Collectives on Two-tier Direct Networks

EuroMPI – 2012

Nikhil Jain, JohnMark Lau, Laxmikant Kale

26th September, 2012

PPL UIUC PARALLEL PROGRAMMING LABORATORY

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

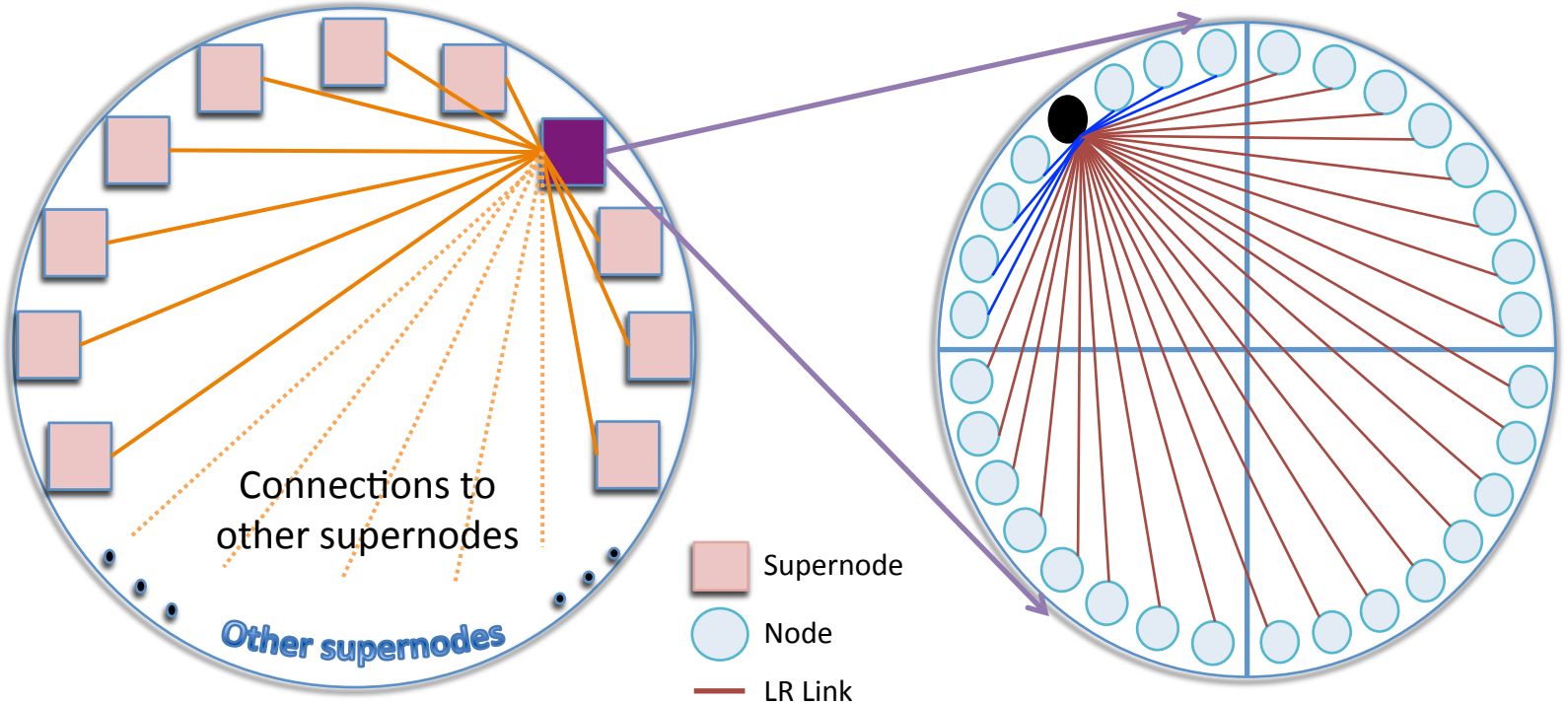
Motivation

- Collectives are an important component of parallel programs
  - Impacts performance and scalability
  - Performance of large message collectives constrained by network bandwidth
- Topology aware implementations are required to extract best performance
- Clos, fat-tree and torus are low-radix and have large diameters
  - Mutiplicity of hops make them congestion prone
  - Carefully designed collective algorithms

