Bigtable: A Distributed Storage System For Structured Data

Fay Chang, Jeffrey Dean, et al.

OSDI'06: Seventh Symposium on Operating System Design And Implementation

Cassandra - A Decentralized Structured Storage System

Avinash Lakshman , Prashant Malik

ACM SIGOPS Operating Systems Review 2010

Presented By :Polykarpos Thomadakis

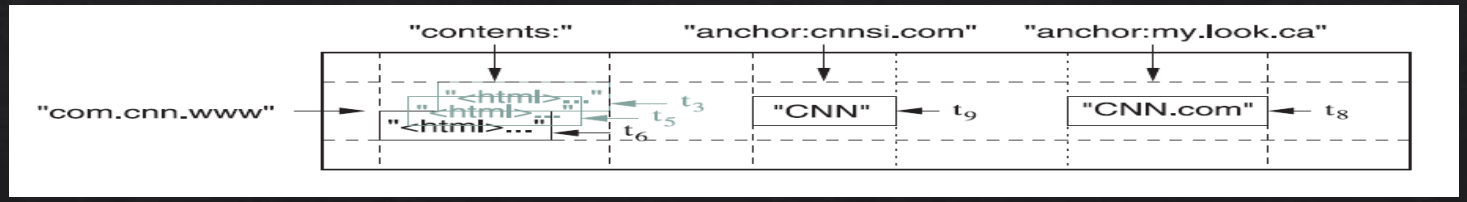
CS 834 – Introduction to Information Retrieval

Fall 2017

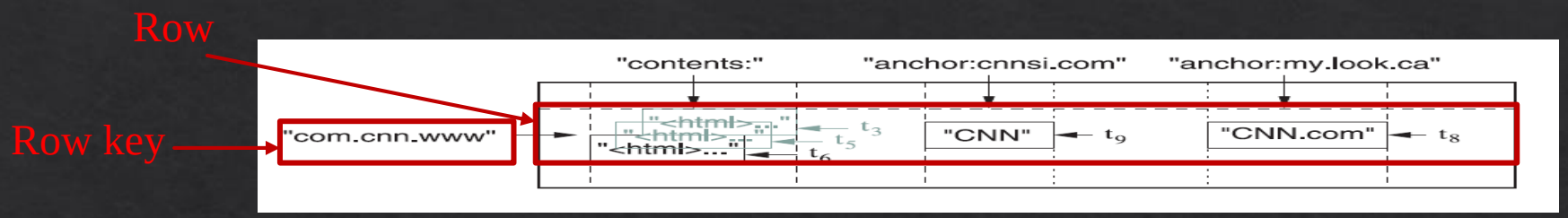
Old Dominion University

Big Table

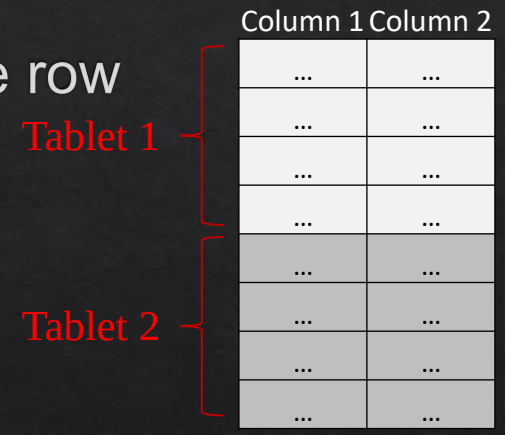
- A sparse, distributed, persistent multidimensional sorted map
- Indexed by a row key, column key, and a timestamp
- Each value in the map is an uninterpreted array of bytes
  - (row:string, column:string, time:int64) → string
- Not a full relational data model
- Supports dynamic control over data layout and format



