# MPI Collectives Algorithms II

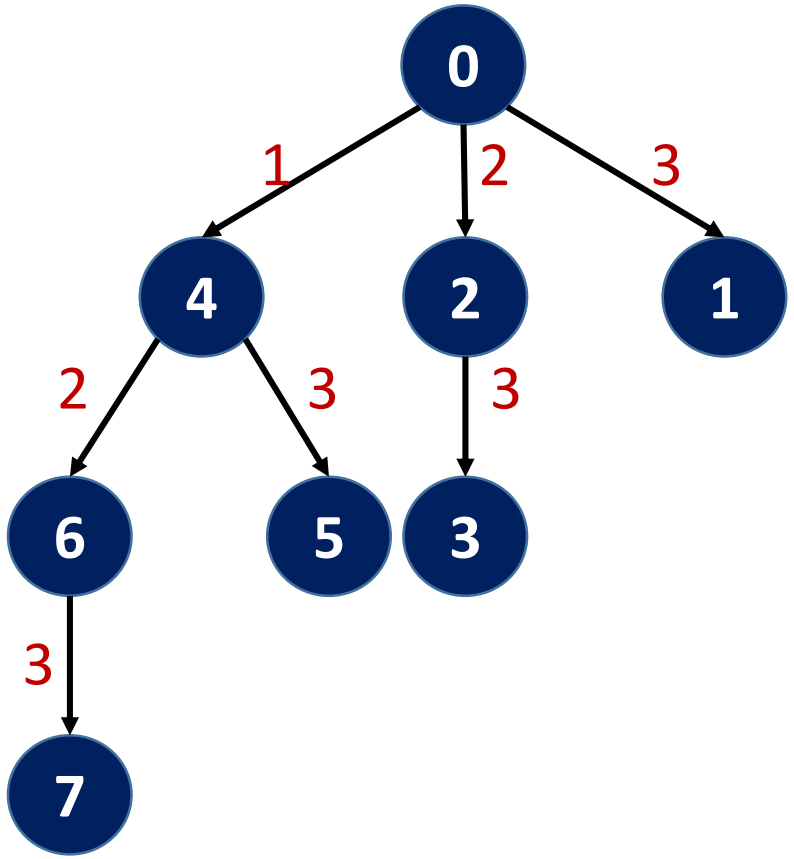
Mar 9, 2021

# Algorithms till now…

- MPI_Allgather
  - Ring
  - Recursive doubling
  - Bruck
- MPI_Scatter
  - Binomial
- MPI_Bcast
  - Binomial
  - Van de Geijn (MPI_Scatter + MPI_Allgather)

# Bcast

- int MPI_Bcast (buffer, count, datatype, root, comm)
- Sends and receives
- Eager protocol may play a role

