Implementation and Analysis of Nonblocking Collective Operations on SCI Networks

Christian Kaiser

Torsten Hoefler

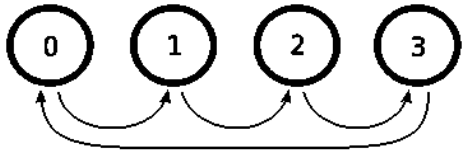
Boris Bierbaum, Thomas Bemmerl

LEHRSTUHL FÜR BETRIEBSSYSTEME
Univ.-Prof. Dr. habil. Thomas Bemmerl

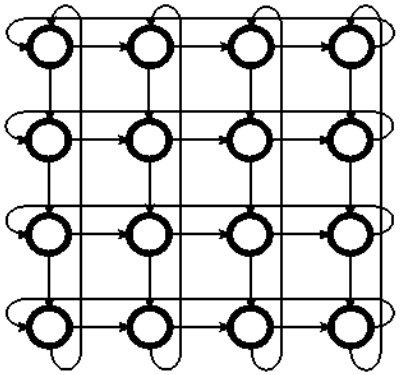
RWTH AACHEN UNIVERSITY

Scalable Coherent Interface (SCI)

Ringlet:

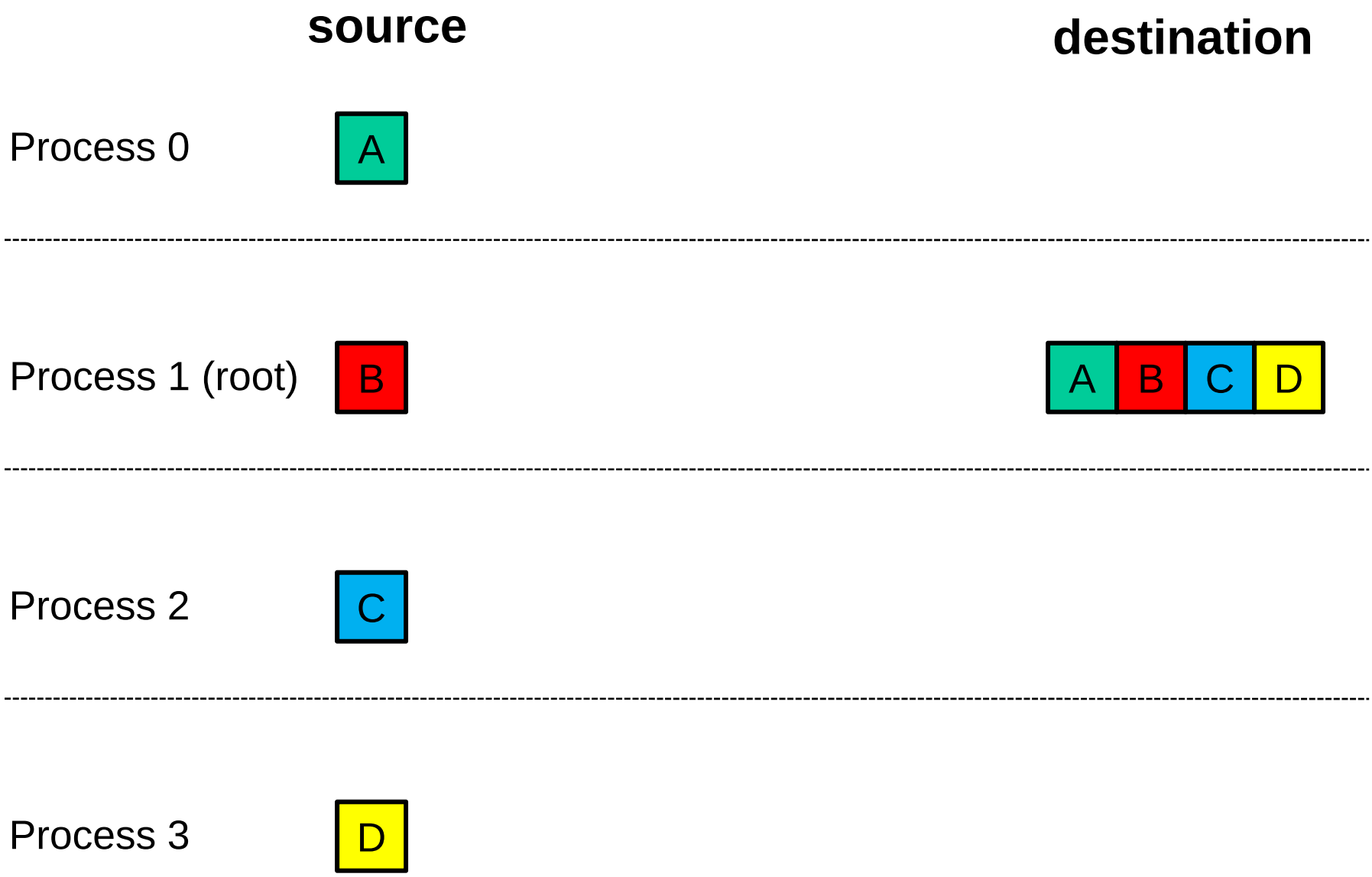


2D Torus:

