Design and Evaluation of Topology-aware Scatter and AllGather Algorithms for Dragonfly Networks
Nathanaël Cheriere, Matthieu Dorier

To cite this version:
Nathanaël Cheriere, Matthieu Dorier. Design and Evaluation of Topology-aware Scatter and AllGather Algorithms for Dragonfly Networks. Supercomputing 2016, Nov 2016, Salt Lake City, United States. Supercomputing 2016. hal-01400271

HAL Id: hal-01400271
https://inria.hal.science/hal-01400271
Submitted on 21 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Design and Evaluation of Topology-aware Scatter and AllGather Algorithms for Dragonfly Networks

Nathanaël Cheriere¹, Matthieu Dorier².
¹ ENS Rennes, ² Argonne National Laboratory

New efficient communication algorithms for Dragonfly
- Cost of traditional topologies too high at scale
- Dragonfly: a new topology
- Opportunity for new efficient communication algorithms

Challenge: Global links are potential bottlenecks

AllGather

Existing algorithms
RING
Default in MPICH for 80 KiB and more
BRUCK - Bruck’s algorithm
Default in MPICH for less than 80 KiB

Our contribution
TAR - Topology Aware Ring
Idea: minimize global link utilization
1. Build a smart ring
2. Send data

Minimize data transfers on global links

Scatter

Existing Algorithms
TRE - Binomial Tree
Default in MPICH

Our contribution
GLF - Global Link First
Use global links only during the first phase

Send data to terminals one by one
Minimize data transfers on global links

LIN twice faster than the others!

Sending 1 KiB per terminal
With uniform background traffic

Major impact factor: the buffer size!
For TREE, GLF, and LLF:
- Large amounts of data transferred
- Fills buffers, slows down transfer
For LIN:
- 1 KiB transferred each time
- No buffer saturated, no slowdown

Conclusion
- Topology and hardware characteristics impact the performance of algorithms
- LIN is twice faster than LLF, GLF, and TREE
- No significant differences between LLF, GLF, and TREE

Why is it complex to make AllGather hardware-aware?
- A lot more data sent over the network creating more contention
- Overall run time determined by the slowest link due to the ring structure
- Terminals transfer on the same path (TAR, RING), no balance across buffers

Ongoing work
- Broadcast as Scatter followed by AllGather
- Computation and communication algorithms such as Reduce
- Energy efficiency of communication algorithms

Conclusion
- Topology awareness matters: TAR wins!
- BRUCK still better for small transfers
- Background traffic does not change the behavior