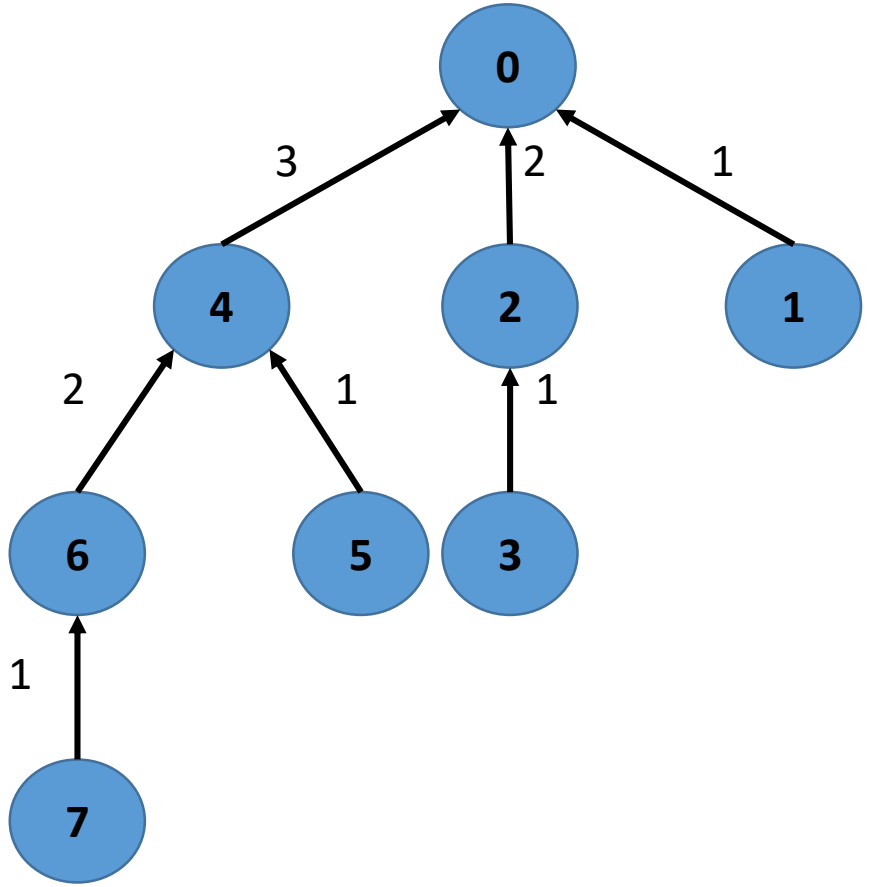


# Gather

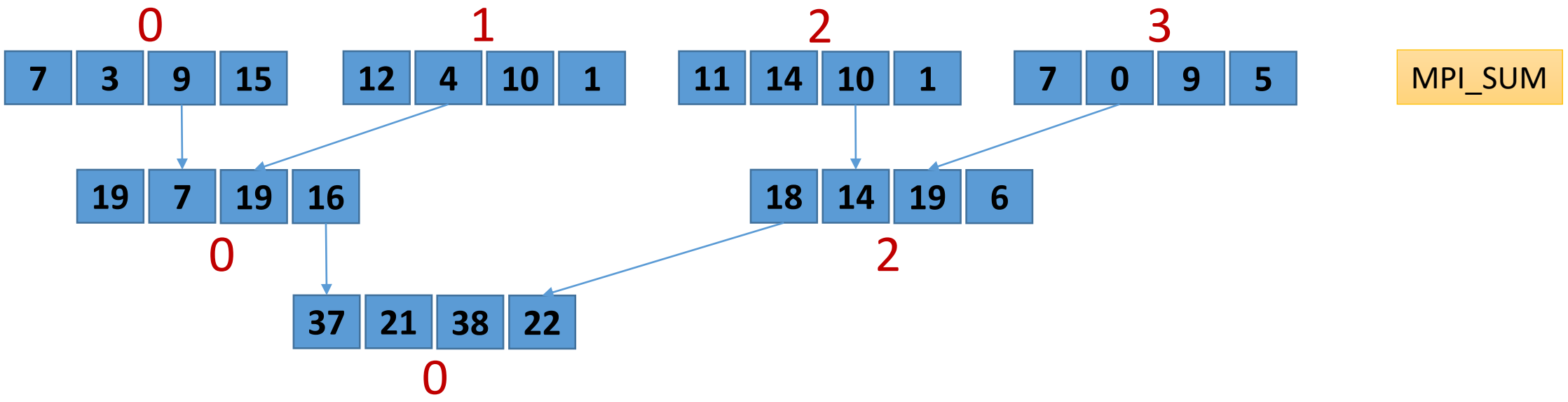
Binomial tree

Time: (log p)*L + n/p*(p-1)*(1/B)

L = latency, B = bandwidth

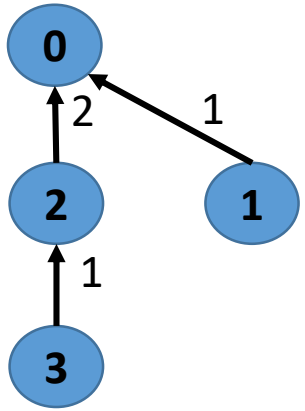


# Reduce – Recursive doubling

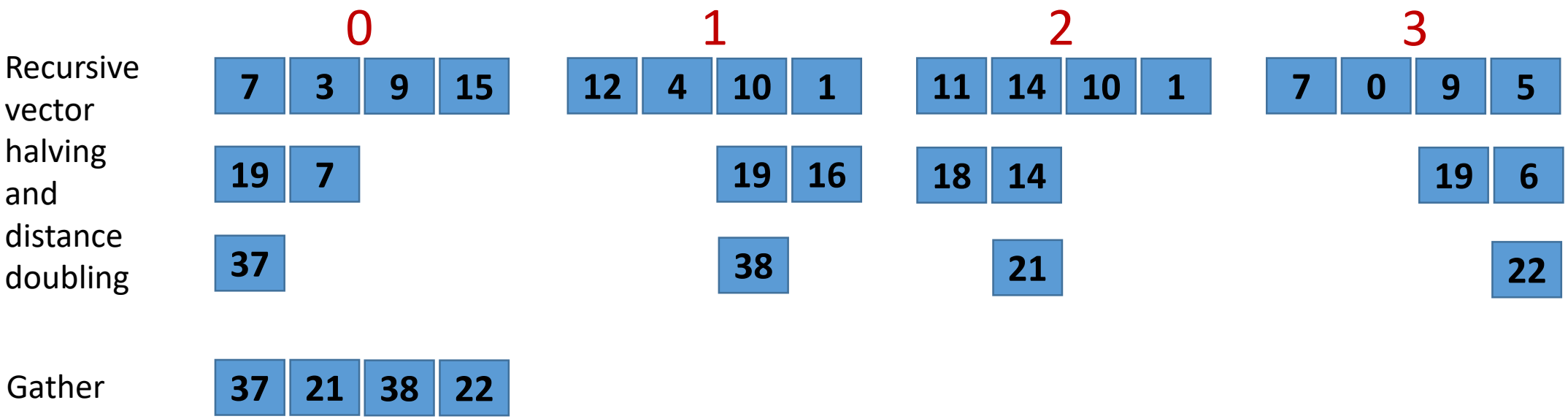



Time: log p (L + n*(1/B) + n*c)
L = latency, B = bandwidth
c = compute time per byte

*Used for short messages*



# Reduce – Rabenseifner's Algorithm




Time:

log p * L + (p-1)/p*(n/B) + (p-1)/p*n*c (reduce-scatter) +

log p * L + (p-1)/p*(n/B) (gather using binomial)