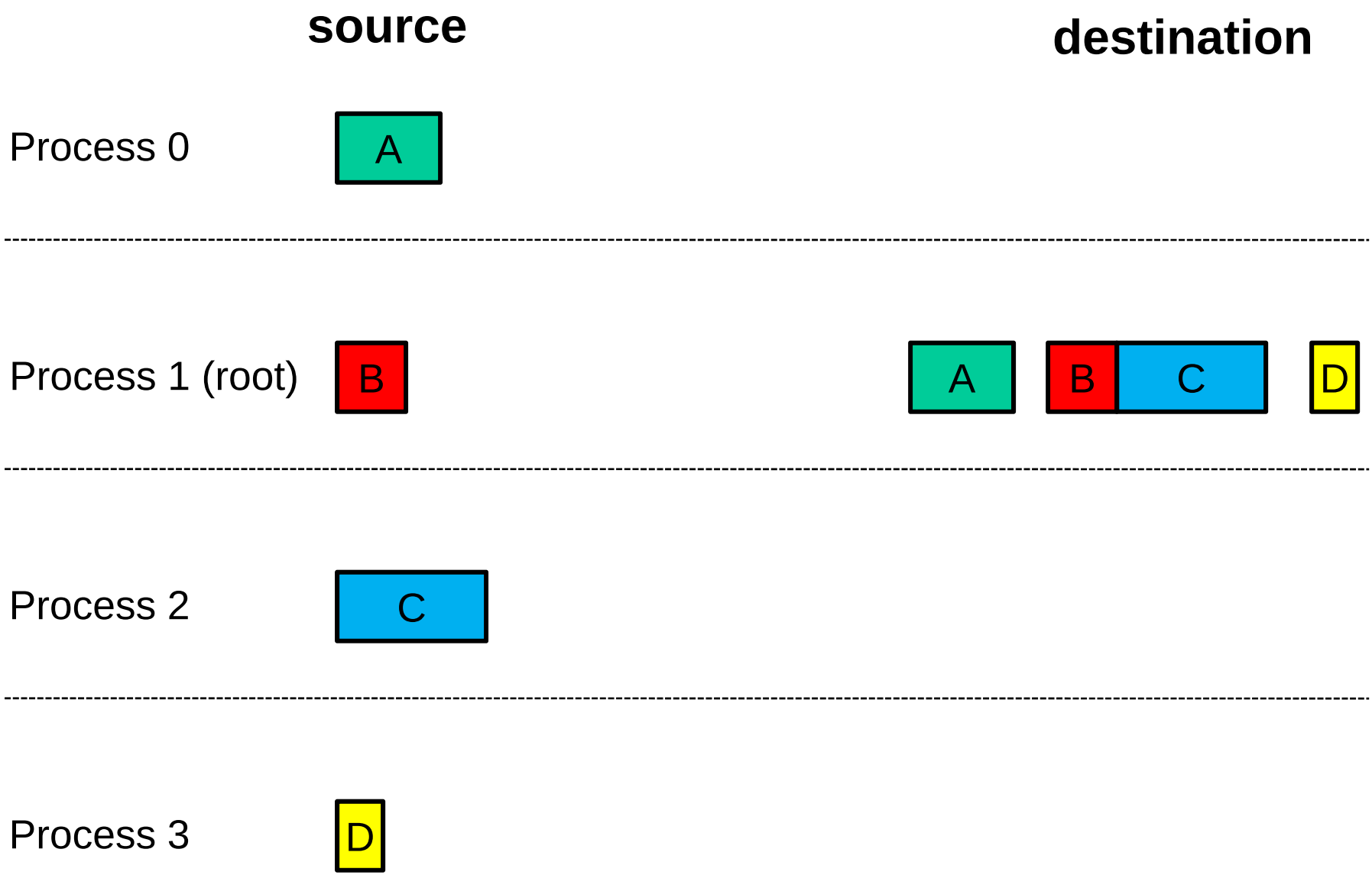


- IEEE Std 1596-1992
- Memory Coupled Clusters
- Data Transfer: PIO and DMA
- SISCI User-Level Interface
- 16 x Intel Pentium D, 2.8 GHz
- SCI: D352 (IB: Mellanox DDR x4)