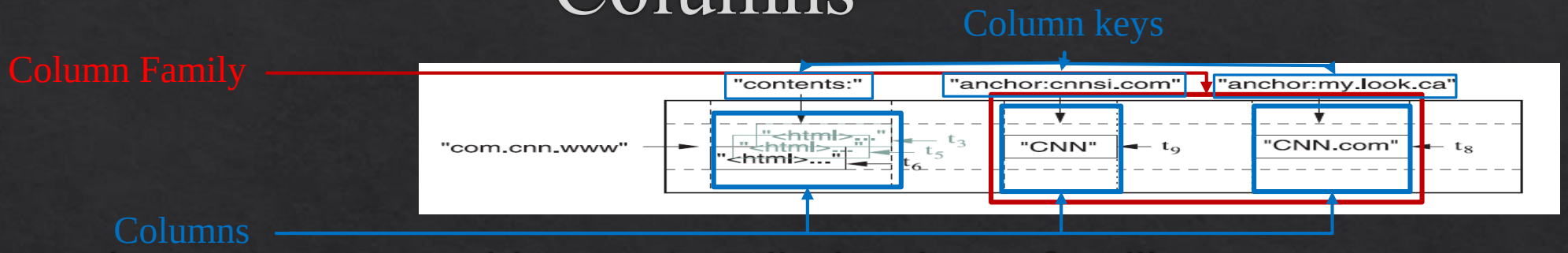
Rows



- Row keys are arbitrary strings
- Reads/Writes under a single row key are atomic
  - Makes it easier to reason about concurrent updates on the row
- Data maintained in *lexicographic* order by row key
- Rows with consecutive keys consist a *tablet*
  - Unit of distribution and load balancing
- As a result, reads of *short* row ranges require communication with only a *small* number of machines
  - Select *good keys* to get *good locality* for data accesses

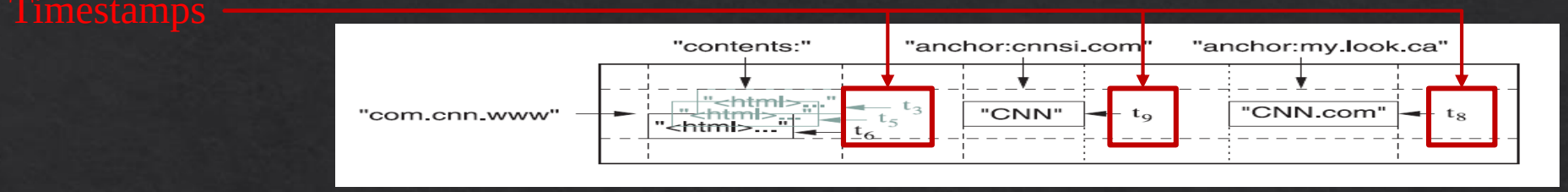


Columns



- Column keys are grouped into *sets* called *column families*
  - Basic unit of *access control*
  - Columns in a family are usually of the *same type* (compressed together)
- A column key must belong to a family to store data in it
  - Accessed using the syntax : *family:quantifier*
- *Column families* should be *small* in number (hundreds at most) and change *rarely*
- In contrast, number of columns is not bounded

Timestamps



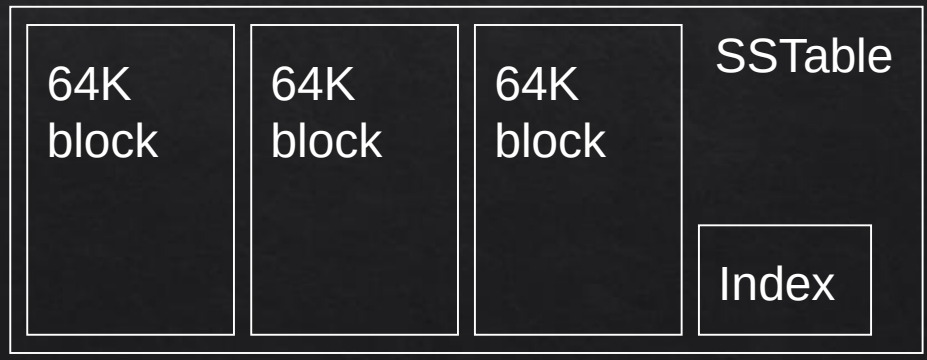
- Each cell can contain *multiple* versions of the same data
  - Indexed by *timestamps*
- *Timestamps* can be assigned by
  - Bigtable, *real time* representation in microseconds
  - Client application
- Different versions stored in decreasing timestamp order
  - Most recent version is read first
- Automatic garbage collection
  - Keep n last versions of the data
  - Keep all versions not older than a time period