n = data size

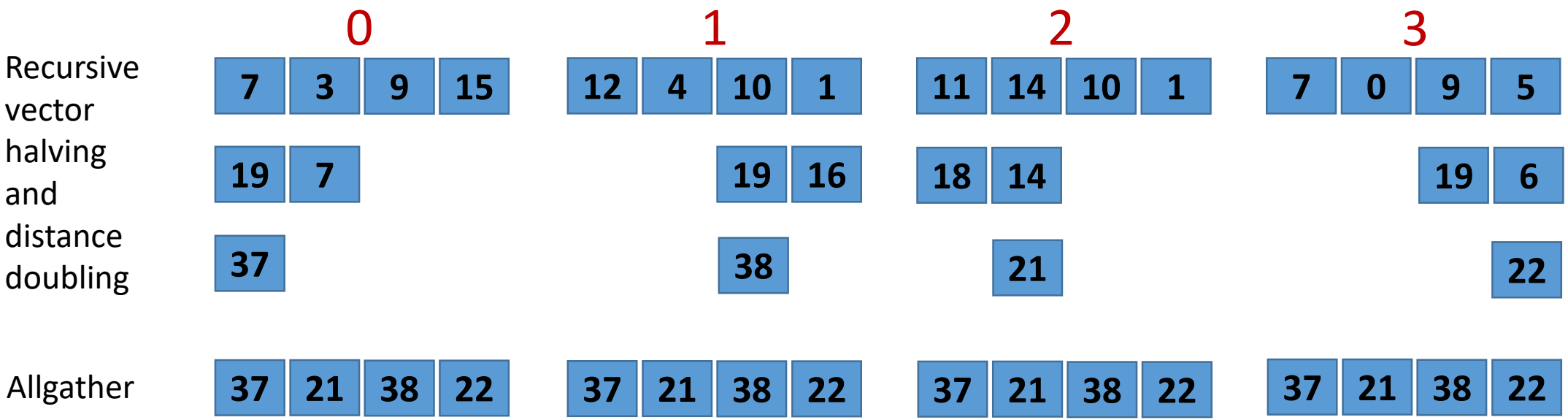
p = #processes

c = compute time

L = latency

B = bandwidth

# Allreduce – Rabenseifner's Algorithm




Time:

log p * L + (p-1)/p*(n/B) + (p-1)/p*n*c (reduce-scatter) +

log p * L + (p-1)/p*(n/B) (allgather using recursive vector doubling and distance halving)

n = data size L = latency

p = #processes B = bandwidth

c = compute time

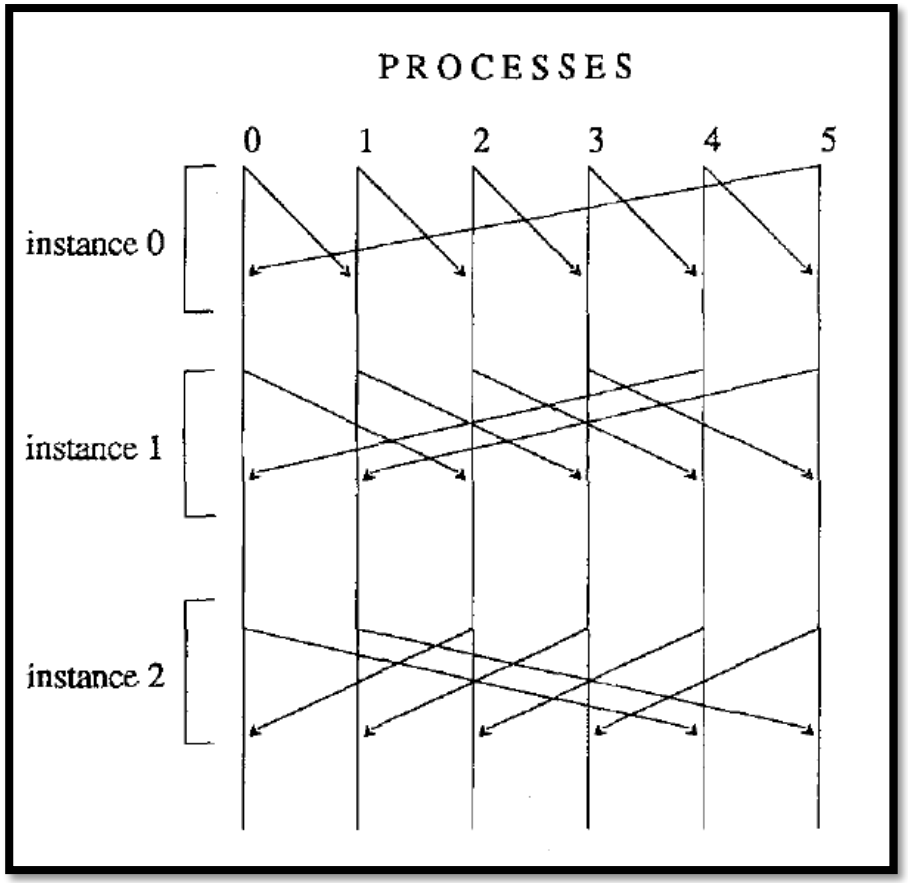
# Allreduce Algorithms – Summary

- Reduce (recursive doubling) followed by broadcast (binomial)
  - Time: [ log p (L + n*(1/B) + n*c) ] + [ log p (L + n*(1/B) ]

