

Contents

1	Performance and scalability of parallel algorithms	1
1.1	Asymptotics	1
1.2	Performance metrics of sequential algorithms	2
1.3	Performance metrics of parallel algorithms	3
1.3.1	Parallel time $T(n, p)$ and speedup $S(n, p)$	3
1.3.2	Parallel cost $C(n, p)$, work $W(n, p)$, and efficiency $E(n, p)$	5
1.3.3	An example: parallel binary reduction	6
1.3.4	Sources of inefficiency of parallel algorithms and ways to eliminate it . .	7
1.4	The scalability of parallel algorithms	7
1.4.1	Improving the cost and efficiency by reducing p	7
1.4.1.1	ROWSIMPARADD = Row-wise simulation of PARADD	7
1.4.1.2	COLSIMPARADD = Column-wise simulation of PARADD	8
1.4.2	The scalability problem	9
1.4.3	Cost scalability and isoefficiency functions ψ_1, ψ_2	9
1.4.3.1	Time scalability	12
1.4.4	Amdahl's law	14
1.4.5	Gustafson's law	14
2	Parallel architectures and models	15
2.1	Taxonomies of parallel architectures	15
2.1.1	Instruction and data flow taxonomy	16
2.1.1.1	Single Instruction Multiple Data (SIMD) architecture model . .	16
2.1.1.2	Multiple Instruction Multiple Data (MIMD) architecture model	17
2.1.1.3	Comparison of SIMD and MIMD, SPMD style, and data parallel languages	17
2.1.2	Memory organization taxonomy	19
2.1.3	Interconnection network taxonomy	20
2.2	Parallel Random Access Machine model	22
2.2.1	PRAM submodels and the computational power	23
2.2.2	Optimal and fully parallel PRAM algorithms	24
2.2.3	Simulations of large PRAMs on small PRAMs of the same type	25
2.2.3.1	Coarse-grain (cost-preserving) simulation of PRAM algorithms on PRAMs with less processors	25
2.2.3.2	Fine-grain (work-preserving) simulation of PRAM algorithms on PRAMs with less processors and Brent's scheduling principle	25
2.2.3.3	Simulation of large-memory PRAM algorithms on small-memory PRAMs	26
2.2.4	Simulation of stronger PRAMs on weaker ones	26
2.3	Asynchronous PRAM	29
2.3.1	Implementation of barriers	30

2.3.2	Simulations of PRAM algorithms on APRAMs	30
2.3.3	Design of APRAM algorithms	32
3	Interconnection networks	34
3.1	Basic notions and terminology	34
	Letters, alphabets, and strings.	34
	Modulo arithmetics and logarithms.	34
	Graphs and subgraphs.	34
	Node degree and regularity.	35
	Isomorphism and automorphism.	35
	Union and Cartesian product.	35
	Vertex and edge symmetry.	36
	Paths, distances, diameters, and connectivity.	37
	Bipartite and Hamiltonian graphs.	38
	Hierarchical recursivity and scalability of graph topologies. . . .	38
	Digraphs.	38
3.2	Requirements on INs for parallel computers	38
	Small and fixed vertex degree.	38
	Small diameter and small average distance.	38
	Symmetry.	39
	Hierarchical recursivity.	39
	Efficient and incremental scalability.	39
	High connectivity and fault tolerance and small fault diameter or fault average distance.	39
	Bisection width: controversial.	39
	Support for efficient routing and collective communication. . . .	39
	Embeddability of and into other topologies.	39
	Hamiltonicity and 2-colorability.	40
3.3	Direct strictly orthogonal networks	40
3.3.1	Binary hypercube of dimension n , Q_n	40
	Basic structural properties.	40
	Routing and distances.	41
	Hypercube-based multiprocessor machines.	41
3.3.2	n -dimensional mesh of dimension sizes $z_1, z_2, \dots, z_n$, $M(z_1, z_2, \dots, z_n)$. .	42
	Basic structural properties.	42
	Routing and distances.	43
	Mesh-based multiprocessor machines.	43
3.3.3	n -dimensional torus of dimension sizes $z_1, z_2, \dots, z_n$, $K(z_1, z_2, \dots, z_n)$. . .	43
	Basic structural properties.	43
	Routing and distances.	44
	Torus-based multiprocessor machines.	44
3.3.4	Comparisons of hypercubes, meshes, and tori	44
3.4	Direct sparse hypercubic networks	45
3.4.1	Cube-connected cycles of dimension n , CCC_n	45
	Basic structural properties.	45

