Argonne NATIONAL LABORATORY

ALCF MPI Benchmarks: Understanding Machine Specific Communication Behavior

Vitali Morozov, *Jiayuan Meng, Venkatram Vishwanath,*

Jeff Hammond, Kalyan Kumaran, and Michael Papka

Argonne National Laboratory

Fifth International Workshop on Parallel Programming Models and Systems Software for High-End Computing

September 10, 2012

U.S. DEPARTMENT OF ENERGY

Outline

- Introduction and Motivation
- Benchmark Suite and Evaluation
  - Blue Gene/P, Blue Gene/Q, Ethernet cluster, Infiniband cluster
  - Messaging Rate
  - Ping-pong Latency: intranode, internode, farthest
  - Single-node aggregate bandwidth
  - Bi-section bandwidth
  - Collective operations: broadcast, barrier, allreduce
  - Halo exchange
  - Communication-computation overlap
- Conclusions and lessons learnt

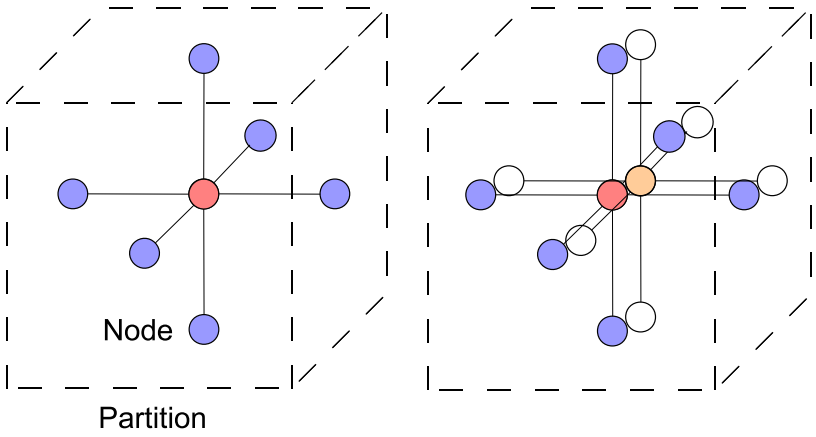
Introduction and Motivation

- Micro benchmarks
  - Intel MPI Benchmarks, SKaMPI, Phloem
- Mini apps
  - SPEC MPI Benchmarks, NAS Parallel Benchmark
- ALCF applications
  - Point to point communication patterns
  - Use of communicators
  - Use of data types
  - Use of synchronization
- Hardware specifics
  - Partitioning, unusual / unexpected behaviour

Evaluation Setup

- IBM Blue Gene/P Intrepid
  - 40 racks, 1024 nodes/rack, 4 CPUs/node
  - 3D torus, collective, barrier
  - 0.425 GB/s, 6+6 links
- IBM Blue Gene/Q Mira
  - 48 racks, 1024 nodes/rack, 16+4 CPUs/ node
  - 5D torus, 2 GB/s, 10+10+2 links
- Eureka
  - 100 nodes, 2 4xXeon E5405
  - 2 Quadro FX5600, 10 Gb Ethernet
- Magellan
  - 100 nodes, 2 8xOpteron
  - 2 Fermi M2070, QDR Infiniband

Messaging Rate



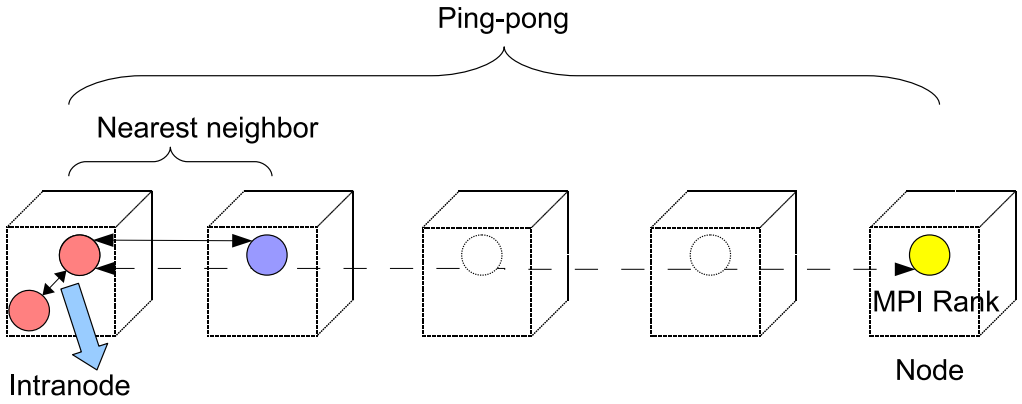

```
for (i = 0; i < iter; i++) {
  total = window * num_targets;
  for (w = 0; w < window; w++)
    for (j = 0; j < num_targets; j++)
      MPI_Irecv(&rb, 0, MPI_CHAR, targets[j],
                tag, comm, &req[total++]);
  total = 0;
  for (w = 0; w < window; w++)
    for (j = 0; j < num_targets; j++)
      MPI_Isend(&sb, 0, MPI_CHAR, targets[j],
                tag, comm, &req[total++]);
  MPI_Waitall(2*window*num_targets, req, stat);
}
```

