

# **Communication Performance Measurement and Analysis on Commodity Clusters**

**A DISSERTATION  
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE  
OF THE UNIVERSITY OF ADELAIDE  
FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY**

By

Nor Asilah Wati Abdul Hamid

March 13, 2008

# Table of Content

<b>ABSTRACT</b> .....	<b>xiv</b>
<b>DECLARATION</b> .....	<b>xvi</b>
<b>LIST OF PUBLICATIONS</b> .....	<b>xvii</b>
<b>ACKNOWLEDGEMENT</b> .....	<b>xviii</b>
<b>CHAPTER 1</b> .....	<b>1</b>
<b>Introduction</b>	
1.1 Parallel computing architectures.....	2
1.2 MPI communications performance.....	4
1.3 Research Rationale.....	6
1.4 Research Aims and Overview.....	8
1.4.1 Comparison of Different Benchmark Software .....	9
1.4.2 Improvements to MPIBench .....	10
1.4.3 Performance Analysis and Investigation of Communication Performance on Different Communication Networks.....	10
1.4.4 Analysis of Algorithm Selection for Optimizing Collective Communication with MPICH for Ethernet and Myrinet Networks .....	11
1.4.5 Performance Evaluation on ccNUMA Shared Memory Machine SGI Altix 3000.....	12
1.5 Thesis Outline .....	13
<b>CHAPTER 2</b> .....	<b>14</b>
<b>Parallel Computing</b>	
2.1 Parallel Computers.....	14
2.1.1 Shared Memory MIMD Systems .....	16
2.1.2 Distributed Memory MIMD Systems .....	19
2.1.3 Distributed Memory System with SMP Nodes.....	22
2.2 Cluster Computer Interconnect.....	24
2.3 Parallel Programming .....	27

2.4 MPI Benchmark Software.....	29
2.5 Performance Analysis with MPIBench.....	30
2.6 Variation of Communication Performance.....	32
2.7 Improving the Communication Performance of Cluster Computers .....	33
<b>Chapter 3 .....</b>	<b>37</b>
<b>Comparison of MPI Benchmark Programs on Shared Memory and Distributed Memory Machines</b>	
3.1 Introduction.....	37
3.2 Related Work .....	38
3.3 MPI Benchmark Measurement Technique .....	38
3.3.1 Mpptest .....	40
3.3.2 Pallas MPI Benchmark .....	41
3.3.3 MPBench.....	42
3.3.4 SKaMPI.....	43
3.3.5 MPIBench .....	45
3.4 MPI Benchmark Functionality and Ease of Use.....	47
3.4.3 Presentation of output .....	49
3.5 Machines Used.....	50
3.5.1 ccNUMA Shared Memory Machine .....	50
3.5.2. Distributed Memory Machine.....	53
3.6 Point-to-Point Communication .....	54
3.6.1 MPI_Send/MPI_Recv .....	55
3.6.2 Bandwidth for MPI_Send/MPI_Recv .....	60
3.7 MPI_Sendrecv.....	64
3.8 Barrier .....	67
3.9 Broadcast.....	69
3.10 Scatter and Gather.....	76
3.11 Alltoall .....	79
3.12 Other Collective Communication .....	80
3.13 Discussion.....	80

<b>CHAPTER 4.....</b>	<b>82</b>
<b>Improvements for MPIBench</b>	
4.1 Introduction.....	82
4.2 Cache Effects .....	83
4.3 Testing the MPIBench Globally Synchronized Clock.....	84
4.4 Improved Measurement for Collective Communication .....	88
4.5 User-specified Communication Pattern for Point-to-Point Communications .....	89
4.6 Ring Pattern for Point-to-Point Communication .....	90
4.7 Programming Errors Fixed .....	95
4.8 Analysis of results over arbitrary set of processes.....	95
4.9 Added Options to Ease of Use .....	96
4.10 Future Work in MPIBench.....	96
<b>CHAPTER 5.....</b>	<b>98</b>
<b>Averages, Distributions and Scalability of MPI Communication Times for Ethernet and Myrinet Networks</b>	
5.1 Introduction.....	98
5.2 Related Work .....	100
5.3 Methodology.....	101
5.4 Point-to-Point Communication .....	102
5.4.1 Send/Receive.....	102
5.4.2 Combined Send and Receive .....	112
5.5 Barrier .....	114
5.6 Broadcast.....	116
5.7 Scatter and Gather.....	125
5.7.1 Scatter .....	125
5.7.2 Gather.....	131
5.8 Alltoall .....	136
5.9 Summary.....	146

