Optimizing neighborhood collectives with Multisends

-Sameer Kumar

(with edits by Torsten Hoefler)

Motivation: Neighborhood Collectives

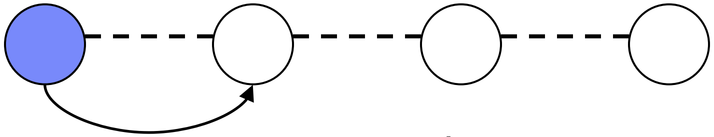
- **Each processor exchanges data with a different subset of ranks in a communicator**
- **Examples**
  - Boundry exchange
    - Neighborhood gather
  - 3D FFT with pencil decomposition
    - Neighborhood alltoall
  - Molecular dynamics real space communication
    - Neighborhood gather and reduce
- **No support in MPI 2.0 for neighborhood collectives**
  - Alltoallv, gatherv are typically not efficient with sparse neighborhoods

Multisend Interface

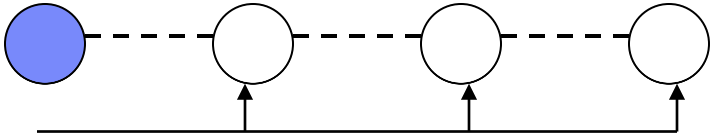
Data Movement via Multisends

- **Multicast**
  - Send data to a collection of processors
  - For example line broadcasts on Blue Gene
- **Network acceleration**
  - Network broadcasts (Quadrics, IB, Blue Gene)
  - Network combine operations for reduce (Blue Gene)
  - Network barriers (Blue Gene)
- **Many-to-many**
  - Scatter and gather operations
- **Processors typically only involved at the end-points of a multisend**

