Performance of MPI

Operations and Computer Technologies Div.
RIKEN Center for Computational Science

Contents

Introduction

- Results of Intel MPI Benchmarks (IMB)
  - Point-to-point communication
  - Collective communication
- Measurement conditions

Language version	lang/tcsds-1.2.33	
IMB version	IMB-v2021.2	
IMB options	Memory size =3GiB/process	(-mem 3.0)
	Max. iteration time=100 sec.	(-time 100.0)
	Max. message length=4MiB	(default)
	(job scripts are attached in the last of this document)	
# of Parallels	P2P : 1 process / node, 1D torus ・384 nodes, 384 processes, node shape=384:torus	
	Collective: 4 processes / node, 3D torus ・384 nodes, 1536 processes, node shape=4x6x16:torus:strict-io ・3,072 nodes, 12,288 processes, node shape=16x12x16:torus:strict-io ・27,648 nodes, 110,592 processes, node shape=48x12x48:torus:strict_io	

2.Point-to-point communication

Targets: PingPong, PingPing, Sendrecv, Exchange

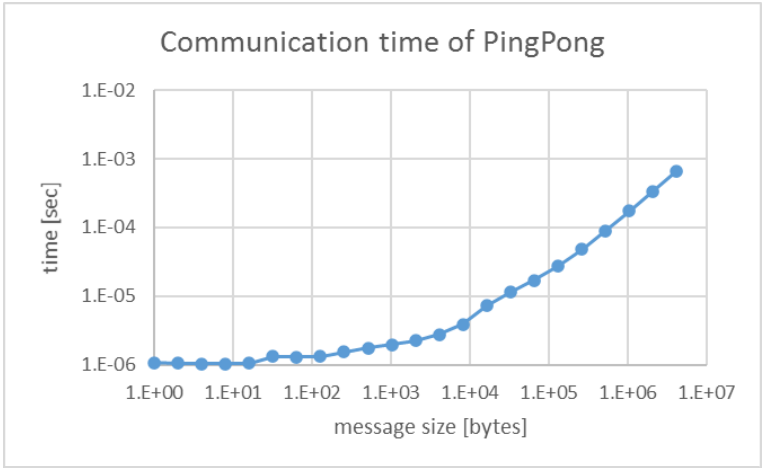
# of parallels : 1 process / node, 1D torus

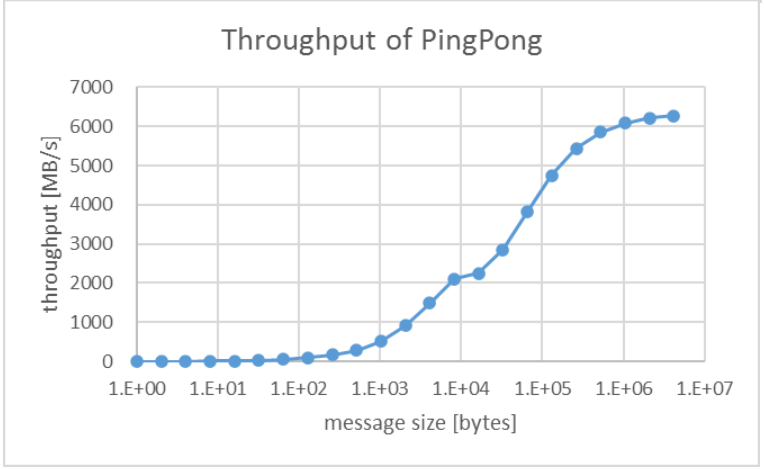
- 384 nodes, 384 processes, node shape=384:torus

2.1 PingPong

2 nodes, 2 procs (1 procs/node)

Message size [bytes]	Communication time [sec]	Throughput [MB/s]
0	1.05E-06	0
1	1.08E-06	1
2	1.07E-06	2
4	1.04E-06	4
8	1.05E-06	8
16	1.07E-06	15
32	1.33E-06	24
64	1.31E-06	49
128	1.32E-06	97
256	1.56E-06	164
512	1.78E-06	287
1024	1.98E-06	516
2048	2.25E-06	910
4096	2.76E-06	1484
8192	3.90E-06	2102
16384	7.27E-06	2253
32768	1.15E-05	2851
65536	1.71E-05	3827
131072	2.76E-05	4753
262144	4.83E-05	5433
524288	8.95E-05	5858
1048576	1.72E-04	6081
2097152	3.38E-04	6211
4194304	6.68E-04	6275

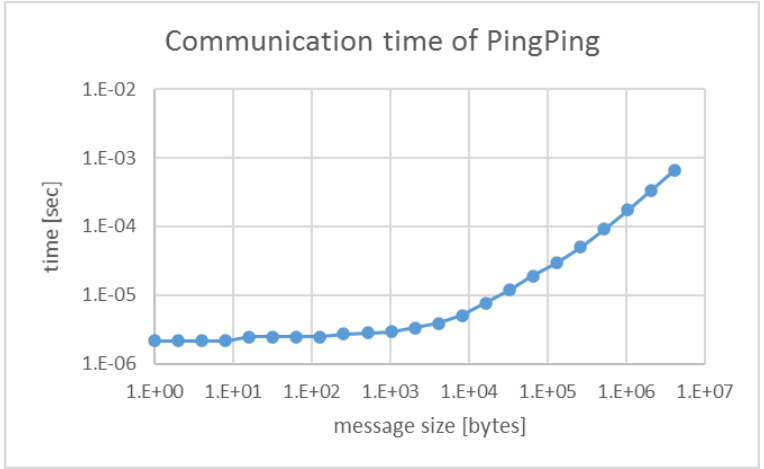


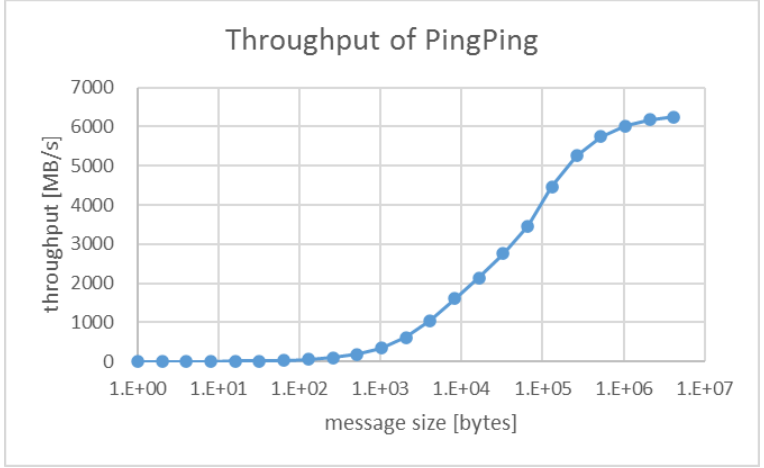


2.2 PingPing

2 nodes, 2 procs (1 procs/node)

Message size [bytes]	Communication time [sec]	Throughput [MB/s]
0	2.05E-06	0
1	2.16E-06	0
2	2.16E-06	1
4	2.16E-06	2
8	2.17E-06	4
16	2.48E-06	6
32	2.49E-06	13
64	2.47E-06	26
128	2.49E-06	51
256	2.71E-06	94
512	2.85E-06	180
1024	2.96E-06	346
2048	3.33E-06	615
4096	3.90E-06	1050
8192	5.10E-06	1607
16384	7.65E-06	2141
32768	1.19E-05	2751
65536	1.89E-05	3461
131072	2.94E-05	4465
262144	4.98E-05	5262
524288	9.13E-05	5745
1048576	1.74E-04	6023
2097152	3.39E-04	6181
4194304	6.70E-04	6257

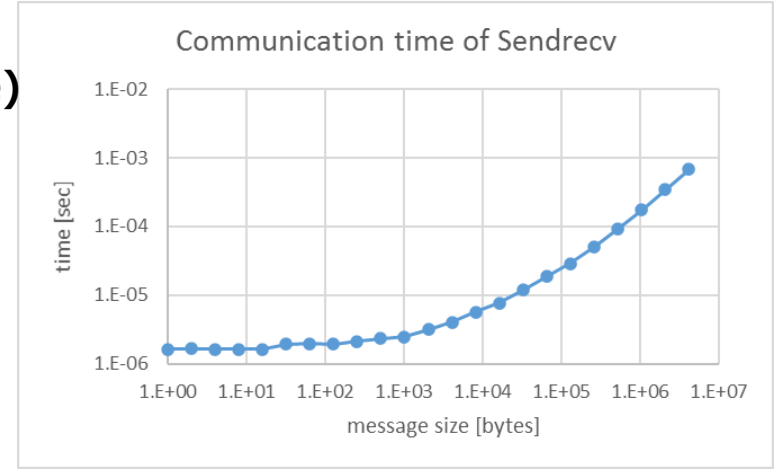


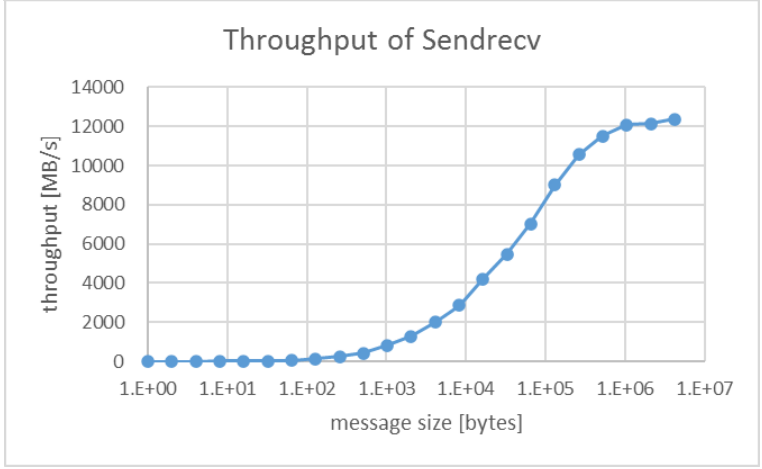


2.3 Sendrecv

384 nodes, 384 procs(1 procs/node)

Message size [bytes]	Communication time [sec]	Throughput [MB/s]
0	1.63E-06	0
1	1.65E-06	1
2	1.66E-06	2
4	1.65E-06	5
8	1.63E-06	10
16	1.65E-06	19
32	1.96E-06	33
64	1.97E-06	65
128	1.96E-06	131
256	2.15E-06	238
512	2.35E-06	436
1024	2.48E-06	824
2048	3.17E-06	1294
4096	4.07E-06	2012
8192	5.70E-06	2876
16384	7.75E-06	4226
32768	1.20E-05	5470
65536	1.86E-05	7032
131072	2.91E-05	9005
262144	4.97E-05	10546
524288	9.12E-05	11495
1048576	1.74E-04	12070
2097152	3.45E-04	12141
4194304	6.78E-04	12381

