

# Contents

Preface .....	xv
Acknowledgments .....	xviii
About the Author .....	xix
<b>CHAPTER 1 Why Parallel Computing? .....</b>	<b>1</b>
1.1 Why We Need Ever-Increasing Performance .....	2
1.2 Why We're Building Parallel Systems .....	3
1.3 Why We Need to Write Parallel Programs .....	3
1.4 How Do We Write Parallel Programs? .....	6
1.5 What We'll Be Doing .....	8
1.6 Concurrent, Parallel, Distributed .....	9
1.7 The Rest of the Book .....	10
1.8 A Word of Warning .....	10
1.9 Typographical Conventions .....	11
1.10 Summary .....	12
1.11 Exercises .....	12
<b>CHAPTER 2 Parallel Hardware and Parallel Software .....</b>	<b>15</b>
2.1 Some Background .....	15
2.1.1 The von Neumann architecture .....	15
2.1.2 Processes, multitasking, and threads .....	17
2.2 Modifications to the von Neumann Model .....	18
2.2.1 The basics of caching .....	19
2.2.2 Cache mappings .....	20
2.2.3 Caches and programs: an example .....	22
2.2.4 Virtual memory .....	23
2.2.5 Instruction-level parallelism .....	25
2.2.6 Hardware multithreading .....	28
2.3 Parallel Hardware .....	29
2.3.1 SIMD systems .....	29
2.3.2 MIMD systems .....	32
2.3.3 Interconnection networks .....	35
2.3.4 Cache coherence .....	43
2.3.5 Shared-memory versus distributed-memory .....	46
2.4 Parallel Software .....	47
2.4.1 Caveats .....	47
2.4.2 Coordinating the processes/threads .....	48
2.4.3 Shared-memory .....	49

	2.4.4	Distributed-memory .....	53
	2.4.5	Programming hybrid systems .....	56
<b>2.5</b>		Input and Output .....	56
<b>2.6</b>		Performance .....	58
	2.6.1	Speedup and efficiency .....	58
	2.6.2	Amdahl's law .....	61
	2.6.3	Scalability .....	62
	2.6.4	Taking timings .....	63
<b>2.7</b>		Parallel Program Design .....	65
	2.7.1	An example .....	66
<b>2.8</b>		Writing and Running Parallel Programs .....	70
<b>2.9</b>		Assumptions .....	70
<b>2.10</b>		Summary .....	71
	2.10.1	Serial systems .....	71
	2.10.2	Parallel hardware .....	73
	2.10.3	Parallel software .....	74
	2.10.4	Input and output .....	75
	2.10.5	Performance .....	75
	2.10.6	Parallel program design .....	76
	2.10.7	Assumptions .....	76
<b>2.11</b>		Exercises .....	77
<b>CHAPTER 3</b>		<b>Distributed-Memory Programming with MPI .....</b>	<b>83</b>
<b>3.1</b>		Getting Started .....	84
	3.1.1	Compilation and execution .....	84
	3.1.2	MPI programs .....	86
	3.1.3	MPI_Init and MPI_Finalize .....	86
	3.1.4	Communicators, MPI_Comm_size and MPI_Comm_rank .....	87
	3.1.5	SPMD programs .....	88
	3.1.6	Communication .....	88
	3.1.7	MPI_Send .....	88
	3.1.8	MPI_Recv .....	90
	3.1.9	Message matching .....	91
	3.1.10	The status_p argument .....	92
	3.1.11	Semantics of MPI_Send and MPI_Recv .....	93
	3.1.12	Some potential pitfalls .....	94
<b>3.2</b>		The Trapezoidal Rule in MPI .....	94
	3.2.1	The trapezoidal rule .....	94
	3.2.2	Parallelizing the trapezoidal rule .....	96

3.3	Dealing with I/O .....	97
3.3.1	Output .....	97
3.3.2	Input .....	100
3.4	Collective Communication.....	101
3.4.1	Tree-structured communication.....	102
3.4.2	MPI_Reduce .....	103
3.4.3	Collective vs. point-to-point communications .....	105
3.4.4	MPI_Allreduce .....	106
3.4.5	Broadcast .....	106
3.4.6	Data distributions .....	109
3.4.7	Scatter .....	110
3.4.8	Gather .....	112
3.4.9	Allgather .....	113
3.5	MPI Derived Datatypes .....	116
3.6	Performance Evaluation of MPI Programs.....	119
3.6.1	Taking timings .....	119
3.6.2	Results .....	122
3.6.3	Speedup and efficiency .....	125
3.6.4	Scalability .....	126
3.7	A Parallel Sorting Algorithm .....	127
3.7.1	Some simple serial sorting algorithms .....	127
3.7.2	Parallel odd-even transposition sort .....	129
3.7.3	Safety in MPI programs .....	132
3.7.4	Final details of parallel odd-even sort .....	134
3.8	Summary .....	136
3.9	Exercises .....	140
3.10	Programming Assignments .....	147
<b>CHAPTER 4</b>	<b>Shared-Memory Programming with Pthreads.....</b>	<b>151</b>
4.1	Processes, Threads, and Pthreads .....	151
4.2	Hello, World .....	153
4.2.1	Execution.....	153
4.2.2	Preliminaries .....	155
4.2.3	Starting the threads .....	156
4.2.4	Running the threads .....	157
4.2.5	Stopping the threads .....	158
4.2.6	Error checking .....	158
4.2.7	Other approaches to thread startup .....	159
4.3	Matrix-Vector Multiplication .....	159
4.4	Critical Sections .....	162

