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

Rick Cattel's clustering from
"Scalable SQL and NoSQL Data Stores"
SIGMOD Record, 2010

extensible record stores

document stores

key-value stores

Rick Cattel 2010

Terminology

- Document = nested values, extensible records (think XML or JSON)
- Extensible record = families of attributes have a schema, but new attributes may be added
- Key-Value object = a set of key-value pairs. No schema, no exposed nesting

Rick Cattel 2010

NoSQL Features

- Ability to horizontally scale “simple operation” throughput over many servers
  - Simple = key lookups, read/write of 1 or few records
- The ability to replicate and partition data over many servers
  - Consider “sharding” and “horizontal partitioning” to be synonyms
- A simple API – no query language
- A weaker concurrency model than ACID transactions
- Efficient use of distributed indexes and RAM for data storage
- The ability to dynamically add new attributes to data records

Rick Cattel 2010

ACID v.s. BASE

- ACID = Atomicity, Consistency, Isolation, and Durability
- BASE = Basically Available, Soft state, Eventually consistent

Don't use "BASE" – it didn't stick.

Aside:
Consistency: "Any data written to the database must be valid according to all defined rules"

Rick Cattel 2010

Major Impact Systems (Rick Cattel)

- "**Memcached** demonstrated that in-memory indexes can be highly scalable, distributing and replicating objects over multiple nodes."
- "**Dynamo** pioneered the idea of [using] eventual consistency as a way to achieve higher availability and scalability: data fetched are not guaranteed to be up-to-date, but updates are guaranteed to be propagated to all nodes eventually."
- "**BigTable** demonstrated that persistent record storage could be scaled to thousands of nodes, a feat that most of the other systems aspire to."

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

Memcached

- Main-memory caching service
  - basic system: no persistence, replication, fault-tolerance
  - Many extensions provide these features
  - Ex: membrain, membase
- Mature system, still in wide use
- Important concept: *consistent hashing*

"Regular" Hashing

Assign M data keys to N servers

assign each key to server = k mod N

Example: N=3

key 0 -> server 0
key 1 -> server 1
key 2 -> server 2
key 3 -> server 0
key 4 -> server 1
…

data keys
k0 = 367
k1 = 452
k2 = 776
…

server 1 2 3 …. 64

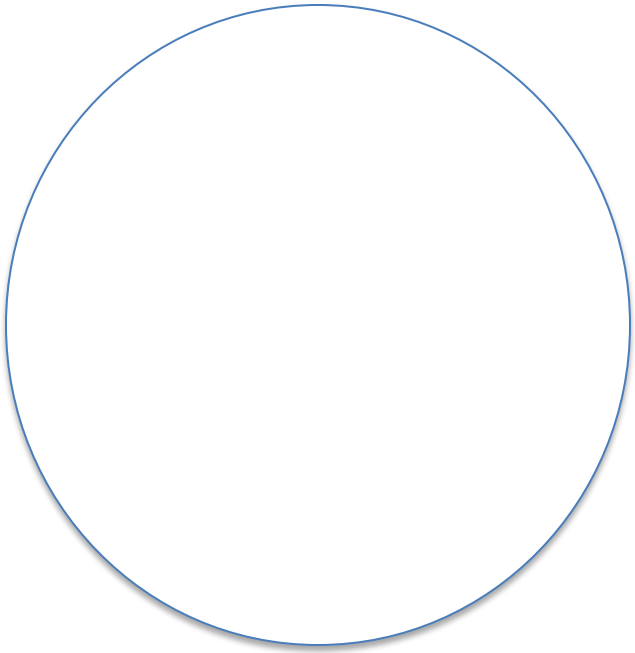
What happens if I increase the number of servers from N to 2N?

***Every** existing key needs to be remapped, and we're screwed.*

Consistent Hashing

server id = 1
server id = 2
server id = 3
…

data key = 367
data key = 452
data key = 776
…



Consistent Hashing: Routing

server id = 1
server id = 2
server id = 3
…