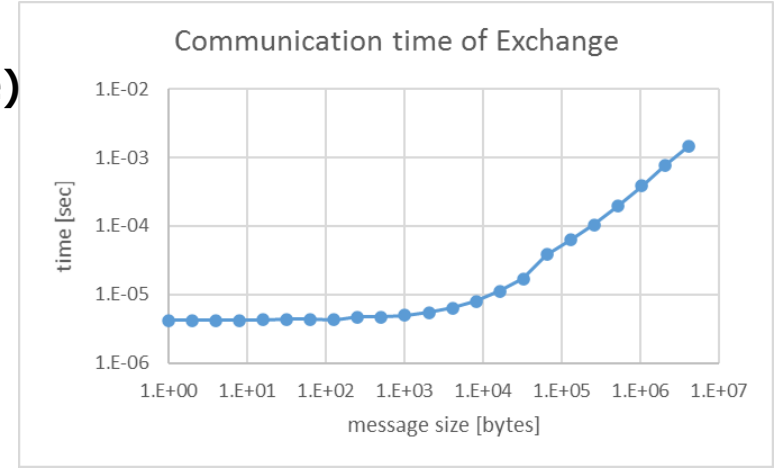


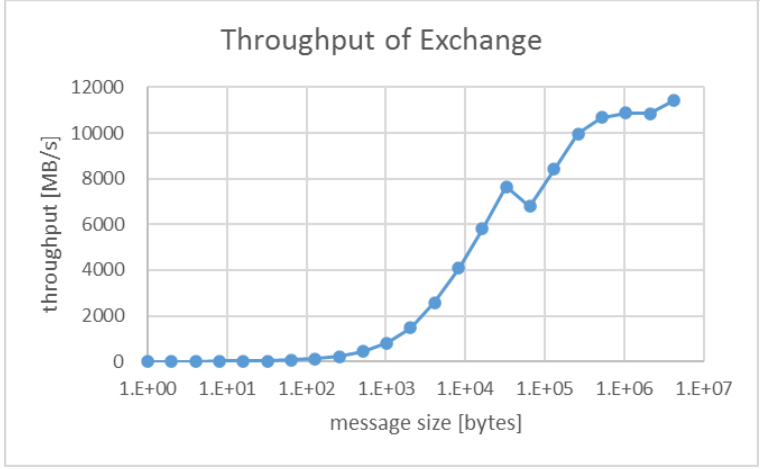


2.4 Exchange

384 nodes, 384 procs(1 procs/node)

Message size [bytes]	Communication time [sec]	Throughput [MB/s]
0	3.87E-06	0
1	4.20E-06	1
2	4.20E-06	2
4	4.21E-06	4
8	4.21E-06	8
16	4.31E-06	15
32	4.39E-06	29
64	4.39E-06	58
128	4.33E-06	118
256	4.70E-06	218
512	4.74E-06	432
1024	5.02E-06	816
2048	5.52E-06	1483
4096	6.37E-06	2572
8192	8.01E-06	4091
16384	1.13E-05	5796
32768	1.71E-05	7657
65536	3.86E-05	6797
131072	6.23E-05	8411
262144	1.05E-04	9971
524288	1.97E-04	10667
1048576	3.86E-04	10880
2097152	7.73E-04	10847
4194304	1.47E-03	11405





3. Collective communications

Targets: Allreduce, Reduce, Allgather, Allgatherv, Gather, Gatherv, Scatter, Scatterv, Alltoall, Alltoallv, Bcast, Barrier

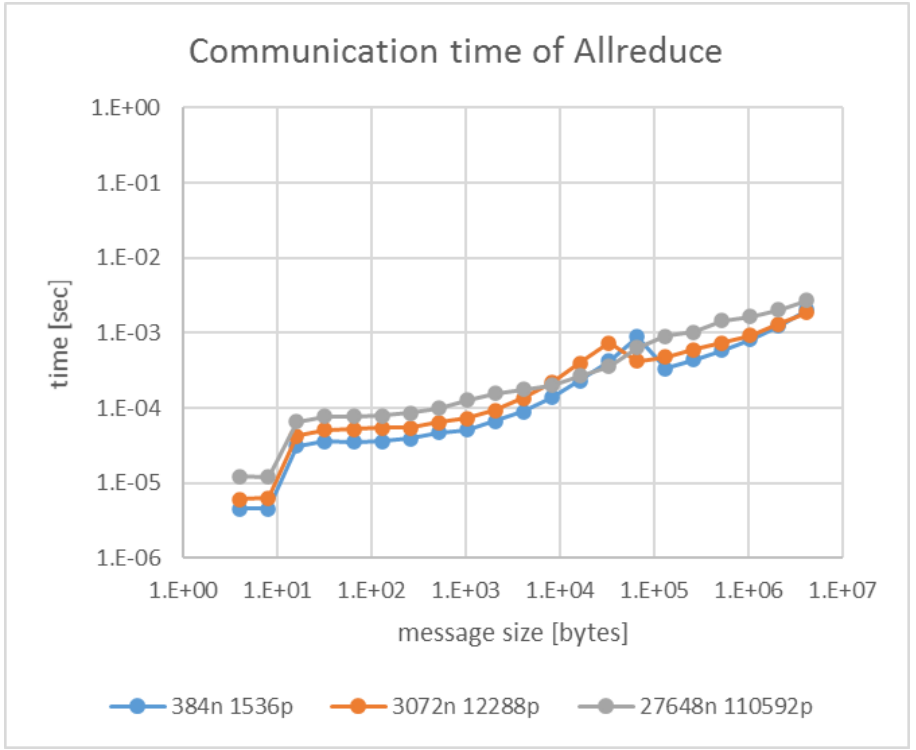
# of parallels : 4 processes / node, 3D torus

- 384 nodes, 1,536 processes, node shape=4x6x16:torus:strict-io
- 3,072 nodes, 12,288 processes, node shape=16x12x16:torus:strict-io
- 27,648 nodes, 110,592 processes, node shape=48x12x48:torus:strict_io

3.1 Allreduce

Communication time [sec]

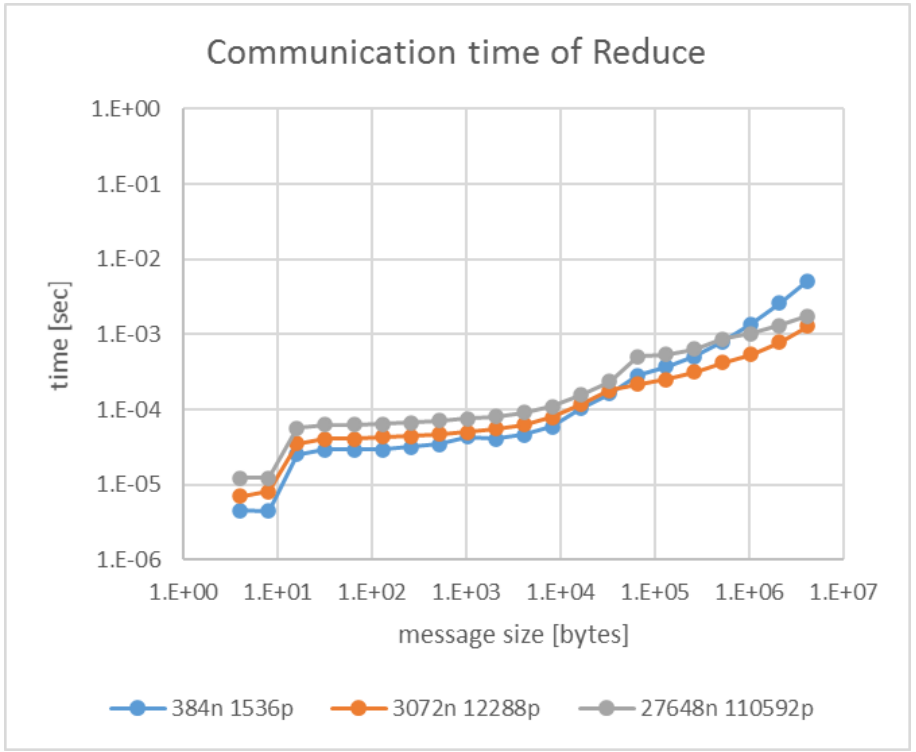
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.20E-07	1.30E-07	1.40E-07
4	4.55E-06	6.11E-06	1.24E-05
8	4.56E-06	6.29E-06	1.21E-05
16	3.11E-05	4.23E-05	6.62E-05
32	3.58E-05	5.10E-05	7.81E-05
64	3.56E-05	5.25E-05	7.80E-05
128	3.63E-05	5.42E-05	7.89E-05
256	3.91E-05	5.50E-05	8.54E-05
512	4.74E-05	6.46E-05	9.89E-05
1024	5.18E-05	7.26E-05	1.27E-04
2048	6.74E-05	9.42E-05	1.55E-04
4096	8.95E-05	1.36E-04	1.77E-04
8192	1.37E-04	2.20E-04	2.01E-04
16384	2.30E-04	3.94E-04	2.67E-04
32768	4.24E-04	7.40E-04	3.57E-04
65536	8.84E-04	4.22E-04	6.39E-04
131072	3.37E-04	4.81E-04	9.07E-04
262144	4.41E-04	6.04E-04	1.02E-03
524288	5.83E-04	7.37E-04	1.48E-03
1048576	8.27E-04	9.26E-04	1.67E-03
2097152	1.23E-03	1.31E-03	2.03E-03
4194304	2.02E-03	1.89E-03	2.74E-03



3.2 Reduce

Communication time [sec]

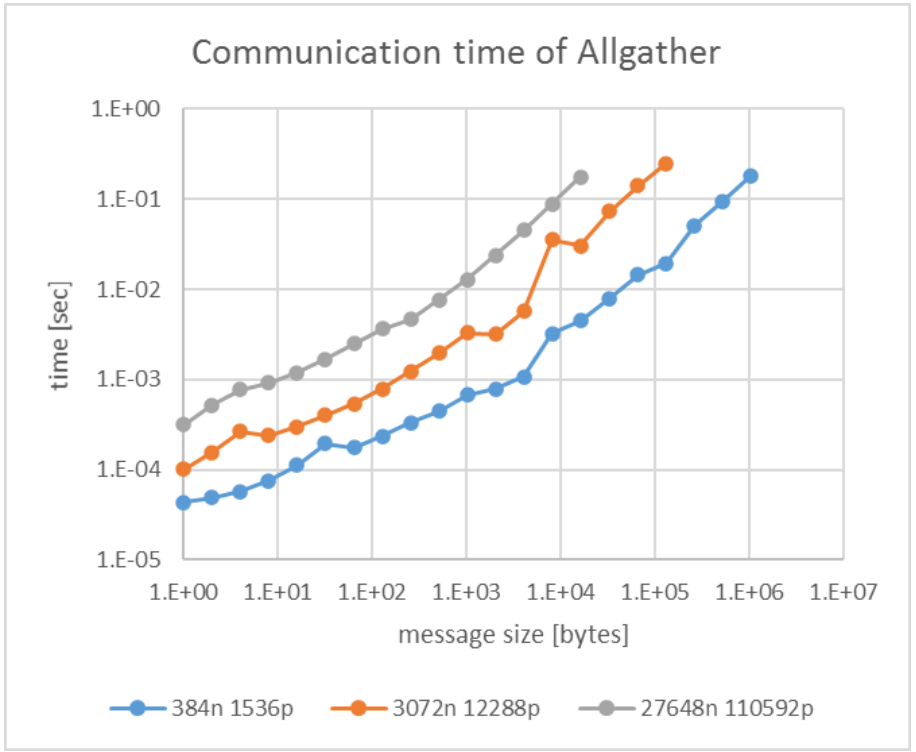
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.20E-07	1.10E-07	1.30E-07
4	4.55E-06	7.06E-06	1.24E-05
8	4.51E-06	8.10E-06	1.24E-05
16	2.55E-05	3.55E-05	5.63E-05
32	2.93E-05	4.05E-05	6.33E-05
64	2.94E-05	4.10E-05	6.34E-05
128	2.93E-05	4.36E-05	6.40E-05
256	3.17E-05	4.41E-05	6.76E-05
512	3.49E-05	4.76E-05	7.20E-05
1024	4.37E-05	5.07E-05	7.59E-05
2048	4.05E-05	5.52E-05	8.11E-05
4096	4.65E-05	6.31E-05	9.10E-05
8192	5.91E-05	7.90E-05	1.11E-04
16384	1.03E-04	1.18E-04	1.55E-04
32768	1.64E-04	1.80E-04	2.35E-04
65536	2.85E-04	2.17E-04	5.07E-04
131072	3.71E-04	2.52E-04	5.47E-04
262144	5.12E-04	3.18E-04	6.37E-04
524288	7.99E-04	4.19E-04	8.74E-04
1048576	1.38E-03	5.42E-04	1.03E-03
2097152	2.61E-03	7.85E-04	1.32E-03
4194304	5.12E-03	1.30E-03	1.76E-03



