**Big Data**

A curated list of awesome big data frameworks, resources and other awesomeness. Inspired by [awesome-php](https://github.com/ziadoz/awesome-php), [awesome-python](https://github.com/vinta/awesome-python), [awesome-ruby](https://github.com/Sdogruyol/awesome-ruby), [hadoopecosystemtable](http://hadoopecosystemtable.github.io/) & [big-data](http://usefulstuff.io/big-data/).

Your contributions are always welcome!

* [Awesome Big Data](https://github.com/onurakpolat/awesome-bigdata#awesome-bigdata)
  + [RDBMS](https://github.com/onurakpolat/awesome-bigdata#rdbms)
  + [Frameworks](https://github.com/onurakpolat/awesome-bigdata#frameworks)
  + [Distributed Programming](https://github.com/onurakpolat/awesome-bigdata#distributed-programming)
  + [Distributed Filesystem](https://github.com/onurakpolat/awesome-bigdata#distributed-filesystem)
  + [Key-Map Data Model](https://github.com/onurakpolat/awesome-bigdata#key-map-data-model)
  + [Document Data Model](https://github.com/onurakpolat/awesome-bigdata#document-data-model)
  + [Key-value Data Model](https://github.com/onurakpolat/awesome-bigdata#key-value-data-model)
  + [Graph Data Model](https://github.com/onurakpolat/awesome-bigdata#graph-data-model)
  + [NewSQL Databases](https://github.com/onurakpolat/awesome-bigdata#newsql-databases)
  + [Columnar Databases](https://github.com/onurakpolat/awesome-bigdata#columnar-databases)
  + [Time-Series Databases](https://github.com/onurakpolat/awesome-bigdata#time-series-databases)
  + [SQL-like processing](https://github.com/onurakpolat/awesome-bigdata#sql-like-processing)
  + [Data Ingestion](https://github.com/onurakpolat/awesome-bigdata#data-ingestion)
  + [Service Programming](https://github.com/onurakpolat/awesome-bigdata#service-programming)
  + [Scheduling](https://github.com/onurakpolat/awesome-bigdata#scheduling)
  + [Machine Learning](https://github.com/onurakpolat/awesome-bigdata#machine-learning)
  + [Benchmarking](https://github.com/onurakpolat/awesome-bigdata#benchmarking)
  + [Security](https://github.com/onurakpolat/awesome-bigdata#security)
  + [System Deployment](https://github.com/onurakpolat/awesome-bigdata#system-deployment)
  + [Applications](https://github.com/onurakpolat/awesome-bigdata#applications)
  + [Search engine and framework](https://github.com/onurakpolat/awesome-bigdata#search-engine-and-framework)
  + [MySQL forks and evolutions](https://github.com/onurakpolat/awesome-bigdata#mysql-forks-and-evolutions)
  + [PostgreSQL forks and evolutions](https://github.com/onurakpolat/awesome-bigdata#postgresql-forks-and-evolutions)
  + [Memcached forks and evolutions](https://github.com/onurakpolat/awesome-bigdata#memcached-forks-and-evolutions)
  + [Embedded Databases](https://github.com/onurakpolat/awesome-bigdata#embedded-databases)
  + [Business Intelligence](https://github.com/onurakpolat/awesome-bigdata#business-intelligence)
  + [Data Visualization](https://github.com/onurakpolat/awesome-bigdata#data-visualization)
  + [Internet of things and sensor data](https://github.com/onurakpolat/awesome-bigdata#internet-of-things-and-sensor-data)
  + [Interesting Readings](https://github.com/onurakpolat/awesome-bigdata#interesting-readings)
  + [Interesting Papers](https://github.com/onurakpolat/awesome-bigdata#interesting-papers)
  + [Videos](https://github.com/onurakpolat/awesome-bigdata#videos)
  + [Books](https://github.com/onurakpolat/awesome-bigdata#books)
* [Other Awesome Lists](https://github.com/onurakpolat/awesome-bigdata#other-awesome-lists)

**RDBMS**

* [MySQL](https://www.mysql.com/) - The world's most popular open source database.
* [PostgreSQL](https://www.postgresql.org/) - The world's most advanced open source database.
* [Oracle Database](http://www.oracle.com/us/corporate/features/database-12c/index.html) - object-relational database management system.
* [Teradata](http://www.teradata.com/products-and-services/teradata-database/) - High-performance MPP data warehouse platform.

**Frameworks**

* [IBM Streams](https://www.ibm.com/analytics/us/en/technology/stream-computing/) - Platform for distributed processing and real-time analytics. Integrates with many of the popular technologies in the Big Data ecosystem (Kafka, HDFS, Spark, etc.)
* [Apache Hadoop](http://hadoop.apache.org/) - Framework for distributed processing. Integrates MapReduce (parallel processing), YARN (job scheduling) and HDFS (distributed file system).
* [Tigon](https://github.com/caskdata/tigon) - High Throughput Real-time Stream Processing Framework.
* [Pachyderm](http://pachyderm.io/) - Pachyderm is a data storage platform built on Docker and Kubernetes to provide reproducible data processing and analysis.

**Distributed Programming**

* [AddThis Hydra](https://github.com/addthis/hydra) - Distributed data processing and storage system originally developed at AddThis.
* [AMPLab SIMR](http://databricks.github.io/simr/) - Run Spark on Hadoop MapReduce v1.
* [Apache APEX](https://apex.apache.org/) - A unified, enterprise platform for big data stream and batch processing.
* [Apache Beam](http://incubator.apache.org/projects/beam.html) - An unified model and set of language-specific SDKs for defining and executing data processing workflows.
* [Apache Crunch](http://crunch.apache.org/) - A simple Java API for tasks like joining and data aggregation that are tedious to implement on plain MapReduce.
* [Apache DataFu](http://incubator.apache.org/projects/datafu.html) - Collection of user-defined functions for Hadoop and Pig developed by LinkedIn.
* [Apache Flink](http://flink.apache.org/) - High-performance runtime, and automatic program optimization.
* [Apache Gearpump](http://gearpump.apache.org/) - Real-time big data streaming engine based on Akka.
* [Apache Gora](http://gora.apache.org/) - Framework for in-memory data model and persistence.
* [Apache Hama](http://hama.apache.org/) - BSP (Bulk Synchronous Parallel) computing framework.
* [Apache MapReduce](https://wiki.apache.org/hadoop/MapReduce/) - Programming model for processing large data sets with a parallel, distributed algorithm on a cluster.
* [Apache Pig](https://pig.apache.org/) - High level language to express data analysis programs for Hadoop.
* [Apache REEF](http://reef.apache.org/) - Retainable evaluator execution framework to simplify and unify the lower layers of big data systems.
* [Apache S4](http://incubator.apache.org/projects/s4.html) - Framework for stream processing, implementation of S4.
* [Apache Spark](http://spark.apache.org/) - Framework for in-memory cluster computing.
* [Apache Spark Streaming](http://spark.apache.org/docs/0.7.3/streaming-programming-guide.html) - Framework for stream processing, part of Spark.
* [Apache Storm](http://storm.apache.org/) - Framework for stream processing by Twitter also on YARN.
* [Apache Samza](http://samza.apache.org/) - Stream processing framework, based on Kafka and YARN.
* [Apache Tez](http://tez.apache.org/) - Application framework for executing a complex DAG (Directed Acyclic Graph) of tasks, built on YARN.
* [Apache Twill](https://incubator.apache.org/projects/twill.html) - Abstraction over YARN that reduces the complexity of developing distributed applications.
* [Cascalog](http://cascalog.org/) - Data processing and querying library.
* [Cheetah](http://vldbarc.org/pvldb/vldb2010/pvldb_vol3/I08.pdf) - High Performance, Custom Data Warehouse on Top of MapReduce.
* [Concurrent Cascading](http://www.cascading.org/) - Framework for data management/analytics on Hadoop.
* [Damballa Parkour](https://github.com/damballa/parkour) - MapReduce library for Clojure.
* [Datasalt Pangool](https://github.com/datasalt/pangool) - Alternative MapReduce paradigm.
* [DataTorrent StrAM](https://www.datatorrent.com/) - Real-time engine is designed to enable distributed, asynchronous, real time in-memory big-data computations in as unblocked a way as possible, with minimal overhead and impact on performance.
* [Facebook Corona](https://www.facebook.com/notes/facebook-engineering/under-the-hood-scheduling-mapreduce-jobs-more-efficiently-with-corona/10151142560538920) - Hadoop enhancement which removes single point of failure.
* [Facebook Peregrine](http://peregrine_mapreduce.bitbucket.org/) - Map Reduce framework.
* [Facebook Scuba](https://www.facebook.com/notes/facebook-engineering/under-the-hood-data-diving-with-scuba/10150599692628920) - Distributed in-memory datastore.
* [Google Dataflow](https://googledevelopers.blogspot.it/2014/06/cloud-platform-at-google-io-new-big.html) - Create data pipelines to help themæingest, transform and analyze data.
* [Google MapReduce](https://research.google.com/archive/mapreduce.html) - Map reduce framework.
* [Google MillWheel](https://research.google.com/pubs/pub41378.html) - Fault tolerant stream processing framework.
* [IBM Streams](https://www.ibm.com/analytics/us/en/technology/stream-computing/) - Platform for distributed processing and real-time analytics. Provides toolkits for advanced analytics like geospatial, time series, etc. out of the box.
* [JAQL](https://code.google.com/p/jaql/) - Declarative programming language for working with structured, semi-structured and unstructured data.
* [Kite](http://kitesdk.org/docs/current/) - A set of libraries, tools, examples, and documentation focused on making it easier to build systems on top of the Hadoop ecosystem.
* [Metamarkets Druid](http://druid.io/) - Framework for real-time analysis of large datasets.
* [Netflix PigPen](https://github.com/Netflix/PigPen) - map-reduce for Clojure which compiles to Apache Pig.
* [Nokia Disco](http://discoproject.org/) - MapReduce framework developed by Nokia.
* [Onyx](http://www.onyxplatform.org/) - Distributed computation for the cloud.
* [Pinterest Pinlater](https://medium.com/@Pinterest_Engineering/pinlater-an-asynchronous-job-execution-system-b8664cb8aa7d) - Asynchronous job execution system.
* [Pydoop](http://crs4.github.io/pydoop/) - Python MapReduce and HDFS API for Hadoop.
* [Rackerlabs Blueflood](http://blueflood.io/) - Multi-tenant distributed metric processing system
* [Skale](https://github.com/skale-me/skale-engine) - High performance distributed data processing in NodeJS.
* [Stratosphere](http://stratosphere.eu/) - General purpose cluster computing framework.
* [Streamdrill](https://streamdrill.com/) - Useful for counting activities of event streams over different time windows and finding the most active one.
* [streamsx.topology](https://github.com/IBMStreams/streamsx.topology) - Libraries to enable building IBM Streams application in Java, Python or Scala.
* [Tuktu](https://github.com/UnderstandLingBV/Tuktu) - Easy-to-use platform for batch and streaming computation, built using Scala, Akka and Play!
* [Twitter Heron](https://github.com/twitter/heron) - Heron is a realtime, distributed, fault-tolerant stream processing engine from Twitter replacing Storm.
* [Twitter Scalding](https://github.com/twitter/scalding) - Scala library for Map Reduce jobs, built on Cascading.
* [Twitter Summingbird](https://github.com/twitter/summingbird) - Streaming MapReduce with Scalding and Storm, by Twitter.
* [Twitter TSAR](https://blog.twitter.com/engineering/en_us/a/2014/tsar-a-timeseries-aggregator.html) - TimeSeries AggregatoR by Twitter.
* [Wallaroo](http://www.wallaroolabs.com/community) - The ultrafast and elastic data processing engine. Big or fast data - no fuss, no Java needed.

