1.1 MixedMode

1.1.1 *Summary*

OpenMP/MPI Mixed-Mode Microbenchmarks [6] is a set of low-level microbenchmarks to test the performance of hybrid MPI - OpenMP programming model. The mixed-mode microbenchmarks provide analogues for the typical operations found in MPI microbenchmark suites, for both point-to-point and collective communication patterns.
Two main considerations are captured in the microbenchmarks:
The effects of intra-thread communication and synchronisation by measuring both the times of MPI library calls and the reading and writing of buffers.

1.1.2 *TestCase*

1) Preparation and Compilation

The code is available on PRACE SVN.
(https://prace.osd.sara.nl/svn/trunk/pracewp74a/PABS/applications/mixedmode/)
After downloading the code, provide other JuBE run xml files (same as the case of HPCC):

- Edit/add CURIE setup on compile.xml: Compilation options are the same as HPCC
- Create the running command file ‘Nbench-Intel-Nehalem-CURIE.xml‘ (replacing #NODES# to $nodes which is given on job running xml file)
- Edit execute.xml

Error and edit

- The code on PRACE SVN results in a compilation error on parallelEnvironment:
  - 'implicit none' statement is not placed in the right place, relative to MPI header inclusion. To resolve this error, #MPI_INCLUDE# line is placed after implicit none in parallelEnvironment.f90.jube.
- Job submission script under platform/Intel-Nehalem-CURIE has been corrected to include the "NODES" option. That is,
  - #MSUB -N #NODES# in the job submission script.
- MPI changes
  - MPI on CURIE by default (bullx 1.X) does not support thread_multiple communication, which results in an error on multiple pingpong and haloexchange. That is a quite important issue on mixedMode run which necessitates the communication between threads.
  - How to determine whether the system-supported MPI is the multiple-thread mode:
    In OpenMPI, ompi_info command will show
    Thread support: posix (MPI_THREAD_MULTIPLE: no, progress: no)
    Then it does not support multiple thread communication
  - How to change/enable:
    Change MPI compiler which provides multiple thread communication.
    Unfortunately Bull MPIs on Curie does not support multiple thread, MPC which is the MPI evaluation by CCC does not work properly (not available F90 compiler). It is requested on Curie support and the answer is awaiting.
  - On haloexcahnge, the code has the potential compatibility issue against OpenMPI configurations:
    On parallelEnvironment.f90, it declares
    CALL MPI_Cart_Create(comm,1,dims,PERIODS,REORDER,commCart,ierr)

with PERIODS and REORDER declared as logical values (old expression)
Now these variables changed to integers, for the compatibility with C languages. Variable types are changed accordingly. That should be the OpenMPI-related problems, but 'use mpi' results in
parallelEnvironment.f90(141): error #6285: There is no matching specific subroutine for this generic subroutine call. [MPI_CART_CREATE]
If 'include 'mpif.h' won't return an error, while during runtime it returns error due to the above problem
Use of "include 'mpif.h'" and change of periods and reorder datatypes to integer solve all problems

2) Parameters

mixedMode benchmark gets the input file for parameter determination and benchmark selection. There are two groups of parameters exist:

- Execution setup, where min/max data sizes and target times are determined.
  In the input file, the top three lines, which specify the minimum data size, maximum data size and target time for each benchmark are required (and in that order). The data size starts at the minimum data size and is doubled until the maximum data size is reached. The target time parameter is used to keep the execution time for each test approximately constant. It is used as follows: for a given data size the benchmark is run for a certain number of iterations. If the execution time is less than the target time the test is re-run with twice the number of iterations until the target time is met.
- Benchmark cases.
  Roughly 4 different groups are of common interest: barrier, collectives, haloexchange, pingpong.

4 groups of simulations are taken place:

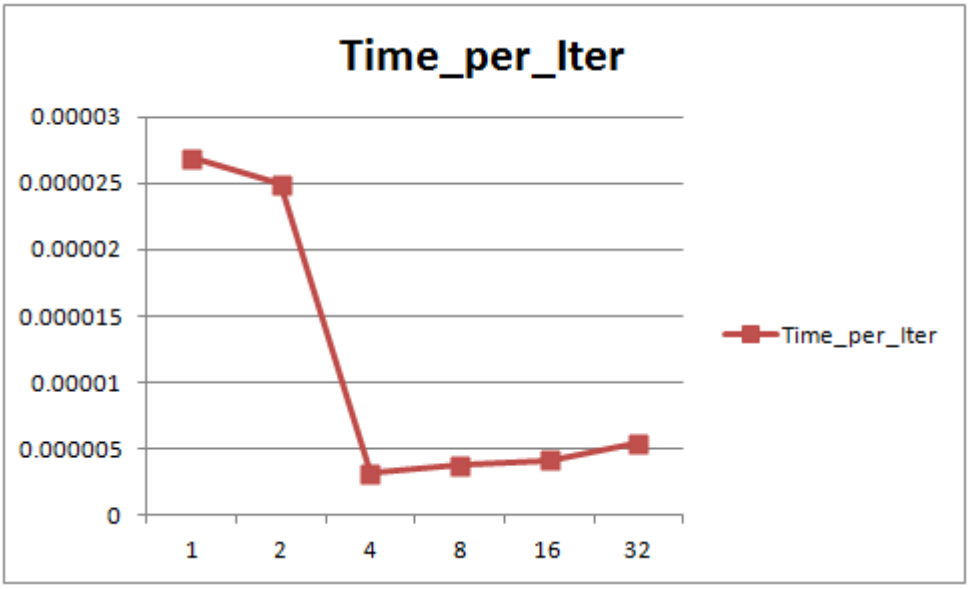
- Pingpong and pingping, max size of 4194304, between 2 nodes
- Haloexchange, max size of 4194304, among 4 nodes
- Collective, max 262144, 4 nodes
- Barrier, max 2, 1 node

1.1.3 *Results and Analysis*

a) Case 1: Barrier test

The barrier test has been performed to check the overhead on barrier within a node. Time for performing barrier has been measured from 1 to 32 cores within a single node, changing the number of threads to satisfy the full use of CPU cores in a single node. It is related with the synchronization between CPU cores, thus it should be less with smaller number of processors for pure MPI case. In this hybrid case, that tendency is observed at more than 4 MPI ranks, but time is longer with 1 or 2 MPI ranks, when threads on each rank spans more than 2 physical CPU sockets. (On CURIE, 4 physical sockets each with 8 cores constitute a node.) We presume that the overhead between threads on different sockets (by calling OMP MASTER OpenMP command) has higher overhead than MPI barrier.

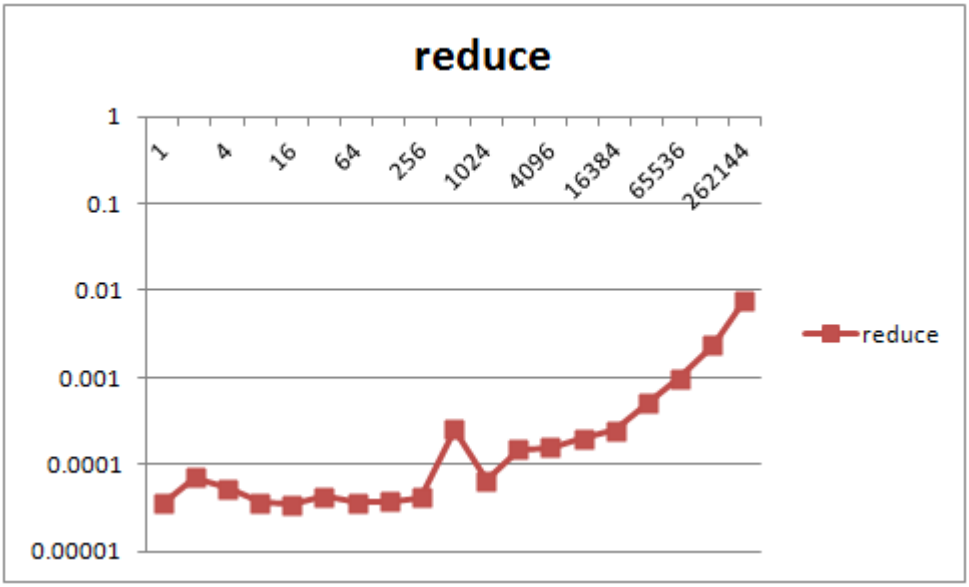
Processors	Time_per_Iter
1	2.69E-05
2	2.49E-05
4	3.15E-06
8	3.78E-06
16	4.12E-06
32	5.38E-06

