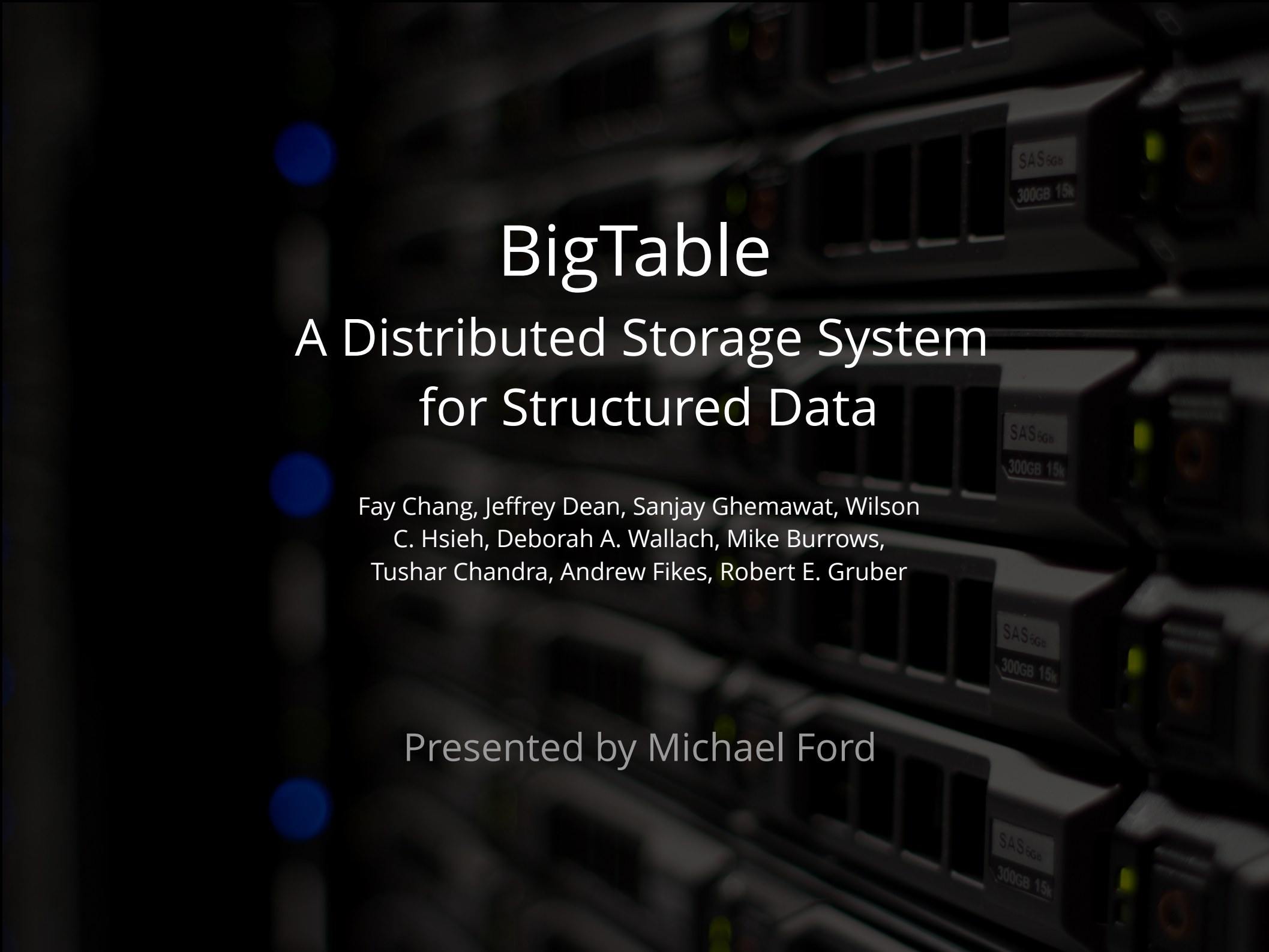


BigTable

A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber

Presented by Michael Ford

Google™ code

YouTube

Blogger™

Google books

NoSQL

- Non SQL? Non relational? Not only SQL?
- DBMS without relational model
- Key Features:
  - Simplicity of data model
  - Control of availability
  - Horizontal scaling
  - Flexibility

NoSQL – CAP Theorem

- Impossible for a distributed system to accomplish all three of:
  - Consistency
  - Availability
  - Partition Tolerance
- NoSQL systems often sacrifice **consistency** by choosing **eventual consistency** over ACID Transactions


KEY-VALUE
ORACLE
Coherence
redis
IBM
DB2
neo4j
INFINITEGRAPH
Couchbase
cassandra
mongoDB
APACHE
HBASE

