[**Hadoop/Hbase**](http://alans.se/blog/2010/hadoop-hbase-cygwin-windows-7-x64/#.U-m2TvmSxic)

Use Apache HBase when you need random, real-time read/write access to your Big Data. This project’s goal is the hosting of very large tables billions of rows X millions of columns atop clusters of commodity hardware. Apache HBase is an open-source, distributed, versioned, non-relational database modeled after Google’s Bigtable: A Distributed Storage System for Structured Data by Chang et al. Just as Bigtable leverages the distributed data storage provided by the Google File System, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.

[**Cassandra**](http://cassandra.apache.org/download/)

The Apache Cassandra database is the right choice when you need scalability and high availability without compromising performance. Linear scalability and proven fault-tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. Cassandra’s support for replicating across multiple datacentres is best-in-class, providing lower latency for your users and the peace of mind of knowing that you can survive regional outages. Cassandra’s data model offers the convenience of column indexes with the performance of log-structured updates, strong support for denormalization and materialized views, and powerful built-in caching.

[**Hypertable**](http://hypertable.com/documentation/installation/quick_start_standalone)

Hypertable is a high performance, open source, massively scalable database modeled after Bigtable, Google’s proprietary, massively scalable database.  This page provides a brief overview of Hypertable, comparing it with a relational database, highlighting some of its unique features, and illustrating how it scales.

[**Accumulo**](https://accumulo.apache.org/)

Accumulo is based on Google’s BigTable design and is built on top of Apache Hadoop, Zookeeper, and Thrift. Apache Accumulo features a few novel improvements on the BigTable design in the form of cell-based access control and a server-side programming mechanism that can modify key/value pairs at various points in the data management process.

[**Amazon SimpleDB**](http://download.cnet.com/SDB-Explorer-for-Amazon-SimpleDB/3000-18500_4-76037201.html)

Amazon SimpleDB is a highly available and flexible non-relational data store that offloads the work of database administration. Developers simply store and query data items via web services requests and Amazon SimpleDB does the rest. Unbound by the strict requirements of a relational database, Amazon SimpleDB is optimized to provide high availability and flexibility, with little or no administrative burden. Behind the scenes, Amazon SimpleDB creates and manages multiple geographically distributed replicas of your data automatically to enable high availability and data durability. The service charges you only for the resources actually consumed in storing your data and serving your requests. You can change your data model on the fly, and data is automatically indexed for you. With Amazon SimpleDB, you can focus on application development without worrying about infrastructure provisioning, high availability, software maintenance, schema and index management, or performance tuning.

[**Cloud Data**](https://spideroak.com/)

Cloud Data is Distributed Large scale Structured Data Storage, and open source project implementing Google’s Bigtable. It can be found on Github. It appears to be the project of a Korean developer named YKKwon.

[**HPCC**](http://azure.microsoft.com/en-us/solutions/big-compute/)

HPCC (High-Performance Computing Cluster), also known as DAS (Data Analytics Supercomputer), is an open source, data-intensive computing system platform developed by LexisNexis Risk Solutions. The HPCC platform incorporates a software architecture implemented on commodity computing clusters to provide high-performance, data-parallel processing for applications utilizing big data. The HPCC platform includes system configurations to support both parallel batch data processing (Thor) and high-performance online query applications using indexed data files (Roxie). The HPCC platform also includes a data-centric declarative programming language for parallel data processing called ECL

[**Flink**](http://flink.incubator.apache.org/)

Apache Flink is an open source system for expressive, declarative, fast, and efficient data analysis. Flink combines the scalability and programming flexibility of distributed MapReduce-like platforms with the efficiency, out-of-core execution, and query optimization capabilities found in parallel databases.

[**Splice**](http://www.biomedcentral.com/1471-2105/14/S14/S13)

Splice Machine is essentially a Hadoop implementation of the Java-powered Apache Derby database project. Hadoop was built to run Java apps across clusters of machines, and so Splice Machine simply applies the Hadoop distributed-application method to Derby database workloads. The resulting system runs standard ANSI SQL-99 queries, but Splice Machine provides services for handling specific flavors of SQL, such as Oracle PL/SQL or Microsoft T-SQL

**Document Store Database**:

[**MongoDB**](http://www.mongodb.org/downloads)

MongoDB is an open-source database used by companies of all sizes, across all industries and for a wide variety of applications. It is an agile database that allows schemas to change quickly as applications evolve, while still providing the functionality developers expect from traditional databases, such as secondary indexes, a full query language and strict consistency. MongoDB is built for scalability, performance and high availability, scaling from single server deployments to large, complex multi-site architectures. By leveraging in-memory computing, MongoDB provides high performance for both reads and writes. MongoDB’s native replication and automated failover enable enterprise-grade reliability and operational flexibility

[**Elastic Search**](http://www.elasticsearch.org/download/)

Elasticsearch is a search server based on Lucene. It provides a distributed, multitenant-capable full-text search engine with a RESTful web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License.

[**Couchbase Server**](http://www.couchbase.com/download)

Couchbase Server originally known as Membase, is an open source, distributed (shared-nothing architecture) NoSQL document-oriented database that is optimized for interactive applications. These applications must service many concurrent users; creating, storing, retrieving, aggregating, manipulating and presenting data. In support of these kinds of application needs, Couchbase is designed to provide easy-to-scale key-value or document access with low latency and high sustained throughput. It is designed to be clustered from a single machine to very large scale deployments.

[**CouchDB**](http://couchdb.apache.org/)

CouchDB is a database that completely embraces the web. Store your data with JSON documents. Access your documents and query your indexes with your web browser, via HTTP. Index, combine, and transform your documents with JavaScript. CouchDB works well with modern web and mobile apps. You can even serve web apps directly out of CouchDB. And you can distribute your data, or your apps, efficiently using CouchDB’s incremental replication. CouchDB supports master-master setups with automatic conflict detection.