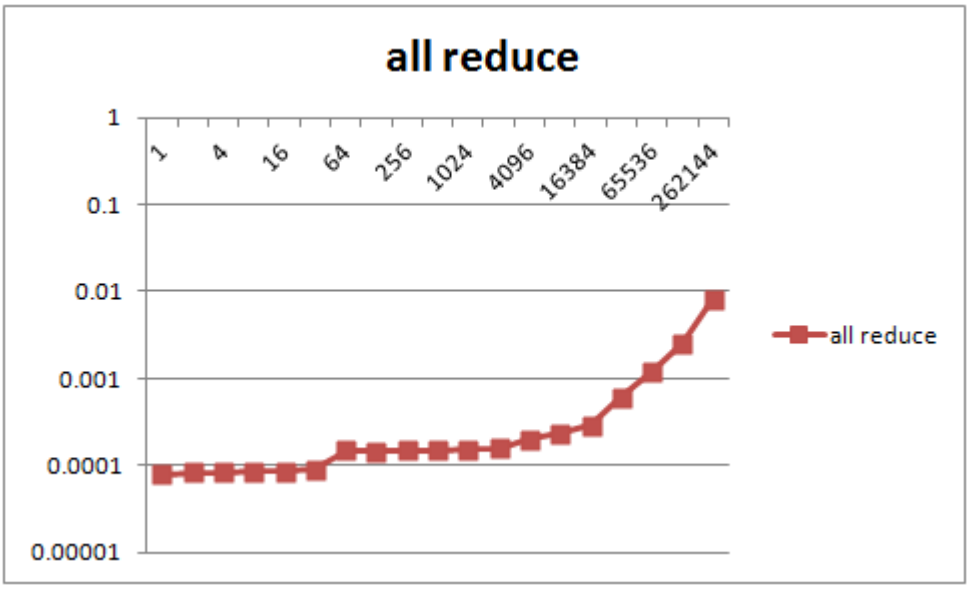


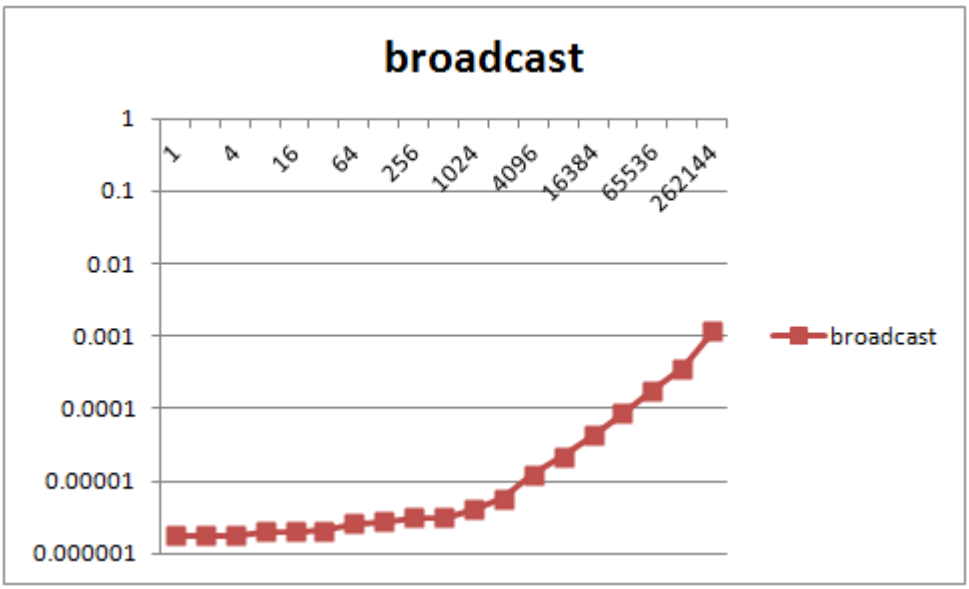
b) Case 2: Collective calls

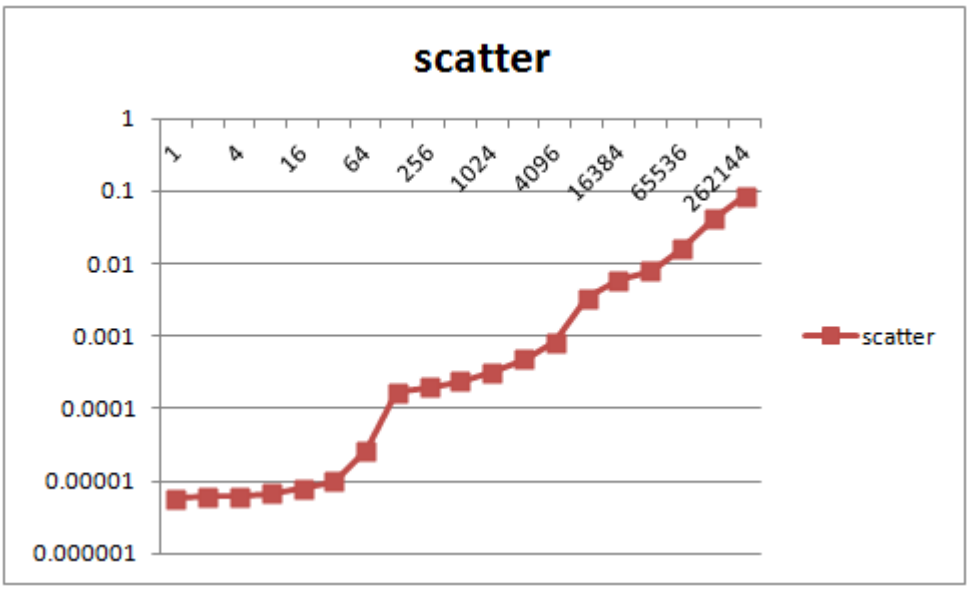
Time for 6 different collective calls (reduce, all-reduce, broadcast, scatter, gather, all-to-all) are measured on 4 nodes. Again, MPI ranks per core ranges from 1 to 32 cores, along with varying number of threads per core to satisfy the full use of CPU cores in a single node.

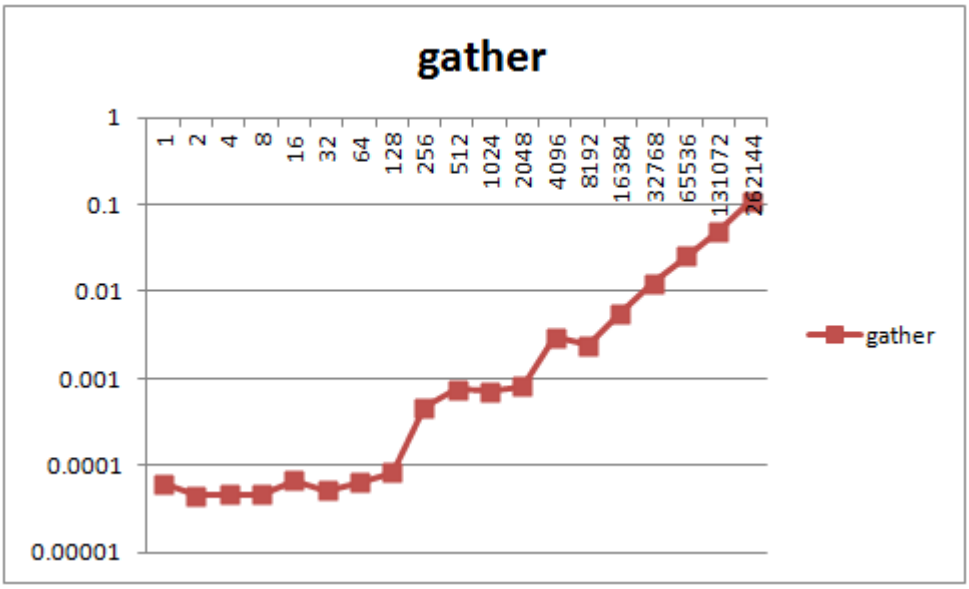
The following graphs show communication time per iteration for those 6 different tests, in case of pure MPI. Integer (4-byte) array is communicated.

