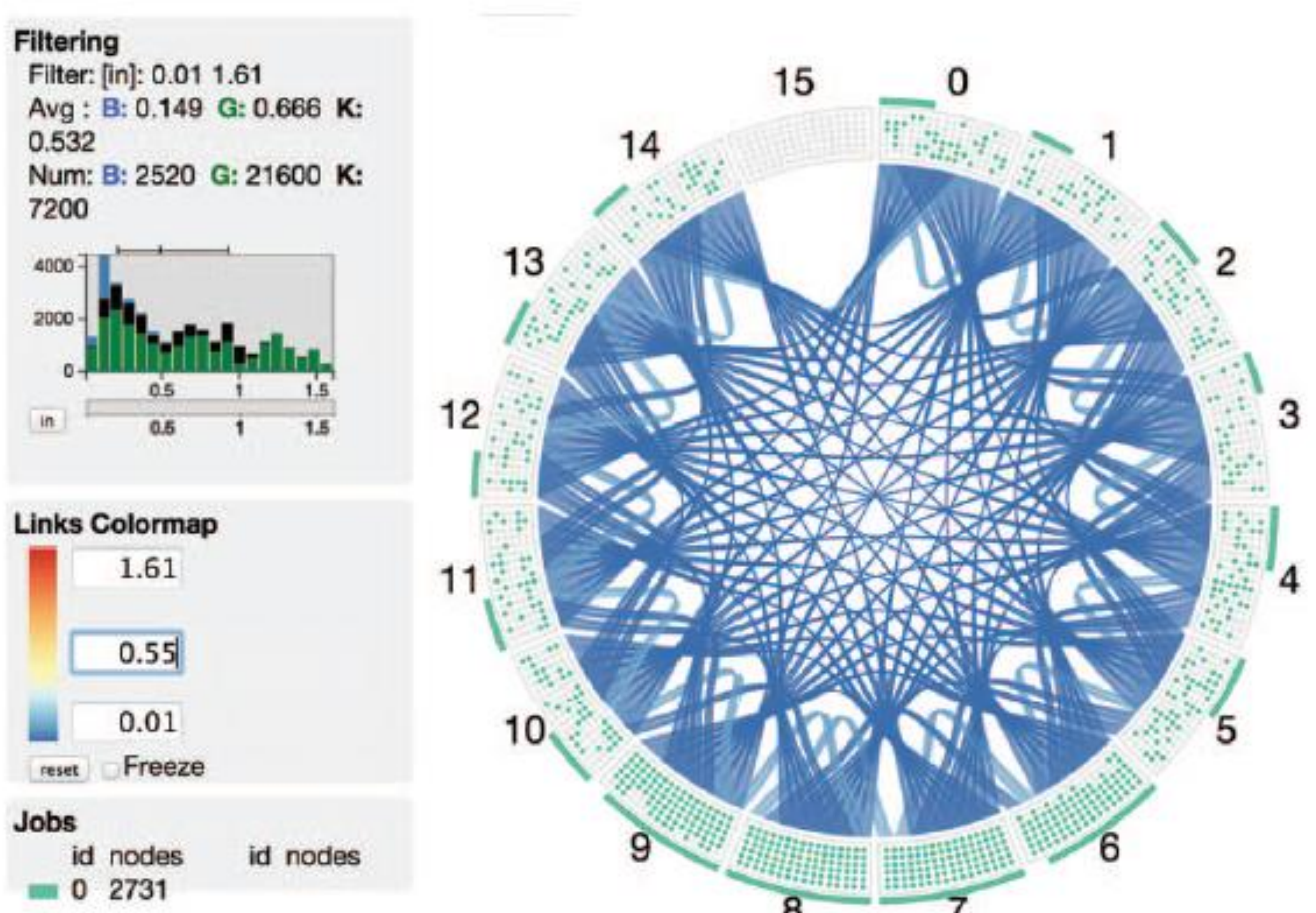
Link Prediction (Isolated)