Building Blocks

- GFS
  - Stores log and data files
  - Uses SSTables
- Cluster management system
  - Scheduling jobs
  - Managing resources
  - Dealing with machine failures
  - Monitoring machine status
- Chubby
  - Highly-available and persistent distributed file and lock service

SSTables

- An ordered immutable map from keys to values
  - Keys and values arbitrary byte strings
- Contains a sequence of data blocks plus an index
  - Index is loaded into memory when SSTable is opened
- A lookup can be performed with a single disk seek
  - Find block from in memory index
  - Read the block from disk
- Can optionally be mapped into memory
  - Without touching the disk



Implementation

Three major components

- A library linked into every client
- One master server
- Many tablet servers

Master server

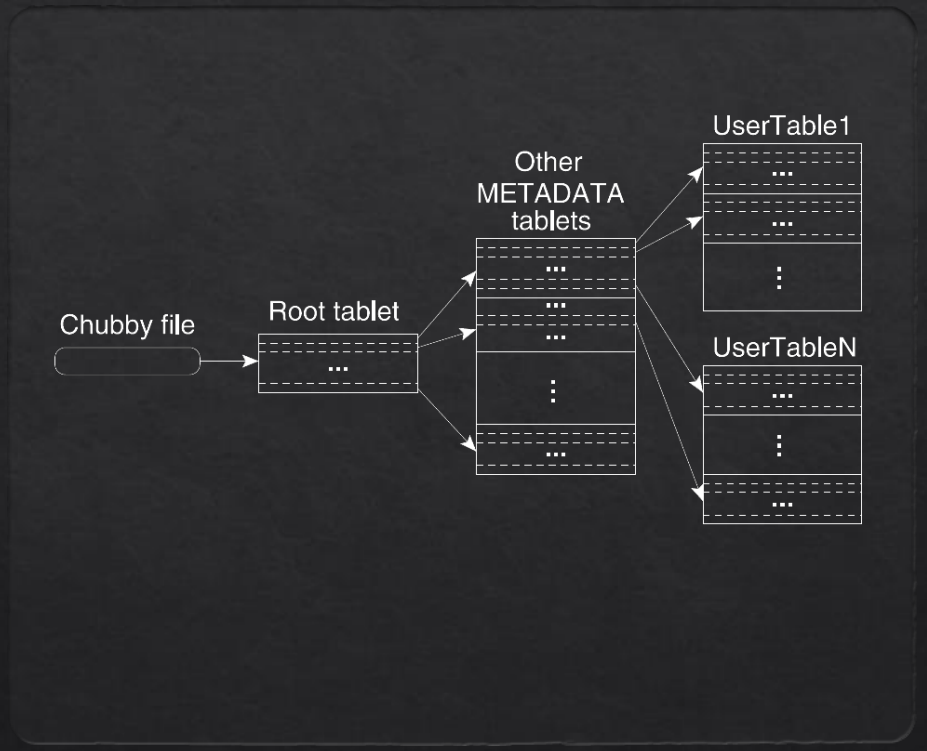
- Assigns tablets to tablet servers
- Detects addition and expiration of tablet servers
- Performs load balancing
- Garbage collection
- Schema changes (e.g table and family column creations)

Table server

- Manages a set of tablets
- Handles read/write requests to its tablets
- Splits tables that have grown too large

Tablet Location

- Three-level hierarchy
  - File stored in Chubby, contains the location of the root tablet
  - Root tablet contains the location of all tablets in special METADATA table
  - User tablets
- Client libraries cache tablet locations
- If tablet location unknown for a client
  - Recursively move up the tablet location hierarch