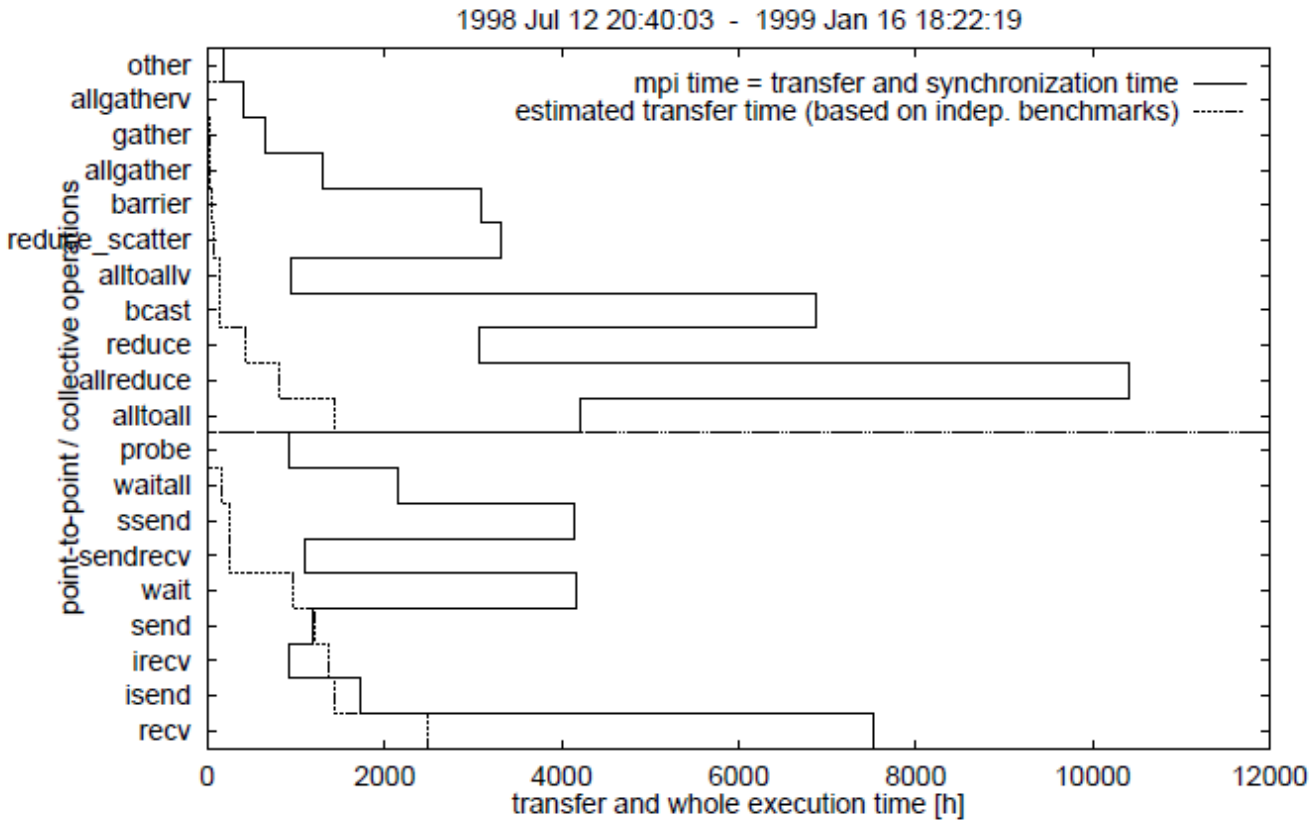
- Reduce-scatter followed by allgather (recursive doubling)
  - Time: 2 * log p * L + 2(p-1)/p * (n/B) + (p-1)/p * n * c

# Barrier

- Dissemination algorithm described in: Debra Hensgen et al., "Two Algorithms for Barrier Synchronization“, IJPP, 1988.
- In step k, 0 <= k <= (ceiling(lgp)-1), process i sends to process (i + 2^k) % p and receives from process (i - 2^k + p) % p
- Every process incorporates the message it received so far
- Uses ceiling(lg p) steps



# MPI Profiles

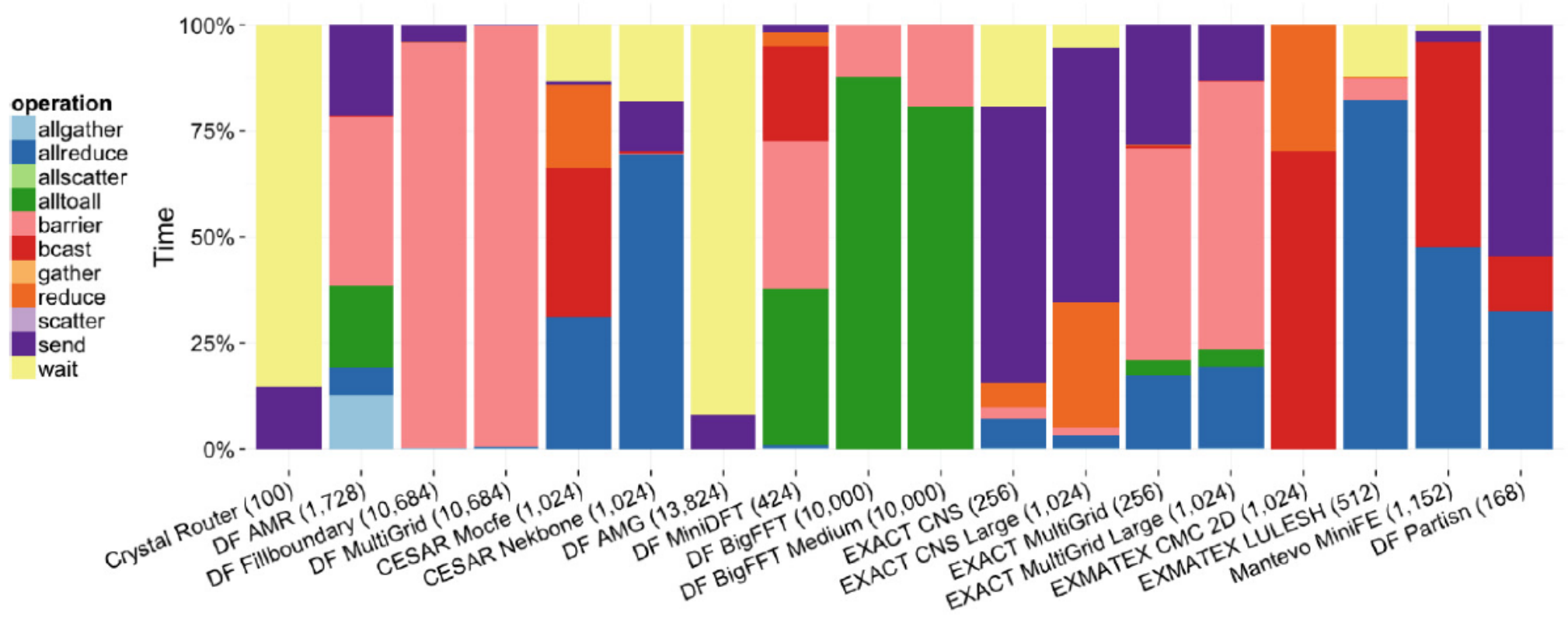



- 68 users have been profiled
- Total 425,288 node-hours on an Cray T3E system at HLRS, Stuttgart

| MPI routines' group | in ...% of all jobs | weighted by CPU time |
|---|---|---|
| blocking point to point | 66.304 % | 65.858 % |
| nonblocking pt-to-pt | 29.264 % | 22.007 % |
| persistent pt-to-pt | 0.010 % | 0.000 % |
| pack and unpack | 2.352 % | 0.118 % |
| collective communication | 92.545 % | 99.996 % |
| derived datatype | 23.530 % | 1.468 % |
| group and sub-communicator | 28.953 % | 8.945 % |
| inter-communicator | 0.029 % | 0.009 % |
| attribute caching / inquiry | 7.872 % | 0.459 % |
| error handler handling | 0.019 % | 0.000 % |
| topology creation | 13.568 % | 15.685 % |
| wtime measurement | 24.521 % | 12.671 % |

