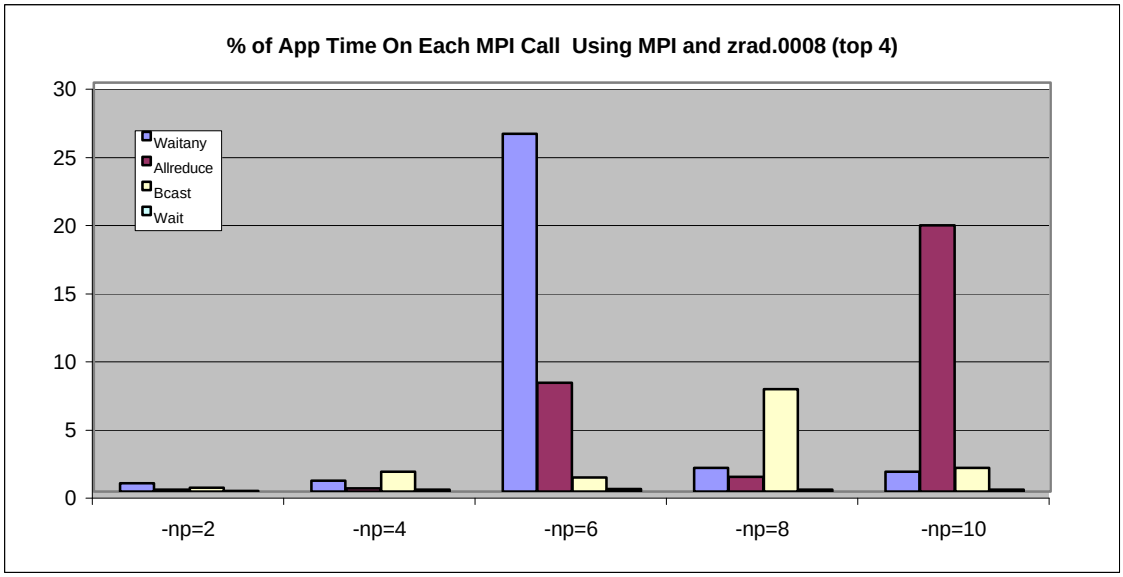
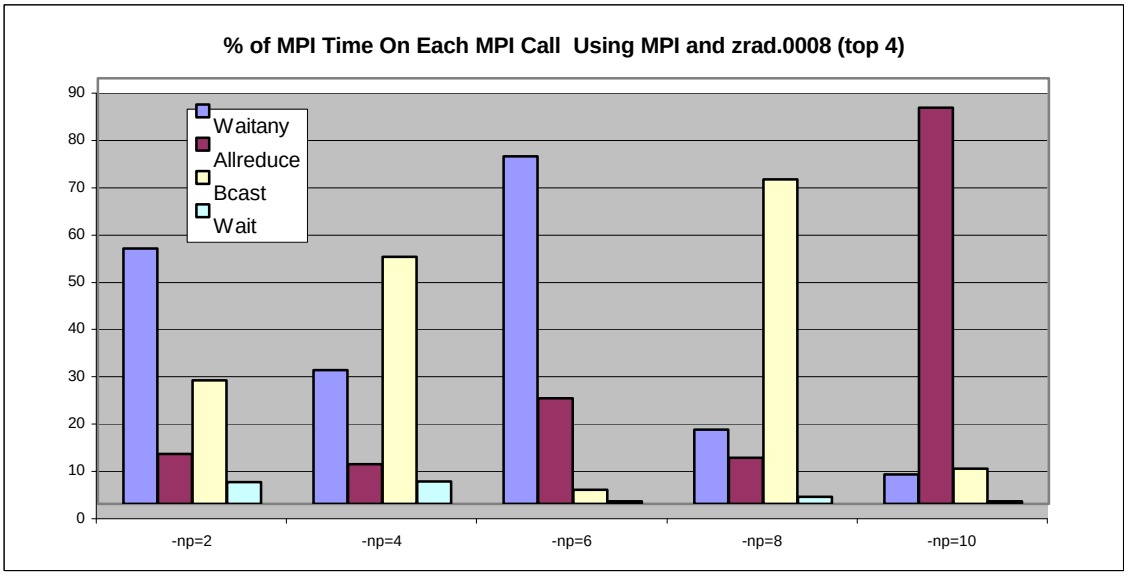
MPI Profile (mpiP) on IRS BenchMark Application