4.5	Busy-Waiting .....	165
4.6	Mutexes .....	168
4.7	Producer-Consumer Synchronization and Semaphores.....	171
4.8	Barriers and Condition Variables.....	176
4.8.1	Busy-waiting and a mutex .....	177
4.8.2	Semaphores .....	177
4.8.3	Condition variables.....	179
4.8.4	Pthreads barriers .....	181
4.9	Read-Write Locks .....	181
4.9.1	Linked list functions.....	181
4.9.2	A multi-threaded linked list.....	183
4.9.3	Pthreads read-write locks .....	187
4.9.4	Performance of the various implementations .....	188
4.9.5	Implementing read-write locks .....	190
4.10	Caches, Cache Coherence, and False Sharing .....	190
4.11	Thread-Safety.....	195
4.11.1	Incorrect programs can produce correct output.....	198
4.12	Summary .....	198
4.13	Exercises .....	200
4.14	Programming Assignments .....	206
<b>CHAPTER 5</b>	<b>Shared-Memory Programming with OpenMP .....</b>	<b>209</b>
5.1	Getting Started.....	210
5.1.1	Compiling and running OpenMP programs.....	211
5.1.2	The program .....	212
5.1.3	Error checking .....	215
5.2	The Trapezoidal Rule .....	216
5.2.1	A first OpenMP version .....	216
5.3	Scope of Variables .....	220
5.4	The Reduction Clause .....	221
5.5	The parallel for Directive .....	224
5.5.1	Caveats .....	225
5.5.2	Data dependences.....	227
5.5.3	Finding loop-carried dependences.....	228
5.5.4	Estimating $\pi$ .....	229
5.5.5	More on scope .....	231
5.6	More About Loops in OpenMP: Sorting .....	232
5.6.1	Bubble sort .....	232
5.6.2	Odd-even transposition sort.....	233
5.7	Scheduling Loops .....	236
5.7.1	The schedule clause.....	237
5.7.2	The static schedule type.....	238

	5.7.3	The dynamic and guided schedule types.....	239
	5.7.4	The runtime schedule type.....	239
	5.7.5	Which schedule?.....	241
<b>5.8</b>		Producers and Consumers.....	241
	5.8.1	Queues.....	241
	5.8.2	Message-passing.....	242
	5.8.3	Sending messages.....	243
	5.8.4	Receiving messages.....	243
	5.8.5	Termination detection.....	244
	5.8.6	Startup.....	244
	5.8.7	The atomic directive.....	245
	5.8.8	Critical sections and locks.....	246
	5.8.9	Using locks in the message-passing program.....	248
	5.8.10	critical directives, atomic directives, or locks?.....	249
	5.8.11	Some caveats.....	249
<b>5.9</b>		Caches, Cache Coherence, and False Sharing.....	251
<b>5.10</b>		Thread-Safety.....	256
	5.10.1	Incorrect programs can produce correct output.....	258
<b>5.11</b>		Summary.....	259
<b>5.12</b>		Exercises.....	263
<b>5.13</b>		Programming Assignments.....	267
<b>CHAPTER 6</b>		<b>Parallel Program Development.....</b>	<b>271</b>
<b>6.1</b>		Two $n$ -Body Solvers.....	271
	6.1.1	The problem.....	271
	6.1.2	Two serial programs.....	273
	6.1.3	Parallelizing the $n$ -body solvers.....	277
	6.1.4	A word about I/O.....	280
	6.1.5	Parallelizing the basic solver using OpenMP.....	281
	6.1.6	Parallelizing the reduced solver using OpenMP.....	284
	6.1.7	Evaluating the OpenMP codes.....	288
	6.1.8	Parallelizing the solvers using pthreads.....	289
	6.1.9	Parallelizing the basic solver using MPI.....	290
	6.1.10	Parallelizing the reduced solver using MPI.....	292
	6.1.11	Performance of the MPI solvers.....	297
<b>6.2</b>		Tree Search.....	299
	6.2.1	Recursive depth-first search.....	302
	6.2.2	Nonrecursive depth-first search.....	303
	6.2.3	Data structures for the serial implementations.....	305
	6.2.4	Performance of the serial implementations.....	306
	6.2.5	Parallelizing tree search.....	306

6.2.6	A static parallelization of tree search using pthreads .....	309
6.2.7	A dynamic parallelization of tree search using pthreads .....	310
6.2.8	Evaluating the pthreads tree-search programs .....	315
6.2.9	Parallelizing the tree-search programs using OpenMP .....	316
6.2.10	Performance of the OpenMP implementations .....	318
6.2.11	Implementation of tree search using MPI and static partitioning .....	319
6.2.12	Implementation of tree search using MPI and dynamic partitioning .....	327
<b>6.3</b>	<b>A Word of Caution .....</b>	<b>335</b>
<b>6.4</b>	<b>Which API? .....</b>	<b>335</b>
<b>6.5</b>	<b>Summary .....</b>	<b>336</b>
6.5.1	Pthreads and OpenMP .....	337
6.5.2	MPI .....	338
<b>6.6</b>	<b>Exercises .....</b>	<b>341</b>
<b>6.7</b>	<b>Programming Assignments .....</b>	<b>350</b>
<b>CHAPTER 7</b>	<b>Where to Go from Here .....</b>	<b>353</b>
<b>References</b>	<b>.....</b>	<b>357</b>
<b>Index</b>	<b>.....</b>	<b>361</b>