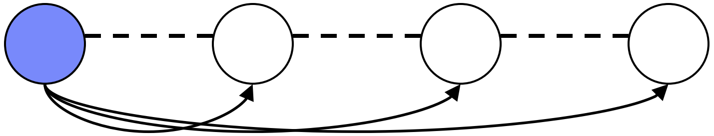
Multisend Examples



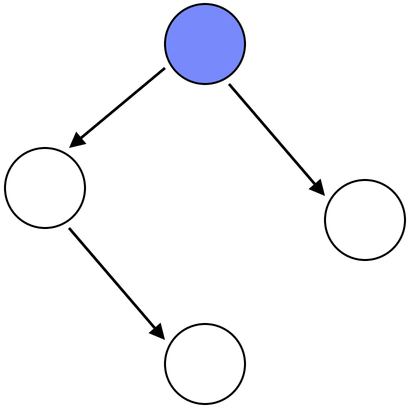
Pt-to-pt Send



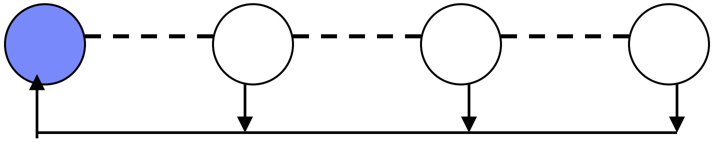
Network Line Broadcast



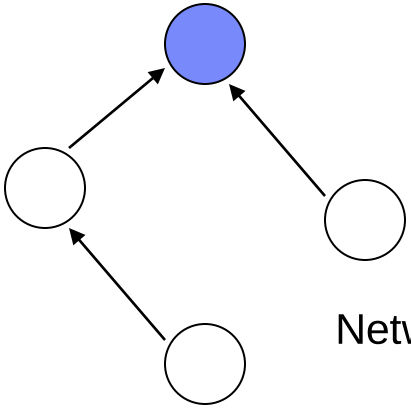
Pt-to-pt Multicast



Network Global Broadcast



Network Line Reduce



Network Global Reduce

DCMF Multicast Interface

DCMF_Multicast (		cb_dispatchMulticast(	
dispatchid,	/* Recv dispatch handler id */	recvRequest,	/*Buffer for recv msg. state*/
request,	/*Buffer to store msg. state*/	srcrank,	/* rank of the sender */
cb_done,	/* Callback that is invoked when multicast completes */	bytes,	/* Size of the multicast*/
		....	
conn_id,	/* Connection identifier tag*/	conn_id,	/*Connection Identifier tag */
persistid,	/* Identify the persistent communication pattern */	....	
		....	
srcbuf,	/* source buffer */	rcvbuf,	/* Buffer for multicast payload*/
size ,	/* size of the message*/	rcvsize,	/* Size of the message */
ranklist,	/* Array of dst. ranks*/	recv_done,	/*Recv completion callback called when all bytes have arrived */
nranks,	/* number of destinations */		
.....	/* Other parameters */	......	/* Other Parameters */
);		);	

DCMF Manytomany Interface

```
DCMF_Manytomany (
        dispatchid,          /* Recv dispatch handler id */
        request,             /* Buffer to store msg. state*/
        cb_done,             /* Callback that is invoked
                             when many-to-many completes*/
        conn_id,             /* Connection Identifier tag */
        persistid,           /*Id of persistent comm. pattern*/
        rIndex,              /* location on the receiver where
                             this sender's message is placed*/
        srcbuf,              /* Base address of source buffer */
        size_vec,            /* Vector of bytes to each dst.*/
        displ_vec,           /* Displacements to each dst. */
        ranklist,            /* List of destination ranks */
        nranks,              /* Number of ranks */
        ....                 /* Other parameters */
);
```

```
cb_dispatchManytomany (
        ....
        recvRequest,         /*Buffer for msg state*/
        rcvbuf,              /*Base address of recv buffer*/
        rcvlens,             /* Vector of recv lengths */
        conn_id,             /*Connection Identifier tag*/
        rcvdispls,           /* Vector of displacements*/
        ....
        nranks,              /*Num ranks to recv from */
        ......
        recv_done,           /*Callback to be called when
                             all data has arrived */
        ....
        ....
        ....
);
```

Advantages of MultiSends

- **Collectives can be built on top of network primitives such as a global broadcast or allreduce**
- **Sending k messages to k destinations is often faster than sending k pt-to-pt messages**
  - On Blue Gene performance can be improved about 5-10x

