系统设计
Distributed System Design 2

本节主讲人：北丐

扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: ninechapter
微博: http://www.weibo.com/ninechapter
知乎: http://zhuanlan.zhihu.com/jiuzhang
官网: http://www.jiuzhang.com

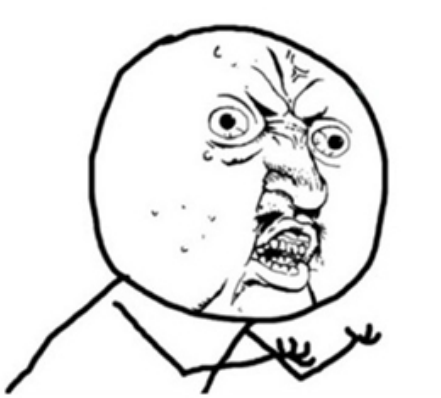
今日课程大纲

- Design a Bigtable
  - Google, Facebook, Amazon, Alibaba
  - NoSQL database 设计框架和原理
  - SStable 读和写
  - 如何建 Index
  - Bloom Filter

- Map Reduce Problems
  - Google, LinkedIn, Apple
  - 多台机器并行处理数据
  - Count Word Frequency
  - Build Inverted Index

Bigtable

Interviewer: What is bigtable?



What is bigtable?

NoSQL DataBase	Company
Bigtable	Google
Hbase	Open Source of Bigtable
Cassandra	Facebook

为什么我们要讲bigtable 的实现?

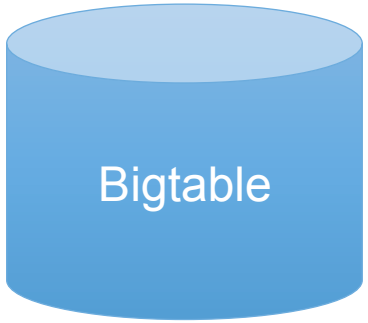
1. Google面试题
2. 解决相类似系统设计题,比如:Look up service
3. 追问NoSQL How to scale的原理

