

Introduction to Parallel Programming

Subodh Kumar



Contents

<i>List of Figures</i>	xiii
Introduction	xvii
Concurrency and Parallelism	xvii
Why Study Parallel Programming	xvii
What Is in This Book	xix
1 An Introduction to Parallel Computer Architecture	1
1.1 Parallel Organization	1
SISD: Single Instruction, Single Data	2
SIMD: Single Instruction, Multiple Data	2
MIMD: Multiple Instruction, Multiple Data	3
MISD: Multiple Instruction, Single Data	3
1.2 System Architecture	4
1.3 CPU Architecture	5
1.4 Memory and Cache	8
1.5 GPU Architecture	11
1.6 Interconnect Architecture	13
Routing	13
Links	13
Types and Quality of Networks	14
Torus Network	16
Hypercube Network	18
Cross-Bar Network	19
Shuffle-Exchange Network	20
Clos Network	21
Tree Network	22
Network Comparison	24
1.7 Summary	24
2 Parallel Programming Models	31
2.1 Distributed-Memory Programming Model	32
2.2 Shared-Memory Programming Model	33
2.3 Task Graph Model	35
2.4 Variants of Task Parallelism	37
2.5 Summary	39

3	Parallel Performance Analysis	45
3.1	Simple Parallel Model	46
3.2	Bulk-Synchronous Parallel Model	47
	BSP Computation Time	48
	BSP Example	49
3.3	PRAM Model	52
	PRAM Computation Time	55
	PRAM Example	55
3.4	Parallel Performance Evaluation	57
	Latency and Throughput	57
	Speed-up	58
	Cost	58
	Efficiency	59
	Scalability	59
	Iso-efficiency	60
3.5	Parallel Work	62
	Brent's Work-Time Scheduling Principle	63
3.6	Amdahl's Law	63
3.7	Gustafson's Law	65
3.8	Karp-Flatt Metric	66
3.9	Summary	67
4	Synchronization and Communication Primitives	75
4.1	Threads and Processes	75
4.2	Race Condition and Consistency of State	77
	Sequential Consistency	78
	Causal Consistency	82
	FIFO and Processor Consistency	82
	Weak Consistency	84
	Linearizability	85
4.3	Synchronization	85
	Synchronization Condition	86
	Protocol Control	86
	Progress	86
	Synchronization Hazards	88
4.4	Mutual Exclusion	90
	Lock	90
	Peterson's Algorithm	91

Bakery Algorithm	94
Compare and Swap	95
Transactional Memory	96
Barrier and Consensus	97
4.5 Communication	99
Point-to-Point Communication	99
RPC	102
Collective Communication	102
4.6 Summary	104
5 Parallel Program Design	111
5.1 Design Steps	112
Granularity	112
Communication	113
Synchronization	114
Load Balance	115
5.2 Task Decomposition	115
Domain Decomposition	116
Functional Decomposition	120
Task Graph Metrics	123
5.3 Task Execution	124
Preliminary Task Mapping	125
Task Scheduling Framework	126
Centralized Push Scheduling Strategy	127
Distributed Push Scheduling	129
Pull Scheduling	129
5.4 Input/Output	130
5.5 Debugging and Profiling	132
5.6 Summary	133
6 Middleware: The Practice of Parallel Programming	139
6.1 OpenMP	139
Preliminaries	140
OpenMP Thread Creation	140
OpenMP Memory Model	141
OpenMP Reduction	143
OpenMP Synchronization	144
Sharing a Loop's Work	147
Other Work-Sharing Pragmas	150

SIMD Pragma	151
Tasks	153
6.2 MPI	155
MPI Send and Receive	156
Message-Passing Synchronization	158
MPI Data Types	161
MPI Collective Communication	164
MPI Barrier	167
MPI Reduction	167
One-Sided Communication	169
MPI File IO	173
MPI Groups and Communicators	176
MPI Dynamic Parallelism	177
MPI Process Topology	178
6.3 Chapel	180
Partitioned Global Address Space	180
Chapel Tasks	181
Chapel Variable Scope	183
6.4 Map-Reduce	184
Parallel Implementation	185
Hadoop	186
6.5 GPU Programming	188
OpenMP GPU Off-Load	188
Data and Function on Device	191
Thread Blocks in OpenMP	193
CUDA	194
CUDA Programming Model	195
CPU-GPU Memory Transfer	197
Concurrent Kernels	198
CUDA Synchronization	199
CUDA Shared Memory	202
CUDA Parallel Memory Access	203
False Sharing	206
6.6 Summary	207
7 Parallel Algorithms and Techniques	211
7.1 Divide and Conquer: Prefix-Sum	212
Parallel Prefix-Sum: Method 1	214

Parallel Prefix-Sum: Method 2	215
Parallel Prefix-Sum: Method 3	215
7.2 Divide and Conquer: Merge Two Sorted Lists	217
Parallel Merge: Method 1	218
Parallel Merge: Method 2	219
Parallel Merge: Method 3	222
Parallel Merge: Method 4	226
7.3 Accelerated Cascading: Find Minima	227
7.4 Recursive Doubling: List Ranking	230
7.5 Recursive Doubling: Euler Tour	231
7.6 Recursive Doubling: Connected Components	233
7.7 Pipelining: Merge-Sort	238
Basic Merge-Sort	238
Pipelined Merges	240
4-Cover Property Analysis	245
Merge Operation per Tick	248
7.8 Application of Prefix-Sum: Radix-Sort	249
7.9 Exploiting Parallelism: Quick-Sort	250
7.10 Fixing Processor Count: Sample-Sort	254
7.11 Exploiting Parallelism: Minimum Spanning Tree	257
Parallel Priority Queue	260
MST with Parallel Priority Queue	263
7.12 Summary	264
<i>Bibliography</i>	269
<i>Index</i>	277

Figures

1.1	Shared-memory vs. distributed-memory architecture	4
1.2	Parallel computing cluster	4
1.3	Computing system	5
1.4	Computing core	6
1.5	Cache coherence	9
1.6	GPU architecture	12
1.7	Completely connected network	14
1.8	A bus network	14
1.9	Ring network	16
1.10	$4 \times 4 \times 4$ 3D Mesh network	17
1.11	A 4×4 2D torus	17
1.12	A 1D torus layout with no long links	18
1.13	Hypercube network	19
1.14	Cross-bar	19
1.15	Omega network	20
1.16	Butterfly network	21
1.17	A Clos network	22
1.18	Tree network	22
1.19	Fat tree network	23
1.20	Folded Clos network	23
1.21	Clos = Fat tree network	24
2.1	Distributed-memory programming model	32
2.2	Shared-memory programming model	34
2.3	Task graph	36
2.4	Data parallelism	37
2.5	Task pipeline	38
2.6	Example task graphs	42
3.1	BSP computation model	48
3.2	Binary tree-like computation tree	51
3.3	PRAM computation model	54
3.4	Efficiency curve: speed-up vs. processor count	60
3.5	Maximum speed-up possible with different processor counts	65
3.6	Maximum speed-up possible by scaling problem size with processor count	66

7.14	Compute $\text{Ranklist}(\text{list1}, \text{list2})$ given $\text{Ranklist}(\text{list1}, X)$ and $\text{Ranklist}(X, \text{list2})$	245
7.15	Merging using cover	247
7.16	Partition a list	251
7.17	CRCW Quick-sort	253
7.18	Sampling a list for sample-sort	256
7.19	Minimum spanning tree of a weighted graph	257
7.20	Examples of Binomial trees of rank 0 to rank 4	261