3.3 Allgather

Communication time [sec]

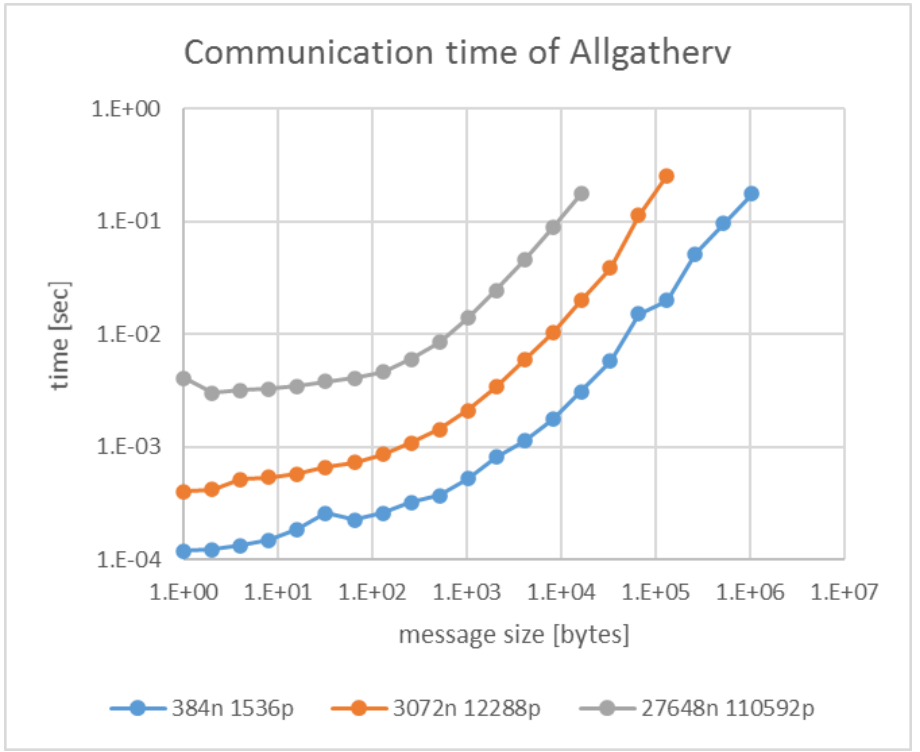
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.50E-07	1.60E-07	1.70E-07
1	4.35E-05	1.02E-04	3.15E-04
2	4.88E-05	1.55E-04	5.22E-04
4	5.78E-05	2.68E-04	7.69E-04
8	7.60E-05	2.39E-04	9.12E-04
16	1.13E-04	3.00E-04	1.19E-03
32	1.95E-04	4.02E-04	1.67E-03
64	1.77E-04	5.40E-04	2.48E-03
128	2.37E-04	7.85E-04	3.63E-03
256	3.34E-04	1.24E-03	4.69E-03
512	4.44E-04	1.95E-03	7.68E-03
1024	6.80E-04	3.27E-03	1.29E-02
2048	7.89E-04	3.16E-03	2.38E-02
4096	1.08E-03	5.74E-03	4.51E-02
8192	3.24E-03	3.55E-02	8.79E-02
16384	4.53E-03	3.07E-02	1.76E-01
32768	7.87E-03	7.37E-02	
65536	1.44E-02	1.42E-01	
131072	1.96E-02	2.51E-01	
262144	5.11E-02		
524288	9.53E-02		
1048576	1.82E-01		



3.4 Allgatherv

Communication time [sec]

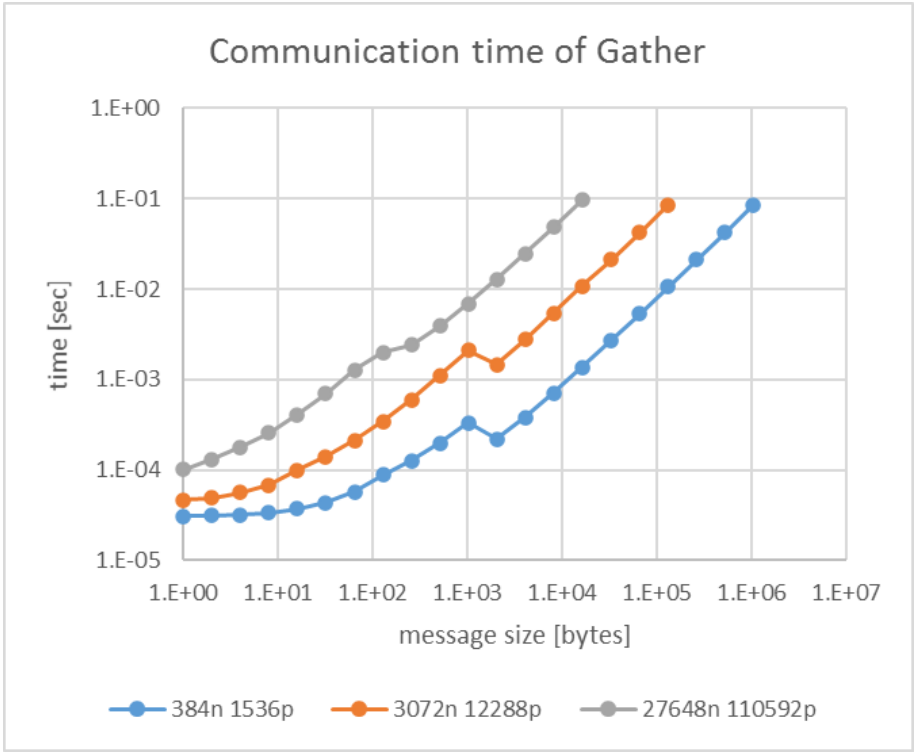
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.99E-06	1.61E-05	1.25E-04
1	1.20E-04	4.03E-04	4.10E-03
2	1.25E-04	4.23E-04	3.04E-03
4	1.35E-04	5.19E-04	3.19E-03
8	1.50E-04	5.39E-04	3.28E-03
16	1.86E-04	5.79E-04	3.47E-03
32	2.59E-04	6.65E-04	3.85E-03
64	2.26E-04	7.29E-04	4.12E-03
128	2.61E-04	8.63E-04	4.67E-03
256	3.24E-04	1.09E-03	6.03E-03
512	3.75E-04	1.43E-03	8.54E-03
1024	5.27E-04	2.12E-03	1.39E-02
2048	8.17E-04	3.41E-03	2.48E-02
4096	1.14E-03	5.93E-03	4.60E-02
8192	1.78E-03	1.04E-02	8.90E-02
16384	3.11E-03	2.00E-02	1.77E-01
32768	5.80E-03	3.90E-02	
65536	1.51E-02	1.14E-01	
131072	2.00E-02	2.58E-01	
262144	5.18E-02		
524288	9.58E-02		
1048576	1.77E-01		



3.5 Gather

Communication time [sec]

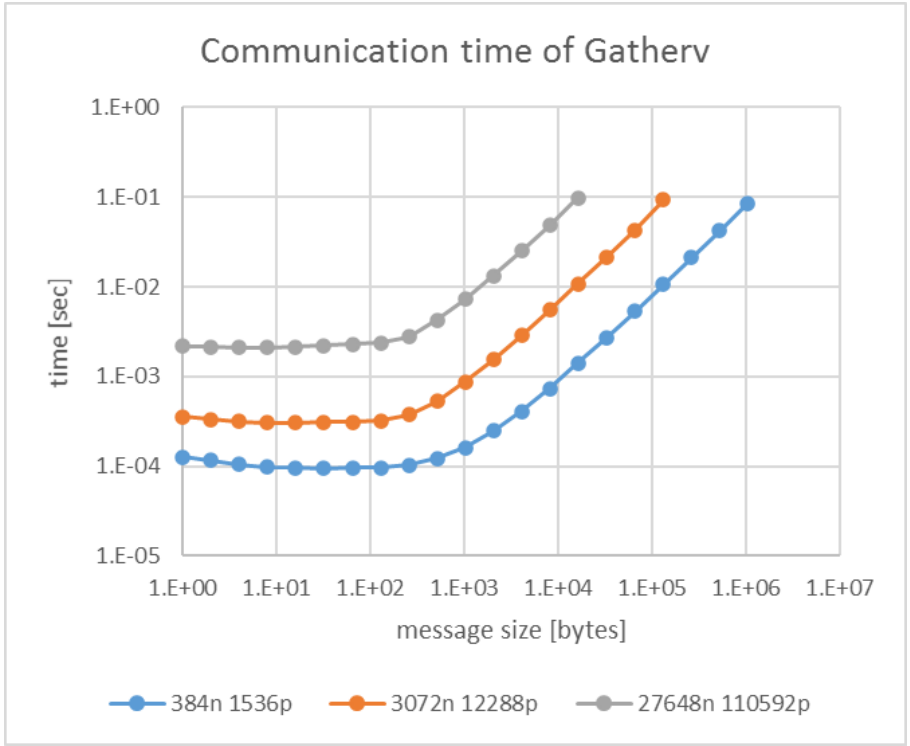
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.20E-07	1.20E-07	1.40E-07
1	3.06E-05	4.66E-05	1.02E-04
2	3.13E-05	4.90E-05	1.31E-04
4	3.20E-05	5.60E-05	1.78E-04
8	3.38E-05	6.87E-05	2.59E-04
16	3.71E-05	9.94E-05	4.08E-04
32	4.39E-05	1.40E-04	6.99E-04
64	5.73E-05	2.11E-04	1.27E-03
128	8.82E-05	3.44E-04	1.99E-03
256	1.27E-04	6.02E-04	2.40E-03
512	1.98E-04	1.11E-03	3.89E-03
1024	3.30E-04	2.12E-03	6.86E-03
2048	2.21E-04	1.46E-03	1.28E-02
4096	3.85E-04	2.77E-03	2.46E-02
8192	7.13E-04	5.40E-03	4.83E-02
16384	1.37E-03	1.08E-02	9.62E-02
32768	2.70E-03	2.14E-02	
65536	5.35E-03	4.26E-02	
131072	1.07E-02	8.57E-02	
262144	2.13E-02		
524288	4.25E-02		
1048576	8.49E-02		



