MPI vs BitTorrent : Switching Between Large-Message Broadcast Algorithms in the Presence of Bottleneck Links

Kiril Dichev
Kiril.Dichev@ucdconnect.ie

Alexey Lastovetsky
Alexey.Lastovetsky@ucd.ie

Heterogeneous Computing Laboratory
http://hcl.ucd.ie

HeteroPar'2012
August 27, 2012
Rhodes Island, Greece

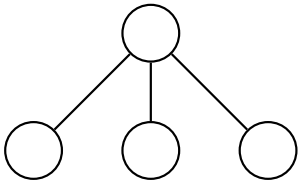
Motivation

- All collectives in MPI are tree-based
  - On heterogeneous networks, efficient communication trees are needed
  - Construction of such communication trees complex
  - Tree-based collectives make sense for small messages
  - . . . But we question them for large messages
- **Objective**: Minimize the total runtime of large-message broadcasts on heterogeneous networks
- **Approach**: Examine algorithms both from HPC and distributed computing

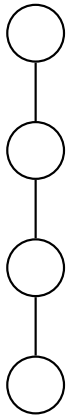
Outline

Outline

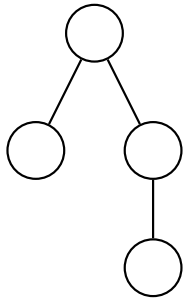
The MPI Way of Broadcasting Messages



flat



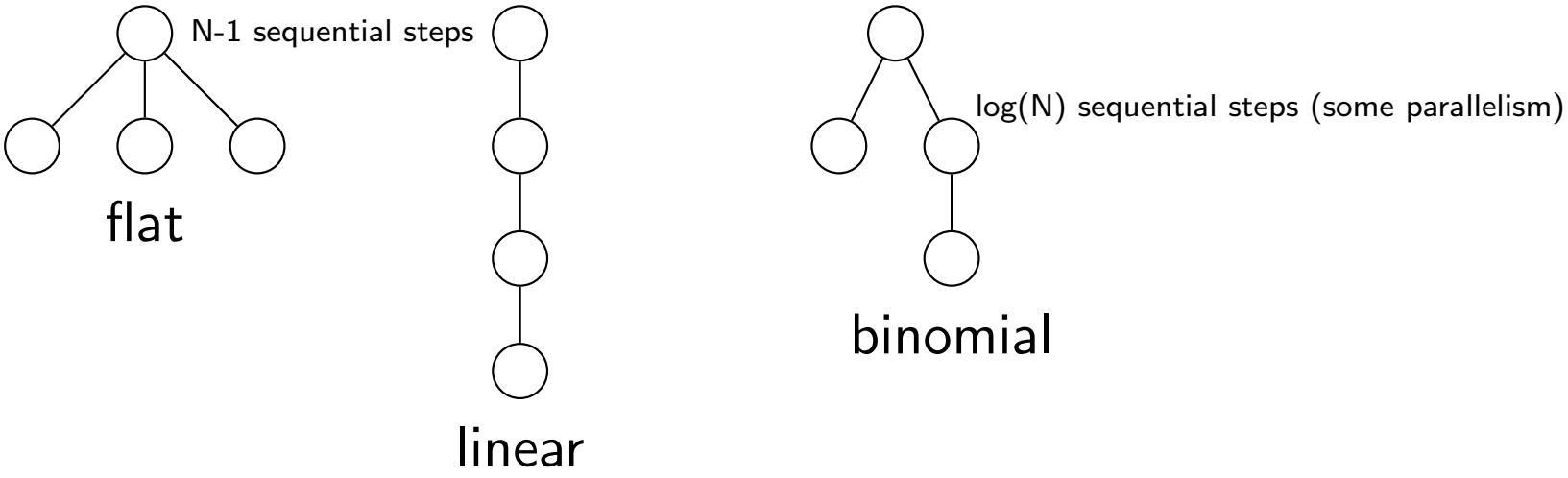
linear



binomial

- MPI always uses trees to schedule a broadcast
- Examples of trees: flat, linear or binomial tree