**Distributed Filesystem**

* [Ambry](https://github.com/linkedin/ambry) - a distributed object store that supports storage of trillion of small immutable objects as well as billions of large objects.
* [Apache HDFS](http://hadoop.apache.org/) - a way to store large files across multiple machines.
* [Apache Kudu](http://kudu.apache.org/) - Hadoop's storage layer to enable fast analytics on fast data.
* [BeeGFS](https://www.beegfs.io/content/) - formerly FhGFS, parallel distributed file system.
* [Ceph Filesystem](http://ceph.com/ceph-storage/file-system/) - software storage platform designed.
* [Disco DDFS](http://disco.readthedocs.org/en/latest/howto/ddfs.html) - distributed filesystem.
* [Facebook Haystack](https://www.facebook.com/note.php?note_id=76191543919) - object storage system.
* [Google Colossus](http://static.googleusercontent.com/media/research.google.reverse-proxy.org/en/us/university/relations/facultysummit2010/storage_architecture_and_challenges.pdf) - distributed filesystem (GFS2).
* [Google GFS](http://static.googleusercontent.com/media/research.google.com/en/archive/gfs-sosp2003.pdf) - distributed filesystem.
* [Google Megastore](https://research.google.com/pubs/pub36971.html) - scalable, highly available storage.
* [GridGain](https://www.gridgain.com/) - GGFS, Hadoop compliant in-memory file system.
* [Lustre file system](http://wiki.lustre.org/) - high-performance distributed filesystem.
* [Microsoft Azure Data Lake Store](https://hadoop.apache.org/docs/current/hadoop-azure-datalake/index.html) - HDFS-compatible storage in Azure cloud
* [Quantcast File System QFS](https://www.quantcast.com/about-us/quantcast-file-system/) - open-source distributed file system.
* [Red Hat GlusterFS](http://gluster.org/) - scale-out network-attached storage file system.
* [Seaweed-FS](https://github.com/chrislusf/seaweedfs) - simple and highly scalable distributed file system.
* [Alluxio](http://www.alluxio.org/) - reliable file sharing at memory speed across cluster frameworks.
* [Tahoe-LAFS](https://www.tahoe-lafs.org/trac/tahoe-lafs) - decentralized cloud storage system.
* [Baidu File System](https://github.com/baidu/bfs) - distributed filesystem.

**Document Data Model**

* [Actian Versant](https://www.actian.com/data-management/ingres-sql-rdbms/) - commercial object-oriented database management systems .
* [Crate Data](https://crate.io/) - is an open source massively scalable data store. It requires zero administration.
* [Facebook Apollo](http://www.infoq.com/news/2014/06/facebook-apollo) - Facebook’s Paxos-like NoSQL database.
* [jumboDB](http://comsysto.github.io/jumbodb/) - document oriented datastore over Hadoop.
* [LinkedIn Espresso](https://engineering.linkedin.com/data) - horizontally scalable document-oriented NoSQL data store.
* [MarkLogic](http://www.marklogic.com/) - Schema-agnostic Enterprise NoSQL database technology.
* [Microsoft Azure DocumentDB](https://azure.microsoft.com/en-us/services/cosmos-db/) - NoSQL cloud database service with protocol support for MongoDB
* [MongoDB](https://www.mongodb.com/) - Document-oriented database system.
* [RavenDB](https://ravendb.net/) - A transactional, open-source Document Database.
* [RethinkDB](https://rethinkdb.com/) - document database that supports queries like table joins and group by.

**Key Map Data Model**

**Note**: There is some term confusion in the industry, and two different things are called "Columnar Databases". Some, listed here, are distributed, persistent databases built around the "key-map" data model: all data has a (possibly composite) key, with which a map of key-value pairs is associated. In some systems, multiple such value maps can be associated with a key, and these maps are referred to as "column families" (with value map keys being referred to as "columns").

Another group of technologies that can also be called "columnar databases" is distinguished by how it stores data, on disk or in memory -- rather than storing data the traditional way, where all column values for a given key are stored next to each other, "row by row", these systems store all *column* values next to each other. So more work is needed to get all columns for a given key, but less work is needed to get all values for a given column.

The former group is referred to as "key map data model" here. The line between these and the [Key-value Data Model](https://github.com/onurakpolat/awesome-bigdata#key-value-data-model) stores is fairly blurry.

The latter, being more about the storage format than about the data model, is listed under [Columnar Databases](https://github.com/onurakpolat/awesome-bigdata#columnar-databases).

You can read more about this distinction on Prof. Daniel Abadi's blog: [Distinguishing two major types of Column Stores](http://dbmsmusings.blogspot.com/2010/03/distinguishing-two-major-types-of_29.html).

* [Apache Accumulo](http://accumulo.apache.org/) - distributed key/value store, built on Hadoop.
* [Apache Cassandra](http://cassandra.apache.org/) - column-oriented distributed datastore, inspired by BigTable.
* [Apache HBase](http://hbase.apache.org/) - column-oriented distributed datastore, inspired by BigTable.
* [Baidu Tera](https://github.com/baidu/tera) - an Internet-scale database, inspired by BigTable.
* [Facebook HydraBase](https://code.facebook.com/posts/321111638043166/hydrabase-the-evolution-of-hbase-facebook/) - evolution of HBase made by Facebook.
* [Google BigTable](http://static.googleusercontent.com/media/research.google.com/en/archive/bigtable-osdi06.pdf) - column-oriented distributed datastore.
* [Google Cloud Datastore](https://cloud.google.com/datastore/docs/concepts/overview) - is a fully managed, schemaless database for storing non-relational data over BigTable.
* [Hypertable](http://www.hypertable.org/) - column-oriented distributed datastore, inspired by BigTable.
* [InfiniDB](https://github.com/infinidb/infinidb/) - is accessed through a MySQL interface and use massive parallel processing to parallelize queries.
* [Tephra](https://github.com/caskdata/tephra) - Transactions for HBase.
* [Twitter Manhattan](https://blog.twitter.com/engineering/en_us/a/2014/manhattan-our-real-time-multi-tenant-distributed-database-for-twitter-scale.html) - real-time, multi-tenant distributed database for Twitter scale.
* [ScyllaDB](http://www.scylladb.com/) - column-oriented distributed datastore written in C++, totally compatible with Apache Cassandra.

