Index

A

ABA problem, 225
abort, 79
acceptors, 286
access methods, 10
access patterns, 15
accuracy, 196
ACID properties, 79, 94, 97, 218
acknowledgments, 183
admission queue, 88
after-images, 91
anti-entropy and dissemination
    bitmap version vectors, 248
    digest reads, 246
    gossip dissemination, 250-255
    hinted handoff, 246
    Merkle trees, 247
    overview of, 243, 255
    read repair, 245
Apache Cassandra
    conflict resolution in, 233
    hinted writes in, 246
    range tombstones in, 137
    time window compaction in, 143
Apache Kudu, 14
Apache ORC, 14
Apache Parquet, 14
Apache Zookeeper, 283
append-only structures, 22, 129, 130
arbitrary faults, 193
ARIES, 92
arrays, 30
asynchronous communication, 187, 245
asynchronous systems, 190, 197
at-least-once delivery, 186
at-most-once delivery, 186
atomic broadcast, 281-285
atomic commitment, 258
atomic registers, 220
atomicity, 79, 218, 225
auto-sharding, 271
availability, 216
availability lists, 57

B

B-Trees, basics of
    B-Tree hierarchy, 35
    B-Tree lookup algorithm, 38
    B-Tree lookup complexity, 37
    B-Tree node merges, 41
    B-Tree node splits, 39
    counting keys, 38
    point and range queries using, 35
    separator keys in, 36
    similarities between BSTs and B-Trees, 33
B-Trees, implementing
    binary search, 67-67
    compression, 73-74
    page headers, 61-67
    rebalancing, 70
    right-only appends, 71-72
    vacuum and maintenance, 74-76
B-Trees, variants of
    abstracting node updates, 113
    Bw-Trees, 120-124, 165
    cache-oblivious B-Trees, 124
    copy-on-write B-Trees, 112, 165
    FD-Trees, 117-120, 165

lazy B-Trees, 114-117
B+-Trees, 36
backoff strategy, 181
backpressure, 175
backup copies, 11
backward links, 62
backward-oriented validation, 98
balanced search trees, 27
balancing operations, 35
bandwidth, 175
base nodes, 121
before-images, 91
benchmarking tools, 4
BerkeleyDB, 2
best effort availability and consistency, 217
best effort broadcast, 280
best fit strategy, 57
big-endian byte order, 47
BigTable, 15, 272
binary encoding
Booleans, enums, and flags, 49
page layouts, 47
primitive types, 47
strings and variable-size data, 49
versioning, 58
binary large objects (BLOBs), 53
binary search algorithm, 67
binary search trees (BSTs)
balancing, 27
similarities to B-Trees, 33
structure of, 26
trees for disk-based storage, 28
bit rot, 181
Bitcask, 153, 165
bitmap version vectors, 248
bitmasks, 50
Blink-Trees, 64, 107
block device, 31
block padding, 73
blocking atomic commitment, 263
blocking operations, 245
blocks, 21, 30, 52
Bloom filters, 146
Booleans, 49
branches, 36
breadcrumbs, 69
bridges, 118
broadcasting, 243, 280
buffer management
cache eviction, 83
caching semantics, 83
circular buffers, 85, 86
database-specific, 82
locking pages in cache, 84
overview of, 81
page replacement, 85-88
buffer manager, 10
buffer pools, 81
buffering, 21, 130, 165 (see also buffer management)
bulk-loading, 72
bully algorithm, 207
busy-wait algorithms, 104
Bw-Trees (Buzzword-Trees)
buffering, immutability, and ordering, 165
compare-and-swap, 121
consolidation and garbage collection, 123
overview of, 120
structural modification operations, 122
update chains, 121
byte-order, 47
Byzantine consensus, 305-309
Byzantine faults, 193

