

PERFORMANCE MODELLING OF MESSAGE-PASSING  
PARALLEL PROGRAMS

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

By

Duncan A. Grove, B.E.(Comp. Sys.)(Hons)

May 30, 2003



# Contents

<b>Abstract</b>	<b>xi</b>
<b>Declaration</b>	<b>xiii</b>
<b>Acknowledgements</b>	<b>xv</b>
<b>1 Parallel Computing</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Parallel Computers . . . . .	3
1.3 Parallel Programs . . . . .	5
1.4 Performance Modelling . . . . .	9
1.5 Thesis Outline . . . . .	11
<b>2 Performance Modelling Techniques</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Amdahl . . . . .	15
2.3 Fortune and Wylie . . . . .	15
2.4 Hoare; Milner; Alur and Dill . . . . .	16
2.5 Valiant . . . . .	18
2.6 Hockney . . . . .	19
2.7 Saavedra and Smith . . . . .	20
2.8 Culler <i>et al.</i> . . . . .	20
2.9 Grama <i>et al.</i> . . . . .	22
2.10 Adve . . . . .	23
2.11 Singh <i>et al.</i> . . . . .	25
2.12 Mehra <i>et al.</i> . . . . .	26
2.13 Parashar and Hariri . . . . .	28
2.14 Skillicorn . . . . .	30
2.15 Crovella and LeBlanc . . . . .	32
2.16 Mraz; Tabe <i>et al.</i> . . . . .	33
2.17 Clement, Quinn and Steed . . . . .	35
2.18 Islam . . . . .	37
2.19 Jonkers . . . . .	38

2.20	van Gemund . . . . .	41
2.21	Labarta and Girona <i>et al.</i> . . . . .	45
2.22	Dunlop and Hey <i>et al.</i> . . . . .	46
2.23	Becker <i>et al.</i> . . . . .	48
2.24	Gautama . . . . .	49
2.25	Tam and Wang . . . . .	50
2.26	Kranzlmüller and Schaubschläger . . . . .	50
2.27	Magnusson <i>et al.</i> ; Hughes <i>et al.</i> . . . . .	52
2.28	An Overview of the Approaches . . . . .	53
<b>3</b>	<b>The PEVPM Performance Model</b>	<b>55</b>
3.1	Introduction . . . . .	55
3.2	Key Features of Performance Models . . . . .	56
3.3	Scope . . . . .	61
3.4	Modelling Message-Passing Codes . . . . .	62
3.4.1	Modelling Local Processing . . . . .	63
3.4.2	Modelling Communication Events . . . . .	65
3.4.3	Combining Processing and Communication Models . . . . .	71
3.4.4	The Modelling Formalisms . . . . .	72
3.4.5	Building a PEVPM Model . . . . .	82
3.5	Automatic Performance Evaluation . . . . .	88
3.6	Advantages of the PEVPM Approach . . . . .	96
3.7	Implications for Other Parallel Methodologies . . . . .	98
3.8	Summary . . . . .	99
<b>4</b>	<b>Benchmarking Point-to-Point Communication</b>	<b>101</b>
4.1	Introduction . . . . .	101
4.2	Existing Message-Passing Benchmarks . . . . .	102
4.2.1	Genesis/PARKBENCH . . . . .	103
4.2.2	NetPIPE . . . . .	103
4.2.3	Pallas MPI Benchmarks . . . . .	104
4.2.4	MPBench . . . . .	104
4.2.5	Mpptest . . . . .	105
4.2.6	SKaMPI . . . . .	106
4.2.7	Profiling Tools . . . . .	106
4.2.8	Limitations of Existing Techniques . . . . .	107
4.3	Design and Implementation of MPIBench . . . . .	108
4.3.1	Constructing a Timing Harness . . . . .	108
4.3.2	An Accurate Global Clock . . . . .	109

