NUMBERS
0MQ, 122
10gen, 180

A
A/B testing, 3–4, 368, 400
AbstractChannelSelector class, Flume, 97–98
AbstractSource class, Flume, 105–106
Acceptor, Paxos, 38
ACID (Atomicity, Consistency, Isolation, Durability), 20, 21
activation function, 380–389
ActiveMQ, 19, 64, 71, 77, 102–103
Advanced Message Queuing Protocol (AMQP), 141
agent command, Flume, 115
agents, Flume, 92–95, 114–115
Aggarwal, Charu, 329
aggregation. See also stochastic optimization
Cassandra, 214
and distributed hash table stores, 216
MongoDB, 190–199
multi-resolution time-series, 290–295
and relational databases, 217
timed counting, 285–290
Trident, 147–150
Agrawala, Maneesh, 301
AJAX (Asynchronous JavaScript and XML), 22–23
Algorithm R, 327
Amazon
DynamoDB, 203, 216
EC2, 83, 84, 204, 205
Kinesis, 74
Route 53 DNS service, 83
AMQP (Advanced Message Queuing Protocol), 141
animation, D3, 271–272
ANN (artificial neural network), 380
anomaly detection, 396
change detection, 399–400
outlier detection, 397–399
Apache Commons Math Library, 377
Apache Flume. See Flume
Apache Kafka. See Kafka
Apache ZooKeeper. See ZooKeeper
approximation
sketches, 331–332
Bloom filters, 338–347
Count-Min, 356–364
Distinct Value, 347–355
hash functions, 332–336
registers, 332
working with sets, 336–338
statistical analysis, 305–306
numerical libraries, 306
random number generation, 319–324
sampling procedures, 324–329
architectures
Apache YARN, 152–153
Cassandra, 204
checklist, 30–34
components, 16–24
features, 24–27
Lambda Architecture, 223
languages, 27–30
arcTo command, HTML5 Canvas, 257
artificial neural network (ANN), 380
ASCII data type, CQL, 209
Asynchronous JavaScript and XML (AJAX), 22–23
Atomicity, Consistency, Isolation, Durability (ACID), 20, 21
attribution process, 364
Avro sink, Flume, 108
Avro source, Flume, 98–99
B
Backhoe Event, 37
backpressure, 104
backpropagation, 384–389
BackType, 20, 119–120
base reservoir algorithm, 326, 327–329
be command, Kafka, 82
Bernoulli distribution, 378
beta distributions, 314, 322
biased streaming sampling, 327–329
BIGINT data type, CQL, 209
BigTable, 170, 203, 216
binomial coefficient, 311
binomial distributions, 311
Birthday Paradox, 335–336
BLOB data type, CQL, 209
Bloom filters, 338–347
algorithm, 338–340
cardinality estimation, 342–343
intersections, 341–342
size, 340–341
unions, 341–342
variations, 344–347
bolts, 120
basic, 135
counting in, 286–288
implementing, 130–136
logging, 135–136
rich, 131–133
BOOLEAN data type, CQL, 209
Boost Library, 306
Bootstrap, 237, 238, 277
Bostock, Mike, 280, 302
bot networks, 343
Box-Muller method, 322
brokers, Kafka
configuring, 81–88
interacting with, 89–92
multi-broker clusters, 88–89
replication, 78–79
space management and, 77
starting clusters, 88
BPOPFLUSH command, Redis, 174
byte code, 29