**Key-value Data Model**

* [Aerospike](http://www.aerospike.com/) - NoSQL flash-optimized, in-memory. Open source and "Server code in 'C' (not Java or Erlang) precisely tuned to avoid context switching and memory copies."
* [Amazon DynamoDB](https://aws.amazon.com/dynamodb/) - distributed key/value store, implementation of Dynamo paper.
* [Badger](https://open.dgraph.io/post/badger/) - a fast, simple, efficient, and persistent key-value store written natively in Go.
* [Bolt](https://github.com/boltdb/bolt) - an embedded key-value database for Go.
* [BTDB](https://github.com/Bobris/BTDB) - Key Value Database in .Net with Object DB Layer, RPC, dynamic IL and much more
* [BuntDB](https://github.com/tidwall/buntdb) - a fast, embeddable, in-memory key/value database for Go with custom indexing and geospatial support.
* [Edis](https://github.com/cbd/edis) - is a protocol-compatible Server replacement for Redis.
* [ElephantDB](https://github.com/nathanmarz/elephantdb) - Distributed database specialized in exporting data from Hadoop.
* [EventStore](https://geteventstore.com/) - distributed time series database.
* [GridDB](https://github.com/griddb/griddb_nosql) - suitable for sensor data stored in a timeseries.
* [HyperDex](https://github.com/rescrv/HyperDex) - a scalable, next generation key-value and document store with a wide array of features, including consistency, fault tolerance and high performance.
* [LinkedIn Krati](https://github.com/linkedin-sna/sna-page/tree/master/krati) - is a simple persistent data store with very low latency and high throughput.
* [Linkedin Voldemort](http://www.project-voldemort.com/voldemort/) - distributed key/value storage system.
* [Oracle NoSQL Database](http://www.oracle.com/technetwork/database/database-technologies/nosqldb/overview/index.html) - distributed key-value database by Oracle Corporation.
* [Redis](https://redis.io/) - in memory key value datastore.
* [Riak](https://github.com/basho/riak) - a decentralized datastore.
* [Storehaus](https://github.com/twitter/storehaus) - library to work with asynchronous key value stores, by Twitter.
* [SummitDB](https://github.com/tidwall/summitdb) - an in-memory, NoSQL key/value database, with disk persistance and using the Raft consensus algorithm.
* [Tarantool](https://github.com/tarantool/tarantool) - an efficient NoSQL database and a Lua application server.
* [TiKV](https://github.com/pingcap/tikv) - a distributed key-value database powered by Rust and inspired by Google Spanner and HBase.
* [Tile38](https://github.com/tidwall/tile38) - a geolocation data store, spatial index, and realtime geofence, supporting a variety of object types including latitude/longitude points, bounding boxes, XYZ tiles, Geohashes, and GeoJSON
* [TreodeDB](https://github.com/Treode/store) - key-value store that's replicated and sharded and provides atomic multirow writes.

**Graph Data Model**

* [AgensGraph](http://www.agensgraph.com/) - a new generation multi-model graph database for the modern complex data environment.
* [Apache Giraph](http://giraph.apache.org/) - implementation of Pregel, based on Hadoop.
* [Apache Spark Bagel](http://spark.apache.org/docs/0.7.3/bagel-programming-guide.html) - implementation of Pregel, part of Spark.
* [ArangoDB](https://www.arangodb.com/) - multi model distributed database.
* [DGraph](https://github.com/dgraph-io/dgraph) - A scalable, distributed, low latency, high throughput graph database aimed at providing Google production level scale and throughput, with low enough latency to be serving real time user queries, over terabytes of structured data.
* [EliasDB](https://github.com/krotik/eliasdb) - a lightweight graph based database that does not require any third-party libraries.
* [Facebook TAO](https://www.facebook.com/notes/facebook-engineering/tao-the-power-of-the-graph/10151525983993920) - TAO is the distributed data store that is widely used at facebook to store and serve the social graph.
* [GCHQ Gaffer](https://github.com/gchq/Gaffer) - Gaffer by GCHQ is a framework that makes it easy to store large-scale graphs in which the nodes and edges have statistics.
* [Google Cayley](https://github.com/cayleygraph/cayley) - open-source graph database.
* [Google Pregel](http://kowshik.github.io/JPregel/pregel_paper.pdf) - graph processing framework.
* [GraphLab PowerGraph](https://turi.com/products/create/docs/) - a core C++ GraphLab API and a collection of high-performance machine learning and data mining toolkits built on top of the GraphLab API.
* [GraphX](https://amplab.cs.berkeley.edu/publication/graphx-grades/) - resilient Distributed Graph System on Spark.
* [Gremlin](https://github.com/tinkerpop/gremlin) - graph traversal Language.
* [Infovore](https://github.com/paulhoule/infovore) - RDF-centric Map/Reduce framework.
* [Intel GraphBuilder](https://01.org/graphbuilder/) - tools to construct large-scale graphs on top of Hadoop.
* [MapGraph](https://www.blazegraph.com/mapgraph-technology/) - Massively Parallel Graph processing on GPUs.
* [Neo4j](https://neo4j.com/) - graph database writting entirely in Java.
* [OrientDB](http://orientdb.com/) - document and graph database.
* [Phoebus](https://github.com/xslogic/phoebus) - framework for large scale graph processing.
* [Titan](http://thinkaurelius.github.io/titan/) - distributed graph database, built over Cassandra.
* [Twitter FlockDB](https://github.com/twitter-archive/flockdb) - distributed graph database.
* [NodeXL](https://nodexl.codeplex.com/) - A free, open-source template for Microsoft® Excel® 2007, 2010, 2013 and 2016 that makes it easy to explore network graphs.

**Columnar Databases**

**Note** please read the note on [Key-Map Data Model](https://github.com/onurakpolat/awesome-bigdata#key-map-data-model) section.

* [Columnar Storage](http://the-paper-trail.org/blog/columnar-storage/) - an explanation of what columnar storage is and when you might want it.
* [Actian Vector](http://www.actian.com/) - column-oriented analytic database.
* [C-Store](http://db.lcs.mit.edu/projects/cstore/) - column oriented DBMS.
* [ClickHouse](https://clickhouse.yandex/) - an open-source column-oriented database management system that allows generating analytical data reports in real time.
* [EventQL](http://eventql.io/) - a distributed, column-oriented database built for large-scale event collection and analytics.
* [MonetDB](https://www.monetdb.org/) - column store database.
* [Parquet](http://parquet.apache.org/) - columnar storage format for Hadoop.
* [Pivotal Greenplum](https://pivotal.io/pivotal-greenplum) - purpose-built, dedicated analytic data warehouse that offers a columnar engine as well as a traditional row-based one.
* [Vertica](https://www.vertica.com/) - is designed to manage large, fast-growing volumes of data and provide very fast query performance when used for data warehouses.
* [SQream DB](http://sqream.com/) - A GPU powered big data database, designed for analytics and data warehousing, with ANSI-92 compliant SQL, suitable for data sets from 10TB to 1PB.
* [Google BigQuery](https://cloud.google.com/bigquery/what-is-bigquery) Google's cloud offering backed by their pioneering work on Dremel.
* [Amazon Redshift](https://aws.amazon.com/redshift/) Amazon's cloud offering, also based on a columnar datastore backend.
* [IndexR](https://github.com/shunfei/indexr) an open-source columnar storage format for fast & realtime analytic with big data.

**NewSQL Databases**

* [Actian Ingres](http://www.actian.com/products/operational-databases/) - commercially supported, open-source SQL relational database management system.
* [Amazon RedShift](http://aws.amazon.com/redshift/) - data warehouse service, based on PostgreSQL.
* [BayesDB](http://probcomp.csail.mit.edu/bayesdb/index.html) - statistic oriented SQL database.
* [Bedrock](http://bedrockdb.com/) - a simple, modular, networked and distributed transaction layer built atop SQLite.
* [CitusDB](https://www.citusdata.com/) - scales out PostgreSQL through sharding and replication.
* [Cockroach](https://github.com/cockroachdb/cockroach) - Scalable, Geo-Replicated, Transactional Datastore.
* [Comdb2](https://github.com/bloomberg/comdb2) - a clustered RDBMS built on optimistic concurrency control techniques.
* [Datomic](http://www.datomic.com/) - distributed database designed to enable scalable, flexible and intelligent applications.
* [FoundationDB](https://foundationdb.com/) - distributed database, inspired by F1.
* [Google F1](https://research.google.com/pubs/pub41344.html) - distributed SQL database built on Spanner.
* [Google Spanner](https://research.google.com/archive/spanner.html) - globally distributed semi-relational database.
* [H-Store](http://hstore.cs.brown.edu/) - is an experimental main-memory, parallel database management system that is optimized for on-line transaction processing (OLTP) applications.
* [Haeinsa](https://github.com/VCNC/haeinsa) - linearly scalable multi-row, multi-table transaction library for HBase based on Percolator.
* [HandlerSocket](https://www.percona.com/doc/percona-server/5.5/performance/handlersocket.html) - NoSQL plugin for MySQL/MariaDB.
* [InfiniSQL](http://www.infinisql.org/) - infinity scalable RDBMS.
* [MemSQL](http://www.memsql.com/) - in memory SQL database witho optimized columnar storage on flash.
* [NuoDB](http://www.nuodb.com/) - SQL/ACID compliant distributed database.
* [Oracle TimesTen in-Memory Database](http://www.oracle.com/technetwork/database/database-technologies/timesten/overview/index.html) - in-memory, relational database management system with persistence and recoverability.
* [Pivotal GemFire XD](http://gemfirexd.docs.pivotal.io/latest/) - Low-latency, in-memory, distributed SQL data store. Provides SQL interface to in-memory table data, persistable in HDFS.
* [SAP HANA](https://hana.sap.com/abouthana.html) - is an in-memory, column-oriented, relational database management system.
* [SenseiDB](http://senseidb.github.io/sensei/) - distributed, realtime, semi-structured database.
* [Sky](http://skydb.io/) - database used for flexible, high performance analysis of behavioral data.
* [SymmetricDS](http://www.symmetricds.org/) - open source software for both file and database synchronization.
* [Map-D](https://www.mapd.com/) - GPU in-memory database, big data analysis and visualization platform
* [TiDB](https://github.com/pingcap/tidb) - TiDB is a distributed SQL database. Inspired by the design of Google F1.
* [VoltDB](https://www.voltdb.com/) - claims to be fastest in-memory database