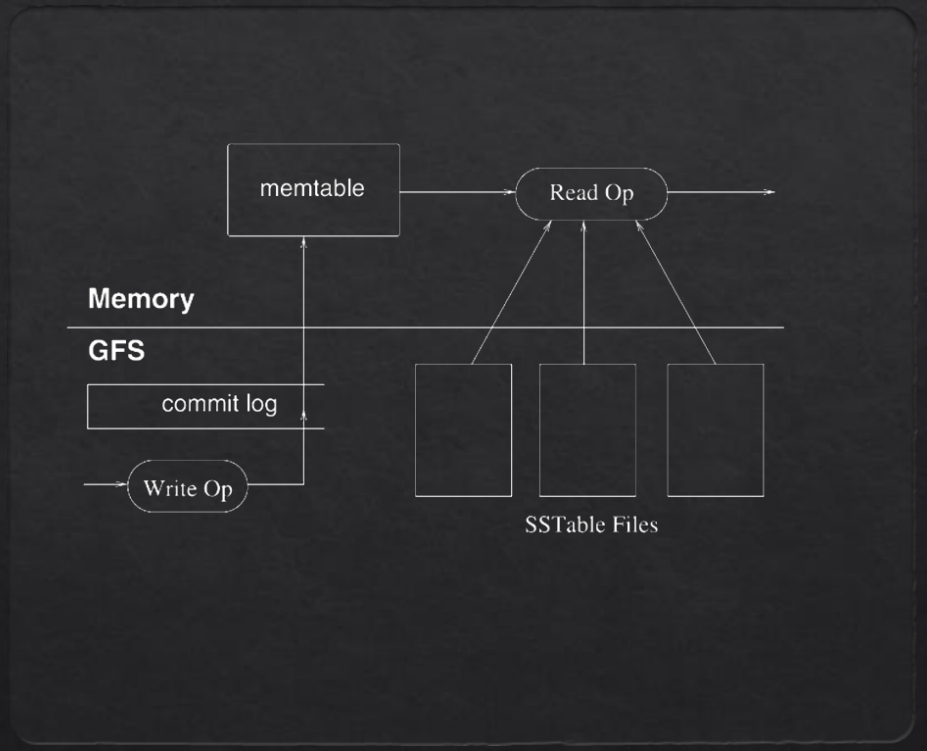


Tablet Assignment

- Each tablet is assigned to one tablet server at a time
- Master keeps track of live tablet servers, assignment of tablets to servers and unassigned tablets
- Chubby is used to keep track of tablet servers
  - Tablet server starts -> creates and acquires lock on a uniquely name file in a Chubby directory.
  - This directory is used by master to discover servers
- A tablet server stops serving its tablets if it loses its lock (due to some failure)
  - Will attempt to reacquire lock if file still exists
  - Kills itself if it fails

Tablet Serving

- The *persistent* state of a tablet is stored in *GFS*
- Updates are committed to a *commit log*
  - Recently commited ones stored sorted in a buffer called *memtable*
  - Older ones in a sequence of SSTable files
- To *recover* a tablet, a server
  - Reads metadata from the METADATA table containing
    - A list of *SSTables*
    - A set of *redo points*