Complexity for Small-Message Broadcasts



- Process number N and message size M determine complexity
- Typical broadcast complexity:
  - $O(M * log(N))$ (binomial tree)
  - $O(M * N)$ (linear/flat tree)

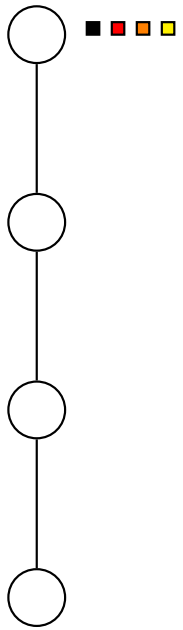
Pipelining in Large-Message MPI Broadcasts

- Complexity differs for large M and moderate N: O(M)
- Reason: Pipelining of fragmented message
- Related research [1]:
  - Trees with small nodal degree best for pipelined broadcasts
  - Linear tree is best tree-based algorithm

[1] Patarasuk, P. Faraj, A., Yuan, X.: Pipelined broadcast on Ethernet switched clusters. In: Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 29th International. p. 10 pp. (April 2006)

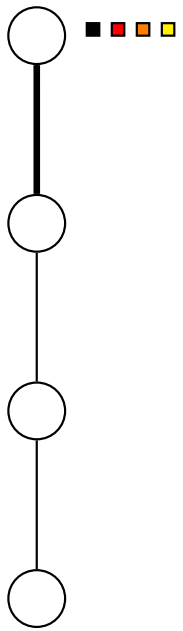
Pipelining in Large-Message MPI Broadcasts

Pipelined linear algorithm
(Open MPI)



Pipelining in Large-Message MPI Broadcasts

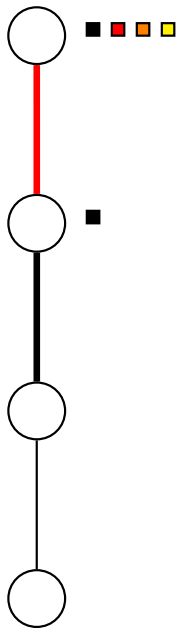
Pipelined linear algorithm
(Open MPI)



Step 1

Pipelining in Large-Message MPI Broadcasts

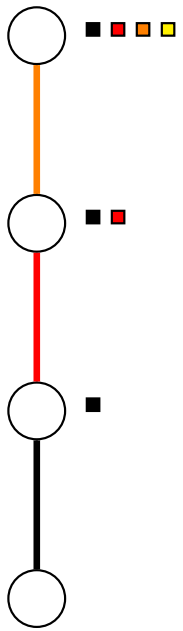
Pipelined linear algorithm
(Open MPI)



Step 2

Pipelining in Large-Message MPI Broadcasts

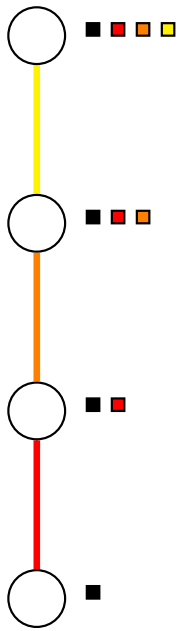
Pipelined linear algorithm
(Open MPI)



Step 3

Pipelining in Large-Message MPI Broadcasts

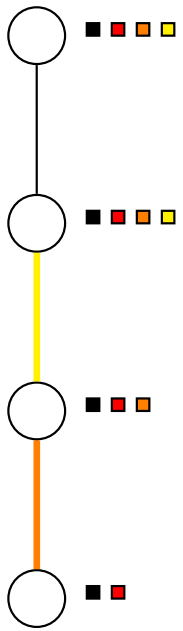
Pipelined linear algorithm
(Open MPI)