4.3.3	Communication Patterns . . . . .	112
4.3.4	Generation of Results . . . . .	115
4.4	Benchmarking Experiments . . . . .	117
4.4.1	Machines Used . . . . .	117
4.4.2	Tests Performed . . . . .	120
4.5	Results for <code>MPI_Isend</code> . . . . .	122
4.5.1	Inter-node, end-to-end completion time . . . . .	122
4.5.2	Intra-node, end-to-end completion time . . . . .	136
4.5.3	Inter-node, local completion time . . . . .	139
4.6	Results for <code>MPI_Sendrecv</code> . . . . .	144
4.7	Analytical Models . . . . .	151
4.8	Stability and Interference . . . . .	162
4.9	Summary . . . . .	167
<b>5</b>	<b>Benchmarking Collective Communication</b>	<b>169</b>
5.1	Introduction . . . . .	169
5.2	Results for <code>MPI_Bcast</code> . . . . .	170
5.3	Results for <code>MPI_Barrier</code> . . . . .	184
5.4	Results for <code>MPI_Scatter</code> and <code>MPI_Gather</code> . . . . .	190
5.5	Results for <code>MPI_Alltoall</code> . . . . .	202
5.6	Discussion of Collective Computation . . . . .	213
5.7	Summary . . . . .	214
<b>6</b>	<b>Case Studies</b>	<b>217</b>
6.1	Introduction . . . . .	217
6.2	Jacobi Iteration . . . . .	219
6.3	Bag of Tasks . . . . .	231
6.4	Fast Fourier Transform . . . . .	239
6.5	Summary . . . . .	249
<b>7</b>	<b>Conclusions and Further Work</b>	<b>251</b>
<b>A</b>	<b>PEVPM Definitions</b>	<b>257</b>
<b>B</b>	<b>Using MPIBench</b>	<b>261</b>
B.1	Running MPIBench . . . . .	261
B.2	Customising MPIBench . . . . .	263
	<b>Bibliography</b>	<b>265</b>

# List of Algorithms

1	PEVPM Process Sweep . . . . .	90
2	PEVPM Match Sweep . . . . .	91

# List of Tables

1	Benchmarking experiments carried out in this thesis . . . . .	121
2	Perseus.2-64x1.barrier predictions and measurements . . . . .	187
3	APAC_NF.4-32x1-4.barrier measurements . . . . .	189

# List of Figures

## Parallel Computing

1	Common interconnection networks in parallel computers. . . . .	4
---	--	---

## PEVPM design

2	Inter-node communication pathway between MPI processes . . . . .	68
3	Performance distributions for point-to-point communication . . . . .	70
4	Computational basis of the PEVPM . . . . .	89
5	Locking semantics for 3+ process interactions . . . . .	93
6	PEVPM match-sweep line processing . . . . .	94

## MPIBench design

7	An iteration of the MPIBench clock synchronisation algorithm . . . . .	110
8	Clock drift and the MPIBench clock synchronisation algorithm . . . . .	111
9	Process placement for benchmarking balanced communication . . . . .	113
10	Example program with intra- and inter-node communication . . . . .	114
11	Raw output from MPIBench for perseus.32x1.bcast.small . . . . .	116