MPI with ZRAD0008

	-np=2		-np=4		-np=6		-np=8		-np=10	
Call	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**
Waitany	0.62	53.94	0.79	28.22	26.23	73.5	1.72	15.67	1.46	6.28
Allreduce	0.12	10.51	0.24	8.44	7.98	22.35	1.08	9.81	19.51	83.72
Bcast	0.3	26.06	1.47	52.25	1.05	2.95	7.52	68.55	1.74	7.44
Wait	0.05	4.55	0.13	4.8	0.18	0.5	0.16	1.46	0.13	0.57
Isend	0.03	2.71	0.1	3.69	0.15	0.43	0.41	3.74	0.01	0.05
Waitall	0.02	1.49	0.06	2.03	0.04	0.12	0.02	0.2	0.02	0.09
Irecv	0.01	0.73	0.02	0.56	0.02	0.05	0.06	0.57	0.05	0.23





* "Waitany","AllReduce" and "Bcast" takes almost 30% of application time in the case unevenly distribute blocks among nodes. The case of "-np=10" almost the same "-np=8", it is due to number of block is smaller then number of node.

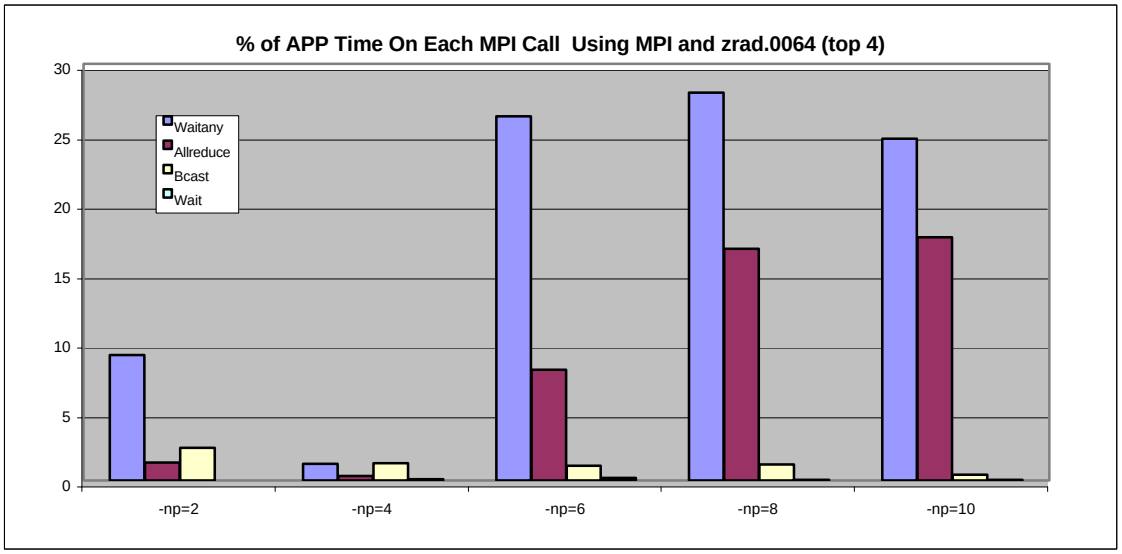
zrad.0008 * This is an 8 block problem, suitable for testing with at least 2 MPI Processes and up to 8 CPU's. The following types of runs should work well