Step 4

Pipelining in Large-Message MPI Broadcasts

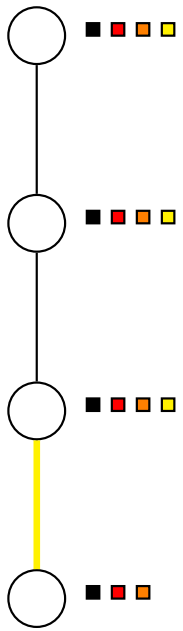
Pipelined linear algorithm
(Open MPI)



Step 5

Pipelining in Large-Message MPI Broadcasts

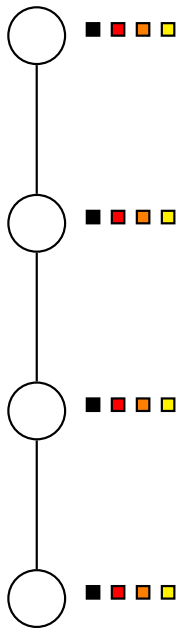
Pipelined linear algorithm
(Open MPI)



Step 6

Pipelining in Large-Message MPI Broadcasts

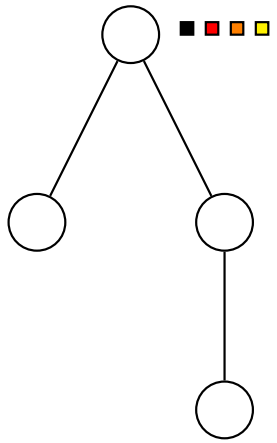
Pipelined linear algorithm
(Open MPI)



Done after 6 steps

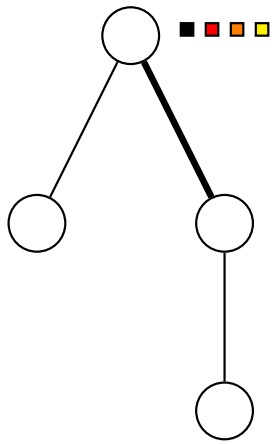
Pipelining in Large-Message MPI Broadcasts

Binomial scatter / Ring allgather algorithm (MPICH2)



Pipelining in Large-Message MPI Broadcasts

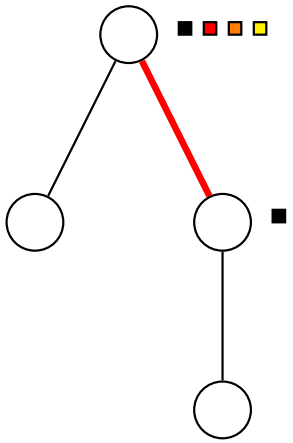
Binomial scatter / Ring allgather algorithm (MPICH2)



Step 1

Pipelining in Large-Message MPI Broadcasts

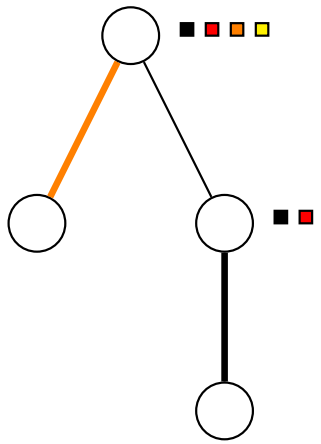
Binomial scatter / Ring allgather algorithm (MPICH2)



Step 2

Pipelining in Large-Message MPI Broadcasts

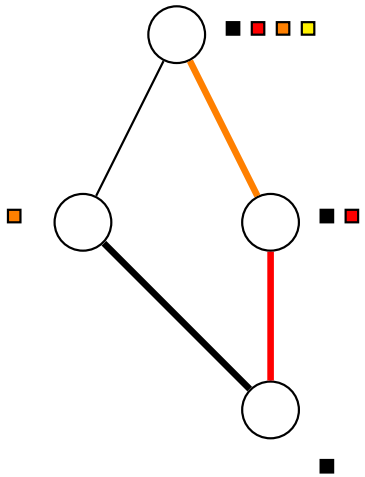
Binomial scatter / Ring allgather algorithm (MPICH2)