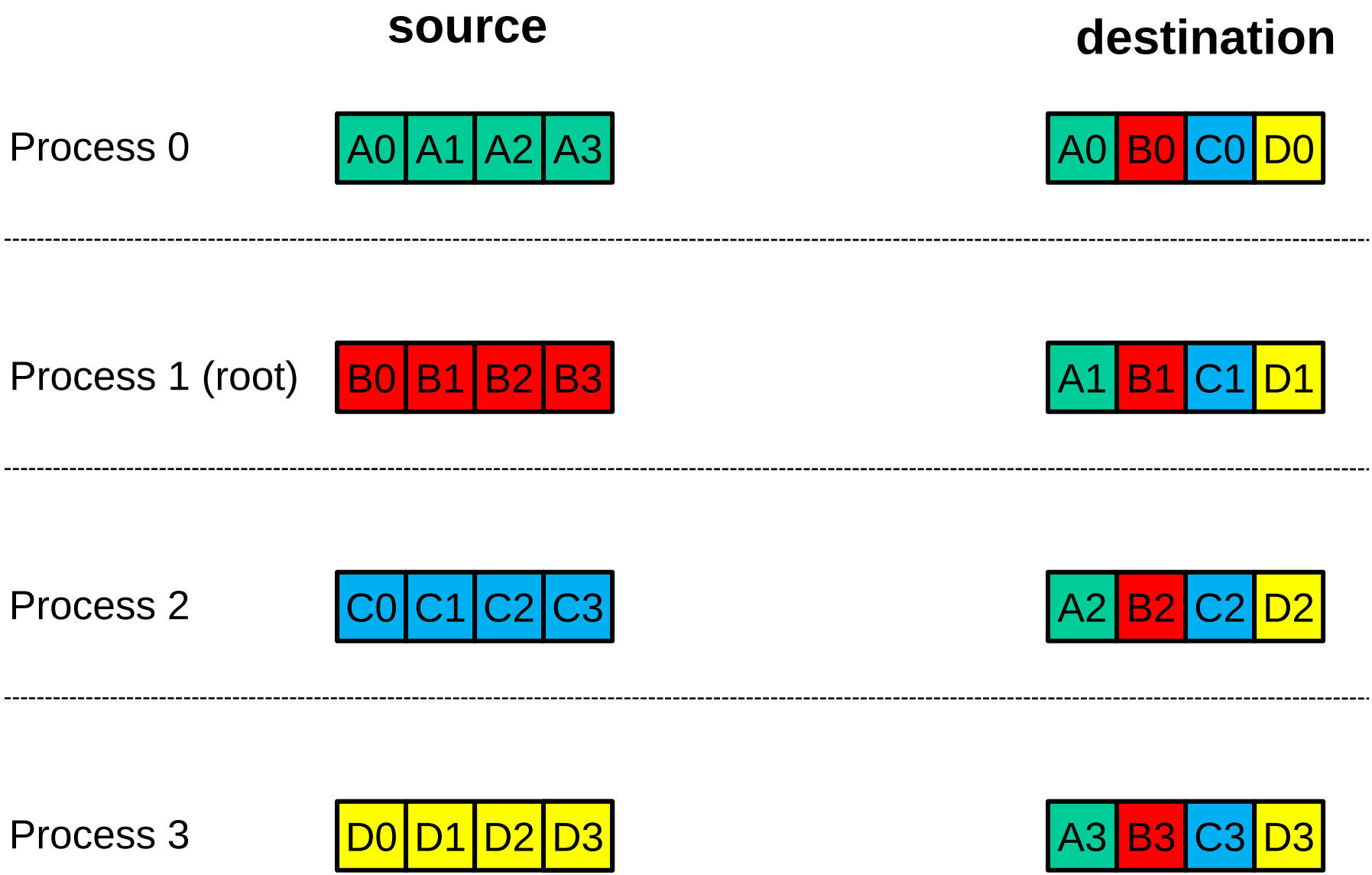
Collective Operations: GATHER



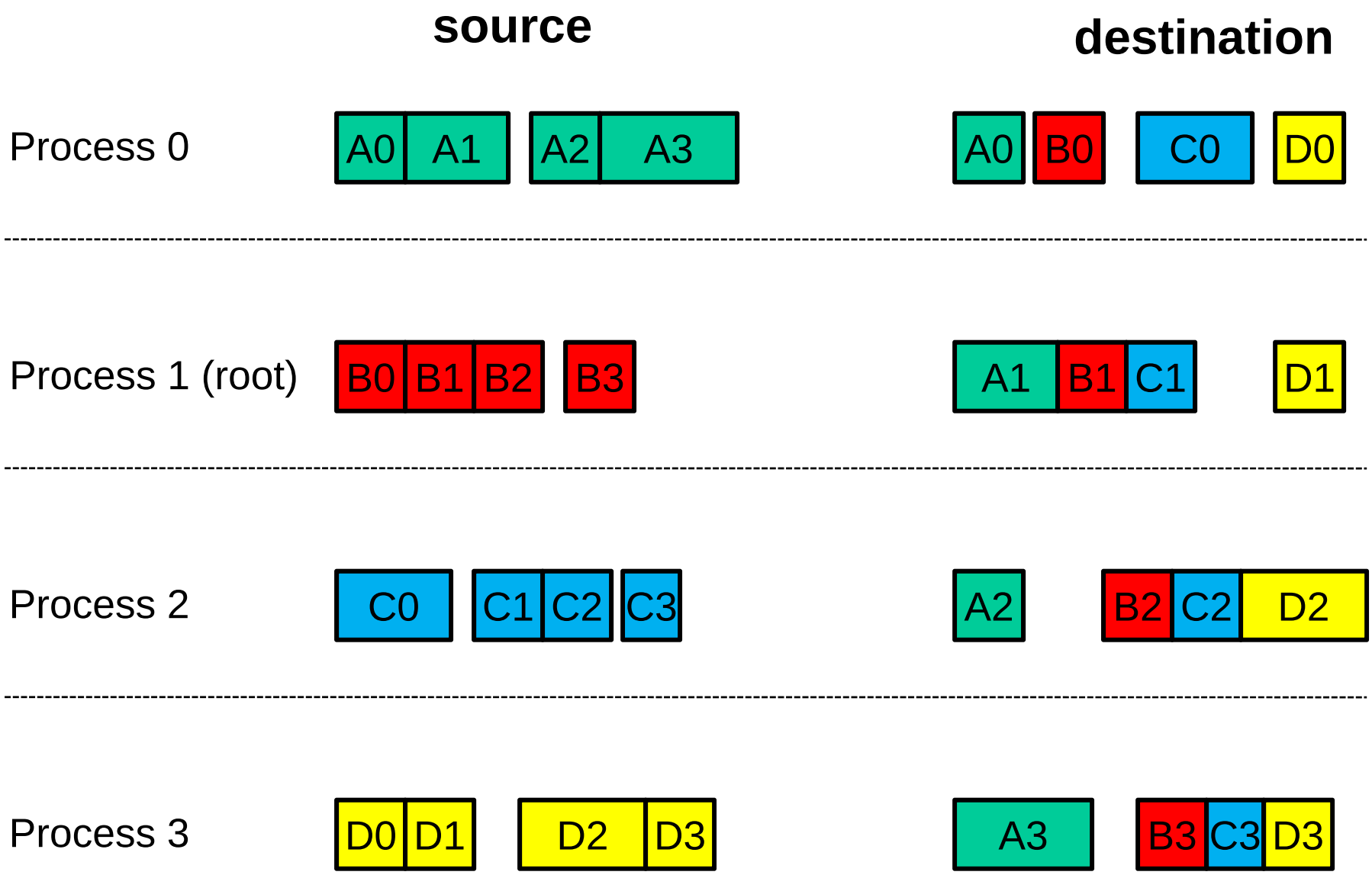
Collective Operations: GATHERV



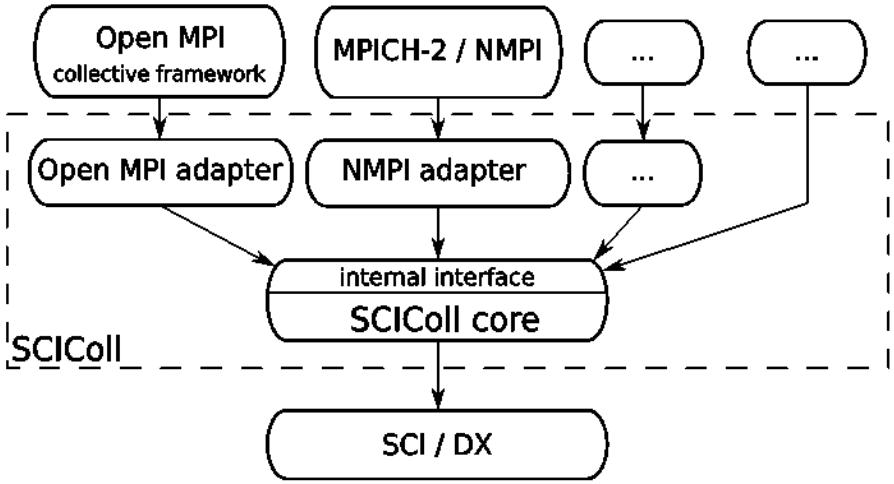
Collective Operations: ALLTOALL



Collective Operations: ALLTOALLV



The SCI Collectives Library

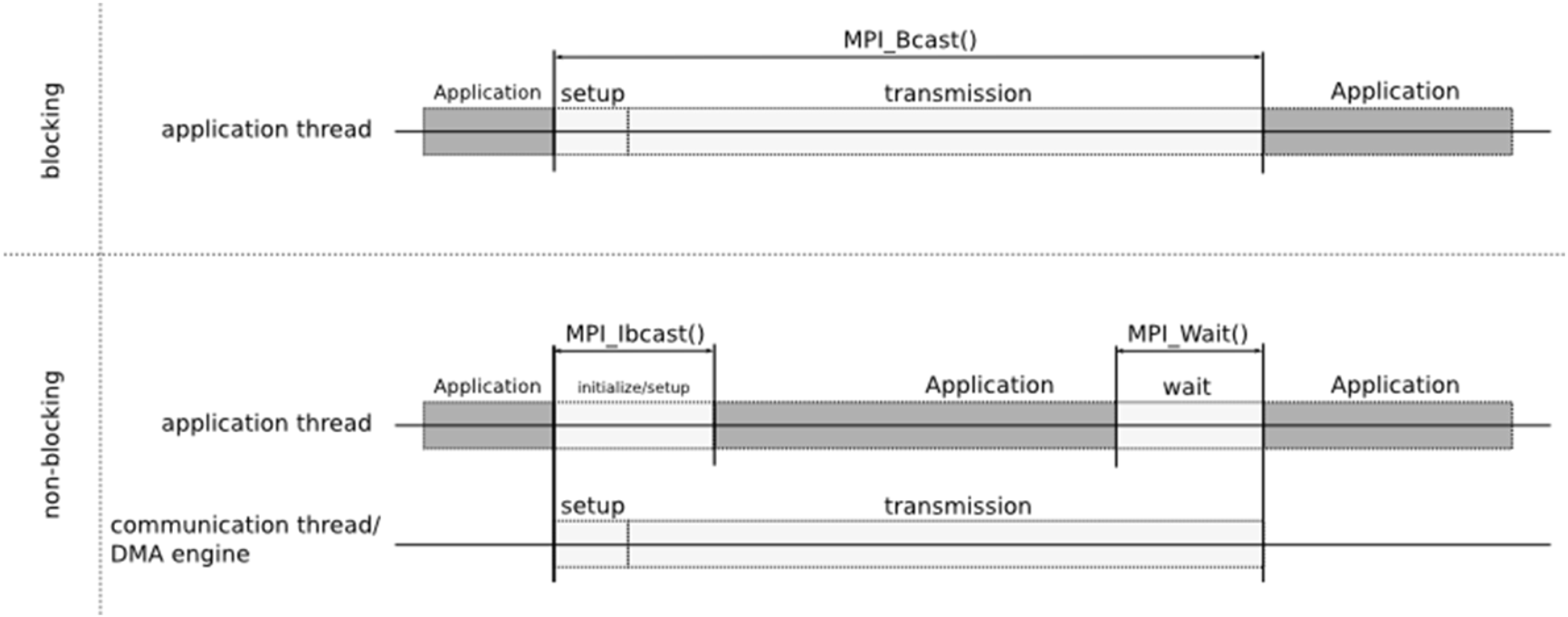


Purpose:

- **Study collective communication algorithms for SCI clusters**
- **Support multiple MPI libraries: Open MPI, NMPI**
