

CONTENTS

Preface	ix
1 Introduction	1
1.1 From networked systems to distributed systems	3
1.1.1 Distributed versus decentralized systems	3
1.1.2 Why making the distinction is relevant	7
1.1.3 Studying distributed systems	8
1.2 Design goals	10
1.2.1 Resource sharing	10
1.2.2 Distribution transparency	11
1.2.3 Openness	15
1.2.4 Dependability	18
1.2.5 Security	21
1.2.6 Scalability	24
1.3 A simple classification of distributed systems	32
1.3.1 High-performance distributed computing	32
1.3.2 Distributed information systems	37
1.3.3 Pervasive systems	43
1.4 Pitfalls	52
1.5 Summary	53
2 Architectures	55
2.1 Architectural styles	56
2.1.1 Layered architectures	57
2.1.2 Service-oriented architectures	62
2.1.3 Publish-subscribe architectures	68
2.2 Middleware and distributed systems	73
2.2.1 Middleware organization	74
2.2.2 Modifiable middleware	78
2.3 Layered-system architectures	78
2.3.1 Simple client-server architecture	79
2.3.2 Multitiered Architectures	80
2.3.3 Example: The Network File System	83
2.3.4 Example: The Web	85
2.4 Symmetrically distributed system architectures	88
2.4.1 Structured peer-to-peer systems	90
2.4.2 Unstructured peer-to-peer systems	92

2.4.3	Hierarchically organized peer-to-peer networks	95
2.4.4	Example: BitTorrent	96
2.5	Hybrid system architectures	98
2.5.1	Cloud computing	98
2.5.2	The edge-cloud architecture	100
2.5.3	Blockchain architectures	104
2.6	Summary	108
3	Processes	111
3.1	Threads	112
3.1.1	Introduction to threads	113
3.1.2	Threads in distributed systems	122
3.2	Virtualization	127
3.2.1	Principle of virtualization	127
3.2.2	Containers	133
3.2.3	Comparing virtual machines and containers	138
3.2.4	Application of virtual machines to distributed systems	139
3.3	Clients	141
3.3.1	Networked user interfaces	141
3.3.2	Virtual desktop environment	144
3.3.3	Client-side software for distribution transparency	148
3.4	Servers	149
3.4.1	General design issues	149
3.4.2	Object servers	154
3.4.3	Example: The Apache Web server	159
3.4.4	Server clusters	161
3.5	Code migration	167
3.5.1	Reasons for migrating code	167
3.5.2	Models for code migration	171
3.5.3	Migration in heterogeneous systems	174
3.6	Summary	177
4	Communication	181
4.1	Foundations	183
4.1.1	Layered Protocols	183
4.1.2	Types of Communication	190
4.2	Remote procedure call	192
4.2.1	Basic RPC operation	192
4.2.2	Parameter passing	197
4.2.3	RPC-based application support	201
4.2.4	Variations on RPC	205
4.3	Message-oriented communication	208
4.3.1	Simple transient messaging with sockets	208
4.3.2	Advanced transient messaging	213

4.3.3	Message-oriented persistent communication	220
4.3.4	Example: Advanced Message Queuing Protocol (AMQP)	227
4.4	Multicast communication	232
4.4.1	Application-level tree-based multicasting	232
4.4.2	Flooding-based multicasting	236
4.4.3	Gossip-based data dissemination	240
4.5	Summary	245
5	Coordination	247
5.1	Clock synchronization	249
5.1.1	Physical clocks	250
5.1.2	Clock synchronization algorithms	253
5.2	Logical clocks	260
5.2.1	Lamport's logical clocks	260
5.2.2	Vector clocks	266
5.3	Mutual exclusion	272
5.3.1	Overview	272
5.3.2	A centralized algorithm	273
5.3.3	A distributed algorithm	274
5.3.4	A token-ring algorithm	276
5.3.5	A decentralized algorithm	277
5.3.6	Example: Simple locking with ZooKeeper	280
5.4	Election algorithms	283
5.4.1	The bully algorithm	283
5.4.2	A ring algorithm	285
5.4.3	Example: Leader election in ZooKeeper	286
5.4.4	Example: Leader election in Raft	289
5.4.5	Elections in large-scale systems	290
5.4.6	Elections in wireless environments	294
5.5	Gossip-based coordination	297
5.5.1	Aggregation	297
5.5.2	A peer-sampling service	298
5.5.3	Gossip-based overlay construction	299
5.5.4	Secure gossiping	303
5.6	Distributed event matching	306
5.6.1	Centralized implementations	307
5.6.2	Secure publish-subscribe solutions	313
5.7	Location systems	315
5.7.1	GPS: Global Positioning System	315
5.7.2	When GPS is not an option	317
5.7.3	Logical positioning of nodes	318
5.8	Summary	322
6	Naming	325