C

C-Store, 13
cache-aware structures, 125
cache-oblivious B-Trees, 124
caching, 12, 29, 81-85
Calvin, 266
candidate/ordinary optimization, 209
candidates, 86
CAP (consistency, availability, and partition tolerance), 216-219
cascading failures, 180
causal consistency, 229-233
cell offsets, 55
cells
cell layout, 54
organizing into pages, 56
checkpointing, 12, 83, 90, 309
checksumming, 59
child offsets, 38
churn, 254
circuit breakers, 180
circular buffers, 85, 86
clean pages, 115
ClickHouse, 14

client centric consistency model, 233
client/server model, 8
CLOCK algorithm, 86
clocks, 168, 176, 232
cluster-wide metadata, 243
clustered files, 19
clustered indexes, 7, 19
CockroachDB, 268
code examples, obtaining and using, xviii
cohorts, 260
cold storage, 7
collisions, 293
column families, 15
column keys, 16
column stores, 7
column-oriented data layout, 12-15
columns, 12
comments and questions, xviii
commit
  atomic commitment, 259
  three-phase commits, 264-266
  two-phase commits (2PCs), 259-264
commit log, 88
commit operation, 79
committed version, 99
common ground, 305
common knowledge, 187
compaction
  defined, 76
  leveled compaction, 142
  process of, 134
  size-tiered compaction, 143
  time window compaction, 143
  tombstones and, 141
compare-and-swap instructions, 104, 121, 225
compensation log records (CLR), 90
complete schedules, 94
completeness, 196
completion events, 219
complexity, 178
compression, 73, 151
concurrency control
  categories of, 93
  concurrency in LSM trees, 155
  in distributed systems, 171-174
  interference between operations, 103
  isolation levels, 96
  lock-based concurrency control, 100-108
  multiversion concurrency control (MVCC), 99
  optimistic concurrency control (OCC), 98
  pessimistic (conservative) concurrency control (PCC), 99
  transaction isolation, 95
concurrent computing, 173
concurrent operations, 219
conflict resolution, 140, 177, 232
Conflict-Free Replicated Data Types (CRDTs), 238
consensus
  Byzantine consensus, 305-309
  linearizability using, 226
  overview of, 279
  Paxos, 285-300
  Raft, 300-305
conservative 2PL, 101
conservative concurrency control, 93
consistency
  consistency guarantees, 79
  convergent consistency, 251
  defined, 218
  eventual consistency, 234
  external consistency, 270
  linearizable consistency, 216
  operation consistency, 222
  overview of, 215
  state consistency, 177
  tunable consistency, 235
consistency models
  causal consistency, 229-233
  client centric consistency model, 233
  linearizability, 223-227
  monotonic reads/writes models, 233
  overview of, 222
  read-own-writes consistency model, 233
  role of, 172, 215
  sequential consistency, 227
  strict consistency, 223
  writes-follow-reads model, 234
consistent hashing, 271
convergent consistency, 251
cooperation, 169
cooperative broadcasts, 243
coordination, 169
coordination avoidance, 275
coordinators, 205, 260
copy-on-write, 22

copy-on-write B-Trees, 112, 165
correct schedules, 94
CP systems, 217
crash faults, 191
crash-recovery, 192
cursors, 137
cyclic redundancy checks (CRCs), 59

D

data
  bulk-loading, 72
  compressing, 73, 151
  deserialization of, 48
  disk-based structures for, 29-32
  file formats, 45-60
  preventing loss of, 11, 59
  serialization of, 48
  variable-size data, 49, 55, 57
data entries, 18
data files, 17-18
data logs, 91
data records
  direct versus primary key references, 20
  ordering of, 22
  storing and retrieving, 17
data replication, 215
database management system (DBMS), 1
  (see also database systems)
database partitioning, 270-271
database systems
  architecture of, 1, 8-10
  buffering, immutability, and ordering in, 21, 165
  column- versus row-oriented, 12-16
  comparing, 2-6
  data files and index files, 17-21
  data storage and distribution, xvi
  memory- versus disk-based, 10-12
  performance and scalability of, 313
  prerequisites to learning about, xiv
  purpose of, 1
  terminology used in, 7
  topics covered, xv
dead processes, 195
deadlocks, 101
decision state, 189
decoupling, 176
deduplication, 185
defragmentation, 76
delete entry, 137
  (see also tombstones)
deletion markers, 18, 75
  (see also tombstones)
delta nodes, 121
density, 131
deserialization, 48
dies, 30
digest reads, 246
direct coordination, 248
dirty pages, 81, 83, 90
dirty reads, 95
dirty writes, 96
disk-based data structures
  hard disk drives, 30
  on-disk structures, 32
  solid state drives (SSDs), 30, 157, 161
  uses for, 29
disk-based DBMS
  durability in, 12
  versus memory-based, 10
  trees for disk-based storage, 28
disk-resident tables, 132
dissemination, 169, 250-255
distinguished learners, 287
distinguished proposers, 291
distributed algorithms, 168
distributed systems
  abstractions, 181
  benefits of, 167
  challenges of, 168
  concurrent execution, 171-174
  failure models, 191-193
  fallacies of, 174-181
  FLP Impossibility Problem, 189
  versus single-node systems, 171
  system synchrony, 190
  terminology used in, 168
  Two Generals' Problem, 187
distributed transactions
  coordination avoidance, 275
  making operations appear atomic, 258
  overview of, 257
  three-phase commits, 264-266
  two-phase commits (2PCs), 259-264
  with Calvin, 266
  with Percolator, 272-274
  with Spanner, 268
divider cells, 36