Tablet Serving Operations

Write Operations

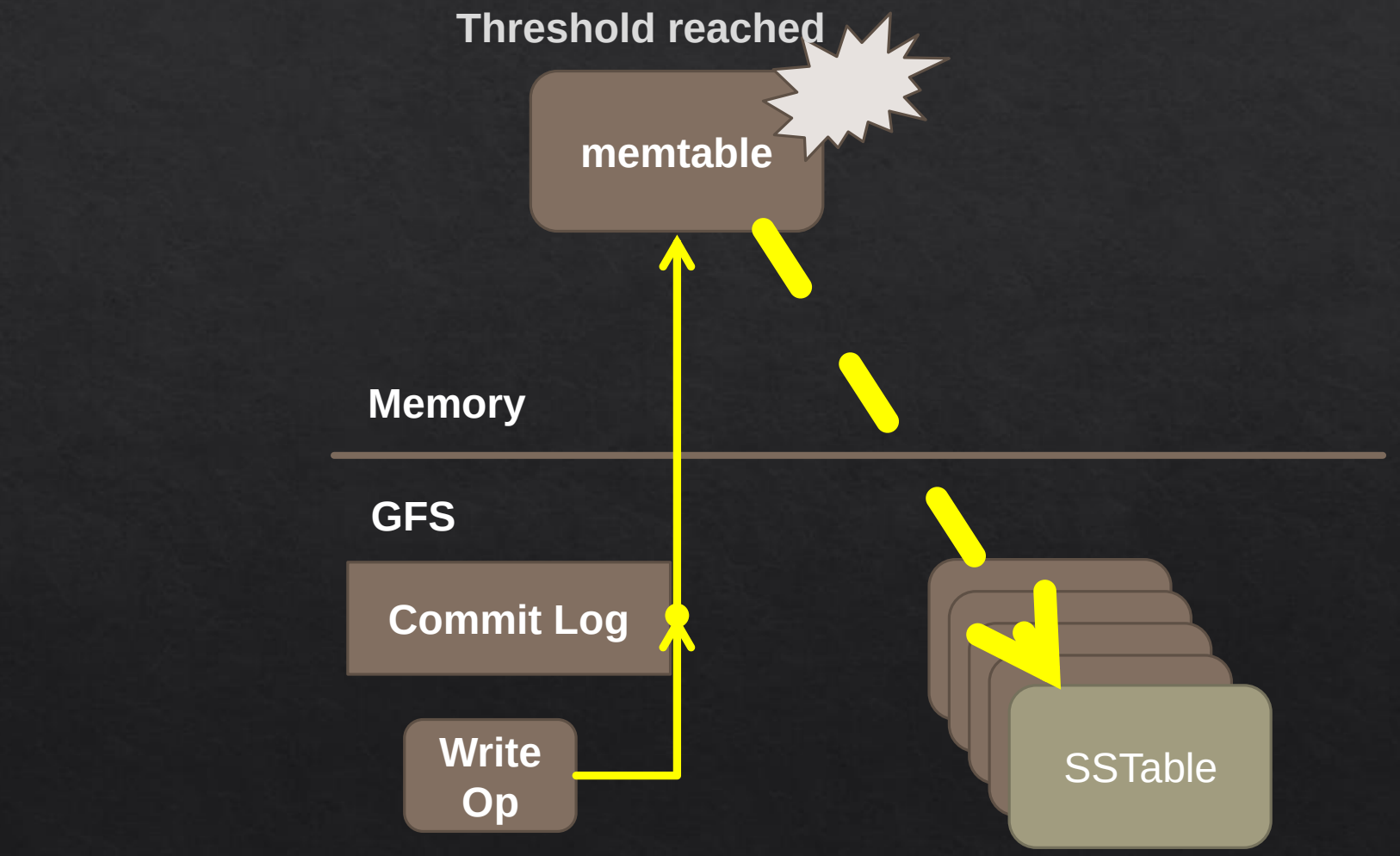
- Server checks if operation is *valid*
- A valid mutation is written to the *commit log*
- *After commitment*, the contents of the write are inserted into the *memtable*

Read Operations

- Server checks if operation is *valid*
- A valid operation is executed on a merged view of the sequence of SSTables and the memtable
- Efficient since both are stored lexicographically sorted