Observations

- Link traffic increases with job size
- Some create significantly high traffic
- Maximum traffic on ? links → hot-spots

Isolated Network Traffic (64K cores)



Filtering

Filter: [in]: 0.01 1.61

Avg : B: 0.149 G: 0.666 K: 0.532

Num: B: 2520 G: 21600 K: 7200

Links Colormap

1.61

0.55

0.01

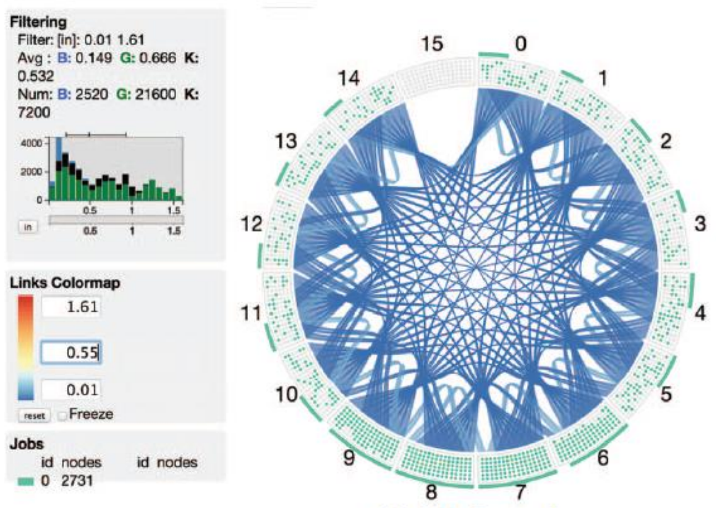
reset ☐ Freeze

Jobs

	id	nodes	id	nodes
🟩	0	2731		

(a) 2D Stencil

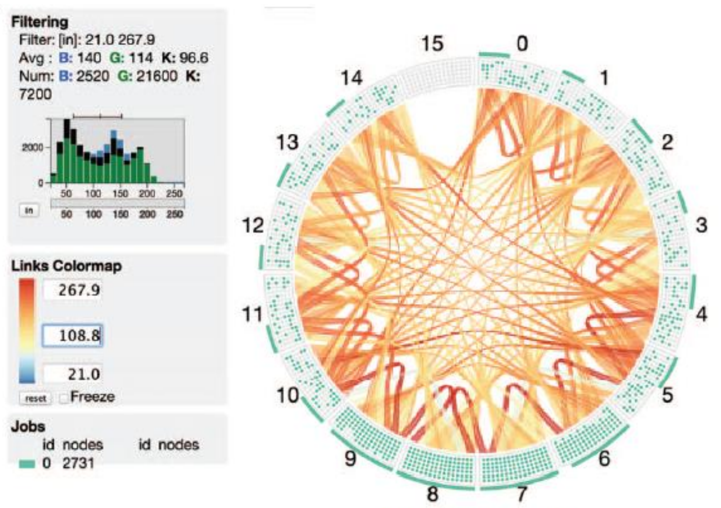
Isolated Network Traffic (64K cores)


Filtering
Filter: [in]: 0.01 1.61
Avg : B: 0.149 G: 0.666 K: 0.532
Num: B: 2520 G: 21600 K: 7200