(see also separator keys)
dormant certificates, 137
(see also tombstones)
dots, 248
dotted causal containers, 249
durability, 11, 80, 83

E

efficiency, 196
Egalitarian Paxos (EPaxos), 293
election timeouts, 302
entropy, 244
enumerated types (enums), 49
epoch-based reclamation, 124
epochs, 266, 283
erase blocks, 31
eventual consistency, xiii, 234
eviction, 81, 83, 85
eviction policy, 85
exactly-once delivery, 186
execution engine, 10
execution plan, 9
expanding phase, 100
exponent, 48
external consistency, 270

F

fadvise flag, 82
failed processes, 195, 219
failure detection
- challenges of, 195
- failure sites, 195
- gossip and, 200
- group failures, 201
- heartbeats and pings, 196-199
- overview of, 195
- reversing problem statements, 201
- terminology used in, 195

failure detectors
- close enough failure detection, 199
- purpose of, 179, 195
- timeout-free, 197

failure models
- arbitrary faults, 193
- crash faults, 191
- failure handling, 193
- omission faults, 192
- purpose of, 191
- role of, 173

failure scenarios, 288, 304
fair path, 197
fair-loss links, 182
false-negatives, 195
false-positives, 195
fanout, 28, 32, 35, 251
Fast Paxos, 292
fastpath, 72
fault tolerance, 173, 215
faulty processes, 195
FD-Trees (Flash Disk Trees)
- fractional cascading, 118
- logarithmic runs, 119
- overview of, 117

fences, 119
fields, 12
file formats
- binary encoding, 47-50
- cell layout, 54
- checksumming, 59
- general principles, 50
- on-disk structures, 45-46
- page structure, 52
- slotted pages, 52
- variable-size data, 57
- versioning, 58

filesystem logging, 159
filter entries, 120
(see also tombstones)
finite duplication, 183
first committer wins, 272
first fit strategy, 57
first in, first out (FIFO) strategy, 86, 185
fit strategy, 57
flags, 50
flash drives, 30
Flash Translation Layer (FTL), 31, 157
Flexible Paxos, 296
floating-point numbers, 48
FLP Impossibility Problem, 189
flush finalization, 156
flushing, 81, 83, 90, 132
followers, 283
force operation, 90
force/no-force policy, 91
forward links, 62
forward-oriented validation, 98
fraction, 48
fractional cascading, 118

fractured reads, 276
fragmentation, 75, 160
free page list, 76
freeblocks, 57
freelist, 76
fsync(), 89
FUSE (failure notification service), 201
fuzzy checkpoints, 90
fuzzy reads, 95

G

garbage collection, 31, 74
ghost records, 75
gossip, 200
gossip dissemination
  gossip mechanics, 251
  gossip protocols, 250
  hybrid gossip, 253
  overlay networks, 251
  partial views, 254
group failures, 201
grow-only sets (G-Sets), 240
growing phase, 100