<b>CHAPTER 6.....</b>	<b>148</b>
<b>Analysis of Algorithm Selection for Optimizing Collective Communication with MPICH for Ethernet and Myrinet Networks</b>	
6.1 Introduction.....	148
6.2 Related Work .....	151
6.3 Methodology.....	153
6.4 Broadcast.....	155
6.5 Alltoall .....	167
6.6 Reduce Scatter .....	172
6.7 Allgather .....	179
6.8 Other Collective Communication .....	183
6.8.2 Reduce.....	184
6.9 Summary .....	185
<b>CHAPTER 7.....</b>	<b>189</b>
<b>Performance Evaluation on ccNUMA Shared Memory Machine SGI Altix 3000</b>	
7.1 Introduction.....	189
7.2 MPI Benchmark Experiments on the Altix.....	190
7.3 Selection of Processors for Benchmarking.....	191
7.4 MPI_Send with Default Settings and Single Copy.....	193
7.5 Point-to-Point Communications.....	195
7.5.1 MPI_Sendrecv.....	200
7.6 Broadcast.....	200
7.7 Barrier .....	203
7.8 Scatter and Gather.....	204
7.9 Alltoall .....	208
7.10 Discussion.....	209
<b>CHAPTER 8.....</b>	<b>210</b>
<b>Conclusion and Further Work</b>	
<b>REFERENCES.....</b>	<b>217</b>

# List of Figures

## CHAPTER 2

Figure 2.1 : Examples of interconnection structures used in shared-memory MIMD systems.....	19
Figure 2.2 : Examples of common networks for Distributed Memory machine .....	22
Figure 2.3 : Block diagram of a system with a “hybrid” network: clusters of four CPUs are connected by a crossbar. ....	24

## CHAPTER 3

Figure 3.1: MPI benchmark measurement technique pseudocode .....	39
Figure 3.2 : Mpptest pseudocode .....	40
Figure 3.3 : SKaMPI pseudocode .....	44
Figure 3.4: MPIBench Pseudocode.....	45
Figure 3.5 : An Altix C-brick with 2 nodes, 2 NUMALink-3 and 2 XIO channels [124].	52
Figure 3.6 : SGI Altix 3000 communications architecture for 128 processors [124].....	53
Figure 3.7 : PMB and Mpptest Point-to-Point pattern.....	55
Figure 3.8 : SKaMPI and MPBench Point-to-Point pattern .....	55
Figure 3.9 : MPIBench Point-to-Point pattern.....	55
Figure 3.10 : Comparison of results from different MPI benchmarks for Point-to-Point (send/receive) communications using 8 processors between default settings and Single Copy (indicate by SC) on SGI Altix.....	57
Figure 3.11 : Comparison of results from different MPI benchmarks for Point-to-Point (send/receive) communications using the same process placement, with a single process on each of 2 different C-Bricks connected by a router, on the SGI Altix....	57
Figure 3.12 : Ratio of Send/Recv time using buffered compared (default) to non-buffered communication for PMB from 2 to 32 Processors .....	58
Figure 3.13 : Comparison of results from different MPI benchmarks for Point-to-Point (send/receive) communications using 8 processors on IBM Linux Cluster. ....	59