回顾第二节课

bigtable的逻辑实现

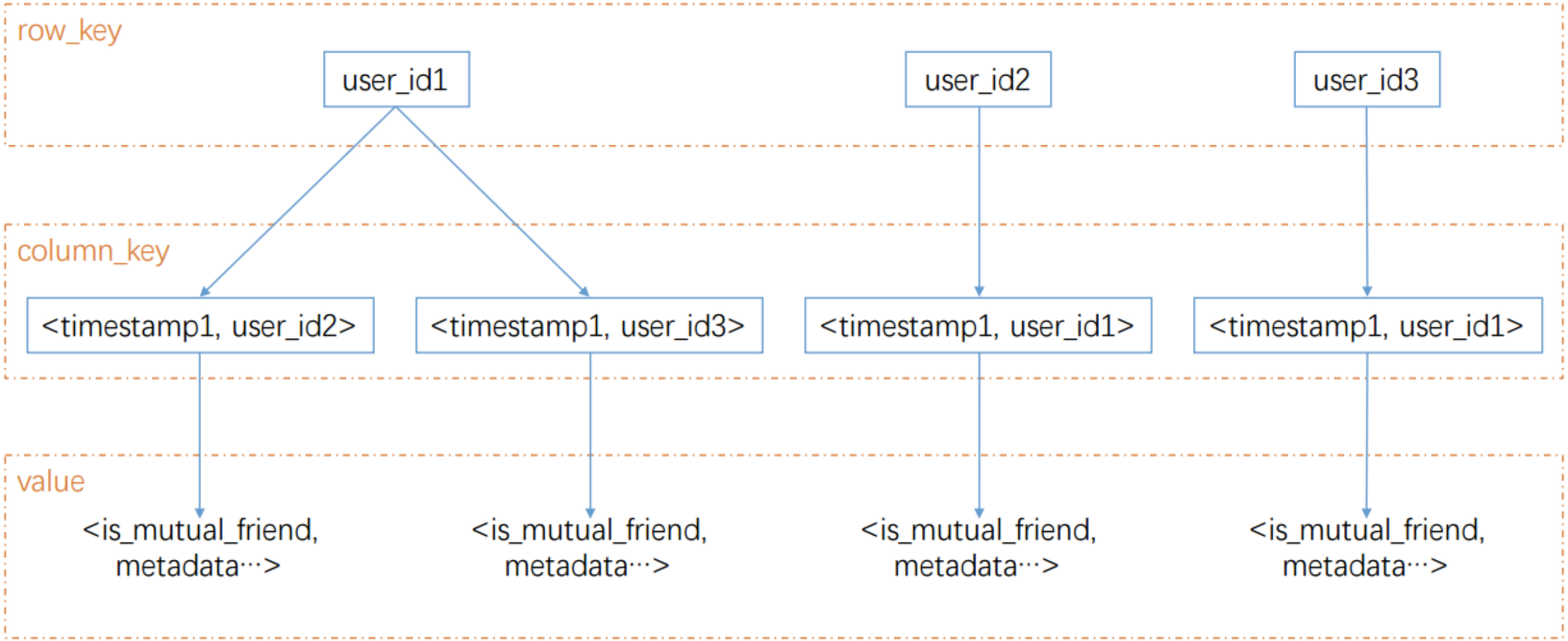
Bigtable

Bigtable	ColumnKey1	ColumnKey2
Row_Key1	value1	value2
Row_Key2	value3	value4



- 三层结构的NoSQL（bigtable）
- 第一层： Row Key
  - 通常是Hash Key
  - 比如Row key = UserId
- 第二层：Column Key
  - 是排序的，可以进行range query
  - 复合值
  - 比如 timestamp+user id
- 第三层：value
  - 一般是一个string
  - 存储对应的信息