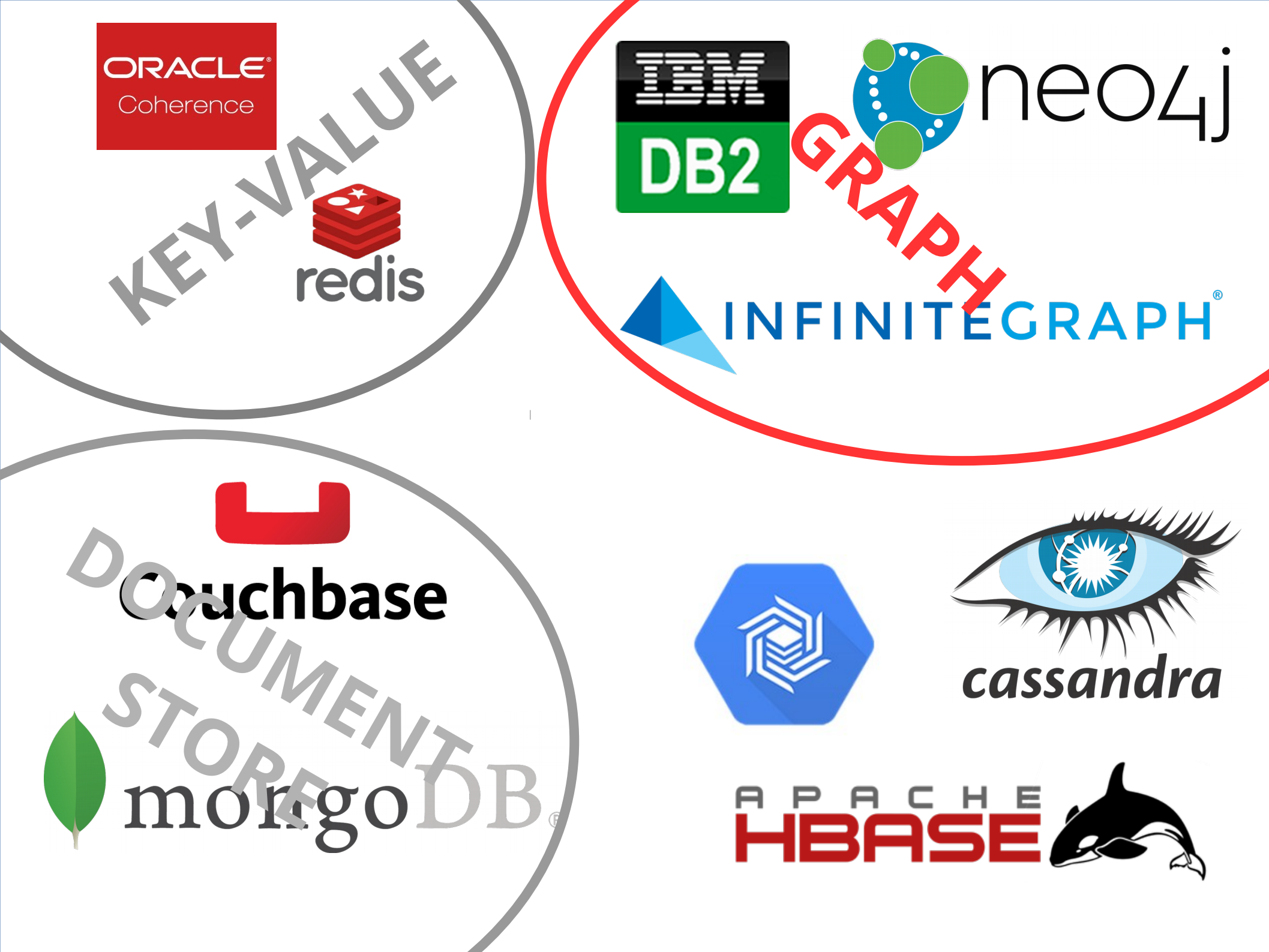
Key	Value
“India”	{“B-25, Sector-58, Noida, India – 201301”
“Romania”	{“IMPS Moara Business Center, Buftea No. 1, Cluj-Napoca, 400606″,City Business Center, Coriolan Brediceanu No. 10, Building B, Timisoara, 300011”}
“US”	{“3975 Fair Ridge Drive. Suite 200 South, Fairfax, VA 22033”}

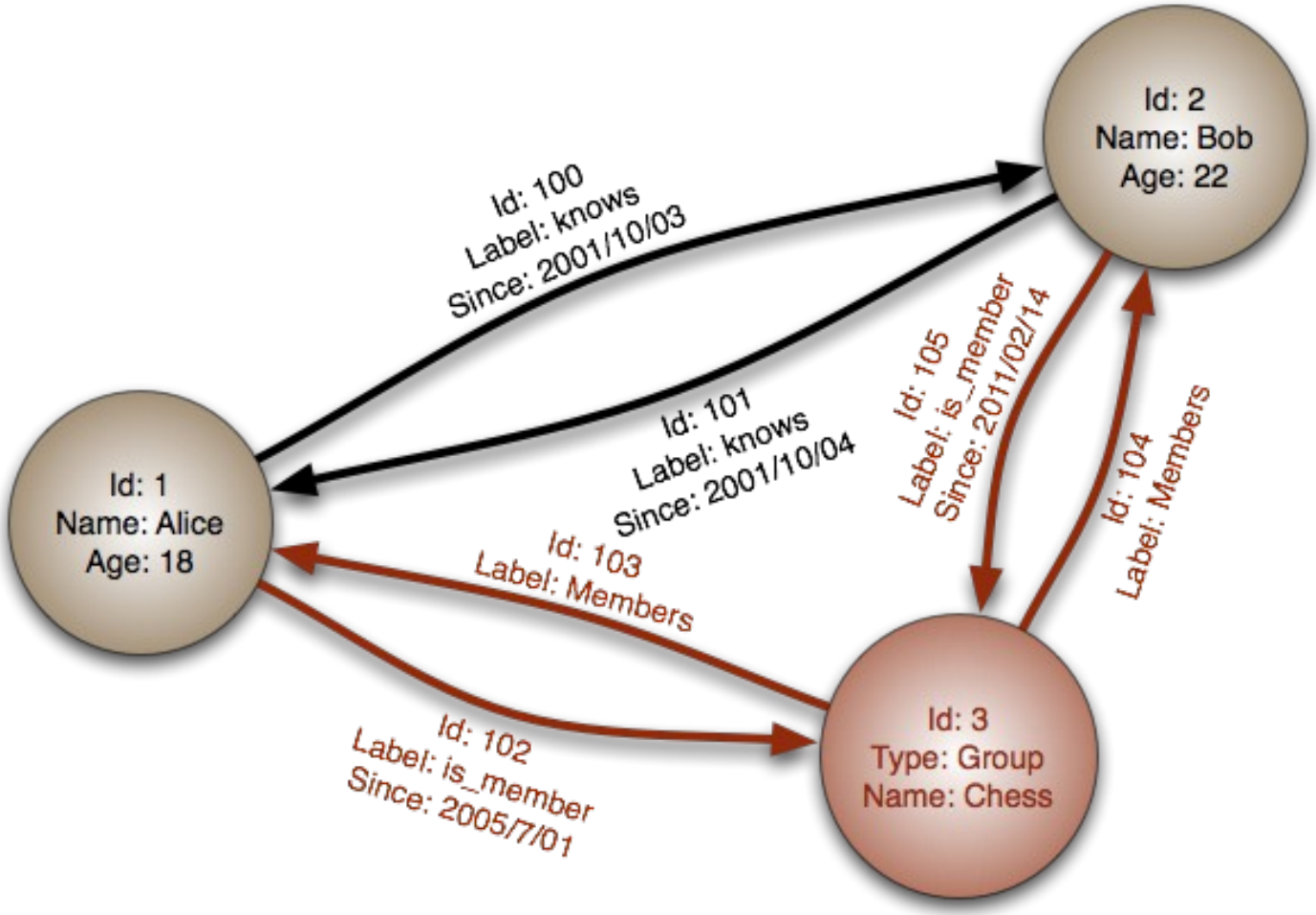
Source https://www.3pillarglobal.com/insights/exploring-the-different-types-of-nosql-databases


