

Contents

Part I High Performance Computing (HPC) with the Message Passing Interface (MPI)

1	A Glance at High Performance Computing (HPC)	3
1.1	What is High Performance Computing (HPC)?	3
1.2	Why Do We Need HPC?	4
1.3	Big Data: The Four Vs (Volume, Variety, Velocity, Value).	5
1.4	Parallel Programming Paradigms: MPI and MapReduce	6
1.5	Granularity: Fined-Grained Versus Coarse-Grained Parallelism	6
1.6	Architectures for Supercomputing: Memory and Network	7
1.7	Speedup	11
1.7.1	Scalability and Iso-efficiency Analysis.	11
1.7.2	Amdahl's Law: Characterizing the Asymptotic Speedup for Fixed Data-Size Problems	12
1.7.3	Gustafson's Law: Scaled Speedup, Increasing Data Size with Resources	14
1.7.4	Simulating Parallel Machines on Sequential Computers	15
1.7.5	Big Data and Parallel Inputs/Outputs (I/O).	16
1.8	Eight Common Fallacies on Distributed Systems	17
1.9	Notes and References	19
1.10	Summary	19
1.11	Exercises.	20
	References	20
2	Introduction to MPI: The Message Passing Interface	21
2.1	MPI for Parallel Programming: Communicating with Messages	21
2.2	Parallel Programming Models, Threads and Processes.	22

2.3	Global Communications Between Processes.	23
2.3.1	Four Basic MPI Primitives: Broadcast, Gather, Reduce, and Total Exchange	24
2.3.2	Blocking Versus Non-blocking and Synchronous Versus Asynchronous Communications	26
2.3.3	Deadlocks from Blocking Communications	28
2.3.4	Concurrency: Local Computations Can Overlap with Communications	32
2.3.5	Unidirectional Versus Bidirectional Communications.	33
2.3.6	Global Computations in MPI: Reduce and Parallel Prefix (Scan)	33
2.3.7	Defining Communication Groups with Communicators	35
2.4	Synchronization Barriers: Meeting Points of Processes	37
2.4.1	A Synchronization Example in MPI: Measuring the Execution Time.	37
2.4.2	The Bulk Synchronous Parallel (BSP) Model.	38
2.5	Getting Started with the MPI: Using OpenMPI	39
2.5.1	The “Hello World” Program with MPI C++.	39
2.5.2	Programming MPI with the C Binding	41
2.5.3	Using MPI with C++ Boost.	41
2.6	Using MPI with OpenMP	42
2.6.1	Programming MPI with the Python Binding.	43
2.7	Main Primitives in MPI	44
2.7.1	MPI Syntax for Broadcast, Scatter, Gather, Reduce and Allreduce.	44
2.7.2	Other Miscellaneous MPI Primitives	45
2.8	Communications on the Ring Topology with MPI	47
2.9	Examples of MPI Programs with Their Speed-Up Analysis	48
2.9.1	The Matrix–Vector Product in MPI.	49
2.9.2	Example of MPI Reduce Operations: Computing the Factorial and Minimum Value of an Array	50
2.9.3	Approximating π with Monte-Carlo Stochastic Integration	52
2.9.4	Monte-Carlo Stochastic Integration for Approximating the Volume of a Molecule	54
2.10	References and Notes	59
2.11	Summary	60
2.12	Exercises.	60
	References	62

3	Topology of Interconnection Networks	63
3.1	Two Important Concepts: Static Versus Dynamic Networks, and Logic Versus Physical Networks	63
3.2	Interconnection Network: Graph Modeling	64
3.3	Some Attributes Characterizing Topologies	65
3.3.1	Degree and Diameter	65
3.3.2	Connectivity and Bisection	65
3.3.3	Criteria for a Good Network Topology	66
3.4	Common Topologies: Simple Static Networks	66
3.4.1	Complete Graphs: The Cliques	66
3.4.2	Star-Shaped Graph	67
3.4.3	Rings and Chordal Rings	67
3.4.4	Meshes (Grids) and Torii (Plural of Torus)	68
3.4.5	The 3D Cube and the Cycle Connected Cube (CCC)	69
3.4.6	Trees and Fat Trees	69
3.5	Topology of Hypercubes and Node Labeling Using the Gray Code	71
3.5.1	Recursive Construction of the Hypercube	71
3.5.2	Numbering Hypercube Nodes Using the Gray Code	72
3.5.3	Generating Gray Codes in C++	73
3.5.4	Converting Gray Codes to and from Binary Codes	75
3.5.5	Cartesian Product \square of Graphs	76
3.6	Some Communication Algorithms on Topologies	78
3.6.1	Communication Primitives on the Oriented Ring	79
3.6.2	Broadcasting on the Hypercube: Tree-Like Communications	86
3.7	Embedding (Logical) Topologies in Other (Physical) Topologies	91
3.8	Complex Regular Topologies	92
3.9	Interconnection Network on a Chip	93
3.10	Notes and References	96
3.11	Summary	96
	References	97
4	Parallel Sorting	99
4.1	Quick Review of Sequential Sorting	99
4.1.1	Main Sequential Sorting Algorithms	99
4.1.2	Complexity of Sorting: A Lower Bound	101
4.2	Parallel Sorting by Merging Lists	102
4.3	Parallel Sorting Using Ranks	102
4.4	Parallel Quicksort	104
4.5	HyperQuickSort	108
4.6	Parallel Sort Regular Sampling (PSRS)	109