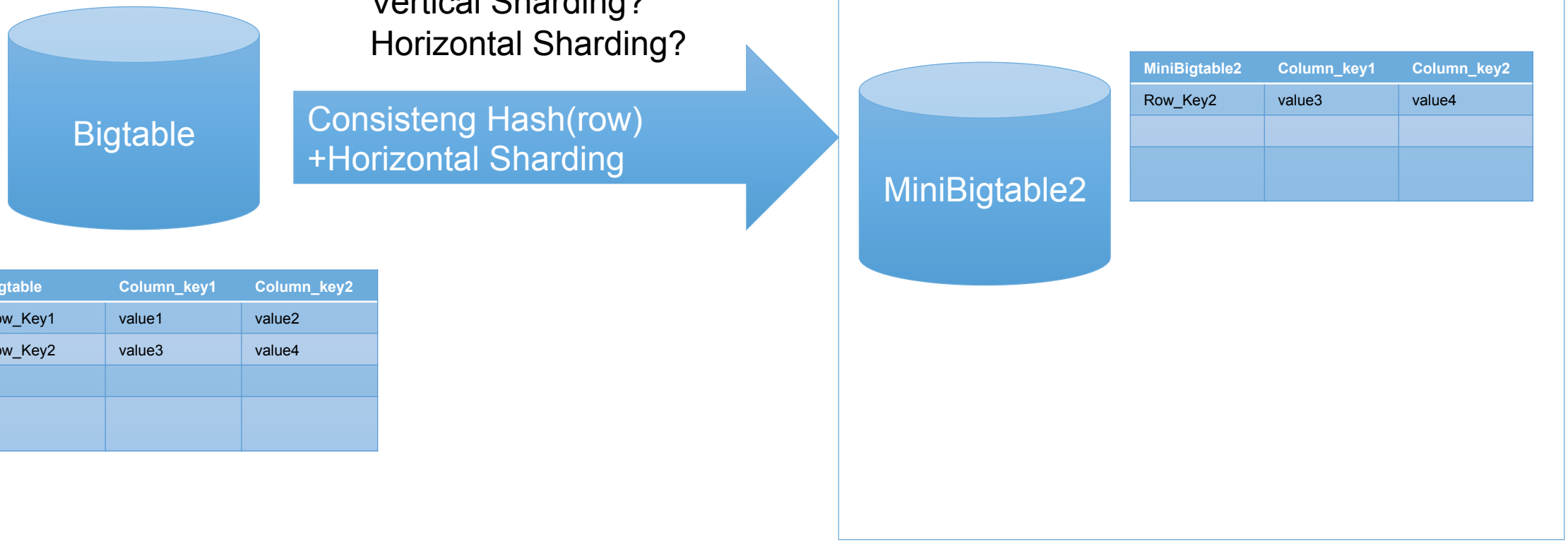
看看Friend Ship Table 的存储



Interviewer: How to Scale?

NoSQL Scale 的原理

Interviewer: How to Scale?



Bigtable	Column_key1	Column_key2
Row_Key1	value1	value2
Row_Key2	value3	value4

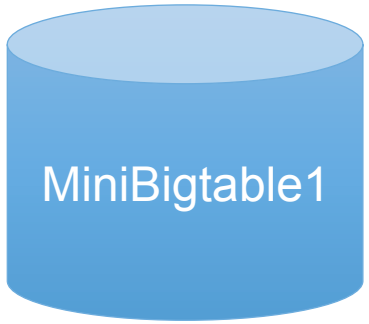
MiniBigtable1	Column_key1	Column_key2
Row_Key1	value1	value2

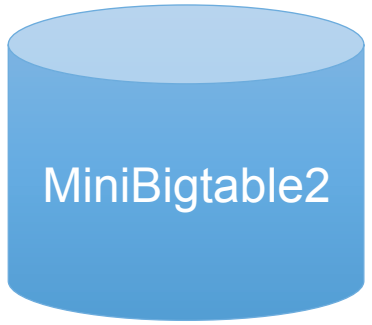
MiniBigtable2	Column_key1	Column_key2
Row_Key2	value3	value4