RPN	BG/P	BG/Q	BG/Q	Eureka	Magellan
	def	def	any_src	Myrinet	Infiniband
1	0.51	0.88	0.91	0.084	0.899
2	0.98	1.75	1.92		
4	1.79	3.19	3.56		
8		6.29	6.63		
16		11.21	11.97		
64		10.25	17.64		

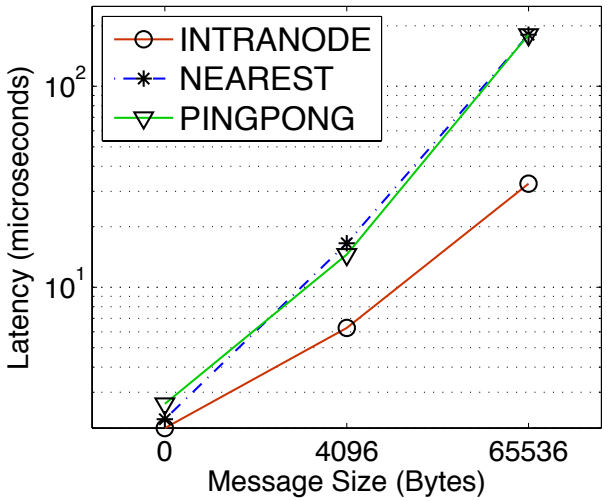
WIN	BG/Q	BG/Q
	4 RPNs	64 RPNs
1	3.27	17.64
2	3.47	15.01
3	3.53	13.56
4	3.36	12.91
5	3.41	12.34
6	3.50	12.02

in millions of messages per second

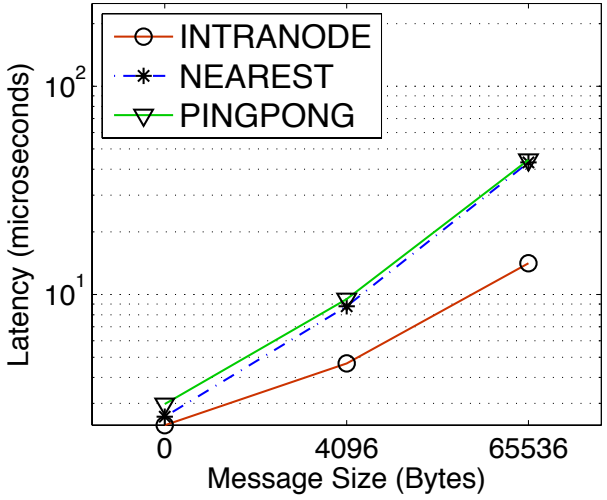
Ping-pong latency



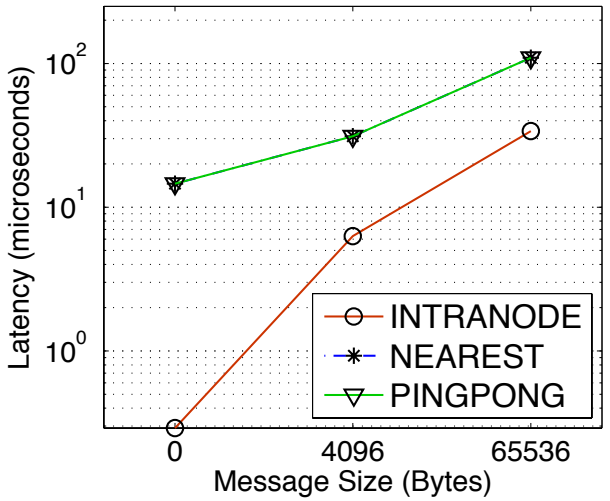

```
t1 = MPI_Wtime();
for (k = 0; k < MAXN; k++) {
  MPI_Send( sb, L, MPI_CHAR, ir1, is1,
       MPI_COMM_WORLD );
  MPI_Recv( rb, L, MPI_CHAR, ir1, ir1,
       MPI_COMM_WORLD, MPI_STATUS_IGNORE);
}
t2 = MPI_Wtime();
t = 1e6 * (t2 – t1) / MAXN / 2.0
```