Figure 3.14 : Comparison of results from different MPI benchmarks for Point-to-Point (send/receive) communications using the same process placement, with a single process on each of 2 different nodes on IBM Linux Cluster. ....	59
Figure 3.15 : PMB Bandwidth Results for 2 until 32 Processors for Default Settings on SGI Altix. ....	62
Figure 3.16 : MPIBench Bandwidth Results for 2 until 32 Processors for Default Settings on SGI Altix. ....	63
Figure 3.17 : PMB Bandwidth Results for 2 until 32 Processors for Default Settings on IBM Linux Cluster. ....	63
Figure 3.18 : MPIBench Bandwidth Results for 2 until 32 Processors on IBM Linux Cluster. ....	64
Figure 3.19 : Comparison between MPI benchmarks for MPI_Sendrecv with MPIBench ring pattern on 8 processors on SGI Altix. ....	66
Figure 3.20 : Comparison between MPI benchmarks for MPI_Sendrecv with MPIBench ring pattern on 8 processors on IBM Linux cluster. ....	67
Figure 3.21 : Comparison between MPI benchmarks for MPI_Barrier for 2 to 128 processors on the SGI Altix. ....	68
Figure 3.22 : Comparison between MPI benchmarks for MPI_Barrier for 2 to 128 processors on the IBM Linux cluster. ....	68
Figure 3.23 : Comparison between MPI benchmarks for MPI_Bcast on 8 processors before tuning the code on SGI Altix. ....	70
Figure 3.24 : Comparison between MPI benchmarks for MPI_Bcast on 8 processors after tuning the code on SGI Altix. ....	71
Figure 3.25 : Comparison between MPI benchmarks for MPI_Bcast on 8 processors on IBM Linux Cluster. ....	71
Figure 3.26 : Node time produce by SKaMPI for MPI_Bcast at 4MBytes for 8 cpus on SGI Altix. ....	73
Figure 3.27 : Distribution result produce by MPIBench for MPI_Bcast at 4MBytes for 8 cpus on SGI Altix. ....	74
Figure 3.28 : Node time produce by SKaMPI for MPI_Bcast at 4MBytes for 8 cpus on IBM Linux Cluster. ....	74

Figure 3.29 : Distribution result produce by MPIBench for MPI_Bcast at 4MBytes for cpus on IBM Linux Cluster.....	75
Figure 3.30 : Minimum, Average and Maximum time from MPIBench for MPI_Bcast at 4MBytes for 8 cpus on IBM Linux Cluster. ....	75
Figure 3.31 : Comparison between MPI benchmarks for MPI_Scatter for 32 processors on SGI Altix.....	76
Figure 3.32 : Comparison between MPI benchmarks for MPI_Scatter for 32 processors on IBM Linux Cluster.....	77
Figure 3.33 : Comparison between MPI benchmarks for MPI_Gather for 32 processors on SGI Altix.....	78
Figure 3.34 : Comparison between MPI benchmarks for MPI_Gather for 32 processors on IBM Linux Cluster.....	78
Figure 3.35 : Comparison between MPI benchmarks for MPI_Alltoall on 32 processors on SGI Altix.....	79
Figure 3.36 : Comparison between MPI benchmarks for MPI_Alltoall on 32 processors on IBM Linux Cluster.....	80

## CHAPTER 4

Figure 4.1: Point-to-Point with 2 processors using MPI_Wtime at 128 Bytes.....	86
Figure 4.2 : Point-to-Point with 2 processors using Clock Cycle at 128 Bytes.....	86
Figure 4.3 : Point-to-Point with 2 processors using MPI_Wtime at 256 Kbytes. ....	87
Figure 4.4 : Point-to-Point with 2 processors using Clock Cycle at 256 KBytes. ....	87
Figure 4.5: MPI Bcast on Ethernet for 128 CPUs at 64KByte. ....	88
Figure 4.6 : MPI Alltoall on Ethernet for 64 CPUs at 4KByte.....	89
Figure 4.7 : MPIBench User-specified Point-to-Point Communication Pattern.....	90
Figure 4.8 : Ring Pattern for 4 CPU and 2 CPU per node.....	91
Figure 4.9 : Average times for MPI_Sendrecv with Ring pattern from 4 to 64 CPUs on SGI Altix.....	93
Figure 4.10 : Average times for MPI_Sendrecv with Ring pattern from 4 to 64 CPUs on IBM Linux Cluster.....	93
Figure 4.11 : Distribution for 4 CPUs on SGI Altix at 256KByte.....	94
Figure 4.12 : Distribution for 4 CPUs for Myrinet on Hydra at 256KByte.....	94