- **Support arbitrary communication libraries: LibNBC**

Nonblocking Collectives (NBC)

Purpose: Overlap of Computation and Communication



NBC in MPI

MPI-2.0 JoD: Split Collectives

MPI_BCAST_BEGIN(buffer, count, datatype, root, comm)

MPI_BCAST_END(buffer, comm)

MPI-2.1:

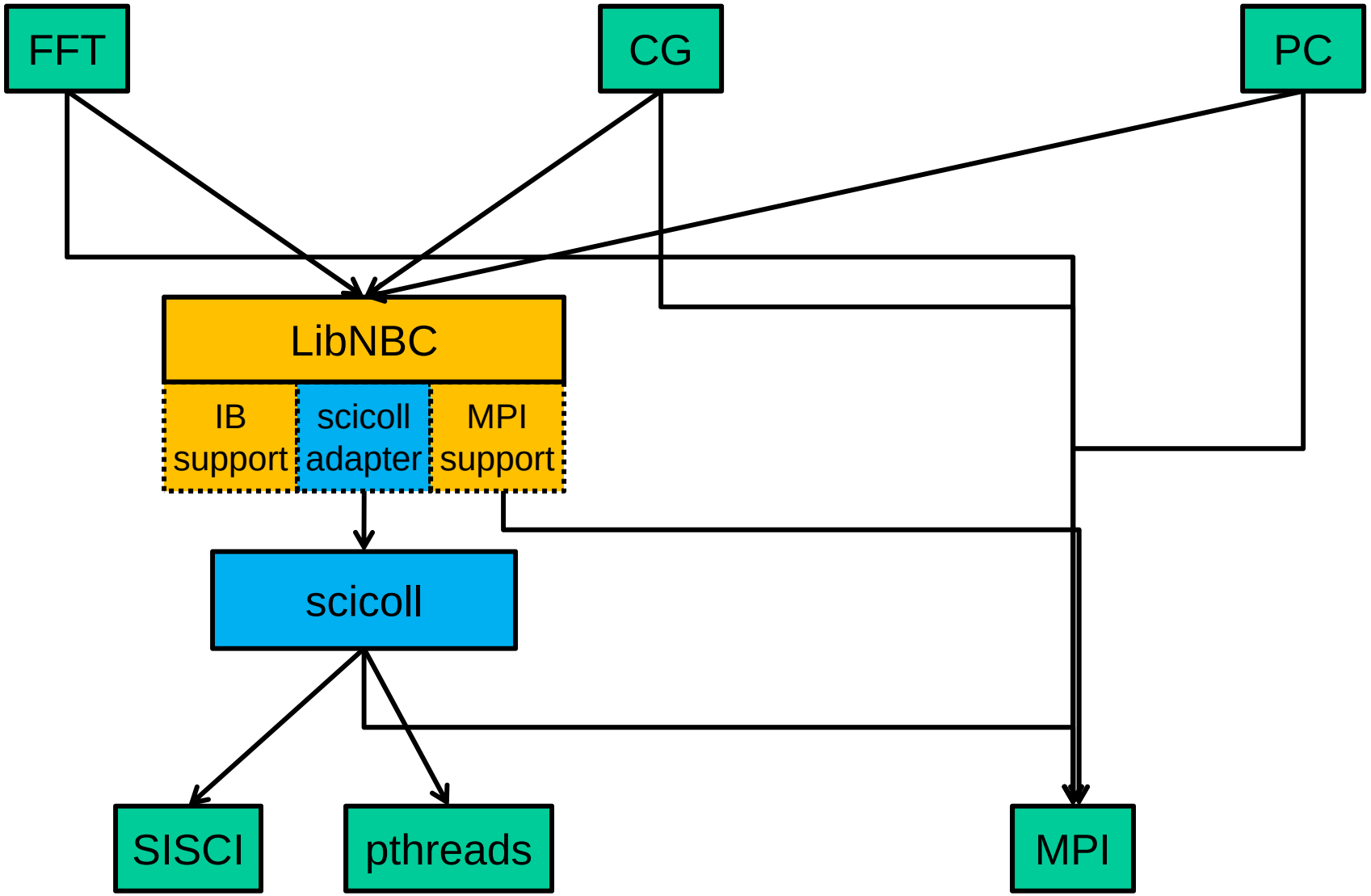
- Implement with nonblocking Point-to-Point operations
- Blocking collectives in separate thread