BG/P 1rack

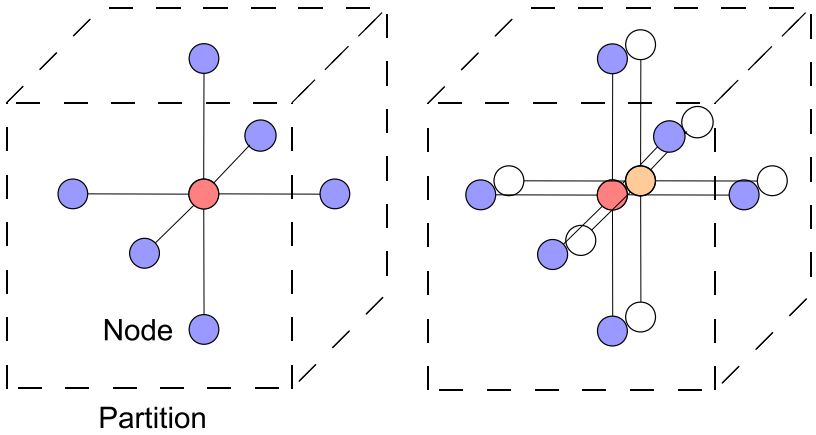


BG/Q 1rack



Myrinet 100 nodes

Single node aggregate bandwidth




```
for ( k = 0; k < MAXN; k++ ) {
  for ( i = 0; i < N; i++ ) {
    MPI_Isend( sb, LENGTH, MPI_BYTE,
        ir1[i], is1, comm, &req[i] );
    MPI_Irecv( rb, LENGTH, MPI_BYTE,
        ir1[i], ir1[i], comm, &req[i+N] );
  }
  MPI_Waitall( 2*N, req, stat );
}
```

512 KB message size

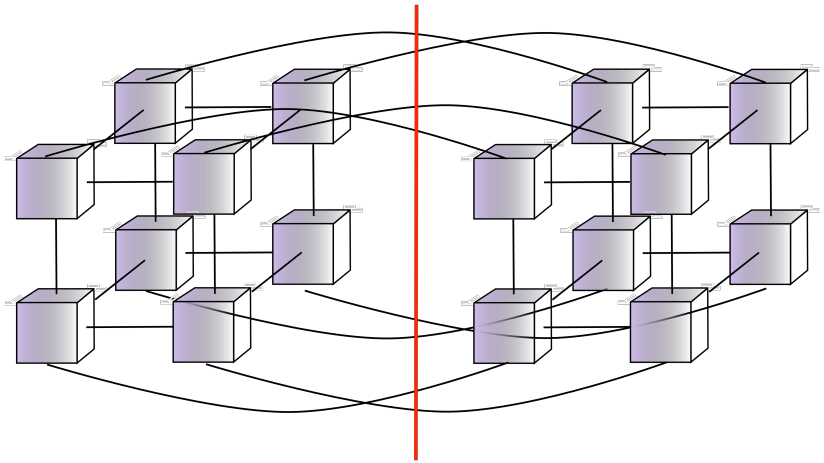
RPN	BG/P	BG/Q
	def	def
1	4.28	31.49
2	4.38	35.45
4	4.37	35.54
16		35.59
64		32.07
Peak	4.53	36.00