[**RethinkDB**](http://rethinkdb.com/docs/install/)

RethinkDB is an open-source, distributed database built to store JSON documents and scale to multiple machines with very little effort. It’s easy to set up and learn, and it has a pleasant query language that supports really useful queries like table joins, groupings, and aggregations

[**RavenDB**](http://ravendb.net/download)

RavenDB is also a 2nd generation document database. What we mean by saying that is that a lot of thought has been put on making sure it it does everything right. Features like Includes, Live Projections and Multi-map, and design decisions like making it Safe-By-Default, are all in to make sure RavenDB provides a real added value, and is not just yet another NoSQL solution

[**MarkLogic Server**](http://developer.marklogic.com/free-developer)

MarkLogic Server is an Enterprise NoSQL Database It fuses together database internals, search-style indexing, and application server behaviors into a unified system. It uses XML documents as its data model, and stores the documents within a transactional repository. It indexes the words and values from each of the loaded documents, as well as the document structure. And, because of its unique Universal Index, MarkLogic doesn’t require advance knowledge of the document structure (its “schema”) nor complete adherence to a particular schema. Through its application server capabilities, it’s programmable and extensible. MarkLogic Server (referred to from here on as just “MarkLogic”) clusters on commodity hardware using a shared-nothing architecture and differentiates itself in the market by supporting massive scale and fantastic performance customer deployments have scaled to hundreds of terabytes of source data while maintaining sub-second query response time.

[**Clusterpoint Server**](http://www2.clusterpoint.com/download/)

Clusterpoint Server is a database software for high-speed storage and large-scale processing of XML and JSON data on clusters of commodity hardware. It works as a schema free document-oriented DBMS platform with an open source API. Clusterpoint solves the problem of latency in Big data. End-users can instantly search billions of documents and do fast analytics in structured and unstructured data.

[**NeDB**](https://github.com/louischatriot/nedb)

NeDB is not intended to be a replacement of large-scale databases such as MongoDB! Its goal is to provide you with a clean and easy way to query data and persist it to disk, for web applications that do not need lots of concurrent connections, for example a continuous integration and deployment server and desktop applications built with Node Webkit. NeDB was benchmarked against the popular client-side database TaffyDB and NeDB is much, much faster.

[**Terrastore**](https://code.google.com/p/terrastore/downloads/list)

Terrastore is a modern document store which provides advanced scalability and elasticity features without sacrificing consistency. Terrastore is based on Terracotta, so it relies on an industry-proven, fast (and cool) clustering technology. Terrastore is accessed through the universally supported HTTP protocol. Terrastore is a distributed document store supporting single-cluster and multi-cluster deployments. Terrastore automatically scales your data: documents are partitioned and distributed among your nodes, with automatic and transparent re-balancing when nodes join and leave.

[**JasDB**](http://www.oberasoftware.com/downloads/)

JasDB is a NoSQL database using a document-based storage mechanism. It was developed with ease of use and minimal configuration in mind to provide an alternative to current document-based implementations out there, to add something new to the industry and give users more choices. JasDB can be installed and configured in almost no time at all.

[**RaptorDB**](http://raptordbkv.codeplex.com/releases/view/113297)

RaptorDB is a JSON based, NoSQL document store database that offers automatic hybrid bitmap indexing and LINQ query filters. This document-store can be used for the back-end store of forums, Blogs, Wikis, Content Management systems and websites. Users only need to know C# programming language to start using RaptorDB.

[**Djondb**](http://djondb.com/downloads.html)

A document-oriented database is a computer program designed for storing, retrieving, and managing document-oriented information, also known as semi-structured data. DjonDB is one type of document DB. All the documents in Djondb are stored in files and organized by namespace in the data folder and stored in JSON format.

[**EDB**](http://www.enterprisedb.com/downloads/postgres-postgresql-downloads)

EDB is an embedded database engine that provides core functionality for a Microsoft Windows CE application. By using EDB, a developer can create an object store called a volume that can contain multiple databases. The volume is file-based and therefore can be easily copied or moved. EDB is an updated and enhanced version of CEDB and provide support for: 1. Transactions, 2. Access by multiple users, 3. Multiple sort orders, key properties, and databases, 4. Enhanced performance, especially with larger databases

[**Amisa Server**](http://db-engines.com/en/system/Amisa+Server)

Amisa Server is a high performance general purpose database management system (DBMS) built from the ground up to power the next generation of data storage and retrieval applications. Amisa Server outperforms every workload optimized system currently available so completely eliminates the need to deploy multiple specialized systems for a single development initiative. Amisa Server saves money by reducing time to market, administration time and overall deployment costs. Amisa server implements the AQL programming language to manage and manipulate data. AQL is identical to SQL syntactically and functionally. Amisa server fully integrates a distributed search engine with a declarative query language to completely erase the query limitations on current search systems.

[**DensoDB**](http://densodb.codeplex.com/releases/view/98603)

DensoDB is a new NoSQL document database. Written for .Net environment in c# language. It’s simple, fast and reliable. No need of service installation and communication protocol. The fastest way to use it. You have direct access to the DataBase memory and you can manipulate objects and data in a very fast way. It gives you the power of a distributed scalable fast database, in a server or server-less environment.

[**SisoDB**](http://www.sisodb.com/nuget)

SisoDB is a schemaless document-oriented provider for SQL-Server. Using JSON and key-value storage, it lets you persist object graphs without specifying any mappings or extending any base classes interfaces etc. It lets you perform queries against SQL-server, using lambda expressions. It syncs schemachanges on the fly and can assist you to handle more complex model updates. Basically, it is a simple data access tool

[**SDB**](http://en.opensuse.org/SDB:Oracle_database_installation)

SDB works as persistent triple stores using relational databases. SDB uses an SQL database for the storage and query of RDF data. Many databases are supported, both Open Source and proprietary. An SDB store can be accessed and managed with the provided command line scripts and via the Jena API.

[**UnQLite**](http://unlite.org/downloads.html)

UnQLite is an in-process software library which implements a self-contained, serverless, zero-configuration, transactional NoSQL database engine. UnQLite is a document store database similar to MongoDB, Redis, CouchDB etc. as well a standard Key/Value store similar to BerkeleyDB, LevelDB. UnQLite is an embedded NoSQL (Key/Value store and Document-store) database engine. Unlike most other NoSQL databases, UnQLite does not have a separate server process. UnQLite reads and writes directly to ordinary disk files. A complete database with multiple collections, is contained in a single disk file. The database file format is cross-platform, you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures

[**ThruDB**](https://github.com/tjake/thrudb)

ThruDB is a set of simple services built on top of the Facebook Apache Thrift framework that provides indexing and document storage services for building and scaling websites. Its purpose is to offer web developers flexible, fast and easy-to-use services that can enhance or replace traditional data storage and access layers.