C
callback pyramid of doom, 230–231
callback-driven programming, 229–230
Camas, 75, 218–221, 222–223
CamusWrapper class, 219–221
capped collections, MongoDB, 183–184
cardinality, 9–10
Cash Register models, 356, 357
Cassandra, 203–214
cluster setup, 205–206
configuration options, 206–207
CQL (Cassandra Query Language), 207–208
insert and update operations, 211–214
keyspaces, 208–211
reading data from, 214
server architecture, 204
central moments, 310
change detection, 399–400
channel selectors, Flume, 95–98
channels, Flume, 110–112
checkExists command, ZooKeeper, 60, 62–63
chi-square distribution, 313–314, 322–324
classic Storm, 120
Clojure, 28–29
clusters
Cassandra, 205–207
horizontal scaling, 26–27
Kafka, 75, 79, 88–89
Redis, 179–180
Storm, 120–126
ZooKeeper, 42–47
collection, 31
collections, MongoDB, 182–184
Colt numerical library, 306, 311, 322–324
Comet, 23
Command-R interface, 293–295
complement of a set, 336
conditional probability, 307–309, 315
configuration and coordination, 35
maintaining distributed state, 36–39
motivation for, 36
ZooKeeper. See ZooKeeper
consistency, 20–21
Cassandra, 203
Redis, 170
ZooKeeper, 41
consistent hashing, 168–169
Consumer implementation, Kafka, 91–92
continuous data delivery, 7–8
continuous distributions, 312–314
correlation, 315
Count-Min sketch algorithm, 356–363
Heavy Hitters list, 358–360
implementation, 356, 357–358
point queries, 356–357
top-K lists, 358–360
COUNTER data type, CQL, 209
counting
timed, 285–290
Word Count example, 149–150
Counting Bloom Filters, 344–346
covariance, 315
CQL (Cassandra Query Language), 203, 207–208.
See also Cassandra
cqlsh command, CQL, 207–214
cubism.js project, 302
Curator client, 56–63
adding to Maven project, 56–57
connecting to ZooKeeper, 57–59
using watches with, 62–63
working with znodes, 59–62
Curator recipes, 63–70
distributed queues, 63–68
leader elections, 68–70
CuratorFramework class, ZooKeeper, 57–59
CuratorFrameworkFactory class, ZooKeeper, 57–59
custom sinks, Flume, 109–110
custom sources, Flume, 105–107

D
D3.js, 29–30, 262–272
animation, 271–272
attributes and styling, 263–264
inserting elements, 263
joining selections and data, 265–267
layouts, 269–271
removing elements, 263
scales and axes, 267–269
selecting elements, 263
shape generators, 264–265
strip charts, 298–299
dashboard example, 238–243, 251–254
data collection, 16–17, 31
Data Driven Documents. See D3.js
data flow, 17–19, 31–32
distributed systems, 72–74
Flume. See Flume
Kafka. See Kafka
Samza integration, 157
data grids, 215, 217
data models, 368–369
Flume, 95
forecasting with, 389–396
linear, 373–378
logistic regression, 378–379
neural network, 380–389
time-series, 369–373
data processing
coordination, 118–119
merges, 119
overview, 118–119
partitions, 119
with Samza. See Samza
with Storm. See Storm
transactional, 119
data sets, high-cardinality, 9–10
data storage, 20–22
Cassandra, 203–214
consistent hashing, 168–169
data grids, 215, 217
MongoDB, 180–203
Redis, 170–180
relational databases, 215, 217
technology selection considerations, 215–217
warehousing, 217–223
data types
CQL (Cassandra Query Language), 209
Redis, 173, 175
data visualization
D3 framework, 262–272
HTML5 Canvas, 254–260
Inline SVG, 260–262
NVD3, 272–274
Vega.js, 274–277
databases
Cassandra. See Cassandra
MongoDB, 182–183, 182–184
NoSQL. See NoSQL storage systems
round-robin, 290
sharding. See sharding
datum command, NVD3, 273
DECIMAL data type, CQL, 209
DECR command, Redis, 173
decrement command, Redis, 173
deep learning, 3, 383
delete command, ZooKeeper, 40, 62
delta method, 317–319
DHTs (distributed hash tables), 217
dimension reduction, 330, 364
discrete distributions, 310–312
Distinct Value (DV) sketches, 347–348
HyperLogLog algorithm, 351–355
Min-Count algorithm, 348–351
distributed hash tables (DHTs), 217
distributed queues, 63–68
Distributed Remote Procedure Calls (DRPC), 142–144
distributions
continuous, 312–314
definition of, 24
Delta Method, 317–319
discrete, 310–312
generating specific, 321–324
inequalities, 319
inferring parameters, 316–317
joint, 315–316
statistical, 310
document stores
MongoDB, 180–203
selection considerations, 216
double data type, CQL, 209
double hashing, 334–335
DRPC (Distributed Remote Procedure Calls), 142–144
dump command, ZooKeeper, 45
DV (Distinct Value) sketches, 347–348
HyperLogLog algorithm, 351–355
Min-Count algorithm, 348–351
dyadic intervals, 361–362
DynamoDB, 203, 216