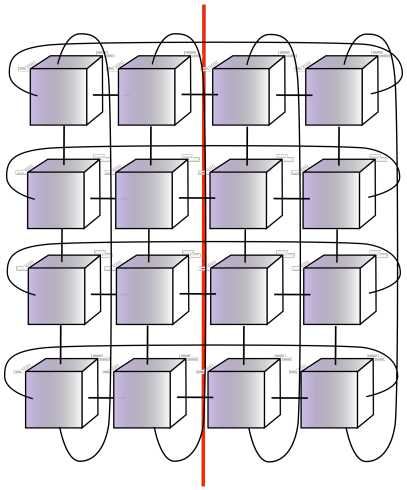
in GB/s

Bi-section bandwidth

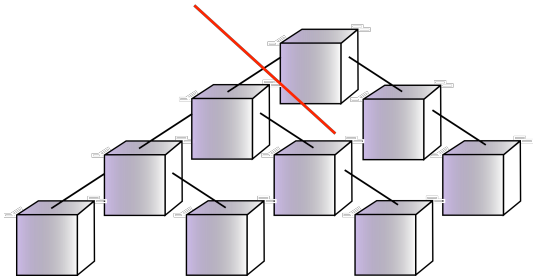
- Capacity of the network across the narrowest bi section

```
for ( i = 0; i < N; i++ ) {
    MPI_Isend( sb, LENGTH, MPI_BYTE,
        ir1[i], is1, comm, &req[i] );
    MPI_Irecv( rb, LENGTH, MPI_BYTE,
        ir1[i], ir1[i], comm, &req[i
  +N] );
    MPI_Waitall( 2, req, stat );
}
```





512 KB message size

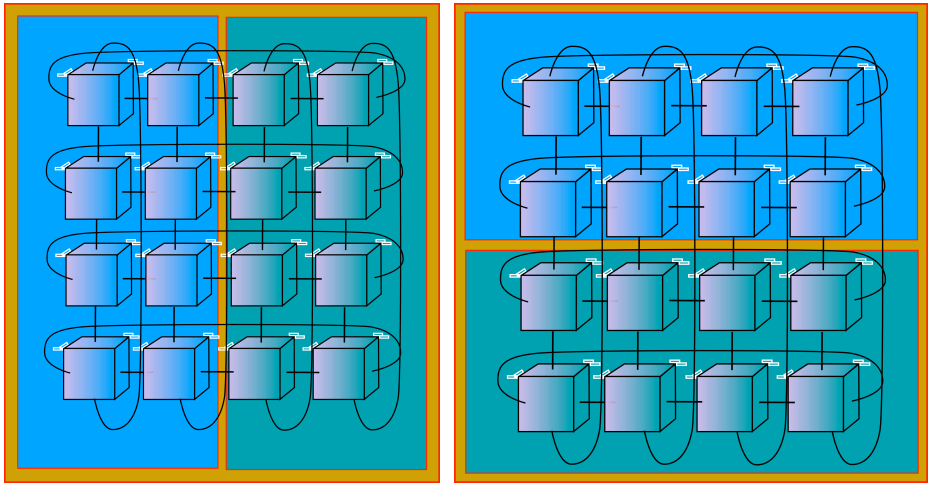


RPN	BG/P	BG/Q	BG/Q
Nodes:	512	512	8192
1	95.3	887	3545
4	95.9	910	3638
16		909	3638
Peak	96.8	922	3686

in GB/s

Collective operations

- Operations
  - Broadcast, Allreduce, Barrier
- Communicators
  - World, Copy, World-1, Half
- Data types
  - Arrays, user-defined



	MPI ranks per node							
1 rack	BG/P			BG/Q				
	1	2	4	1	4	8	16	64
Barrier	1.19	2.29	3.06	2.25	3.56	3.63	3.73	9.21
Broadcast 2	4.58	7.31	7.87	4.01	4.27	4.33	4.51	8.70
Broadcast 1K	20.08	26.80	27.53	9.30	9.49	9.49	9.61	17.19
Allreduce 1	5.52	6.60	7.41	4.34	4.95	5.44	6.15	14.53
Allreduce 256	28.58	37.86	43.26	5.85	8.49	8.97	11.41	29.76

in micro seconds

Collective operations: Half World

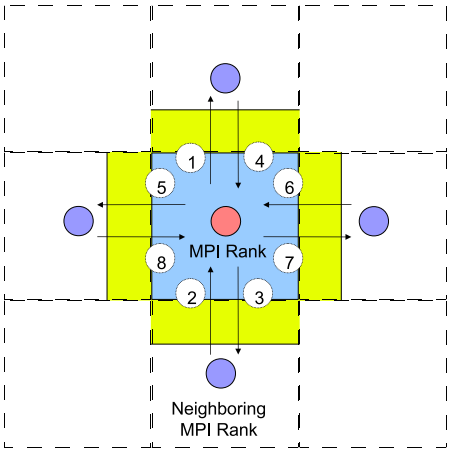
World	MPI ranks per node							
1 rack		BG/P				BG/Q		
	1	2	4	1	4	8	16	64
Barrier	1.19	2.29	3.06	2.25	3.56	3.63	3.73	9.21
Broadcast 2	4.58	7.31	7.87	4.01	4.27	4.33	4.51	8.70
Broadcast 1K	20.08	26.80	27.53	9.30	9.49	9.49	9.61	17.19
Allreduce 1	5.52	6.60	7.41	4.34	4.95	5.44	6.15	14.53
Allreduce 256	28.58	37.86	43.26	5.85	8.49	8.97	11.41	29.76