**Key Value / Tuple Store databases**:

[**Amazon DynamoDB**](http://aws.amazon.com/tools/?nc1=h_d_dl)

DynamoDB is a fast, fully managed NoSQL database service that makes it simple and cost-effective to store and retrieve any amount of data, and serve any level of request traffic. Its reliable throughput and single-digit millisecond latency make it a great fit for gaming, ad tech, mobile and many other applications.

[**Azure Table storage**](http://azurestorageexplorer.codeplex.com/releases/view/54430)

Azure Table services provides the potential to store enormous amounts of data, while enabling efficient access and persistence. The services simplify storage, saving you from jumping through all the hoops required to work with a relational database—constraints, views, indices, relationships and stored procedures. You just deal with data, data, data. Azure Tables use keys that enable efficient querying, and you can employ one—the PartitionKey—for load balancing when the table service decides it’s time to spread your table over multiple servers. A table doesn’t have a specified schema. It’s simply a structured container of rows (or entities) that doesn’t care what a row looks like. You can have a table that stores one particular type, but you can also store rows with varying structures in a single table.

[**Riak**](http://basho.com/riak/)

Riak uses a simple key/value model for object storage. Objects in Riak consist of a unique key and a value, stored in a flat namespace called a bucket. You can store anything you want in Riak: text, images, JSON/XML/HTML documents, user and session data, backups, log files, and more.

[**Redis**](http://redis.io/download)

Redis is a “NoSQL” key-value data store. More precisely, it is a data structure server. Not like MongoDB (which is a disk-based document store), though MongoDB could be used for similar key/value use cases. The closest analog is probably to think of Redis as Memcached, but with built-in persistence (snapshotting or journaling to disk) and more datatypes. Those two additions may seem pretty minor, but they are what make Redis pretty incredible. Persistence to disk means you can use Redis as a real database instead of just a volatile cache. The data won’t disappear when you restart, like with memcached.

[**Aerospike**](http://www.aerospike.com/download/server/3.3.12/)

Aerospike is the world’s fastest, most reliable in-memory open source NoSQL database that operates with unprecedented speed at scale on just a handful of servers. Aerospike enables a new class of applications that combine transactions and hot analytics, and process billions of objects, 20K-2M+ transactions per second (TPS) and 100GB-100TB+ of data with predictable sub-millisecond latency and ACID reliability. The first flash-optimized in-memory NoSQL database, Aerospike can run in pure RAM with spinning disks or as a hybrid memory database with RAM and flash. This enables our customers reap the benefits of the highest price-to-performance ratio available today. Aerospike has been powering a wide range of context driven application – from web portals to universal profile stores for real-time bidding and cross-channel marketing platforms.

[**FoundationDB**](https://foundationdb.com/get)

FoundationDB supports ACID transactions with high performance while maintaining the NoSQL benefit of scalability with distributed processing. Most NoSQL databases make no attempt to support ACID transactions. Those that do usually make fundamental compromises, such as supporting only local transactions on a single key, document, etc. FoundationDB supports global transactions over any number of keys. Read more about the importance of global transactions in the Transaction Manifesto.

[**LevelDB**](https://code.google.com/p/leveldb/downloads/list)

LevelDB is based on concepts from Google’s BigTable database system. The tablet implementation for the BigTable system was developed starting in about 2004, and is based on a different Google internal code base than the LevelDB code. That code base relies on a number of Google code libraries that are not themselves open sourced, so directly open sourcing that code would have been difficult. LevelDB stores keys and values in arbitrary byte arrays, and data is sorted by key. It supports batching writes, forward and backward iteration, and compression of the data via Google’s Snappy compression library. LevelDB is not a SQL database. Like other NoSQL and Dbm stores, it does not have a relational data model, it does not support SQL queries, and it has no support for indexes. Applications use LevelDB as a library, as it does not provide a server or command-line interface.

[**Berkeley DB**](http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/downloads/index.html)

Berkeley DB (BDB) is a software library that provides a high-performance embedded database for key/value data. Berkeley DB is written in C with API bindings for C++, C#, PHP, Java, Perl, Python, Ruby, Tcl, Smalltalk, and many other programming languages. BDB stores arbitrary key/data pairs as byte arrays, and supports multiple data items for a single key. Berkeley DB is not a relational database. BDB can support thousands of simultaneous threads of control or concurrent processes manipulating databases as large as 256 terabytes, on a wide variety of operating systems including most Unix-like and Windows systems, and real-time operating systems. Berkeley DB is also used as the common name for three distinct products; Oracle Berkeley DB, Berkeley DB Java Edition, and Berkeley DB XML. These three products all share a common ancestry and are currently under active development at Oracle Corporation.

[**Oracle NoSQL Database**](http://www.oracle.com/technetwork/database/database-technologies/nosqldb/downloads/index.html)

The Oracle NoSQL Database is a distributed key-value database. It is designed to provide highly reliable, scalable and available data storage across a configurable set of systems that function as storage nodes. Data is stored as key-value pairs, which are written to particular storage node(s), based on the hashed value of the primary key. Storage nodes are replicated to ensure high availability, rapid failover in the event of a node failure and optimal load balancing of queries. Customer applications are written using an easy-to-use Java/C API to read and write data.

[**GenieDB**](http://www.geniedb.com/)

GenieDB, a provider of distributed relational database technology, has launched a new database-as-a-service (DBaaS) offering, the GenieDB Globally Distributed MySQL-as-a-Service. The new GenieDB offering is a scalable DBaaS that enables enterprises to use the GenieDB automated platform to build Web-scale applications with the benefit of geographical database distribution. Geo-distribution provides enterprises with continuous availability during regional outages and better application response time for globally distributed users. Unlike many other database solutions, GenieDB enables developers to meet the challenges of cloud environments without having to give up critical database capabilities or abandoning investments in existing database infrastructure,” said Cary Breese, CEO of GenieDB, in a statement. “The technology provides an easy-to-use platform that overcomes the difficulties of managing a fully distributed database in the cloud, while allowing organizations to continue to use native MySQL.”

[**BangDB**](http://www.iqlect.com/download.php)

Multiflavored, distributed, transactional, high performance NoSQL database written in C/C++ from scratch for scale out apps suitable for heavy lifting. BangDB is available as Embedded Datastore, Client Server Model, Data Grid / Elastic Data Store.