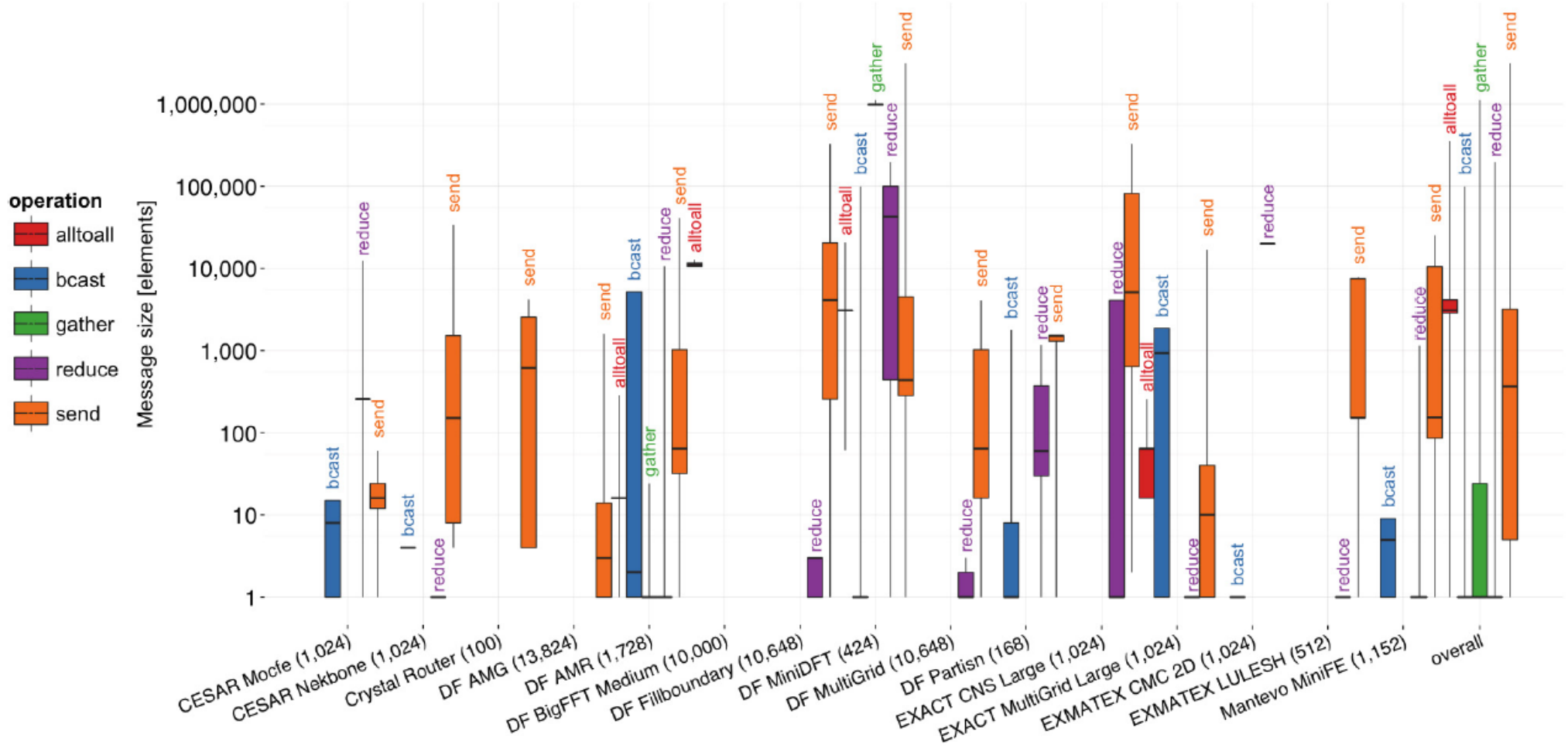
[Source: Rabenseifner et al., Automatic MPI Counter Profiling, CUG 2000]