H

half-split nodes, 107, 122
HaloDB, 2
hard disk drives (HDDs), 30
hardware failures, 181
harvest and yield, 219
hash collision, 147, 246
hash-organized tables (hashed files), 18
hazard pointers, 150
HBase, 15
head elements, 120
head trees, 118
headers, 50
heap-organized tables (heap files), 18
heartbeat counters, 200
heartbeat protocols, 179, 196-199
high availability, 218
high keys, 64, 107
hinted handoff, 246
histories, 257
horizontal scaling, xiii, 167
hot data, 7
hotspotting, 181
hybrid gossip, 253, 254
Hybrid Partial View (HyParView) protocol, 254
hybrid transactional and analytical processing (HTAP), 8

I

idempotent operations, 184
IEEE Standard for Floating-Point Arithmetic, 48
immediate eviction, 85
immutability, 21, 165
immutable storage structures, 129, 165
implicit primary keys, 20
in-memory database management systems, 10
in-memory tables, 135
in-place updates, 25, 75, 130
incomplete checkpoints, 91
index entries, 36
index files, 17, 18
index-organized tables (IOT), 7, 18
indexes
  clustered versus nonclustered, 7, 19
  defined, 18
  primary and secondary, 19
  purpose of, 17
  SSTable-Attached Secondary Indexes (SASI), 146
indirection pointers, 67
indivisibility, 79
infective processes, 250
initial state, 189
InnoDB, 2
insertion points, 67
instantaneous processing, 175
internal nodes, 35
Invariant Confluence (I-Confluence), 275
invitation algorithm, 210
invocation events, 219
isolation, 79
isolation levels
  already committed, 96
  purpose of, 95
  read and write anomalies, 95
  serializability, 94
iterators, 137

J

jitter, 181

K

kernel page cache, 82
key hashes, 271
keydir, 153

L

last-write-wins register (LWW register), 239
latch crabbing, 105
latch upgrading, 107
latch-free, log-structured, access-method aware (LLAMA), 160
latches, 103
latency, 174, 251
lazy B-Trees
  benefits of, 114
  Lazy-Adaptive Tree (LA-Tree), 116, 165
  WiredTiger, 114, 165
Lazy-Adaptive Tree (LA-Tree), 116, 165
lazy-push, 253
leader election
  bully algorithm, 207
  candidate/ordinary optimization, 209
  in Raft, 302
  invitation algorithm, 210
  next-in-line failover, 208
  overview of, 205
  ring algorithm, 211
  stable leader election, 213
leaders, 260, 283, 291
leaf nodes, 35, 52
learners, 287
leases, 227, 291
least-frequently used (LFU), 87
least-recently used (LRU), 86
least-significant byte (LSB), 47
LevelDB, 2
leveled compaction, 142
libmdbx, 2
Lightning Memory-Mapped Database (LMDB), 113
linearizability, 97, 216, 218, 223-227
linearization point, 225
links
  defined, 168
  exactly-once delivery, 186
  failure sites, 195
  fair-loss links, 182
  message acknowledgments, 183
  message order, 185
  message retransmit problems, 184
  message retransmits, 184
  perfect links, 186
  stubborn links, 184
little-endian byte order, 47
liveness guarantees, 195, 205
LMDB, 2
load-balancing, 70
local clocks, 176
local execution, 178, 226
lock manager, 10, 80
lock table, 269
lock-based concurrency control
  Blink-Trees, 107
  deadlocks, 101
  latch crabbing, 105
  latch upgrading and pointer chasing, 107
  latches, 103
  locks, 102
  overview of, 100
  readers-writer lock (RW lock), 103
locking, 84
log buffer, 90
log manager, 80
log sequence number (LSN), 90
log stacking
  drawbacks of, 157
  filesystem logging, 159
  Flash Translation Layer (FTL), 157
log-structured merge-trees (LSM trees)
  concurrency in LSM trees, 155
  immutable storage offered by, 129
  LSM Tree lookups, 137
  maintenance in, 141
  merge-iteration, 137-140
  overview of, 130
  reconciliation, 140
  structure of, 132-136
  updates and deletes, 136
log-structured storage (LSS)
  implementation details, 145-152
  LLAMA and mindful stacking, 160
  log stacking, 157-160
  LSM Trees, 130-143
  mutable versus immutable storage, 129
  read, write, and space amplification, 143-145
  unordered LSM storage, 152-155
logarithmically sized sorted runs, 119

logical clocks, 168, 232
logical logs, 91
logical space, 74
loss of interest function, 250
lost update anomaly, 97
lost updates, 96

