|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | BigTable | DynamoBD | LevelDB | Hypertable | MongoDB | CouchDB | Cassandra | HBase |
| Licencia | Propietario | Propietario | Open Source | Open Source | AGPL (Drivers: Apache) | Open Source | Apache | Apache |
| Balanceo de carga |  |  |  |  | shard |  |  |  |
| Modelo de Datos | Clave/valor | Clave/valor | *BigTable* | *BigTable* | Base de datos Documental | clave-valor/documental | *BigTable* | *BigTable* |
| Infraestructura | GFS | Dynamo (SSD) |  | *HDFS*, *MapR*, *Ceph*, *KFS* o el sistema de ficheros local | [*Volatile memory*](http://vschart.com/list/volatile-memory/)  [*File Syste*](http://vschart.com/list/file-system/)*m* |  |  | HDFS, S3, S3N, EBS |
| Replicación | maestro/esclavo(lectura y escritura de maestro) | maestro/esclavo |  |  | maestro/esclavo | Bidireccional  maestro-maestro | *Parecido Dynamo* | maestro/esclavo |
| Consultas |  |  |  |  | javascript |  | [*Cassandra Query Language*](http://en.wikipedia.org/w/index.php?title=Cassandra_Query_Language&action=edit&redlink=1)*(CQL)* |  |
| APIs | C++, Interfaz cliente | AWS | C++ | *HQL* (*Hypertable* *Query* *Language*) y *Thrift API* | Java,PHP,Ruby  C#,Python,  JavaScript,Haskell,Perl,C++,Erlang  Scala,C |  | *Java* (*Hector*), *Python* (*Pycassa*), *PHP* (*PHPcassa*) | *Java*, *Jython*, *Groovy*, *Scala* y *JRuby*, y *REST* y *Thrift* |
| Lenguaje de implementación | C/C++ | Java |  | C + + | C + + | Erlang | Java | Java |
| Map / Reduce | Si,(Google) | Si,(Amazon Elastic MapReduce) |  |  | No | Si | Si(Hadoop) | Si(Hadoop) |
| Protocolo |  |  |  | Thrift,biblioteca de C + +, o HQL shell | BSON | HTTP / REST |  |  |
| Empresas | Google | Amazon |  | Baidu | Craigslist, Foursquare, Shutterfly, Intuit |  | Digg, Twiter, Rackspace, Facebook, Apache | Facebook |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | BigTable | DynamoBD | LevelDB | Hypertable | MongoDB | CouchDB | Cassandra | HBase |
| Plataforma/s | Multiplataforma | Multiplataforma |  |  | Linux  Windows  Mac OS X  Solaris |  | Multiplataforma |  |
| Escalabilidad |  |  |  |  |  | Altamente escalable |  |  |
| Consistencia | Si | si | Si | Si | Si | No | No | Si |
| Alta  Disponibilidad | Si | Si | No | No | No | Si | Si | No |
| Persistencia | Si | No | Si | Si | Si | Si | Si | Si |
| Tolerancia a fallos | Si | No | No | Si | Si | Si | Si | Si |
| Auto fragmentación |  |  |  |  |  | Si |  |  |
| Formato de archivo de almacenamiento de datos | SSTable |  | SSTable |  |  |  |  | HFile |
| Modelo de integridad | Lock | ACID |  | Lock | BASE | MVCC | MVCC | Lock |
| Lock Service | Chubby |  |  | Hyperspace |  |  |  | ZooKeeper |
| *Cell Versions* | Si |  |  | Si |  |  |  | Si |
| Coste | - |  |  |  |  | Bajo |  |  |

ACID:  Atomicidad, Consistencia, Aislamiento y Durabilidad

MVCC: operaciones de escritura no bloquean lectura

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | BigTable | DynamoBD | LevelDB | Hypertable | MongoDB | CouchDB | Cassandra | HBase |
| Mejor Uso | diseñado para escalar a través de cientos o miles de máquinas | para soluciónes grande Data Base |  |  | consultas dinámicas, con frecuencia escrituras, rara vez leen los datos estadísticos | acumulación, en ocasiones cambiar datos con consultas predefinidas | escribir con frecuencia, leer menos | lectura aleatoria escribir en grandes bases de datos |

<http://www.larsgeorge.com/2009/11/hbase-vs-bigtable-comparison.html>