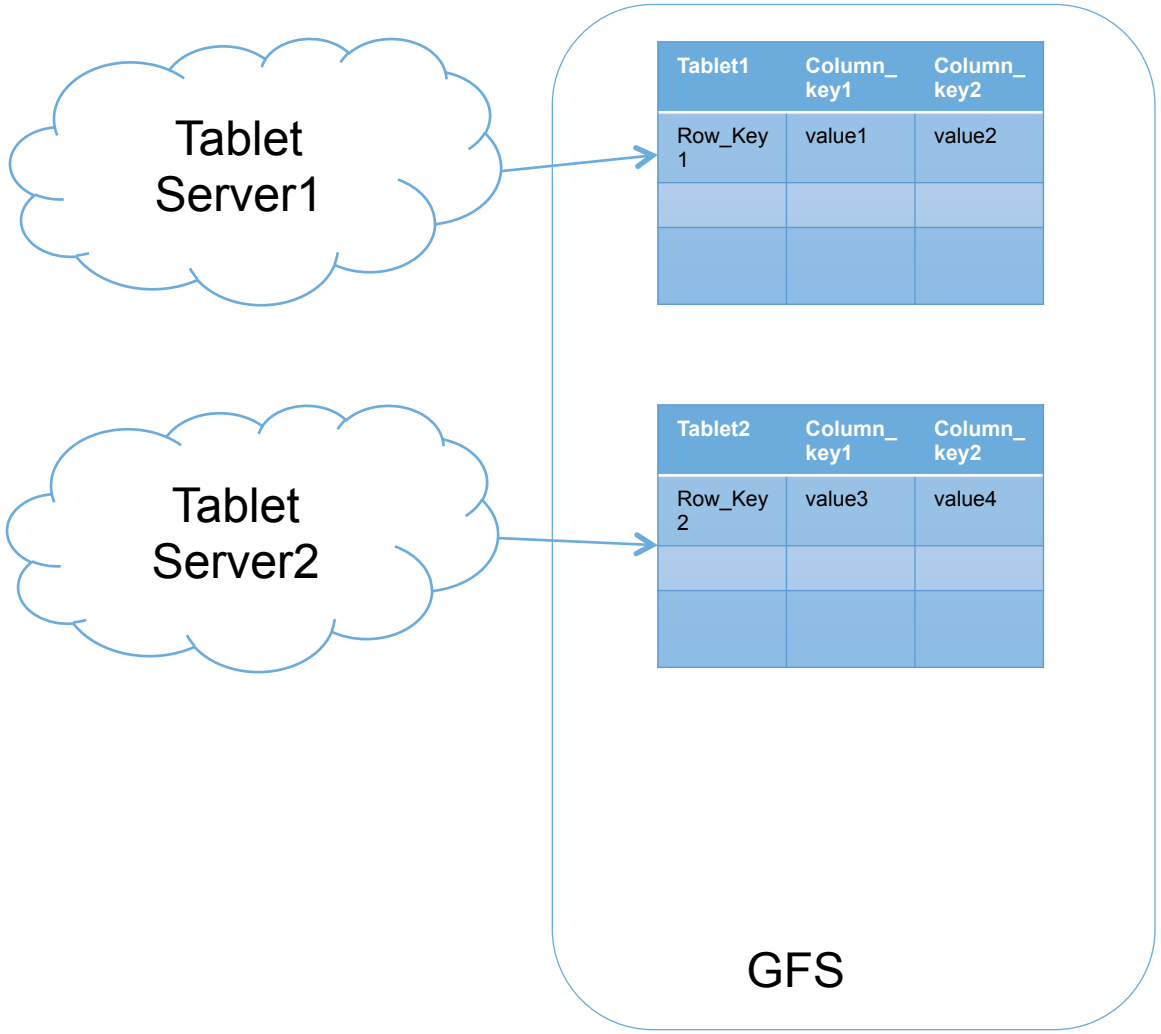
Tablet Server

Key
- Tablet = Minibigtable
- Tablet 存在Tablet server上面。
- 一个Tablet Server上面就是一个小的DB

- Question?
  - How to manage Tablet Server?