E
EC2, Amazon, 83, 84, 204, 205
data servers, 16, 17, 31
Memcached, 171
Elasticsearch, 107, 115
empty sets, 336
envi command, ZooKeeper, 45–46
ephemeral znodes, 40
epoch number (Zab), 41
error correcting form, 391
Etc, 39, 70
ETL (extract-transform-load) tools, 217
  Hadoop as, 218–223
EVAL command, Redis, 177
EVALSHA command, Redis, 177
Event-Driven sources, Flume, 105–107
EventEmitter class, Node, 230–231, 243
EventSource class, Node, 245
eventual consistency, 21
  Cassandra, 203
  Redis, 170
exclusive-or (XOR) pattern, 387–388
Exec source, Flume, 103–104
EXISTS command, Redis, 172
expectation, 309–310
  Delta Method, 317–319
  method of moments, 317
  exponential distributions, 314, 321–322
  exponential moving average, 372–373
  exponential smoothing methods, 390–393
  exponentially biased reservoir sampling, 328–329
express.js framework, 237

F
factorial functions, 308, 335–336
fat jars, 125–126
feed-forward networks
  backpropagation algorithm, 384–389
  multi-layer implementations, 381–384
File channel, Flume, 111
fill command, HTML5 Canvas, 257
filterDefinition class, Storm, 133–135
first success distribution, 311
Fisher-Yates Shuffle, 325–327
Flajolet
  HyperLogLog algorithm, 351–355
  stochastic averaging, 349
float data type, CQL, 209
flow management, 71–72
distributed data flows, 72–74
  Kafka. See Kafka
  Flume. See Flume
Flume
  agents, 92–95, 114–115
  channels, 110–112
  crosspath integration, 114
  custom component integration, 114
  data model, 95
  interceptors, 112–114
  plug-in integration, 114
  sink processors, 110
  sinks, 107–110
  sources, 98
    Avro, 98–99
    custom, 105–107
    Exec, 103–104
    HTTP, 101–102
    Java Message System (JMS), 102–103
    Netcat, 99–100
Spool Directory, 104–105
  Syslog, 100–101
  Thrift, 99
Storm connections, 141
FNV (Fowler, Noll, and Vo) hash, 333–334
forecasting, 389–396
  exponential smoothing methods, 390–393
  neural network methods, 394–396
  regression methods, 393–394
  four-letter words, ZooKeeper, 45
  Fowler, Noll, and Vo (FNV) hash, 333–334
frequency tables, 356
functions
  activation, 380–389
  factorial functions, 308, 335–336
  hash functions, 332–336
  inner functions, 236
  moment generating, 317, 319
  outer functions, 236
  sigmoid functions, 380, 382

G
gamma distributions, 314, 322–324
generalized linear models (GLMs), 378
generalized linear models (GLMs), 378
generalized linear models (GLMs), 378
generalized linear models (GLMs), 378
GET command, Redis, 172
getChildren command, ZooKeeper, 40, 60,
  61–62
getChildren command, ZooKeeper, 40,
  60–61, 62
Giroire, Frédéric, 348
GLMs (generalized linear models), 378
GNU Lesser General Public License (LGPL), 306
GNU Scientific Library, 306
Go language, 30
Google
  BitTable, 170, 203, 216
  Go language, 30
  HyperLogLog++, 354–355
  Protocol Buffers, 17
  V8 engine, 20, 228
gossip protocol, Cassandra, 204–206
gradient descent, 379, 394