Step 3

Pipelining in Large-Message MPI Broadcasts

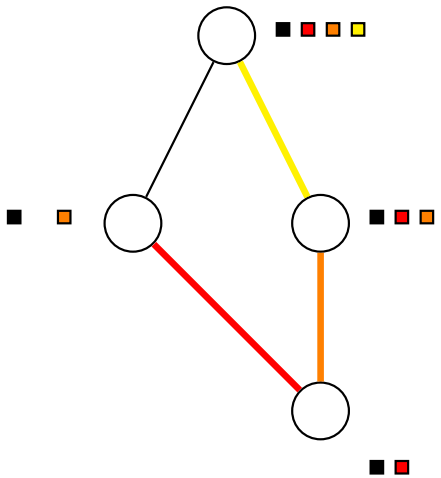
Binomial scatter / Ring allgather algorithm (MPICH2)



Step 4

Pipelining in Large-Message MPI Broadcasts

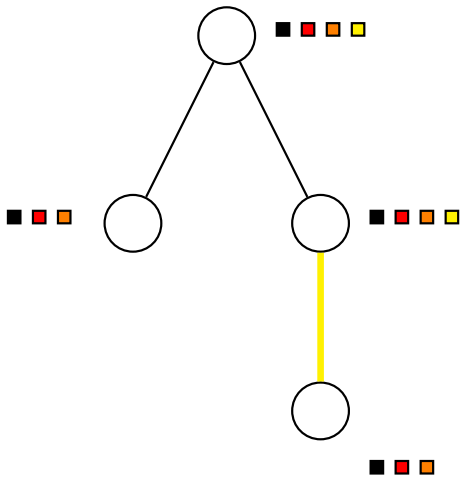
Binomial scatter / Ring allgather algorithm (MPICH2)



Step 5

Pipelining in Large-Message MPI Broadcasts

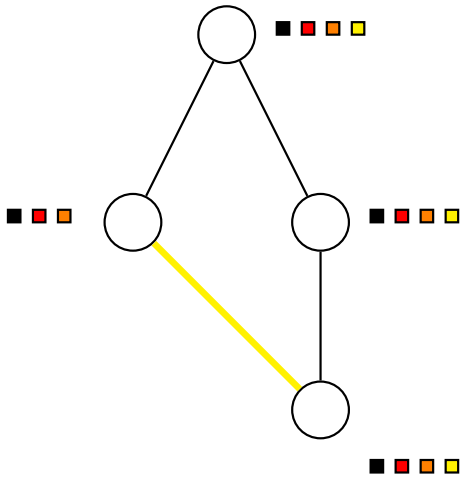
Binomial scatter / Ring allgather algorithm (MPICH2)



Step 6

Pipelining in Large-Message MPI Broadcasts

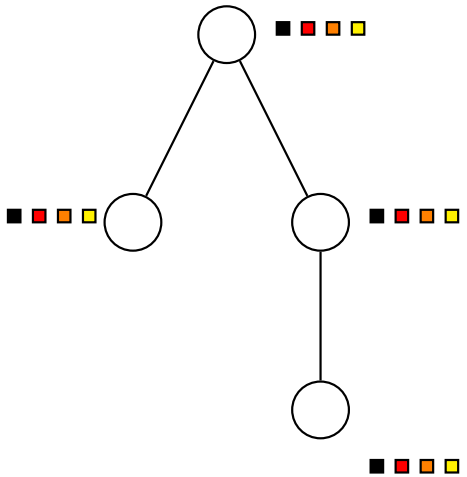
Binomial scatter / Ring allgather algorithm (MPICH2)



Step 7

Pipelining in Large-Message MPI Broadcasts

Binomial scatter / Ring allgather algorithm (MPICH2)



Done after 7 steps

Outline

Overview of BitTorrent Protocol

- Protocol invented by Bram Cohen[2]
- Various objectives related to peer-to-peer systems – high-performance not central
- Non-determinism, randomness, and unpredictability
- Hard to analyze complexity
- Early efforts rely on practical observations
- Some work[3] suggests complexity O(M) in wide-area networks