ORACLE
Coherence
KEY-VALUE
redis
IBM
DB2
neo4j
INFINITEGRAPH
DOCUMENT
STORE
Couchbase
mongoDB
cassandra
APACHE
HBASE

```
{officeName:”3Pillar Noida”,
{Street: “B-25, City:”Noida”, State:”UP”, Pincode:”201301”}
}
{officeName:”3Pillar Timisoara”,
{Boulevard:”Coriolan Brediceanu No. 10”, Block:”B, Ist Floor”, City: “Timisoara”, Pinc
}
{officeName:”3Pillar Cluj”,
{Latitude:”40.748328”, Longitude:”-73.985560”}
}
```

Source https://www.3pillarglobal.com/insights/exploring-the-different-types-of-nosql-databases


ORACLE
Coherence
KEY-VALUE
redis
IBM
DB2
neo4j
GRAPH
INFINITEGRAPH
Couchbase
DOCUMENT
STORE
mongoDB
cassandra
APACHE
HBASE



Source https://www.3pillarglobal.com/insights/exploring-the-different-types-of-nosql-databases


ORACLE
Coherence
KEY-VALUE
redis
IBM
DB2
neo4j
GRAPH
INFINITEGRAPH®
Couchbase
DOCUMENT
STORE
mongoDB®
cassandra
APACHE
HBASE

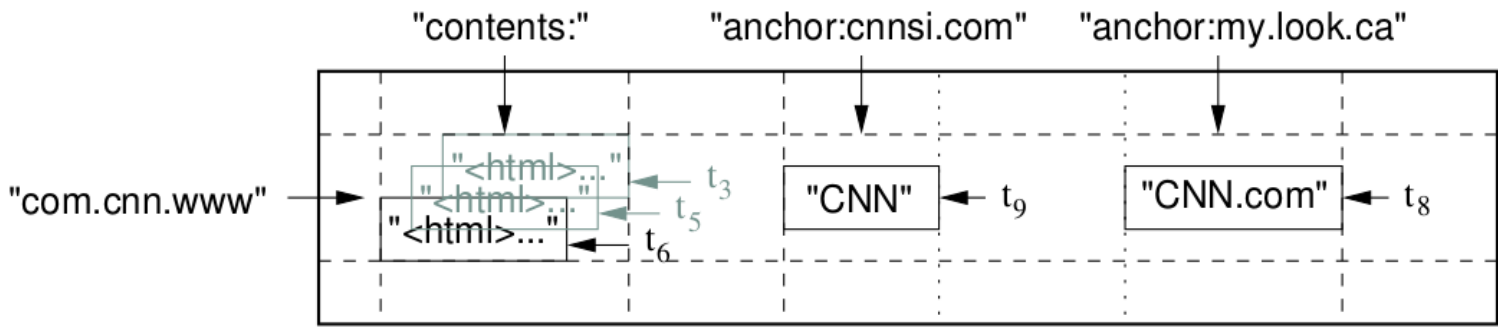
BigTable

Overview

- Multi-dimensional, sparse, sorted map with tabular structure
- Dynamic control of schema:
  - Data layout
  - Data format
- Schema choice determines data locality
- Therefore user choices significantly affect speed

Indexing

- Rows
  - Sorted alphabetically, keeps data together if using reverse naming
- Column-families
  - Need to provide column-family:column to retrieve data
  - Columns in family are compressed together so should be same type
  - Designed to have few families but many columns
- Timestamps
  - Default assignment real-time (microseconds)
  - Can be client assigned
  - Garbage collection can be set to automatically keep only n most recent