Links Colormap
1.61
0.55
0.01
reset Freeze

Jobs
id nodes id nodes
0 2731

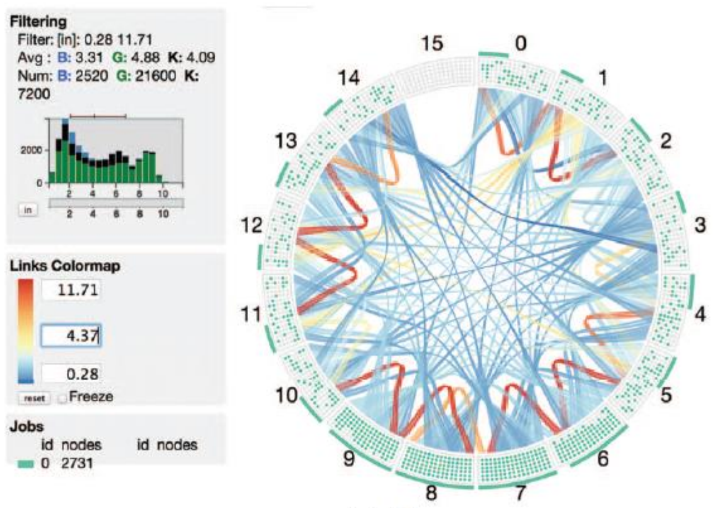
(a) 2D Stencil


Filtering
Filter: [in]: 21.0 267.9
Avg : B: 140 G: 114 K: 96.6
Num: B: 2520 G: 21600 K: 7200

Links Colormap
267.9
108.8
21.0
reset Freeze

Jobs
id nodes id nodes
0 2731

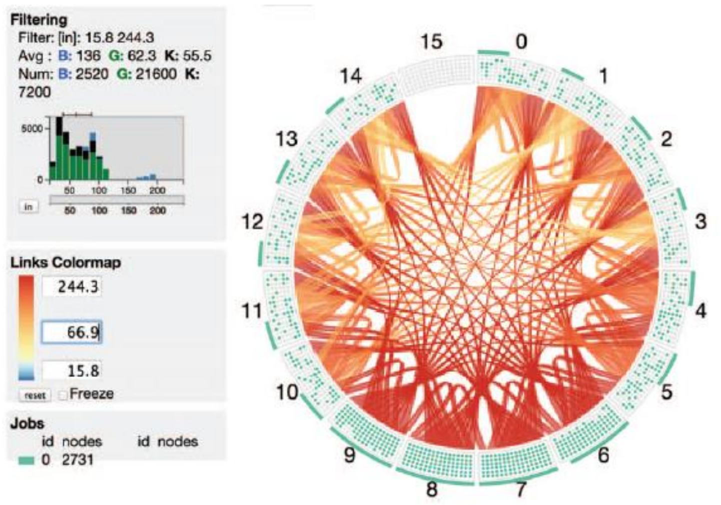
(b) 4D Stencil


Filtering
Filter: [in]: 0.28 11.71
Avg : B: 3.31 G: 4.88 K: 4.09
Num: B: 2520 G: 21600 K: 7200