<b>MPI_Isend (inter-node) measurements</b>		
12	Perseus.2-64x1-2.isend.small.averages . . . . .	123
13	Perseus.2-64x1-2.isend.large.averages . . . . .	124
14	Perseus.64x2.isend.small.3dhistograms . . . . .	125
15	Perseus.32x1.isend.large.2dhistograms . . . . .	127
16	Perseus.64x1.isend.large.2dhistograms . . . . .	128
17	Orion.2-32x1-4.isend.small.averages . . . . .	129
18	Orion.2-32x1-4.isend.large.averages . . . . .	130
19	Orion.32x1-4.isend.small.3dhistograms . . . . .	131
20	Orion.32x1.isend.large.2dhistograms . . . . .	132
21	APAC_NF.2-32x1-4.isend.small.averages . . . . .	134
22	APAC_NF.2-32x1-4.isend.large.averages . . . . .	134
23	APAC_NF.32x1-4.isend.small.3dhistograms . . . . .	135
24	APAC_NF.32x1.isend.large.2dhistograms . . . . .	136
<b>MPI_Isend (intra-node) measurements</b>		
25	Perseus-Orion-APAC_NF.1x2-4.isend.small.averages . . . . .	137
26	Perseus-Orion-APAC_NF.1x2-4.isend.large.averages . . . . .	137
<b>MPI_Isend (local completion) measurements</b>		
27	Perseus.2-64x1-2.isendlocal.small.averages . . . . .	140
28	Perseus.2-64x1-2.isendlocal.large.averages . . . . .	140
29	Orion.2-32x1-4.isendlocal.small.averages . . . . .	142
30	Orion.2-32x1-4.isendlocal.large.averages . . . . .	142
31	APAC_NF.2-32x1-4.isendlocal.small.averages . . . . .	143
32	APAC_NF.2-32x1-4.isendlocal.large.averages . . . . .	143
<b>MPI_Sendrecv measurements</b>		
33	Perseus.2-64x1-2.sendrecv.small.averages . . . . .	146
34	Perseus.2-64x1-2.sendrecv.large.averages . . . . .	146
35	Orion.2-32x1-4.sendrecv.small.averages . . . . .	148
36	Orion.2-32x1-4.sendrecv.large.averages . . . . .	148
37	Orion.2x1.sendrecv.512.2dhistograms . . . . .	149
38	APAC_NF.2-32x1-4.sendrecv.small.averages . . . . .	150
39	APAC_NF.2-32x1-4.sendrecv.large.averages . . . . .	150
<b>Analytical models for MPI_Isend (inter-node) measurements</b>		
40	Perseus.32x1.isend.512.2dhistogram.fit . . . . .	158
41	Perseus.32x1.isend.16384.2dhistogram.fit . . . . .	158
42	Orion.32x1.isend.512.2dhistogram.fit . . . . .	159

43	Orion.32x1.isend.28672.2dhistogram.fit . . . . .	159
44	APAC_NF.32x1.isend.512.2dhistogram.fit . . . . .	160
45	APAC_NF.32x1.isend.16384.2dhistogram.fit . . . . .	160
<b>MPI_Bcast models and measurements</b>		
46	A software-based broadcast tree for 16 processes . . . . .	171
47	Perseus.4-64x1-2.bcast.small.averages . . . . .	172
48	Perseus.4-64x1-2.bcast.large.averages . . . . .	172
49	Perseus.32x1.bcast.65536.2dhistogram . . . . .	173
50	Perseus.4-64x1-2.bcast.large.outliers . . . . .	173
51	Perseus.16x1.bcast.128.2dhistogram . . . . .	174
52	Figure 46 augmented with per process delays . . . . .	176
53	Orion.4-32x1-4.bcast.large.averages . . . . .	179
54	Orion.16x1-4.32768.2dhistograms . . . . .	179
55	APAC_NF.4-32x1-4.bcast.large.averages . . . . .	181
56	APAC_NF.32x1-4.bcast.16384.2dhistograms . . . . .	181
57	APAC_NF.32x1.bcast.16384.2dhistogram (root process only) . . . . .	182
58	APAC_NF.16x1.bcast.32.2dhistogram (switch serialisation) . . . . .	182
59	APAC_NF.8x1.bcast.large.averages (software-based) . . . . .	183
60	APAC_NF.8x1.bcast-sw.16384.2dhistogram (software-based) . . . . .	183
<b>MPI_Barrier measurements</b>		
61	Perseus.4-64x1.barrier.2dhistograms . . . . .	186
62	Perseus.4-64x2.barrier.2dhistograms . . . . .	186
63	Orion.4-32x1.barrier.2dhistograms . . . . .	188
64	Orion.4-32x4.barrier.2dhistograms . . . . .	188
<b>MPI_Scatter and MPI_Gather measurements</b>		
65	Perseus.4-64x1-2.scatter.large.averages . . . . .	193
66	Perseus.16x1.scatter.65536.2dhistogram . . . . .	193
67	Perseus.4-64x1-2.gather.large.averages . . . . .	195
68	Perseus.8x1.gather.65536.2dhistogram . . . . .	195
69	Orion.4-32x1-4.scatter.large.averages . . . . .	197
70	Orion.16-32x1.scatter.65536.2dhistograms . . . . .	197
71	Orion.4-32x1-4.gather.large.averages . . . . .	198
72	Orion.8x1.gather.16384+32768.2dhistograms . . . . .	198
73	APAC_NF.4-32x1-4.scatter.large.averages . . . . .	199
74	APAC_NF.32x1.scatter.65536.2dhistogram . . . . .	199
75	APAC_NF.4-32x1-4.gather.large.averages . . . . .	200