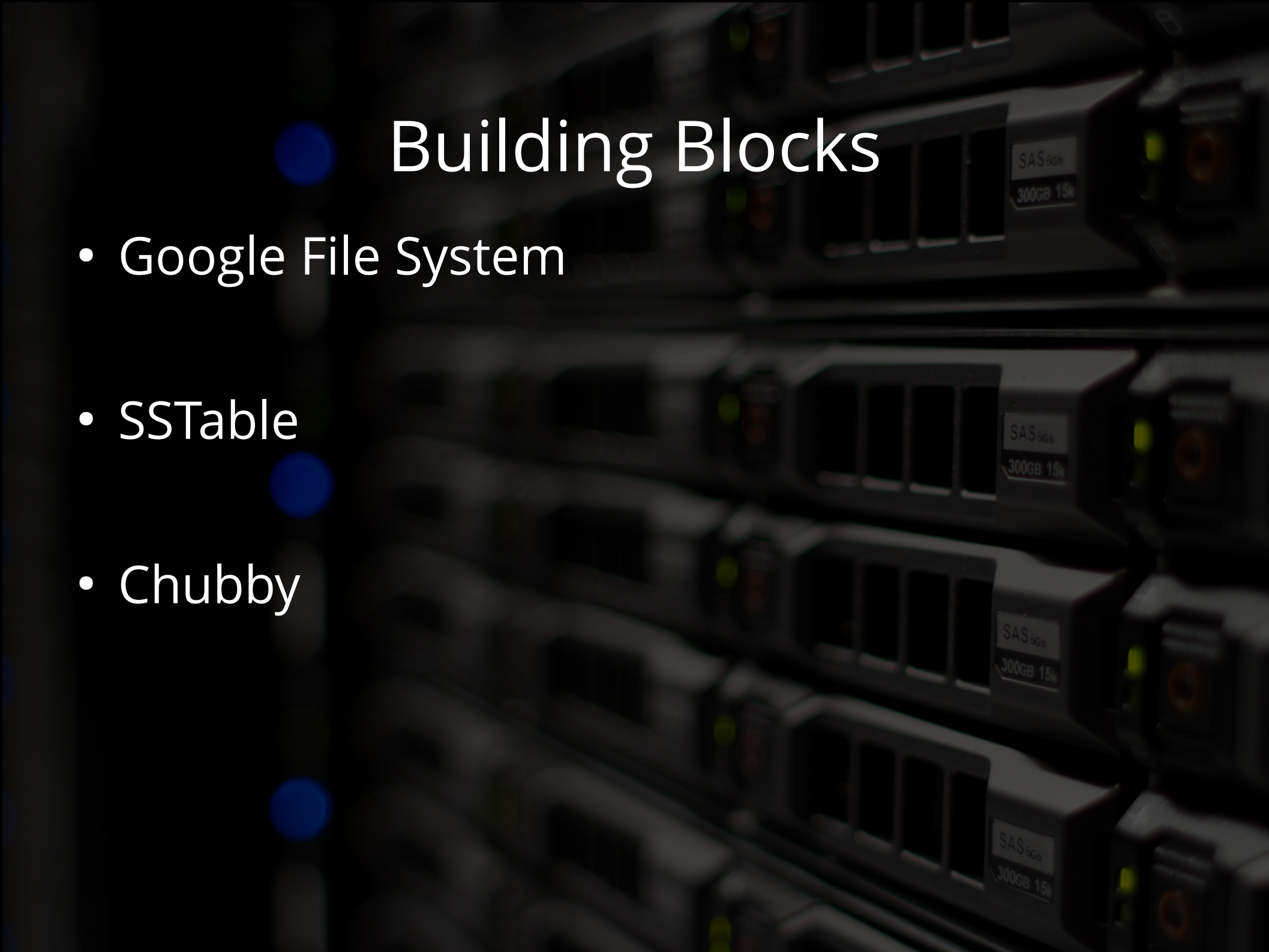

"contents:"
"anchor:cnnsi.com"
"anchor:my.look.ca"
"com.cnn.www"
"<html>..."
"<html>..."
"<html>..."
t3
t5
t6
"CNN"
t9
"CNN.com"
t8

Partitioning: Tablets

- Range of rows
- Unit of data distribution
- 100-200 MB each
- Automatic splitting

Building Blocks

- Google File System
- SSTable
- Chubby



Google File System

- Distributed data storage system
- Files divided into **fixed-size chunks**
- Chunks replicated 3 times across **chunk servers**
- Chunk location and metadata managed with **master server**

Sorted String Table - SSTable

- Key-value map datatype
- Immutable
- Sorted sequentially for easy lookup

Chubby

- Distributed lock service
- Determines lock status with **Paxos consensus algorithm**
- 5 chubby replicas
  - 1 of which is master
- Used for:
  - Ensure there is only 1 active master
  - Store location of root tablet
  - Discover and delete tablet servers
  - Store column family information
  - Store access control lists


Implementation

General Structure

Tablet servers

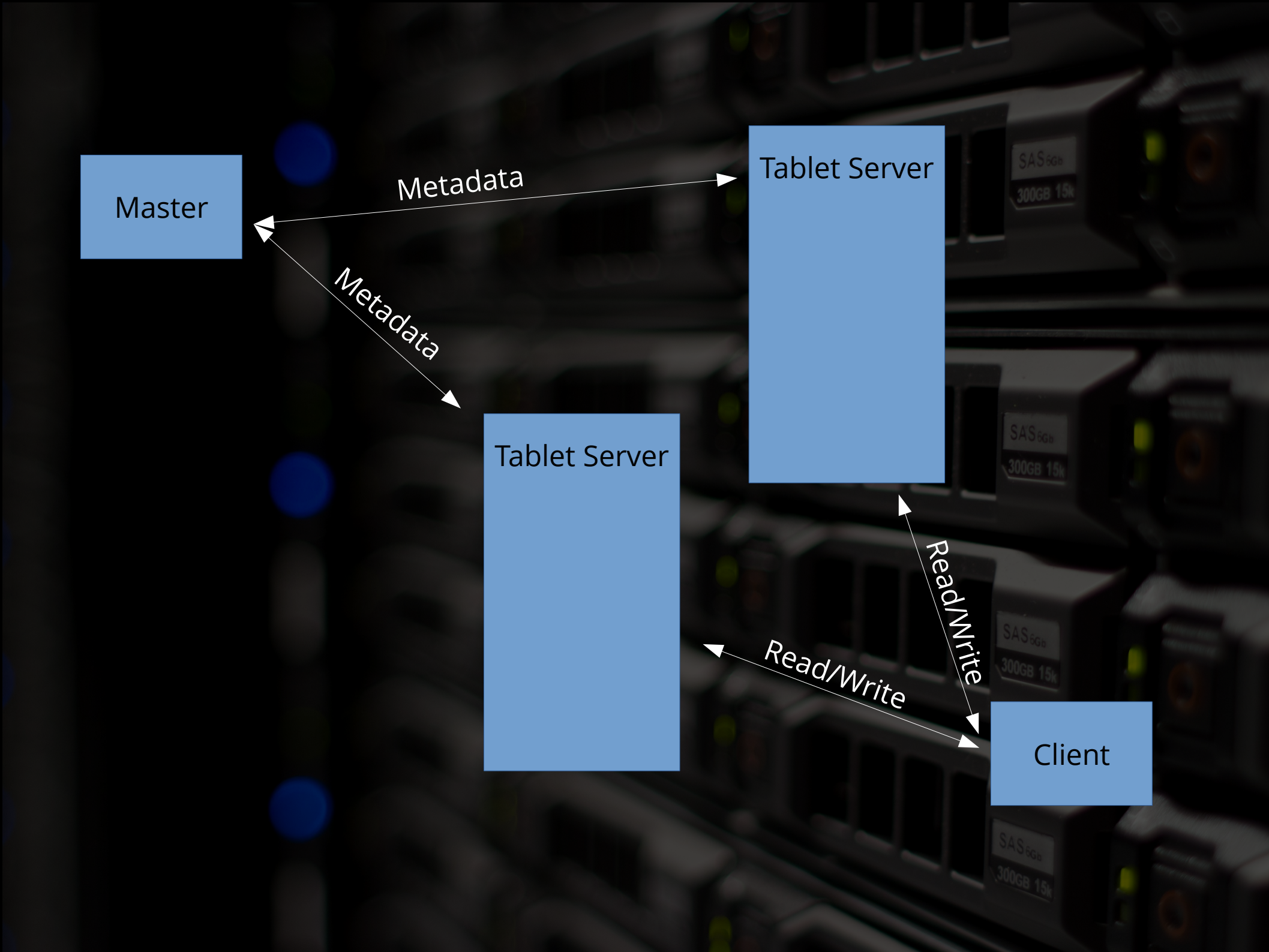
- Manages set of tablets (10-1000 tablets)
- Manages read/write directly with client
- Splits tablets when too big