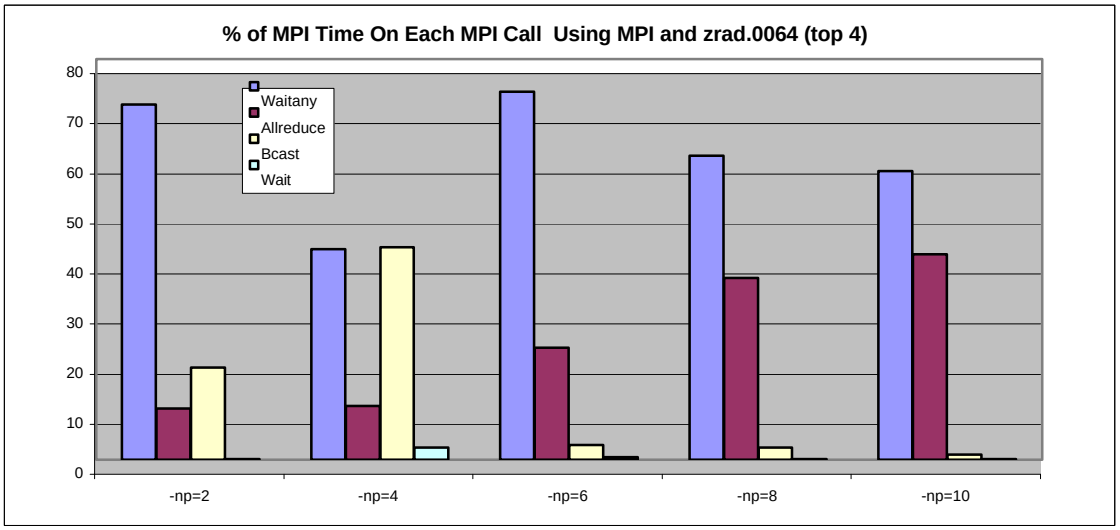
8 MPI Processes

4 MPI Processes at 2 Threads each

MPI with ZRAD0064

	-np=2		-np=4		-np=6		-np=8		-np=10	
Call	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**
Waitany	9.05	70.9	1.22	42.1	26.23	73.5	27.93	60.74	24.6	57.61
Allreduce	1.31	10.26	0.31	10.75	7.98	22.35	16.7	36.33	17.52	41.04
Bcast	2.36	18.45	1.23	42.42	1.05	2.95	1.13	2.45	0.43	1
Wait	0.02	0.18	0.07	2.38	0.18	0.5	0.04	0.09	0.05	0.12
Isend	0.02	0.15	0.02	0.69	0.15	0.43	0	0.01	0.03	0.08
Waitall	0.01	0.05	0.04	1.53	0.04	0.12	0.14	0.31	0.05	0.12
Irecv	0	0.02	0	0.07	0.02	0.05	0.06	0.57	0.05	0.23





* "Waitany","AllReduce" and "Bcast" takes almost 50% of application time when number of block is much larger then number of nodes. It is even worse when block is unevenly distrubuted among nodes, like the case "-np=6", "-np=10"

zrad.0064 This is a 64 block problem, suitable for testing with at least 2 MPI Processes and up to 64 CPU's.The following types of runs should work well