[**Scalaris**](http://sourceforge.net/projects/scalaris/files/)

Scalaris is a scalable, transactional, distributed key-value store. It was the first NoSQL database, that supported the ACID properties for multi-key transactions. It can be used for building scalable Web 2.0 services. Scalaris uses a structured overlay with a non-blocking Paxos commit protocol for transaction processing with strong consistency over replicas. Scalaris is implemented in Erlang.

[**Tokyo Cabnit/Tyrant**](http://fallabs.com/tokyocabinet/)

Tokyo Cabinet is a library of routines for managing a database. The database is a simple data file containing records, each is a pair of a key and a value. Every key and value is serial bytes with variable length. Both binary data and character string can be used as a key and a value. There is neither concept of data tables nor data types. Records are organized in hash table, B+ tree, or fixed-length array. Tokyo Cabinet is developed as the successor of GDBM and QDBM on the following purposes.

[**Voldemort**](http://www.project-voldemort.com/voldemort/)

Voldemort is a distributed data store that is designed as a key-value store used by LinkedIn for high-scalability storage. It is named after the fictional Harry Potter villain Lord Voldemort. Voldemort is still under development. It is neither an object database, nor a relational database. It does not try to satisfy arbitrary relations and the ACID properties, but rather is a big, distributed, fault-tolerant, persistent hash table. A 2012 study comparing systems for storing APM monitoring data reported that Voldemort, Cassandra, and HBase offered linear scalability in most cases, with Voldemort having the lowest latency and Cassandra having the highest throughput.

[**Dynomite**](https://docs.google.com/document/d/1AFSOrjZErJuLERkfqYcJw1IOPOA3FdpshyGGZo9xonw/edit?pli=1)

Dynomite currently provides integrated storage and distribution, requiring developers to adopt a simple, key/value data model to get the availability and scalability advantages. By separating these two functions, developers can take advantage of the sophisticated distribution and scaling techniques of Dynomite with great flexibility in the choice of data model. In this new architecture, Dynomite handles data partitioning, versioning, and read repair, and user-provided storage engines provide persistence and query processing.

[**MemcacheDB**](http://memcached.org/)

MemcacheDB  is a persistence enabled variant of memcached, a general-purpose distributed memory caching system often used to speed up dynamic database-driven websites by caching data and objects in memory. The main difference between MemcacheDB and memcached is that MemcacheDB has its own key-value database system based on Berkeley DB, so it is meant for persistent storage rather than as a cache solution. MemcacheDB is accessed through the same protocol as memcached, so applications may use any memcached API as a means of accessing a MemcacheDB database

[**c-treeACE database**](http://www.faircom.com/ace/download_t.php)

c-tree database is a cross-platform database engine developed by FairCom Corporation. Software developers typically embed the c-treeACE engine within the applications that they create and then deploy the application and engine together as an integrated solution. At its core, c-treeACE uses a record-oriented, Indexed Sequential Access Method (ISAM) structure offering high speed indexing mechanisms over those files. Developers can use these direct access methods to design the data and index structures that closely parallel the needs of their application. This paradigm is sometimes referred to as an application-specific database or an embedded database because of the tightly coupled nature of the application and database.

[**KitaroDB**](http://www.kitarodb.com/)

KitaroDB is a free NoSQL database that runs natively in the WinRT, Win32, and .NET environments. KitaroDB is a fast, efficient data store that supports key-value pairs as well as intrusive keys, and can be used by developers across Microsoft’s platforms. Based on a commercial database driving enterprise applications for more than 25 years, KitaroDB brings NoSQL to WinRT, the new Windows 8 UI, and also supports Win32 and .NET applications. Capable of thousands of operations per second, KitaroDB is nevertheless small enough to fit on client devices leaving resources available for the rest of the application. The easy-to-use interface enables developers to spend their time programming application features, and not worrying about how to push their schemaless data into a rigid schema.”

[**hamsterdb**](http://hamsterdb.com/download/)

hamsterdb runs on a variety of platforms, including tablets and phones, desktop machines and cloud instances. All major operating systems are supported. Unlike other key-value databases, hamsterdb knows about the type of the keys and will use that information to optimize storage and algorithms. A database storing integer keys uses a completely different memory layout than variable length binary keys. This memory layout drastically reduces the file size, reduces I/O, increases performance and improves scalability.

[**STSdb**](http://stsdb.com/)

STSdb is an open-source, client/server and embedded NoSQL database and virtual file system in one. It is built up from scratch without using any third party components. Data is stored in a very flexible key-value format where the key consists of the combination of sub-keys and an associated value. The innovative design makes STSdb perfect for BigData and cloud applications.

[**Tarantool**](http://tarantool.org/download.html)

Tarantool is a NoSQL database running inside a Lua program. It’s created to store and process the most volatile and highly accessible Web data. In Tarantool, all data is maintained in RAM. Data persistence is implemented using a Write Ahead Log and snapshotting. It supports asynchronous replication and hot standby and uses coroutines and asynchronous I/O to implement high-performance lock-free access to data.

[**quasardb**](https://www.quasardb.net/)

quasardb is a distributed, high-performance, associative database designed from the ground up for the most demanding environments. Based on decades of theoretical research and years of prototyping, quasardb stands on the shoulder of giants: it combines breakthroughs from relational databases, operating systems and network distribution to redefine the state of the art. quasardb already withstood the fire of critical environments where failure isn’t an option and will change the way you look at associative databases.

**RaptorDB**

RaptorDB is a JSON based, NoSQL document store database that offers automatic hybrid bitmap indexing and LINQ query filters. This document-store can be used for the back-end store of forums, Blogs, Wikis, Content Management systems and websites. Users only need to know C# programming language to start using RaptorDB.

**TIBCO ActiveSpaces DB**

As the volume, variety, and velocity of data grows exponentially, applications designed using traditional data storage technologies such as relational databases are not able to scale. Two technologies have come forward to address this need, in-memory data grids and NoSQL databases. TIBCO ActiveSpaces takes an approach that is the best of both. On the one hand, it stores data in memory on a cluster of machines for fast read access, and on the other hand, it provides distributed persistence on local file systems for very fast write performance.

[**NessDB**](https://github.com/BohuTANG/nessDB)

NessDB is a very fast key-value, embedded Database Storage Engine (Using log-structured-merge (LSM) trees) with Level-LRU, Bloom-Filter.