M

magic numbers, 62
main memory DBMS
  versus disk-based, 10
  durability in, 11
maintenance, 76
mapping tables, 121
memory cells, 30
memory mapped files, 32
memory-resident components (memtables), 132
memtable switch, 156
merge delta nodes, 123
merge functions, 275
merge reconciliation, 129
merge structural modification operations, 123
merge-iteration, 137-140
merges, 35, 41, 68, 122
Merkle trees, 247
message acknowledgments, 183
message redundancy, 251
message retransmits, 184
messages, 168
metadata, 243
min-heap, 138
mindful stacking, 160
MMAPv1, 2
monarchial leader election, 207
MonetDB, 13
MongoDB
  storage engines available, 2
  terminology used in, 20
  WiredTiger storage engine, 114
monotonic reads/writes models, 233
most-significant byte (MSB), 47
Multi-Paxos, 291
multicomponent LSM Trees, 134, 165
multiple copies, 173
multiple partitions, 294
multishard transactions, 269
multiversion concurrency control (MVCC), 93, 99
multiway trees, 36
mutability, 22
mutable storage structures, 129, 165
MyISAM, 2
MySQL
  append-only write-ahead logs, 89
  overflow pages, 65
  page header content, 61
  storage engines available, 2
  terminology used in, 20

N

network partitions, 179, 217
network unreliability, 179
next-in-line failover, 208
node crashes, 217
node high keys, 64
node merges, 41, 68
node splits, 39, 68
node updates, 113
nodes, 8, 52, 168
Non-Volatile Memory (NVM), 11
nonblocking atomic commitment problem, 259
nonclustered indexes, 7, 19
nonleaf nodes, 52
nonrepeatable reads, 95
notification broadcasts, 243
null-terminated strings, 49

O

occupancy, 35
offsets, 18, 51, 55
omission faults, 192
on-disk data structures, 32, 45
online analytical processing (OLAP), 8
online transaction processing (OLTP), 4, 7
Open-Channel SSDs, 161
operation consistency, 222
operation logs, 91
optimistic concurrency control (OCC), 93, 98
ordering, 21, 165, 221
outsourced heartbeats, 198
overflow, 39
overflow pages, 65
overlay networks, 251
overload, 181
O_DIRECT flag, 82

P

PACELEC conjecture, 217
page caches, 81, 83
page headers
  contents of, 61
  magic numbers, 62
  node high keys, 64
  overflow pages, 65
  rightmost pointers, 63
  sibling links, 62
page ID, 55
page reference events, 87
page replacement
  CLOCK algorithm, 86
  eviction policy, 85
  FIFO algorithm, 86
  least-frequently used (LFU), 87
  LRU-based cache eviction, 86
  usage frequency, 87
page-replacement policy, 85
paged binary trees, 32
pages, 18, 30, 35, 45, 52
paging-in, 81, 87
parallel computing, 173
parent updates, 122
partial failures, 179
partial views, 254
partially synchronous systems, 191
participants, 168
partitioning, 294
pathological trees, 27
Paxos
  distributed transactions in Spanner, 269
  Egalitarian Paxos (EPaxos), 293
  failure scenarios, 288
  Fast Paxos, 292
  Flexible Paxos, 296
  generalized solution to consensus, 297
  Multi-Paxos, 291
  overview of, 285
  Paxos algorithm, 286
  quorums in, 287
peer sampling services, 254
peer-to-peer information exchange, 243
Percolator, 272-274
perfect delivery, 187
perfect links, 186
performance evaluation tools, 4
pessimistic (conservative) concurrency control (PCC), 93, 99
phantom reads, 95
phases, 168
phi-accrual (φ-accrual) failure detector, 199
physical clocks, 168
physical logs, 91
pinning, 83, 84
pipelining, 176
platform byte order, 48
point queries, 35
pointer chasing, 107
pointers, 38
Positive-Negative-Counter (PN-Counter), 239
PostgreSQL
  append-only write-ahead logs, 89
  Blink-Trees, 64
  circular buffers, 85
  versus fsync(), 89
  page header content, 61
  right-only appends (fastpath), 72
  terminology used in, 20
  versioning in, 58
Practical Byzantine Fault Tolerance (PBFT), 306
Pre-Accept phase, 294
predicate deletes, 137
prefetching, 85
prepare operations, 259, 260
primary files, 18
primary indexes, 19-21
primary keys, 20-21
primary pages, 65
primitive types, 47
priority queues, 138
probabilistic interest loss, 251
probation queue, 88
process-local queues, 176
processes
  consistency models and, 223
  defined, 168
  failure sites, 195
  fair-loss links, 182
  infective processes, 250
  susceptible processes, 250
processing time, 175
promises, 262, 286
promotion, 39
proposers, 286