76	APAC_NF.32x1-4.gather.65536.2dhistograms . . . . .	200
<b>MPI_Alltoall measurements</b>		
77	Perseus.4-64x1-2.alltoall.large.outliers . . . . .	204
78	Perseus.4-64x1-2.alltoall.large.averages . . . . .	204
79	Perseus.32-64x1.alltoall.4096.2dhistogram . . . . .	206
80	Perseus.64x2.alltoall.65536.2dhistogram . . . . .	206
81	Orion.4-32x1-4.alltoall.large.averages . . . . .	209
82	Orion.8-32x1-2.alltoall.65536.2dhistogram . . . . .	209
83	APAC_NF.4-32x1-4.alltoall.large.averages . . . . .	212
84	APAC-SC.16x1.alltoall.0-262144.2dhistograms . . . . .	212
<b>Jacobi Iteration code, predictions and measurements</b>		
85	Jacobi Iteration skeleton code. . . . .	220
86	Jacobi Iteration PEVPM annotations. . . . .	222
87	Perseus.2-64x1-2.jacobi.averages . . . . .	225
88	Perseus.2-64x1-2.jacobi.speedups . . . . .	225
89	Orion.2-32x1-4.jacobi.averages . . . . .	226
90	Orion.2-32x1-4.jacobi.speedups . . . . .	226
91	APAC_NF.2-32x1-4.jacobi.averages . . . . .	227
92	APAC_NF.2-32x1-4.jacobi.speedups . . . . .	227
<b>Bag of Tasks predictions and measurements</b>		
93	Bag of Tasks skeleton code. . . . .	232
94	Perseus.2-64x1-2.bots.averages . . . . .	236
95	Perseus.2-64x1-2.bots.speedups . . . . .	236
96	Orion.2-32x1-4.bots.averages . . . . .	237
97	Orion.2-32x1-4.bots.speedups . . . . .	237
98	APAC_NF.2-32x1-4.bots.averages . . . . .	238
99	APAC_NF.2-32x1-4.bots.speedups . . . . .	238
<b>2D Fast Fourier Transform predictions and measurements</b>		
100	2D Fast Fourier Transform skeleton code. . . . .	241
101	Perseus.2-64x1-2.fft.averages . . . . .	243
102	Perseus.2-64x1-2.fft.speedups . . . . .	243
103	Orion.2-32x1-4.fft.averages . . . . .	244
104	Orion.2-32x1-4.fft.speedups . . . . .	244
105	APAC_NF.2-32x1-4.fft.averages . . . . .	245
106	APAC_NF.2-32x1-4.fft.speedups . . . . .	245



# Abstract

Parallel computing is essential for solving very large scientific and engineering problems. An effective parallel computing solution requires an appropriate parallel machine and a well-optimised parallel program, both of which can be selected via performance modelling. This dissertation describes a new performance modelling system, called the Performance Evaluating Virtual Parallel Machine (PEVPM). Unlike previous techniques, the PEVPM system is relatively easy to use, inexpensive to apply and extremely accurate. It uses a novel bottom-up approach, where submodels of individual computation and communication events are dynamically constructed from data-dependencies, current contention levels and the performance distributions of low-level operations, which define performance variability in the face of contention. During model evaluation, the performance distribution attached to each submodel is sampled using Monte Carlo techniques, thus simulating the effects of contention. This allows the PEVPM to accurately simulate a program's execution structure, even if it is non-deterministic, and thus to predict its performance.