64 MPI Processes

32 MPI Processes at 2 Threads each

16 MPI Processes at 4 Threads each

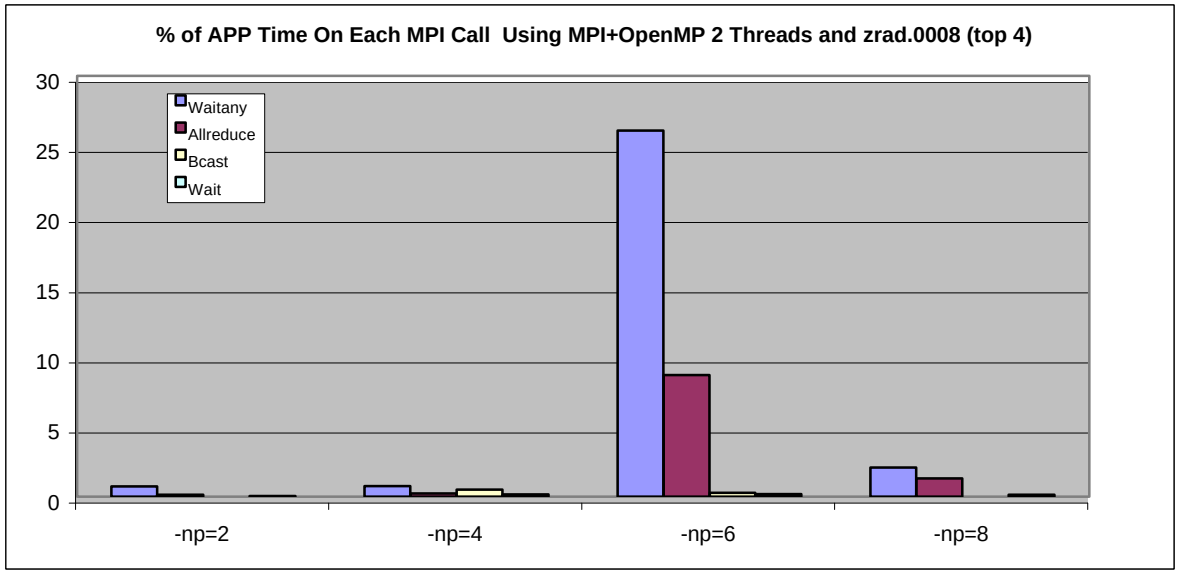
8 MPI Processes at 8 Threads each

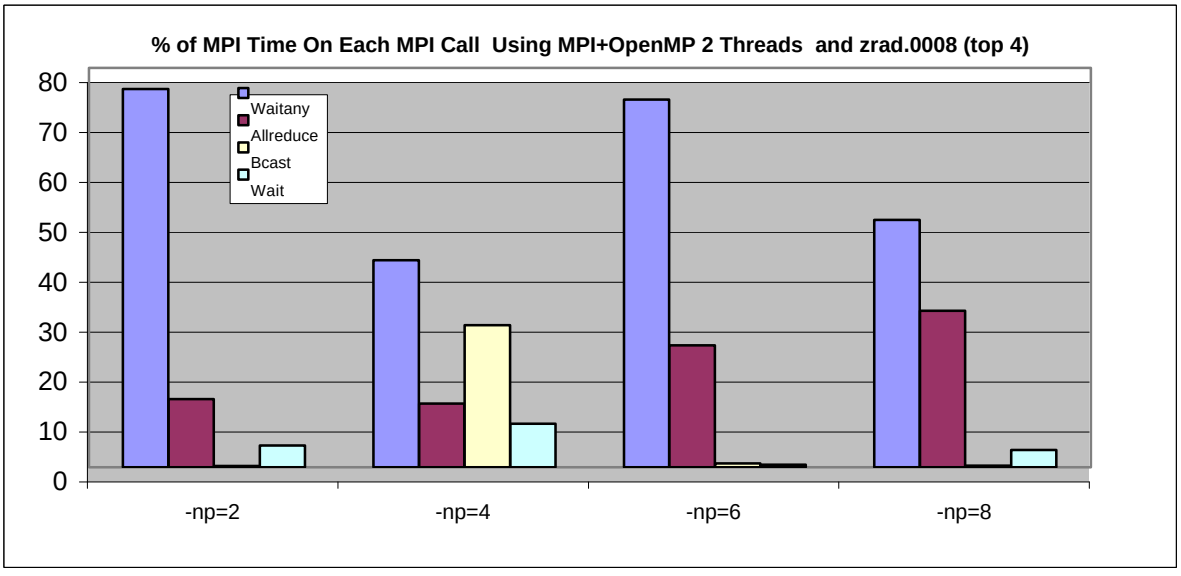
4 MPI Processes at 16 Threads each

2 MPI Processes at 32 Threads each

MPI/OpenMP 2 Threads with ZRAD0008

	-np=2		-np=4		-np=6		-np=8	
Call	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**
Waitany	0.72	75.76	0.75	41.5	26.09	73.63	2.07	49.54
Allreduce	0.13	13.67	0.23	12.74	8.66	24.43	1.31	31.32
Bcast	0	0.23	0.51	28.46	0.28	0.78	0.01	0.31
Wait	0.04	4.38	0.16	8.73	0.18	0.51	0.14	3.45
Isend	0.01	1.07	0.1	5.43	0.18	0.52	0.52	12.53
Waitall	0.01	0.05	0.04	2.38	0.02	0.07	0.04	0.86
Irecv	0.01	1.01	0.01	0.75	0.02	0.06	0.08	1.83