[**HyperDex**](http://hyperdex.org/download/)

HyperDex, a novel distributed key-value store that provides a unique search primitive that enables queries on secondary attributes. The key insightbehind HyperDex is the concept of hyperspace hashing in which objects with multiple attributes are mapped into a multidimensional hyperspace. This mapping leads to efficient implementations not only for retrieval by primary key, but also for partially-specied secondary attribute searches and range queries. A novel chaining protocol enables the system to achieve strong consistency, maintain availability and guarantee fault tolerance.

[**Symas Lightning Memory Mapped Database (LMDB)**](https://github.com/deephacks/lmdb)

LMDB is an ultra-fast, ultra-compact key-value embedded data store developed by Symas for the OpenLDAP Project. It uses memory-mapped files, so it has the read performance of a pure in-memory database while still offering the persistence of standard disk-based databases, and is only limited to the size of the virtual address space

[**PickleDB**](https://github.com/patx/pickledb)

PickleDB is a simple store of kind key/value that was written by Harrison Erd. It Easy integrate with your python code. It has a limited capacity to work with large dataset, due that works with it in memory and then dump it to a file

[**Light Cloud**](https://github.com/Plurk/LightCloud)

Distributed and persistent key-value database Built on Tokyo Tyrant. One of the fastest key-value databases. Can store millions of keys on very few servers – tested in production. LightCloud is a distributed and horizontal scaleable database

[**Hibari**](http://sourceforge.net/projects/hibari/)

Hibari Cloud Database is a distributed non-relational database management system (Distributed Non-RDBMS) for cloud computing to support explosively growing data volume. Hibari is a distributed, high availability key-value data store that focuses on the “C”onsistency and “A”vailability aspects of Brewer’s CAP Theorem.

[**Genome**](https://www.biostars.org/p/9202/)

These databases collect genome sequences, annotate and analyze them, and provide public access. Some add curation of experimental literature to improve computed annotations. These databases may hold many species genomes, or a single model organism genome.

**Graph Databases**:

[**Neo4J**](http://www.neo4j.org/download)

Neo4J is a Java-based open source NoSQL graph database. With a graph database, which can search social network data, connections between data are explored. Neo4j can solve problems that require repeated network probing (the database is filled with nodes, which are then linked), and the company stresses Neo4j’s high performance. The importance of graph database technology as well as Neoo4j’s potential in the mobile space. Eifrem also stressed his confidence in Java, despite recent security issues affecting the platform.

[**InfiniteGraph**](http://www.objectivity.com/downloads/)

InfiniteGraph is a distributed graph database implemented in Java, and is from a class of NOSQL (or Not Only SQL) data technologies focused on graph data structures. Graph data typically consist of objects or things (nodes) and various relationships (edges) that may connect two or more nodes. Developers may use Infinitegraph to build web and mobile applications and services that need to solve graph problems or answer.

[**DEX**](http://www.pcdj.com/support/dex-2-support/)

DEX is based on a graph database model, that is basically characterized by three properties: data structures are graphs or any other structure similar to a graph; data manipulation and queries are based on graph-oriented operations; and there are data constraints to guarantee the integrity of the data and its relationships. A DEX graph is a Labeled Directed Attributed Multigraph. Labeled because nodes and edges in a graph belong to types. Directed because it supports directed edges as well as undirected. Attributed because both nodes and edges may have attributes and Multigraph meaning that there may be multiple edges between the same nodes even if they are from the same edge type.

[**Titan**](http://keylines.com/titan)

Titan is a scalable graph database optimized for storing and querying graphs containing hundreds of billions of vertices and edges distributed across a multi-machine cluster. Titan is a transactional database that can support thousands of concurrent users executing complex graph traversals.

[**Infogrid**](http://infogrid.org/trac/wiki/Docs/Downloads)

InfoGrid is a Web Graph Database with a many additional software components that make the development of REST-ful web applications on a graph foundation easy. InfoGrid is open source, and is being developed in Java as a set of projects. Provides an abstract common interface to storage technologies such as SQL databases and distributed NoSQL hashtables. This enables an InfoGrid GraphDatabase to persist its data using any of several different storage technologies but with the same API for application developers.

[**HypergraphDB**](http://www.hypergraphdb.org/downloads)

HypergraphDB is open source data storage mechanism based on powerful knowledge management formalism known as directed hypergraphs. While a persistent memory model designed mostly for knowledge management, AI and semantic web projects, it can also be used as an embedded object-oriented database for Java projects of all sizes. Or a graph database. Or a (non-SQL) relational database. HyperGraphDB application components implement various domain models, standards, algorithms and domain-specific tools, taking advantage of its generality. Every entity in those components is ultimately a HyperGraphDB atom, which makes it possible to integrate and compose them naturally.

[**Trinity**](http://sourceforge.net/projects/gt-db/)

General purpose graph computation faces a great challenge of random data access. Meanwhile, the RAM capacity limit forms a scale bound of single machine solutions for general purpose graph processing. Trinity is a general purpose distributed graph system over a memory cloud. Memory cloud is a globally addressable, in-memory key-value store over a cluster of machines. Through the distributed in-memory storage, Trinity provides fast random data access power over a large data set. This makes Trinity a natural large graph processing platform. With the power of fast graph exploration and distributed parallel computing, Trinity supports both low-latency online query processing and high-throughput offline analytics on billion-node scale large graphs.

[**AllegroGraph**](http://franz.com/downloads.lhtml)

AllegroGraph is a modern, high-performance, persistent graph database. AllegroGraph uses efficient memory utilization in combination with disk-based storage, enabling it to scale to billions of quads while maintaining superior performance. AllegroGraph supports SPARQL, RDFS++, and Prolog reasoning from numerous client applications.

[**WHITE Database**](http://whitedb.org/)

The Workplace Health Indicator Tracking and Evaluation (WHITE™) database is a web-based system that centralizes information on incident tracking and case management for the BC health authorities. The information enables the healthcare sector to reduce and/or eliminate workplace injuries, provide prompt clinical and workplace interventions to reduce disability and time loss, and evaluate the effectiveness of health and safety programs.