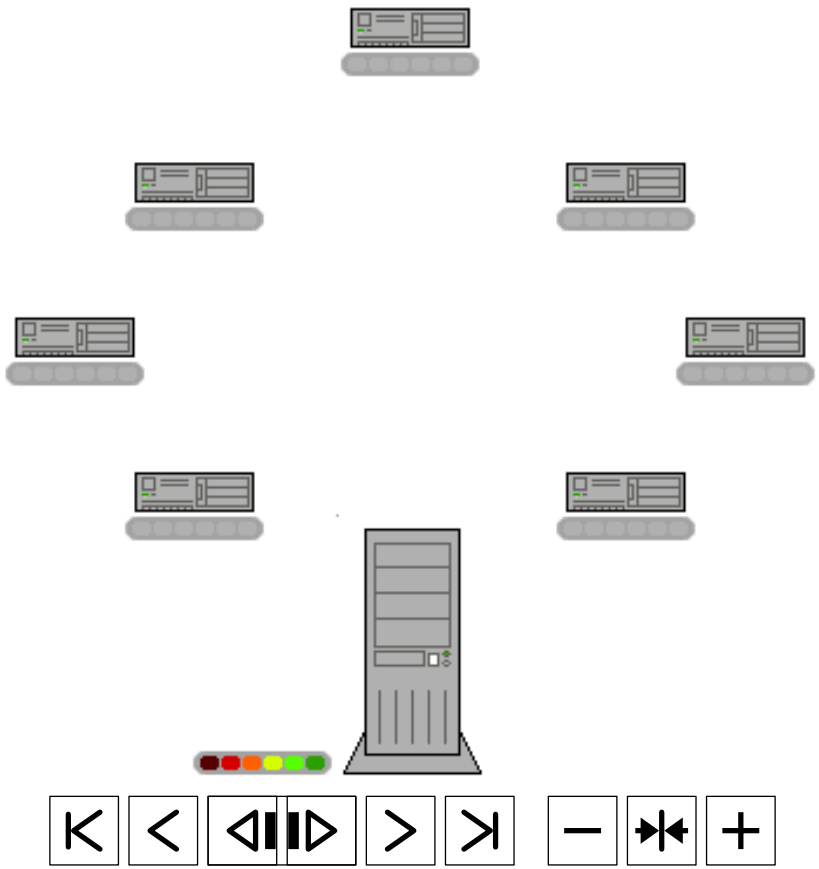
[2] Cohen, B.: Incentives build robustness in BitTorrent (2003)

[3] Izal, M., Urvoy-Keller, G., Biersack, E., Felber, P., Al Hamra, A., Garcs-Erice, L.: Dissecting BitTorrent: Five months in a torrents lifetime.

BitTorrent – A Different Algorithm

Example

Source: Wikipedia



Related Work

- Independently and in distributed computing, BitTorrent-inspired research in Vrije Universiteit [4]
- Communication libraries shown to perform better than an optimized MPI library on emulated wide-area networks
- No attempts to use BitTorrent on HPC clusters

[4] Burger, M.d.: High-throughput multicast communication for grid applications. Ph.D. thesis, Vrije Universiteit Amsterdam (2009)

Outline

Experimental Setup

- Original BitTorrent client written by Bram Cohen used
- Minor modifications (remove I/O and send dummy data, add simple profiling)
- BitTorrent clients used as MPI programs

Experimental Setup (2)

- We use the Bordeaux clusters of Grid'5000 for our experiments
- 'Bordeplage' cluster is connected through bottleneck link to the other clusters:
  - Single 1 Gigabit link for all inter-cluster communication
  - Larger latency through traversal of more switches