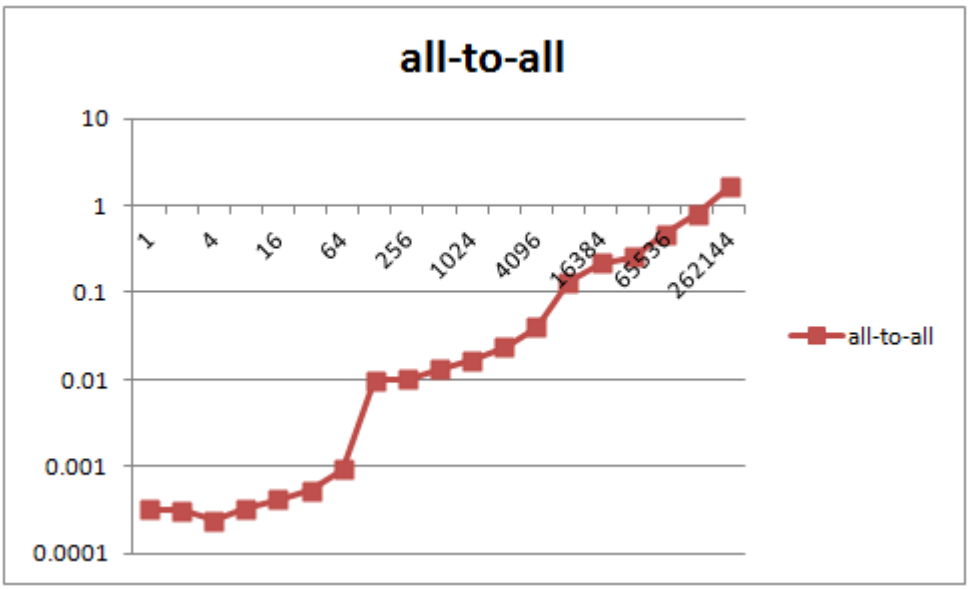



all reduce
1
0.1
0.01
0.001
0.0001
0.00001
1
4
16
64
256
1024
4096
16384
65536
262144
all reduce

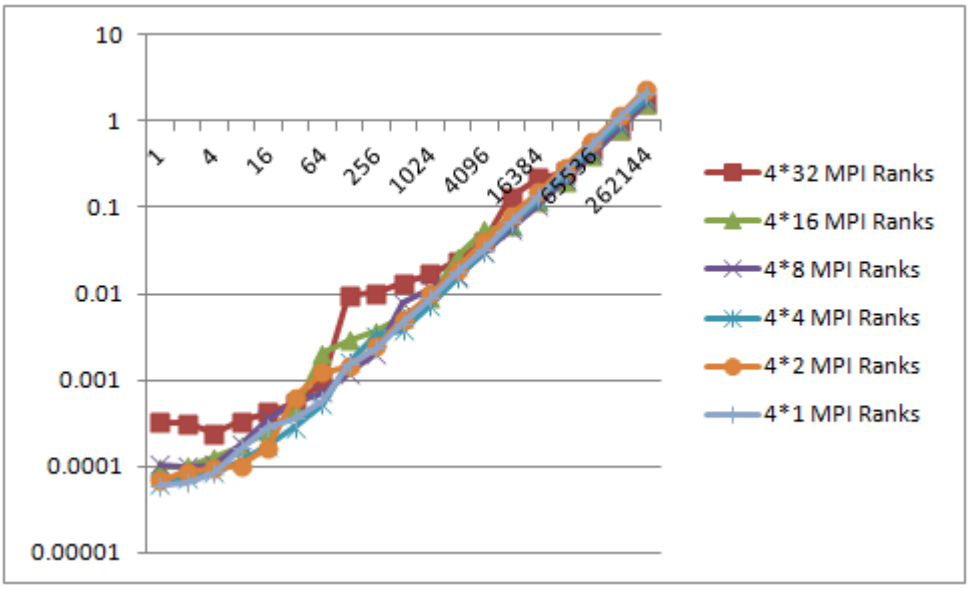

broadcast
1
0.1
0.01
0.001
0.0001
0.00001
0.000001
1
4
16
64
256
1024
4096
16384
65536
262144
broadcast


scatter
1
0.1
0.01
0.001
0.0001
0.00001
0.000001
1
4
16
64
256
1024
4096
16384
65536
262144
scatter





Now the time for all-to-all communication at different composition is depicted. Tendency is almost linear between array size and communication time. Time does not differ much between different number of threads.

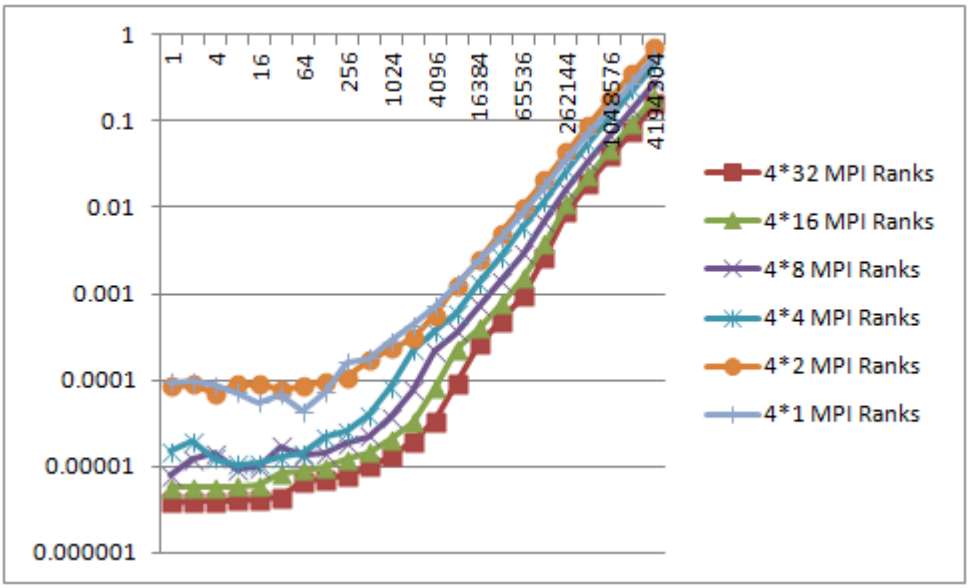


array size	4*32 MPI Ranks	4*16 MPI Ranks	4*8 MPI Ranks	4*4 MPI Ranks	4*2 MPI Ranks	4*1 MPI Ranks
1	0.000319	9.77E-05	0.000106	6.28E-05	6.8E-05	5.96E-05
2	0.000313	9.84E-05	0.0001	7.37E-05	8.71E-05	6.43E-05
4	0.000241	0.000126	0.000105	8.57E-05	9.5E-05	8.38E-05
8	0.000328	0.000169	0.000179	0.000117	0.000103	0.000161
16	0.000423	0.000261	0.000355	0.000176	0.000167	0.000281
32	0.000531	0.000442	0.000543	0.00028	0.000626	0.000359
64	0.000927	0.002063	0.00072	0.000522	0.001235	0.000582
128	0.009679	0.002876	0.001182	0.001617	0.001428	0.001532
256	0.010189	0.00371	0.002025	0.00327	0.002472	0.002331
512	0.013211	0.005295	0.007716	0.003797	0.00507	0.004573
1024	0.016664	0.008759	0.011528	0.007265	0.009912	0.008903
2048	0.023664	0.026703	0.015882	0.015254	0.019374	0.017871
4096	0.04023	0.054792	0.030595	0.031503	0.041185	0.032693
8192	0.130663	0.062821	0.056329	0.063134	0.081788	0.067023
16384	0.220157	0.117169	0.105831	0.120949	0.151497	0.134121
32768	0.258915	0.201208	0.20971	0.236647	0.291881	0.266832
65536	0.473534	0.403379	0.42111	0.480074	0.587966	0.533736
131072	0.804078	0.810602	0.843203	0.953593	1.148214	1.098953
262144	1.616524	1.61157	1.681836	1.878699	2.365967	2.131736