prospective leaders, 283
protected queue, 88
push/lazy push multicast trees (Plumtrees), 253

Q

query optimizer, 9
query plan, 9
query processor, 9
questions and comments, xviii
queue capacity, 175
queueing techniques, 104
quickbalance, 72
quorums, 236, 245, 287

R

Raft
  failure scenarios, 304
  leader role in, 302
  overview of, 300
random I/O, 31
random processes, 250
range queries, 35, 154
range tombstones, 137
RCFile, 14
read amplification, 143-145
read anomalies
  dirty read, 95
  fractured read, 276
  nonrepeatable read, 95
  phantom read, 95
read atomic isolation level, 276
read committed isolation level, 96
read performance, 130
read phase, 98
read repair, 245
read sets, 267
read skews, 272
read uncommitted isolation level, 96
Read-Atomic Multi Partition (RAMP) transactions, 275
read-only transactions, 268
read-own-writes consistency model, 233
read-time data repair, 177
read-write transactions, 268
readers-writer lock (RW lock), 103
rebalancing, 70
recency conflicts, 248
reconciliation, 140
recoverability, 258
recovery
  ARIES algorithm, 92
  Byzantine consensus, 309
  log semantics, 90
  operation versus data logs, 91
  steal and force policies, 91
  write-ahead log (WAL), 88
recovery manager, 10
redo operations, 91
redundancy, 173
redundant messages, 251
referencing, 83
registers, 219, 239
regular registers, 220
reliability, 174, 186
reliable broadcasts, 281
remote execution, 10, 178
remove delta nodes, 123
removed processes, 250
repeatable read isolation level, 96
repeating history, 92
replica sets, 270
replica states, 248
replicas, 168
replicated logs, 291
replication and consistency
  CAP conjecture, 216-219
  Conflict-Free Replicated Data Types (CRDTs), 238
  consistency models, 222-233
  eventual consistency, 234
  ordering, 221
  overview of, 215, 240
  quorums, 236
  session models, 233
  shared memory, 219
  strong eventual consistency, 238
  system availability, 216
  tunable consistency, 235
  witness replicas, 236
Reusable Infrastructure for Linearizability (RIFL), 227
Riak, 20, 246
right-only appends, 71
rightmost pointers, 63
ring algorithm, 211
robustness, 251
RocksDB, 2
rollback operations, 259

root nodes, 26, 35
routing keys, 270
row keys, 16
row locators, 18
row-oriented data layout, 12-15
rows, 12
RUM Conjecture, 144