**Time-Series Databases**

* [Axibase Time Series Database](http://axibase.com/products/axibase-time-series-database/) - Integrated time series database on top of HBase with built-in visualization, rule-engine and SQL support.
* [Chronix](http://chronix.io/) - a time series storage built to store time series highly compressed and for fast access times.
* [Cube](http://square.github.io/cube/) - uses MongoDB to store time series data.
* [Heroic](https://spotify.github.io/heroic/#!/index) - is a scalable time series database based on Cassandra and Elasticsearch.
* [InfluxDB](https://www.influxdata.com/) - distributed time series database.
* [Kairosdb](https://github.com/kairosdb/kairosdb) - similar to OpenTSDB but allows for Cassandra.
* [Newts](https://opennms.github.io/newts/) - a time series database based on Apache Cassandra.
* [OpenTSDB](http://opentsdb.net/) - distributed time series database on top of HBase.
* [Prometheus](https://prometheus.io/) - a time series database and service monitoring system.
* [Beringei](https://github.com/facebookincubator/beringei) - Facebook's in-memory time-series database.
* [TrailDB](http://traildb.io/) - an efficient tool for storing and querying series of events.
* [Druid](https://github.com/druid-io/druid/) Column oriented distributed data store ideal for powering interactive applications
* [Riak-TS](http://basho.com/products/riak-ts/) Riak TS is the only enterprise-grade NoSQL time series database optimized specifically for IoT and Time Series data.
* [Akumuli](https://github.com/akumuli/Akumuli) Akumuli is a numeric time-series database. It can be used to capture, store and process time-series data in real-time. The word "akumuli" can be translated from esperanto as "accumulate".
* [Rhombus](https://github.com/Pardot/Rhombus) A time-series object store for Cassandra that handles all the complexity of building wide row indexes.
* [Dalmatiner DB](https://github.com/dalmatinerdb/dalmatinerdb) Fast distributed metrics database
* [Blueflood](https://github.com/rackerlabs/blueflood) A distributed system designed to ingest and process time series data
* [Timely](https://github.com/NationalSecurityAgency/timely) Timely is a time series database application that provides secure access to time series data based on Accumulo and Grafana.
* [SiriDB](https://github.com/transceptor-technology/siridb-server) Highly-scalable, robust and fast, open source time series database with cluster functionality.

**SQL-like processing**

* [Actian SQL for Hadoop](http://www.actian.com/analytic-database/vectorh-sql-hadoop) - high performance interactive SQL access to all Hadoop data.
* [Apache Drill](http://drill.apache.org/) - framework for interactive analysis, inspired by Dremel.
* [Apache HCatalog](https://cwiki.apache.org/confluence/display/HCATALOG/Index) - table and storage management layer for Hadoop.
* [Apache Hive](http://hive.apache.org/) - SQL-like data warehouse system for Hadoop.
* [Apache Calcite](http://calcite.apache.org/) - framework that allows efficient translation of queries involving heterogeneous and federated data.
* [Apache Phoenix](http://phoenix.apache.org/index.html) - SQL skin over HBase.
* [Aster Database](http://www.teradata.com/products-and-services/Teradata-Aster/teradata-aster-database) - SQL-like analytic processing for MapReduce.
* [Cloudera Impala](https://www.cloudera.com/products/apache-hadoop/impala.html) - framework for interactive analysis, Inspired by Dremel.
* [Concurrent Lingual](http://www.cascading.org/projects/lingual/) - SQL-like query language for Cascading.
* [Datasalt Splout SQL](http://www.datasalt.com/products/splout-sql/) - full SQL query engine for big datasets.
* [Facebook PrestoDB](https://prestodb.io/) - distributed SQL query engine.
* [Google BigQuery](https://research.google.com/pubs/pub36632.html) - framework for interactive analysis, implementation of Dremel.
* [PipelineDB](https://www.pipelinedb.com/) - an open-source relational database that runs SQL queries continuously on streams, incrementally storing results in tables.
* [Pivotal HDB](https://pivotal.io/pivotal-hdb) - SQL-like data warehouse system for Hadoop.
* [RainstorDB](http://rainstor.com/products/rainstor-database/) - database for storing petabyte-scale volumes of structured and semi-structured data.
* [Spark Catalyst](https://github.com/apache/spark/tree/master/sql) - is a Query Optimization Framework for Spark and Shark.
* [SparkSQL](https://databricks.com/blog/2014/03/26/spark-sql-manipulating-structured-data-using-spark-2.html) - Manipulating Structured Data Using Spark.
* [Splice Machine](https://www.splicemachine.com/) - a full-featured SQL-on-Hadoop RDBMS with ACID transactions.
* [Stinger](https://hortonworks.com/innovation/stinger/) - interactive query for Hive.
* [Tajo](http://tajo.apache.org/) - distributed data warehouse system on Hadoop.
* [Trafodion](https://wiki.trafodion.org/wiki/index.php/Main_Page) - enterprise-class SQL-on-HBase solution targeting big data transactional or operational workloads.

**Data Ingestion**

* [Amazon Kinesis](https://aws.amazon.com/kinesis/) - real-time processing of streaming data at massive scale.
* [Apache Chukwa](http://chukwa.apache.org/) - data collection system.
* [Apache Flume](http://flume.apache.org/) - service to manage large amount of log data.
* [Apache Kafka](http://kafka.apache.org/) - distributed publish-subscribe messaging system.
* [Apache Sqoop](http://sqoop.apache.org/) - tool to transfer data between Hadoop and a structured datastore.
* [Cloudera Morphlines](https://github.com/cloudera/cdk/tree/master/cdk-morphlines) - framework that help ETL to Solr, HBase and HDFS.
* [Embulk](http://www.embulk.org/) - open-source bulk data loader that helps data transfer between various databases, storages, file formats, and cloud services.
* [Facebook Scribe](https://github.com/facebookarchive/scribe) - streamed log data aggregator.
* [Fluentd](http://www.fluentd.org/) - tool to collect events and logs.
* [Google Photon](https://research.google.com/pubs/pub41318.html) - geographically distributed system for joining multiple continuously flowing streams of data in real-time with high scalability and low latency.
* [Heka](https://github.com/mozilla-services/heka) - open source stream processing software system.
* [HIHO](https://github.com/sonalgoyal/hiho) - framework for connecting disparate data sources with Hadoop.
* [Kestrel](https://github.com/papertrail/kestrel) - distributed message queue system.
* [LinkedIn Databus](https://engineering.linkedin.com/data) - stream of change capture events for a database.
* [LinkedIn Kamikaze](https://github.com/linkedin/kamikaze) - utility package for compressing sorted integer arrays.
* [LinkedIn White Elephant](https://github.com/linkedin/white-elephant) - log aggregator and dashboard.
* [Logstash](https://www.elastic.co/products/logstash) - a tool for managing events and logs.
* [Netflix Suro](https://github.com/Netflix/suro) - log agregattor like Storm and Samza based on Chukwa.
* [Pinterest Secor](https://github.com/pinterest/secor) - is a service implementing Kafka log persistance.
* [Linkedin Gobblin](https://github.com/linkedin/gobblin) - linkedin's universal data ingestion framework.
* [Skizze](https://github.com/skizzehq/skizze) - sketch data store to deal with all problems around counting and sketching using probabilistic data-structures.
* [StreamSets Data Collector](https://github.com/streamsets/datacollector) - continuous big data ingest infrastructure with a simple to use IDE.
* [Yahoo Pulsar](https://github.com/apache/incubator-pulsar) - a distributed pub-sub messaging platform with a very flexible messaging model and an intuitive client API.
* [Alooma](https://www.alooma.com/integrations/mysql) - data pipeline as a service enabling moving data sources such as MySQL into data warehouses.

**Service Programming**

* [Akka Toolkit](http://akka.io/) - runtime for distributed, and fault tolerant event-driven applications on the JVM.
* [Apache Avro](http://avro.apache.org/) - data serialization system.
* [Apache Curator](http://curator.apache.org/) - Java libaries for Apache ZooKeeper.
* [Apache Karaf](http://karaf.apache.org/) - OSGi runtime that runs on top of any OSGi framework.
* [Apache Thrift](http://thrift.apache.org/) - framework to build binary protocols.
* [Apache Zookeeper](http://zookeeper.apache.org/) - centralized service for process management.
* [Google Chubby](https://research.google.com/archive/chubby.html) - a lock service for loosely-coupled distributed systems.
* [Hydrosphere Mist](https://github.com/Hydrospheredata/mist) - a service for exposing Apache Spark analytics jobs and machine learning models as realtime, batch or reactive web services.
* [Linkedin Norbert](https://engineering.linkedin.com/data) - cluster manager.
* [OpenMPI](https://www.open-mpi.org/) - message passing framework.
* [Serf](https://www.serf.io/) - decentralized solution for service discovery and orchestration.
* [Spotify Luigi](https://github.com/spotify/luigi) - a Python package for building complex pipelines of batch jobs. It handles dependency resolution, workflow management, visualization, handling failures, command line integration, and much more.
* [Spring XD](https://github.com/spring-projects/spring-xd) - distributed and extensible system for data ingestion, real time analytics, batch processing, and data export.
* [Twitter Elephant Bird](https://github.com/twitter/elephant-bird) - libraries for working with LZOP-compressed data.
* [Twitter Finagle](https://twitter.github.io/finagle/) - asynchronous network stack for the JVM.