	Routing and distances.	46
	CCC-based multiprocessors.	46
3.4.2	Wrapped butterfly of dimension n , wBF_n	46
	Basic structural properties.	46
	Routing and distances.	47
3.4.3	Ordinary butterfly of dimension n , oBF_n	47
	Basic structural properties.	47
	Routing and distances.	47
	Butterfly-based multiprocessors.	47
3.5	Indirect (multistage) sparse hypercubic networks	48
3.5.1	Theory of generic binary unidirectional MINs	48
3.5.2	Basic properties of unidirectional binary delta MINs	49
	Deterministic greedy routing in k -D delta binary MINs.	50
3.5.3	Rearrangeable unidirectional MINs	50
3.5.4	Bidirectional delta MINs	51
3.5.5	Comparison of direct hypercubic networks and indirect hypercubic MINs	52
4	Embeddings and simulations of INs	54
4.1	The embedding problem	54
4.1.1	Basic definitions for static embeddings	54
4.1.2	Some properties and lower bounds	56
4.2	Embeddings into the hypercube	57
4.2.1	Embeddings of paths and cycles	58
4.2.2	Static embeddings of trees	59
	4.2.2.1 Complete binary trees	59
	Straightforward embedding based on inorder labelling with $\overline{dil} = 1.5$.	60
	Nearly optimal embedding of CBT_n into Q_{n+1}	61
	4.2.2.2 Other trees	62
4.2.3	Dynamic randomized embeddings of trees	62
4.2.4	Binary n -level Divide&Conquer (D&C) computations on hypercubes . .	63
	Optimal implementation of a pipelined D&C computation on Q_n .	63
	Optimal implementation of a one-wave D&C computation on Q_n .	63
4.2.5	Embeddings of meshes and tori	64
4.2.6	Embeddings of hypercubic networks	65
4.3	Embeddings into meshes and tori	66
4.3.1	Embeddings of 1-D meshes/tori into high-D meshes/tori	66
4.3.2	Embeddings of hypercubes into meshes/tori	67
4.3.3	Embeddings of 2-D tori into 1-D tori	68
4.3.4	Embeddings of 2-D meshes into 2-D meshes	68
4.3.5	Binary n -level D&C computations on 2-D meshes	70
	An efficient implementation of a one-wave D&C computation on 2-D square meshes.	71
	An implementation of the pipelined D&C computation on 2-D square meshes.	71
4.4	Embeddings of linear arrays/cycles into any network	72
4.5	Embeddings into hypercubic topologies	73