3.6 Gatherv

Communication time [sec]

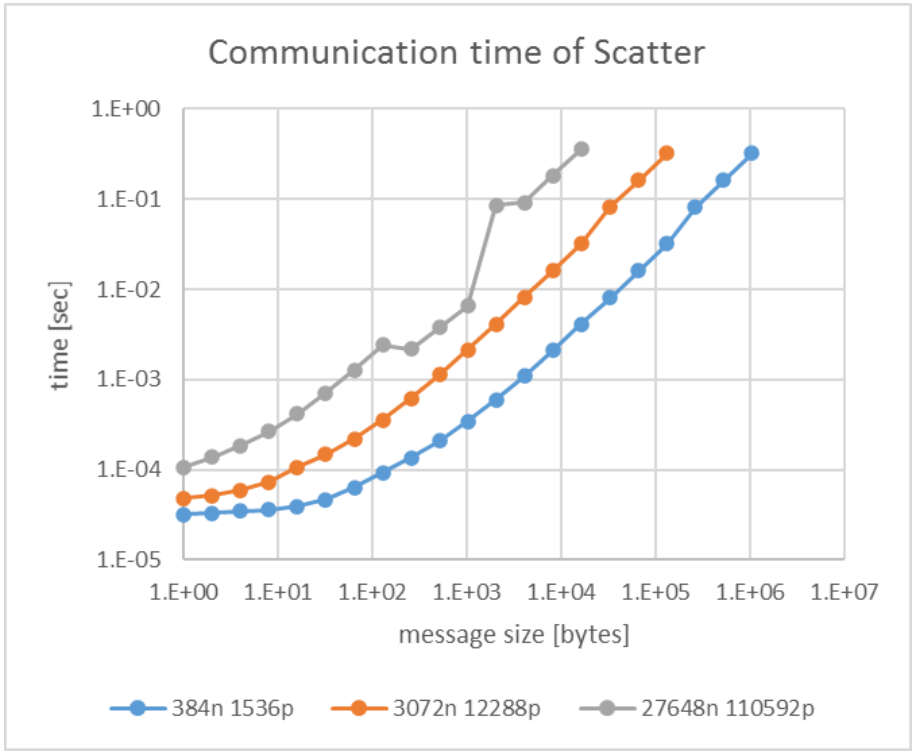
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	8.21E-05	1.42E-04	5.48E-04
1	1.27E-04	3.56E-04	2.18E-03
2	1.16E-04	3.36E-04	2.15E-03
4	1.06E-04	3.14E-04	2.12E-03
8	9.78E-05	3.07E-04	2.12E-03
16	9.61E-05	3.05E-04	2.13E-03
32	9.53E-05	3.09E-04	2.23E-03
64	9.56E-05	3.12E-04	2.29E-03
128	9.57E-05	3.22E-04	2.36E-03
256	1.03E-04	3.74E-04	2.79E-03
512	1.22E-04	5.28E-04	4.29E-03
1024	1.60E-04	8.69E-04	7.29E-03
2048	2.48E-04	1.53E-03	1.32E-02
4096	4.12E-04	2.85E-03	2.51E-02
8192	7.40E-04	5.48E-03	4.87E-02
16384	1.41E-03	1.08E-02	9.77E-02
32768	2.73E-03	2.14E-02	
65536	5.38E-03	4.32E-02	
131072	1.07E-02	9.38E-02	
262144	2.13E-02		
524288	4.25E-02		
1048576	8.50E-02		



3.7 Scatter

Communication time [sec]

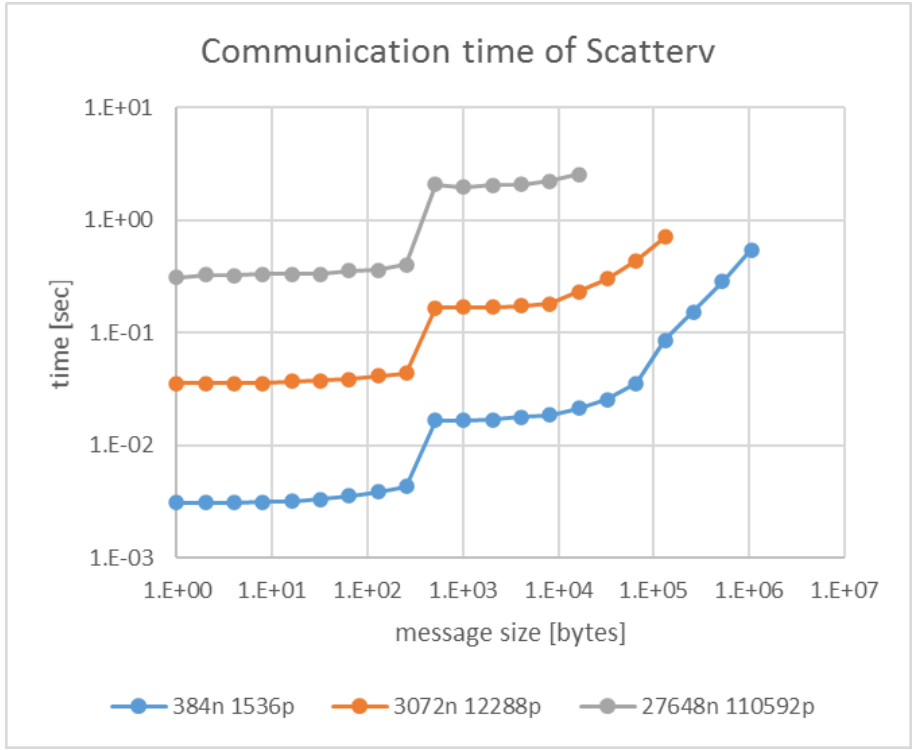
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.20E-07	1.10E-07	1.40E-07
1	3.19E-05	4.80E-05	1.07E-04
2	3.28E-05	5.15E-05	1.38E-04
4	3.46E-05	5.96E-05	1.85E-04
8	3.60E-05	7.33E-05	2.67E-04
16	3.91E-05	1.07E-04	4.18E-04
32	4.68E-05	1.48E-04	7.10E-04
64	6.32E-05	2.20E-04	1.28E-03
128	9.27E-05	3.54E-04	2.42E-03
256	1.36E-04	6.16E-04	2.17E-03
512	2.09E-04	1.13E-03	3.78E-03
1024	3.42E-04	2.13E-03	6.57E-03
2048	5.99E-04	4.15E-03	8.51E-02
4096	1.10E-03	8.16E-03	9.20E-02
8192	2.11E-03	1.62E-02	1.81E-01
16384	4.11E-03	3.22E-02	3.58E-01
32768	8.12E-03	8.25E-02	
65536	1.61E-02	1.62E-01	
131072	3.21E-02	3.21E-01	
262144	8.19E-02		
524288	1.61E-01		
1048576	3.21E-01		



3.8 Scatterv

Communication time [sec]

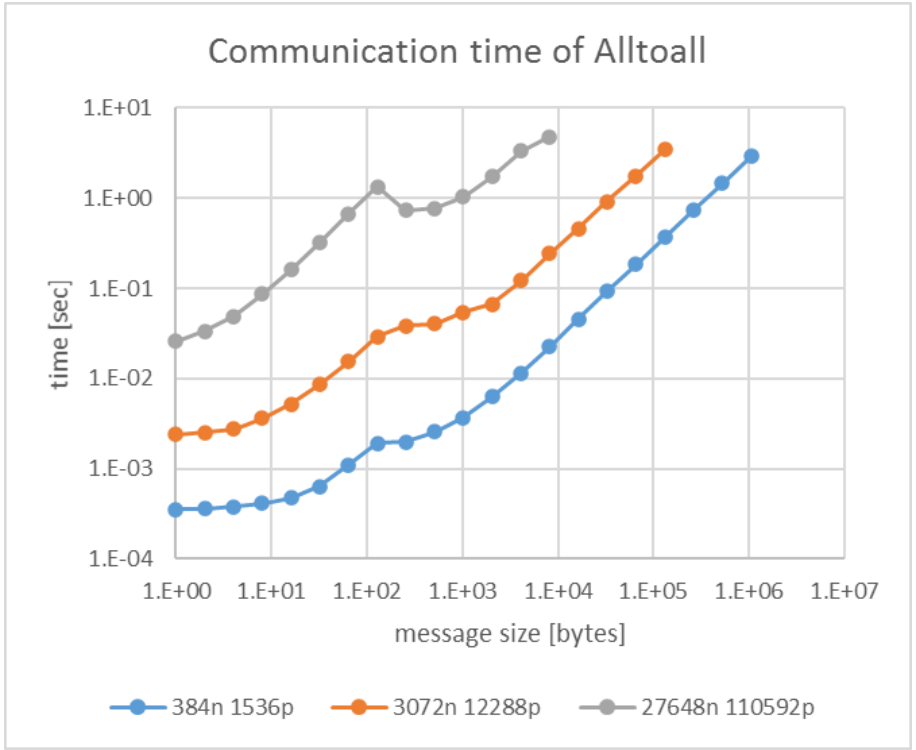
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	8.76E-06	3.73E-05	2.66E-04
1	3.10E-03	3.58E-02	3.10E-01
2	3.11E-03	3.57E-02	3.31E-01
4	3.12E-03	3.57E-02	3.25E-01
8	3.14E-03	3.57E-02	3.32E-01
16	3.18E-03	3.71E-02	3.32E-01
32	3.35E-03	3.77E-02	3.33E-01
64	3.58E-03	3.89E-02	3.55E-01
128	3.86E-03	4.16E-02	3.60E-01
256	4.35E-03	4.39E-02	4.06E-01
512	1.66E-02	1.68E-01	2.09E+00
1024	1.67E-02	1.69E-01	1.97E+00
2048	1.70E-02	1.69E-01	2.05E+00
4096	1.79E-02	1.75E-01	2.08E+00
8192	1.88E-02	1.81E-01	2.23E+00
16384	2.15E-02	2.34E-01	2.55E+00
32768	2.58E-02	3.02E-01	
65536	3.55E-02	4.40E-01	
131072	8.65E-02	7.15E-01	
262144	1.53E-01		
524288	2.88E-01		
1048576	5.49E-01		



3.9 Alltoall

Communication time [sec]