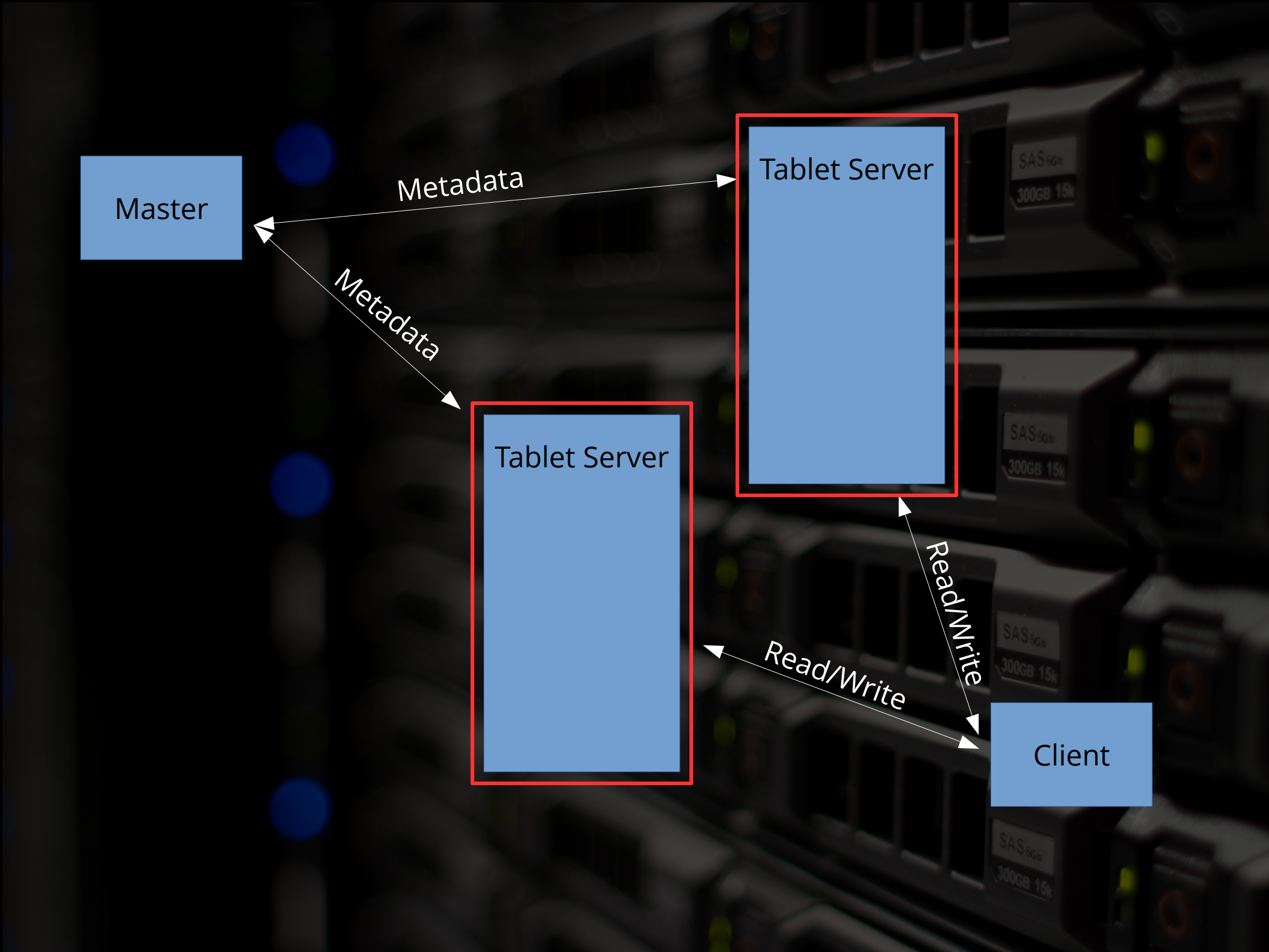
Master server

- Assigns tablets to tablet servers
- Managing tablet servers and load
- Garbage collection

Client

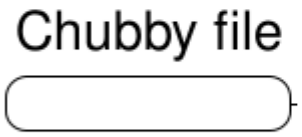
- Can iterate over: column families, columns within a family, rows


Master
Metadata
Metadata
Tablet Server
Tablet Server
Read/Write
Read/Write
Client


Master
Metadata
Metadata
Tablet Server
Tablet Server
Read/Write
Read/Write
Client

Tablet Servers

- Tablet assignment managed by master server
  - Acquires exclusive chubby lock in servers directory


Chubby file

Tablet Servers

- Tablet assignment managed by master server
  - Acquires exclusive chubby lock in servers directory
- Tablet location storing
  - Chubby File
  - Root tablet
  - Metadata tablets
  - Tables containing data tablets

