Contents

List of Figures 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
# Contents

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
# Table of Contents

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
Table of Contents

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
## Contents

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

Bibliography | 269 |
Index | 277 |