Neighborhood collectives via DCMF Multisends

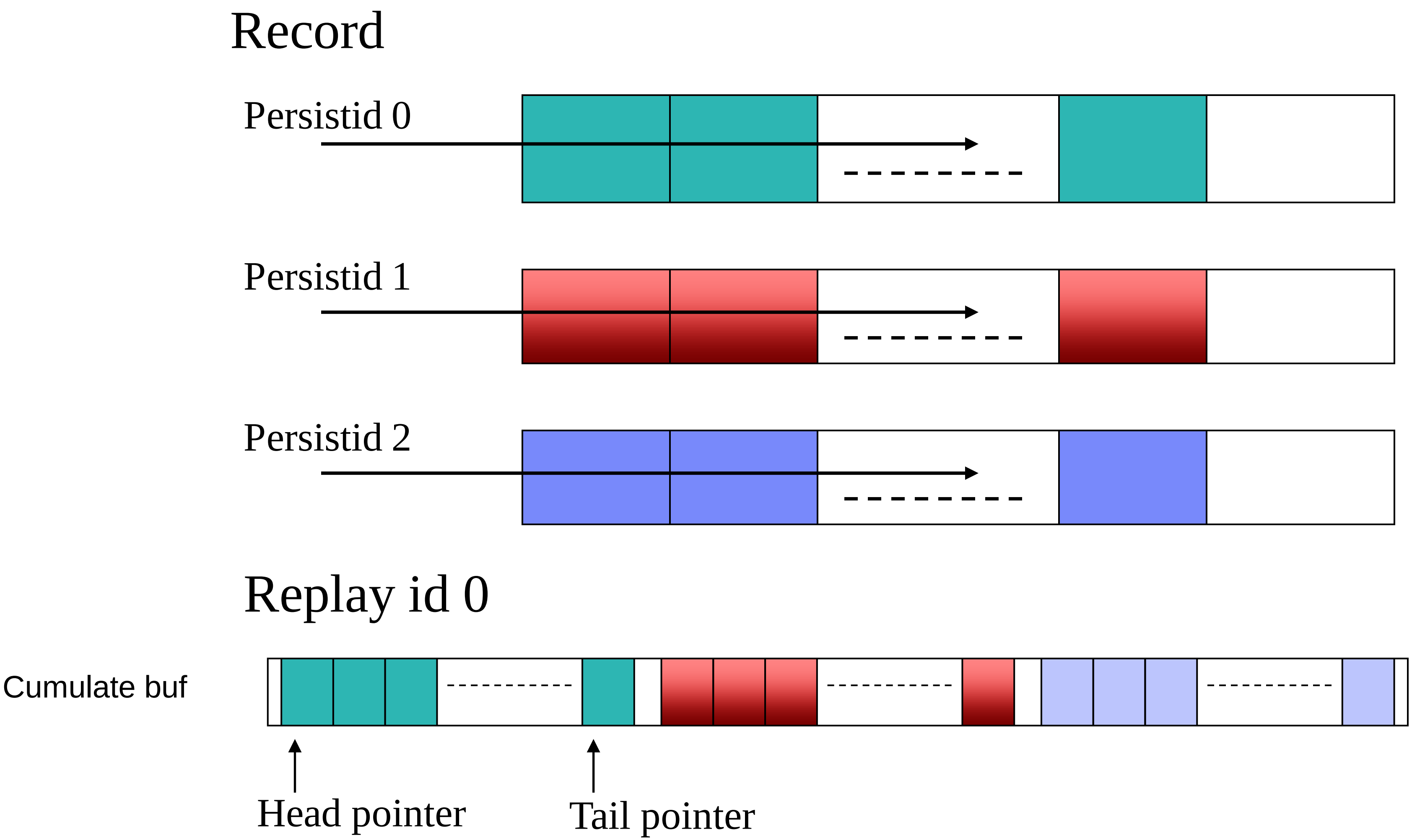
- **Application directly initiates multisend calls**
  – Each task sends data to a different small subset of a communicator
  – Space and CPU optimization

Optimization of Applications with neighborhood collective communication via Multisends, Sameer Kumar, Philip Heidelberger, Dong Chen, Michael Hines, to appear in IPDPS 2010

Record Replay

- **Take advantage of persistent communication**
- **Multisends can record multicast and manytomany calls**
- **Record operation stores message descriptors in a separate FIFO**
- **Replay just sets head and tail pointers in the DMA at very low overheads**