Tablet Serving

Tablet Storage

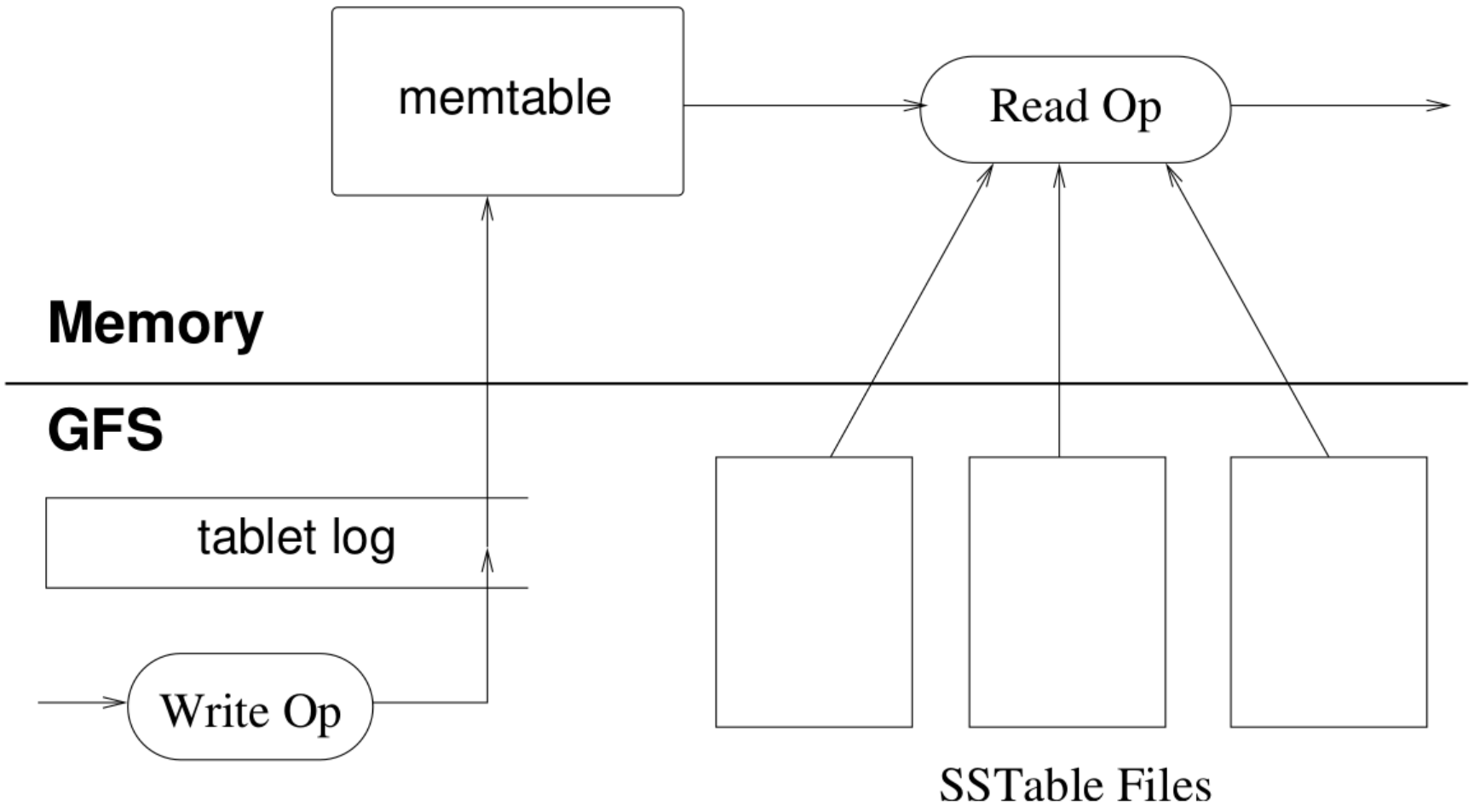
- Recent commits logged in *memtable*
- Old commits logged in SSTables

Tablet Read

- SSTable locations from metadata
- Constructs tablet by merging SSTables and memtable using commits