H
Hadoop, 218–223
  for ETL processes, 223
  event vs. processing time, 222
  ingesting data from Flume, 221
  ingesting data from Kafka, 218–221
  map-reduce processing, 19–20
hash functions, 332–336
  and the Birthday Paradox, 335–336
  double hashing, 334–335
  independent, 333–334
hash tables
  distributed stores, 216
  double hashing, 334–335
  Redis, 172–173
Hazelcast, 215
Heaviside step function, 380, 382
Heavy Hitters list, 358–360
Heer, Jeffrey, 301
HGET command, Redis, 173
HGETALL command, Redis, 173
Hidden Markov Models, 399
high availability, 24–25
high-cardinality data sets, 9–10
high-speed canvas charts, 299–301
HINCRBY command, Redis, 173
HINCRBYFLOAT command, Redis, 173
Hive, 223
HSET command, Redis, 173
HSET command, Redis, 173
Holt-Winters models, 390–393
horizon charts, 282, 301–302
horizontal scaling, 26–27
Host intercept, Flume, 112–113
HSET command, Redis, 173
HTML5 Canvas, 254–260
HTTP source, Flume, 101–102
Hummingbird, 300–301
hyperbolic tangent, 380, 382–384
hypergeometric distributions, 311
HyperLogLog algorithm, 351–356
implementing, 352–354
improvements to, 354–355
real-time unique visitor pivot tables, 355
HyperLogLog++, 354–355

I
identity matrix, 377
IDL (interface definition language), 17, 98
immediate mode implementations, 24
impressions, 340
in-memory data grids, 215, 217
in-sync replicas, 26, 78, 199
inclusion-exclusion principle, 337–338, 355
INCR command, Redis, 173
INCRBY command, Redis, 173
INCRBYFLOAT command, Redis, 173
INCRBYFLOAT command, Redis, 173
independent hash functions, 333–334
indexing, MongoDB
  basic, 184–185
  full text, 186
  geospatial, 185–186
  optional parameters, 186–187
INSET data type, CQL, 209
INFO command, Redis, 172
Inline SVG, 260–262
inner functions, 236
insert and update operations
  Cassandra, 211–214
  MongoDB, 188–189
installing
  Kafka, 80–81
  ZooKeeper server, 42–44
INT data type, CQL, 209
intercept term, 374
interceptors, Flume, 112–114
interface definition language (IDL), 17, 98
Internet of Things, 5–7
intersection of a set, 336

J
Jacquard, 343
Jacquard Similarity, 344, 350–351
Java, 27–28
Java client, ZooKeeper
  adding ZooKeeper to Maven project, 47–48
  connecting, 48–56
Java Database Connection (JDBC) channel, Flume, 111–112
Java Management Extensions (JMX), 45, 89
Java Message System (JMS) source, Flume, 17
Java, 27–28
JDBC (Java Database Connection) channel, Flume, 111–112
Jenkins hash, 333
Jensen's inequality, 319
JMX (Java Management Extensions), 45, 89
jobs, Samza, 157–166
  configuring, 158–160
  executing, 165–166
  implementing stream tasks, 16
  initializing tasks, 161–163
  packaging for YARN, 163–165
  preparing job application, 158
  task communication, 160
joint distributions, 314–316, 315–316
Just a Bunch of Disks (JBOD), 84, 111, 205
JDBC (Java Database Connection) channel, Flume, 111–112
K
Kafka, 74
  brokers, 75, 89–92
  configuring environment, 80–89
  design and implementation, 74–79
  installing, 80–91
  prerequisites, 81
  replication, 78–79, 84–88
  Samza integration, 157
  Storm integration, 140–141
Kenshoo, 141
kernels
  exponential moving average, 372–373
  weighted moving average, 370–372
  key-value stores, 21, 169–170
  Cassandra, 203–214
  Redis, 170–180
  keys, Cassandra, 208–211
kill command, Storm, 125
Kinesis, 74
Kirsch and Mitzenmacher, 334
Kong, Nicholas, 301
Kreps, Jay, 76