Two-tier Direct Networks

- New network topology
  - IBM PERCS
  - Dragonfly - Aries
- High radix network with mutiple levels of connections
- At level 1, multi-core chips (nodes) are clustered to form supernodes/racks
- At level 2, connections are provided between the supernodes

Two-tier Direct Networks



Second tier connections (of one supernode)

First tier connections within a SN (of one node)

Topology Oblivious Algorithms

Scatter/Gather	Binomial Tree
Allgather	Ring, Recursive Doubling
Broadcast	DeGeijn's Scatter with Allgather
Reduce-scatter	Pairwise Exchange
Reduce	Rabenseifner's Reduce-Scatter with Gather

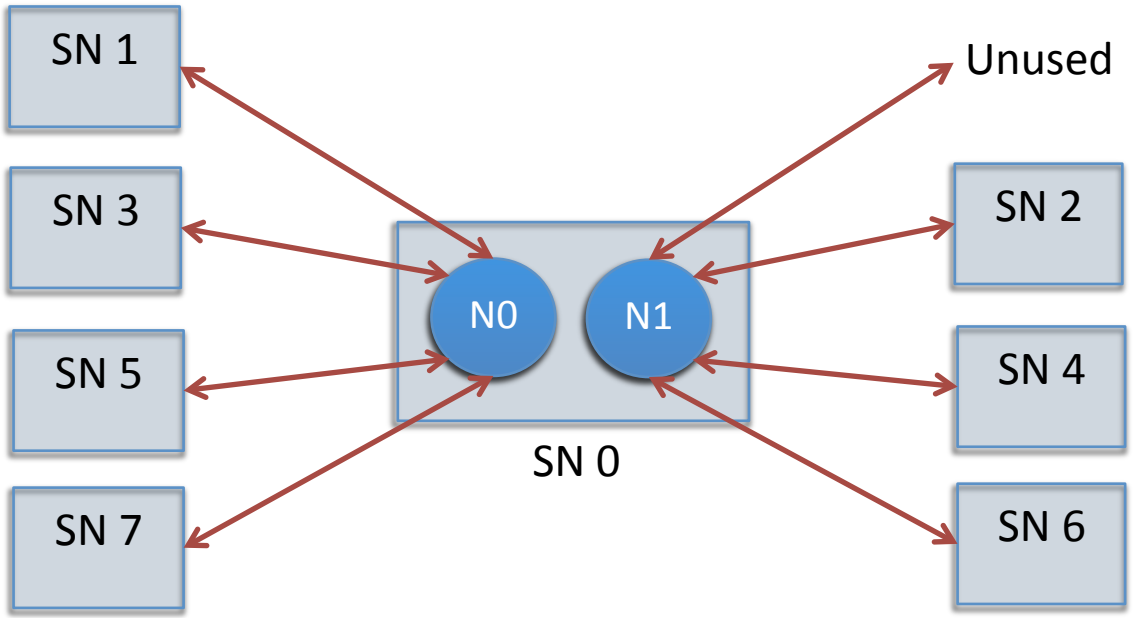
Topology Aware Algorithms

- Blue Gene
  - Multi-color non-overlapping spanning trees
  - Generalized n-dimensional bucket algorithm
- Tofu
  - The Trinaryx3 Allreduce
- Clos
  - Distance-halving allgather algorithm

Assumptions/Conventions

- Task – (core ID, node ID, supernode ID)
  - core ID and node ID are local
- All-to-all connection between nodes of a supernode
- *nps* – nodes per supernode, *cpn* – cores per node
- Connection from a supernode to other supernodes originate from nodes in a round robin manner
  - Link from supernode *S1* to supernode *S2* originates at node *(S2 modulo nps)* in supernode *S1*