Tablet Write

- Commited to log
- Contents inserted into memtable


memtable
Read Op
Memory
GFS
tablet log
Write Op
SSTable Files

Compaction

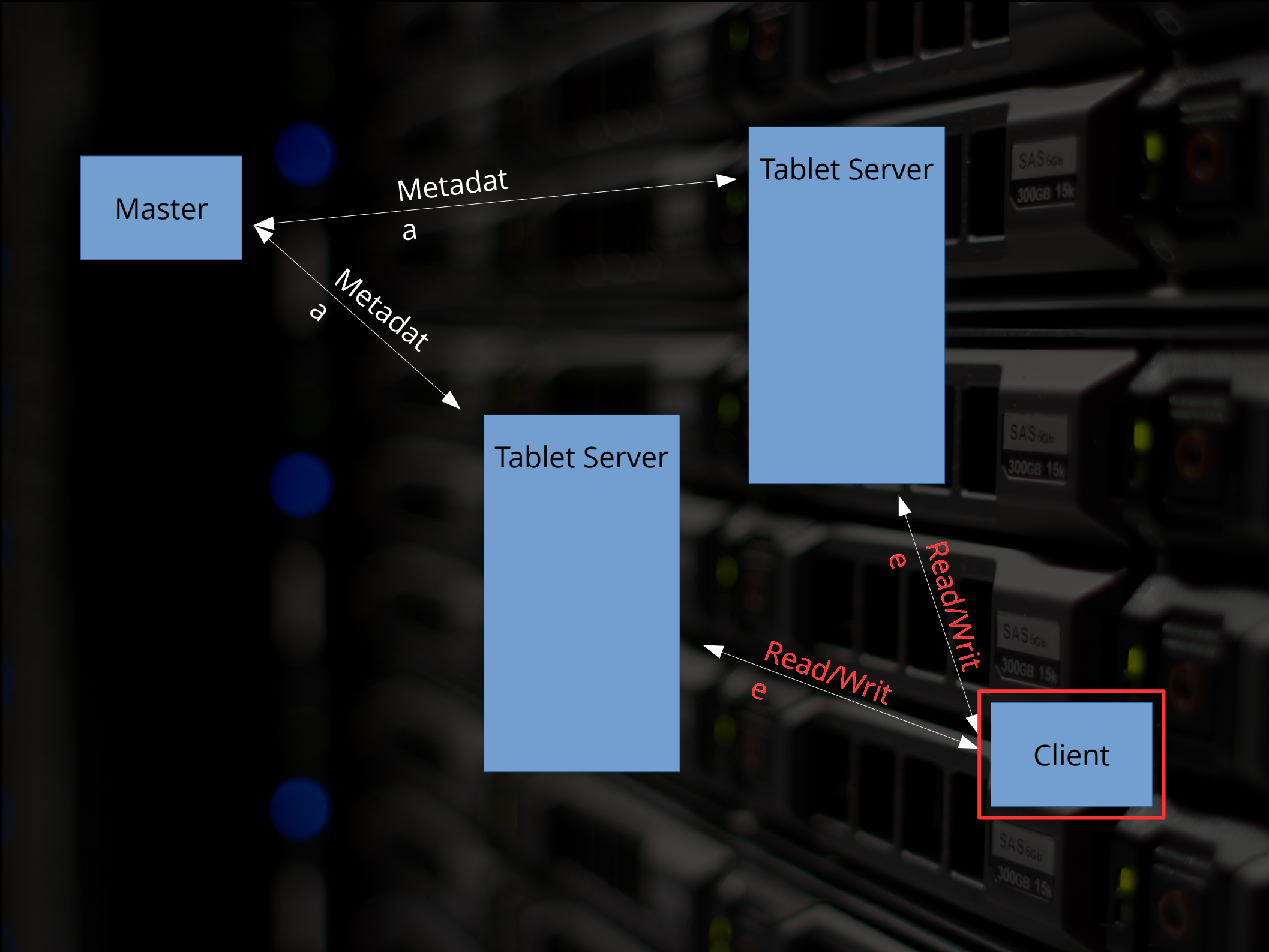
Problem: memtable fills up with commits

Minor compaction

- Memtable made into new SSTable
- New memtable created

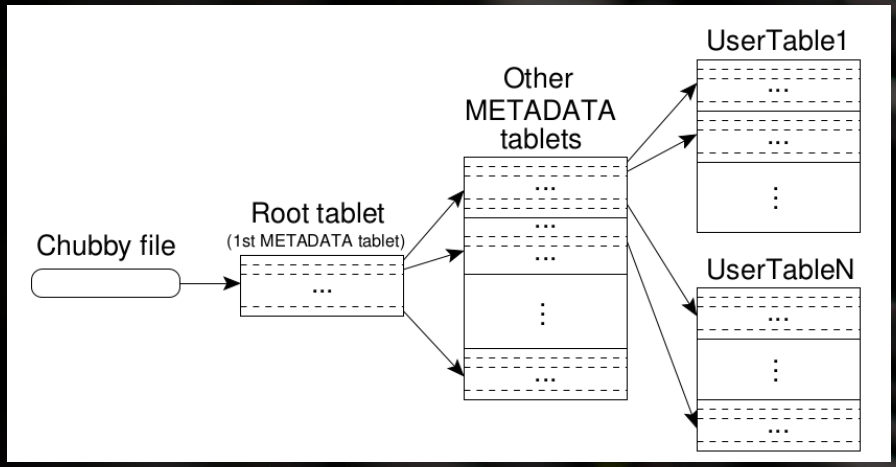
Major compaction

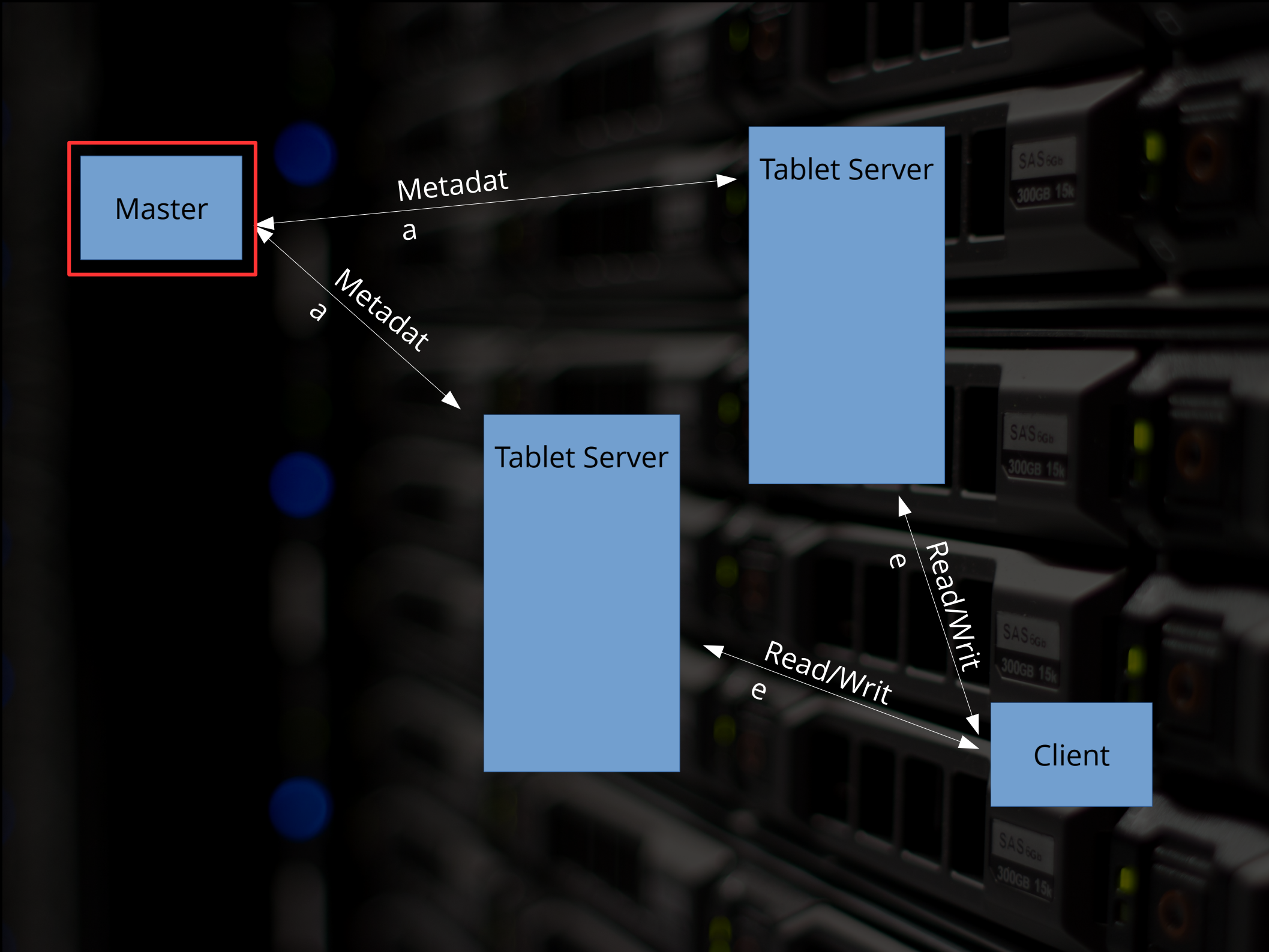
- Combines memtable and SSTables with no deletions