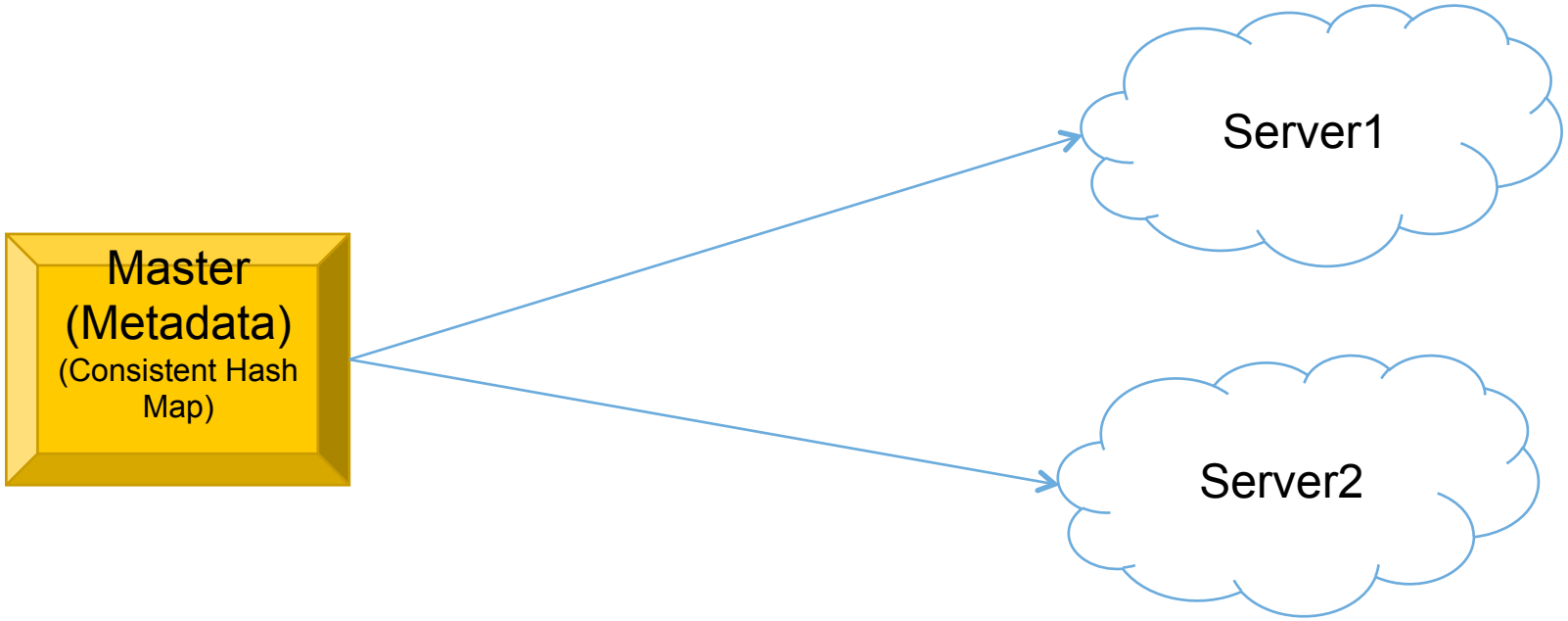




Interviewer: How to manage Tablet Server?

Interviewer: How to manager Tablet Server?

- Key
  - Master + Slave
  - Master has HashMap[key, server address]



Interviewer: How do we read/ write in bigtable?

后端系统设计日经题