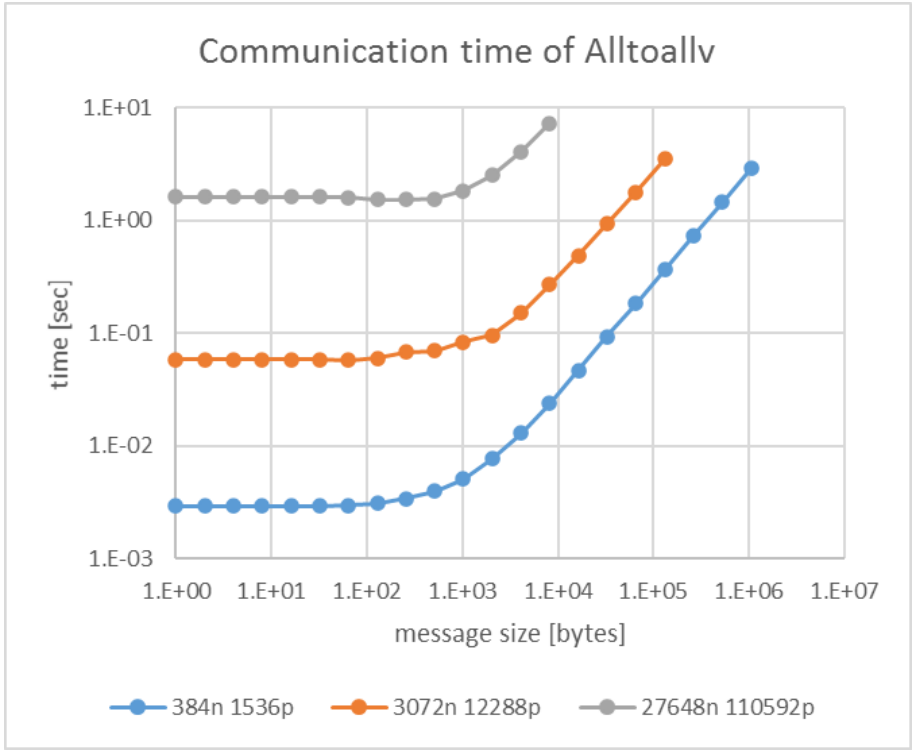
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.10E-07	1.20E-07	1.40E-07
1	3.52E-04	2.40E-03	2.59E-02
2	3.58E-04	2.51E-03	3.33E-02
4	3.78E-04	2.78E-03	4.90E-02
8	4.12E-04	3.61E-03	8.66E-02
16	4.79E-04	5.22E-03	1.60E-01
32	6.33E-04	8.62E-03	3.20E-01
64	1.11E-03	1.53E-02	6.67E-01
128	1.93E-03	2.90E-02	1.34E+00
256	1.99E-03	3.84E-02	7.35E-01
512	2.58E-03	4.07E-02	7.67E-01
1024	3.67E-03	5.43E-02	1.04E+00
2048	6.27E-03	6.73E-02	1.73E+00
4096	1.16E-02	1.22E-01	3.33E+00
8192	2.24E-02	2.42E-01	4.76E+00
16384	4.56E-02	4.62E-01	
32768	9.20E-02	9.10E-01	
65536	1.83E-01	1.76E+00	
131072	3.66E-01	3.47E+00	
262144	7.30E-01		
524288	1.46E+00		
1048576	2.92E+00		



3.10 Alltoallv

Communication time [sec]

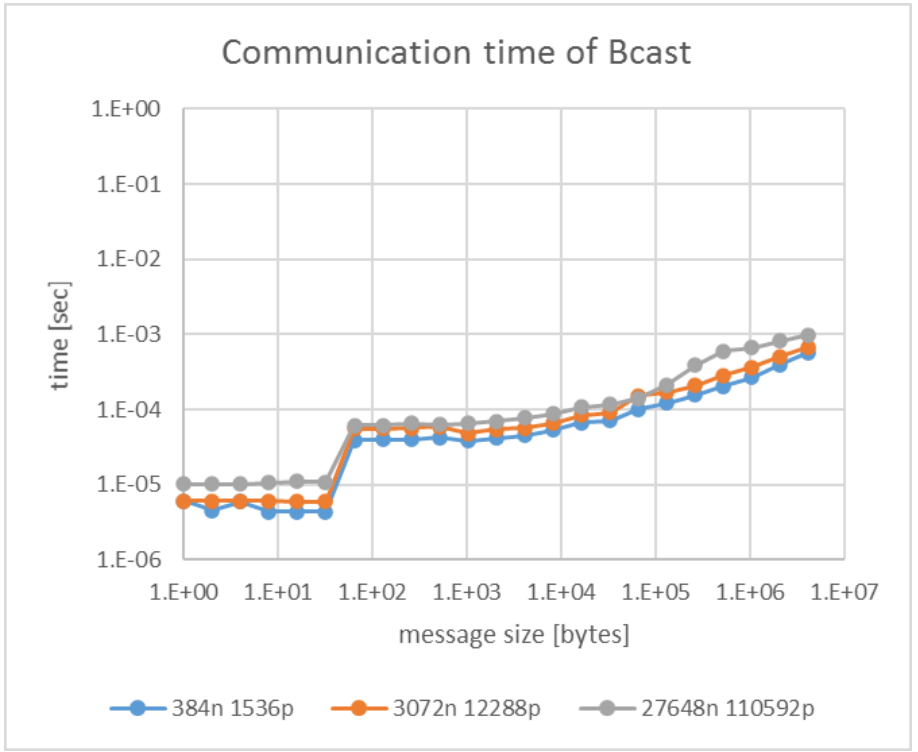
Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.63E-03	2.96E-02	8.10E-01
1	2.96E-03	5.86E-02	1.63E+00
2	2.96E-03	5.87E-02	1.63E+00
4	2.96E-03	5.87E-02	1.62E+00
8	2.96E-03	5.86E-02	1.62E+00
16	2.97E-03	5.88E-02	1.63E+00
32	2.96E-03	5.88E-02	1.63E+00
64	2.98E-03	5.83E-02	1.61E+00
128	3.12E-03	6.05E-02	1.55E+00
256	3.42E-03	6.82E-02	1.53E+00
512	4.01E-03	7.01E-02	1.57E+00
1024	5.12E-03	8.40E-02	1.83E+00
2048	7.71E-03	9.66E-02	2.53E+00
4096	1.31E-02	1.52E-01	4.12E+00
8192	2.38E-02	2.71E-01	7.31E+00
16384	4.70E-02	4.91E-01	
32768	9.33E-02	9.41E-01	
65536	1.85E-01	1.79E+00	
131072	3.68E-01	3.50E+00	
262144	7.32E-01		
524288	1.46E+00		
1048576	2.92E+00		



3.11 Bcast

Communication time [sec]

Message size [bytes]	384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
0	1.10E-07	1.00E-07	1.10E-07
1	6.11E-06	6.15E-06	1.02E-05
2	4.61E-06	6.05E-06	1.02E-05
4	6.00E-06	6.11E-06	1.02E-05
8	4.37E-06	6.15E-06	1.06E-05
16	4.40E-06	6.01E-06	1.11E-05
32	4.37E-06	5.98E-06	1.08E-05
64	3.96E-05	5.56E-05	6.12E-05
128	3.98E-05	5.61E-05	6.17E-05
256	4.04E-05	5.65E-05	6.52E-05
512	4.29E-05	5.96E-05	6.34E-05
1024	3.84E-05	4.84E-05	6.60E-05
2048	4.14E-05	5.50E-05	7.05E-05
4096	4.56E-05	5.74E-05	7.78E-05
8192	5.39E-05	6.63E-05	8.72E-05
16384	6.78E-05	8.34E-05	1.08E-04
32768	7.12E-05	9.06E-05	1.18E-04
65536	1.01E-04	1.53E-04	1.43E-04
131072	1.22E-04	1.69E-04	2.15E-04
262144	1.58E-04	2.09E-04	3.93E-04
524288	2.03E-04	2.84E-04	5.96E-04
1048576	2.66E-04	3.63E-04	6.73E-04
2097152	4.00E-04	5.10E-04	8.11E-04
4194304	5.77E-04	6.76E-04	9.81E-04



3.12 Barrier

Communication time [sec]

384 nodes 1536 procs	3072 nodes 12288 procs	27648 nodes 110592 procs
4.32E-06	5.78E-06	9.83E-06

4. All to all

Latency and throughput

Measure the performance of Alltoall

- **Since the performance of Alltoall depends on the bisection bandwith, the performance will be better when the shape of X,Y,Z axes (excluding a,b,x axes) takes cubic.**
  - Performance was measured in the form shown in the table.
- **Measurement conditions**
  - Using strict option to fix the form
  - Language version : lang/tcsds-1.2.31
  - Benchmark : osu-micro-benchmarks-5.7.1
  - # of parallels : 1 process / node
- **Note**
  - Form of 4x6x16 takes long Z axis due to the limit of resource group.

Table. Measured forms

Form	2x3x2	2x3x4	4x3x4	4x6x4	4x6x8	4x6x16	8x6x16	8x12x16	16x12x16	16x24x16	16x24x32
# of nodes	12	24	48	96	192	384	768	1536	3072	6144	12288
resource group	small-torus						large				

Measure the performance of Alltoall (cont.)

- **Performance was measured using default algorithm (unspecified) or marked (○) algorithms in the table due to various restrictions (ex. crp is only available on 1 node.)**
  - In case of default mode, one of algorithms in the table is selected.

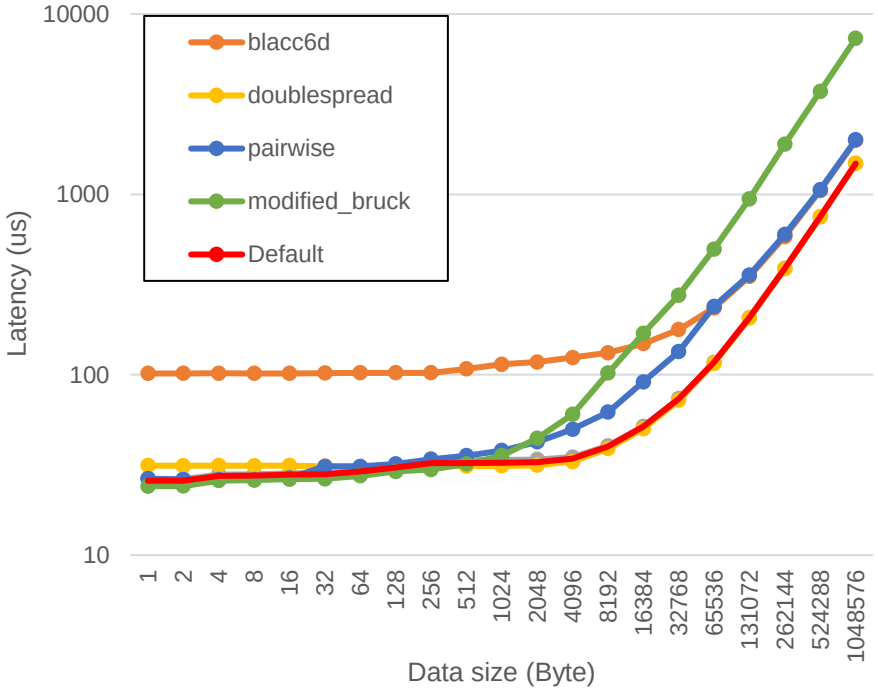
8.4.1.4 coll_select_alltoall_algorithm (Specifying the Algorithm of the MPI_ALLTOALL Routine)

Fix the algorithm to be executed in the MPI_ALLTOALL routine in the program to specific algorithm at all times.

