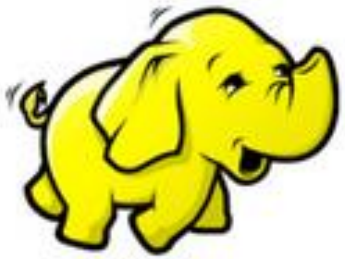


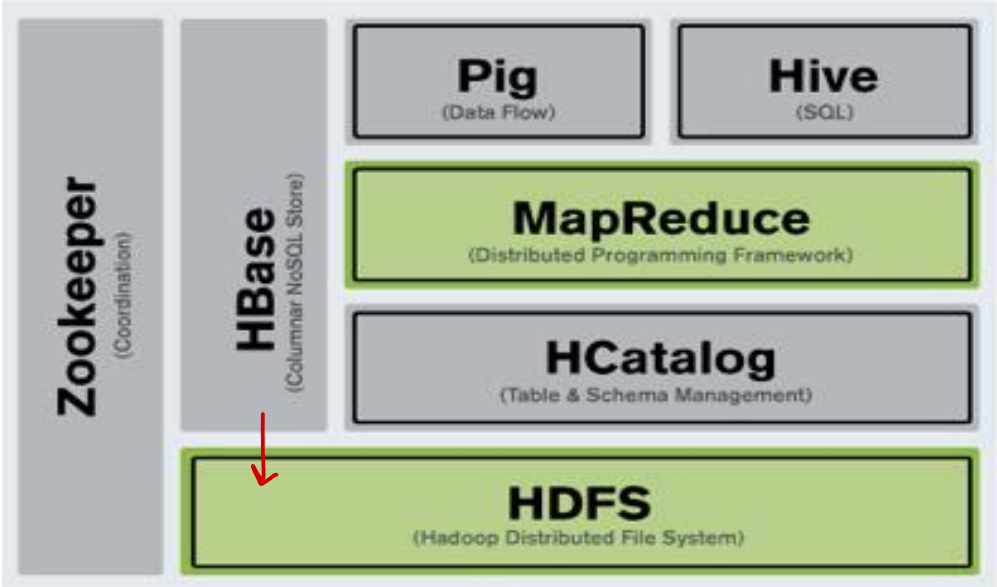
Apache HBase

Sunbeam Infotech

HBase Introduction

- HBase is an open-source non-relational distributed database.
- It is modeled after Google's Big-Table and open-sourced under Apache License.
- HBase is developed in Java as part Apache Hadoop project.
- HBase contributed by developers at Facebook, Cloudera, Hortonworks, etc.
- It is cluster based database running on top of Hadoop HDFS.
- HBase data files are stored in HDFS.

- HBase development:
  - 2006: Google Big Table paper
  - 2007: Hadoop's contrib
  - 2008: Hadoop's sub-project
  - 2010: Apache top level project
  - 2011: HBase 0.92 release



HBase vs HDFS

- Distributed systems to scale to thousands of nodes.
- HDFS – Batch processing over big files
  - Not good for record lookup.
  - Not good for small incremental batches.
  - Not designed for update & delete.
- HBase – Distributed Column database
  - Low latency record lookup (by row id/key).
  - Support for inserting & updating records. DML

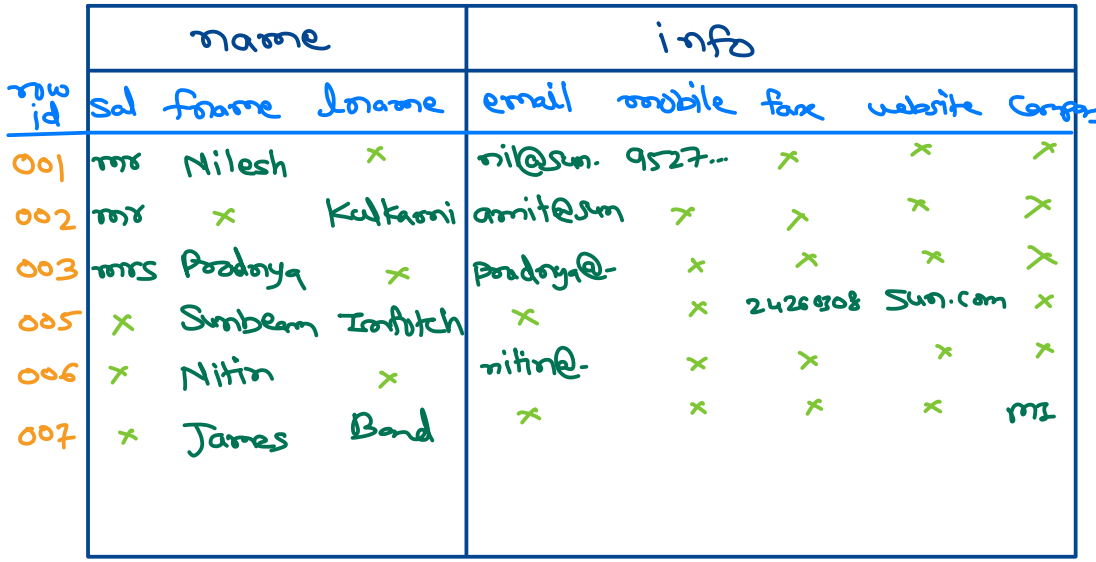
	HDFS + MR	Hbase
Write pattern	Append only	Random write, Bulk loading
Read pattern	Scan whole file	Random read, small range read or full table scan
SQL performance	Very good	Slower efficient with Phonix
Structured storage	User-defined, Avro or Sequential files	Sparse column family data model
Max data size	30+ PB	~ 1 PB

