CS 245: Database System Principles

Notes 13: BigTable, HBASE, Cassandra

Hector Garcia-Molina

Sources

- HBASE: The Definitive Guide, Lars George, O'Reilly Publishers, 2011.
- Cassandra: The Definitive Guide, Eben Hewitt, O'Reilly Publishers, 2011.
- BigTable: A Distributed Storage System for Structured Data, F. Chang et al, ACM Transactions on Computer Systems, Vol. 26, No. 2, June 2008.

Lots of Buzz Words!

- "Apache Cassandra is an open-source, distributed, decentralized, elastically scalable, highly available, fault-tolerant, tunably consistent, column-oriented database that bases its distribution design on Amazon's dynamo and its data model on Google's Big Table."
- Clearly, it is buzz-word compliant!!

Basic Idea: Key-Value Store

Table T:

key	value
k1	v1
k2	v2
k3	v3
k4	v4

Basic Idea: Key-Value Store

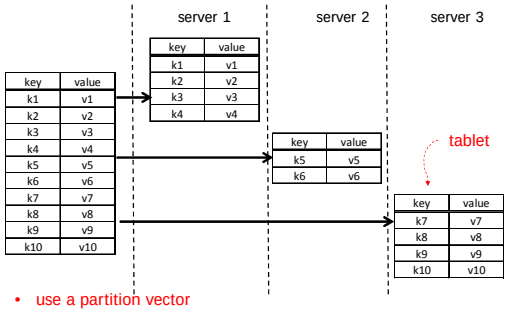
Table T:

key	value
k1	v1
k2	v2
k3	v3
k4	v4

keys are sorted

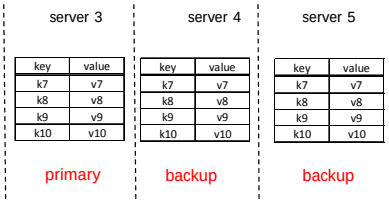
- API:
  - lookup(key) → value
  - lookup(key range) → values
  - getNext → value
  - insert(key, value)
  - delete(key)
- Each row has timestemp
- Single row actions atomic
  (but not persistent in some systems?)
- No multi-key transactions
- No query language!

Fragmentation (Sharding)



- use a partition vector
- "auto-sharding": vector selected automatically

Tablet Replication

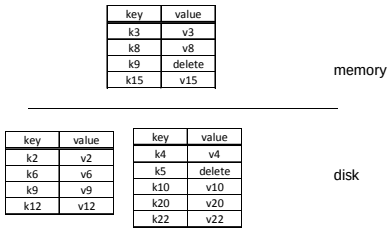


- Cassandra:
  Replication Factor (# copies)
  R/W Rule: One, Quorum, All
  Policy (e.g., Rack Unaware, Rack Aware, ...)
  Read all copies (return fastest reply, do repairs if necessary)
- HBase: Does not manage replication, relies on HDFS

Need a "directory"

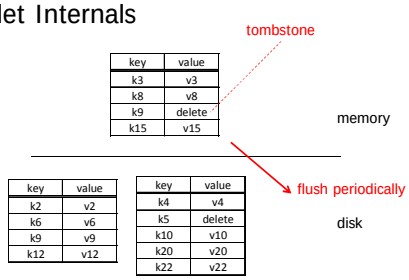
- Table Name: Key → Server that stores key
  → Backup servers
- Can be implemented as a special table.

Tablet Internals



Design Philosophy (?): Primary scenario is where all data is in memory.
Disk storage added as an afterthought

Tablet Internals



- tablet is merge of all segments (files)
- disk segments imutable
- writes efficient; reads only efficient when all data in memory
- periodically reorganize into single segment

Column Family

K	A	B	C	D	E
k1	a1	b1	c1	d1	e1
k2	a2	null	c2	d2	e2
k3	null	null	null	d3	e3
k4	a4	b4	c4	e4	e4
k5	a5	b5	null	null	null

Column Family

K	A	B	C	D	E
k1	a1	b1	c1	d1	e1
k2	a2	null	c2	d2	e2
k3	null	null	null	d3	e3
k4	a4	b4	c4	e4	e4
k5	a5	b5	null	null	null

- for storage, treat each row as a single "super value"
- API provides access to sub-values
  (use family:qualifier to refer to sub-values
  e.g., price:euros, price:dollars)
- Cassandra allows "super-column":
  two level nesting of columns
  (e.g., Column A can have sub-columns X & Y)