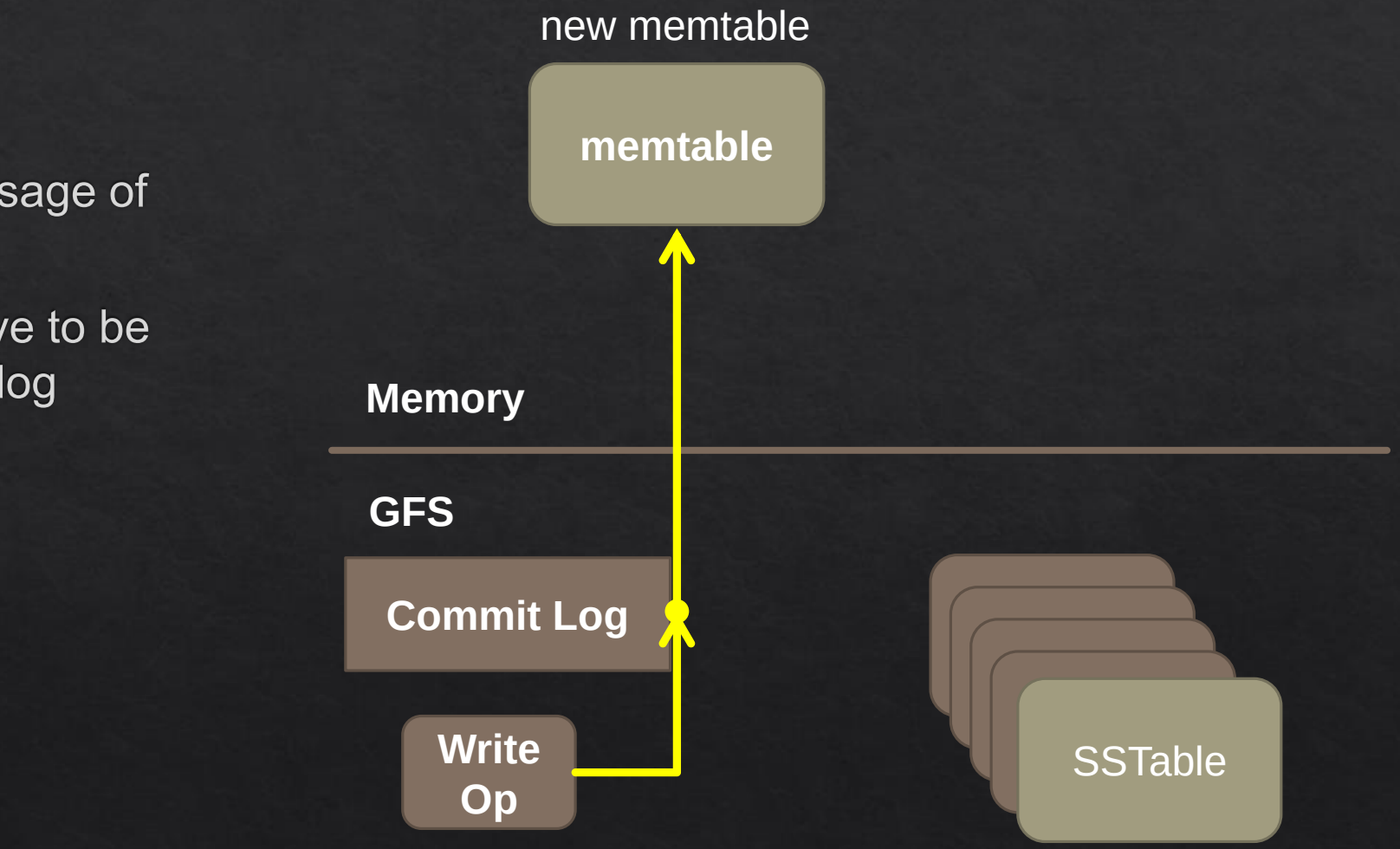
Compactions

- Minor compaction



Compactions

- Minor compaction
  - Shrinks the memory usage of the tablet server
  - Reduces data that have to be read from the commit log during recovery



Compactions

- Merging Compaction
  - Number of SSTables increases arbitrary
  - Bound their number by merging a set of SSTables and the memtable into a new SSTable
  - The SSTables and memtable can be discarded after compaction
  - Can contain special deletion entries
- Major Compaction
  - Rewrites all SSTables into one SSTable
  - Contains no deleted information or data
  - Allows to reclaim resources
  - Ensures deletion of data in a timely fashion, important for sensitive data

Cassandra

- A *distributed storage system*
- Managing very large amounts of structured data
  - Spread out across many commodity servers
- *Highly available* service
- *No single point of failure*
- Aimed to run on top of *hundreds of nodes*