Assumptions/Conventions



Connections among supernodes

- Focus on large messages – startup cost ignored

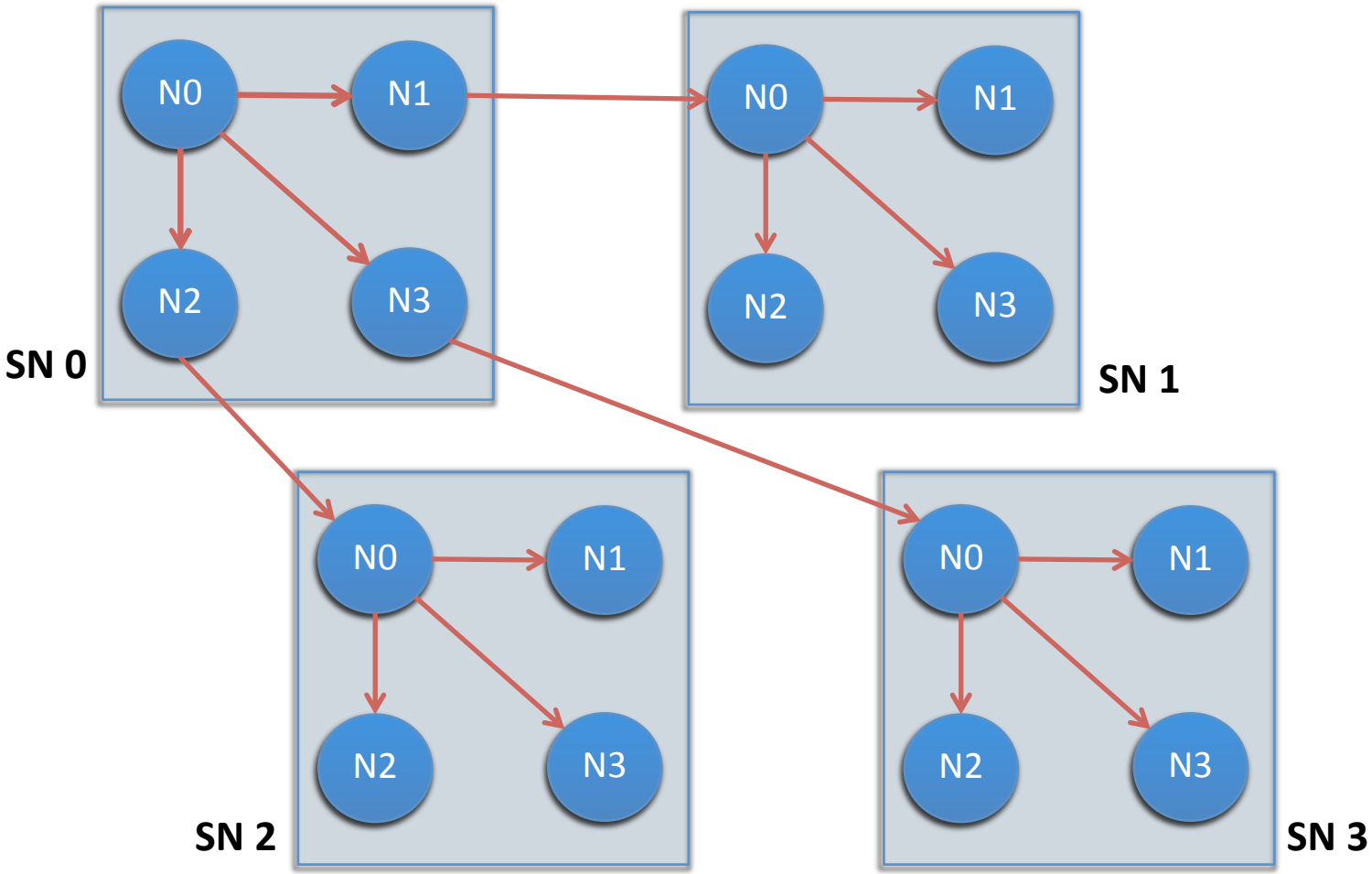
Two-tier Algorithms

- How to take advantage of the cliques and multiple levels of connections?
- SDTA – Stepwise Dissemination, Transfer or Aggregation
- Simultaneous exchange of data within level 1
- Minimize amount of data transfered at level 2
- (0, 0, 0) assumed root

Scatter using SDTA

- (0, 0, 0) → (0, *, 0)
  - Data sent to (0, x, 0) is data that belongs to supernodes connected to (0, x, 0)
- (0, *, 0) scatters the data to corresponding nodes (in other supernodes)
- (0, x, *) distributes the data within their supernode
- (0, *, *) provides data to other cores in their node

Step 2 – Transfer to other supernodes



Broadcast using SDTA

- Can be done using an algorithm similar to scatter – not optimal
- (0, 0, 0) divides data into *nps* chunks; sends chunk *x* to (0, x, 0)
- (0, *, 0) sends data to exactly one connected node (in other supernode)
- Every node that receives data acts like a broadcast source
  - Sends data to all other nodes in their supernodes
  - These nodes forward data to other supernodes
  - The recepient in other supernodes share it within their supernodes