Links Colormap
11.71
4.37
0.28
reset Freeze

Jobs
id nodes id nodes
0 2731

(c) Many-to-many

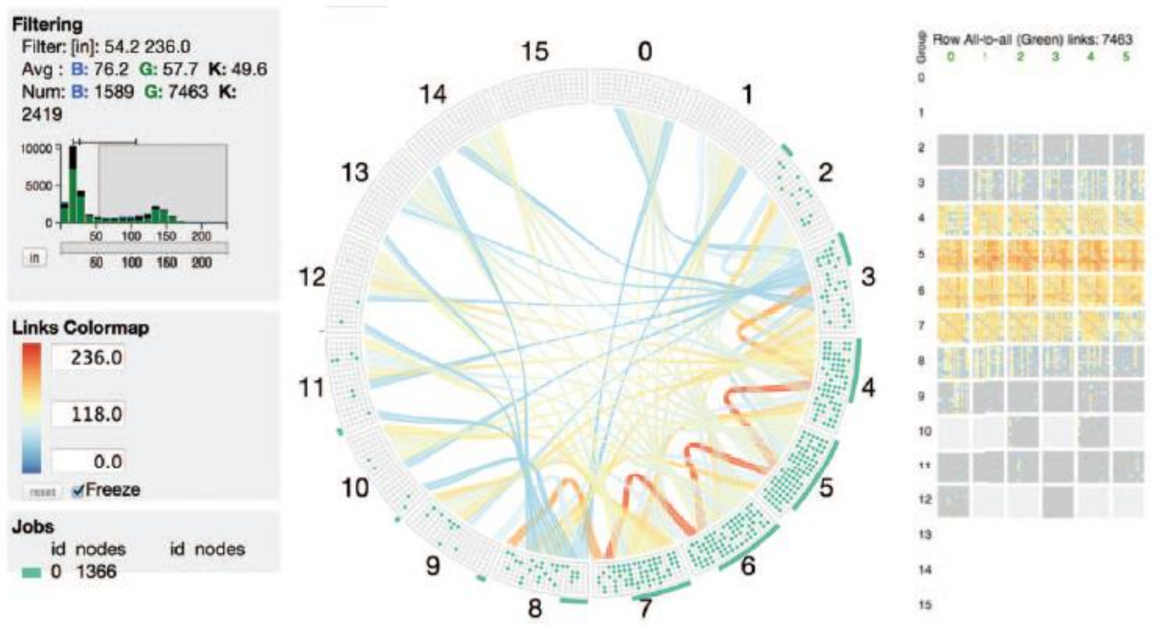

Filtering
Filter: [in]: 15.8 244.3
Avg : B: 136 G: 62.3 K: 55.5
Num: B: 2520 G: 21600 K: 7200

Links Colormap
244.3
66.9
15.8
reset Freeze

Jobs
id nodes id nodes
0 2731

(d) Spread

Job Placement Effect



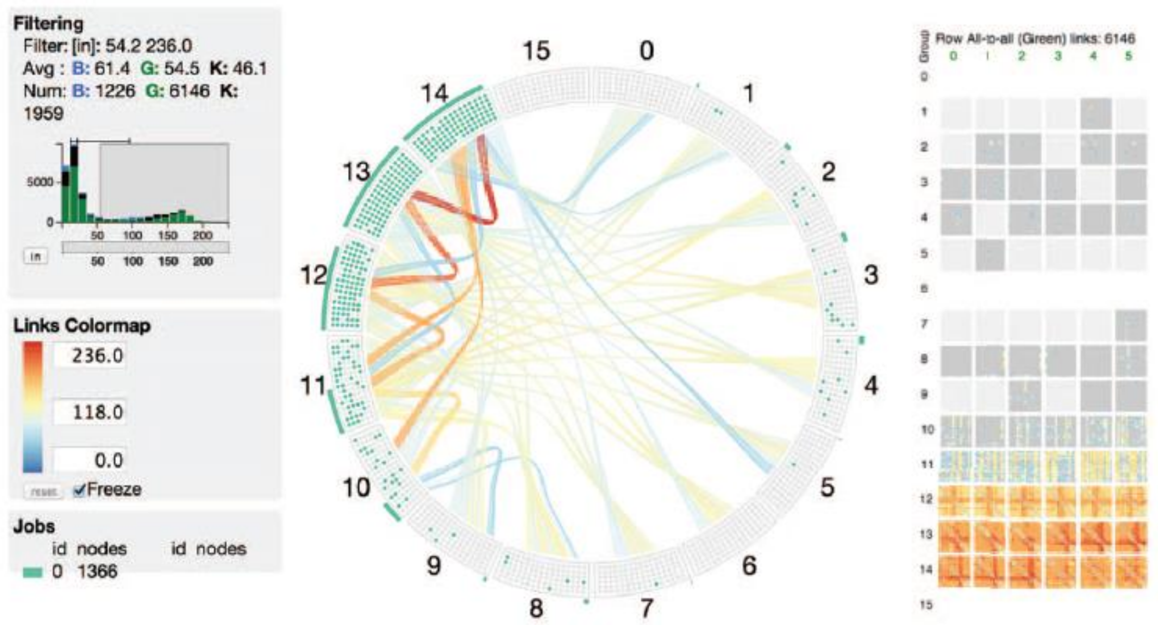
Filtering
Filter: [in]: 54.2 236.0
Avg : B: 76.2 G: 57.7 K: 49.6
Num: B: 1589 G: 7463 K: 2419