**Scheduling**

* [Apache Airflow](https://github.com/apache/incubator-airflow) - a platform to programmatically author, schedule and monitor workflows.
* [Apache Aurora](http://aurora.apache.org/) - is a service scheduler that runs on top of Apache Mesos.
* [Apache Falcon](http://falcon.apache.org/) - data management framework.
* [Apache Oozie](http://oozie.apache.org/) - workflow job scheduler.
* [Azure Data Factory](https://docs.microsoft.com/en-us/azure/data-factory/data-factory-introduction) - cloud-based pipeline orchestration for on-prem, cloud and HDInsight
* [Chronos](http://mesos.github.io/chronos/) - distributed and fault-tolerant scheduler.
* [Linkedin Azkaban](https://azkaban.github.io/) - batch workflow job scheduler.
* [Schedoscope](https://github.com/ottogroup/schedoscope) - Scala DSL for agile scheduling of Hadoop jobs.
* [Sparrow](https://github.com/radlab/sparrow) - scheduling platform.

**Machine Learning**

* [Azure ML Studio](https://studio.azureml.net/) - Cloud-based AzureML, R, Python Machine Learning platform
* [brain](https://github.com/harthur/brain) - Neural networks in JavaScript.
* [Cloudera Oryx](https://github.com/cloudera/oryx) - real-time large-scale machine learning.
* [Concurrent Pattern](http://www.cascading.org/projects/pattern/) - machine learning library for Cascading.
* [convnetjs](https://github.com/karpathy/convnetjs) - Deep Learning in Javascript. Train Convolutional Neural Networks (or ordinary ones) in your browser.
* [DataVec](https://github.com/deeplearning4j/DataVec) - A vectorization and data preprocessing library for deep learning in Java and Scala. Part of the Deeplearning4j ecosystem.
* [Deeplearning4j](https://github.com/deeplearning4j) - Fast, open deep learning for the JVM (Java, Scala, Clojure). A neural network configuration layer powered by a C++ library. Uses Spark and Hadoop to train nets on multiple GPUs and CPUs.
* [Decider](https://github.com/danielsdeleo/Decider) - Flexible and Extensible Machine Learning in Ruby.
* [ENCOG](http://www.heatonresearch.com/encog/) - machine learning framework that supports a variety of advanced algorithms, as well as support classes to normalize and process data.
* [etcML](http://www.etcml.com/) - text classification with machine learning.
* [Etsy Conjecture](https://github.com/etsy/Conjecture) - scalable Machine Learning in Scalding.
* [GraphLab Create](https://dato.com/products/create/) - A machine learning platform in Python with a broad collection of ML toolkits, data engineering, and deployment tools.
* [H2O](https://github.com/h2oai/h2o-3/) - statistical, machine learning and math runtime with Hadoop. R and Python.
* [Keras](https://github.com/fchollet/keras) - An intuitive neural net API inspired by Torch that runs atop Theano and Tensorflow.
* [Mahout](http://mahout.apache.org/) - An Apache-backed machine learning library for Hadoop.
* [MLbase](http://www.mlbase.org/) - distributed machine learning libraries for the BDAS stack.
* [MLPNeuralNet](https://github.com/nikolaypavlov/MLPNeuralNet) - Fast multilayer perceptron neural network library for iOS and Mac OS X.
* [MOA](http://moa.cms.waikato.ac.nz/) - MOA performs big data stream mining in real time, and large scale machine learning.
* [MonkeyLearn](https://monkeylearn.com/) - Text mining made easy. Extract and classify data from text.
* [ND4J](https://github.com/deeplearning4j/nd4j) - A matrix library for the JVM. Numpy for Java.
* [nupic](https://github.com/numenta/nupic) - Numenta Platform for Intelligent Computing: a brain-inspired machine intelligence platform, and biologically accurate neural network based on cortical learning algorithms.
* [PredictionIO](http://predictionio.incubator.apache.org/index.html) - machine learning server buit on Hadoop, Mahout and Cascading.
* [RL4J](https://github.com/deeplearning4j/rl4j) - Reinforcement learning for Java and Scala. Includes Deep-Q learning and A3C algorithms, and integrates with Open AI's Gym. Runs in the Deeplearning4j ecosystem.
* [SAMOA](http://samoa.incubator.apache.org/) - distributed streaming machine learning framework.
* [scikit-learn](https://github.com/scikit-learn/scikit-learn) - scikit-learn: machine learning in Python.
* [Spark MLlib](http://spark.apache.org/docs/0.9.0/mllib-guide.html) - a Spark implementation of some common machine learning (ML) functionality.
* [Sibyl](https://users.soe.ucsc.edu/~niejiazhong/slides/chandra.pdf) - System for Large Scale Machine Learning at Google.
* [TensorFlow](https://github.com/tensorflow/tensorflow) - Library from Google for machine learning using data flow graphs.
* [Theano](https://github.com/theano) - A Python-focused machine learning library supported by the University of Montreal.
* [Torch](https://github.com/torch) - A deep learning library with a Lua API, supported by NYU and Facebook.
* [Velox](https://github.com/amplab/velox-modelserver) - System for serving machine learning predictions.
* [Vowpal Wabbit](https://github.com/JohnLangford/vowpal_wabbit/wiki) - learning system sponsored by Microsoft and Yahoo!.
* [WEKA](http://www.cs.waikato.ac.nz/ml/weka/) - suite of machine learning software.
* [BidMach](https://github.com/BIDData/BIDMach) - CPU and GPU-accelerated Machine Learning Library.

**Benchmarking**

* [Apache Hadoop Benchmarking](https://issues.apache.org/jira/browse/MAPREDUCE-3561) - micro-benchmarks for testing Hadoop performances.
* [Berkeley SWIM Benchmark](https://github.com/SWIMProjectUCB/SWIM/wiki) - real-world big data workload benchmark.
* [Intel HiBench](https://github.com/intel-hadoop/HiBench) - a Hadoop benchmark suite.
* [PUMA Benchmarking](https://issues.apache.org/jira/browse/MAPREDUCE-5116) - benchmark suite for MapReduce applications.
* [Yahoo Gridmix3](http://yahoohadoop.tumblr.com/post/98294079296/gridmix3-emulating-production-workload-for) - Hadoop cluster benchmarking from Yahoo engineer team.
* [Deeplearning4j Benchmarks](https://github.com/deeplearning4j/dl4j-benchmark)

**Security**

* [Apache Ranger](http://ranger.apache.org/) - Central security admin & fine-grained authorization for Hadoop
* [Apache Eagle](http://eagle.apache.org/) - real time monitoring solution
* [Apache Knox Gateway](http://knox.apache.org/) - single point of secure access for Hadoop clusters.
* [Apache Sentry](http://incubator.apache.org/projects/sentry.html) - security module for data stored in Hadoop.
* [BDA](https://github.com/kotobukki/BDA/) - The vulnerability detector for Hadoop and Spark

**System Deployment**

* [Apache Ambari](http://ambari.apache.org/) - operational framework for Hadoop mangement.
* [Apache Bigtop](http://bigtop.apache.org/) - system deployment framework for the Hadoop ecosystem.
* [Apache Helix](http://helix.apache.org/) - cluster management framework.
* [Apache Mesos](http://mesos.apache.org/) - cluster manager.
* [Apache Slider](https://github.com/apache/incubator-slider) - is a YARN application to deploy existing distributed applications on YARN.
* [Apache Whirr](http://whirr.apache.org/) - set of libraries for running cloud services.
* [Apache YARN](https://hortonworks.com/hadoop/yarn/) - Cluster manager.
* [Brooklyn](http://brooklyncentral.github.io/) - library that simplifies application deployment and management.
* [Buildoop](http://buildoop.github.io/) - Similar to Apache BigTop based on Groovy language.
* [Cloudera HUE](http://gethue.com/) - web application for interacting with Hadoop.
* [Facebook Prism](http://www.wired.com/2012/08/facebook-prism/) - multi datacenters replication system.
* [Google Borg](https://www.wired.com/2013/03/google-borg-twitter-mesos/all/) - job scheduling and monitoring system.
* [Google Omega](https://www.youtube.com/watch?v=0ZFMlO98Jkc) - job scheduling and monitoring system.
* [Hortonworks HOYA](https://hortonworks.com/blog/introducing-hoya-hbase-on-yarn/) - application that can deploy HBase cluster on YARN.
* [Marathon](https://github.com/mesosphere/marathon) - Mesos framework for long-running services.