# MPI Communication Times Survey



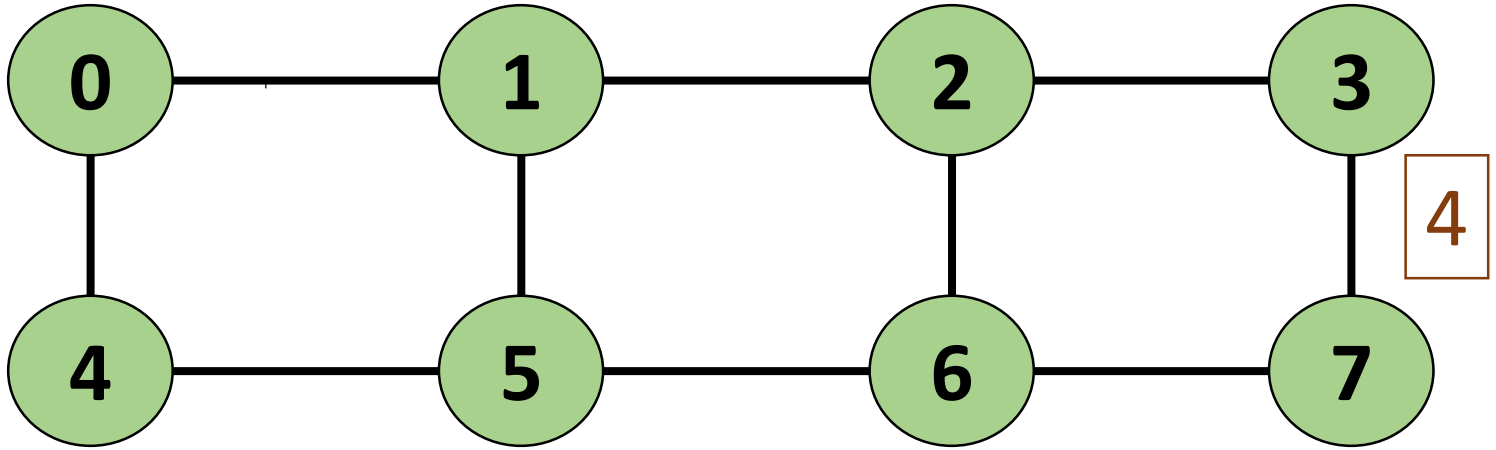

Source: Klenk and Froning, An Overview of MPI Characteristics of Exascale Proxy Applications

# Message Size Variation

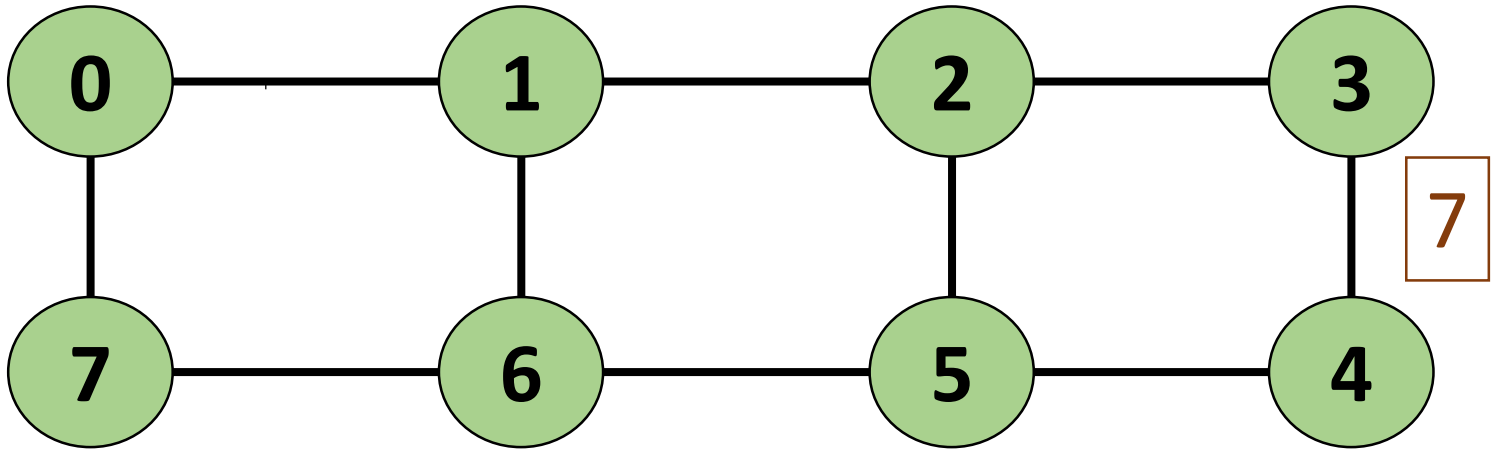



Source: Klenk and Froning, An Overview of MPI Characteristics of Exascale Proxy Applications