Links Colormap
236.0
118.0
0.0
reset ✓Freeze

Jobs

id	nodes	id	nodes
🟩 0	1366		

Row All-to-all (Green) links: 7463



Filtering
Filter: [in]: 54.2 236.0
Avg : B: 61.4 G: 54.5 K: 46.1
Num: B: 1226 G: 6146 K: 1959

Links Colormap
236.0
118.0
0.0
reset ✓Freeze

Jobs

id	nodes	id	nodes
🟩 0	1366		

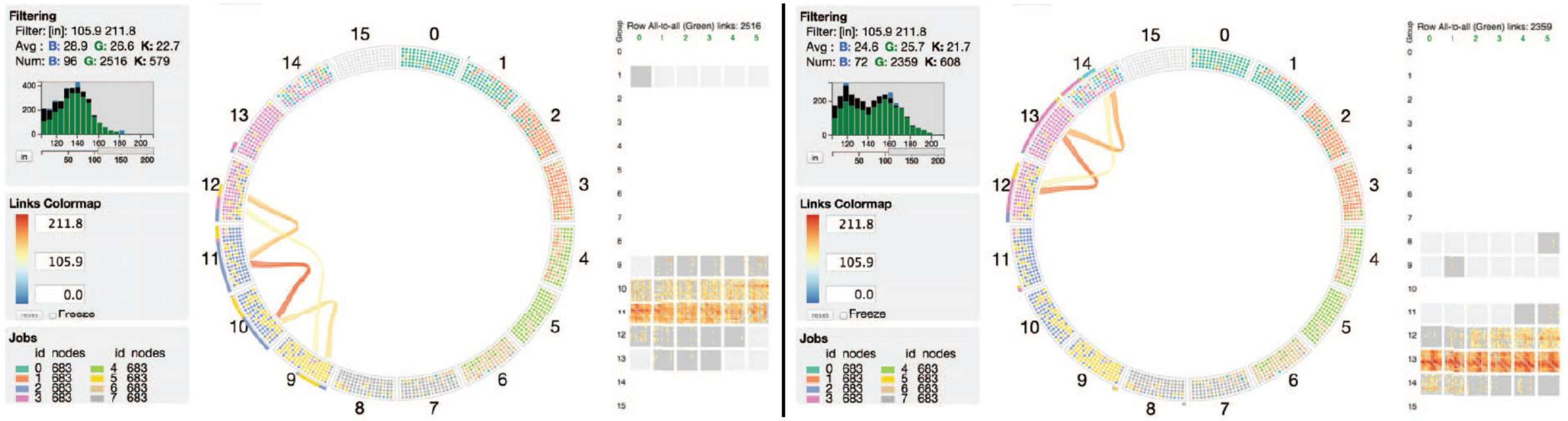
Row All-to-all (Green) links: 6146

Scattered vs. Compact placement of 4D stencil on 32K cores

In scattered
- More blue links used (1589)
- Lower maximum traffic (191 MB)

In compact
- Less blue links used (1226)
- Higher maximum traffic (236 MB)

Job Placement Effects (Relative)


Filtering
Filter: [in]: 105.9 211.8
Avg : B: 28.9 G: 26.6 K: 22.7
Num: B: 96 G: 2516 K: 579

Links Colormap
211.8
105.9
0.0
Freeze

Jobs

id	nodes	id	nodes
0	683	4	683
1	683	5	683
2	683	6	683
3	683	7	683

Row All-to-all (Green) links: 2516

Filtering
Filter: [in]: 105.9 211.8
Avg : B: 24.6 G: 25.7 K: 21.7
Num: B: 72 G: 2359 K: 608

