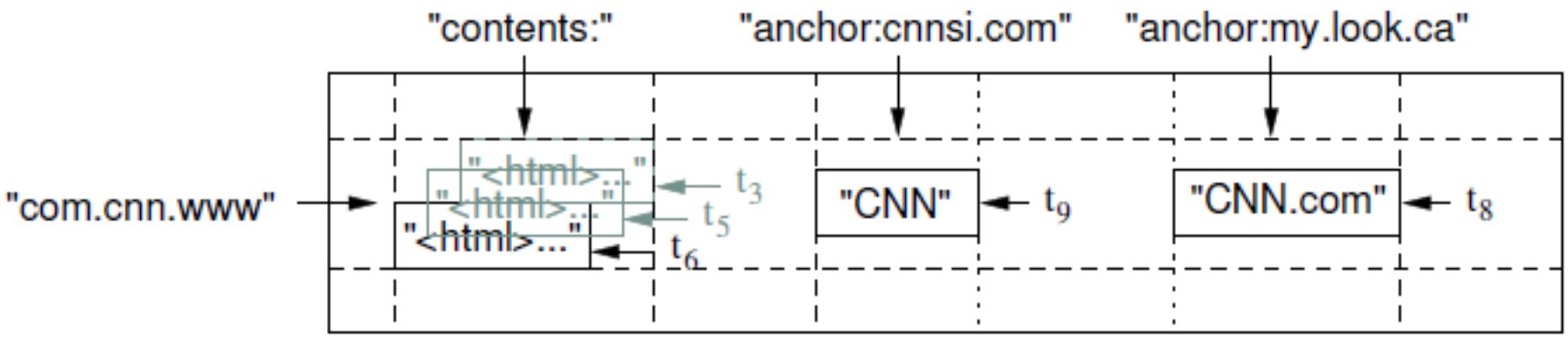
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
UNIVERSITY OF CRETE

HY590.45
Modern Topics in Scalable Storage Systems

Kostas Magoutis
magoutis@csd.uoc.gr
http://www.csd.uoc.gr/~hy590-45

Bigtable data model



Writing to a table

```
// Open the table
Table *T = OpenOrDie("/bigtable/web/webtable");

// Write a new anchor and delete an old anchor
RowMutation r1(T, "com.cnn.www");
r1.Set("anchor:www.c-span.org", "CNN");
r1.Delete("anchor:www.abc.com");
Operation op;
Apply(&op, &r1);
```

Reading from a table

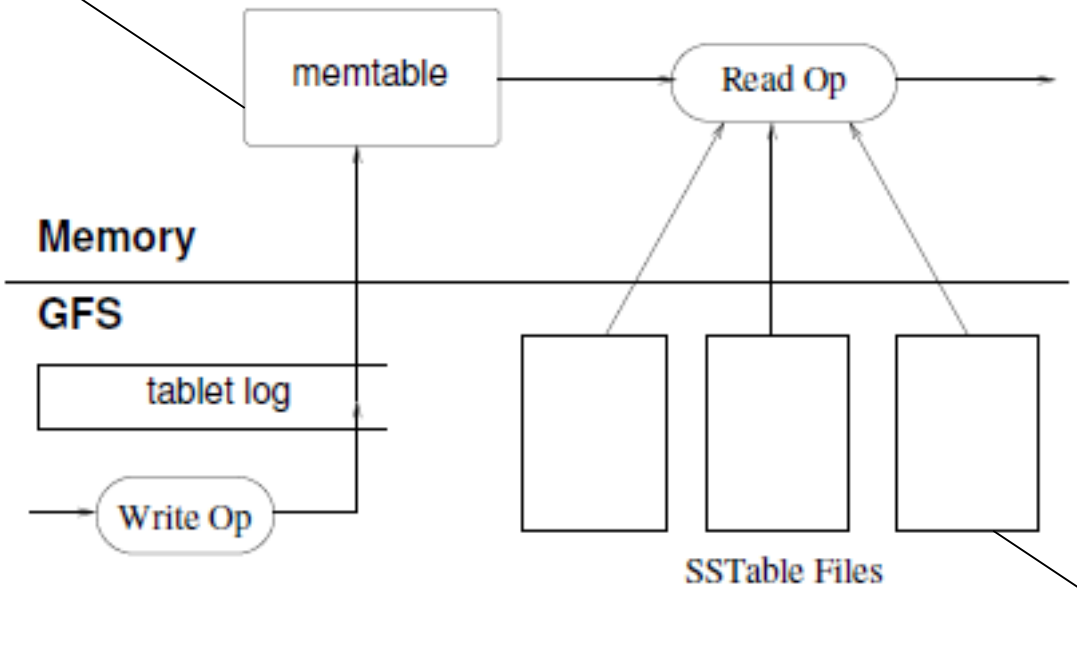
```
Scanner scanner(T);
ScanStream *stream;
stream = scanner.FetchColumnFamily("anchor");
stream->SetReturnAllVersions();
scanner.Lookup("com.cnn.www");
for (; !stream->Done(); stream->Next()) {
  printf("%s %s %lld %s\n",
         scanner.RowName(),
         stream->ColumnName(),
         stream->MicroTimestamp(),
         stream->Value());
}
```