Master
Metadat
a
Metadat
a
Tablet Server
Tablet Server
Read/Writ
e
Read/Writ
e
Client

Client Features

- Caches tablet locations
- If location inaccurate, moves up tablet server hierarchy
- Atomic single-row transactions




Master
Metadat
a
Metadat
a
Tablet Server
Tablet Server
e
Read/Writ
Read/Writ
e
Client

Master Server

Jobs

1) Managing tablet servers

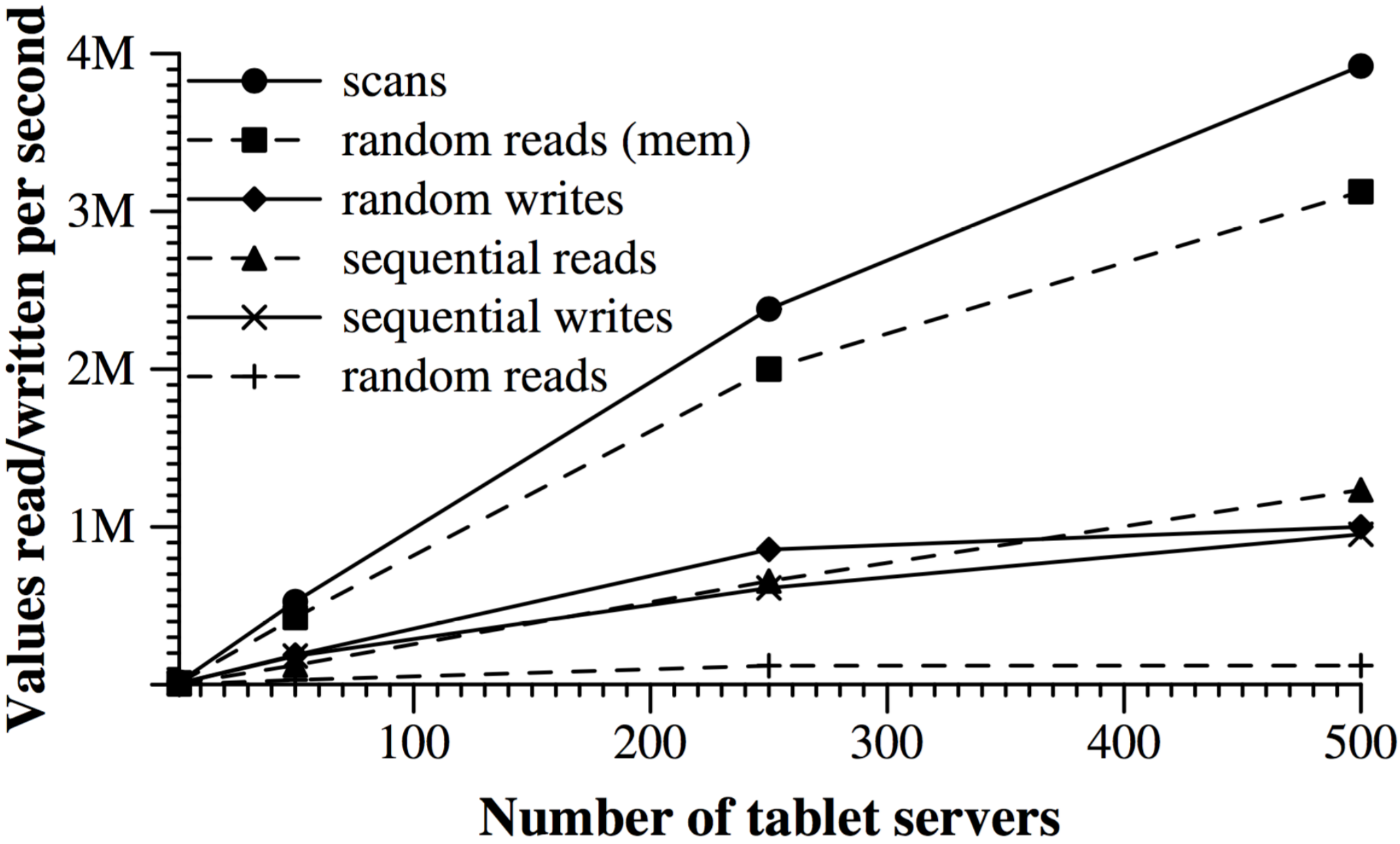
2) Assigning tablets

3) Garbage collection

Experiments and Examples

Benchmarks

- Sequential write
- Sequential read
- Scan
- Random reads
- Random write


Values read/written per second
4M
3M
2M
1M
scans
random reads (mem)
random writes
sequential reads
sequential writes
random reads
100
200
300
400
500
Number of tablet servers

Future directions

- Apache Cassandra, Hbase
- Google Cloud
- Google Spanner