|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Year*  | *System/ Paper*  | *Scale to 1000s*  |  *Primary Index*  | *Secondary Indexes*  | *Transactions*  | *Joins/ Analytics*  | *Integrity Constraints*  | *Views*  |  *Language/ Algebra*  | *Data model*  | *my label*  |
| 1971 |  RDBMS  | **O**  | ✔  | ✔ | ✔  | ✔  | ✔  | ✔  | ✔  | **tables**  | **sql-like**  |
| 2003 |  memcached  | ✔  | ✔  | **O**  | **O**  | **O**  | **O**  | **O**  | **O**  | **key-val**  | **nosql**  |
| 2004 |  MapReduce  | ✔  | **O**  | **O**  | **O**  | ✔  | **O**  | **O**  | **O**  | **key-val**  | **batch**  |
| 2005 |  CouchDB  | ✔  | ✔  | ✔  | **record**  | **MR**  | **O**  | ✔  | **O**  | **document**  | **nosql**  |
| 2006 |  BigTable (Hbase)  | ✔  | ✔  | ✔  | **record**  | **compat. w/MR**  | **/**  | **O**  | **O**  | **ext. record**  | **nosql**  |
| 2007 |  MongoDB  | ✔  | ✔  | ✔  | **EC, record**  | **O**  | **O**  | **O**  | **O**  | **document**  | **nosql**  |
| 2007 |  Dynamo  | ✔  | ✔  | **O**  | **O**  | **O**  | **O**  | **O**  | **O**  | **key-val**  | **nosql**  |
| 2008 |  Pig  | ✔  | **O**  | **O**  | **O**  | ✔  | **/**  | **O**  | ✔  | **tables**  | **sql-like**  |
| 2008 |  HIVE  | ✔  | **O**  | **O**  | **O**  | ✔  | ✔  | **O**  | ✔  | **tables**  | **sql-like**  |
| 2008 |  Cassandra  | ✔  | ✔  | ✔  | **EC, record**  | **O**  | ✔  | ✔  | **O**  | **key-val**  | **nosql**  |
| 2009 |  Voldemort  | ✔  | ✔  | **O**  | **EC, record**  | **O**  | **O**  | **O**  | **O**  | **key-val**  | **nosql**  |
| 2009 |  Riak  | ✔  | ✔  | ✔  | **EC, record**  | **MR**  | **O**  |   |   | **key-val**  | **nosql**  |
| 2010 |  Dremel  | ✔  | **O**  | **O**  | **O**  | **/**  | ✔  | **O**  | ✔  | **tables**  | **sql-like**  |
| 2011 |  Megastore  | ✔  | ✔  | ✔  | **entity groups**  | **O**  | **/**  | **O**  | **/**  | **tables**  | **nosql**  |
| 2011 |  Tenzing  | ✔  | **O**  | **O**  | **O**  | **O**  | ✔  | ✔  | ✔  | **tables**  | **sql-like**  |
| 2011 |  Spark/Shark  | ✔  | **O**  | **O**  | **O**  | ✔  | ✔  | **O**  | ✔  | **tables**  | **sql-like**  |
| 2012 |  Spanner  | ✔  | ✔  | ✔  | ✔  | **?**  | ✔  | ✔  | ✔  | **tables**  | **sql-like**  |
| 2012 |  Accumulo  | ✔  | ✔  | ✔  | **record**  | **compat. w/MR**  | **/**  | **O**  | **O**  | **ext. record**  | **nosql**  |
| 2013 |  Impala  | ✔  | **O**  | **O**  | **O**  | ✔  | ✔  | **O**  | ✔  | **tables**  | **sql-like**  |

DynamoDB

Key features:

* Service Level Agreement (SLN): at the
* 99th percentile, and not on mean/median/variance (otherwise, one penalizes the heavy users)
* “Respond within 300ms for 99.9% of its requests”

Dynamo (2)

Key features:

* DHT with replication:
	+ Store value at k, k+1, …, k+N-1
	+ Eventual consistency through vector clocks
* Reconciliation at read time:
	+ Writes never fail (“poor customer experience”)
	+ Conflict resolution: “last write wins” or application specific

Vector Clocks

Each data item associated with a list of (server, timestamp) pairs indicating its version history.

Vector Clocks Example

* A client writes D1 at server SX:

D1 ([SX,1])

* Another client reads D1, writes back D2; also handled by SX:

D2 ([SX,2]) (D1 garbage collected)

* Another client reads D2, writes back D3; handled by server SY:

D3 ([SX,2], [SY,1])

* Another client reads D2, writes back D4; handled by server SZ:

D4 ([SX,2], [SZ,1])

* Another client reads D3, D4: CONFLICT !

|  |  |  |
| --- | --- | --- |
|  Data 1  | Data 2  | Conflict?  |
| ([SX,3],[SY,6])  | ([SX,3],[SZ,2])  |  |
| ([SX,3])  | ([SX,5])  |  |
| ([SX,3],[SY,6])  | ([SX,3],[SY,6],[SZ,2])  |  |
| ([SX,3],[SY,10])  | ([SX,3],[SY,6],[SZ,2])  |  |
| ([SX,3],[SY,10])  | ([SX,3],[SY,20],[SZ,2])  |  |

Configurable Consistency

* R = Minumum number of nodes that participate in a successful read
* W = Minumum number of nodes that participate in a successful write
* N = Replication factor
* If R + W > N, you can claim consistency
* But R + W < N means lower latency.