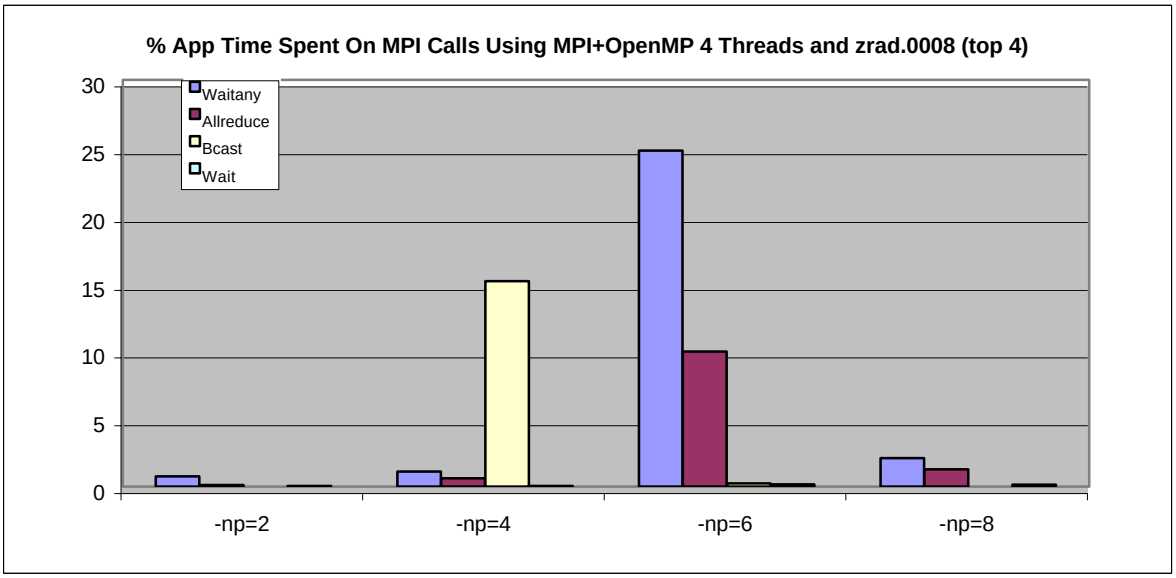


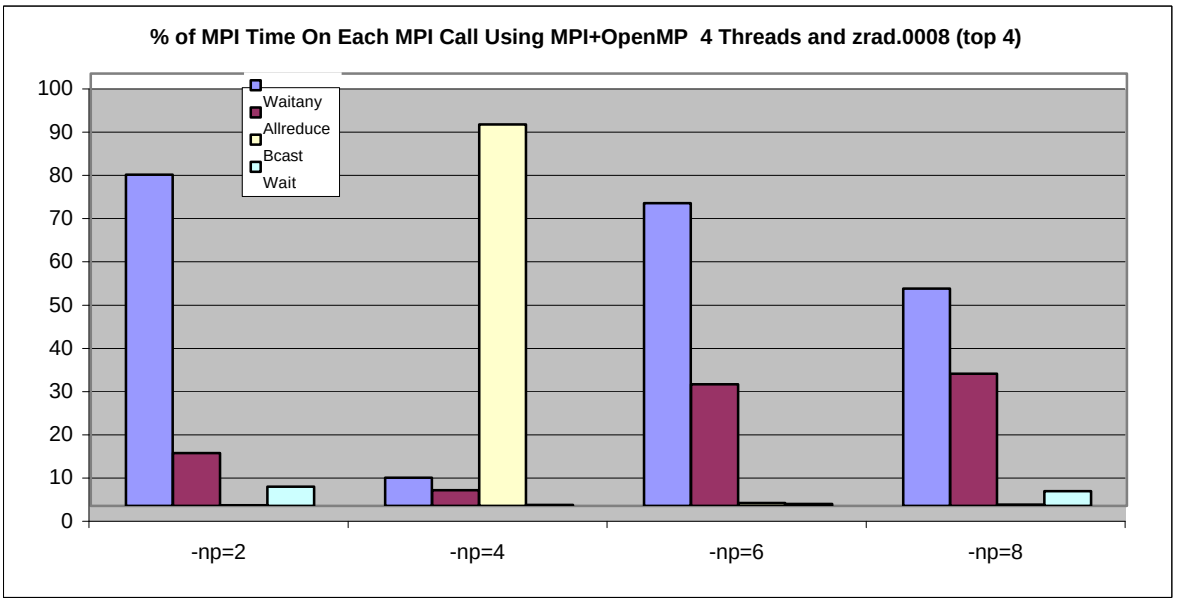


* MPI/OpenMP reduces significantly MPI time, when distributes multiple blocks to each single node. The communcation among processes within a node is faster then communication among processes between nodes.

MPI/OpenMP 2 Threads with ZRAD0008

	-np=2		-np=4		-np=6		-np=8	
Call	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**	**App%**	**MPI%**
Waitany	0.75	76.58	1.12	6.53	24.79	70.05	2.09	50.25
Allreduce	0.12	12.25	0.62	3.61	9.96	28.14	1.27	30.58
Bcast	0	0.15	15.16	88.24	0.23	0.66	0.01	0.33
Wait	0.04	4.42	0.04	0.23	0.16	0.45	0.14	3.41
Isend	0.04	3.71	0.16	0.96	0.21	0.58	0.52	12.59
Waitall	0.02	1.88	0.05	0.28	0.02	0.05	0.04	0.85
Irecv	0.01	0.98	0.02	0.14	0.02	0.06	0.08	1.83