Used	Value of MCA Parameter	Contents
	crp	Use the algorithm crp tuned for Tofu interconnect.
○	blacc6d	Use the algorithm blacc6d tuned for Tofu interconnect.
○	blacc3d	Use the algorithm blacc3d tuned for Tofu interconnect.
○	doublespread	Use the algorithm doublespread tuned for Tofu interconnect.
	two_proc	Use the algorithm two_proc implemented with the Open MPI.
	linear_sync	Use the algorithm linear_sync implemented with the Open MPI.
○	modified_bruck	Use the algorithm modified_bruck implemented with the Open MPI.
○	pairwise	Use the algorithm pairwise implemented with the Open MPI.
	linear	Use the algorithm linear implemented with the Open MPI.

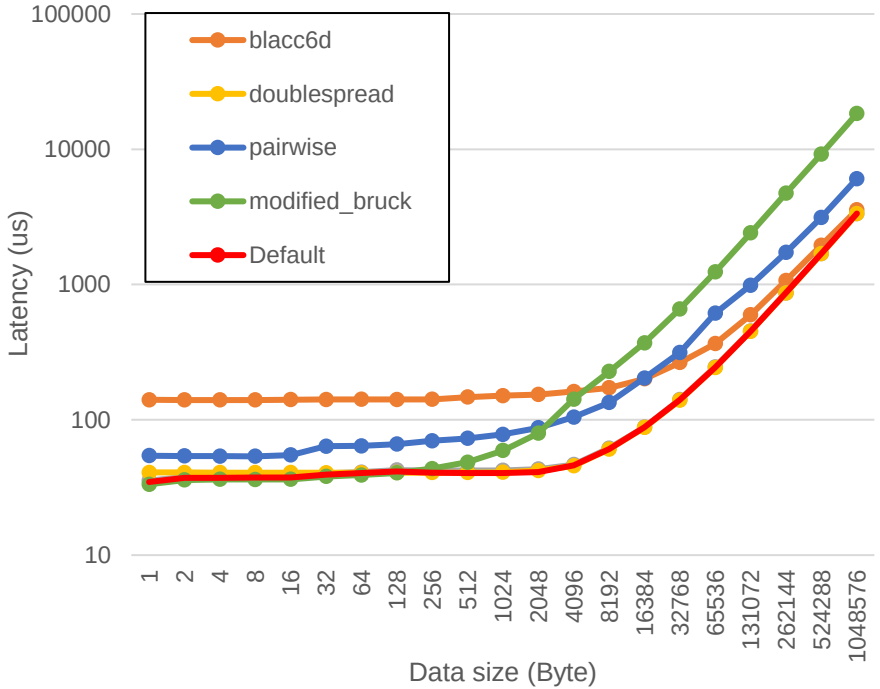
Latency (1/6)

12 nodes (2x3x2)



- **blacc3d was unavailable**

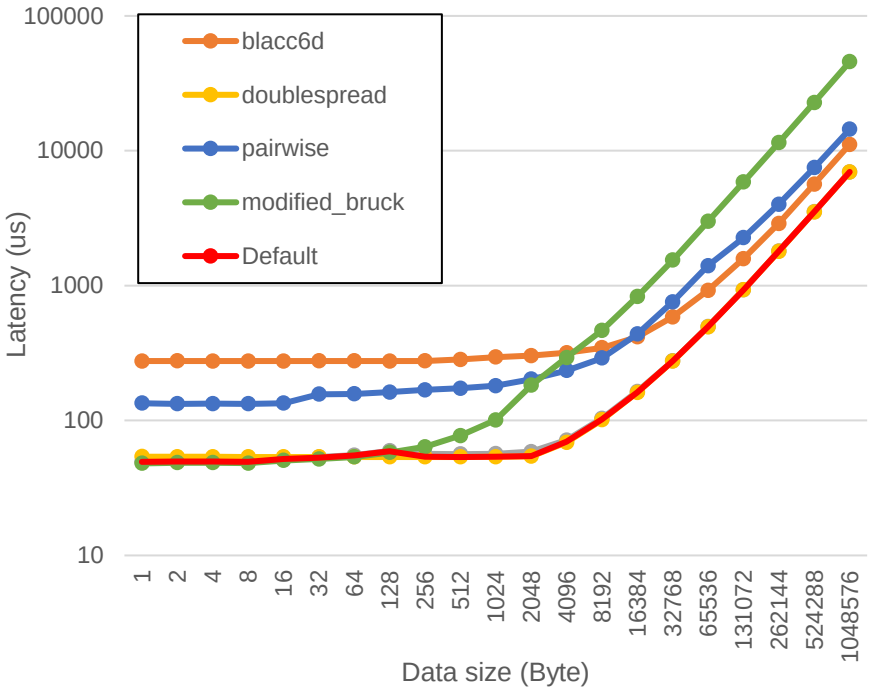
24 nodes (2x3x4)



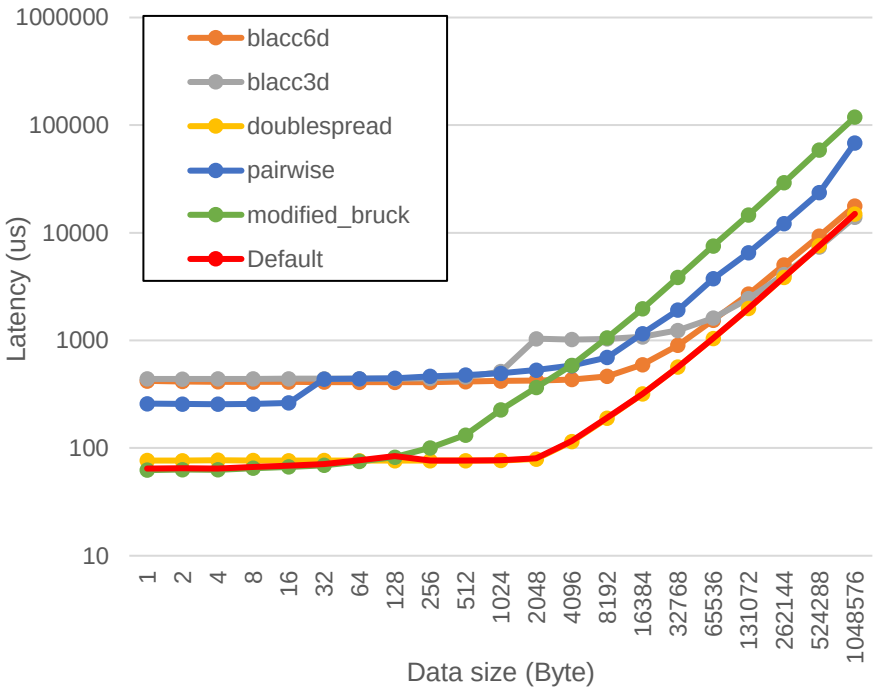
- **blacc3d was unavailable**

Latency (2/6)

48 nodes (4x3x4)



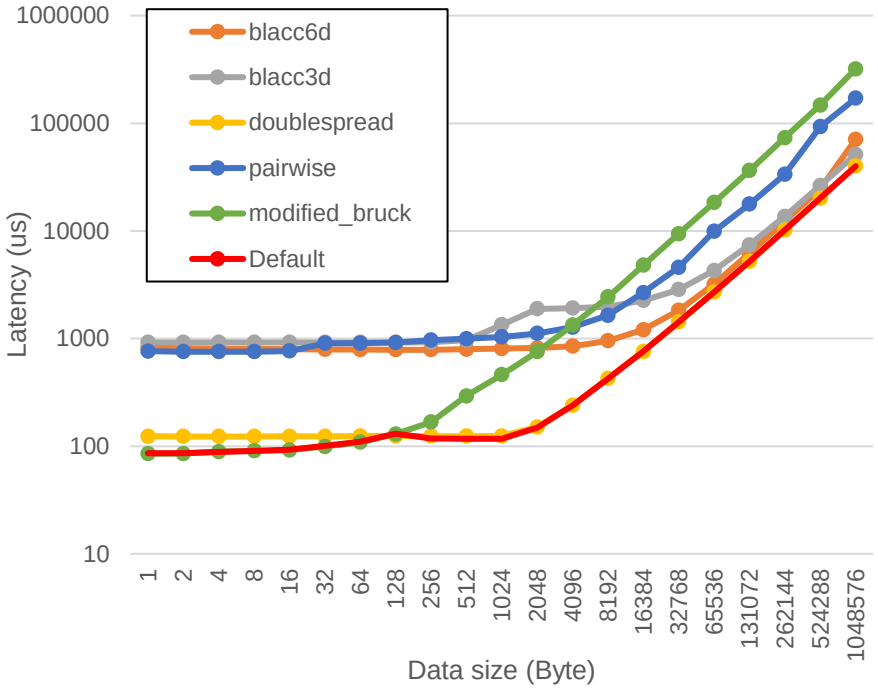
96 nodes (4x6x4)



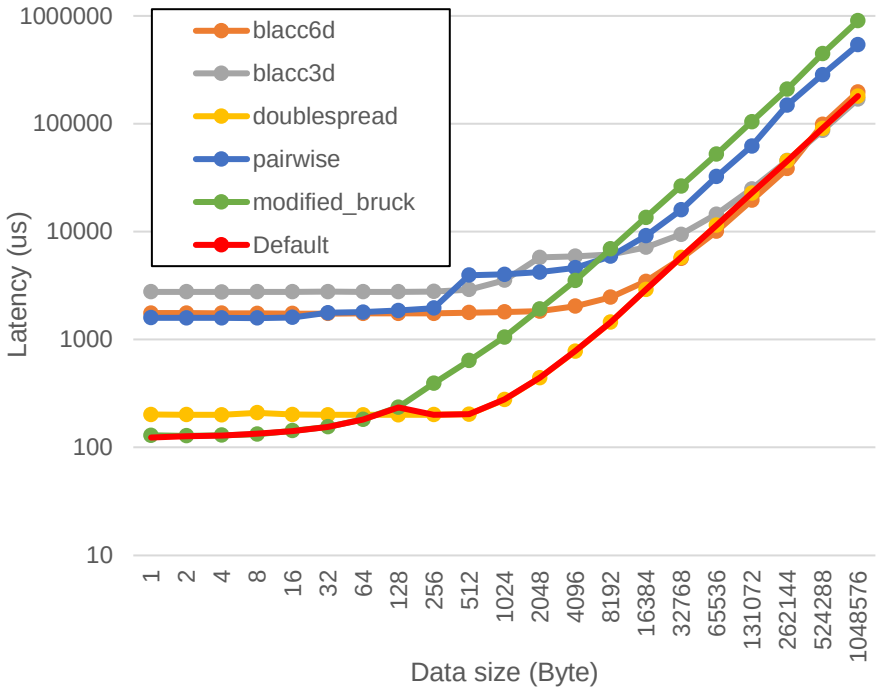
- **blacc3d was unavailable**

Latency (3/6)

192 nodes (4x6x8)