## CHAPTER 5

Figure 5.1 : Average Time for MPI_Send/MPI_Recv on Myrinet (MY) and Ethernet (ET).....	103
Figure 5.2 : Distribution of MPI_Send/Recv for Myrinet at 128 CPUs for 16 KByte. .	105
Figure 5.3: Distribution of MPI_Send/Recv times for Myrinet at 128 CPUs for 64 KByte. ....	106
Figure 5.4 : Distribution of MPI_Send/Recv times for Ethernet at 128 CPUs for 16 KByte. ....	106
Figure 5.5 : Distribution of MPI_Send/Recv for Ethernet at 128 CPUs for 64 KByte. .	107
Figure 5.6: Examples of the calculation of Min, Mean and Std. Dev. for 32 CPUs at 16 KByte for Myrinet.....	109
Figure 5.7 : Average Time for MPI_Sendrecv on Myrinet (MY) and Ethernet (ET). ...	113
Figure 5.8 : Average time for MPI_Barrier Myrinet and Ethernet. ....	115
Figure 5.9 : Distribution of MPI_Barrier times for Ethernet at 64,128 and 200 CPUs..	115
Figure 5.10 : Distribution of MPI_Barrier times for Myrinet at 64,128 and 200 CPUs.	116
Figure 5.11 : Average time for MPI_Bcast on Myrinet.....	117
Figure 5.12 : Average time for MPI_Bcast on Ethernet .....	118
Figure 5.13 : Myrinet at 128 CPUs for 256 KByte.....	119
Figure 5.14 : Ethernet at 128 CPUs for 8 KByte. ....	120
Figure 5.15 : Ethernet at 128 CPUs for 16 KByte. ....	120
Figure 5.16 : Ethernet at 128 CPUs for 32 KByte. ....	121
Figure 5.17 : Ethernet at 128 CPUs for 64 KByte. ....	121
Figure 5.18 : Ethernet at 128 CPUs for 256 KByte. ....	122
Figure 5.19 : Ethernet at 32 CPUs for 256 KByte. ....	123
Figure 5.20 : Minimum and Average Time for each CPU on Ethernet for 32 CPUs at 256 KByte .....	124
Figure 5.21 : Myrinet at 32 CPUs for 256 KByte.....	124
Figure 5.22 : Minimum and Average Time for each CPU on Myrinet for 32 CPUs at 256 KByte. ....	125
Figure 5.23 : Average time for MPI_Scatter on Myrinet.....	127
Figure 5.24 : Average time for MPI_Scatter on Ethernet. ....	128
Figure 5.25 : Myrinet at 128 CPUs for 64 KByte.....	128

Figure 5.26 : Minimum, Maximum and Average Time for each CPU on Myrinet for 128 CPUs at 64 KByte.....	129
Figure 5.27 : Ethernet at 128 CPUs for 64 KByte.....	129
Figure 5.28 : Minimum, Maximum and Average Time for each CPU on Ethernet for 128 CPUs at 64 KByte.....	130
Figure 5.29 : Minimum, Maximum and Average Time for each CPU on Ethernet for 16 CPUs at 64 KByte.....	130
Figure 5.30 : Average time for MPI_Gather on Myrinet.....	131
Figure 5.31 : Average time for MPI_Gather on Ethernet.....	132
Figure 5.32 : Myrinet for 128 CPUs at 64 KByte.....	133
Figure 5.33 : Minimum, Maximum and Average Time for each CPU on Myrinet for 128 CPUs at 64 KByte.....	134
Figure 5.34 : Ethernet for 128 nodes at 64 KByte.....	134
Figure 5.35 : Minimum, Maximum and Average Time for each CPU on Ethernet for 128 CPUs at 64 KByte.....	135
Figure 5.36 : Minimum and Average Time for each CPU on Ethernet for 16 CPUs at 64 KByte.....	135
Figure 5.37 : Average time for MPI_Alltoall on Myrinet for 4 to 200 CPUs.....	137
Figure 5.38 : Average time for MPI_Alltoall on Ethernet for 4 to 200 CPUs.....	138
Figure 5.39 : Myrinet at 128 CPUs for 2KByte.....	138
Figure 5.40 : Ethernet at 128 CPUs for 64Byte.....	139
Figure 5.41 : Minimum, Maximum and Average time on Ethernet for 128 CPUs at 64 Byte.....	139
Figure 5.42 : Ethernet at 128 CPUs for 256 Byte.....	140
Figure 5.43 : Minimum, Maximum and Average time on Ethernet for 128 CPUs at 256 Byte.....	140
Figure 5.44 : Ethernet at 128 CPUs for 1KByte.....	141
Figure 5.45 : Ethernet at 128 CPUs for 2KByte.....	141
Figure 5.46 : Ethernet at 64 CPUs for 2 KByte.....	143
Figure 5.47 : Ethernet at 64 CPUs for 4 KByte.....	143
Figure 5.48 : Ethernet at 64 CPUs for 8 KByte.....	144
Figure 5.49 : Ethernet at 64 CPUs for 16 KByte.....	144

Figure 5.50: Ethernet at 64 CPUs for 32 KByte. ....	145
Figure 5.51 : Myrinet at 64 CPUs for 32 KByte.....	145