4.5.0.1	Normal hypercube algorithms	74
5	Routing algorithms and switching techniques	76
5.1	Architecture of the communication subsystem	76
5.1.1	Computing node and router	76
5.1.2	Units of communication	77
5.1.3	The design space of routing algorithms	77
5.1.3.1	Routing decisions: when and where made	78
5.1.3.2	Adaptiveness	78
5.1.3.3	Minimality	78
5.1.3.4	Progressiveness	78
5.1.4	Implementation of the routing algorithm	79
5.2	Performance metrics	79
5.3	Basic switching techniques	80
5.3.1	Circuit switching (CS)	80
	Communication latency.	81
5.3.2	Store-and-forward (SF) switching	81
	Communication latency.	81
5.3.3	Virtual cut-through (VCT) switching	81
	Communication latency.	82
5.3.4	Wormhole (WH) switching	82
	Communication latency.	83
5.3.5	Simplified formulae for the communication latency	84
	Distance sensitive switching.	84
	Distance insensitive switching.	84
5.4	Why is distance-insensitive switching important?	84
5.5	Deadlocks	84
5.5.1	How to solve the deadlock problem?	85
5.5.2	Channel dependence graph	85
5.5.3	Deadlock avoidance by routing restrictions	86
5.5.3.1	Dimension-ordered routing	86
5.5.3.2	$Up^*/Down^*$ routing algorithms in irregular networks	87
5.5.4	Deadlock avoidance based on virtual channels	88
5.5.4.1	Deadlock avoidance in tori	89
6	Prefix computations and Eulerian tour techniques	92
6.1	Parallel reduction algorithms	92
6.2	Parallel prefix sum (PPS) algorithms	93
6.2.1	PPS on EREW PRAMs	93
6.2.2	PPS on trees	94
	Generic PPS on indirect trees.	94
	Generic PPS on direct trees.	94
	PPS on an arbitrary topology	95
6.2.3	PPS on hypercubes	95
6.2.4	PPS on meshes	96
	PPS on SF meshes.	96

	PPS on WH meshes.	96
6.3	Scalability of the PPS	96
6.4	Applications of the PPS	98
	Packing problem.	98
	RadixSort.	98
	Carry-look-ahead parallel binary adder.	98
	Tridiagonal system of equations.	99
6.5	Segmented PPS (prefix sum) (SPPS)	100
6.5.1	EREW parallel QuickSort	101
6.6	Prefix computations on linked lists	102
6.6.1	List ranking and PPS on linked lists	102
6.6.2	The problem of scalability of pointer jumping	103
6.6.3	The idea of a scalable list ranking	104
6.7	Eulerian tour techniques (ETT) in trees	104
	Eulerian graphs.	104
	Adjacency list representation.	104
6.7.1	Parallel construction of Eulerian tours and paths in spanning trees	105
	Eulerian tour construction and the labyrinth rule.	105
	Transformation of an Eulerian tour into an Eulerian path.	106
	Eulerian path ranking.	106
6.7.2	Solving graph problems using the ETT	106
6.7.2.1	Computing all parents in parallel	106
6.7.2.2	Computing all subtree sizes in parallel	107
6.7.2.3	Computing levels of all vertices in parallel	108
6.7.2.4	Preorder numbering	108
7	Parallel sorting algorithms	110
7.1	Introduction to comparison-based parallel sorting	110
7.2	Indirect sorting networks	110
7.3	Direct sorting networks	111
7.3.1	Scalability of sorting algorithms	112
7.4	A naive EREW PRAM(N, p) sorting algorithm	113
7.5	Oblivious sorting and 0-1 sorting theorem	113
7.6	Sorting on mesh networks	115
7.6.1	Even-odd transposition sorting (parallel bubble-sort) on 1-D meshes	115
	Scalability of EOTSORT on $M(p)$	115
7.6.2	SHEARSORT on 2-D meshes	115
	Scalability of SHEARSORT on $M(\sqrt{p}, \sqrt{p})$	116
7.6.3	The lower bound on 2-D mesh snake-like sorting	117
7.6.4	Lexicographic sorting on 3-D mesh $M(n, n, n)$	117
	Scalability of the 3DSORT algorithm on $M(\sqrt[3]{p}, \sqrt[3]{p}, \sqrt[3]{p})$	118
7.7	Sorting on hypercubic networks	119
7.7.1	EVENODDMERGESORT (EOMS)	119
7.7.2	EVENEVENMERGESORT (EEMS)	121
7.7.3	BITONICMERGESORT (BMS)	122