**Applications**

* [411](https://github.com/etsy/411) - an web application for alert management resulting from scheduled searches into Elasticsearch.
* [Adobe spindle](https://github.com/adobe-research/spindle) - Next-generation web analytics processing with Scala, Spark, and Parquet.
* [Apache Kiji](http://www.kiji.org.s3-website-us-east-1.amazonaws.com/) - framework to collect and analyze data in real-time, based on HBase.
* [Apache Metron](http://metron.apache.org/) - a platform that integrates a variety of open source big data technologies in order to offer a centralized tool for security monitoring and analysis.
* [Apache Nutch](http://nutch.apache.org/) - open source web crawler.
* [Apache OODT](http://oodt.apache.org/) - capturing, processing and sharing of data for NASA's scientific archives.
* [Apache Tika](https://tika.apache.org/) - content analysis toolkit.
* [Argus](https://github.com/salesforce/Argus) - Time series monitoring and alerting platform.
* [AthenaX](https://github.com/uber/AthenaX) - a streaming analytics platform that enables users to run production-quality, large scale streaming analytics using Structured Query Language (SQL).
* [Atlas](https://github.com/Netflix/atlas) - a backend for managing dimensional time series data.
* [Countly](https://count.ly/) - open source mobile and web analytics platform, based on Node.js & MongoDB.
* [Domino](https://www.dominodatalab.com/) - Run, scale, share, and deploy models — without any infrastructure.
* [Eclipse BIRT](http://www.eclipse.org/birt/) - Eclipse-based reporting system.
* [ElastAert](https://github.com/Yelp/elastalert) - ElastAlert is a simple framework for alerting on anomalies, spikes, or other patterns of interest from data in ElasticSearch.
* [Eventhub](https://github.com/Codecademy/EventHub) - open source event analytics platform.
* [Hermes](https://github.com/allegro/hermes) - asynchronous message broker built on top of Kafka.
* [HIPI Library](http://hipi.cs.virginia.edu/) - API for performing image processing tasks on Hadoop's MapReduce.
* [Hunk](https://www.splunk.com/en_us/download/hunk.html) - Splunk analytics for Hadoop.
* [Imhotep](http://opensource.indeedeng.io/imhotep/) - Large scale analytics platform by indeed.
* [MADlib](http://madlib.incubator.apache.org/community/) - data-processing library of an RDBMS to analyze data.
* [Kapacitor](https://github.com/influxdata/kapacitor) - an open source framework for processing, monitoring, and alerting on time series data.
* [Kylin](http://kylin.apache.org/) - open source Distributed Analytics Engine from eBay.
* [PivotalR](https://github.com/pivotalsoftware/PivotalR) - R on Pivotal HD / HAWQ and PostgreSQL.
* [Rakam](https://github.com/rakam-io/rakam) - open-source real-time custom analytics platform powered by Postgresql, Kinesis and PrestoDB.
* [Qubole](https://www.qubole.com/) - auto-scaling Hadoop cluster, built-in data connectors.
* [Sense](https://sense.io/) - Cloud Platform for Data Science and Big Data Analytics.
* [SnappyData](https://github.com/SnappyDataInc/snappydata) - a distributed in-memory data store for real-time operational analytics, delivering stream analytics, OLTP (online transaction processing) and OLAP (online analytical processing) built on Spark in a single integrated cluster.
* [Snowplow](https://github.com/snowplow/snowplow) - enterprise-strength web and event analytics, powered by Hadoop, Kinesis, Redshift and Postgres.
* [SparkR](http://amplab-extras.github.io/SparkR-pkg/) - R frontend for Spark.
* [Splunk](https://www.splunk.com/) - analyzer for machine-generated data.
* [Sumo Logic](https://www.sumologic.com/) - cloud based analyzer for machine-generated data.
* [Talend](http://www.talend.com/products/big-data/) - unified open source environment for YARN, Hadoop, HBASE, Hive, HCatalog & Pig.
* [Warp](https://warp.one/) - query by example tool for big data (OS X app)

**Search engine and framework**

* [Apache Lucene](http://lucene.apache.org/) - Search engine library.
* [Apache Solr](http://lucene.apache.org/solr/) - Search platform for Apache Lucene.
* [Elassandra](https://github.com/strapdata/elassandra) - is a fork of Elasticsearch modified to run on top of Apache Cassandra in a scalable and resilient peer-to-peer architecture.
* [ElasticSearch](https://www.elastic.co/) - Search and analytics engine based on Apache Lucene.
* [Enigma.io](https://www.enigma.com/) – Freemium robust web application for exploring, filtering, analyzing, searching and exporting massive datasets scraped from across the Web.
* [Facebook Unicorn](https://www.facebook.com/publications/219621248185635/) - social graph search platform.
* [Google Caffeine](https://googleblog.blogspot.it/2010/06/our-new-search-index-caffeine.html) - continuous indexing system.
* [Google Percolator](https://research.google.com/pubs/pub36726.html) - continuous indexing system.
* [TeraGoogle](https://github.com/onurakpolat/awesome-bigdata/blob/master) - large search index.
* [HBase Coprocessor](https://blogs.apache.org/hbase/entry/coprocessor_introduction) - implementation of Percolator, part of HBase.
* [Lily HBase Indexer](http://ngdata.github.io/hbase-indexer/) - quickly and easily search for any content stored in HBase.
* [LinkedIn Bobo](http://senseidb.github.io/bobo/) - is a Faceted Search implementation written purely in Java, an extension to Apache Lucene.
* [LinkedIn Cleo](https://github.com/linkedin/cleo) - is a flexible software library for enabling rapid development of partial, out-of-order and real-time typeahead search.
* [LinkedIn Galene](https://engineering.linkedin.com/search/did-you-mean-galene) - search architecture at LinkedIn.
* [LinkedIn Zoie](https://github.com/senseidb/zoie) - is a realtime search/indexing system written in Java.
* [MG4J](http://mg4j.di.unimi.it/) - MG4J (Managing Gigabytes for Java) is a full-text search engine for large document collections written in Java. It is highly customisable, high-performance and provides state-of-the-art features and new research algorithms.
* [Sphinx Search Server](http://sphinxsearch.com/) - fulltext search engine.

**MySQL forks and evolutions**

* [Amazon RDS](https://aws.amazon.com/rds/) - MySQL databases in Amazon's cloud.
* [Drizzle](http://www.drizzle.org/) - evolution of MySQL 6.0.
* [Google Cloud SQL](https://cloud.google.com/sql/docs/) - MySQL databases in Google's cloud.
* [MariaDB](https://mariadb.org/) - enhanced, drop-in replacement for MySQL.
* [MySQL Cluster](https://www.mysql.com/products/cluster/) - MySQL implementation using NDB Cluster storage engine.
* [Percona Server](https://www.percona.com/software/mysql-database/percona-server) - enhanced, drop-in replacement for MySQL.
* [ProxySQL](https://github.com/renecannao/proxysql) - High Performance Proxy for MySQL.
* [TokuDB](https://www.percona.com/) - TokuDB is a storage engine for MySQL and MariaDB.
* [WebScaleSQL](http://webscalesql.org/) - is a collaboration among engineers from several companies that face similar challenges in running MySQL at scale.

**PostgreSQL forks and evolutions**

* [HadoopDB](http://db.cs.yale.edu/hadoopdb/hadoopdb.html) - hybrid of MapReduce and DBMS.
* [IBM Netezza](http://www-01.ibm.com/software/data/netezza/) - high-performance data warehouse appliances.
* [Postgres-XL](http://www.postgres-xl.org/) - Scalable Open Source PostgreSQL-based Database Cluster.
* [RecDB](http://www-users.cs.umn.edu/~sarwat/RecDB/) - Open Source Recommendation Engine Built Entirely Inside PostgreSQL.
* [Stado](http://www.stormdb.com/community/stado) - open source MPP database system solely targeted at data warehousing and data mart applications.
* [Yahoo Everest](https://www.scribd.com/doc/3159239/70-Everest-PGCon-RT) - multi-peta-byte database / MPP derived by PostgreSQL.
* [TimescaleDB](http://www.timescale.com/) - An open-source time-series database optimized for fast ingest and complex queries
* [PipelineDB](https://www.pipelinedb.com/) - The Streaming SQL Database. An open-source relational database that runs SQL queries continuously on streams, incrementally storing results in tables

**Memcached forks and evolutions**

* [Facebook McDipper](https://www.facebook.com/notes/facebook-engineering/mcdipper-a-key-value-cache-for-flash-storage/10151347090423920) - key/value cache for flash storage.
* [Facebook Memcached](https://www.facebook.com/notes/facebook-engineering/scaling-memcache-at-facebook/10151411410803920) - fork of Memcache.
* [Twemproxy](https://github.com/twitter/twemproxy) - A fast, light-weight proxy for memcached and redis.
* [Twitter Fatcache](https://github.com/twitter/fatcache) - key/value cache for flash storage.
* [Twitter Twemcache](https://github.com/twitter/twemcache) - fork of Memcache.

**Embedded Databases**

* [Actian PSQL](http://www.actian.com/products/operational-databases/) - ACID-compliant DBMS developed by Pervasive Software, optimized for embedding in applications.
* [BerkeleyDB](https://www.oracle.com/database/berkeley-db/index.html) - a software library that provides a high-performance embedded database for key/value data.
* [HanoiDB](https://github.com/krestenkrab/hanoidb) - Erlang LSM BTree Storage.
* [LevelDB](https://github.com/google/leveldb) - a fast key-value storage library written at Google that provides an ordered mapping from string keys to string values.
* [LMDB](https://symas.com/mdb/) - ultra-fast, ultra-compact key-value embedded data store developed by Symas.
* [RocksDB](http://rocksdb.org/) - embeddable persistent key-value store for fast storage based on LevelDB.