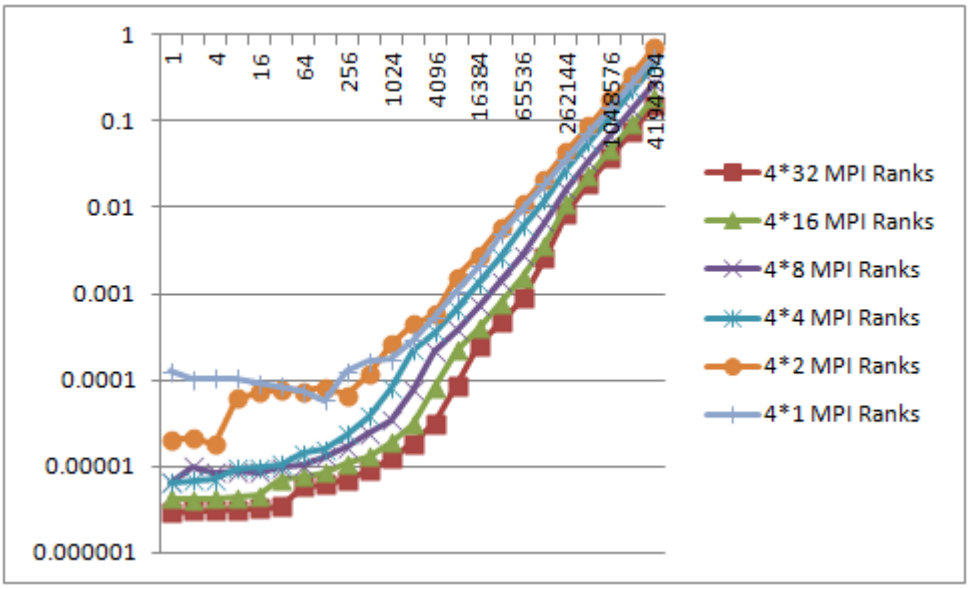
c) Case 3: Haloexchange

Haloexchange time is measured on 4 node MPI task simulation. Again, number of MPI ranks per node varies from 1 to 32, with the inverse proportion on number of threads per core so that the whole CPU cores participate in simulation. Three different MPI communications are performed on this simulation. They are serial communication (communicator located out of OpenMP thread call, so the master thread only operates the communication), funnelled communication (communicator located inside OpenMP thread call while a single thread takes care of the whole communication) and threaded communication (communicator located inside OpenMP thread call and all threads participate in communication). In principle, threaded

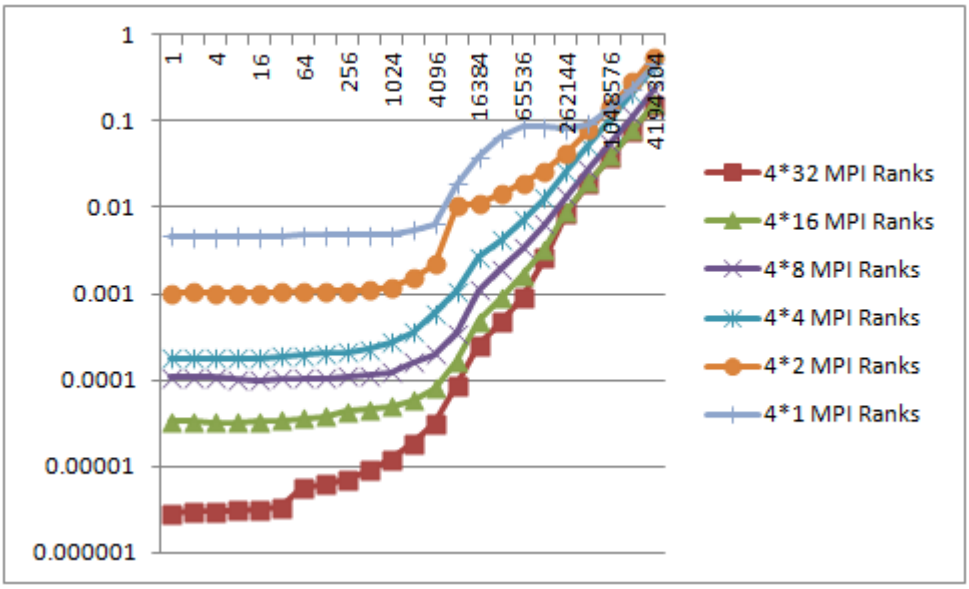
communication should run fastest. Three results from serial, funnelled and threaded communications are presented below. Like previous result, communication time takes more with 16 or 32 threads. It again concludes that the data copy between different sockets take more time.



array size	4*32 MPI Ranks	4*16 MPI Ranks	4*8 MPI Ranks	4*4 MPI Ranks	4*2 MPI Ranks	4*1 MPI Ranks
1	3.75E-06	5.64E-06	7.78E-06	1.47E-05	8.38E-05	9.43E-05
2	3.91E-06	5.66E-06	1.2E-05	1.93E-05	8.91E-05	9.52E-05
4	3.94E-06	5.71E-06	1.41E-05	1.24E-05	6.95E-05	8.58E-05
8	4.03E-06	5.8E-06	9.27E-06	1.05E-05	8.87E-05	7.16E-05
16	4.07E-06	5.99E-06	9.64E-06	1.08E-05	8.97E-05	5.49E-05
32	4.29E-06	8.3E-06	1.71E-05	1.28E-05	7.84E-05	6.9E-05
64	6.68E-06	8.86E-06	1.33E-05	1.41E-05	8.35E-05	4.36E-05
128	7.02E-06	9.74E-06	1.41E-05	2.17E-05	9.71E-05	7.16E-05
256	7.84E-06	1.2E-05	1.84E-05	2.54E-05	0.000109	0.000159
512	1E-05	1.44E-05	2.21E-05	3.84E-05	0.00017	0.00018
1024	1.31E-05	2.02E-05	3.7E-05	8.13E-05	0.000235	0.000284
2048	1.93E-05	3.2E-05	7.49E-05	0.000223	0.000301	0.000431
4096	3.29E-05	7.93E-05	0.000215	0.000371	0.000556	0.000717
8192	9.2E-05	0.00023	0.000362	0.000601	0.001256	0.001296
16384	0.000263	0.000402	0.000713	0.001335	0.002448	0.002471
32768	0.00048	0.000762	0.0014	0.002624	0.005046	0.004464
65536	0.000923	0.001503	0.00284	0.005967	0.01019	0.008871
131072	0.002657	0.003738	0.006856	0.011969	0.020675	0.017627
262144	0.008812	0.010886	0.016405	0.026925	0.043942	0.036594
524288	0.01938	0.023279	0.034418	0.056159	0.089816	0.072382
1048576	0.038983	0.047307	0.06924	0.114805	0.180618	0.141657
2097152	0.077839	0.094977	0.139592	0.230683	0.363333	0.280667
4194304	0.156932	0.192619	0.283281	0.462807	0.715509	0.554379