## CHAPTER 6

Figure 6.1: 8 CPUs broadcast on Myrinet. ....	159
Figure 6.2 : 32 CPU Broadcast on Myrinet. ....	160
Figure 6.3 : 8 CPU Broadcast on Ethernet.....	161
Figure 6.4 : 32 CPU Broadcast on Ethernet.....	162
Figure 6.5 : Broadcast for 8 CPUs with 2 ppn for 16 KByte to 1 Mbyte on Ethernet....	163
Figure 6.6 : Broadcast for 8 CPUs with 1 ppn for 16 KByte to 1 Mbyte on Ethernet...	163
Figure 6.7 : Comparison between test results and model for 2 ppn for 32 CPUs for medium message size on Myrinet.....	165
Figure 6.8 : Comparison between test results and model for 1 ppn for 32 CPUs for medium message size on Myrinet.....	165
Figure 6.9 : Comparison between test results and model for 2ppn for 32 CPUs for large message size on Myrinet.....	166
Figure 6.10 : Comparison between test results and model for 1 ppn for 32 CPUs for large message size on Myrinet.....	166
Figure 6.11 : 8 CPU and 2 ppn for Alltoall on Myrinet.....	169
Figure 6.12 : 8 CPU and 2ppn for Alltoall on Ethernet.....	169
Figure 6.13 : 32 CPU and 2 ppn for Alltoall on Myrinet.....	170
Figure 6.14 : 32 CPU and 2 ppn Alltoall on Ethernet.....	170
Figure 6.15: 32 CPU and 1 ppn for Alltoall on Myrinet.....	171
Figure 6.16 : Comparison between test results and model for 2ppn for 32 CPUs on Myrinet.....	171
Figure 6.17 : Comparison between test results and model for 1ppn for 32 CPUs on Myrinet.....	172
Figure 6.18 : 8 CPU on Myrinet for Reduce Scatter.....	175
Figure 6.19 : 8 CPU on Ethernet for Reduce Scatter.....	176
Figure 6.20 : 32 CPU on Myrinet for Reduce Scatter.....	176
Figure 6.21 : 32 CPU and 2 ppn on Ethernet for Reduce Scatter.....	177

Figure 6.22 : Results for 32 CPUs and 1 ppn for Reduce Scatter on Myrinet.....	177
Figure 6.23 : Comparison between test results and model for 2ppn for 32 CPUs on Myrinet.....	178
Figure 6.24 : Comparison between test results and model for 1ppn for 32 CPUs on Myrinet.....	178
Figure 6.25 : 8 CPU and 2ppn on Myrinet for Allgather.....	181
Figure 6.26 : 8 CPU and 2ppn on Ethernet for Allgather .....	181
Figure 6.27 : 32 CPU and 2 ppn on Myrinet for Allgather.....	182
Figure 6.28 : 32 CPU and 2 ppn on Ethernet for Allgather .....	182
Figure 6.29 : Expected performance for 32 CPU and 1ppn for Gigabit Ethernet .....	187

## CHAPTER 7

Figure 7.1 : 8 CPUs for Point-to-Point at 256 KBytes using processor number from 0 to 7 and 16 to 23.....	191
Figure 7.2 : Communication for 32 processor for different group of processor for 256KBytes.....	193
Figure 7.3 : Average time for point-to-point using the default setting and single copy.	194
Figure 7.4 : Bandwidth for point-to-point using the default setting and single copy. ....	194
Figure 7.5 : Point-to-Point performance for small message sizes. ....	197
Figure 7.6 : Point-to-Point performance for large message sizes. ....	197
Figure 7.7 : Probability distributions for MPI point-to-point communications using 48 and 64 processors for 256 KByte message size.....	199
Figure 7.8 : Probability distributions for MPI point-to-point communications using 48 and 64 processors for 256 KByte message size using Single Copy options.....	199
Figure 7.9 : Performance of MPI_Bcast as a function of data size on 2 to 128 CPUs..	201
Figure 7.10 : Distribution results for MPI_Bcast at 64 Bytes on 32 cpus. ....	202
Figure 7.11 : Distribution result for MPI_Bcast at 256Kbytes on 32 cpus.....	203
Figure 7.12 : Average time for an MPI barrier operation for 2 to 128 processors. ....	204
Figure 7.13 : Performance for MPI_Scatter for 2 to 128 processors.....	205
Figure 7.14 : Distribution for MPI_Scatter for 64 processors at 256Kbytes .....	205
Figure 7.15 : Performance for MPI_Gather for 2 to 128 processors .....	207