7.7.4	Implementation of MergeSort algorithms on hypercubic networks	124
7.7.4.1	Implementation of the EOMS algorithm on a butterfly network	124
7.7.4.2	Implementation of the EOMS algorithm on a hypercube . . .	125
7.7.4.3	Implementation of the BMS algorithm on a butterfly and hypercube	126
7.7.5	Implementation of the BMS algorithm on PRAMs and 2-D meshes . . .	126
	Scalability of the MESHBMS algorithm.	128
	Complexity and scalability of the 3DSORT algorithm revised. . .	128
8	Permutations in mesh-based and hypercubic networks	129
8.1	Permutation routing in meshes	129
8.1.1	Greedy permutation routing in 1-D SF meshes	129
8.1.2	Greedy permutation routing in SF 2-D meshes	130
8.1.3	Higher-dimensional meshes	131
8.1.4	Time-memory trade-off problem for permutation routing in meshes . . .	132
8.1.5	Randomized permutation routing	132
8.1.6	Sorting-based permutation routing in meshes	133
8.1.7	Off-line permutation routing in meshes	134
8.2	Permutation routing in hypercubic MINs	136
8.2.1	Greedy permutation routing in butterflies	137
8.2.1.1	Permutations with the worst-case greedy routing	137
8.2.1.2	Time complexity of the worst-case greedy permutation routing	138
8.2.1.3	Permutations with optimal greedy routing in butterflies	139
	Packing, spreading, and monotone permutations.	139
	Spreading permutations.	140
	Monotone permutations.	140
	Translation and complement permutations.	141
	Permutation routing in a hypercube.	141
8.2.2	Improving the time complexity of the worst-case permutations in hypercubic networks	142
8.2.3	Sorting-based permutation routing in hypercubic networks	142
8.2.4	Off-line permutation routing in Benes networks	142
8.2.5	Applications of the Benes off-line routing algorithms	144
	Indirect butterflies.	144
	Direct butterflies.	144
	Simulations on hypercubes.	146
9	Communication algorithms	147
9.1	Taxonomy of communication problems and models	147
	Basic collective communication operations (CCOs).	147
	Communication models and metrics.	147
	An example of calculations of lower bounds for CCOs.	148
9.2	One-to-all broadcast (OAB)	149
9.2.1	OAB in SF networks	149
9.2.1.1	Lower and upper bounds	149
	Lower bounds.	149

- Upper bounds in general all-output-port networks. 149
- Upper bounds in general 1-port networks. 149
- 9.2.1.2 Optimal OAB in EREW PRAM, 1-port cliques, and hypercubes: RD 150
 - RD in hypercubes. 150
- 9.2.1.3 Dimension-ordered trees in meshes 150
 - All-output-port meshes. 150
 - 1-port meshes. 151
- 9.2.1.4 Dimension ordered trees in tori 152
- 9.2.2 OAB in WH networks 152
 - 9.2.2.1 Lower and upper bounds 152
 - 9.2.2.2 1-port hypercubes 152
 - 9.2.2.3 All-output-port hypercubes 152
 - Double tree algorithm. 153
 - Ho-Kao algorithm. 153
 - 9.2.2.4 1-port tori. 154
 - RD in 1-D tori. 154
 - RD in 1-D meshes. 155
 - Dimension-ordered RD in higher-D meshes and tori. 155
 - 9.2.2.5 All-output-port tori. 155
 - An optimal OAB algorithm in 2-port WH 1-D tori. 156
 - Diagonal-based OAB algorithm (DIAGMESH OAB) in 4-port WH 2-D tori. 156
- 9.3 Multicast 157
 - 9.3.1 ADOC-based MC algorithm in 1-port WH hypercubes 157
 - 9.3.2 1-port WH 2-D meshes 158
- 9.4 One-to-all scatter (OAS) 159
 - 9.4.1 Noncombining OAS 159
 - 9.4.1.1 Lower bounds 159
 - 9.4.1.2 1-port SF networks 160
 - 9.4.1.3 1-port WH networks 160
 - 9.4.1.4 All-port SF networks 160
 - All-port meshes or tori. 160
 - All-port hypercubes. 161
 - 9.4.1.5 All-port WH networks 161
 - 9.4.2 Combining OAS 161
 - Lower bounds. 162
 - SF 1-D meshes and tori. 162
 - WH meshes and tori. 162
 - Hypercubes. 163
- 9.5 All-to-all broadcast (AAB) 163
 - 9.5.1 SF combining AAB 163
 - 9.5.1.1 All-port full-duplex networks 163
 - Lower and upper bounds. 164
 - 9.5.1.2 All-port half-duplex networks 164