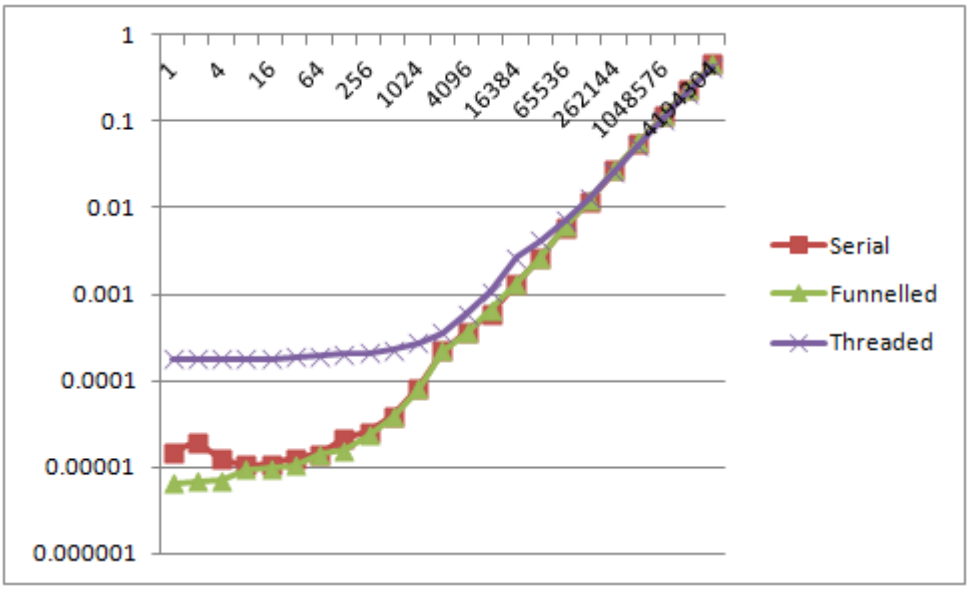


array size	4*32 MPI Ranks	4*16 MPI Ranks	4*8 MPI Ranks	4*4 MPI Ranks	4*2 MPI Ranks	4*1 MPI Ranks
1	2.9E-06	4.2E-06	6.55E-06	6.47E-06	2E-05	0.000125
2	3.06E-06	4.16E-06	9.92E-06	6.89E-06	2.14E-05	0.0001
4	3.05E-06	4.2E-06	8.43E-06	7.11E-06	1.77E-05	0.000106
8	3.13E-06	4.32E-06	8.71E-06	9.52E-06	6.1E-05	0.000105
16	3.21E-06	4.62E-06	8.53E-06	9.74E-06	7.23E-05	9.19E-05
32	3.44E-06	7.03E-06	9.75E-06	1.05E-05	7.78E-05	8.39E-05
64	5.86E-06	7.54E-06	1.01E-05	1.4E-05	7.37E-05	7.45E-05
128	6.21E-06	8.41E-06	1.3E-05	1.57E-05	8.15E-05	5.84E-05
256	7.01E-06	1.06E-05	1.67E-05	2.35E-05	6.71E-05	0.000125
512	8.94E-06	1.3E-05	2.41E-05	3.81E-05	0.000117	0.000164
1024	1.23E-05	1.89E-05	3.4E-05	7.98E-05	0.000259	0.000173
2048	1.84E-05	3.04E-05	7.6E-05	0.000221	0.00044	0.000291
4096	3.14E-05	8.28E-05	0.000213	0.00036	0.000579	0.000539
8192	8.57E-05	0.000223	0.000381	0.000668	0.001534	0.001056
16384	0.000255	0.0004	0.000723	0.001342	0.002827	0.002093
32768	0.000471	0.000781	0.001388	0.002641	0.005874	0.004891
65536	0.000897	0.001546	0.002762	0.006044	0.01101	0.009489
131072	0.002582	0.003662	0.006825	0.012185	0.021184	0.018896
262144	0.008677	0.010908	0.016557	0.027662	0.04519	0.036654
524288	0.019263	0.023159	0.034135	0.056911	0.087722	0.072488
1048576	0.038537	0.04723	0.069592	0.114997	0.175234	0.141718
2097152	0.077274	0.093639	0.13953	0.23087	0.348118	0.279573
4194304	0.153982	0.189669	0.28076	0.462419	0.732286	0.554078



array size	4*32 MPI Ranks	4*16 MPI Ranks	4*8 MPI Ranks	4*4 MPI Ranks	4*2 MPI Ranks	4*1 MPI Ranks
1	2.82E-06	3.28E-05	0.000107	0.00018	0.001023	0.004605
2	2.95E-06	3.33E-05	0.000107	0.00018	0.001037	0.004534
4	2.99E-06	3.26E-05	0.000108	0.000177	0.001025	0.004563
8	3.06E-06	3.26E-05	0.000102	0.000178	0.001024	0.004619
16	3.15E-06	3.33E-05	9.95E-05	0.000179	0.001025	0.004548
32	3.35E-06	3.43E-05	0.000104	0.000187	0.001051	0.004692
64	5.71E-06	3.55E-05	0.000104	0.000194	0.001066	0.004767
128	6.07E-06	3.8E-05	0.000106	0.000202	0.001066	0.004734
256	6.95E-06	4.3E-05	0.000111	0.000211	0.00106	0.004793
512	8.9E-06	4.6E-05	0.000113	0.000226	0.001095	0.00475
1024	1.19E-05	4.96E-05	0.000124	0.000276	0.001203	0.004736
2048	1.83E-05	5.74E-05	0.00016	0.000361	0.001504	0.005405
4096	3.13E-05	8.02E-05	0.0002	0.000597	0.002195	0.006434
8192	8.72E-05	0.000166	0.000344	0.001057	0.010503	0.018915
16384	0.000255	0.000481	0.001108	0.002661	0.011119	0.03836
32768	0.000469	0.000894	0.001923	0.004196	0.014108	0.065974
65536	0.000902	0.001651	0.003415	0.007081	0.01909	0.084656
131072	0.002577	0.003313	0.006382	0.013013	0.026711	0.085877
262144	0.008682	0.009175	0.013432	0.026381	0.043243	0.081627
524288	0.019047	0.019808	0.028144	0.053518	0.078469	0.094093
1048576	0.038645	0.039887	0.056472	0.104868	0.144898	0.137201
2097152	0.077327	0.079919	0.116403	0.210336	0.284097	0.236741
4194304	0.155631	0.163249	0.236441	0.426464	0.553894	0.461462

We compare three cases of serial, funnelled and threaded communication times. The following graph shows three different communication modes at time of 8 threads per rank. On threaded mode, the initial overhead (in opening the communication channel) is larger than other ones. On the other hands, with large communication size, threaded mode provided the better communication speed.

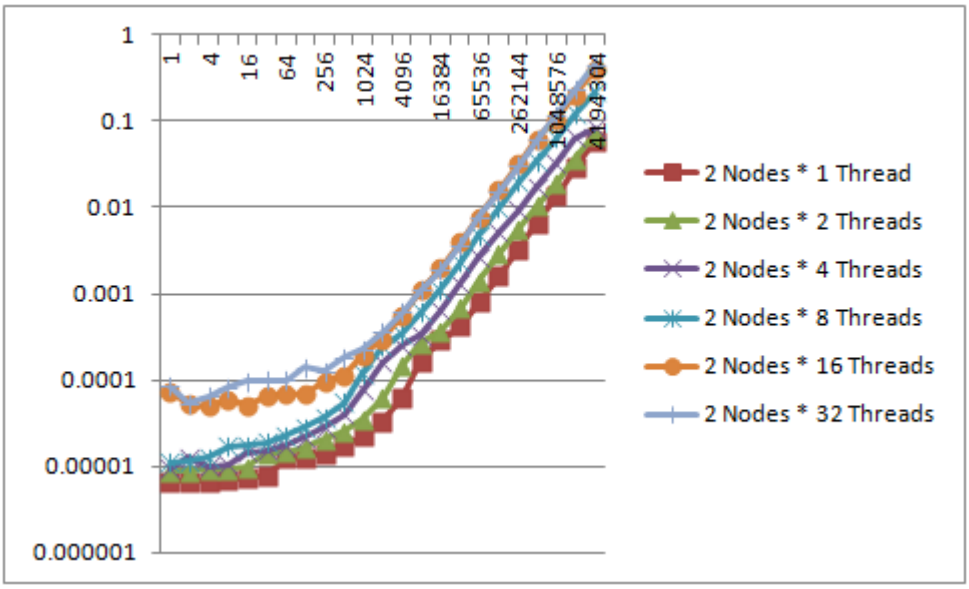


array size	Serial	Funnelled	Threaded
1	1.47E-05	6.47E-06	0.00018
2	1.93E-05	6.89E-06	0.00018
4	1.24E-05	7.11E-06	0.000177
8	1.05E-05	9.52E-06	0.000178
16	1.08E-05	9.74E-06	0.000179
32	1.28E-05	1.05E-05	0.000187
64	1.41E-05	1.4E-05	0.000194
128	2.17E-05	1.57E-05	0.000202
256	2.54E-05	2.35E-05	0.000211
512	3.84E-05	3.81E-05	0.000226
1024	8.13E-05	7.98E-05	0.000276
2048	0.000223	0.000221	0.000361
4096	0.000371	0.00036	0.000597
8192	0.000601	0.000668	0.001057
16384	0.001335	0.001342	0.002661
32768	0.002624	0.002641	0.004196
65536	0.005967	0.006044	0.007081
131072	0.011969	0.012185	0.013013
262144	0.026925	0.027662	0.026381
524288	0.056159	0.056911	0.053518
1048576	0.114805	0.114997	0.104868
2097152	0.230683	0.23087	0.210336
4194304	0.462807	0.462419	0.426464