Obtaining these performance distributions required the development of a new benchmarking tool, called MPIBench. Unlike previous tools, which simply measure average message-passing time over a large number of repeated message transfers, MPIBench uses a highly accurate and globally synchronised clock to measure the performance of individual communication operations. MPIBench was used to benchmark three parallel computers, which encompassed a wide range of network performance capabilities, namely those provided by Fast Ethernet, Myrinet and QsNet. Network contention, a problem ignored by most research in this area, was found to cause extensive performance variation during message-passing operations. For point-to-point communication, this variation was best described by Pearson 5 distributions. Collective communication operations were able to be modelled using their constituent point-to-point operations. In cases of severe contention, extreme outliers were common in the observed performance distributions, which were shown to be the result of lost messages and their subsequent retransmit timeouts.

The highly accurate benchmark results provided by MPIBench were coupled with the PEVPM models of a range of parallel programs, and simulated by the PEVPM. These case studies proved that, unlike previous modelling approaches, the PEVPM technique successfully unites generality, flexibility, cost-effectiveness and accuracy in one performance modelling system for parallel programs. This makes it a valuable tool for the development of parallel computing solutions.



# Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. This thesis contains no material which has been previously published or written by another person, except where due reference has been made in the text. I consent to this copy of my thesis being available for loan and photocopying from the University Library.

---

Duncan A. Grove, B.E.(Comp. Sys.)(Hons)

May 30, 2003



# Acknowledgements

Firstly, my sincerest thanks go to my supervisors, Dr. Paul Coddington and Prof. Ken Hawick. Ken, your exuberance for parallel and distributed computing whetted my desire to undertake advanced research in computer science, if not my taste for whiskey; slàinte! Paul, your earnest and steadfast mentorship helped reveal to me the true nature of scientific enquiry, for which you have earned my deepest respect and gratitude. I would also like to thank my friend and unofficial advisor – on matters of thesis, life, the universe and everything – Dr. Francis Vaughan: Francis, you were often wrong, but somehow managed to always help me find the right answer!

Many thanks are due to the University of Adelaide and in particular its Department of Computer Science, as well as the Advanced Computational Systems and Research Data Networks CRCs, for supporting my research and giving me the chance to present the fruits of that labour at conferences around the world. I am also indebted to the School of Informatics at the University of Wales, Bangor, for inviting me to spend six months there to pursue my work. Likewise, I am grateful to the Centre for High Performance Computing and Applications at the University of Adelaide and the Australian Partnership for Advanced Computing for making supercomputer time available to me.

Studies aside, I am deeply thankful to the many friends who filled my life with laughter and joy throughout my PhD candidature. While it is impossible to name them all, some demand special mention: Benji, our lunches and walks by the river have held immeasurable pleasure for me; Craig, your insight for happenings in the department provided no end of fun; Kate – yes, you won, but the impetus from our show-down did us both much good; Richard, Jezz and Mike, thank you for looking over bits of thesis draft – it was much appreciated; and finally, to the Hungers crowd – I can't imagine a better bunch of friends to have had along for the ride.

Most importantly, I owe an immense debt of gratitude to my family. Gray, Bron and Lach: I could never be as happy as I am without each of you, who are so precious to me, in my life. Mum and Dad, you are more deserving of my thanks than anyone for bringing this thesis to fruition. Dad, your continual interest, advice, enthusiasm and support were my life-raft; Mum, your love and encouragement were my ration-kit. Last, and most of all, I would like to thank my fiancée, Alex, for, quite simply, everything: Alee, your companionship means the world to me.





Another turning point, a fork stuck in the road.  
Time grabs you by the wrist, directs you where to go.  
So make the best of this test, and don't ask why.  
It's not a question, but a lesson learned in time.  
It's something unpredictable, but in the end is right.  
I hope you had the time of your life.

So take the photographs, and still frames in your mind.  
Hang it on a shelf of good health and good time.  
Tattoos of memories and dead skin on trial.  
For what it's worth, it was worth all the while.  
It's something unpredictable, but in the end is right.  
I hope you had the time of your life.

*Good Riddance, Green Day*