High-level architecture

- Each table is split into *tablets*
- Each tablet is assigned to a tablet server
- Single master, infrequently used
- Assisted by
  - Chubby, for coordination
  - GFS, for file and log storage

Tablet representation

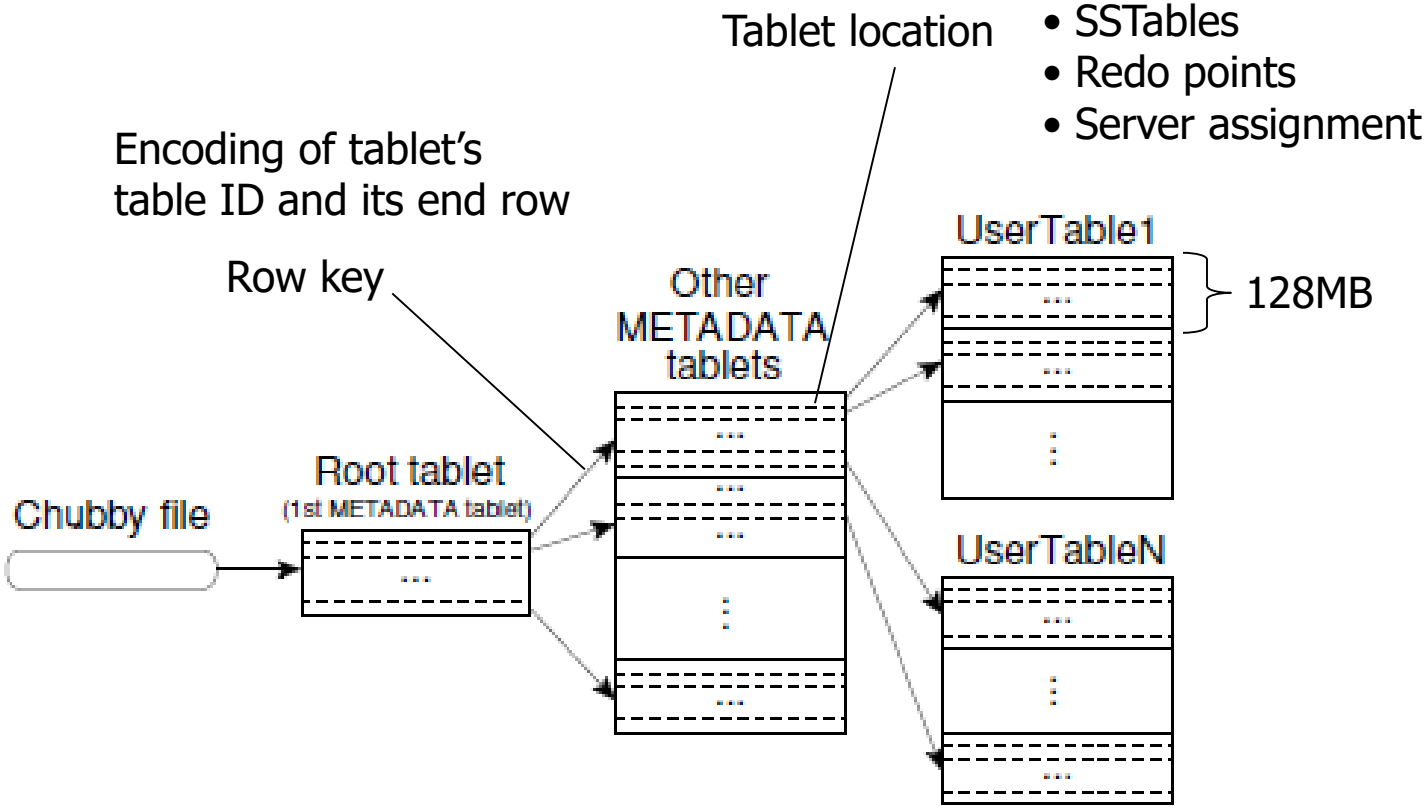
Sorted memory buffer storing recently committed updates