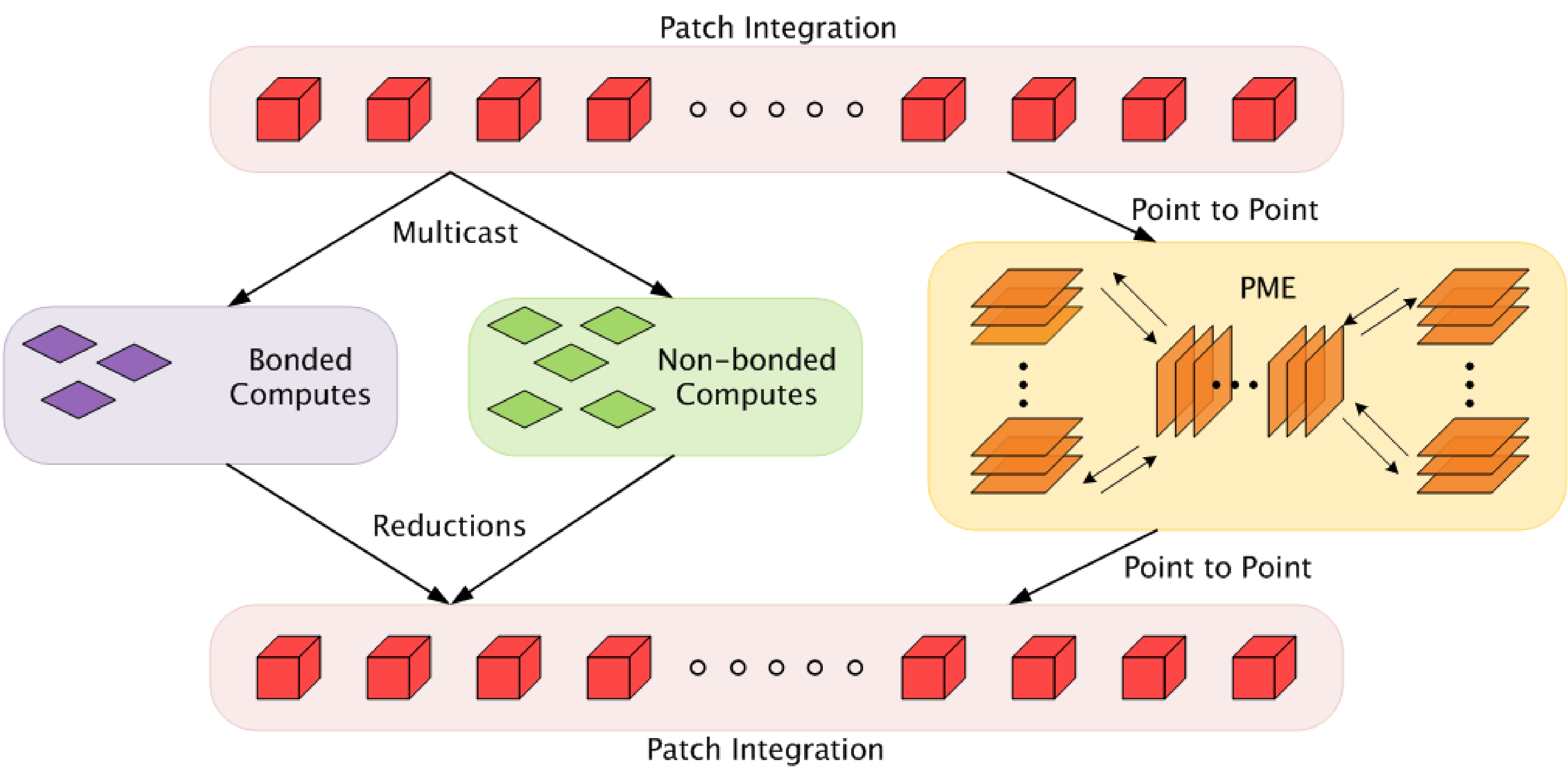
Record Replay



- A 16 byte scatter is 2.7x faster via record replay
- A 16 byte allgather is 2.9x faster via record replay
- MPI standard needs to be enhanced to support persistent collectives