[**Virtuoso**](https://github.com/openlink/virtuoso-opensource)

Virtuoso Universal Server is a middleware and database engine hybrid that combines the functionality of a traditional RDBMS, ORDBMS, virtual database, RDF, XML, free-text, web application server and file server functionality in a single system. Rather than have dedicated servers for each of the aforementioned functionality realms, Virtuoso is a “universal server”; it enables a single multithreaded server process that implements multiple protocols. The open source edition of Virtuoso Universal Server is also known as OpenLink Virtuoso. The software has been developed by OpenLink Software with Kingsley Uyi Idehen and Orri Erling as the chief software architects.

[**VertxDB**](http://www.dekorte.com/projects/opensource/vertexdb/)

VertexDB is a high performance graph database server that supports automatic garbage collection. It uses the HTTP protocol for requests and JSON for its response data format and the API is inspired by the FUSE filesystem API plus a few extra methods for queries and queues. VertexDB is composed of nodes which are folders of key/value pairs. Keys are stored in lexical ordering and can be any string not containing a forward slash character

[**FlockDB**](http://webscripts.softpedia.com/script/Database-Tools/FlockDB-66248.html)

FlockDB is an open source distributed, fault-tolerant graph database for managing wide but shallow network graphs. It was initially used by Twitter to store relationships between users, e.g. followings and favorites. FlockDB differs from other graph databases, e.g. Neo4j in that it is not designed for multi-hop graph traversal but rather for rapid set operations, not unlike the primary use-case for Redis sets. Since it is still in the process of being packaged for outside of Twitter use, the code is still very rough and hence there is no stable release available yet. FlockDB was posted on GitHub shortly after Twitter released its Gizzard framework, which it uses to query the FlockDB distributed datastore.

[**BrightstarDB**](https://www.nuget.org/packages/BrightstarDB/)

BrightstarDB was created with the goal of making the benefits of the flexible, schema-free RDF model available to .NET developers in an easy-to-use persistent store. BrightstarDB is, at its core, an RDF data store capable of handling millions of RDF triples; but unlike many other stores, BrightstarDB does not force the programmer to use an unfamiliar RDF-based API. Instead we built two layers on top; one that enables the use of .NET’s dynamic objects for retrieval and update; and another that provides a full “contract-first” entity model allowing you to define an application’s domain model as .NET interfaces with minimal annotation and then use LINQ to query the data store and a “context object” pattern that will be familiar to users of the .NET Entity Framework for entity creation and update operations.

**Multimodel Databases**

[**ArangoDB**](https://www.arangodb.org/download)

A distributed open-source database with a flexible data model for documents, graphs, and key-values. Build high performance applications using a convenient sql-like query language or JavaScript extensions.

[**OrientDB**](http://www.orientechnologies.com/download/)

OrientDB is an Open Source NoSQL DBMS with the features of both Document and Graph DBMSs. Written in Java, it is incredibly fast: it can store up to 150,000 records per second on common hardware. Even for a Document based database, the relationships are managed as in Graph Databases with direct connections among records. You can traverse parts of or entire trees and graphs of records in a few milliseconds. Supports schema-less, schema-full and schema-mixed modes. Has a strong security profiling system based on user and roles and supports SQL amongst the query languages. Thanks to the SQL layer, it’s straightforward to use for those skilled in the relational database world.

[**DatomicDB**](https://github.com/Datomic/mbrainz-sample)

Datomic is a new database designed as a composition of simple services. It strives to strike a balance between the capabilities of the traditional RDBMS and the elastic scalability of the new generation of redundant distributed storage systems.

[**FatDB**](http://www.dbdirections.com/nycsqlusergroup/enhancing-sql-server-with-a-no-sql-db/)

FatDB is the next generation NoSQL database for Windows that extends database functionality by integrating Map Reduce, a work queue, file management system, high-speed cache, and application services. FatDB is built to integrate tightly with SQL Server so that you can build exciting new applications that leverage relational and unstructured data models.

[**AlchemyDB**](https://code.google.com/p/alchemydatabase/)

Alchemy Database is a low-latency high-TPS NewSQL RDBMS embedded in the NOSQL datastore redis. Extensive datastore-side-scripting is provided via deeply embedded Lua. Unstructured data, can also be stored, as there are no limits on #tables, #indexes, #columns, and sparsely populated rows use minimal memory. AlchemyDB was the first NewSQL database to integrate relational database management system (RDBMS), document store, and graph database capabilities on top of the Redis open-source key-value store.

[**coretxDB**](http://download.cnet.com/Cortex-Pro/3640-2654_4-10262141-2.html)

cortex uses SQLite database engine – fast, reliable and file based, which means, you don’t have to mess with drivers. You can use them through the UI, to keep data organized. Or you can access databases from Cortex scripting language

**Object Databases**:

[**VersantDB**](http://developer.versant.com/developer/downloads)

The Versant Object Database enables developers using object oriented languages to transactionally store their information by allowing the respective language to act as the Data Definition Language (DDL) for the database. In other words, the memory model is the database schema model.In general, persistence in VOD in implemented by declaring a list of classes, then providing a transaction demarcation application programming interface to use cases. Respective language integrations adhere to the constructs of that language, including syntactic and directive sugars.Additional APIs exist, beyond simple transaction demarcation, providing for the more advanced capabilities necessary to address practical issues found when dealing with performance optimization and scalability for systems with large amounts of data, many concurrent users, network latency, disk bottlenecks.

[**Objectivity**](https://download.objectivity.com/approved.aspx)

Objectivity/DB is a commercial object database produced by Objectivity, Inc. It allows applications to make standard C++, Java, Python or Smalltalk objects persistent without having to convert the data objects into the rows and columns used by a relational database management system (RDBMS). Objectivity/DB supports the most popular object oriented languages plus SQL/ODBC and XML. It runs on Linux, LynxOS, UNIX and Windows platforms. All of the languages and platforms interoperate, with the Objectivity/DB kernel taking care of compiler and hardware platform differences.

[**Gemstone**](http://seaside.gemtalksystems.com/downloads.html)

GemStone provides a distributed, server-based, multiuser, transactional Smalltalk runtime system, Smalltalk application partitioning technology, access to relational data, and production-quality scalability and availability. The GemStone object server allows you to bring together object-based applications and existing enterprise and business information in a three-tier, distributed client/server environment.