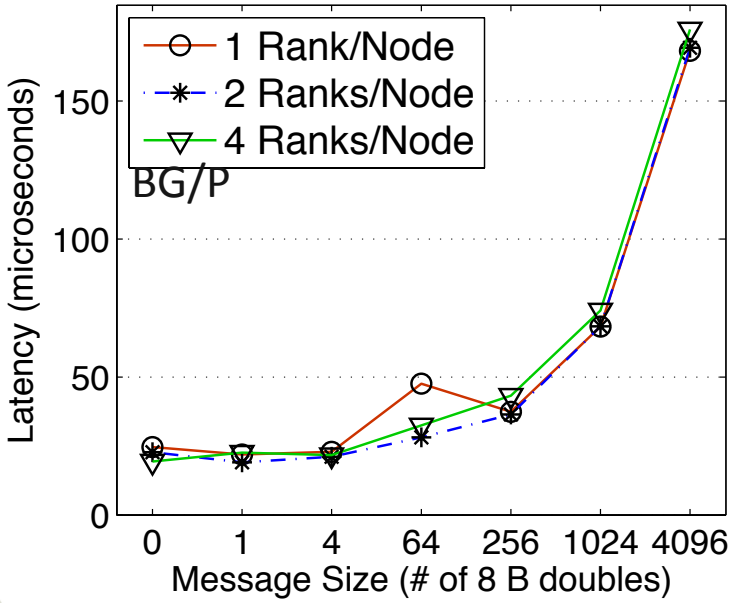
Half World	MPI ranks per node							
1 rack		BG/P				BG/Q		
	1	2	4	1	4	8	16	64
Barrier	10.63	11.94	11.99	2.14	3.45	3.52	3.63	9.37
Broadcast 2	3.16	3.81	4.47	3.91	4.21	4.30	4.46	8.42
Broadcast 1K	65.42	76.61	86.36	9.11	9.39	9.62	9.42	16.73
Allreduce 1	17.91	21.41	22.51	4.20	4.78	5.40	6.04	14.47
Allreduce 256	154.6	170.6	183.46	5.77	8.47	8.96	11.26	29.48

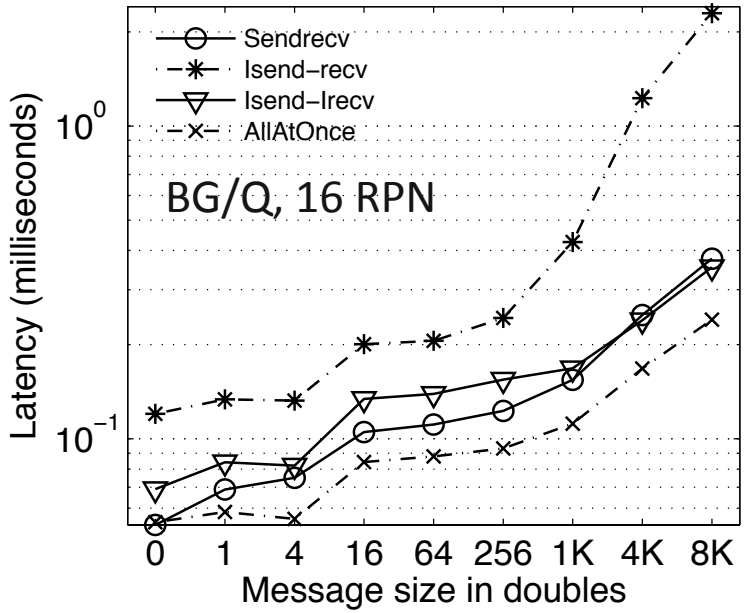
in micro seconds

Halo Exchange

- Sendrecv
  - Synchronous, dimensions in order
- Isend-recv
  - Sync at recv, dimensions in order
- Isend-irecv
  - Async, dimensions in order
- AllAtOnce
  - Posted Irecv, Isends, sync at Waitall







Communication/Computation Overlap

Eff = ((t2 – t1) – (t4 – t3)) / (t4’ – t1’)