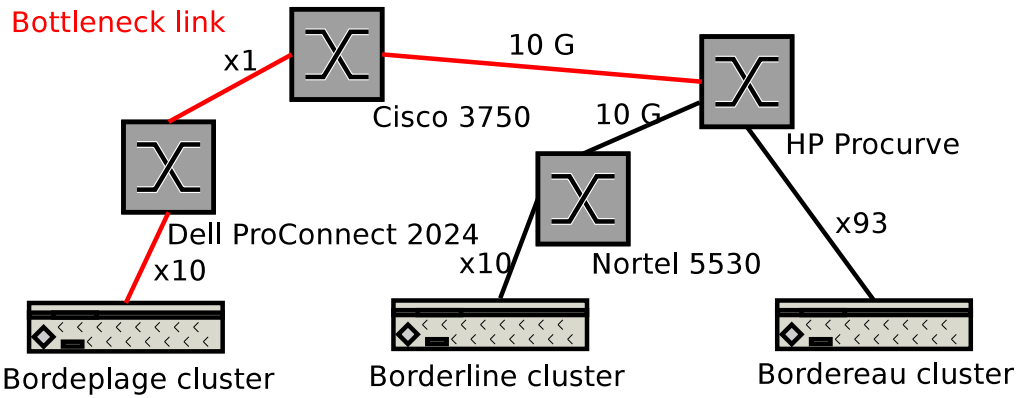
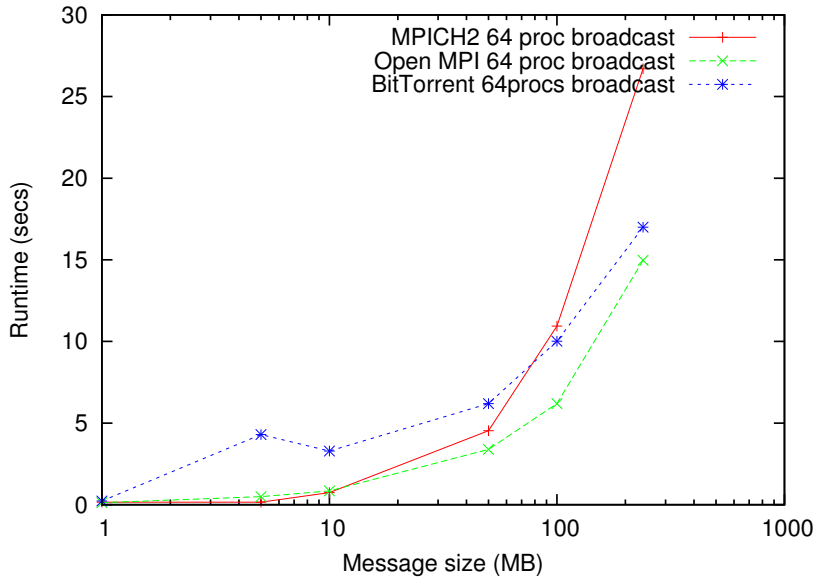


Figure: Topology of the Ethernet network on Bordeaux site

Results
Single Cluster

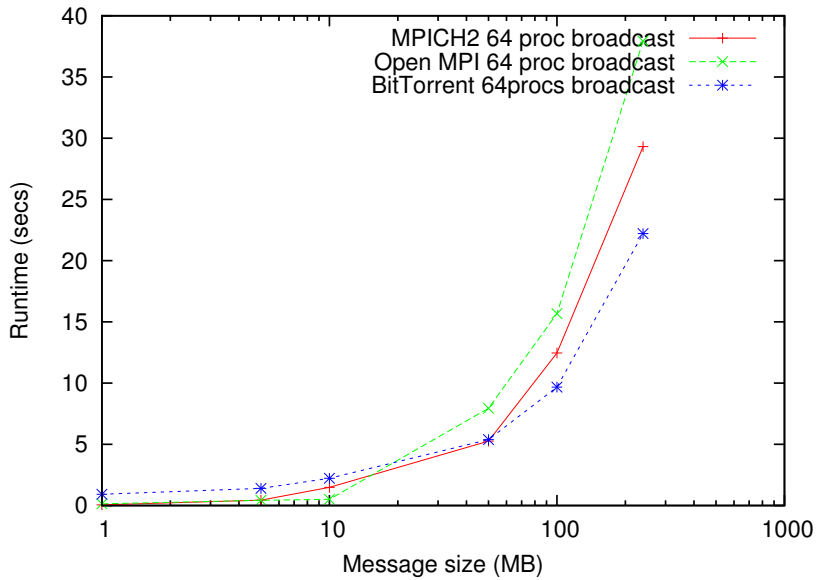
- 64 nodes on a single cluster
- The linear tree algorithm holds its "promise" – best algorithm
- BT is better than expected - better than MPICH2 for messages larger than 50 MB