[**Starcounter**](https://github.com/Starcounter/Starcounter)

Starcounter is, in contrast to OldSQL databases, originally designed to have its main storage in RAM, to utilize modern multi-core CPUs with several level of caches, and to minimize overhead. Starcounter also makes use of a new invention we call VMDBMS, which makes it substantially faster than other in-memory high performance databases. VMDBMS stands for an integration between the application runtime virtual machine (VM) and the database management system (DBMS). As a result of this integration the database data resides all the time in one single place in RAM and is not copied back and forth between the database and the application.

[**HSS Database**](http://highspeed-solutions.net/blog/2011/12/04/hss-database-v1-2-0-released/)

The HSS Database is an object oriented database management system (OODB or ODBMS) for Microsoft .NET, Silverlight and Windows Phone 7. HSS Database gives developers the ability to store and retrieve objects from their applications with extremely high speeds compared to other solutions

[**ZODB**](https://pypi.python.org/pypi/ZODB)

The ZODB is a native object database that stores your objects while allowing you to work with any paradigms that can be expressed in Python. Thereby your code becomes simpler, more robust and easier to understand. A ZODB storage is basically a directed graph of (Python) objects pointing at each other, with a Python dictionary at the root. Objects are accessed by starting at the root, and following pointers until the target object. In this respect, ZODB can be seen as a sophisticated Python persistence layer

[**Magma**](http://magma.maths.usyd.edu.au/magma/)

Magma is an open-source object-oriented database developed entirely in Smalltalk. Magma provides transparent access to a large-scale shared persistent object model. It supports multiple users concurrently via optimistic locking. It uses a simple transaction protocol, including nested transactions, supports collaborative program development via live class evolution, peer-to-peer model sharing and Monticello integration. Magma supports large, indexed collections with robust querying, runs with pretty good performance and provides performance tuning mechanisms. Magma is fault tolerant and includes a small suite of tools. Magma can either work locally or on a remote Magma server. This means, multiple images can access the same database concurrently.

**NEODB**

Neo is a database designed for network­oriented data. This is data that is ordered in complex networks or deep trees. Where the relational model is based on tables, columns and rows, Neo’s primitives are nodes, relationships and properties. Together, these form a large network of information that we call a node space. Neo shines at handling semi­structured data. Semi­structured data is a research term that is quickly gaining ground outside of academia. Simply put, semi­structured data typically has few mandatory but many optional attributes. As a consequence, it usually has a very dynamic structure, sometimes to the point where it varies even between every single element. Data with that degree of variance is difficult to fit in a relational database schema but can be easily represented in the Neo model.

[**Streling**](http://www-01.ibm.com/support/docview.wss?uid=swg24037472)

Sterling is a NoSQL object-oriented database developed especially for Silverlight, Windows Phone 7.0 and .NET. It supports LINQ object queries. The core is light so that the system is flexible and it becomes easy to query the database.

[**EyeDB**](http://www.eyedb.org/)

EyeDB is an Object Oriented Database Management System (OODBMS) based on the ODMG 3 specification, developed and supported by the French company SYSRA. EyeDB provides an advanced object model (inheritance, collections, arrays, methods, triggers, constraints, and reflexivity), an object definition language based on ODMG ODL, an object query and manipulation language based on ODMG OQL and programming interfaces for C++ and Java.

[**FarmerD**](http://www.framerd.org/)

FramerD is a portable distributed object-oriented database designed to support the maintenance and sharing of knowledge bases. Unlike other object-oriented databases, FramerD is optimized for the sort of pointer-intensive data structures used by semantic networks, frame systems, and many intelligent agent applications. FramerD databases readily include millions of searchable frames and may be distributed over multiple networked machines. FramerD includes an extensive scripting language based on Scheme with special support for web-based interfaces. FramerD is implemented in ANSI C and has been compiled for a wide range of platforms, including many varieties of Unix, Mac OS X, WIN32. In addition, experimental Java and Lisp libraries exist for accessing FramerD databases and services.

[**NinjaDB**](https://www.dbninja.com/?page=download)

Ninja Database Pro is deadly good. Ninja Database Pro is a lighting fast, compact, ACID compliant database. It can be used as a database for desktop applications, a Silverlight database, or a Windows Phone 7 database, an Android database with Xamarin’s MonoDroid or an iPhone database with Xamarin’s MonoTouch. It is the first database supporting either object database mode or relational database mode. You choose how to save your child objects as embedded or in a separate table. It supports all the features you expect: LINQ index queries, paging, transactions, constraints, triggers, caching, BLOB, CLOB, Import XML, Export XML, Auto Identity Primary Keys, and foreign key relationships. Industry standard AES encryption and Mini LZO compression are included. Unlike most other databases, Ninja Database Pro can save complex data structures such as double linked lists, multi-dimensional arrays, and dictionaries. Databases can be created in memory, isolated storage, or normal file storage.

[**ObjectDB**](http://www.objectdb.com/object/db/database/download)

ObjectDB is the most productive software for developing Java database applications using the Java Persistence API (JPA). It is the first persistence solution that combines a powerful database with JPA support in one product, saving the need to integrate an external JPA ORM with a database.

**Grid & Cloud Database**:

[**Oracle Coherence**](http://www.oracle.com/technetwork/middleware/downloads/index-087510.html)

Oracle coherence has revolutionized the way clustered application data is cached. Oracle Coherence manages data in clustered applications and application servers as if it were a single application server. Database applications no longer need to query the database directly each time data is required to be retrieved, updated, or deleted. A Coherence cache is a collection of data objects that serves as an intermediary between the database and the client applications. Database data may be loaded into a cache and made available to different applications. Thus, Coherence caches reduce load on the database and provide faster access to database data

[**GemfireDB**](http://www.dbvis.com/download/)

Gemfire is a distributed memory oriented data management platform that pools memory (and CPU, network and optionally local disk) across multiple processes to manage application objects and behavior. GemFire uses dynamic replication and data partitioning techniques to offer continuous availability, very high performance and linear scalability for data intensive applications without compromising on data consistency even when exposed to failure conditions. Besides being a distributed data container, it is an active data management system that uses an optimized low latency distribution layer for reliable asynchronous event notifications along with highly concurrent data structures for storage.

[**Infinispan**](http://infinispan.org/download/)

Infinispan is an extremely scalable, highly available key/value data store and data grid platform. It is 100% open source, and written in Java. The purpose of Infinispan is to expose a data structure that is distributed, highly concurrent and designed ground-up to make the most of modern multi-processor and multi-core architectures. It is often used as a distributed cache, but also as a NoSQL key/value store or object database.