S

safe registers, 220
safety guarantees, 195, 205
sanity checks, 62
scaling horizontally (scaling out), xiii, 167
scaling vertically (scaling up), xiii, 167
schedulers, 267
schedules, 94
search keys, 17
secondary indexes, 19
sectors, 30
seeks, 30
separator keys, 36, 38, 63
sequence numbers, 183
sequencers, 266
sequential consistency, 227
sequential I/O, 30
sequential operations, 219
serial operations, 220
serial schedules, 94
serializability, 94, 96, 257
serialization, 48
session causality, 234
session models, 233
shadow paging, 91
shadowing, 18
sharding, 270
shared memory, 172, 219
shared state, 173
short-time bursts, 176
shrinking phase, 100
sibling links, 62, 107
sign, 48
single-operation consistency models, 222
single-partition transactions, 257
size-tiered compaction, 143
skiplists, 115, 148
sloppy quorums, 246
slot directory, 53
slotted pages, 18, 52, 56, 74
snapshot isolation (SI), 97, 272
snapshot reads, 268
snapshots, 12
Software Defined Flash (SDF), 162
software errors, 181
solid state drives (SSDs), 30, 157, 161
Sophia, 2
Sorted String Tables (SSTables), 145
space amplification, 143-145
Spanner, 268
spanning trees, 252
spanservers, 269
speculative execution, 288
spinning disks, 30
split brain, 205, 208
split delta nodes, 122
split point, 39
split SMOs, 122
split vote, 304
splits, 35, 39, 68, 122
SQLite
    overflow pages, 65
    page header content, 61
    quickbalance, 72
    rightmost pointers, 63
    unoccupied segments (freeblocks), 57
SSTable-Attached Secondary Indexes (SASI), 146
stable checkpoints, 309
stable leader election, 213
state, 168, 168, 222
state consistency, 177
steal/no-steal policy, 91
steps, 168
storage engines
    responsibilities of, 1, 10
    understanding trade-offs, 5
strategy, 57
strict consistency, 223
strings, 30, 49, 53
strong eventual consistency, 238
structural modification operations (SMOs), 122
stubborn links, 184
subtrees, 26, 36
super-peers, 239
susceptible processes, 250
suspected processes, 195
suspicion level, 199
sync checkpoint, 90
synchrony, 173, 190

system availability, 216
system loads, 176

T

table starvation, 143
table views, 155
tables
  defined, 12
  disk-resident tables, 132
  hash-organized tables (hashed files), 18
  heap-organized tables (heap files), 18
  in-memory tables, 135
  index-organized tables, 7, 18
  lock tables, 269
  mapping tables, 121
  Sorted String Tables (SSTables), 145
tablets, 269
TCP protocol, 186
Thomas Write Rule, 100
three-phase commits, 264-266
threshold interest loss, 251
throughput, 4
time window compaction, 143
timeout-free failure detection, 197
timestamp oracle, 273
timestamp ordering, 99
TinyLFU, 87
tombstones, 18, 137, 141
total order multicast, 281
TPC-C benchmark, 4
trailers, 50
transaction isolation, 95
transaction manager, 10, 101, 259, 269
transaction processing and recovery
  ACID properties, 79, 94, 97, 218
  buffer management, 81-88
  concurrency control, 93-108
  recovery, 88-93
Transaction Processing Performance Council (TPC), 4
transitions, 168
transitive coordination, 248
transport subsystem, 8
tree balancing, 27, 70
tree height, 29, 32
trimming, 90, 136
TrueTime, 268
tunable consistency, 235
Two Generals' Problem, 187
two-component LSM Trees, 132, 165
two-level memory hierarchy, 81
two-phase commits (2PCs)
  cohort failures in, 261
  coordinator failures in, 262
  overview of, 259
  versus two-phase locking, 100
two-phase locking (2PL), 100
types, 47

U

ubiquitous B-Trees, 33-42
unclustered indexes, 19
uncommitted version, 99
underflow, 41
undo and redo, 92
unique identifiers, 183
unordered storage structures
  benefits of, 152
  Bitcask, 153, 165
  WiscKey, 154, 165
unresponsive processes, 195
update buffers, 115
update chains, 121
upserts, 140
usage frequency, 87, 87

V

vacuum, 76
validation, 62
validation phase, 98
van Emde Boas layout, 125
vector clocks, 232
versioning, 58
Vertica, 13
vertical scaling, 167
virtual disks, 81
virtual IDs, 14
virtual synchrony, 282
vLogs (value logs), 154

W

wait-die restriction, 102
waits-for graphs, 101
wall clocks, 168
wear leveling, 158
wide column stores, 7, 15
WiredTiger, 2, 114, 165

WiscKey, 154, 165
witness replicas, 236
wound-wait restriction, 102
write amplification, 143-145
write anomalies, 96
write performance, 130
write phase, 98
write sets, 267
write skews, 96, 97, 272
write-ahead log (WAL), 83, 88
write-ahead log truncation, 156
write-once registers, 292
write-write conflicts, 272
writes-follow-reads model, 234

Y

Yahoo! Cloud Serving Benchmark (YCSB), 4
yield, 219
YugaByte, 268

Z

Zookeeper Atomic Broadcast (ZAB), 283