	Better SF combining AAB algorithms in bipartite networks. . . .	164
	Better SF combining AAB algorithms in strongly orientable net- works.	164
9.5.1.3	1-port SF networks	166
	Lower and upper bounds.	166
	1-D full-duplex SF meshes and tori.	166
	Higher-D meshes or tori.	167
	Hypercubes.	168
9.5.2	SF noncombining AAB in all-port full-duplex model	168
9.5.2.1	Lower bounds	168
9.5.2.2	$M(z_1, z_2)$	168
9.5.2.3	Vertex-symmetric topologies	169
	2-D and 3-D tori.	169
	Edge-disjoint Hamiltonian cycles.	169
	Hypercubes.	170
9.5.3	WH combining AAB	171
9.6	All-to-all scatter (AAS)	171
9.6.1	Lower bounds	172
9.6.2	SF combining AAS	172
9.6.2.1	Hypercubes	172
9.6.2.2	1-port half-duplex 1-D torus	173
9.6.2.3	2-port full-duplex 1-D torus	173
9.6.2.4	1-port or 2-port higher-D meshes/tori	173
9.6.3	WH combining AAS	174
9.6.3.1	Hypercubes	174
9.6.3.2	Meshes/tori	174
	The BINEXCHAAS algorithm.	174
	The QUADEXCHAAS algorithm.	175
9.6.4	WH noncombining AAS	175
9.6.4.1	Hypercubes	175
9.6.4.2	2-D meshes/tori with XY routing	176
10	Parallel algorithms for linear algebra	177
10.1	Basic definitions	177
10.2	Basic matrix mapping schemes	177
10.2.1	Striped mapping	177
10.2.2	Checkerboard mapping	178
10.3	Matrix transposition	178
10.3.1	Striped mapped matrix	178
10.3.2	Checkerboard mapped matrix	179
10.3.2.1	SF 2-D mesh	179
10.3.3	WH 2-D mesh	180
10.3.4	Hypercube	180
10.4	Matrix-vector multiplication (MVM)	181
10.4.1	Row-wise mapped matrix	181

- 10.4.2 Column-wise mapped matrix 181
- 10.4.3 Checkerboard mapped matrix 182
- 10.5 Matrix-matrix multiplication $\mathcal{C} = \mathcal{A}\mathcal{B}$ 183
 - 10.5.1 A naive algorithm 183
 - 10.5.2 A less naive algorithm 183
 - 10.5.3 Cannon’s algorithm 184
 - 10.5.4 Fox’s algorithm Broadcast-Multiply-Roll (BMR) 185
- 10.6 Parallel algorithms for systems of linear equations 186
 - 10.6.1 Triangular matrices and back substitution 186
 - 10.6.2 Tridiagonal systems and even-odd reduction 187
 - 10.6.3 Parallel direct methods for solving dense SLE 188
 - 10.6.3.1 LU decomposition 189
 - 10.6.3.2 Optimal mapping for the parallel LU decomposition. 189
 - 10.6.3.3 Implementation of the parallel LU decomposition 190
 - 10.6.4 Iterative methods for solving SLEs 190
 - 10.6.4.1 Parallel Jacobi method 191
 - 10.6.4.2 Parallel Gauss-Seidel method 192
 - 10.6.4.3 Partial differential equations and their discretizations 192
 - Parallel Jacobi algorithm. 193
 - Parallel Gauss-Seidel algorithm. 193
 - Highly parallel Gauss-Seidel algorithm. 194
 - Other discretizations and their parallel solutions. 195