[**Hazelcast**](http://hazelcast.com/resources/speed-database-hazelcast/)

One of the most common use cases that In Memory Data Grids (IMDG) like Hazelcast solve is that of the slow or unscalable Relational Database (RDBMS). Scaling a non-performant RDBMS at best involves knowledge of complex configuration techniques and at worst could require the addition of expensive non commodity hardware. In this webinar we will demonstrate how you can easily add Hazelcast into the workflow of your application to solve this issue. Hazelcast can be used to solve the problem of slow reads by caching data in memory and it can also relieve stress on a Database where slow updates are an issue for your application.

**XML Databases**:

[**EMC Documentum xDB**](http://india.emc.com/products/how-to-buy/index.htm)

EMC Documentum xDB is a high-performance and scalable native XML database that is ideal for data-intensive uses such as archiving data from retired applications. Unlike relational databases, Documentum xDB allows database structures to be easily modified to adapt to changing information requirements. It also handles complex data relationships that are not easily modeled in relational rows and columns.Data will be safe with xDB’s high-availability and disaster-recovery options. xDB also provides a powerful, extensible development and runtime toolset based on XML standards as well as full support for the XQuery language for data and full-text searches.

[**eXist**](http://exist-db.org/exist/apps/homepage/index.html)

eXist is an open source database management system entirely built on XML technology, also called a native XML database. Unlike most relational database management systems, eXist uses XQuery, which is a W3C Recommendation, to manipulate its data.an open-source native XML database which provides an easy-to-use and powerful environment for learning and applying XML languages. We begin with a brief description on how to install EXIST and execute some simple operations. EXIST provides a graphical interface which is pretty easy to use.

[**Sedna**](http://www.sedna.org/download.html)

Sedna is a free native XML database which provides a full range of core database services – persistent storage, ACID transactions, security, indices, hot backup. Flexible XML processing facilities include W3C XQuery implementation, tight integration of XQuery with full-text search facilities and a node-level update language.

[**BaseX**](http://basex.org/products/download/all-downloads/)

BaseX is a native and light-weight XML database management system and XQuery processor, developed as a community project on GitHub. It is specialized in storing, querying, and visualizing large XML documents and collections. BaseX is platform-independent and distributed under a permissive free software license. In contrast to other document-oriented databases, XML databases provide support for standardized query languages such as XPath and XQuery. BaseX is highly conformant to World Wide Web Consortium specifications and the official Update and Full Text extensions. The included GUI enables users to interactively search, explore and analyze their data, and evaluate XPath/XQuery expression in the lifetime.

**Qizx/db**

Qizx/db is a XML Query database engine designed to be embedded in a Java application – typically a Servlet. As such, it is primarily used as a class library. To help experimenting with XML Query and XML databases and developing, Qizx/db also comes with two tools which make it easy to build a database, populate it with XML documents, and perform queries on this database

**BerkeleyDB**

Oracle Berkeley DB XML is an XML database with support for XQuery designed to store and index XML content for fast, scalable and predictable access. It is a C, C++ library that links into your application. Berkeley DB XML provides transactional access, automatic recovery, content compression, on-disk data encryption with AES, fail-over to a hot standby, and replication for high availability. Store, index and query key/value meta-data related to the XML documents as well. Berkeley DB XML provides fast, reliable and scalable persistence for applications that need to manage XML content.

**Multidimensional Databases**:

[**Global**](http://globalsdb.org/downloads)

A Global is a persistent sparse multi-dimensional array, which consists of one or more storage elements or “nodes”. Each node is identified by a node reference. Each node consists of a name and zero or more subscripts The data stored at each level of the global can either be atomic (a single piece of information) or complex (multiple pieces of information stored in ValueLista format) in nature. In its simplest form, a global is a collection of its name, and all of its subscripts. Given this simple definition, a Globals Database will consist of one or more named globals, each with its own set of zero or more subscripts.

[**Intersystem cache**](https://download.intersystems.com/download/register.csp)

At the heart of Caché lies the Caché Database Engine. The database engine is highly optimized for performance, concurrency, scalability, and reliability. There is a high degree of platform-specific optimization to attain maximum performance on each supported platform. Caché is a full-featured database system; it includes all the features needed for running mission-critical applications (including journaling, backup and recovery, and system administration tools). To help reduce operating costs, Caché is designed to require significantly less database administration than other database products. The majority of deployed Caché systems have no database

[**GT.M**](http://sourceforge.net/projects/fis-gtm/)

GT.M is a database engine with scalability proven in the largest real-time core processing systems in production at financial institutions worldwide, as well as in large, well known healthcare institutions, but with a small footprint that scales down to use in small clinics, virtual machines and software appliances. The GT.M data model is a hierarchical associative memory that imposes no restrictions on the data types of the indexes and the content – the application logic can impose any schema, dictionary or data organization suited to its problem domain.\* GT.M’s compiler for the standard M also known as MUMPS scripting language implements full support for ACID (Atomic, Consistent, Isolated, Durable) transactions, using optimistic concurrency control and software transactional memory (STM) that resolves the common mismatch between databases and programming languages

[**SciDB**](http://scidb.sourceforge.net/download.html)

SciDB organizes data as a collection of multidimensional arrays. Just as the relational table is the basis of relational algebra and SQL, the multidimensional array is the basis for SciDB.Array database designed for multidimensional data management and analytics common to scientific, geospatial, financial, and industrial applications.

[**Rasdaman**](http://www.rasdaman.org/wiki/Install)

RasDaMan is a universal domain-independent array DBMS for multidimensional arrays of arbitrary size and structure. A declarative, SQL-based array query language offers flexible retrieval and manipulation. Efficient server-based query evaluation is enabled by an intelligent optimizer and a streamlined storage architecture based on flexible array tiling and compression. RasDaMan is being used in several international projects for the management of geo and healthcare data of various dimensionality.

**Network Model Databases**:

[**Vyhodb**](http://www.vyhodb.com/)

Vyhodb Service oriented, schema-less, network data model DBMS. Client application invokes methods of vyhodb services, which are written in Java and deployed inside vyhodb. Vyhodb services reads and modifies storage data. API: Java, Protocol: RSI – Remote service invocation, Written in: Java, ACID: fully supported, Replication: async master slave, Misc: online backup, License: proprietary.

*Have something to say? Share it in the comments.*