Figure 7.16 : Distribution for MPI_Gather for 64 processors at 4Kbytes .....	207
Figure 7.17 : Performance for MPI_Alltoall for 2 to 128 processors.....	208
Figure 7.18 : Distribution for MPI_Alltoall for 32 processors at 256Kbytes.....	209

## List of Tables

### CHAPTER 1

Table 1.1 : Comparison of the architecture for high performance computer for year 1997 and 2007.....	3
Table 1.2 : Percentage of the most use interconnect from the statistic list at TOP500 website for November 2007 .....	4
Table 1.3 : Protocol Comparison (Ping-Pong application).....	5

### CHAPTER 2

Table 2.1 : Comparison for bandwidth, latency and cost between different interconnec.	27
---	----

### CHAPTER 3

Table 3.1 : SGI Altix brick type.....	52
Table 3.2 : Bandwidth results in MBytes/sec for various numbers of processors using default settings on SGI Altix.....	61
Table 3.3 : Bandwidth results in MBytes/sec for various numbers of processors on IBM Linux Cluster. ....	61
Table 3.4 : Comparison for average communication time (Microsec) between MPI_Send/MPI_Recv with MPI_Sendrecv for MPIBench on SGI Altix. ....	66

### CHAPTER 5

Table 5.1 : Percentage of times that are greater than n times and smaller than N+1 the minimum values for Myrinet. ....	110
Table 5.2 : Percentage of times that are greater than n times the minimum values for Ethernet. ....	111

Table 5.3 : Myrinet, percentage for average plus standard deviation for n = 1,2,3,4. ....	111
Table 5.4 : Ethernet, percentage for average plus standard deviation for n = 1,2,3,4. ...	112
Table 5.5 : Comparison for MPI_Send/MPI_Recv and MPI_Sendrecv Between Myrinet and Ethernet for 256 KByte messages. ....	114
Table 5.6 : Percentage of RTO occurrences for Broadcast for Ethernet on 128 CPUs and estimated average time without RTOs. ....	122
Table 5.7 : Percentage of RTO Occurrences for Alltoall for 32, 64 and 128 CPUs. ....	146

## CHAPTER 6

Table 6.1 : Summary of Algorithms used by MPICH for Broadcast. ....	156
Table 6.2 : Results for 8 CPUs for broadcast on Myrinet. ....	159
Table 6.3 : Results for 32 CPUs for broadcast on Myrinet. ....	160
Table 6.4 : Results for 8 CPUs for broadcast on Ethernet. ....	161
Table 6.5 : Results for 32 CPUs for broadcast on Ethernet. ....	162
Table 6.6 : Comparison results between 8p and 2ppn with 8p and 1ppn for Broadcast on Ethernet. ....	164
Table 6.7 : Comparison between MPICH2 1.0.4 with MPICH 1.2.6 for Broadcast on Ethernet. ....	164
Table 6.8 : Summary of Algorithms used for Alltoall in MPICH. ....	167
Table 6.9 : Summary of Algorithms used for Reduce Scatter in MPICH. ....	173
Table 6.10 : Summary of Algorithms used for Allgather in MPICH. ....	179
Table 6.11 : Summary of Algorithm uses in Allreduce. ....	184
Table 6.12 : Summary of Algorithm uses in Reduce. ....	185

## CHAPTER 7

Table 7.1 : Measured latency (for sending a zero byte message) and bandwidth (for a 4 MByte message) for different numbers of processes on the Altix. Results for MPIBench are for all processes communicating concurrently, so include contention effects. Results for MPBench (in bold font) are for only two communicating processes (processes 0 and N-1) with no network or memory contention. ....	195
--	-----

## ABSTRACT

Cluster computers have become the dominant architecture in high-performance computing. Parallel programs on these computers are mostly written using the Message Passing Interface (MPI) standard, so the communication performance of the MPI library for a cluster is very important. This thesis investigates several different aspects of performance analysis for MPI libraries, on both distributed memory clusters and shared memory parallel computers.