Links Colormap
211.8
105.9
0.0
Freeze

Jobs

id	nodes	id	nodes
0	683	4	683
1	683	5	683
2	683	6	683
3	683	7	683

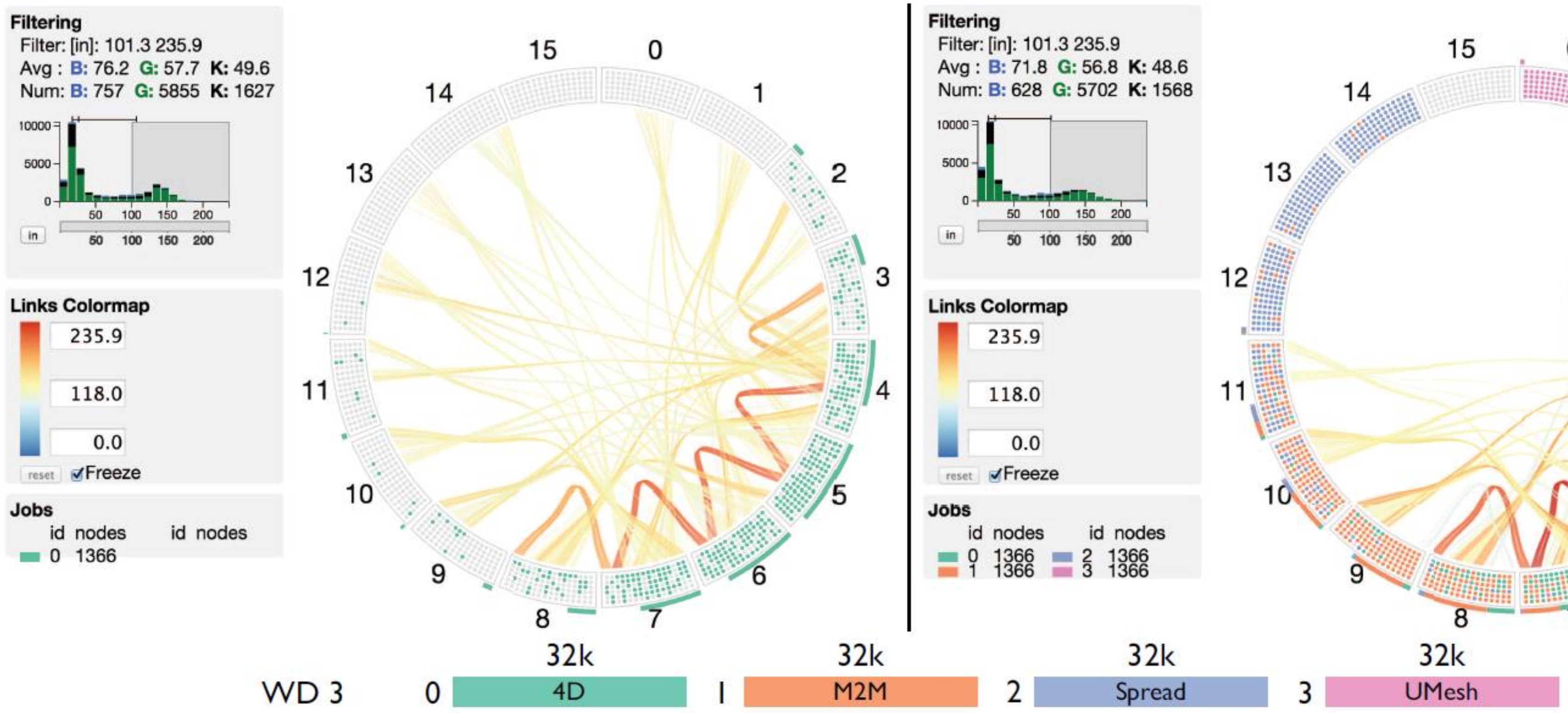
Row All-to-all (Green) links: 2359

	16k	16k	16k	16k	16k	16k	16k	16k
WD 1	0 Spread	1 2D	2 4D	3 4D	4 M2M	5 M2M	6 UMesh	7 UMesh

Scattered, Job 2 (left) and compact, Job 3 (right) job placements (4D stencil on 16K cores with other jobs).