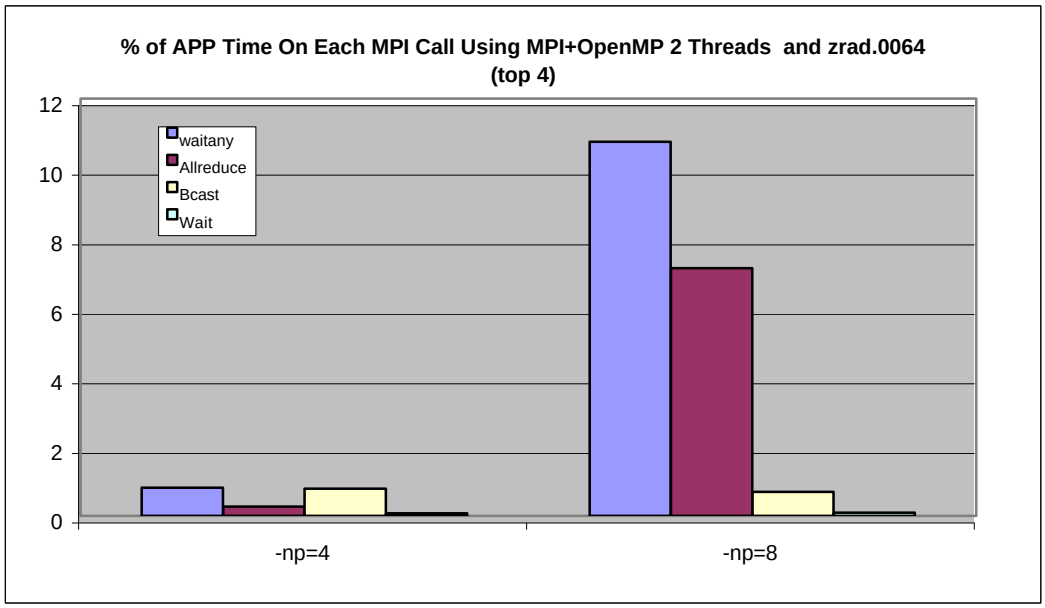


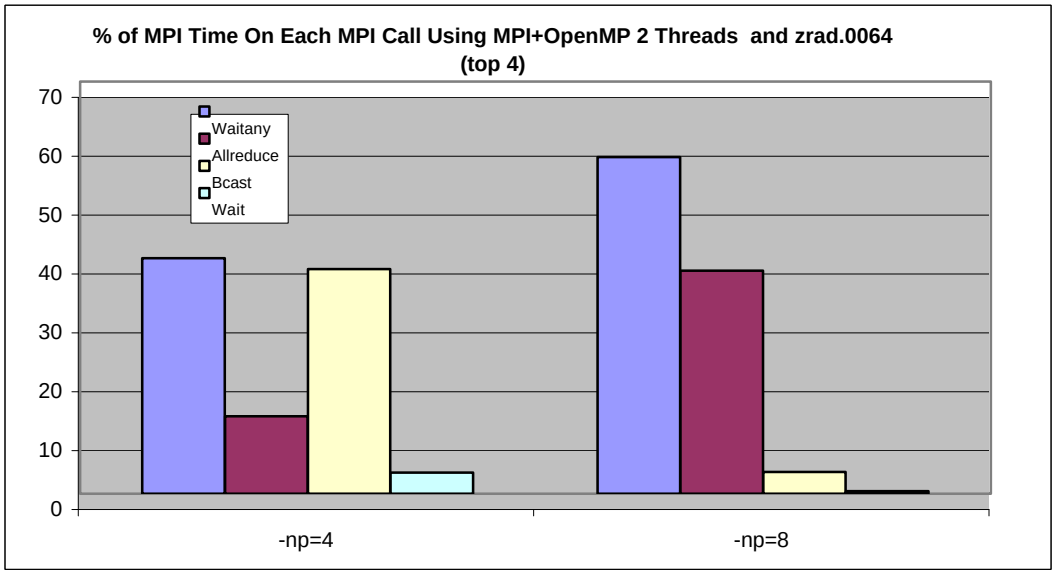


* Compare to MPI/OpenMP with 2 threads;in MPI/OpenMP with 4 threads MPI calls does not show any mjor change. Increasing number of OpenMP threads does not lead to any change MPI time.

MPI/OpenMP 2 Threads with ZRAD0064

	-np=4		-np=8	
Call	**App%**	**MPI%**	**App%**	**MPI%**
Waitany	0.81	39.99	10.76	57.2
Allreduce	0.27	13.15	7.12	37.86
Bcast	0.78	38.16	0.69	3.69
Wait	0.07	3.61	0.09	0.45
Isend	0.02	1.07	0.04	0.22
Waitall	0.08	3.91	0.1	0.55
Irecv	0	0.1	0	0.03