Allgather using SDTA

- All-to-all networks facilitates parallel base broadcast
- Steps:
  - Every nodes shares data with all other nodes in its supernode
  - Every node shares the data (it has so far) with corresponding nodes in other supernodes
  - Nodes share the data within their supernodes
- Majority of communication at level 1 - minimal communication at level 2

Computation Collectives

- Owner core - core that has been assigned a part of the data that needs to be reduced
- Given a clique of *k* cores with size *m* data, consider the following known approach
  - Each core is made owner of size m/k data
  - Every core sends the data corresponding to the owner cores (in their data) to the owner cores – all-to-all network
  - The owner cores reduce the data they own

Multi-phased Reduce

- Perform reduction among cores of every node; collect the data at core 0
- Perform reduction among nodes of every supernode – decide owners carefully
- Perform reduction among supernodes; collect the data at the root

Reduce-scatter

- First two steps same as Reduce
- In reduction among supernodes, choose owners carefully – a supernode is owner of data that should be deposited on cores in it as part of Reduce-scatter
- Nodes in supernodes that contain data scatter it to other nodes within their supernodes

Cost Comparison

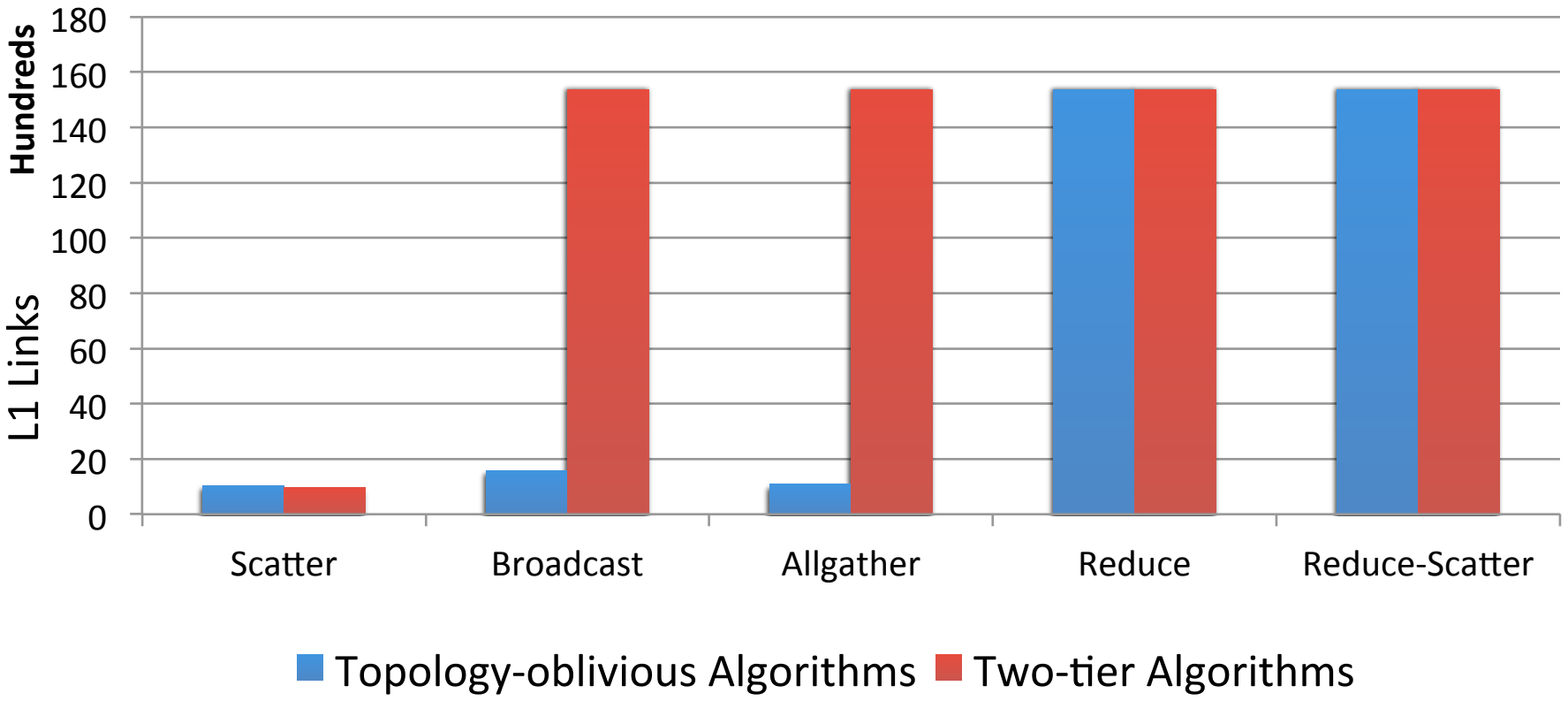
Operation	Base Cost	Two Tier Cost
Scatter	$\frac{p-1}{p}n\beta$	$n\beta * max\{\frac{1}{nps}, \frac{1}{sn}\}$
Gather	$\frac{p-1}{p}n\beta$	$n\beta * max\{\frac{1}{nps}, \frac{1}{sn}\}$
Allgather	$\frac{p-1}{p}n\beta$	$n\beta(\frac{1}{nps} + \frac{1}{sn} + \frac{1}{sn*nps})$
Broadcast	$2\frac{p-1}{p}n\beta$	$n\beta(\frac{3}{nps})$
Reduce-Scatter	$\frac{p-1}{p}(n\beta + n\gamma)$	$n\beta(\frac{1}{nps} + \frac{1}{sn} + \frac{1}{sn*nps}) + 2n\gamma$
Reduce	$\frac{p-1}{p}(2n\beta + n\gamma)$	$n\beta(\frac{1}{nps} + \frac{2}{sn}) + 2n\gamma$

