

Designing Data-intensive Applications

*The Big Ideas Behind Reliable, Scalable,
and Maintainable Systems*

Martin Kleppmann

Table of Contents

Preface	xiii
Part I. Foundations of Data Systems	
1. Reliable, Scalable, and Maintainable Applications	3
Thinking About Data Systems	4
Reliability	6
Hardware Faults	7
Software Errors	8
Human Errors	9
How Important Is Reliability?	10
Scalability	10
Describing Load	11
Describing Performance	13
Approaches for Coping with Load	17
Maintainability	18
Operability: Making Life Easy for Operations	19
Simplicity: Managing Complexity	20
Evolvability: Making Change Easy	21
Summary	22
2. Data Models and Query Languages	27
Relational Model Versus Document Model	28
The Birth of NoSQL	29
The Object-Relational Mismatch	29
Many-to-One and Many-to-Many Relationships	33
Are Document Databases Repeating History?	36

Relational Versus Document Databases Today	38
Query Languages for Data	42
Declarative Queries on the Web	44
MapReduce Querying	46
Graph-Like Data Models	49
Property Graphs	50
The Cypher Query Language	52
Graph Queries in SQL	53
Triple-Stores and SPARQL	55
The Foundation: Datalog	60
Summary	63
3. Storage and Retrieval	69
Data Structures That Power Your Database	70
Hash Indexes	72
SSTables and LSM-Trees	76
B-Trees	79
Comparing B-Trees and LSM-Trees	83
Other Indexing Structures	85
Transaction Processing or Analytics?	90
Data Warehousing	91
Stars and Snowflakes: Schemas for Analytics	93
Column-Oriented Storage	95
Column Compression	97
Sort Order in Column Storage	99
Writing to Column-Oriented Storage	101
Aggregation: Data Cubes and Materialized Views	101
Summary	103
4. Encoding and Evolution	111
Formats for Encoding Data	112
Language-Specific Formats	113
JSON, XML, and Binary Variants	114
Thrift and Protocol Buffers	117
Avro	122
The Merits of Schemas	127
Modes of Dataflow	128
Dataflow Through Databases	129
Dataflow Through Services: REST and RPC	131
Message-Passing Dataflow	136
Summary	139

Part II. Distributed Data

5. Replication	151
Leaders and Followers	152
Synchronous Versus Asynchronous Replication	153
Setting Up New Followers	155
Handling Node Outages	156
Implementation of Replication Logs	158
Problems with Replication Lag	161
Reading Your Own Writes	162
Monotonic Reads	164
Consistent Prefix Reads	165
Solutions for Replication Lag	167
Multi-Leader Replication	168
Use Cases for Multi-Leader Replication	168
Handling Write Conflicts	171
Multi-Leader Replication Topologies	175
Leaderless Replication	177
Writing to the Database When a Node Is Down	177
Limitations of Quorum Consistency	181
Sloppy Quorums and Hinted Handoff	183
Detecting Concurrent Writes	184
Summary	192
6. Partitioning	199
Partitioning and Replication	200
Partitioning of Key-Value Data	201
Partitioning by Key Range	202
Partitioning by Hash of Key	203
Skewed Workloads and Relieving Hot Spots	205
Partitioning and Secondary Indexes	206
Partitioning Secondary Indexes by Document	206
Partitioning Secondary Indexes by Term	208
Rebalancing Partitions	209
Strategies for Rebalancing	210
Operations: Automatic or Manual Rebalancing	213
Request Routing	214
Parallel Query Execution	216
Summary	216
7. Transactions	221
The Slippery Concept of a Transaction	222

The Meaning of ACID	223
Single-Object and Multi-Object Operations	228
Weak Isolation Levels	233
Read Committed	234
Snapshot Isolation and Repeatable Read	237
Preventing Lost Updates	242
Write Skew and Phantoms	246
Serializability	251
Actual Serial Execution	252
Two-Phase Locking (2PL)	257
Serializable Snapshot Isolation (SSI)	261
Summary	266
8. The Trouble with Distributed Systems	273
Faults and Partial Failures	274
Cloud Computing and Supercomputing	275
Unreliable Networks	277
Network Faults in Practice	279
Detecting Faults	280
Timeouts and Unbounded Delays	281
Synchronous Versus Asynchronous Networks	284
Unreliable Clocks	287
Monotonic Versus Time-of-Day Clocks	288
Clock Synchronization and Accuracy	289
Relying on Synchronized Clocks	291
Process Pauses	295
Knowledge, Truth, and Lies	300
The Truth Is Defined by the Majority	300
Byzantine Faults	304
System Model and Reality	306
Summary	310
9. Consistency and Consensus	321
Consistency Guarantees	322
Linearizability	324
What Makes a System Linearizable?	325
Relying on Linearizability	330
Implementing Linearizable Systems	332
The Cost of Linearizability	335
Ordering Guarantees	339
Ordering and Causality	339
Sequence Number Ordering	343

Total Order Broadcast	348
Distributed Transactions and Consensus	352
Atomic Commit and Two-Phase Commit (2PC)	354
Distributed Transactions in Practice	360
Fault-Tolerant Consensus	364
Membership and Coordination Services	370
Summary	373

Part III. Derived Data

10. Batch Processing	389
Batch Processing with Unix Tools	391
Simple Log Analysis	391
The Unix Philosophy	394
MapReduce and Distributed Filesystems	397
MapReduce Job Execution	399
Reduce-Side Joins and Grouping	403
Map-Side Joins	408
The Output of Batch Workflows	411
Comparing Hadoop to Distributed Databases	414
Beyond MapReduce	419
Materialization of Intermediate State	419
Graphs and Iterative Processing	424
High-Level APIs and Languages	426
Summary	429
11. Stream Processing	439
Transmitting Event Streams	440
Messaging Systems	441
Partitioned Logs	446
Databases and Streams	451
Keeping Systems in Sync	452
Change Data Capture	454
Event Sourcing	457
State, Streams, and Immutability	459
Processing Streams	464
Uses of Stream Processing	465
Reasoning About Time	468
Stream Joins	472
Fault Tolerance	476
Summary	479

12. The Future of Data Systems	489
Data Integration	490
Combining Specialized Tools by Deriving Data	490
Batch and Stream Processing	494
Unbundling Databases	499
Composing Data Storage Technologies	499
Designing Applications Around Dataflow	504
Observing Derived State	509
Aiming for Correctness	515
The End-to-End Argument for Databases	516
Enforcing Constraints	521
Timeliness and Integrity	524
Trust, but Verify	528
Doing the Right Thing	533
Predictive Analytics	533
Privacy and Tracking	536
Summary	543
Glossary	553
Index	559