L
Lambda architecture, 22, 223
LCRNNG (Linear Congruential Generator random number generator), 319–320
leader elections
using Curator, 68–70
using ZooKeeper, 49–56
LeaderElection class, ZooKeeper, 49–56
LeaderLatch class, ZooKeeper, 68, 69
LeaderSelector class, ZooKeeper, 68–69
learning rate, 296, 379, 386
least squares model, 373
Lesser General Public License (LGPL), 306
lexical scoping, 236
LGPL (Lesser General Public License), 306
Lightweight Transactions, Cassandra, 212
Linear Congruential Generator random number generator (LCRNNG), 319–320
Linear Counting, 352
linear models, 369, 373–378
multivariate linear regression, 376–378
simple linear regression, 374–376
lineTo command, HTML5 Canvas, 257
LinkedIn
Apache Kafka project, 74
Camus and, 218
Samza project, 151
LIST data type, CQL, 209
lists
Redis, 173–174
top-K, 358–360
lock servers, 118
log collection. See Flume
logistic function, 380, 382
logistic regression, 378–379
long tail, 9
longitudinal data, 6
loosely structured data, 8–9
Lorem Ipsum spout, Storm, 138–140
low latency, 25–26
LPOP command, Redis, 173
LPOLLPUSH command, Redis, 174
LPUUSH command, Redis, 173, 174
LREM command, Redis, 174
Lua, 177–178

M
MACD (Moving Average Convergence Divergence), 399–400
MAP data type, CQL, 209
Markov inequality, 319
Marz, Nathan, 22, 223
Maven, 28
Apache Commons Math library, 377
assembly plug-in, 163–165
Curator, adding, 56–57
Samza packages, 158
topology projects, starting, 125–126
ZooKeeper, adding, 47–48
maximum likelihood estimation, 316–317
mean, 309
Memory channel, Flume, 111
MemoryMapState class, Trident, 148
merges, 119
method of moments, 317
methods
JavaScript, 29
stochastic optimization, 296–297
web communication, 22–23
web rendering, 23–24
metro collections, MongoDB, 189–190, 191–195
MSET command, Redis, 172
Microsoft VML (Vector Markup Language), 23, 260, 261
Min-Count sketch algorithm, 348–351
computing set similarity, 350–351
implementing, 349–350
Min-wise Hashing, 350–351
MirrorMaker, 79, 157
mobile streaming applications, 277–279
moment generating function, 317, 319
momentum backpropagation, 388–389
mongod, 180–181, 201–202
MongoDB, 180–203
basic indexing, 184–185
capped collections, 183–184
collections, 182–184
full text indexing, 186
geospatial indexing, 185–186
insert and update operations, 188–189
metro collections, 189–190, 191–195
model, 180
replication, 199–200
setup, 180–182
sharding, 200–203
mongos, 200–203
Most Recent Event Tracking, 177–178
moveTo command, HTML5 Canvas, 257
moving average, 369–370
Moving Average Convergence Divergence (MACD), 399–400
MSET command, Redis, 172
multi-armed bandit optimization strategy, 368, 400–402
Multi-Paxos, 38–39
multi-resolution time-series aggregation, 290–295
multiplexing selectors, Flume, 96–97
multivariate linear regression, 376–378
MurmurHash, 333, 338–340