Vertical Partitions

K	A	B	C	D	E
k1	a1	b1	c1	d1	e1
k2	a2	null	c2	d2	e2
k3	null	null	null	d3	e3
k4	a4	b4	c4	e4	e4
k5	a5	b5	null	null	null

can be manually implemented as

K	A
k1	a1
k2	a2
k4	a4
k5	a5

K	B
k1	b1
k4	b4
k5	b5

K	C
k1	c1
k2	c2
k4	c4

K	D	E
k1	d1	e1
k2	d2	e2
k3	d3	e3
k4	e4	e4

Vertical Partitions

K	A	B	C	D	E
k1	a1	b1	c1	d1	e1
k2	a2	null	c2	d2	e2
k3	null	null	null	d3	e3
k4	a4	b4	c4	e4	e4
k5	a5	b5	null	null	null

column family

K	A
k1	a1
k2	a2
k4	a4
k5	a5

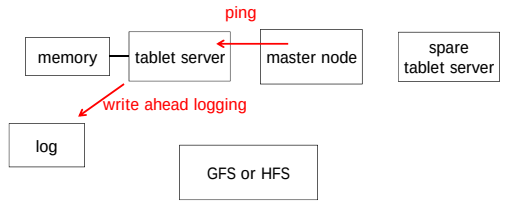
K	B
k1	b1
k4	b4
k5	b5

K	C
k1	c1
k2	c2
k4	c4

K	D	E
k1	d1	e1
k2	d2	e2
k3	d3	e3
k4	e4	e4

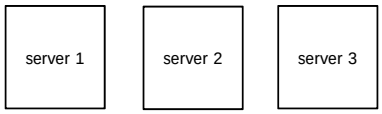
- good for sparse data;
- good for column scans
- not so good for tuple reads
- are atomic updates to row still supported?
- API supports actions on full table; mapped to actions on column tables
- API supports column "project"
- To decide on vertical partition, need to know access patterns

Failure Recovery (BigTable, HBase)



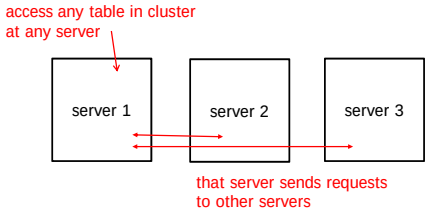
Failure recovery (Cassandra)

- No master node, all nodes in "cluster" equal



Failure recovery (Cassandra)

- No master node, all nodes in "cluster" equal



Bonus Slides*: Are Traditional Databases Dead?

- Heard on Twitter:
  - noSQL rules
  - new DB systems scale better than old ones
  - DBMS too slow …
- Therefore, need new, revolutionary technology!!

* WARNING: Author may be biased :-)

Cautionary Tale

- Lawrence Richard Walters, nicknamed "**Lawnchair Larry**" or the "Lawn Chair Pilot", (April 19, 1949 – October 6, 1993) was an American truck driver who took flight on July 2, 1982 in a homemade aircraft. Dubbed Inspiration I, the "flying machine" consisted of an ordinary patio chair with 45 helium-filled weather balloons attached to it. Walters rose to an altitude of over 15,000 feet (4,600 m) and floated from his point of origin in San Pedro, California into controlled airspace near Los Angeles International Airport.

Associated Press 1983
and http://www.flightdata.com

Parallels

- **Lawnchair Larry**
  - Wanna fly
  - Can't afford airplane
- **T-Gen**
  - Wanna DB services
  - Can't afford real DBMS

Parallels

- **Lawnchair Larry**
  - Wanna fly
  - Can't afford airplane
  - I can do myself!
  - I am off!!
- **T-Gen**
  - Wanna DB services
  - Can't afford real DBMS
  - I can do myself!
  - I am off!!

Parallels

- **Lawnchair Larry**
  - Wanna fly
  - Can't afford airplane
  - I can do myself!
  - I am off!!
  - How do I land???
- **T-Gen**
  - Wanna DB services
  - Can't afford real DBMS
  - I can do myself!
  - I am off!!
  - I need joins???

Parallels

- **Lawnchair Larry**
  - Wanna fly
  - Can't afford airplane
  - I can do myself!
  - I am off!!
  - How do I land???
  - How talk ATC?
  - How to navigate?
  - Need oxygen!!??
- **T-Gen**
  - Wanna DB services
  - Can't afford real DBMS
  - I can do myself!
  - I am off!!
  - I need joins???
  - How to index?
  - Just had crash! Now what?
  - Data inconsistent!
  - Oh? Need to maintain???

Keep Lawnchair Larry in Mind

- Does DBMS technology not cut it and we need to start from scratch??
  - Or are you just being cheap? ☺
- If you think you need a subset of DBMS, will needs change over time?