**Business Intelligence**

* [BIME Analytics](https://www.bimeanalytics.com/?lang=en) - business intelligence platform in the cloud.
* [Chartio](https://chartio.com/) - lean business intelligence platform to visualize and explore your data.
* [datapine](https://www.datapine.com/) - self-service business intelligence tool in the cloud.
* [GoodData](https://www.gooddata.com/) - platform for data products and embedded analytics.
* [Jaspersoft](https://www.jaspersoft.com/) - powerful business intelligence suite.
* [Jedox Palo](https://www.jedox.com/en/) - customisable Business Intelligence platform.
* [Jethrodata](https://jethro.io/) - Interactive Big Data Analytics.
* [Microsoft](http://www.microsoft.com/en-us/server-cloud/solutions/business-intelligence/default.aspx) - business intelligence software and platform.
* [Microstrategy](https://www.microstrategy.com/) - software platforms for business intelligence, mobile intelligence, and network applications.
* [Pentaho](http://www.pentaho.com/) - business intelligence platform.
* [Qlik](http://www.qlik.com/us/) - business intelligence and analytics platform.
* [Redash](https://redash.io/) - Open source business intelligence platform, supporting multiple data sources and planned queries.
* [Saiku](http://www.meteorite.bi/) - open source analytics platform.
* [SpagoBI](http://www.spagobi.org/) - open source business intelligence platform.
* [SparklineData SNAP](http://sparklinedata.com/) - modern B.I platform powered by Apache Spark.
* [Tableau](https://www.tableau.com/) - business intelligence platform.
* [Zoomdata](https://www.zoomdata.com/) - Big Data Analytics.
* [Metabase](https://github.com/metabase/metabase) - The simplest, fastest way to get business intelligence and analytics to everyone in your company

**Data Visualization**

* [Airpal](https://github.com/airbnb/airpal) - Web UI for PrestoDB.
* [AnyChart](http://www.anychart.com/) - fast, simple and flexible JavaScript (HTML5) charting library featuring pure JS API.
* [Arbor](https://github.com/samizdatco/arbor) - graph visualization library using web workers and jQuery.
* [Banana](https://github.com/LucidWorks/banana) - visualize logs and time-stamped data stored in Solr. Port of Kibana.
* [Bloomery](https://github.com/ufukomer/bloomery) - Web UI for Impala.
* [Bokeh](http://bokeh.pydata.org/en/latest/) - A powerful Python interactive visualization library that targets modern web browsers for presentation, with the goal of providing elegant, concise construction of novel graphics in the style of D3.js, but also delivering this capability with high-performance interactivity over very large or streaming datasets.
* [C3](http://c3js.org/) - D3-based reusable chart library
* [CartoDB](https://github.com/CartoDB/cartodb) - open-source or freemium hosting for geospatial databases with powerful front-end editing capabilities and a robust API.
* [chartd](http://chartd.co/) - responsive, retina-compatible charts with just an img tag.
* [Chart.js](http://www.chartjs.org/) - open source HTML5 Charts visualizations.
* [Chartist.js](https://github.com/gionkunz/chartist-js) - another open source HTML5 Charts visualization.
* [Crossfilter](http://square.github.io/crossfilter/) - JavaScript library for exploring large multivariate datasets in the browser. Works well with dc.js and d3.js.
* [Cubism](https://github.com/square/cubism) - JavaScript library for time series visualization.
* [Cytoscape](http://cytoscape.github.io/) - JavaScript library for visualizing complex networks.
* [DC.js](http://dc-js.github.io/dc.js/) - Dimensional charting built to work natively with crossfilter rendered using d3.js. Excellent for connecting charts/additional metadata to hover events in D3.
* [D3](https://d3js.org/) - javaScript library for manipulating documents.
* [D3.compose](https://github.com/CSNW/d3.compose) - Compose complex, data-driven visualizations from reusable charts and components.
* [D3Plus](http://d3plus.org/) - A fairly robust set of reusable charts and styles for d3.js.
* [Echarts](https://github.com/ecomfe/echarts) - Baidus enterprise charts.
* [Envisionjs](https://github.com/HumbleSoftware/envisionjs) - dynamic HTML5 visualization.
* [FnordMetric](https://metrictools.org/) - write SQL queries that return SVG charts rather than tables
* [Freeboard](https://github.com/Freeboard/freeboard) - pen source real-time dashboard builder for IOT and other web mashups.
* [Gephi](https://github.com/gephi/gephi) - An award-winning open-source platform for visualizing and manipulating large graphs and network connections. It's like Photoshop, but for graphs. Available for Windows and Mac OS X.
* [Google Charts](https://developers.google.com/chart/) - simple charting API.
* [Grafana](https://grafana.com/) - graphite dashboard frontend, editor and graph composer.
* [Graphite](http://graphiteapp.org/) - scalable Realtime Graphing.
* [Highcharts](https://www.highcharts.com/) - simple and flexible charting API.
* [IPython](http://ipython.org/) - provides a rich architecture for interactive computing.
* [Kibana](https://www.elastic.co/products/kibana) - visualize logs and time-stamped data
* [Lumify](http://lumify.io/) - open source big data analysis and visualization platform
* [Matplotlib](https://github.com/matplotlib/matplotlib) - plotting with Python.
* [Metricsgraphic.js](https://metricsgraphicsjs.org/) - a library built on top of D3 that is optimized for time-series data
* [NVD3](http://nvd3.org/) - chart components for d3.js.
* [Peity](https://github.com/benpickles/peity) - Progressive SVG bar, line and pie charts.
* [Plot.ly](https://plot.ly/) - Easy-to-use web service that allows for rapid creation of complex charts, from heatmaps to histograms. Upload data to create and style charts with Plotly's online spreadsheet. Fork others' plots.
* [Plotly.js](https://github.com/plotly/plotly.js) The open source javascript graphing library that powers plotly.
* [Recline](https://github.com/okfn/recline) - simple but powerful library for building data applications in pure Javascript and HTML.
* [Redash](https://github.com/getredash/redash) - open-source platform to query and visualize data.
* [ReCharts](http://recharts.org/) - A composable charting library built on React components
* [Shiny](http://shiny.rstudio.com/) - a web application framework for R.
* [Sigma.js](https://github.com/jacomyal/sigma.js) - JavaScript library dedicated to graph drawing.
* [Superset](https://github.com/apache/incubator-superset) - a data exploration platform designed to be visual, intuitive and interactive, making it easy to slice, dice and visualize data and perform analytics at the speed of thought.
* [Vega](https://github.com/trifacta/vega) - a visualization grammar.
* [Zeppelin](https://github.com/ZEPL/zeppelin) - a notebook-style collaborative data analysis.
* [Zing Charts](https://www.zingchart.com/) - JavaScript charting library for big data.

**Internet of things and sensor data**

* [Apache Edgent (Incubating)](http://edgent.apache.org/) - a programming model and micro-kernel style runtime that can be embedded in gateways and small footprint edge devices enabling local, real-time, analytics on the edge devices.
* [Azure IoT Hub](https://azure.microsoft.com/en-us/services/iot-hub/) - Cloud-based bi-directional monitoring and messaging hub
* [TempoIQ](https://www.tempoiq.com/) - Cloud-based sensor analytics.
* [2lemetry](http://2lemetry.com/) - Platform for Internet of things.
* [Pubnub](https://www.pubnub.com/) - Data stream network
* [ThingWorx](https://www.thingworx.com/) - Rapid development and connection of intelligent systems
* [IFTTT](https://ifttt.com/) - If this then that
* [Evrything](https://evrythng.com/)- Making products smart

**Interesting Readings**

* [Big Data Benchmark](https://amplab.cs.berkeley.edu/benchmark/) - Benchmark of Redshift, Hive, Shark, Impala and Stiger/Tez.
* [NoSQL Comparison](https://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis) - Cassandra vs MongoDB vs CouchDB vs Redis vs Riak vs HBase vs Couchbase vs Neo4j vs Hypertable vs ElasticSearch vs Accumulo vs VoltDB vs Scalaris comparison.
* [Monitoring Kafka performance](https://www.datadoghq.com/blog/monitoring-kafka-performance-metrics?ref=awesome) - Guide to monitoring Apache Kafka, including native methods for metrics collection.
* [Monitoring Hadoop performance](https://www.datadoghq.com/blog/monitor-hadoop-metrics?ref=awesome) - Guide to monitoring Hadoop, with an overview of Hadoop architecture, and native methods for metrics collection.

**Interesting Papers**

**2015 - 2016**

* [2015](http://www.vldb.org/pvldb/vol8/p1804-ching.pdf) - **Facebook** - One Trillion Edges: Graph Processing at Facebook-Scale.