MPI-3 Draft:

MPI_IBCAST(buffer, count, datatype, root, comm, request)

MPI_WAIT(request, status)

LibNBC


FFT
CG
PC
LibNBC
IB support
scicoll adapter
MPI support
scicoll
SISCI
pthreads
MPI

Rationale: NBC for SCI

So far:

- Promising results with NBC via LibNBC
- Research done on InfiniBand clusters

Therefore:

What about a very different network architecture?

Implementation considerations:

- Use algorithms different from blocking version?
- PIO vs DMA
- Use background thread?

Available Benchmarks for LibNBC API

Synthetic:

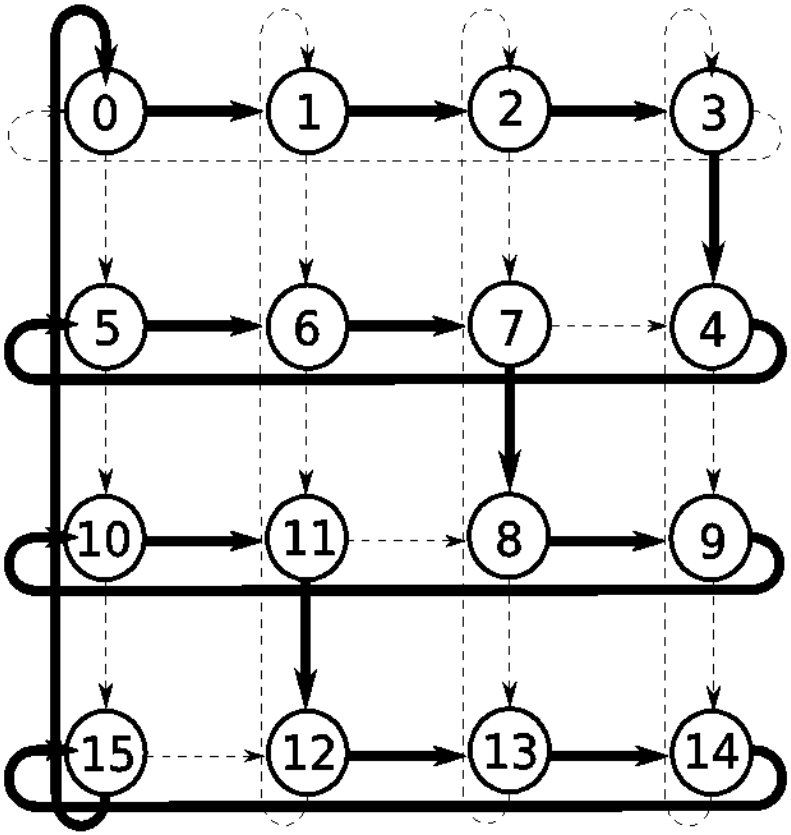
NBCBench: measures the communication overhead / overlap potential

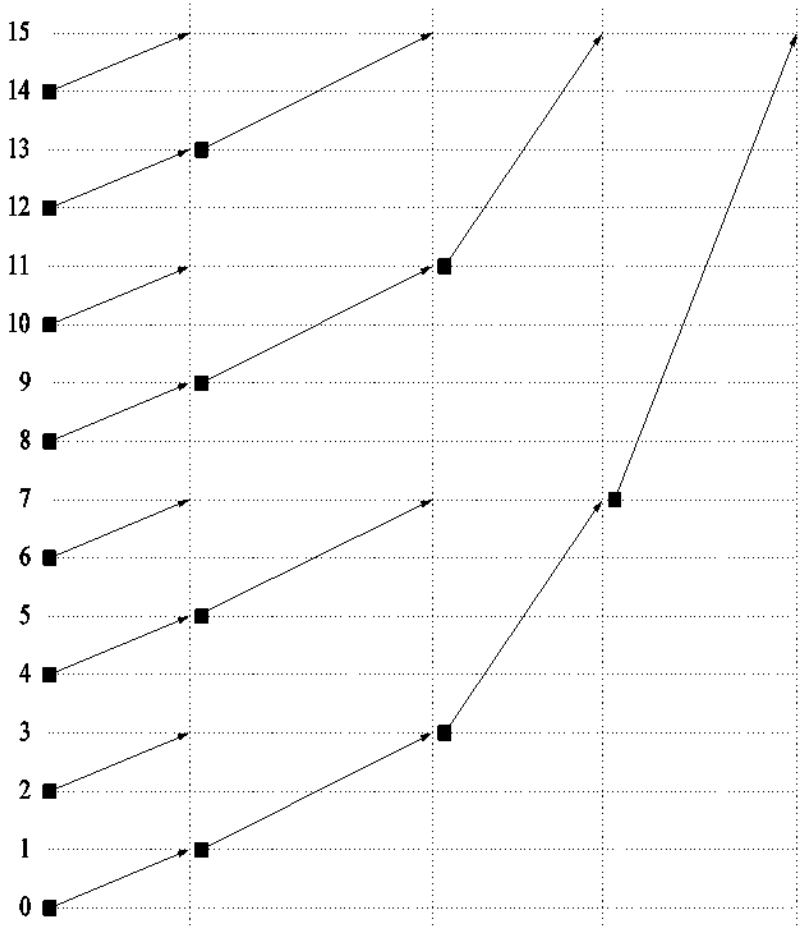
Application Kernels:

- **CG (Alltoallv): 3D Grid, overlaps computation with halo zone exchange**
- **PC (Gatherv): overlaps compression with gathering of previous results**
- **FFT (Alltoall): parallel matrix transpose, overlaps data exchange for z transpose with computation for x and y**