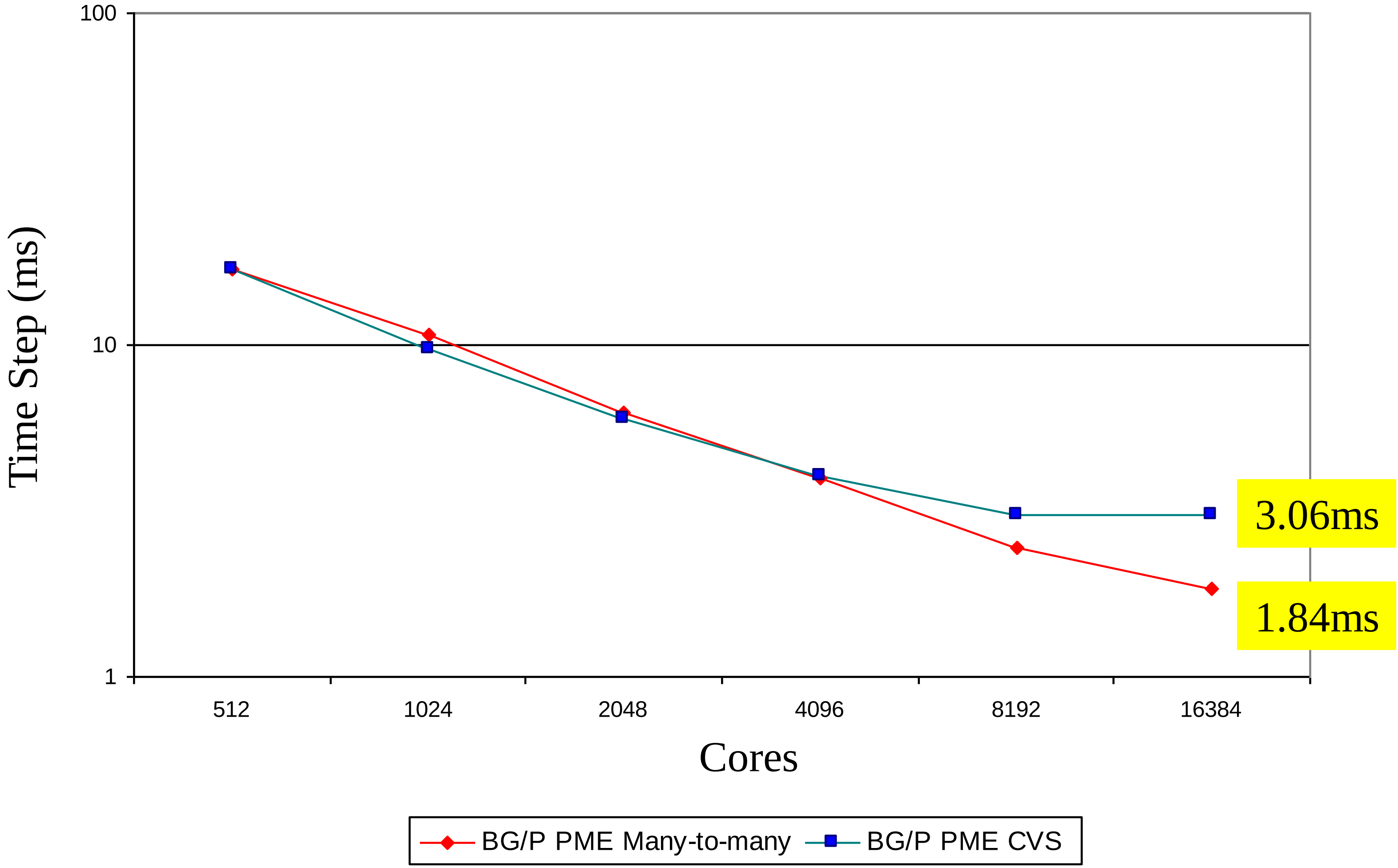
Performance Results

NAMD Molecular Dynamics Application



NAMD PME Optimized by DCMF Multisends

Uses manytomany for sparse alltoalls and multicast for neighborhood broadcasts



92K Atom APoA1 Benchmark with PME every step in Quad Mode

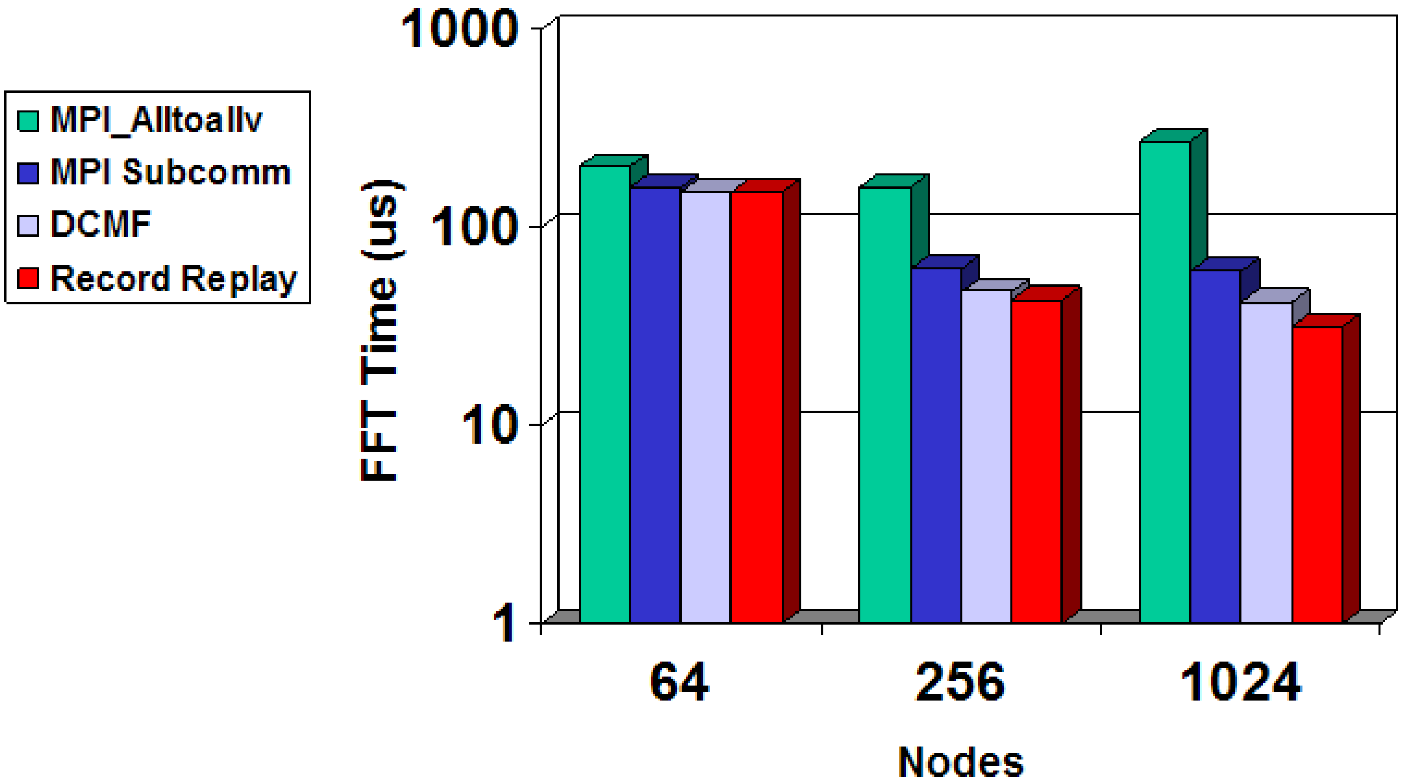
Lattice QCD Benchmark

LQCD loop over iterations:	//Lattice QCD computation loop
Post_receives_0	//Post receives for phase 0
Isend_0	//Send 6 torus near neighbor messages for phase 0
Compute_0	//Phase 0 compute $\frac{(1300*N^4)}{2}$ cycles
Wait_0	//Phase 0 wait
Post_receives_1	//Post receives for phase 1
Isend_1	//Send 6 torus near-neighbor messages for phase 1
Compute_1	//Phase 1 compute $\frac{(1300*N^4)}{2}$ cycles
Wait_1	//Phase 1 wait
Allreduce_0	//Phase 0 CG global sum
Allreduce_1	//Phase 1 CG global sum

Problem Size (N)	MPI Eager			Many-to-many		
N^4 sites/processor	run	comp	wire	run	comp	wire
8	6594	6264	526	6583	6264	526
4	511	392	66	460	392	66
2	104	24.5	8.2	55.7	24.5	8.2