4.7	Sorting on Grids: ShearSort	111
4.8	Sorting Using Comparison Network: Odd–Even Sorting	113
4.9	Merging Sorted Lists Using a Comparison Network	115
4.10	The Bitonic Merge Sort	116
4.11	Notes and References	118
4.12	Summary	120
4.13	Exercises.	120
	References	120
5	Parallel Linear Algebra	121
5.1	Distributed Linear Algebra	121
5.1.1	Linear Algebra for Data Science.	121
5.1.2	Classic Linear Algebra	125
5.1.3	The Matrix–Vector Product: $y = Ax$	126
5.1.4	Data Patterns for Parallelism	127
5.2	Matrix–Vector Product on the Topology of the Oriented Ring.	128
5.3	Matrix Product on the Grid: The Outer Product Algorithm . . .	134
5.4	Matrix Products on the Topology of the 2D Torus	134
5.4.1	Cannon’s Algorithm	136
5.4.2	Fox’s Algorithm: The Broadcast-Multiply-Roll Matrix Product.	138
5.4.3	Snyder’s Algorithm: Accumulating Local Products on the Diagonal	141
5.4.4	Comparisons of Cannon’s, Fox’s and Snyder’s Algorithms	143
5.5	Notes and References	143
5.6	Summary	144
5.7	Exercises.	144
	References	144
6	The MapReduce Paradigm	147
6.1	The Challenge of Processing Big Data Fast!	147
6.2	The Basic Principle of MapReduce.	148
6.2.1	Map and Reduce Processes	148
6.2.2	A Historical Perspective: Map and Reduce in Functional Programming Languages	149
6.3	Data Types and the MapReduce Mechanism	150
6.4	A Complete Example of MapReduce in C++.	152
6.5	Launching MapReduce Jobs and Overview of the MapReduce Architecture	154
6.6	Using MapReduce in MPI with the MR-MPI Library	156
6.7	Notes and References	158
6.8	Summary	159
	References	160

Part II High Performance Computing (HPC) for Data Science (DS)

7	Partition-Based Clustering with k-Means.	163
7.1	Exploratory Data Analysis and Clustering	163
7.1.1	Hard Clustering: Partitioning Data Sets	164
7.1.2	Cost Functions and Model-Based Clustering	165
7.2	The k -Means Objective Function	166
7.2.1	Rewriting the k -Means Cost Function for a Dual Interpretation of Clustering: Group Intra-cluster or Separate Inter-cluster Data	170
7.2.2	Complexity and Tractability of the k -Means Optimization Problem	171
7.3	Lloyd's Batched k -Means Local Heuristic	172
7.4	Initializing k -Means with Global Heuristics	175
7.4.1	Random Seed Initialization (as Known as Forgy's Initialization).	175
7.4.2	Global k -Means: The Best Greedy Initialization	175
7.4.3	k -Means++: A Simple Probabilistically-Guaranteed Initialization.	175
7.5	Application of k -Means to Vector Quantization (VQ)	177
7.5.1	Vector Quantization	177
7.5.2	Lloyd's Local Minima and Stable Voronoi Partitions	178
7.6	A Physical Interpretation of k -Means: The Inertia Decomposition.	179
7.7	Choosing k in k -Means: Model Selection	180
7.7.1	Model Selection via the Elbow Method.	180
7.7.2	Model Selection: Explaining Variance Reduction with k .	180
7.8	Parallel k -Means Clustering on a Cluster of Machines.	181
7.9	Evaluating Clustering Partitions	184
7.9.1	The Rand Index	184
7.9.2	Normalized Mutual Information (NMI)	185
7.10	Notes and References	185
7.11	Summary	187
7.12	Exercises.	187
	References	192
8	Hierarchical Clustering	195
8.1	Agglomerative Versus Divisive Hierarchical Clustering, and Dendrogram Representations	195