Gather(v)

- **Underlying concept**: Hamiltonian Path in a 2D torus
- **Algorithms**: Binary Tree, Binomial Tree, Flat Tree, Sequential Transmission





Gather(v)/Alltoall(v)

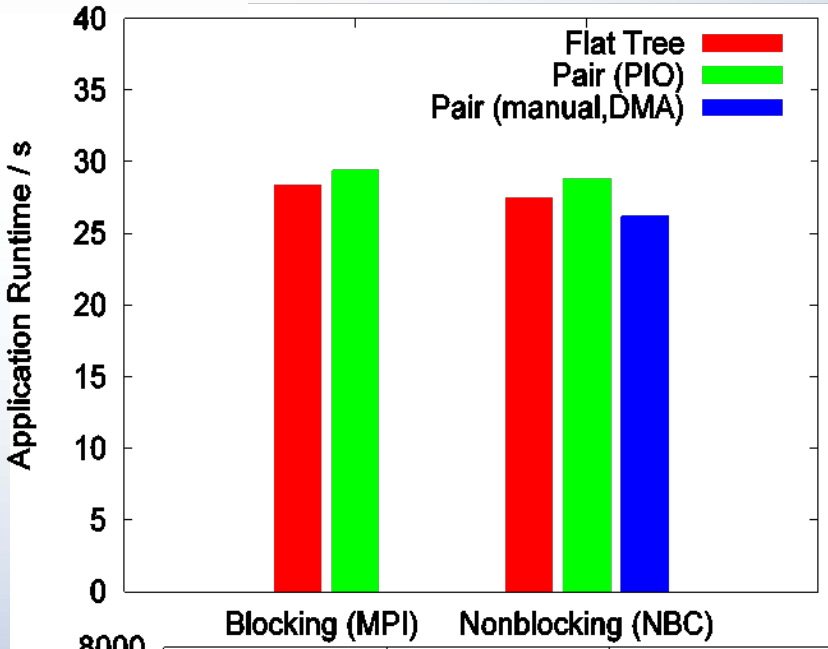
Gather(v):

- Additional progress thread: Binary Tree (PIO), Binomial Tree (PIO), Flat Tree (PIO), Sequential Transmission (PIO, DMA)
- Single Thread with manual progress: Sequential Transmission
- Vector Variant: Flat Tree and Sequential Transmission

Alltoall(v):

- Additional progress thread: Bruck (PIO), Pairwise Exchange (PIO), Ring (PIO), Flat Tree (PIO)
- Single Thread with manual progress: Pairwise Exchange (DMA)
- Vector Variant:Pairwise Exchange, Flat Tree

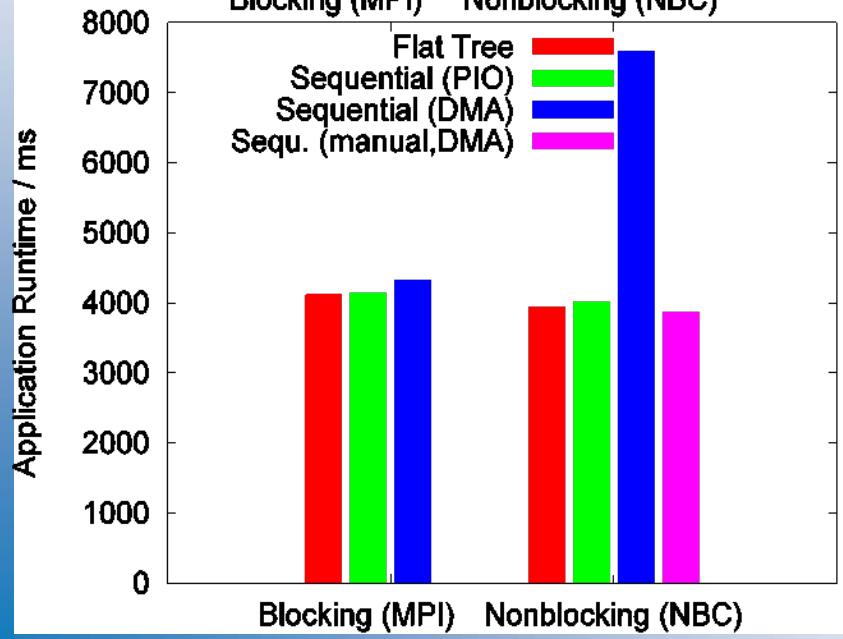
Application Kernels: Algorithms

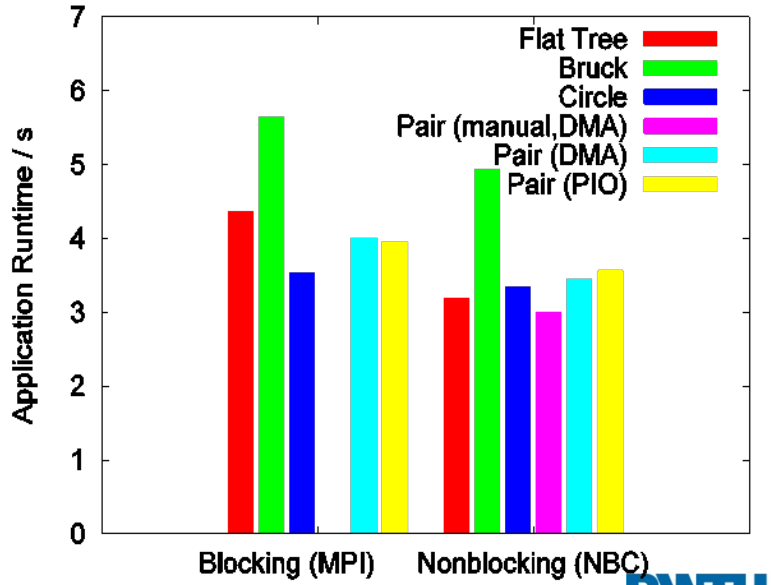


CG (Alltoallv)

PC (Gatherv)