Table 2. Cost Model based Comparison

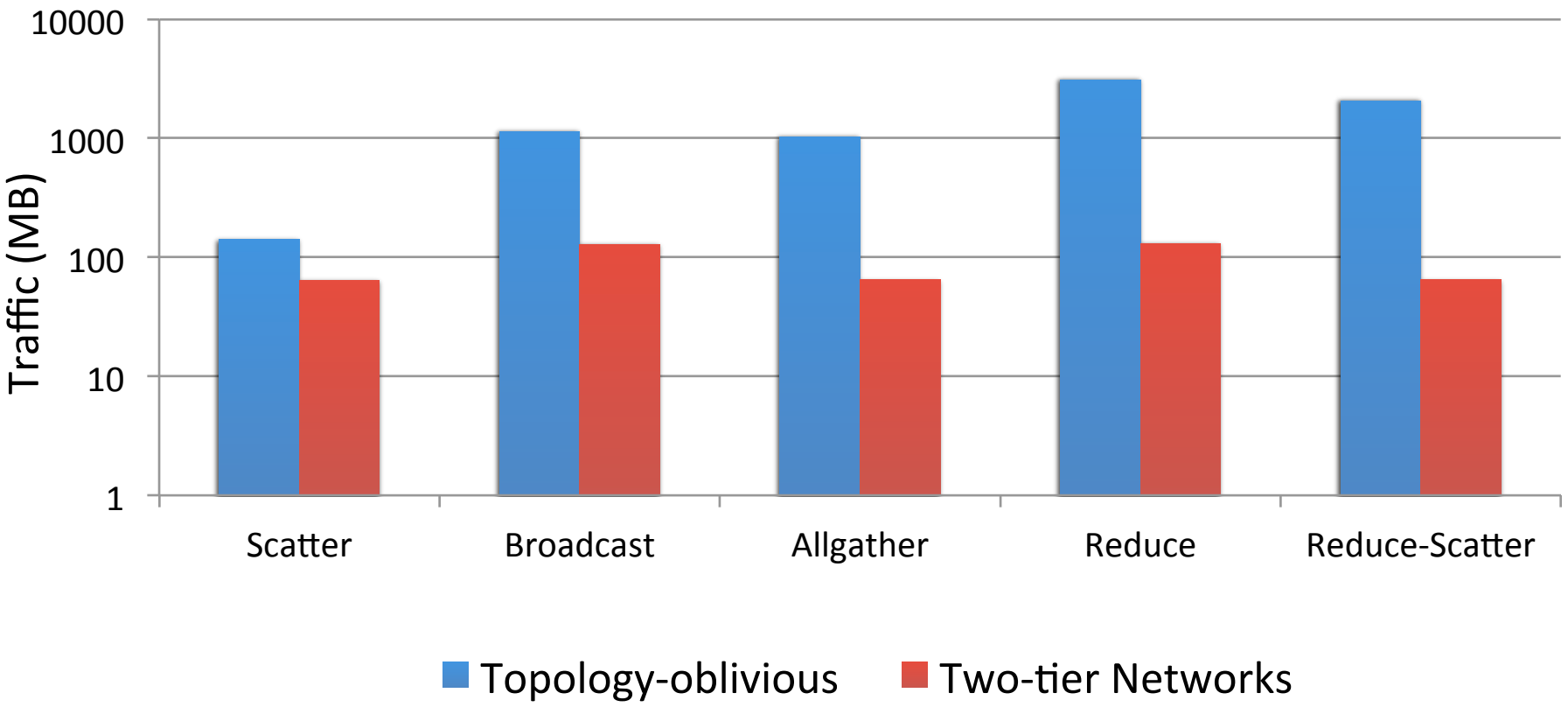
Experiments

- Rank-order mapping
- *pattern-generator* generates a list of communication exchange beween MPI ranks
- *linkUsage* generates the amount of traffic that will flow on each link in the given two-tier network
- 64 supernodes, nps = 16, cpn = 16
- 4032 L2 links and 15360 L1 links in the system