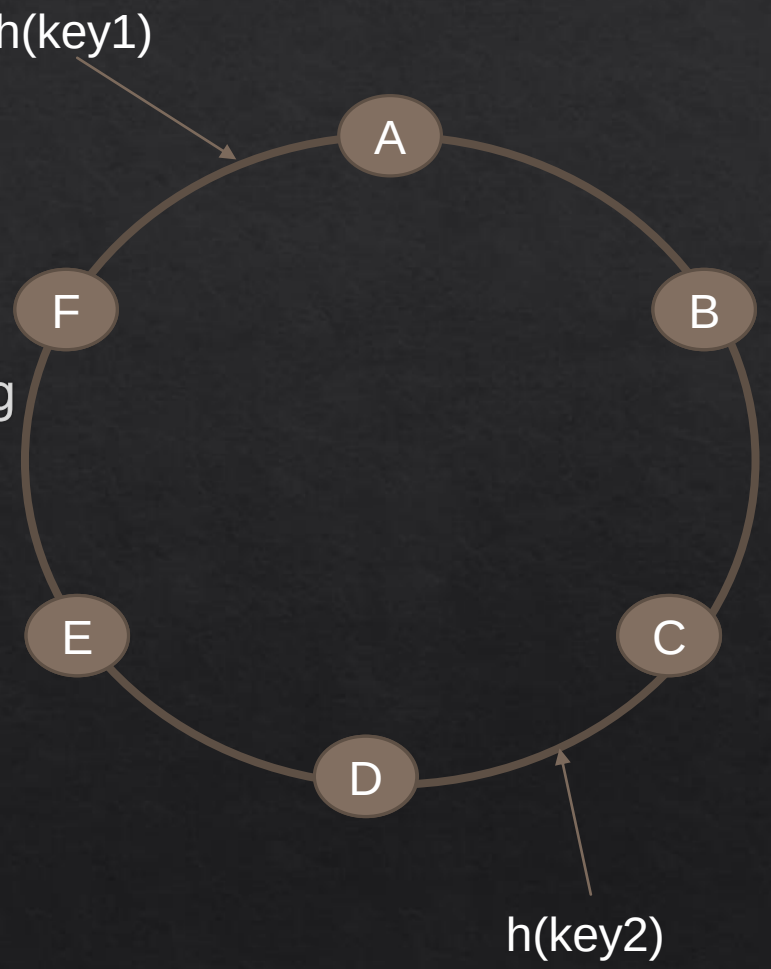
Data Model

- Similar to Bigtable
- A distributed multi dimensional map indexed by a key
- Row key in a table is a string with no size restrictions
- Operations atomic per row
- Columns grouped into families
- Supports two kinds of families
  - Simple
  - Super
    - Can be visualized as a column family within a column family
  - Sorting by Time or Name

Access -> family : super_family : column

Architecture

- Partioning
  - Order preserving hash function
  - Output range is in a ring
  - Nodes assigned a key representing their position in the ring
  - Data items given a key in the ring
    - Assigned to the next node moving clockwise (coordinator)
  - Departure/Arrival of a node affects only its neighbors



Architecture

- Membership
  - Use *gossip* to find other nodes in the system
  - Periodic, Pairwise, inter-node communication
  - Low frequency communication ensures low cost.
  - Random selection of peers.
  - Round by round doubling the peers makes protocol very *robust*.

Architecture

- Replication
  - Data items replicated at N hosts
  - Coordinator responsible
  - Three policies:
    - Rack Unaware
    - Rack Aware
    - Datacenter Aware
- Failure Detection
  - Accrual Failure Detector
  - Assign a sliding window of inter-arrival times of gossip messages per node
    - Assume node as down when threshold is reached

Architecture

- Bootstrapping
  - Two ways to add new node
    - New node gets assigned a random token which gives its position in the ring. It gossips its location to rest of the ring
    - New node reads its config file to contact its initial contact points.
- Scaling
  - New nodes assigned appropriately to alleviate a heavily loaded node
  - Splits the range of data the other node was responsible for

Architecture

- Local Persistence
  - Writes are commited into a log file
  - After commiting the update is performed in memory
  - Much like Bigtable (including compactions)
- Requests
  - Writes: System routes requests to replicas and waits for a quorum of replicas to acknowledge completion
  - Reads: System routes requests to the closest replica or to all replicas waiting for a quorum of responses