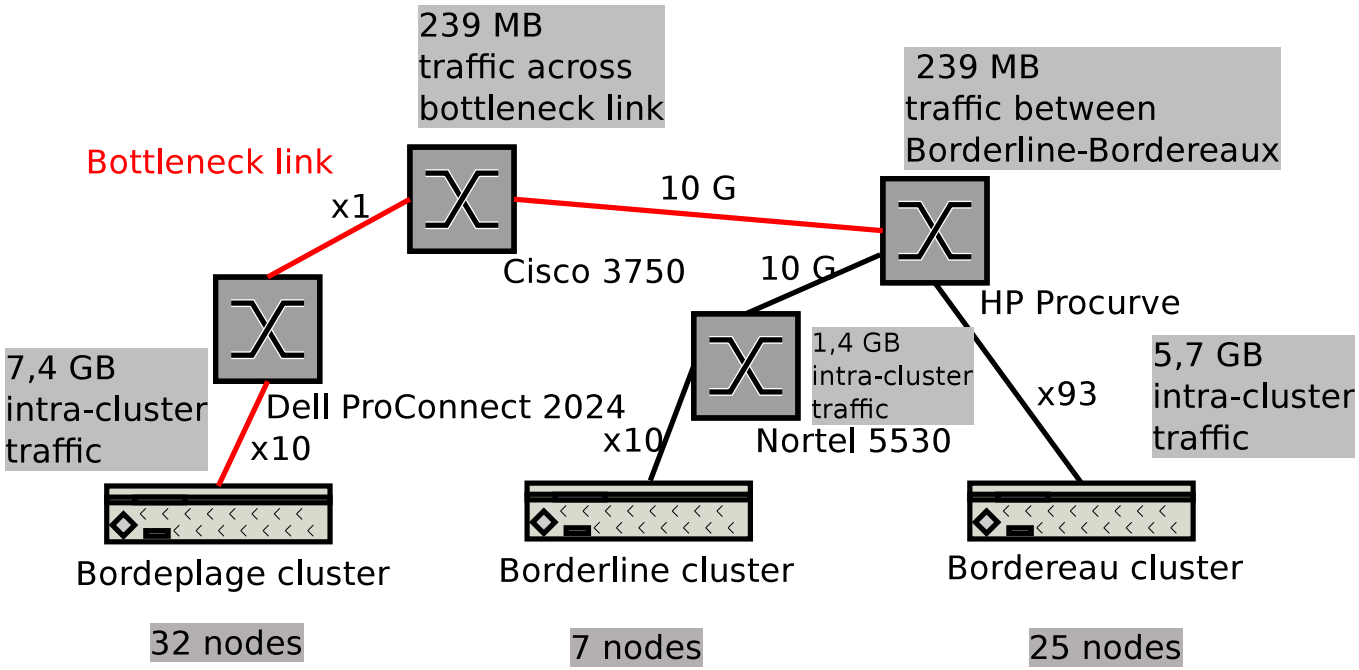
Results

Involving the Bottleneck Link

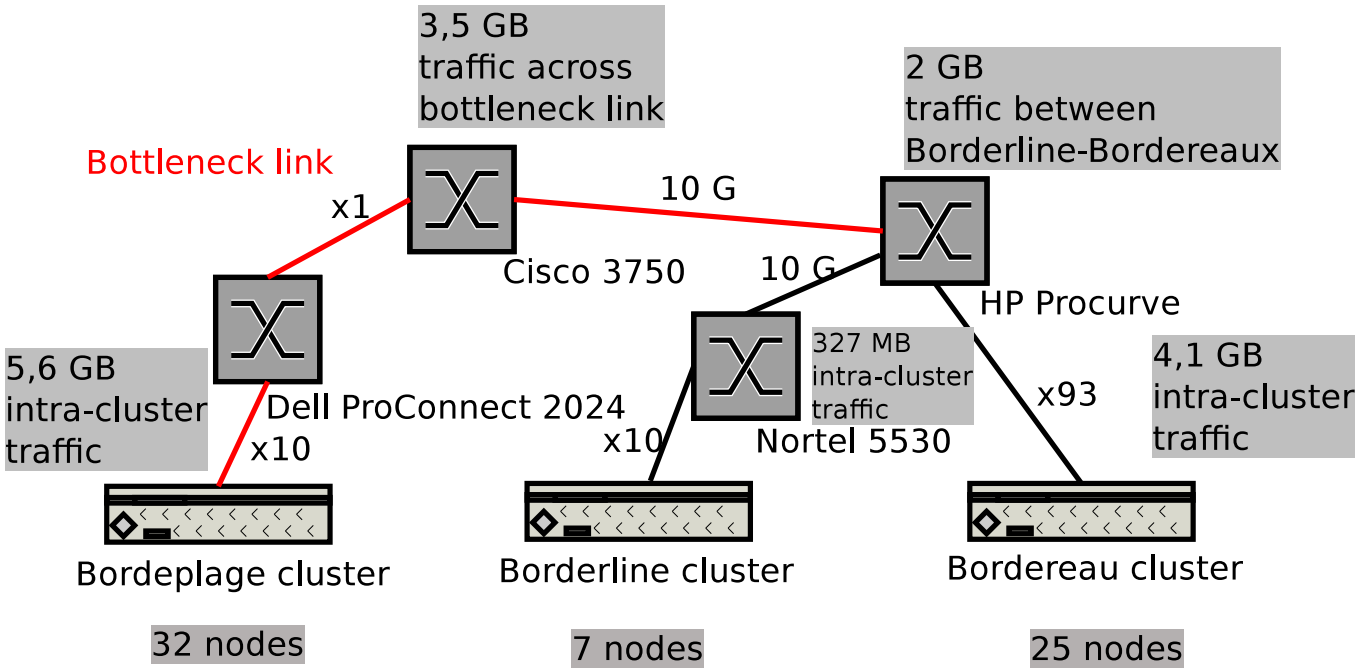
- 64 nodes across the 3 clusters (involving the single bottleneck link)
- As expected, all algorithms are slower
- Linear tree with less than half the original throughput
- BT is the best algorithm for message size of 50 MB or larger



Data Movement in 239 MB Broadcast - Open MPI



Data Movement in 239 MB Broadcast - BitTorrent



Pros of BitTorrent Compared to MPI

- Many parallel connections seem to improve the protocol
- Oblivious of network topology
- Near the optimal MPI algorithm for very large messages on homogeneous networks
- Better than MPI for messages larger than 50 MB on moderately heterogeneous networks
- BT results stable (all results show average, not minimum)

Contras of BitTorrent Compared to MPI

- High performance not the main objective of original protocol
- Good network utilization particular to large-message broadcasts
- Analysis of complexity is difficult

Conclusion

- In this work, we prove that BitTorrent-based collectives can be used in HPC
- In related work, we show that BitTorrent can solve other HPC-related problems as well [5]

[5] Dichev, K., Reid, F., Lastovetsky, A.: Efficient and reliable network tomography in heterogeneous networks using BitTorrent broadcasts and clustering algorithms. To be published in SC'12

Questions?