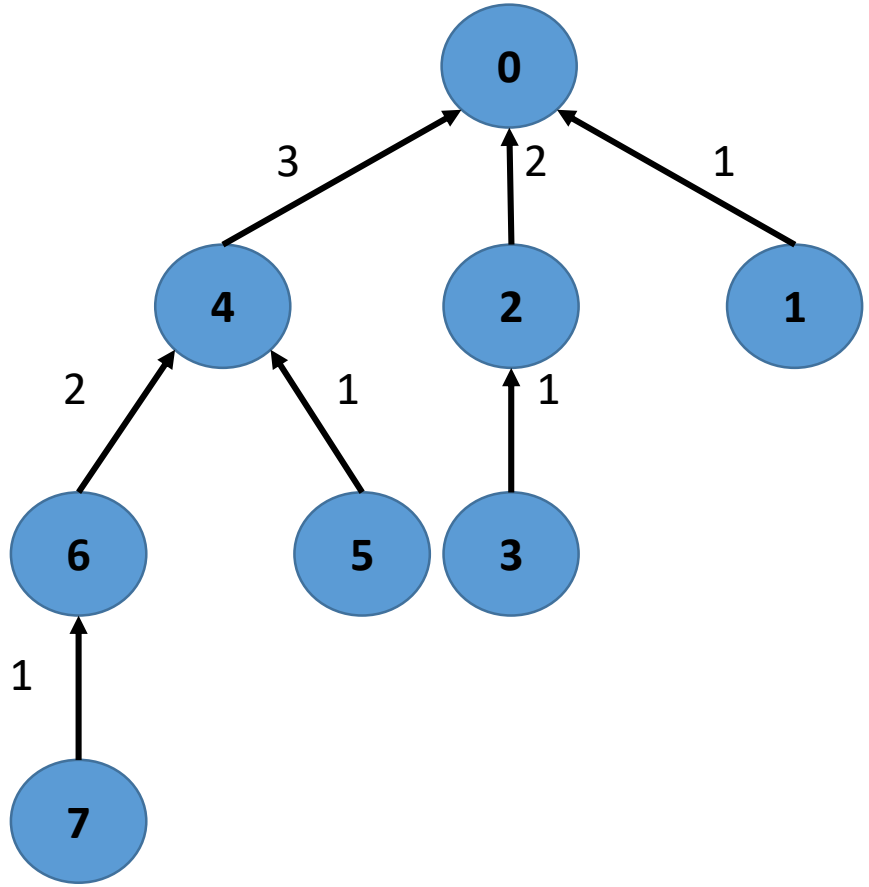
# Effect of Network Topology



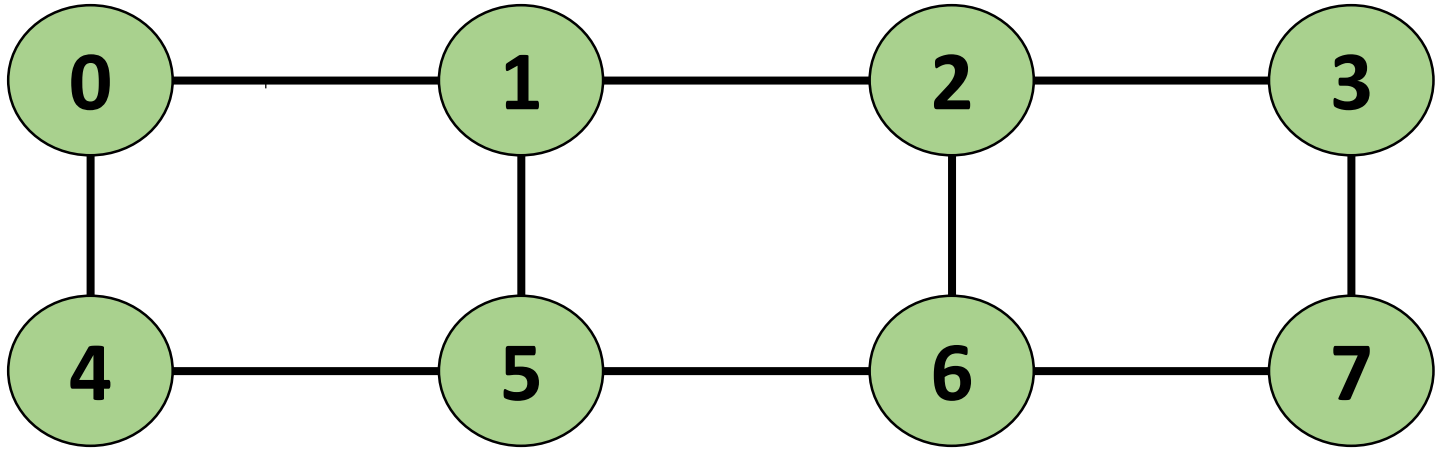

4

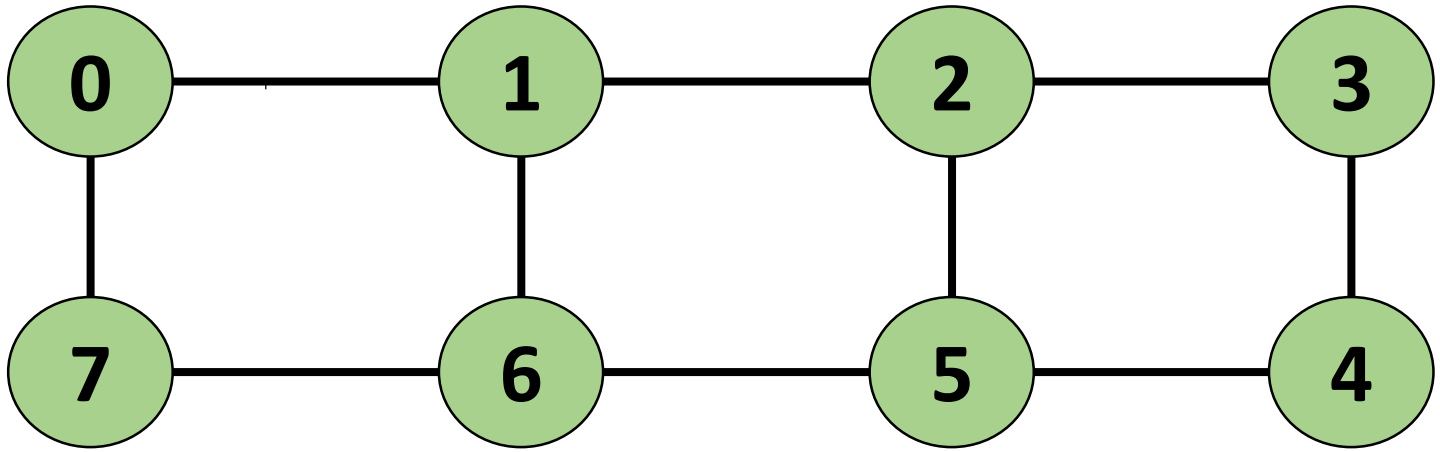



7



# Effect of Network Topology



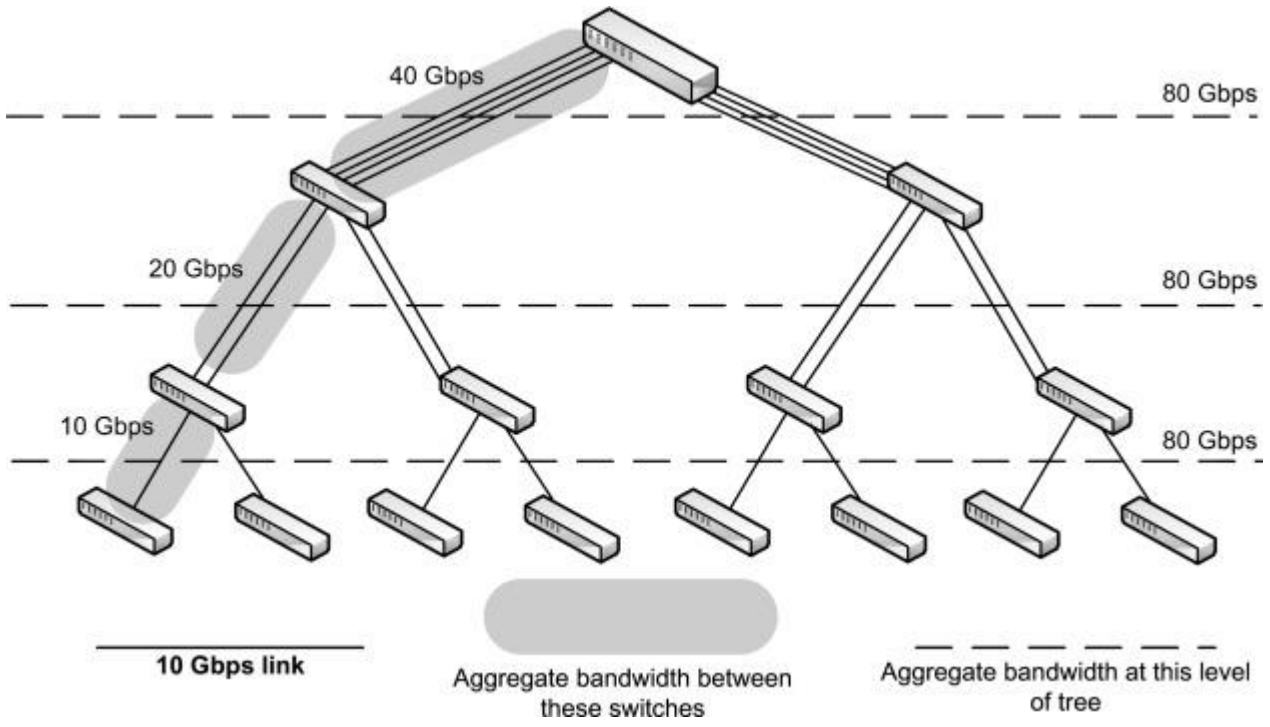



# Designing Topology-Aware Collective Communication Algorithms for Large Scale InfiniBand Clusters: Case Studies with Scatter and Gather

Krishna Kandalla, Hari Subramoni, Abhinav Vishnu and Dhabaleswar K. (DK) Panda

IPDPS Workshops 2010

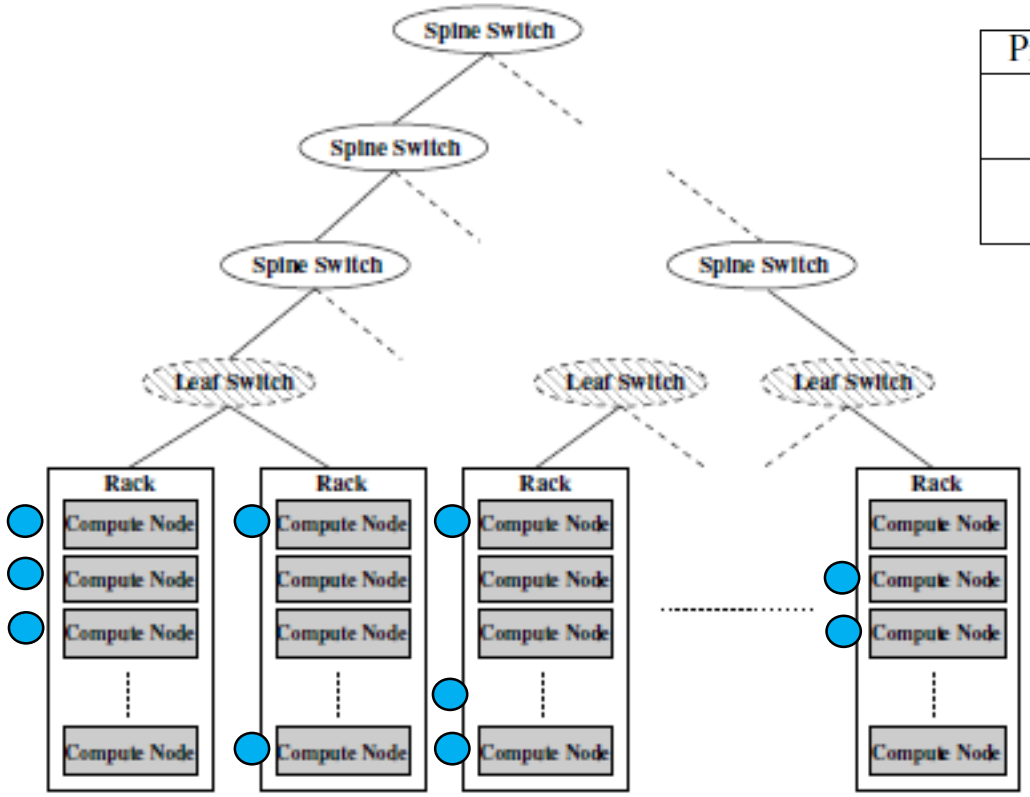
# Network Topology




Fat-tree topology

[www.sciencedirect.com]

# Effect of Topology on Latency




A typical topology of large-scale systems

(TACC Ranger system)

| Process Location | | Number of Hops | MPI Latency (*us*) |
|---|---|---|---|
| Intra-Rack | Intra-Chassis | 0 Hops in Leaf Switch | 1.57 |
| | Inter-Chassis | 1 Hop in Leaf Switch | 2.04 |
| Inter-Rack | | 3 Hops Across Spine Switch | 2.45 |
| | | 5 Hops Across Spine Switch | 2.85 |

- Contiguous node allocation may increase queue waiting times
- Allocated nodes are usually scattered in the system
- Different job request sizes and durations

# Infiniband

- New networking standard
- High throughput, low latency
- Provides good scalability
- Transmits packets of up to 4 KB size
- Higher link widths support >100Gb/s bi-directional bandwidth
- SDR, DDR, QDR, FDR, …
- Supports RDMA
- Used in 36% of top 500 supercomputers