d) Case 4: Ping-pong / Ping-ping

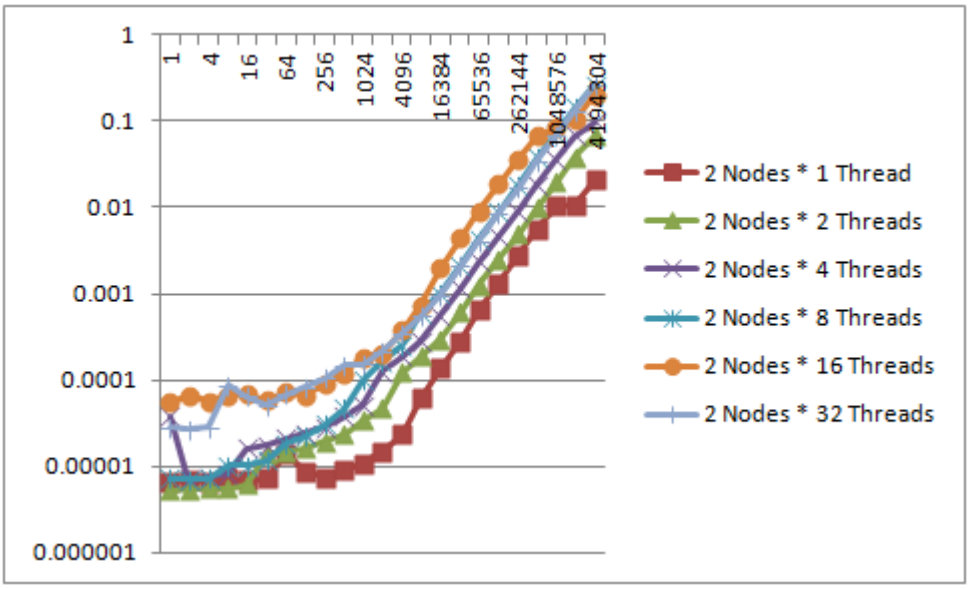
Ping-pong / Ping-ping communication is performed between 2 cores in different node. We vary the number of threads from 1 to 32 and perform three different MPI communication modes at various array sizes.

- Pingpong, serial



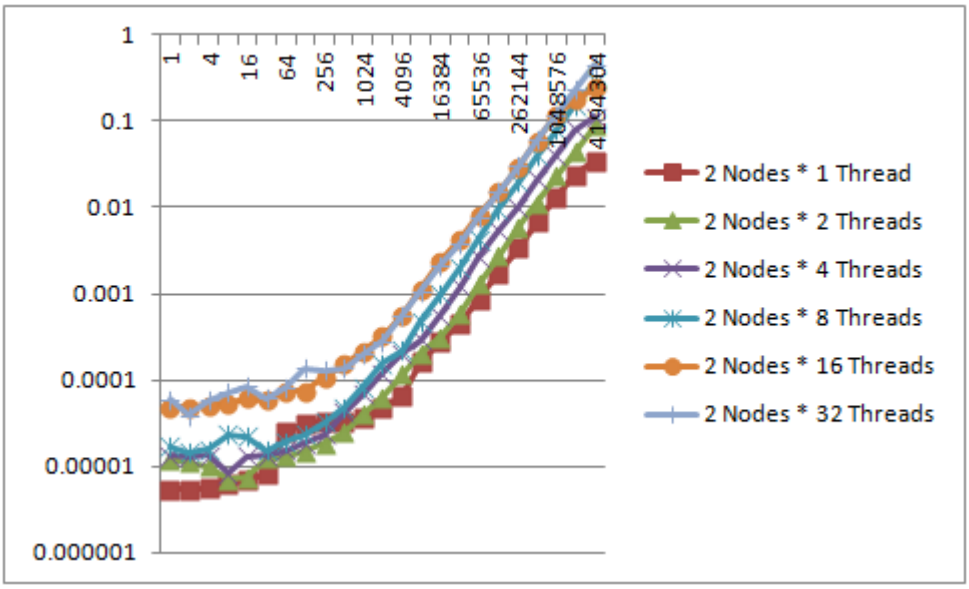
array size	2 Nodes * 1 Thread	2 Nodes * 2 Threads	2 Nodes * 4 Threads	2 Nodes * 8 Threads	2 Nodes * 16 Threads	2 Nodes * 32 Threads
1	6.58E-06	8.69E-06	9.39E-06	1.1E-05	7.36E-05	8.33E-05
2	6.62E-06	8.75E-06	1.27E-05	1.14E-05	5.17E-05	5.41E-05
4	6.69E-06	9.05E-06	9.83E-06	1.27E-05	4.9E-05	6.41E-05
8	7.05E-06	9.18E-06	1.03E-05	1.68E-05	5.97E-05	8.28E-05
16	7.27E-06	9.62E-06	1.46E-05	1.73E-05	5.04E-05	9.76E-05
32	7.8E-06	1.41E-05	1.53E-05	1.88E-05	6.43E-05	9.99E-05
64	1.22E-05	1.46E-05	1.73E-05	2.27E-05	6.94E-05	9.82E-05
128	1.27E-05	1.65E-05	2.14E-05	2.79E-05	6.75E-05	0.00014
256	1.42E-05	2.02E-05	2.84E-05	3.65E-05	9.62E-05	0.000126
512	1.73E-05	2.56E-05	3.97E-05	5.43E-05	0.000113	0.000185
1024	2.2E-05	3.53E-05	7.66E-05	0.000118	0.000192	0.000227
2048	3.27E-05	6.2E-05	0.00016	0.000245	0.000296	0.00036
4096	6.25E-05	0.000148	0.000252	0.000346	0.000557	0.000603
8192	0.000163	0.000269	0.000328	0.000604	0.001095	0.001113
16384	0.000291	0.000367	0.000652	0.00114	0.001997	0.001858
32768	0.00042	0.000684	0.001321	0.002337	0.003977	0.003664
65536	0.000795	0.001408	0.002652	0.004835	0.007717	0.007648
131072	0.001595	0.002858	0.005237	0.009735	0.015752	0.015211
262144	0.003262	0.005681	0.00939	0.019717	0.03185	0.030301
524288	0.006549	0.010508	0.017228	0.034441	0.061649	0.061054
1048576	0.013527	0.019206	0.032577	0.064062	0.106281	0.118814
2097152	0.028976	0.036075	0.063705	0.120918	0.202844	0.23094
4194304	0.058235	0.069881	0.082702	0.220111	0.396916	0.45484