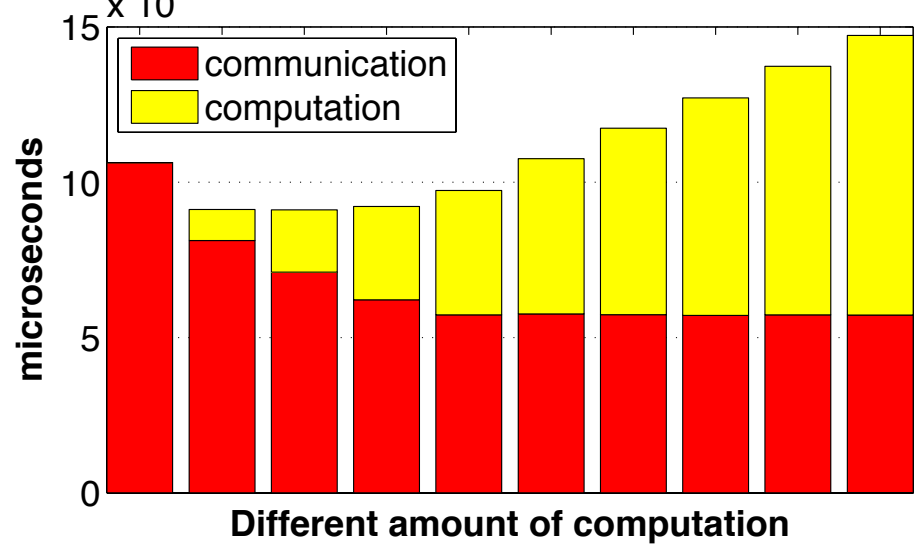
t4’ – t1’: communication time without compute

t2 – t1 : time to post communications

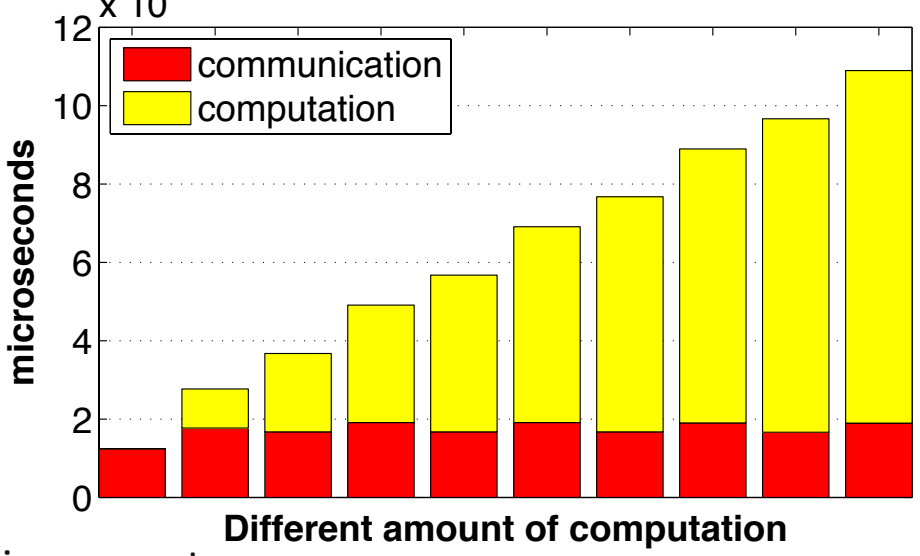
t4 – t3: time to sync posted communications

```
t1 = MPI_Wtime();
MPI_Isend(sbuff, sizes[j], MPI_CHAR, 1,
   tag, comm, &req[0]);
MPI_Irecv(rbuff, sizes[j], MPI_CHAR, 1,
   tag, comm, &req[1]);
t2 = MPI_Wtime();
delay(compute);
t3 = MPI_Wtime();
MPI_Waitall(2, req, sts);
t4 = MPI_Wtime();
```

t4 – t1 interrupts on



t4 – t1 interrupts off



in 10 ms increment

Conclusions and lessons learnt

- ALCF MPI Benchmark suites contains a set of micro benchmarks and miniapps, which reflect ALCF use cases
- Intuition does not always help to project performance of elementary operations: internode vs intranode, single link vs long path, subcommunicator in collectives
- Benchmarking helps to find the best strategy for rank placements within the node in globally
- Scaling of micro operations does not always reflect expectations
- Users rarely follow the “best practice” advices – benchmarking estimates the “punishment”

- ALCF MPI Benchmark is extensively used for platform comparisons, performance projections, and procurement processes