N
naïve set theory, 336
negative binomial distributions, 311–312
Netcat source, Flume, 99–100
Poisson distribution, 312, 321–322
Pollable sources, Flume, 105–107
posterior distributions, 401–402
PostFilter class, Trident, 288
Prepare-Promise cycle, 38–39
probability mass function (PMF), 310, 311, 313
probability theory, 307–309
  continuous distributions, 312–314
  discrete distributions, 310–312
  expectation, 309–310
  joint distributions, 315–316
  statistical distributions, 310
  variance, 309–310
  working with distributions, 316–319
Producer, Kafka, 90
programmatic buying, 340
Proposer, Paxos, 38
ProtoBuf, 17
Protocol Buffers, 17
PUBLISH command, Redis, 178–179
publish/subscribe support, Redis, 178–179

Q
quantile queries, 360–364
quasi-Newton techniques, 379
QueueBuilder class, ZooKeeper, 64–65

R
R Statistical Library, 306
RabbitMQ, 64, 73, 77, 141
Raft, 39
random number generation, 319–325
random variates, 320, 321
range queries, 360–364
real-time architectures
  checklist, 30–34
  components, 16–24
  features, 24–27
  languages, 27–30
rebalance command, Storm, 121
Redis, 170–180
  client notifications, 297
  dashboard example, 251–254
  drawbacks, 32–33
  publish/subscribe support, 178–179
  replication, 179–
  scripting, 176–178
  setup, 170–171
  sharding, 179–180
  working with, 171–176
Redis Cluster, 179
redis-cli tool, 171–172
registers, 332
regression methods, 393–394
regression models, 369, 378–379
Regular Expression Filter interceptor, Flume, 113–114
rejection sampling, 323–324

Netty transport, 123
Network of Workstations (NOW) environments, 35
Network Time Protocol (NTP), 37
  networks
  bot networks, 343–344
  Kafka threads for processing requests, 83
  local topology, 204
  NAS and Cassandra, 205–206
  neural network models, 380–389, 394–396
  NTP (Network Time Protocol), 37
  unreliable connections, 36–37
  neural network models, 380–389, 394–396
  backpropagation, 384–389
  multi-layer, 381–384
Node, 228, 229
  callback pyramid of doom, 230–231
  callback-driven programming, 229–230
  developing web apps, 235–238
  managing projects with NPM, 231–235
  node package manager (NPM), 231–235
  non-blocking I/O mechanisms, 228, 229
  normal distribution, 313–314, 322
  normal equations, 377
NoSQL storage systems, 20–22, 169–170
  Cassandra, 203–214
  MongoDB, 180–203
  Redis, 170–180
notifications, ZooKeeper, 41
NOW (Network of Workstations) environments, 35
NPM (node package manager), 231–235
  NTP (Network Time Protocol), 37
null sets, 336
numeral libraries, 306
NumPy, 306
nutcracker, 179
NVIDIA, 273–274

Obermark, Ron, 76
online advertising, 4–5, 340
operational monitoring, 3
ordinary least squares, 376–378
outer functions, 236
outlier detection, 397–399

partitions, 119
  Kafka, 75
  partition local, Trident, 145
  repartitioning operations, Trident, 147
Paxos algorithm, 38–39, 41
pen movement commands, HTML5 Canvas, 257
  persistence, Trident, 147–150
persistent znodes, 40, 60–62
Pig, 223
PMF (probability mass function), 310, 311, 313
  point queries, 356–357, 361
relational databases, 20–22, 25, 215, 217
consistent hashing, 168–169
repartitioning operations, Trident, 147
Replica Set, MondoDB, 199–200, 202
replicating selectors, Flume, 96
replication, 25
Cassandra, 208
in-sync replicas, 26, 78, 199
Kafka, 78–79, 84–88
MongoDB, 199–200
Redis, 179
reqs command, ZooKeeper, 47
reservoir algorithms, 326–327
residual sum of squares (RSS), 374
retained mode implementations, 24
RFC 6455 (WebSocket), 249–251
RichBaseBolt class, Storm, 131
round-robin
custom stream grouping, 129–130
databases, 290
RPPOP command, Redis, 173
RPOPLPUSH command, Redis, 174
RPUSH command, Redis, 173, 174
RSS (residual sum of squares), 374
ruck command, ZooKeeper, 45