Ordered immutable map from keys to values

- How do compactions work

Metadata: Tablet location



- Clients directly read and cache metadata
- Master, tablet servers update metadata

Tablet master

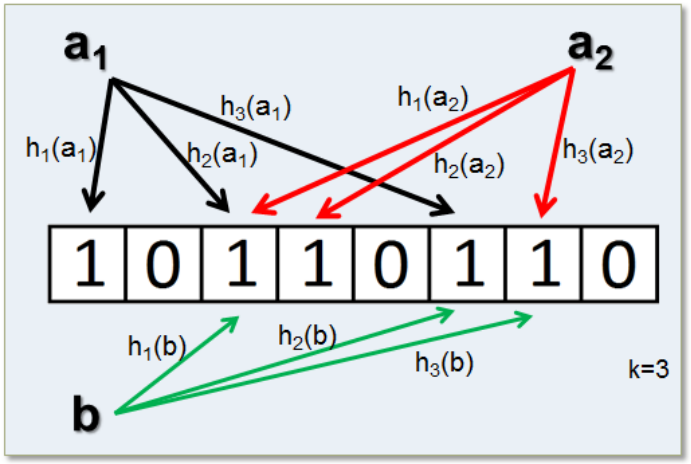
- Responsible for
  - Tablet creation, deletion, merge
    - Splits performed by tablet servers
  - Detecting addition/expiration of tablet servers
    - Based on leases issued by Chubby
  - Assigning tablets to table servers
    - Looks for server with sufficient room, load
    - Discovers current tablet assignment at startup
  - Balancing tablet-server load
    - Migrates tablets between servers when necessary
  - Garbage collection of files in GFS

Coordination service ("Chubby")

- Highly-available lock service based on Paxos
- API
  - File operations (create, read, write, delete, etc.)
  - Locking (acquire, release)
  - Consistent caching
- Bigtable uses Chubby to
  - Ensure at most one active master at a time
  - Discover tablet servers, finalize tablet deaths
  - Store bootstrapping info, schemas, ACLs

Refinements

- Locality groups
- Compression
- Caching for read performance
- Bloom filters



Source: https://redislabs.com/blog/rebloom-bloom-filter-datatype-redis/

Refinements

- Commit-log implementation
- Speeding-up tablet recovery
- Exploiting immutability

Lessons learned

- Large distributed systems are vulnerable to many types of failures, not just partitions and fail-stop
- Important to delay new features until it is clear how these features will be used
- Important to have system-level monitoring
- Value of simple designs