- Pingping, serial



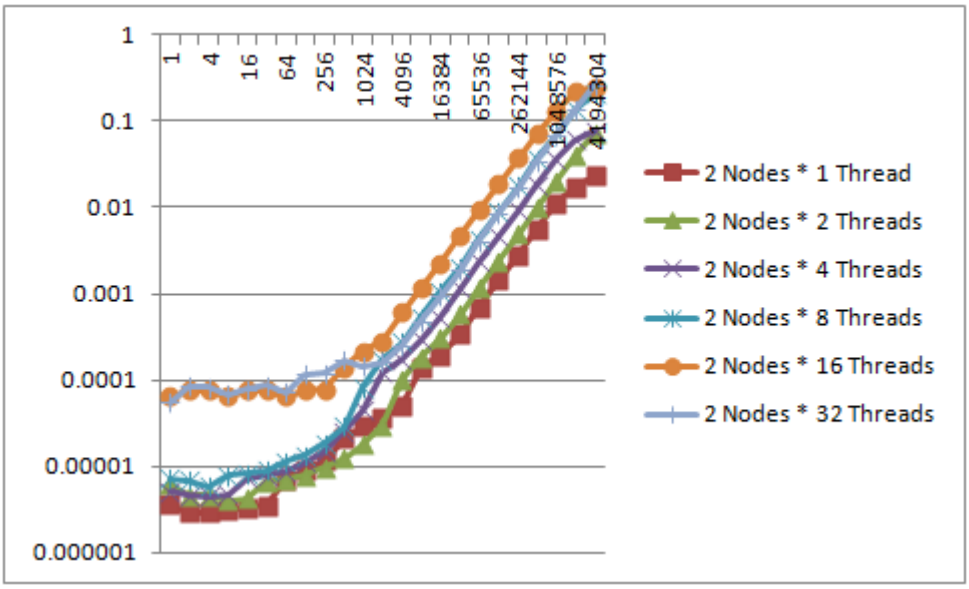
array size	2 Nodes * 1 Thread	2 Nodes * 2 Threads	2 Nodes * 4 Threads	2 Nodes * 8 Threads	2 Nodes * 16 Threads	2 Nodes * 32 Threads
1	6.63E-06	5.28E-06	3.79E-05	7.2E-06	5.43E-05	2.84E-05
2	6.83E-06	5.37E-06	6.8E-06	7.02E-06	6.58E-05	2.69E-05
4	6.73E-06	5.45E-06	7.49E-06	7.16E-06	5.58E-05	2.84E-05
8	7.13E-06	5.59E-06	8.14E-06	1.03E-05	6.66E-05	8.66E-05
16	6.82E-06	6.16E-06	1.62E-05	1.01E-05	6.78E-05	6.54E-05
32	7.12E-06	1.37E-05	1.74E-05	1.18E-05	5.81E-05	5.07E-05
64	1.42E-05	1.47E-05	2.09E-05	1.83E-05	7.19E-05	6.77E-05
128	8.63E-06	1.59E-05	2.37E-05	2.23E-05	6.61E-05	8.27E-05
256	7.46E-06	1.96E-05	2.83E-05	3.04E-05	8.82E-05	0.000104
512	8.87E-06	2.31E-05	3.78E-05	4.67E-05	0.00012	0.000149
1024	1.08E-05	3.36E-05	5.08E-05	9.8E-05	0.000184	0.000151
2048	1.5E-05	4.74E-05	0.000126	0.000163	0.000204	0.000214
4096	2.35E-05	0.000123	0.000187	0.000255	0.000379	0.000352
8192	6.06E-05	0.000188	0.000284	0.000596	0.000735	0.000555
16384	0.000137	0.000299	0.000564	0.001022	0.001987	0.000949
32768	0.000279	0.000604	0.001169	0.002124	0.004437	0.002096
65536	0.000649	0.001248	0.002348	0.004342	0.009138	0.004086
131072	0.001326	0.002495	0.004628	0.008747	0.018587	0.008685
262144	0.002765	0.004903	0.009154	0.017958	0.036344	0.016716
524288	0.005541	0.009825	0.018393	0.037253	0.06739	0.034751
1048576	0.010503	0.019739	0.036553	0.072713	0.083098	0.068424
2097152	0.01072	0.038621	0.067847	0.141933	0.104515	0.133188
4194304	0.020864	0.067412	0.099042	0.262037	0.203096	0.261065

- Pingpong, funnelled



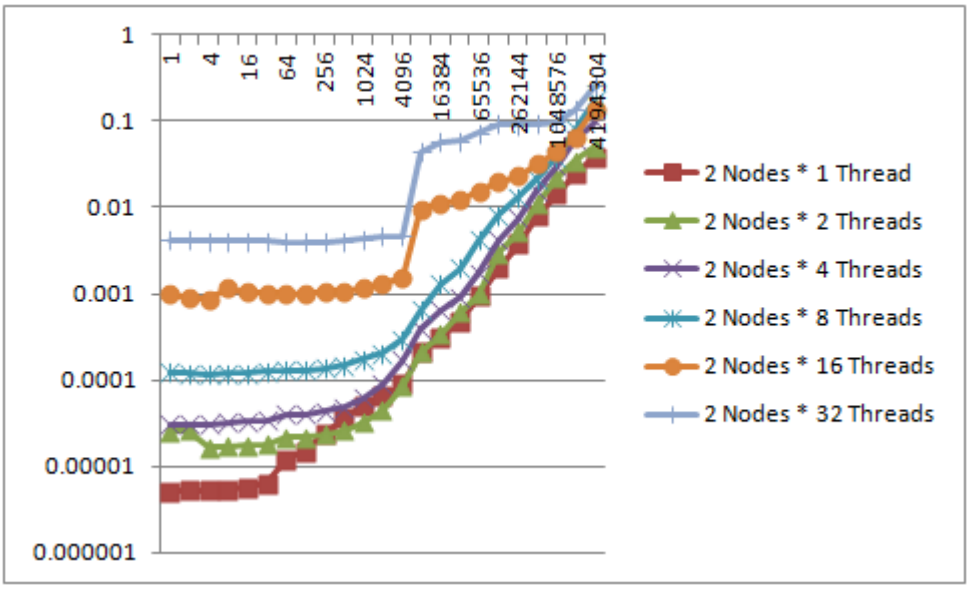
array size	2 Nodes * 1 Thread	2 Nodes * 2 Threads	2 Nodes * 4 Threads	2 Nodes * 8 Threads	2 Nodes * 16 Threads	2 Nodes * 32 Threads
1	5.38E-06	1.18E-05	1.33E-05	1.71E-05	4.81E-05	5.94E-05
2	5.42E-06	1.12E-05	1.28E-05	1.45E-05	4.8E-05	3.92E-05
4	5.55E-06	9.99E-06	1.33E-05	1.55E-05	5.07E-05	5.76E-05
8	6.06E-06	7.01E-06	8.23E-06	2.34E-05	5.36E-05	7.19E-05
16	7.03E-06	7.5E-06	1.3E-05	2.25E-05	6.14E-05	8.34E-05
32	8.07E-06	1.25E-05	1.35E-05	1.53E-05	5.76E-05	6.12E-05
64	2.5E-05	1.31E-05	1.52E-05	1.96E-05	7.45E-05	8.18E-05
128	3.17E-05	1.49E-05	1.84E-05	2.31E-05	7.39E-05	0.000136
256	0.000033	1.84E-05	2.34E-05	3.18E-05	0.000108	0.000129
512	3.34E-05	2.46E-05	4.19E-05	4.84E-05	0.000152	0.000138
1024	3.73E-05	4.02E-05	6.84E-05	8.21E-05	0.000214	0.000197
2048	4.68E-05	6.32E-05	0.000119	0.000155	0.000325	0.00028
4096	6.71E-05	0.000115	0.000203	0.000222	0.000558	0.000565
8192	0.00016	0.000202	0.000285	0.00048	0.001125	0.001077
16384	0.000283	0.000312	0.000557	0.000983	0.002384	0.002128
32768	0.000449	0.000591	0.001237	0.002033	0.004346	0.003909
65536	0.000859	0.001286	0.002641	0.004344	0.008219	0.007724
131072	0.001727	0.002758	0.005425	0.009543	0.015259	0.015172
262144	0.003383	0.005865	0.010232	0.019764	0.029302	0.030254
524288	0.006695	0.011264	0.021045	0.03979	0.05849	0.061054
1048576	0.013147	0.022985	0.040531	0.078947	0.119018	0.118976
2097152	0.02393	0.043992	0.078916	0.155841	0.174679	0.230925
4194304	0.034706	0.087668	0.112225	0.287587	0.244132	0.454999