Links shared with communication-heavy code (Job 2 and 3), 8% higher maximum traffic (right figure).

Inter-job Interference

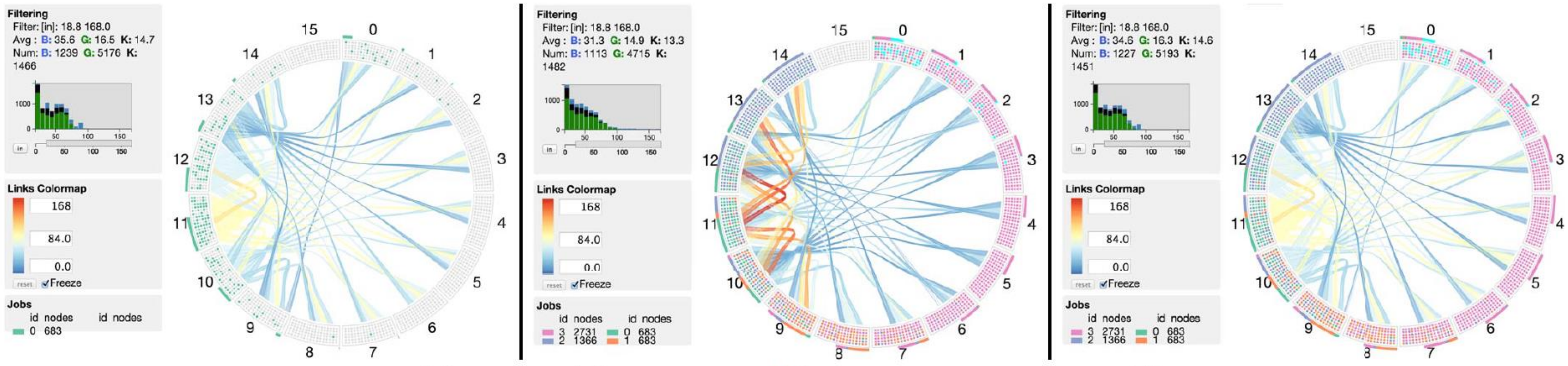


4D stencil (job 0) runs in isolation (left) and 4D stencil (job 0) runs with other jobs (right)

More blue links utilized and lower maximum traffic when other jobs are not running (left)

Job 0’s traffic confined to fewer blue links to share bandwidth with other jobs (right)

Inter-job Interference (Dependent on Job Type)


Filtering
Filter: [in]: 18.8 168.0
Avg : B: 35.6 G: 16.5 K: 14.7
Num: B: 1239 G: 5176 K: 1466

Links Colormap
168
84.0
0.0
reset ✓Freeze

Jobs

id	nodes	id	nodes
🟩 0	683		

Filtering
Filter: [in]: 18.8 168.0
Avg : B: 31.3 G: 14.9 K: 13.3
Num: B: 1113 G: 4715 K: 1482

Links Colormap
168
84.0
0.0
reset ✓Freeze

Jobs

id	nodes	id	nodes
🟪 3	2731	🟩 0	683
🟦 2	1366	🟧 1	683

Filtering
Filter: [in]: 18.8 168.0
Avg : B: 34.6 G: 16.3 K: 14.6
Num: B: 1227 G: 5193 K: 1451

Links Colormap
168
84.0
0.0
reset ✓Freeze

Jobs

id	nodes	id	nodes
🟪 3	2731	🟩 0	683
🟦 2	1366	🟧 1	683

		16k		16k		32k		64k
WD 4	0	Spread	1	2D	2	UMesh	3	4D
WD 5	0	Spread	1	UMesh	2	2D	3	M2M

Congestion increases more when Spread is run with communication-intensive 4D stencil (middle) than when run along with M2M (right)