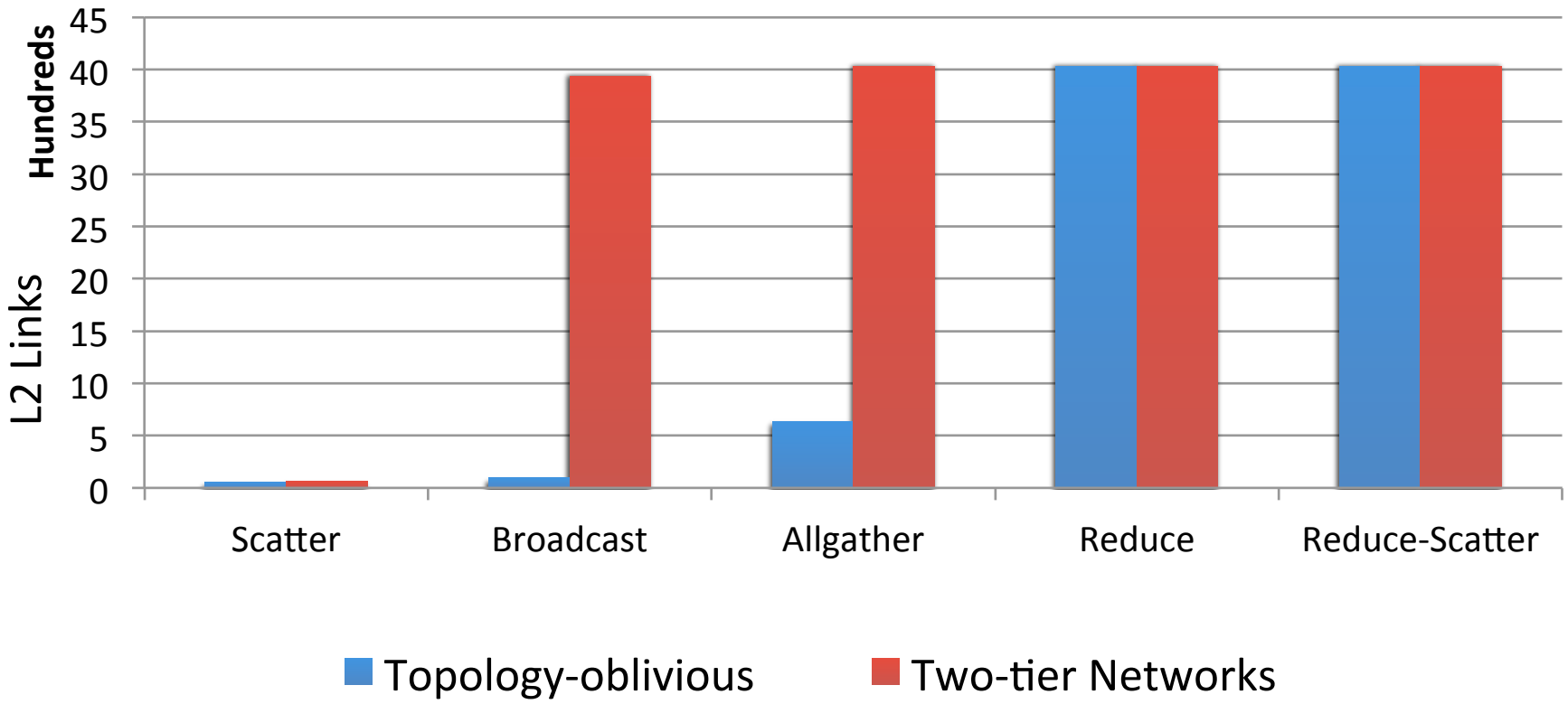
L1 Links Used



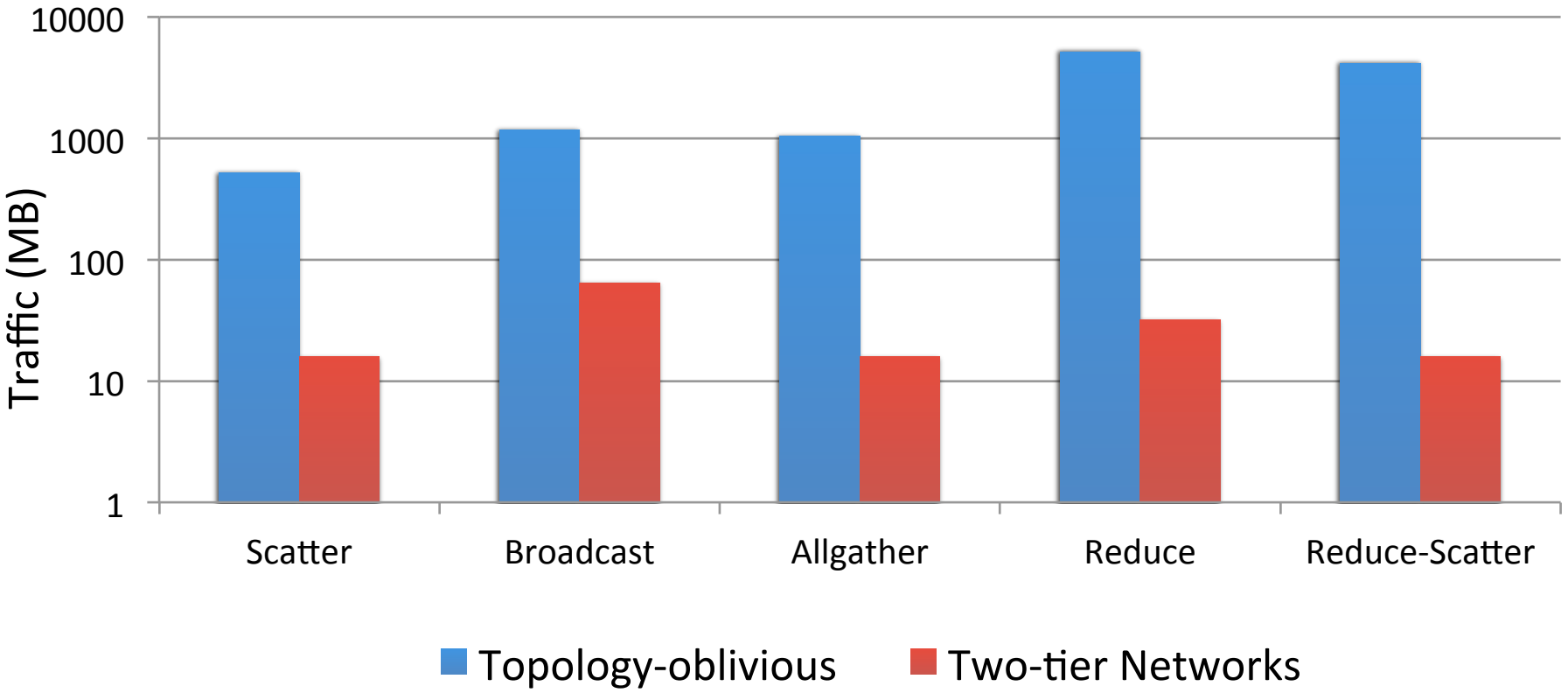
L1 Links Maximum Traffic



L2 Links Used



L2 Links Maximum Traffic



Conclusion and Future Work

- Proposed topology aware algorithms for large message collectives on two-tier direct networks
- Comparison based on cost model and analytical modeling promise good performance
- Implement these algorithms on a real system
- Explore short message collectives