Interviewer: How do we read/write in bigtable?

- Read: Key; Return: Value
- Write: Key,Value; Return: Done
- What is key?
- Bigtable Is NoSQL (Key-Value Store)
- Key= Row_Key+Column_key.

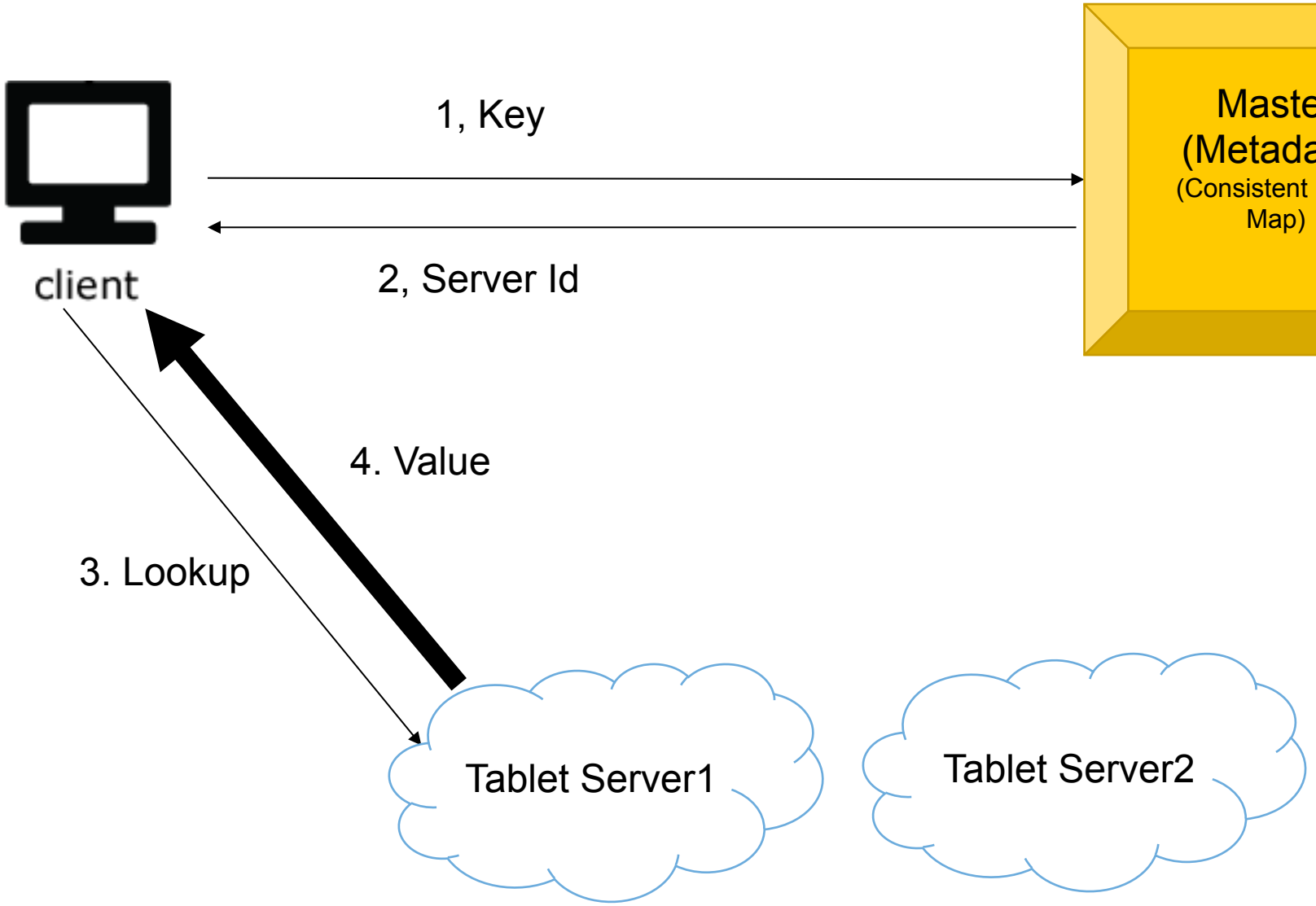
Bigtable	ColumnKey1
Row_Key1	value1
Row_Key2	value3