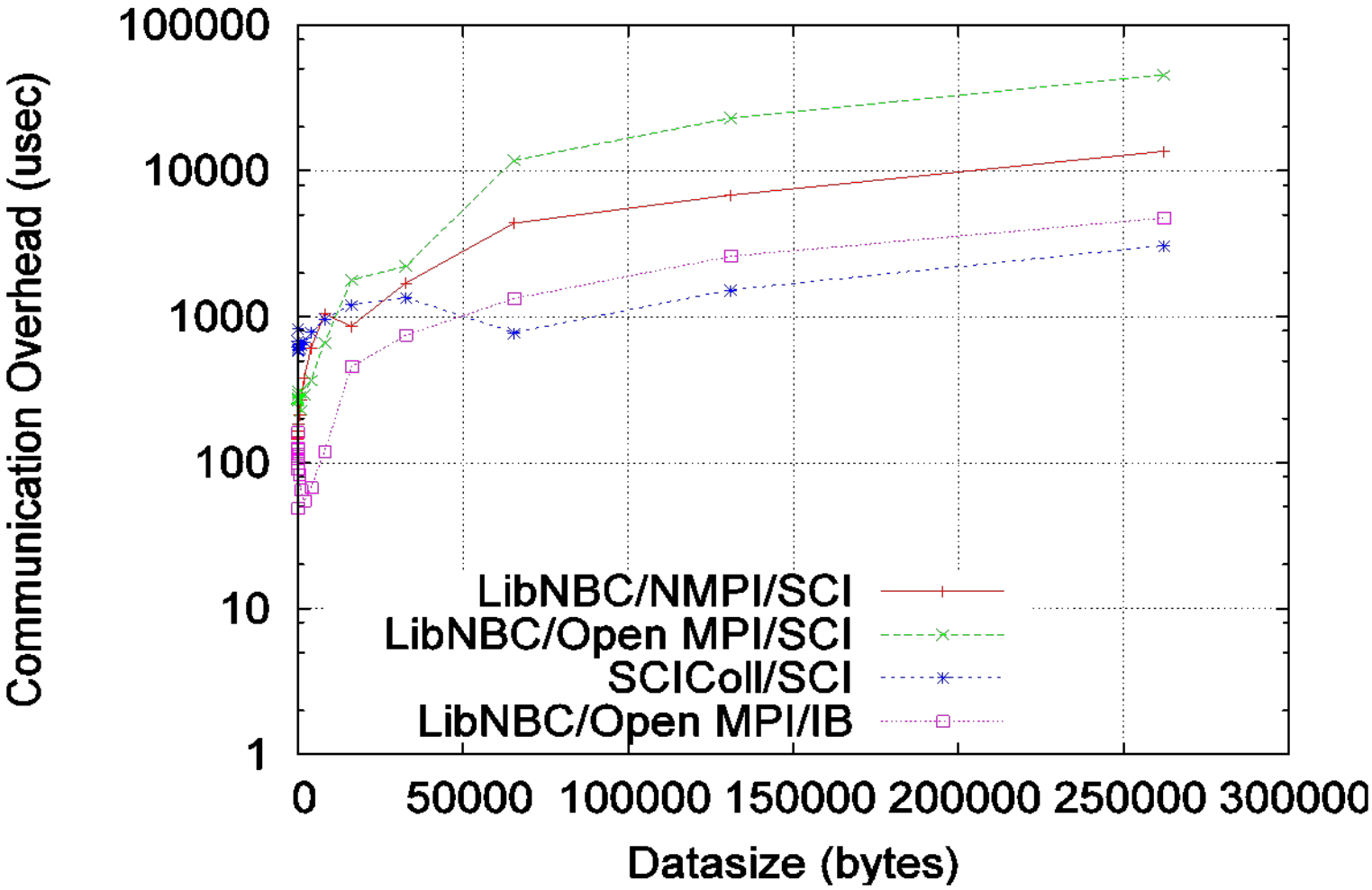
FFT (Alltoall)





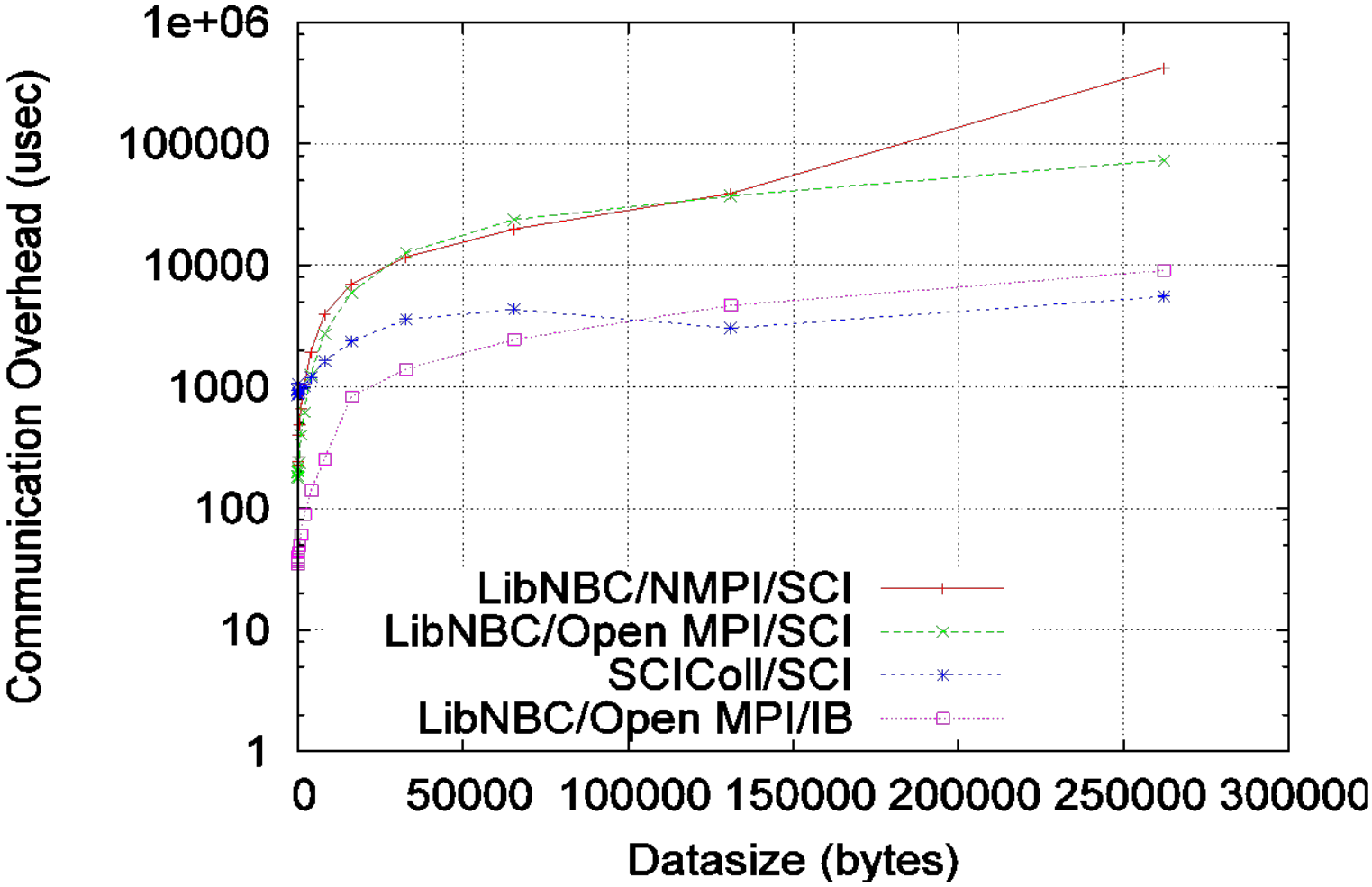
Communication Overhead (NBCBench)