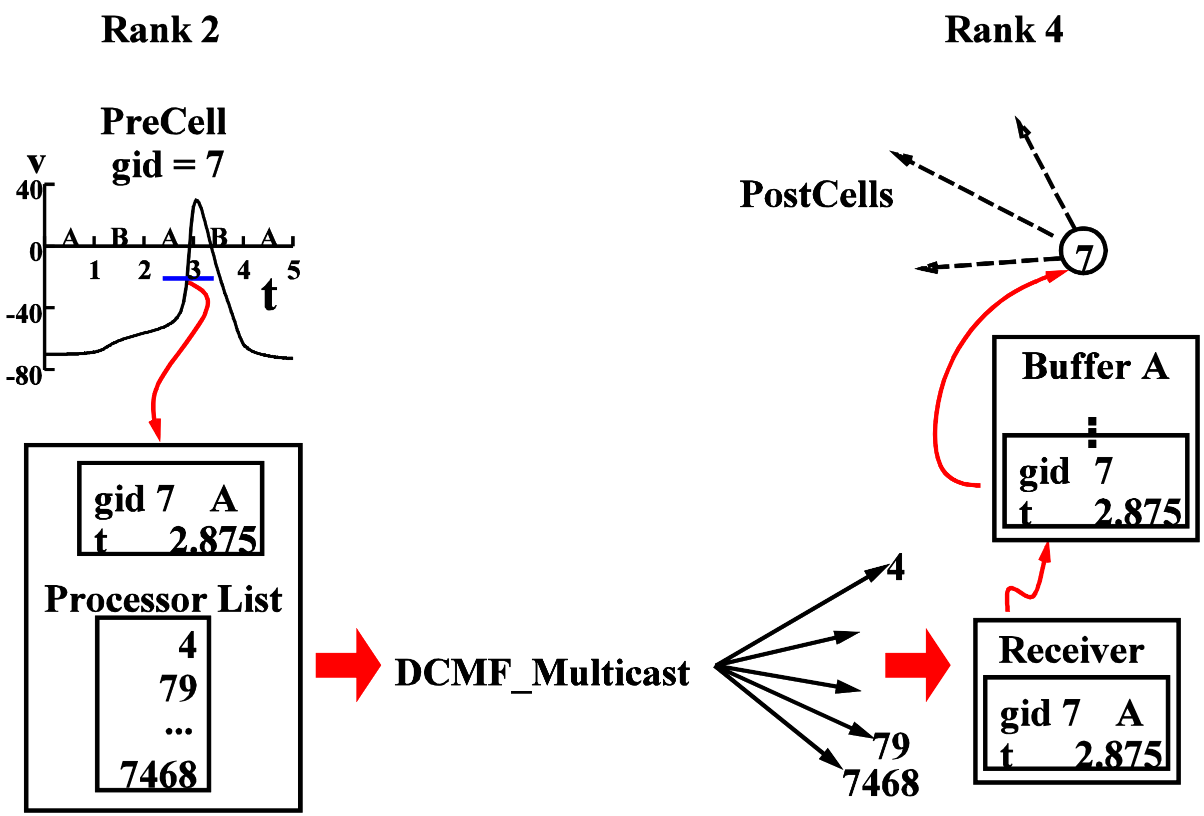
Table 1. 4D Near neighbor benchmark performance (μs) on 512 nodes in Quad mode

3D FFT



- MPI Alltoallv has several zeros in the input vectors
- MPI Alltoall on subcomm has a barrier for synchronization
- DCMF injects descriptors for each destinations

Blue Brain Neuron Application



Neuron Performance

Cores	Cells	Conn.	MPI_Allgather		DCMF_Multicast		Record Replay	
			run	comp	run	comp	run	comp
8192	256K	1k	2.09	0.695	2.06	0.785	1.89	0.684
16384	256K	1k	1.76	0.353	1.25	0.397	0.979	0.347
32768	256K	1k	2.17	0.191	0.834	0.217	0.633	0.187
8192	256K	10k	11.1	6.14	14.9	6.64	14.3	6.04
16384	256K	10k	6.87	3.19	10	3.56	8.88	3.19
32768	256K	10k	4.83	1.61	6.75	1.82	5.87	1.59

Performance of Neuron (seconds) on BG/P

MPI 3 Proposal Discussion

- **Benefits of Neighbor_alltoall() demonstrated**
  – Neighbor_gather() useful if all buffers are the same
- **Neighbor_reduce() is more "intrusive" but might be more useful too**
  – reduces message complexity
- **How important is persistence?**
  – DMA descriptor lists can be updated trivially
  – Is decoupled from neighbor collective semantics

Streaming Completion

- **Brought up at last meeting -> straw vote this meeting!**
- **1st option: MPI_Cwaitany(count, request, index)**
  - returns when first message finishes
  - special new call
- **2nd option: MPI_Sneighbor_alltoall()**
  - returns a list of requests
  - straw-vote: send completion lumped into one?

Nonblocking Collectives Merge

- **Merge done, please <u>review</u>!**

- **Source code will be cleaned up after review**
  - automatically, low error probability

- **What next?**
  - release draft?