- Pingping, funnelled



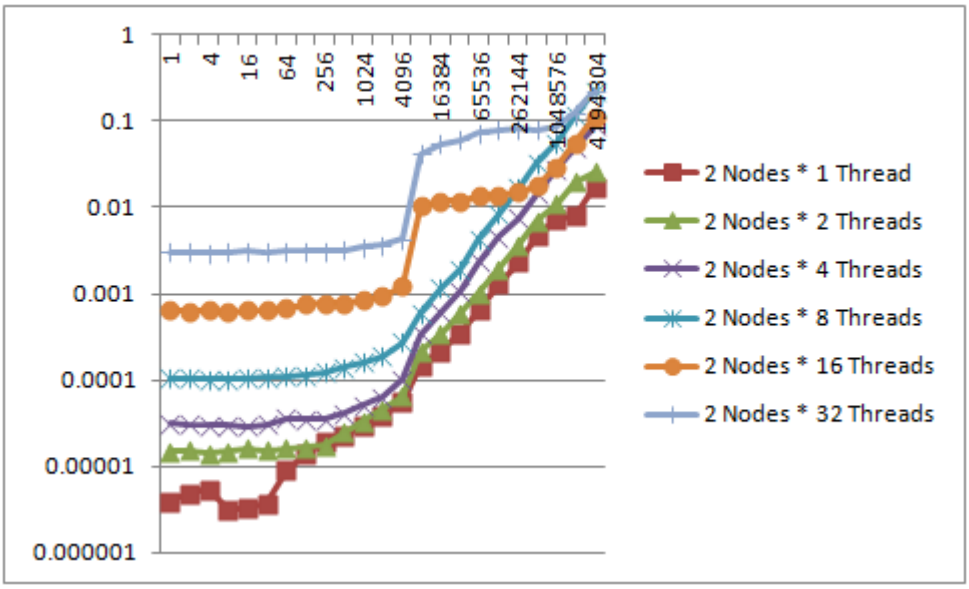
array size	2 Nodes * 1 Thread	2 Nodes * 2 Threads	2 Nodes * 4 Threads	2 Nodes * 8 Threads	2 Nodes * 16 Threads	2 Nodes * 32 Threads
1	3.59E-06	6.16E-06	5.13E-06	7.25E-06	6.51E-05	5.51E-05
2	2.94E-06	4.56E-06	4.56E-06	6.81E-06	7.64E-05	8.34E-05
4	2.94E-06	4.58E-06	4.46E-06	5.81E-06	7.61E-05	8.28E-05
8	3.08E-06	3.98E-06	4.68E-06	7.76E-06	6.46E-05	6.8E-05
16	3.28E-06	4.17E-06	7.14E-06	8.28E-06	7.65E-05	7.77E-05
32	3.55E-06	6.69E-06	7.73E-06	8.89E-06	7.61E-05	8.64E-05
64	7.03E-06	6.98E-06	8.62E-06	1.13E-05	6.47E-05	7.14E-05
128	8.93E-06	7.8E-06	1.09E-05	1.34E-05	7.63E-05	0.000117
256	1.18E-05	9.65E-06	1.47E-05	1.86E-05	7.86E-05	0.00012
512	2.12E-05	1.23E-05	2.5E-05	2.93E-05	0.000142	0.000168
1024	2.9E-05	1.78E-05	4.48E-05	8.14E-05	0.000215	0.000145
2048	3.57E-05	3E-05	0.000124	0.000176	0.000277	0.000161
4096	5.13E-05	0.000103	0.000174	0.000275	0.000616	0.000255
8192	0.000136	0.000182	0.000287	0.000535	0.001146	0.000481
16384	0.000193	0.000308	0.000539	0.001061	0.00224	0.00093
32768	0.000339	0.000583	0.001144	0.002024	0.004836	0.001861
65536	0.000692	0.001151	0.002324	0.004339	0.00934	0.004079
131072	0.001416	0.002408	0.004615	0.008814	0.018626	0.008712
262144	0.002789	0.004844	0.009259	0.017793	0.037617	0.016727
524288	0.005578	0.00982	0.018371	0.036736	0.070807	0.034755
1048576	0.011103	0.020012	0.036076	0.070346	0.1298	0.068375
2097152	0.017293	0.039298	0.060889	0.134039	0.220619	0.133264
4194304	0.022992	0.072309	0.077414	0.213709	0.249917	0.261671

- Pingpong, threaded



array size	2 Nodes * 1 Thread	2 Nodes * 2 Threads	2 Nodes * 4 Threads	2 Nodes * 8 Threads	2 Nodes * 16 Threads	2 Nodes * 32 Threads
1	5.15E-06	2.52E-05	3.03E-05	0.000124	0.00102	0.004125
2	5.17E-06	2.59E-05	3.02E-05	0.000119	0.000883	0.004162
4	5.19E-06	1.63E-05	3.05E-05	0.000116	0.000868	0.004044
8	5.44E-06	1.71E-05	3.26E-05	0.000121	0.001146	0.004057
16	5.61E-06	1.74E-05	3.28E-05	0.000121	0.00104	0.004073
32	6.23E-06	1.77E-05	3.43E-05	0.000125	0.001016	0.004062
64	1.21E-05	2.14E-05	3.94E-05	0.000129	0.00102	0.003958
128	1.48E-05	2.17E-05	4.02E-05	0.000131	0.000974	0.003998
256	2.42E-05	2.34E-05	4.31E-05	0.000137	0.00107	0.003999
512	3.59E-05	2.63E-05	4.81E-05	0.000149	0.001062	0.004025
1024	5.05E-05	3.2E-05	6.06E-05	0.000175	0.001145	0.004285
2048	6.47E-05	4.45E-05	8.81E-05	0.000203	0.001325	0.004592
4096	8.82E-05	8.36E-05	0.000167	0.000297	0.001526	0.004641
8192	0.000212	0.000207	0.000387	0.00065	0.009493	0.044314
16384	0.000311	0.000339	0.00064	0.00128	0.011327	0.057298
32768	0.000483	0.000603	0.000901	0.00197	0.012133	0.059366
65536	0.000928	0.001	0.001711	0.004135	0.015473	0.072885
131072	0.001959	0.00294	0.00414	0.008259	0.019677	0.092451
262144	0.003804	0.005235	0.007506	0.013341	0.023409	0.089889
524288	0.007995	0.010887	0.015589	0.021522	0.031981	0.093493
1048576	0.014191	0.022001	0.030116	0.038147	0.043626	0.09721
2097152	0.024145	0.034305	0.060229	0.079891	0.065909	0.136609
4194304	0.038228	0.050861	0.099186	0.174233	0.140235	0.259729

- Pingping, threaded



array size	2 Nodes * 1 Thread	2 Nodes * 2 Threads	2 Nodes * 4 Threads	2 Nodes * 8 Threads	2 Nodes * 16 Threads	2 Nodes * 32 Threads
1	3.92E-06	1.5E-05	3.11E-05	0.000105	0.000641	0.003001
2	4.79E-06	1.53E-05	3.02E-05	0.000105	0.000631	0.003031
4	5.39E-06	1.42E-05	2.95E-05	0.000103	0.000644	0.002962
8	3.16E-06	1.49E-05	2.98E-05	0.000102	0.000615	0.003031
16	3.26E-06	1.61E-05	2.92E-05	0.000103	0.000648	0.003078
32	3.67E-06	1.54E-05	3.08E-05	0.000108	0.000647	0.003055
64	9E-06	1.6E-05	3.61E-05	0.000111	0.000684	0.003086
128	1.38E-05	1.64E-05	3.48E-05	0.000115	0.000754	0.003074
256	1.96E-05	1.74E-05	3.51E-05	0.000121	0.000772	0.003151
512	2.24E-05	2.47E-05	4.06E-05	0.000142	0.000769	0.003224
1024	2.97E-05	3.32E-05	5.01E-05	0.00016	0.00084	0.003503
2048	3.86E-05	4.61E-05	6.29E-05	0.000189	0.000933	0.003679
4096	5.51E-05	6.5E-05	0.000102	0.000276	0.001213	0.004322
8192	0.000147	0.000215	0.000336	0.00059	0.010493	0.041254
16384	0.000212	0.000335	0.000604	0.001155	0.011914	0.054718
32768	0.000348	0.00059	0.001061	0.001938	0.011731	0.060605
65536	0.00065	0.001026	0.0023	0.004328	0.01368	0.072114
131072	0.001287	0.001943	0.004532	0.008476	0.013875	0.078689
262144	0.002333	0.0037	0.007535	0.016738	0.014989	0.080424
524288	0.004663	0.006882	0.014427	0.032714	0.017764	0.080175
1048576	0.007387	0.01085	0.028108	0.056056	0.02913	0.088296
2097152	0.008035	0.019932	0.048439	0.113705	0.055535	0.124229
4194304	0.017408	0.025423	0.088375	0.224803	0.110942	0.22147

References

[1] HPCC: http://icl.cs.utk.edu/hpcc/
[2] Report on available performance analysis and benchmark tools, PRACE Preparatory Phase Deliverable D6.3.1, November 2008.
[3] Technical Assessment Report of Prototype Systems, PRACE Preparatory Phase Deliverable D5.2, December 2009
[4] JuBE framework, http://www2.fz-juelich.de/jsc/jube/
[5] P-SNAP homepage, http://www.ccs3.lanl.gov/pal/software/psnap/
[6] mixedMode, http://www2.epcc.ed.ac.uk/~markb/mpiopenmpbench/intro.html