逻辑结构

Key:value
(Row_Key1+ColumnKey1):value1
(Row_Key2+ColumnKey2):value3

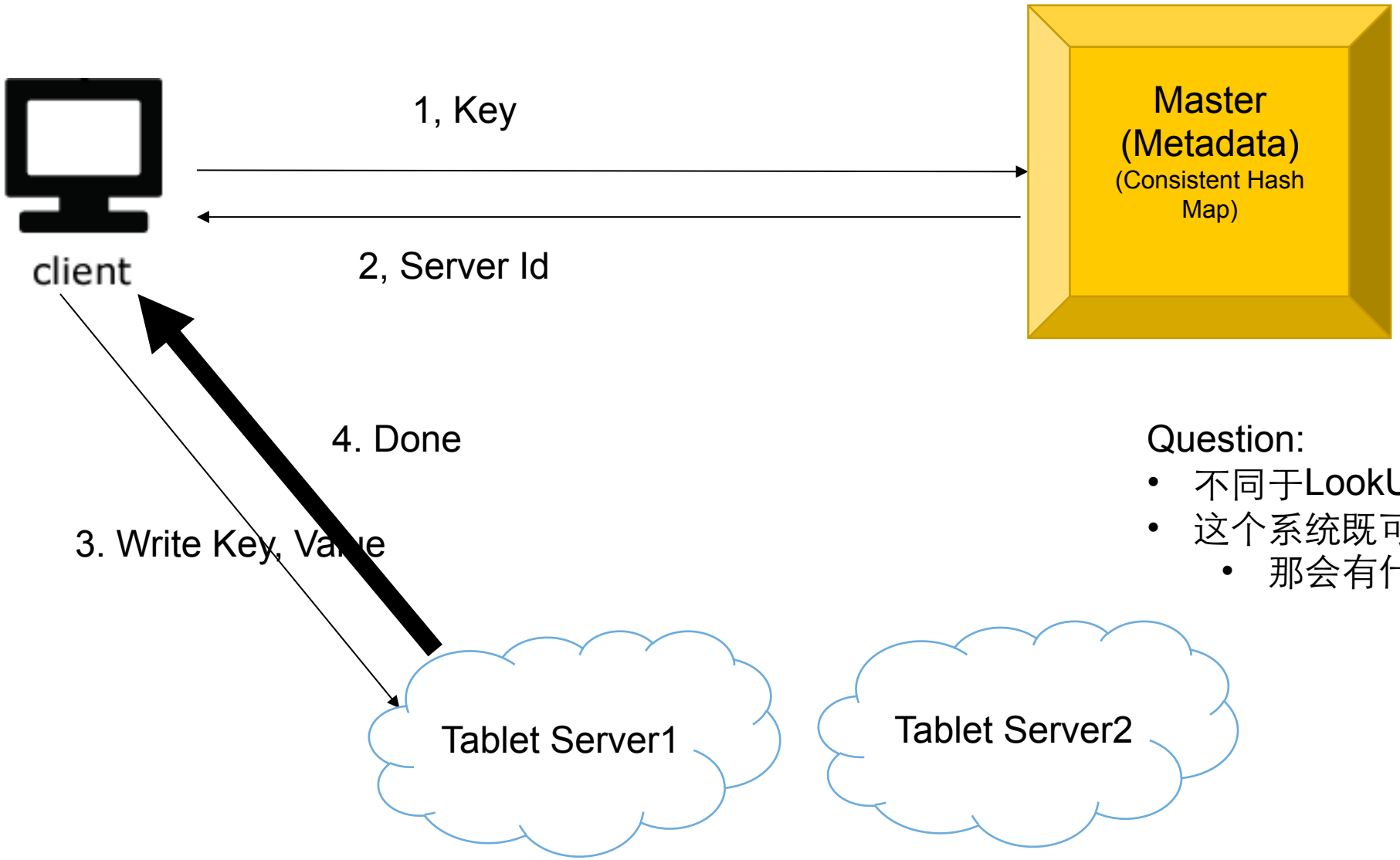
真实文件里面存储

Interview: How to read a key? （宏观）



Client是相对的:
Browser vs Webserver
Webserver vs Database
Database vs GFS

Interview: How to write a key?



Question:

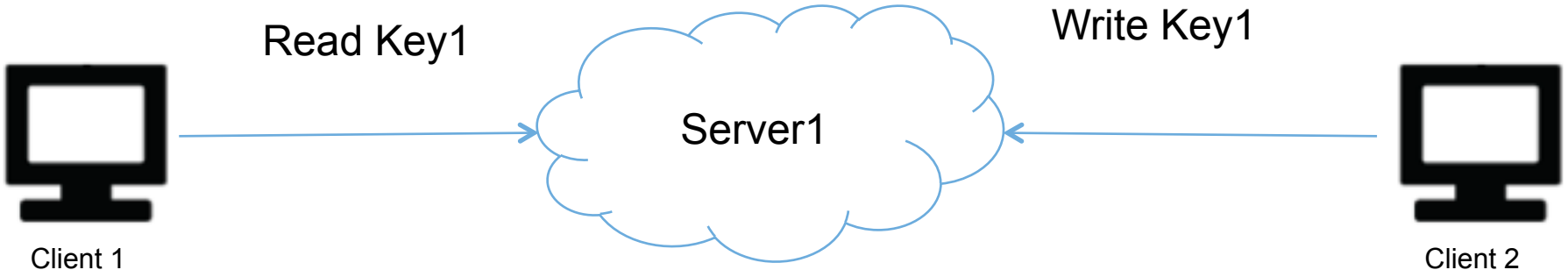
- 不同于LookUp Service
- 这个系统既可以读又可以写?
  - 那会有什么样的问题?

Interviewer: What if we read while we are writing?

写的过程当中，有读请求?

Race Condition

Interviewer: What if we read and write the same key at same time?