**2013 - 2014**

* [2014](http://infolab.stanford.edu/~ullman/mmds/book.pdf) - **Stanford** - Mining of Massive Datasets.
* [2013](https://amplab.cs.berkeley.edu/wp-content/uploads/2013/03/eurosys13-paper83.pdf) - **AMPLab** - Presto: Distributed Machine Learning and Graph Processing with Sparse Matrices.
* [2013](https://amplab.cs.berkeley.edu/wp-content/uploads/2013/01/dmx1.pdf) - **AMPLab** - MLbase: A Distributed Machine-learning System.
* [2013](https://amplab.cs.berkeley.edu/wp-content/uploads/2013/02/shark_sigmod2013.pdf) - **AMPLab** - Shark: SQL and Rich Analytics at Scale.
* [2013](https://amplab.cs.berkeley.edu/wp-content/uploads/2013/05/grades-graphx_with_fonts.pdf) - **AMPLab** - GraphX: A Resilient Distributed Graph System on Spark.
* [2013](http://static.googleusercontent.com/media/research.google.com/en/pubs/archive/40671.pdf) - **Google** - HyperLogLog in Practice: Algorithmic Engineering of a State of The Art Cardinality Estimation Algorithm.
* [2013](http://research.microsoft.com/pubs/200169/now-vldb.pdf) - **Microsoft** - Scalable Progressive Analytics on Big Data in the Cloud.
* [2013](http://static.druid.io/docs/druid.pdf) - **Metamarkets** - Druid: A Real-time Analytical Data Store.
* [2013](http://db.disi.unitn.eu/pages/VLDBProgram/pdf/industry/p764-rae.pdf) - **Google** - Online, Asynchronous Schema Change in F1.
* [2013](http://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/41344.pdf) - **Google** - F1: A Distributed SQL Database That Scales.
* [2013](http://db.disi.unitn.eu/pages/VLDBProgram/pdf/industry/p734-akidau.pdf) - **Google** - MillWheel: Fault-Tolerant Stream Processing at Internet Scale.
* [2013](http://db.disi.unitn.eu/pages/VLDBProgram/pdf/industry/p767-wiener.pdf) - **Facebook** - Scuba: Diving into Data at Facebook.
* [2013](http://db.disi.unitn.eu/pages/VLDBProgram/pdf/industry/p871-curtiss.pdf) - **Facebook** - Unicorn: A System for Searching the Social Graph.
* [2013](https://www.usenix.org/system/files/conference/nsdi13/nsdi13-final170_update.pdf) - **Facebook** - Scaling Memcache at Facebook.

**2011 - 2012**

* [2012](http://vldb.org/pvldb/vol5/p1771_georgelee_vldb2012.pdf) - **Twitter** - The Unified Logging Infrastructure for Data Analytics at Twitter.
* [2012](https://amplab.cs.berkeley.edu/wp-content/uploads/2013/04/blinkdb_vldb12_demo.pdf) - **AMPLab** - Blink and It’s Done: Interactive Queries on Very Large Data.
* [2012](https://www.usenix.org/system/files/login/articles/zaharia.pdf) - **AMPLab** - Fast and Interactive Analytics over Hadoop Data with Spark.
* [2012](https://amplab.cs.berkeley.edu/wp-content/uploads/2012/03/mod482-xin1.pdf) - **AMPLab** - Shark: Fast Data Analysis Using Coarse-grained Distributed Memory.
* [2012](https://www.usenix.org/legacy/event/nsdi11/tech/full_papers/Bolosky.pdf) - **Microsoft** - Paxos Replicated State Machines as the Basis of a High-Performance Data Store.
* [2012](http://research.microsoft.com/pubs/178045/ppaoxs-paper29.pdf) - **Microsoft** - Paxos Made Parallel.
* [2012](https://arxiv.org/pdf/1203.5485.pdf) - **AMPLab** - BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data.
* [2012](http://vldb.org/pvldb/vol5/p1436_alexanderhall_vldb2012.pdf) - **Google** - Processing a trillion cells per mouse click.
* [2012](http://static.googleusercontent.com/media/research.google.com/en/archive/spanner-osdi2012.pdf) - **Google** - Spanner: Google’s Globally-Distributed Database.
* [2011](https://amplab.cs.berkeley.edu/wp-content/uploads/2011/06/euro118-ananthanarayanan.pdf) - **AMPLab** - Scarlett: Coping with Skewed Popularity Content in MapReduce Clusters.
* [2011](https://amplab.cs.berkeley.edu/wp-content/uploads/2011/06/Mesos-A-Platform-for-Fine-Grained-Resource-Sharing-in-the-Data-Center.pdf) - **AMPLab** - Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center.
* [2011](http://static.googleusercontent.com/media/research.google.com/en/pubs/archive/36971.pdf) - **Google** - Megastore: Providing Scalable, Highly Available Storage for Interactive Services.

**2001 - 2010**

* [2010](https://www.usenix.org/legacy/event/osdi10/tech/full_papers/Beaver.pdf) - **Facebook** - Finding a needle in Haystack: Facebook’s photo storage.
* [2010](https://amplab.cs.berkeley.edu/wp-content/uploads/2011/06/Spark-Cluster-Computing-with-Working-Sets.pdf) - **AMPLab** - Spark: Cluster Computing with Working Sets.
* [2010](http://kowshik.github.io/JPregel/pregel_paper.pdf) - **Google** - Pregel: A System for Large-Scale Graph Processing.
* [2010](http://static.googleusercontent.com/media/research.google.com/en/pubs/archive/36726.pdf) - **Google** - Large-scale Incremental Processing Using Distributed Transactions and Notiﬁcations base of Percolator and Caffeine.
* [2010](http://static.googleusercontent.com/media/research.google.com/en/pubs/archive/36632.pdf) - **Google** - Dremel: Interactive Analysis of Web-Scale Datasets.
* [2010](http://leoneu.github.io/) - **Yahoo** - S4: Distributed Stream Computing Platform.
* [2009](http://www.vldb.org/pvldb/2/vldb09-861.pdf) - HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads.
* [2008](http://www.cca08.org/papers/Paper-13-Ariel-Rabkin.pdf) - **AMPLab** - Chukwa: A large-scale monitoring system.
* [2007](http://www.read.seas.harvard.edu/~kohler/class/cs239-w08/decandia07dynamo.pdf) - **Amazon** - Dynamo: Amazon’s Highly Available Key-value Store.
* [2006](http://static.googleusercontent.com/media/research.google.com/en/archive/chubby-osdi06.pdf) - **Google** - The Chubby lock service for loosely-coupled distributed systems.
* [2006](http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/archive/bigtable-osdi06.pdf) - **Google** - Bigtable: A Distributed Storage System for Structured Data.
* [2004](http://static.googleusercontent.com/media/research.google.com/en/archive/mapreduce-osdi04.pdf) - **Google** - MapReduce: Simplied Data Processing on Large Clusters.
* [2003](http://static.googleusercontent.com/media/research.google.com/en/archive/gfs-sosp2003.pdf) - **Google** - The Google File System.

**Videos**

**Books**

**Streaming**

* [Streaming Data](https://www.manning.com/books/streaming-data) - Streaming Data introduces the concepts and requirements of streaming and real-time data systems.
* [Storm Applied](https://www.manning.com/books/storm-applied) - Storm Applied is a practical guide to using Apache Storm for the real-world tasks associated with processing and analyzing real-time data streams.
* [Fundamentals of Stream Processing: Application Design, Systems, and Analytics](http://www.cambridge.org/us/academic/subjects/engineering/communications-and-signal-processing/fundamentals-stream-processing-application-design-systems-and-analytics) - This comprehensive, hands-on guide combining the fundamental building blocks and emerging research in stream processing is ideal for application designers, system builders, analytic developers, as well as students and researchers in the field.
* [Stream Data Processing: A Quality of Service Perspective](http://www.springer.com/us/book/9780387710020) - Presents a new paradigm suitable for stream and complex event processing.
* [Unified Log Processing](https://www.manning.com/books/event-streams-in-action) - Unified Log Processing is a practical guide to implementing a unified log of event streams (Kafka or Kinesis) in your business
* [Kafka Streams in Action](https://www.manning.com/books/kafka-streams-in-action) - Kafka Streams in Action teaches you everything you need to know to implement stream processing on data flowing into your Kafka platform, allowing you to focus on getting more from your data without sacrificing time or effort.
* [Big Data](https://www.manning.com/books/big-data) - Big Data teaches you to build big data systems using an architecture that takes advantage of clustered hardware along with new tools designed specifically to capture and analyze web-scale data.
* [Spark in Action](https://www.manning.com/books/spark-in-action) - Spark in Action teaches you the theory and skills you need to effectively handle batch and streaming data using Spark. Fully updated for Spark 2.0.
* [Kafka in Action](https://www.manning.com/books/kafka-in-action) - Kafka in Action is a fast-paced introduction to every aspect of working with Kafka you need to really reap its benefits.
* [Reactive Data Handling](https://www.manning.com/books/reactive-data-handling) - Reactive Data Handling is a collection of five hand-picked chapters, selected by Manuel Bernhardt, that introduce you to building reactive applications capable of handling real-time processing with large data loads--free eBook!

**Distributed systems**

* [Distributed Systems for fun and profit](http://book.mixu.net/distsys/) – Theory of distributed systems. Include parts about time and ordering, replication and impossibility results.

**Data Visualization**

* [The beauty of data visualization](https://www.youtube.com/watch?v=5Zg-C8AAIGg)
* [Designing Data Visualizations with Noah Iliinsky](https://www.youtube.com/watch?v=R-oiKt7bUU8)
* [Hans Rosling's 200 Countries, 200 Years, 4 Minutes](https://www.youtube.com/watch?v=jbkSRLYSojo)
* [Ice Bucket Challenge Data Visualization](https://www.youtube.com/watch?v=qTEchen97rQ)