8.2	Strategies to Define a Good Linkage Distance	198
8.2.1	A Generic Algorithm for Agglomerative Hierarchical Clustering	199
8.2.2	Choosing the Appropriate Elementary Distance Between Elements	200
8.3	Ward Merging Criterion and Centroids	203
8.4	Retrieving Flat Partitions from Dendrograms	204
8.5	Ultra-metric Distances and Phylogenetic Trees	204
8.6	Notes and References	206
8.7	Summary	207
8.8	Exercises	208
	References	211
9	Supervised Learning: Practice and Theory of Classification with the k-NN Rule	213
9.1	Supervised Learning	213
9.2	Nearest Neighbor Classification: NN-Rule	213
9.2.1	Optimizing Euclidean Distance Computation for Nearest Neighbor Queries	214
9.2.2	Nearest Neighbor (NN) Rules and Voronoi Diagrams	215
9.2.3	Enhancing the NN-Rule with the k -NN Rule by Voting!	216
9.3	Evaluating the Performance of Classifiers	217
9.3.1	Misclassification Error Rate	218
9.3.2	Confusion Matrices and True/False Positive/Negative	218
9.4	Precision, Recall and F -Score	219
9.5	Statistical Machine Learning and Bayes' Minimal Error Bound	220
9.5.1	Non-parametric Probability Density Estimation	220
9.5.2	Probability of Error and Bayes' Error	222
9.5.3	Probability of Error for the k -NN Rule	223
9.6	Implementing Nearest Neighbor Queries on a Computer Cluster	224
9.7	Notes and References	224
9.8	Summary	227
9.9	Exercises	227
	References	229

10	Fast Approximate Optimization in High Dimensions with Core-Sets and Fast Dimension Reduction.	231
10.1	Approximate Optimization for Large-Scale Data-Sets	231
10.1.1	An Example that Illustrates the Needs for High Dimensions	231
10.1.2	Some Distance Phenomena in High Dimensions.	232
10.1.3	Core-Sets: From Big Data-Sets to Tiny Data-Sets!	232
10.2	Core-Sets (Coresets, Core sets): Definition	233
10.3	Core-Sets for the Smallest Enclosing Balls	233
10.4	A Simple Iterative Heuristic for Approximating the Smallest Enclosing Ball	235
10.4.1	Convergence Proof	236
10.4.2	Small Enclosing Balls and Good Margin Linear Separators for SVMs.	238
10.5	Core-Sets for k -Means	238
10.6	Fast Dimension Reduction via Random Projection Matrices	238
10.6.1	The Curse of Dimensionality	238
10.6.2	Two Illustrating Examples of High-Dimensional Tasks	239
10.6.3	Linear Dimension Reduction	240
10.6.4	The Johnson–Lindenstrauss’ Theorem	241
10.6.5	Random Projection Matrices	241
10.7	Notes and References	242
10.8	Summary	242
10.9	Exercises.	243
	References	243
11	Parallel Algorithms for Graphs.	245
11.1	Finding Dense(st) Sub-graphs in Large Graphs	245
11.1.1	Problem Statement	245
11.1.2	Densest Sub-graph Complexity and a Simple Greedy Heuristic	246
11.1.3	A Parallel Heuristic for the Densest Sub-graph.	250
11.2	Testing (Sub)graph Isomorphisms.	253
11.2.1	General Principles of Enumerating Algorithms	254
11.2.2	Ullman’s Algorithm to Test the Sub-graph Isomorphism	255
11.2.3	Parallelizing Enumerating Algorithms	257
11.3	Notes and Discussions	257
11.4	Summary	258
11.5	Exercises.	259
	References	259

Appendix A: Written Exam (3 h) 261

**Appendix B: SLURM: A Resource Manager and Job Scheduler
on Clusters of Machines 273**

Index 277

<http://www.springer.com/978-3-319-21902-8>

Introduction to HPC with MPI for Data Science

Nielsen, F.

2016, XXXIII, 282 p. 101 illus. in color., Softcover

ISBN: 978-3-319-21902-8