6.1	Names, identifiers, and addresses	326
6.2	Flat naming	329
6.2.1	Simple solutions	329
6.2.2	Home-based approaches	331
6.2.3	Distributed hash tables	333
6.2.4	Hierarchical approaches	338
6.2.5	Secure flat naming	343
6.3	Structured naming	344
6.3.1	Name spaces	344
6.3.2	Name resolution	347
6.3.3	The implementation of a name space	352
6.3.4	Example: The Domain Name System	359
6.3.5	Example: The Network File System	369
6.4	Attribute-based naming	375
6.4.1	Directory services	375
6.4.2	Hierarchical implementations: LDAP	376
6.4.3	Decentralized implementations	380
6.5	Named-data networking	385
6.5.1	Basics	385
6.5.2	Routing	387
6.5.3	Security in named-data networking	388
6.6	Summary	389
7	Consistency and replication	391
7.1	Introduction	392
7.1.1	Reasons for replication	393
7.1.2	Replication as scaling technique	394
7.2	Data-centric consistency models	395
7.2.1	Consistent ordering of operations	396
7.2.2	Eventual consistency	406
7.2.3	Continuous consistency	410
7.3	Client-centric consistency models	415
7.3.1	Monotonic reads	417
7.3.2	Monotonic writes	418
7.3.3	Read your writes	420
7.3.4	Writes follow reads	421
7.3.5	Example: client-centric consistency in ZooKeeper	422
7.4	Replica management	423
7.4.1	Finding the best server location	424
7.4.2	Content replication and placement	426
7.4.3	Content distribution	430
7.4.4	Managing replicated objects	434
7.5	Consistency protocols	437
7.5.1	Sequential consistency: Primary-based protocols	438

7.5.2	Sequential consistency: Replicated-write protocols . . .	440
7.5.3	Cache-coherence protocols	443
7.5.4	Implementing continuous consistency	446
7.5.5	Implementing client-centric consistency	448
7.6	Example: Caching and replication in the Web	451
7.7	Summary	458
8	Fault tolerance	461
8.1	Introduction to fault tolerance	462
8.1.1	Basic concepts	463
8.1.2	Failure models	466
8.1.3	Failure masking by redundancy	470
8.2	Process resilience	471
8.2.1	Resilience by process groups	472
8.2.2	Failure masking and replication	474
8.2.3	Consensus in faulty systems with crash failures	475
8.2.4	Example: Paxos	479
8.2.5	Consensus in faulty systems with arbitrary failures	491
8.2.6	Consensus in blockchain systems	502
8.2.7	Some limitations on realizing fault tolerance	503
8.2.8	Failure detection	506
8.3	Reliable client-server communication	508
8.3.1	Point-to-point communication	508
8.3.2	RPC semantics in the presence of failures	509
8.4	Reliable group communication	515
8.4.1	Introduction	515
8.4.2	Scalability in reliable multicasting	518
8.4.3	Atomic multicast	522
8.5	Distributed commit	528
8.6	Recovery	536
8.6.1	Introduction	536
8.6.2	Checkpointing	538
8.6.3	Message logging	541
8.7	Summary	543
9	Security	545
9.1	Introduction to security	546
9.1.1	Security threats, policies, and mechanisms	547
9.1.2	Design issues	548
9.2	Cryptography	555
9.2.1	Basics	555
9.2.2	Symmetric and asymmetric cryptosystems	557
9.2.3	Hash functions	560
9.2.4	Key management	562

9.3	Authentication	571
9.3.1	Introduction to authentication	571
9.3.2	Authentication protocols	572
9.4	Trust in distributed systems	585
9.4.1	Trust in the face of Byzantine failures	586
9.4.2	Trusting an identity	586
9.4.3	Trusting a system	591
9.5	Authorization	593
9.5.1	General issues in access control	593
9.5.2	Attribute-based access control	598
9.5.3	Delegation	601
9.5.4	Decentralized authorization: an example	605
9.6	Monitoring	609
9.6.1	Firewalls	609
9.6.2	Intrusion detection: basics	611
9.6.3	Collaborative intrusion detection	612
9.7	Summary	613
	Index	615
	Bibliography	631
	Glossary	665