Gather

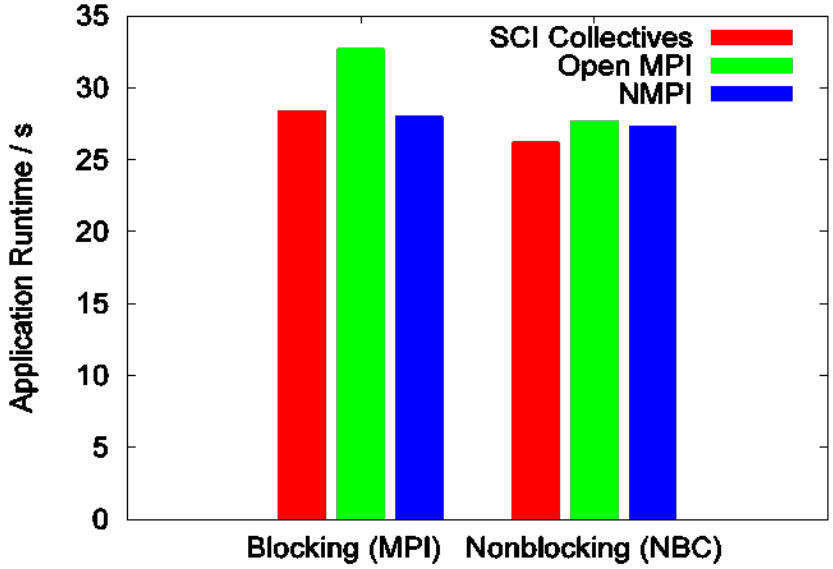


Communication Overhead (NBCBench)

Alltoall



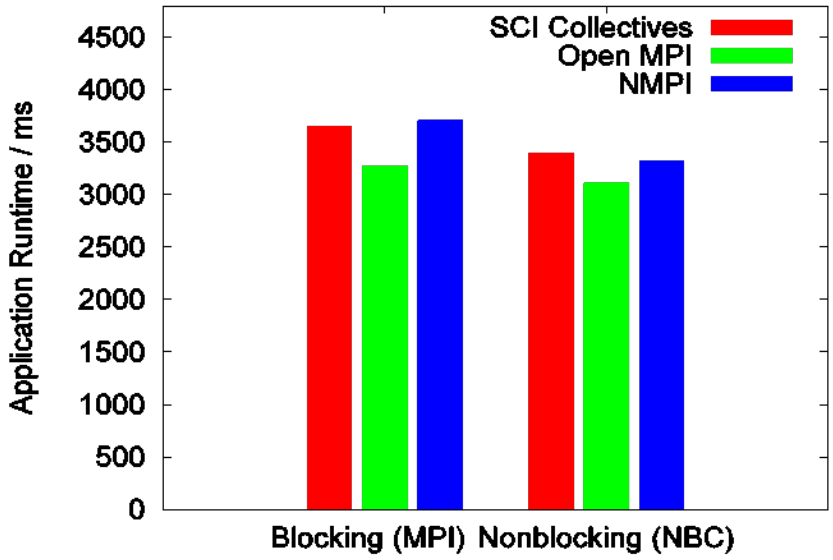
Application Kernels: Performance

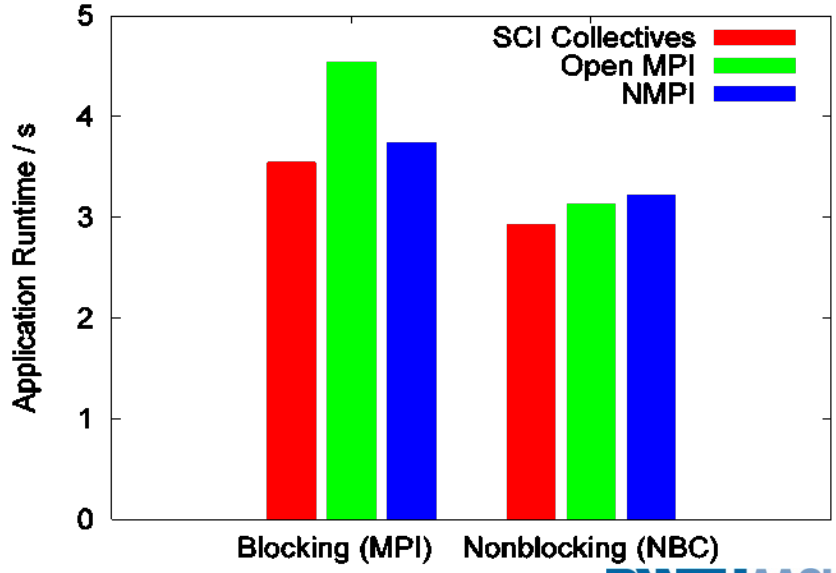


CG (Alltoallv)

PC (Gatherv)

FFT (Alltoall)





Conclusion

What we‘ve done:

Implement nonblocking Gather(v) and Alltoall(v) collective opera-tions on SCI clusters with different algorithms and implementation alternatives

What we found out:

- **Applications can benefit from nonblocking collectives on SCI clusters in spite of inferior DMA performance**
- **Best implementation method: DMA in a single thread, PIO is usually used for blocking collectives**
- **Issues with multiple threads**

The End

Thank you for your attention!