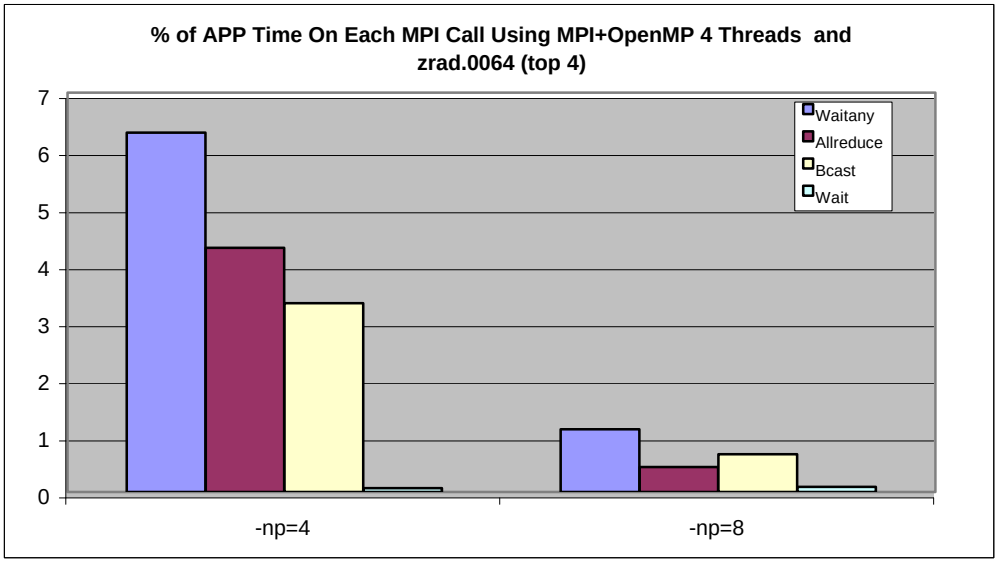


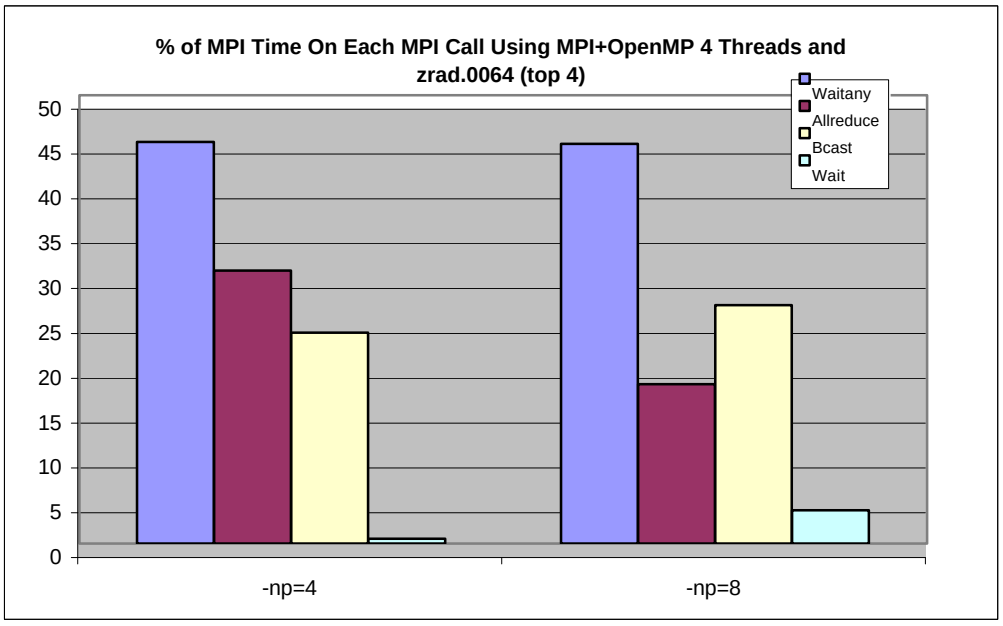


* When increasing input size, It likely need to increase number of nodes. In case of "np=2", "Waitany","Allreduce" and "Bcast" takes longer then "Waitany" in the case of "-np=8". When number of nodes small, there is likely lot of comminication between nodes and also facing problem of call blocking and synchronzation.

MPI/OpenMP 4 Threads with ZRAD0064

	-np=4		-np=8	
Call	**App%**	**MPI%**	**App%**	**MPI%**
Waitany	6.3	44.8	1.1	44.56
Allreduce	4.28	30.46	0.44	17.76
Bcast	3.31	23.52	0.66	26.59
Wait	0.07	0.53	0.09	3.71
Isend	0.02	0.14	0.04	0.22
Waitall	0.08	0.54	0.13	5.1
Irecv	0	0.01	0.01	0.23





* Compare to MPI/OpenMP with 2 threads;in MPI/OpenMP with 4 threads MPI calls does not show any mjor change. Increasing number of OpenMP threads does not lead to any change MPI time.

CONCLUSION

Waitany,"Allrecude" and "Bcast" is three major MPI calls that may have impact on a large scale application like IRS when it runs on cluster environment. There are few factors that may recude MPI call impact on application time:

* Number of nodes:
  When the size of data increase, number of nodes should inceases.

* Distributing Data:
  Minizing data dependecy on each node that help minize data exchange between nodes. This task also depend on number of

* Number of threads:
  This does not directly improve MPI call, but it help to determine how to distribute data among nodes in SMP enviroment.