The performance evaluation was done using MPIBench, a new MPI benchmark program that provides some useful new functionality compared to existing MPI benchmarks. Since there has been only limited previous use of MPIBench, some initial work was done on comparing MPIBench with other MPI benchmarks, and improving its functionality, reliability, portability and ease of use. This work included a detailed comparison of results from the Pallas MPI Benchmark (PMB), SKaMPI, Mpptest, MPBench and MPIBench on both distributed memory and shared memory parallel computers, which has not previously been done. This comparison showed that the results for some MPI routines were significantly different between the different benchmarks, particularly for the shared memory machine.

A comparison was done between Myrinet and Ethernet network performance on the same machine, an IBM Linux cluster with 128 dual processor nodes, using the MPICH MPI library. The analysis focused mainly on the scalability and variability of communication times for the different networks, making use of the capability of MPIBench to generate distributions of MPI communication times. The analysis provided an improved understanding of the effects of TCP retransmission timeouts on Ethernet networks.

This analysis showed anomalous results for some MPI routines. Further investigation showed that this is because MPICH uses different algorithms for small and large message sizes for some collective communication routines, and the message size where this changeover occurs is fixed, based on measurements using a cluster with a single processor per node. Experiments were done to measure the performance of the different algorithms, which demonstrated that for some MPI routines the optimal changeover points were very different between Myrinet and Ethernet networks and for 1 and 2 proc-

essors per node. Significant performance improvements can be made by allowing the changeover points to be tuned rather than fixed, particularly for commodity Ethernet networks and for clusters with more than 1 process per node.

MPIBench was also used to analyse the MPI performance and scalability of a large ccNUMA shared memory machine, an SGI Altix 3000 with 160 processors. The results were compared with a high-end cluster, an AlphaServer SC with Quadrics QsNet interconnect. For most MPI routines the Altix showed significantly better performance, particularly when non-buffered copy was used. MPIBench proved to be a very capable tool for analyzing MPI performance in a variety of different situations.



## DECLARATION

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due to reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

---

NOR ASILAH WATI ABDUL HAMID

Phd Candidate,

Department of Computer Science

University of Adelaide

13 March 2008

## LIST OF PUBLICATIONS

The following papers were written based on the work presented in this thesis.

### *Papers in Refereed Conference Proceedings*

1. N. A. W. A Hamid and P. Coddington. “Analysis of Algorithm Selection for Optimizing Collective Communication with MPICH for Ethernet and Myrinet Networks”, *Proc. of The 8<sup>th</sup> International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT’07), Adelaide, Australia, December 2007.*
2. N. A. W. A Hamid and P. Coddington. “Averages, Distribution and Scalability of MPI Communication Times for Ethernet and Myrinet Networks”, *Proc. of Parallel and Distributed Computing and Network (PDCN’07), 25<sup>th</sup> IASTED International Multi-Conference, Congress Innsbruck, Austria, February 2007.*
3. N. A. W. A Hamid, P. Coddington and F. Vaughan. “Comparison of MPI Benchmark Programs on an SGI Altix ccNUMA Shared Memory Machine”, *Proc. of Workshop on Performance Modeling, Evaluation, and Optimization of Parallel and Distributed Systems (PMEO-PDS’06), Rhodes, Greece, April 2006.*
4. N. A. W. A Hamid, P. Coddington and F. Vaughan. “Performance Analysis of MPI Communications on the SGI Altix 3700”, *Proc. of APAC’05, Gold Coast, Australia, September 2005.*

## ACKNOWLEDGEMENT

The work described in this thesis was carried out at the University of Adelaide in the Department of Computer Science under the supervision of Dr. Paul Coddington.

I would like to express my deepest gratitude to Dr. Paul Coddington for his excellent guidance, enthusiastic supervision and tolerance throughout this research. His dedication and contribution gave me tremendous help in completion of this research and the publication of the papers. I also would like to thank him for his commitment to this research and for giving me the chance to present our papers at conferences around the world.

Many thanks are due to the University of Adelaide and in particular the Department of Computer Science, as well as the South Australian Partnership for Advanced Computing (SAPAC) for making supercomputer time available to me. Special thanks to Patrick Fitzhenry and Grant Ward for their professional advice and patience throughout the experimental program. I also would like to give thanks for the help from a team of programmers from the University of Adelaide and the South Australian Partnership of Advanced Computing, Alex Chichowski, Tim Seely and Paul Martinaitis.

Lastly to my family, my husband Dr. Raizal Saifulnaz Muhammad Rashid and my son Muhammad Haziq Raizal Saifulnaz and my parents for their love, encouragement and continual support throughout the years of doing this research.