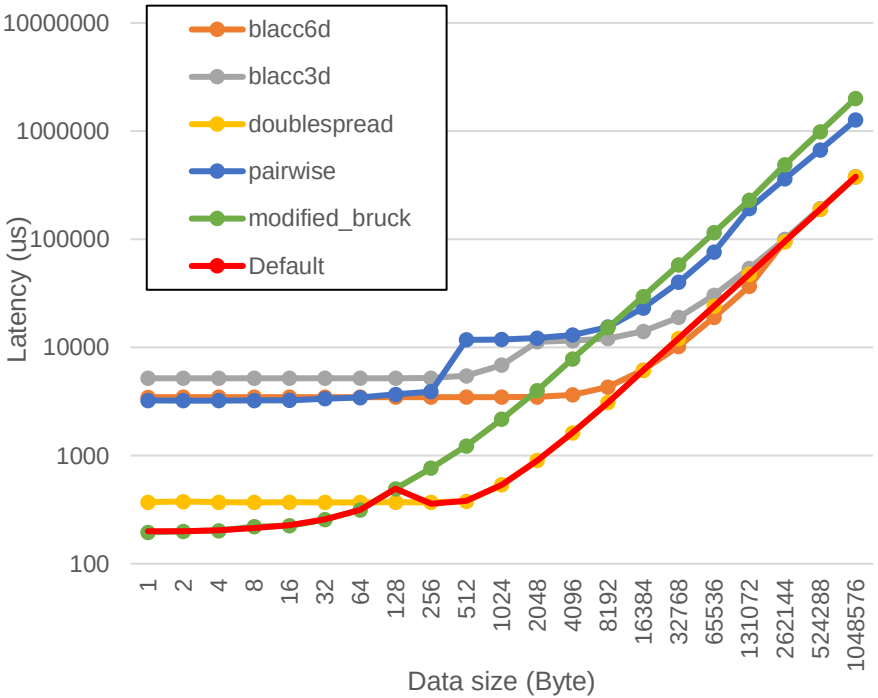


384 nodes (4x6x16)

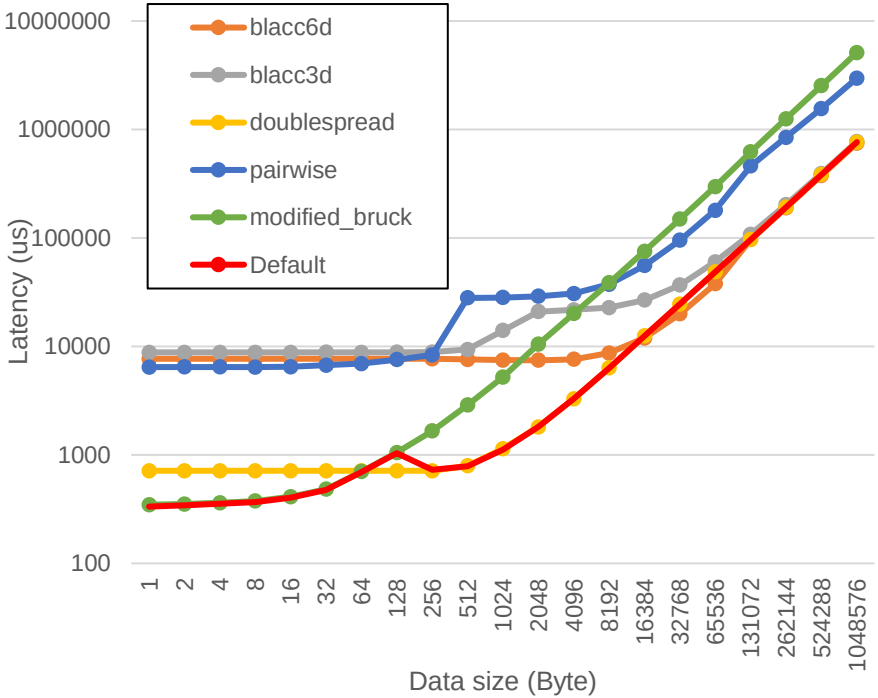


Latency (4/6)

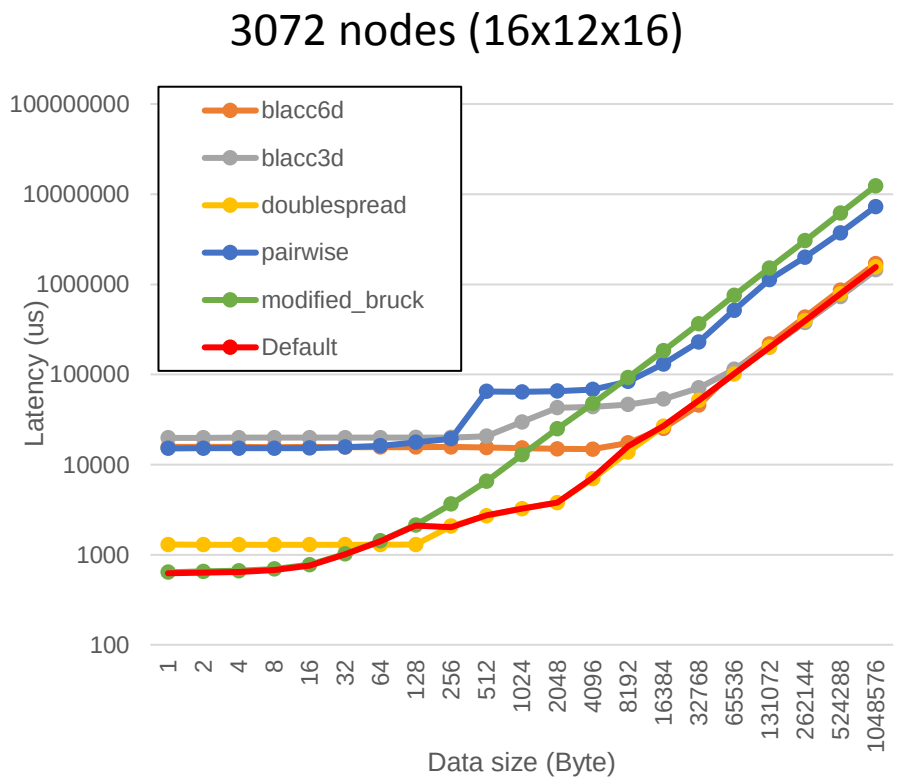
768 nodes (8x6x16)

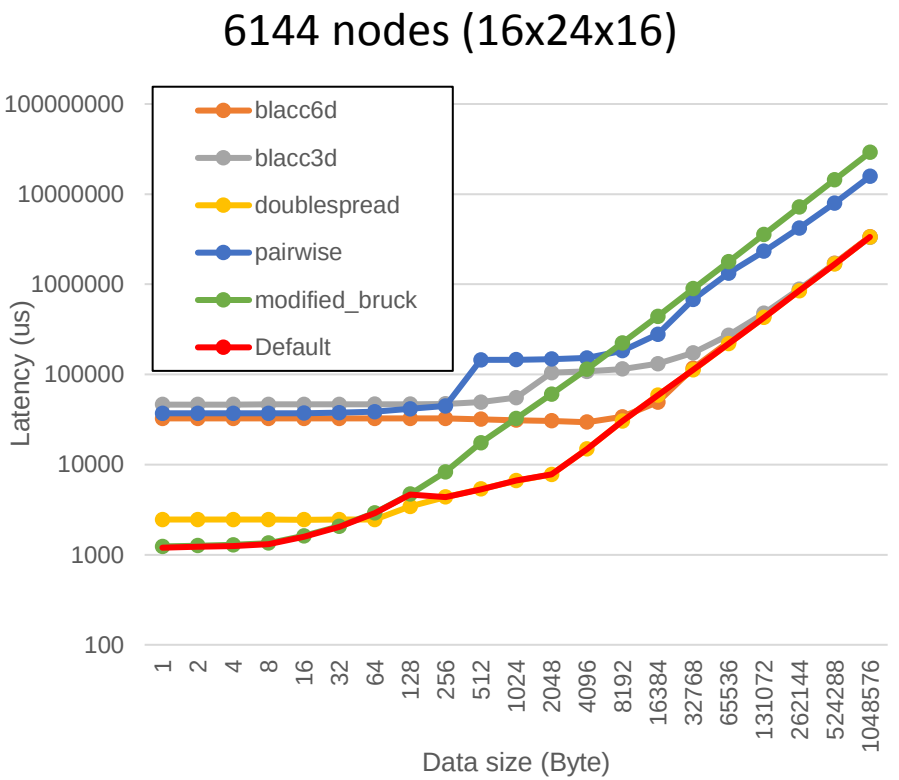


1536 nodes (8x12x16)

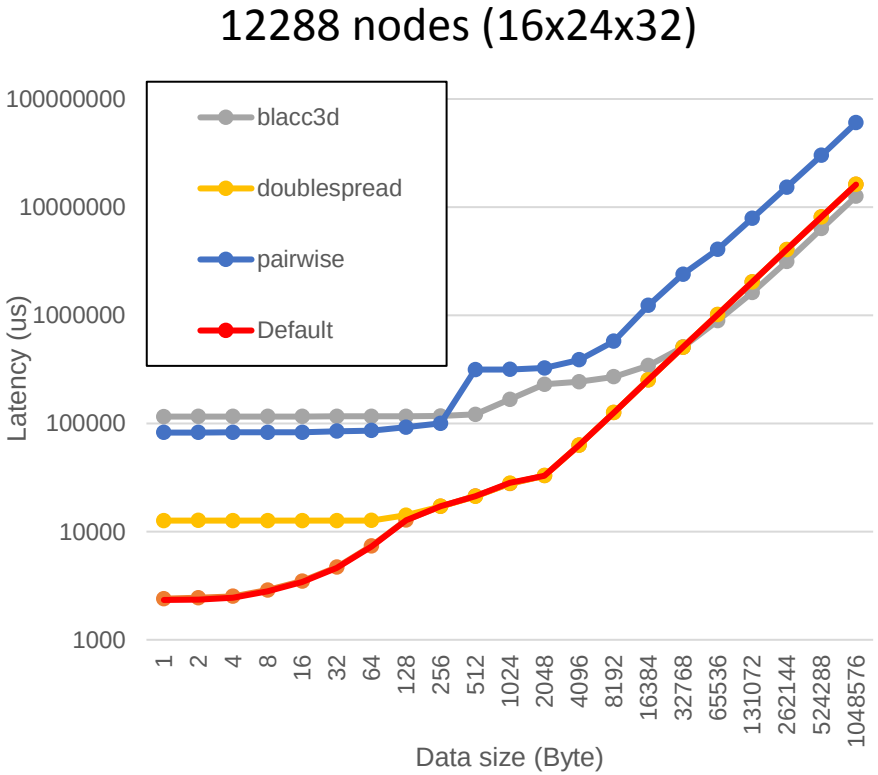


Latency (5/6)