S
SADD command, Redis, 175
sampling, 324–329
biased streaming, 327–329
from fixed population, 325–326
from streaming population, 326–327
Samza, 151
Apache YARN and, 151–153
counting jobs, 289–290
integrating into data flow, 157
jobs, 157–166
multinode, 155–157
single node, 153–155
Scala, 28–29
scatter-gather implementations, 119
SciPy, 306
Scribe, 18, 71–72, 99
scripting, Redis, 177–178
SDTPP command, Redis, 175
second order expansion, 318–319
sensor platforms, 10
sequential snodes, 40–41
Server Sent Events (SSEs), 23, 33–34, 245–249
servers
coordination servers, 118–119
horizontal scalability, 26–27
vertical scaling, 27
SET command, Redis, 172
SET data type, CQL, 209
Set<Ex> interface, 338–340
setData command, ZooKeeper, 40, 41
sets, 336–338
Bloom filters, 338–347
algorithm, 338–340
cardinality estimation, 342–343
intersections, 341–342
size, 340–341
unions, 341–342
variations, 344–347
Distinct Value sketches, 347–348
HyperLogLog algorithm, 351–355
Min-Count algorithm, 348–351
shape generators, D3, 264–265
sharding
MongoDB collections, 200–203
Redis databases, 179–180
sigmoid functions, 380, 382
simple linear regression, 374–376
simple random sampling, 325–326
sink processors, Flume, 110
sinks, Flume, 107–110
SINTER command, Redis, 175
SISMEMBER command, Redis, 175
sketch algorithms, 331–332
Bloom Filter, 338–347
Count-Min, 356–363
hash functions, 332–336
HyperLogLog, 351–356
Min-Count, 348–351
registers, 332
sets, 336–338
SLEEP command, Storm, 126
sliding window reservoir sampling, 328
smoothing methods, 390–393
SMOVE command, Redis, 175
sorted sets, Redis, 175–176
sources of streaming data, 2–7
sources, Flume, 98
Avro, 98–99
custom, 105–107
Exec, 103–104
HTTP, 101–102
Java Message System (JMS), 102–103
Netcat, 99–100
Spool Directory, 104–105
Syslog, 100–101
Thrift, 99
Spectral Bloom Filter, 356
Split Brain Problem, 37
Spool Directory source, Flume, 104–105
SPOP command, Redis, 175
spouts, 120
implementing, 136–141
Lorem Ipsum, 138–140
SPRM command, Redis, 175
SSTAT command, ZooKeeper, 47
SSEs (Server Sent Events), 23, 33–34, 245–249
Stable Bloom Filters, 346–347
STAT command, ZooKeeper, 46–47, 62
Static interceptor, Flume, 113
statistical analysis, 305–306. See also probability theory
numerical libraries, 306
random number generation, 319–324
sampling procedures, 324–329
Stochastic averaging, 296, 349
Stochastic gradient descent, 296
Stochastic optimization, 296–297. See also aggregation
Storage, 20–22
Cassandra, 203–214
Consistent hashing, 168–169
data grids, 215, 217
High-cardinality storage, 9–10
MongoDB, 180–203
Redis, 170–180
Relational databases, 215, 217
technology selection considerations, 215–217
Warehouse, 217–223
Versus other data, 7–10
Streaming web applications
backend server communication, 242–254
dashboard example, 238–242
Node, 229–238
strip charts, D3, 298–299
Stroke command, HTML5 Canvas, 257
Subscribe command, Redis, 178–179
Summation, 285–290
Union command, Redis, 175
Supervisors, Storm, 121
Syslog source, Flume, 100–101
Systems monitoring, 396–400
at least once delivery, 72–73
change detection, 399–400
Outlier detection, 397–399
As source of streaming data, 3
Space management, 77