HBase vs RDBMS

	RDBMS	HBase - NoSQL
Data layout	Row oriented	Column oriented
Transaction	Multi-Row ACID	Single Row only
Query Language	SQL	get/put/scan/… ruby shell for hbase.
Security	Authentication	HDFS+Auth
Indexes	On any column	Only on row-key
Max data Size	TBs	~ 1PB
Read/Write Throughput	1000 queries/sec	million queries/sec

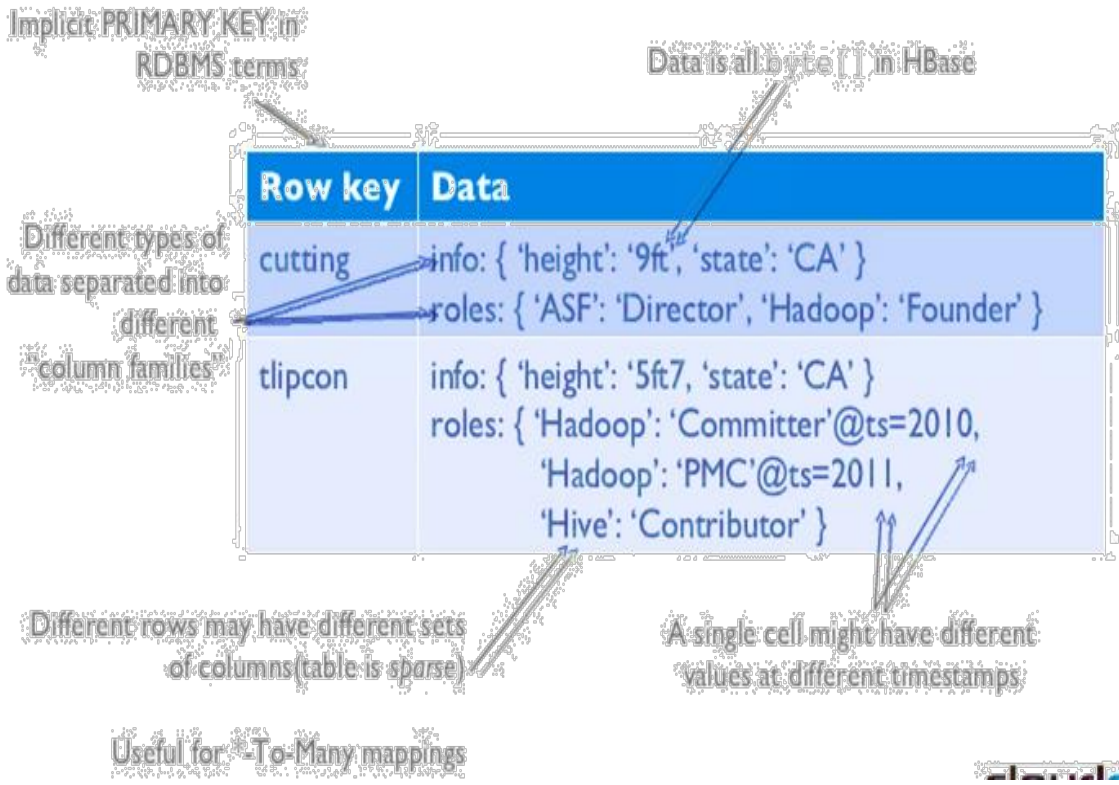
HBase - NoSQL

- HBase being NoSQL, is a schema-less database. Columns can be added on the fly.
- Sparse tables have lot of *null* values and not stored by HBase to save disk space.
- It persists all its data in underlying HDFS. Hence it is reliable, scalable, high performance at cost of distributed servers.
- Each record is associated with a key and is stored in sorted order of keys.
- Data store can store one or more tables.
- Each row in table is indexed by row key.
- Column oriented database:
  - Each table can have one or more column families.
  - Each column family have one or more columns.
  - Columns in family may be different for each row.
- Columns can be added in family dynamically.
- Multiple versions of the values (for each update) as per timestamps (by default).



HBase Data Model

Row ID	Column Family -- Coulmn Qualifiers
001	name : { fname : 'nilesh', lname : 'ghule' } details : { email : 'nilesh@sunbeaminfo.com', mobile : '9527331338', mobile2: '7722093091' }
002	name : { fname : 'nitin', lname : 'kudale' } details : { mobile : '9881208115' }
003	name : { fname : 'sunbeam' } details : { site : 'www.sunbeaminfo.com' mobile : '9881208115', mobile2: '9881208114' phone : '02024260308' }



HBase Data Model

- Intersection of row and column is a cell.
- All cells, row ids, even table, column family & column names are stored as byte array.
- Thus any data type of any size can be stored in each cell.
- With each edit, a new version of the cell is created with new time stamps. Internally stores versions in desc order of time-stamps.
- Key & Version numbers are replicated with each column family.
- Empty cells aren't stored.

Sorted on disk by Row key, Col key, descending timestamp

Row key	Column key	Timestamp	Cell value
cutting	roles:ASF	1273871823022	Director
cutting	roles:Hadoop	1183746289103	Founder
tlipcon	roles:Hadoop	1300062064923	PMC
tlipcon	roles:Hadoop	1293388212294	Committer
tlipcon	roles:Hive	1273616297446	Contributor

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>