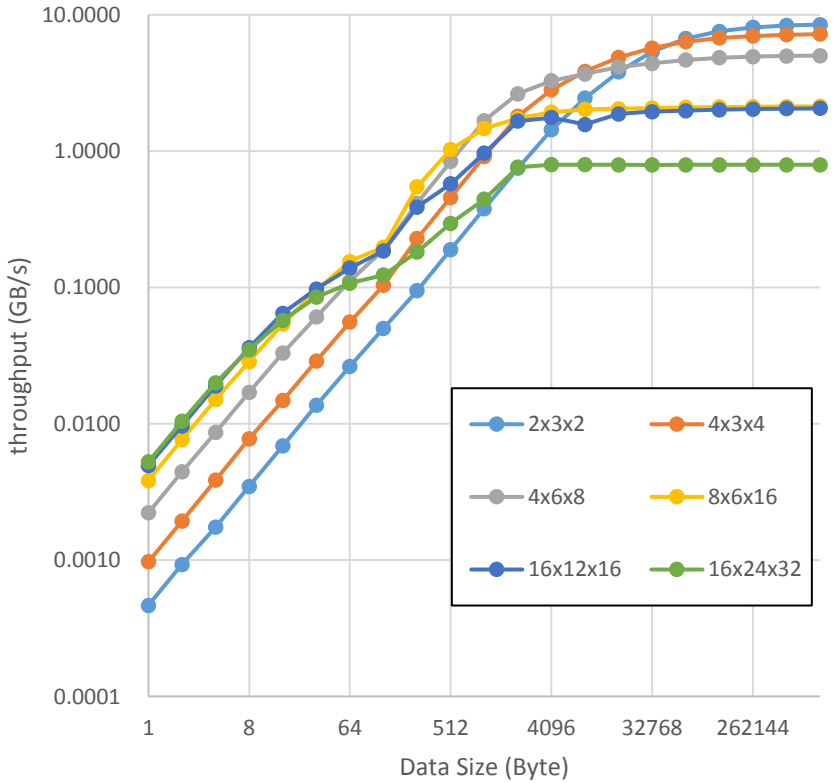
Latency (6/6)



- **blacc6d was unavailable**
- **modified_bruck was omitted due to time constraints**

- **Summary**
  - ***Default*** is fastest
    - When data size is small, ***modified_bruck*** is selected, and when data size is large, ***doublespread*** is selected
      - based on MPI statistics
    - Threshold value is determined on # of nodes
      - When # of nodes is large, threshold value is large and ***doublespread*** is selected
  - Algorithms tuned for Tofu is not fast in every case.
    - When data size is small, performance of ***blacc6d*** is slow
    - When data size is small, performance of ***modified_bruck*** is fastest

Throughput (data size x # of nodes / time)



- **Result using default mode for each node size**
  - Performance of throughput is decreased in proportion of # of nodes.

5. MPI process generation time

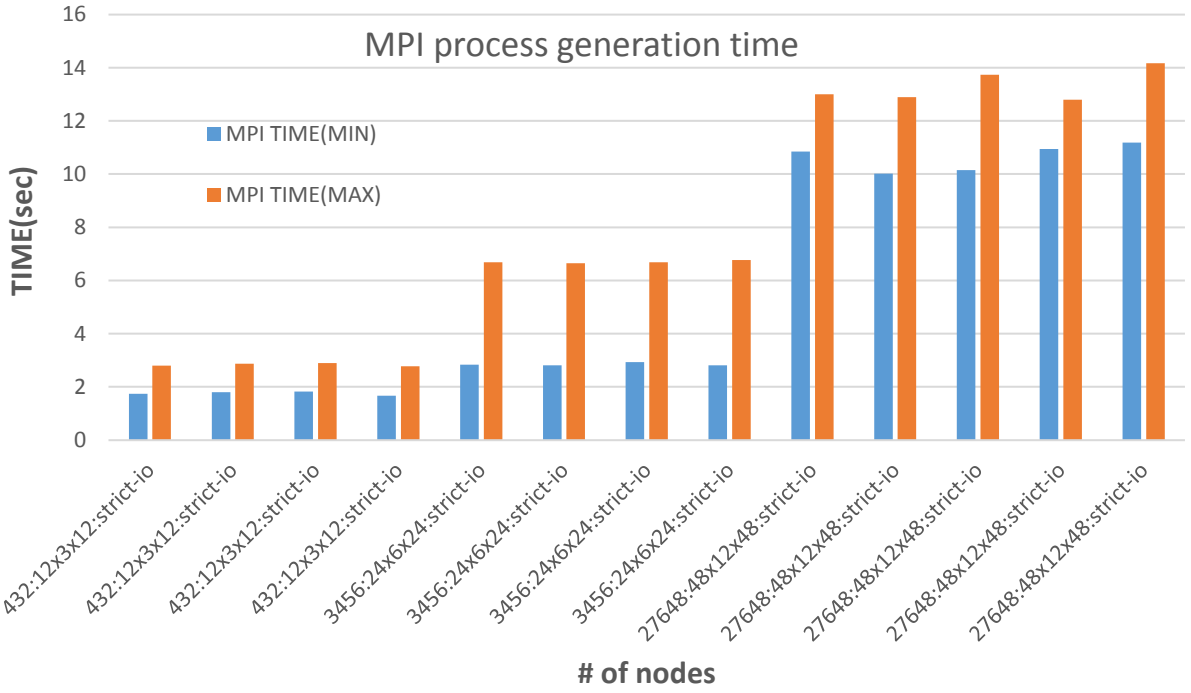
of nodes

- 432:12x3x12:strict-io
- 3456:24x6x24:strict-io
- 27648:48x12x48:strict-io

How to measure

1. Record time stamp before execution of mpiexec in a jobscript - (A)
2. Execute date by mpiexec in a jobscript
3. TCS (PLE) generate a process for date on every node
4. Each process on every node executes date -(B)
5. Record a difference of (A) and (B) in all ranks
6. Calculate minimum time and maximum time of differences on all ranks

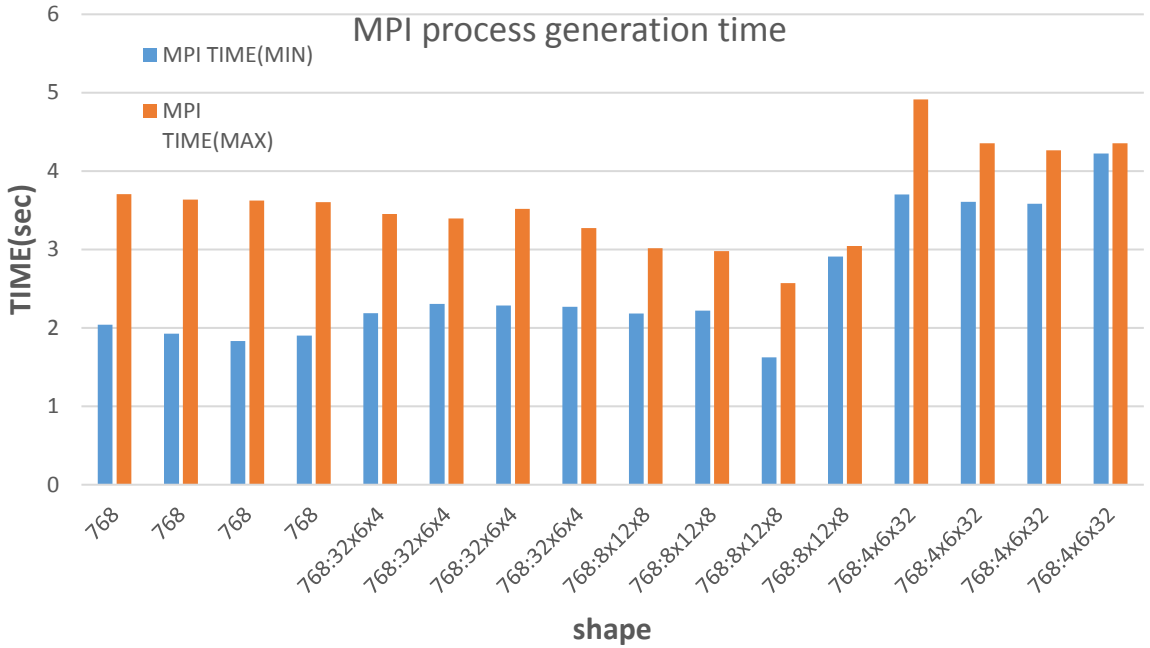
MPI process generation time (# of nodes)



# of Nodes	TIME(MIN)	TIME(MAX)
432:12x3x12	1.734	2.798
432:12x3x12	1.795	2.874
432:12x3x12	1.82	2.898
432:12x3x12	1.666	2.777
3456:24x6x24	2.834	6.684
3456:24x6x24	2.809	6.644
3456:24x6x24	2.937	6.686
3456:24x6x24	2.809	6.77
27648:48x12x48	10.847	12.998
27648:48x12x48	10.012	12.888
27648:48x12x48	10.149	13.734
27648:48x12x48	10.939	12.794
27648:48x12x48	11.188	14.173

- **MPI process generation time increases in proportion to # of nodes**

MPI process generation time (shape)



Nodes(shape)	TIME(MIN)	TIME(MAX)
768:noncont	2.041	3.703
768:noncont	1.926	3.637
768:noncont	1.829	3.621
768:noncont	1.900	3.602
768:32x6x4	2.187	3.450
768:32x6x4	2.303	3.395
768:32x6x4	2.284	3.516
768:32x6x4	2.269	3.272
768:8x12x8	2.183	3.013
768:8x12x8	2.219	2.979
768:8x12x8	1.625	2.568
768:8x12x8	2.909	3.043
768:4x6x32	3.701	4.911
768:4x6x32	3.609	4.353
768:4x6x32	3.584	4.266
768:4x6x32	4.223	4.354

- **MPI process generation time is almost not affected by job shape**

Appendix

- How to compile
- Example of job script

How to compile

```
gtar zxf IMB-v2021.2.tar.gz

cd mpi-benchmarks-IMB-v2021.2
make ¥
  CC=mpifccpx ¥
  CXX=mpiFCCpx ¥
  CFLAGS="-Nclang" ¥
  CXXFLAGS="-Nclang" ¥
  IMB-MPI1
cd ..

ls -l mpi-benchmarks-IMB-v2021.2/src_cpp/IMB-MPI1
```

Example of job script

```
#!/bin/bash -x
#PJM -L elapse=30:00
#PJM -L rscgrp="resource group name"
#PJM -L node=4x6x16:torus:strict-io
#PJM --mpi proc=1536
#PJM -j
#PJM -s

export LANG=C
NUM_PROCS=${PJM_MPI_PROC}

llio_transfer ./IMB-MPI1

export PLE_MPI_STD_EMPTYFILE=off

rm -f output.*
/usr/bin/time -p mpiexec -of-proc output ¥
  ./IMB-MPI1 -npmin ${NUM_PROCS} -time 100.0 -mem 3.0 Allreduce

llio_transfer --purge ./IMB-MPI1
```

Example of job script

```
#!/bin/bash
#PJM -L "node=48x12x48:strict-io"
#PJM -L “rscgrp=“resource group name”
#PJM -L "elapse=00:30:00"

export PLE_MPI_STD_EMPTYFILE=off

date +"%Y-%m-%d %H:%M:%S:%6N (JM)"
mpiexec -stdout-proc ./%n.output.%j/%/1000r/stdout -stderr-
proc ./%n.output.%j/%/1000r/stderr date +"%Y-%m-%d %H:%M:%S:%6N"
```

Update history

Changes	Date
1st release	15 November, 2021
Corrected typos and errors	27 January, 2022
Add the result of MPI process generation time (p.35)	4 April, 2022