T
TCP/IP-based networks, 16
text data type, CQL, 209
Thompson Sampling, 209
Thrift, 17
Sink, 108
Source, 99
tiers, 18, 34
time-series aggregation, 290–295
time-series models, 369
Exponential moving average, 372–373
Moving average, 369–370
Weighted moving average, 370–372
timed counting, 285–290
In bolts, 286–288
In Samza, 289–290
In Trident, 288–289
timestamp data type, CQL, 209
Timestamp interceptor, Flume, 112
timeuuid data type, CQL, 209
top-K lists, 358–360
topics, Kafka, 75
Creating and management, 84–85
Replication and, 78
Topologies, Storm, 120, 124–126,
127–129
TopologyBuilder class, Storm, 127–129
Transactional processing, 119
TransactionalTridentKafkaSpout class, 149–150
Trident, 120, 144
Aggregation, 147–148
Counting events, 288–289
Local operations, 145–147
Partition local aggregation, 148
Repartitioning operations, 147
streams, 144–145
Word Count example, 149–150
TridentTopology class, 145
Turnstile model, 356
Twemproxy, 179–180, 200
Twitter, 5
Bootstrap, 237, 238, 277
Rainbird, 20
Storm. See Storm
Twemproxy, 179–180, 200

U
union of a set, 336
universal hash functions, 334
update command, NVD3, 273
UUID data type, CQL, 209
UUID interceptor, Flume, 113

V
V8 engine, 30, 228
values class, Storm, 132
VARCHAR data type, CQL, 209
variance, 309–310
VARCHAR data type, CQL, 209
Vector Markup Language (VML), 23, 260, 261
Vega.js, 274–277
vertical scaling, 27
visualizing data, 254
D3 framework, 262–272
HTML5 Canvas, 254–260
Inline SVG, 260–262
NVD3, 272–274
Vega.js, 274–277
VML (Vector Markup Language), 23, 260, 261
Voldemort, 216

W
warehousing
Hadoop, 218–223
Lambda Architecture, 223
watches, ZooKeeper, 41
wearables, 6
web applications, 228–229
streaming
backend server communication, 242–254
dashboard example, 238–242
Node, 229–238
weighted moving average, 370–372
Wikipedia edit stream, 282–285
Word Count example, 149–150

X
XMLHttpRequest (XHR), 22–23
XOR (exclusive-or) pattern, 387–388

Y
Yahoo!
Netty transport, 123
Storm-YARN project, 151
ZooKeeper. See ZooKeeper
YARN, 151
architecture, 152–153
background, 151–152
multinode Samza, 155–157
relationship to Samza, 153
single node Samza, 153–155

Z
Zab algorithm, 41–42
zero-th generation systems, 18
zigzag algorithm, 321
ZINCRBY command, Redis, 176
ZINTERSTORE command, Redis, 175–176
znodes, 39–41
Curator framework
ephemeral, 40
operations, 40
persistent, 40
sequential, 40–41
version number, 41
ZooKeeper, 39
clusters, creating, 42–47
consistency, 41
Curator client, 56–63
adding ZooKeeper to Maven projects, 56–57
connecting, 57–59
using watches with, 62–63
working with znodes, 59–62
Curator recipes, 63–70
distributed queues, 63–68
leader elections, 68–70
installing, 42–44
Java client, 47–56
adding ZooKeeper to Maven projects, 47–48
connecting, 48–56
notifications, 41
quorum
choosing size, 44
monitoring, 45–47
watches, 41, 62–63
znodes. See znodes
ZRANGEBYSCORE command, Redis, 176
ZREVRANGEBYSCORE command, Redis, 176
ZUNIONSTORE command, Redis, 175–176