Race Condition

- http://bit.ly/25FBHM4

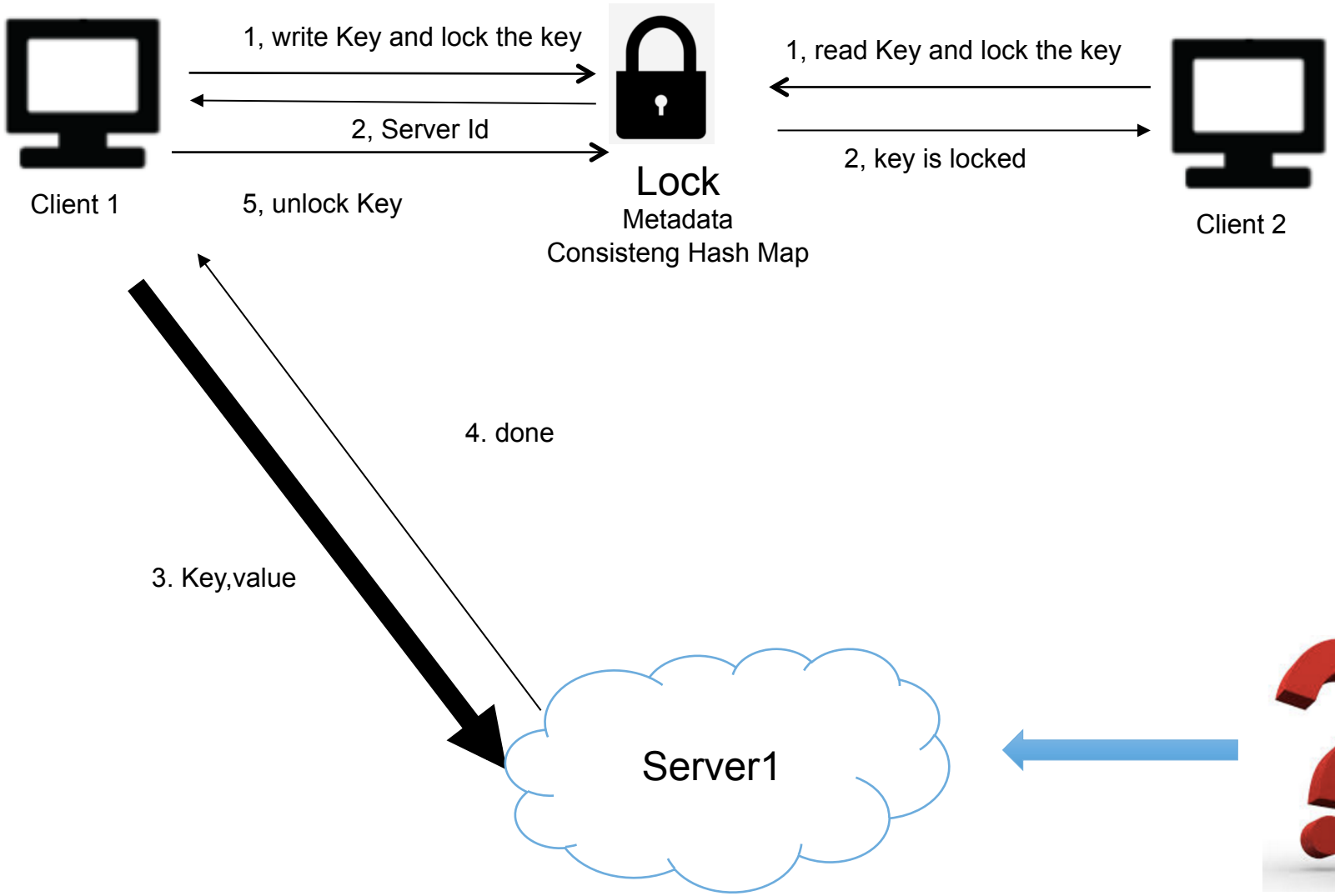
We need a lock

We need a distributed lock

- Chubby
- Zookeeper

- Read More:
- http://bit.ly/1Pukiyt
- http://bit.ly/1TOWIsR

Interview: How to write a key?



Distributed Lock

- The Metadata is store on the Lock
- Lock 本来要存储Metadata那master就不需要存储MetaData了

Summary of Bigtable High Level Overview

- Design
  - Client + Master + Tablet Server + Distributed Lock
- Client
  - Read + Write
- Tablet Server
  - Maintain the Data (Key value pairs)
- Master
  - Shard the file
  - Manage the servers health
  - Load Balance
- Distributed Lock
  - Update MetaData
  - Maintain the MetaData
  - Lock Key

Question?

- How to read and write in Tablet Server in detail?

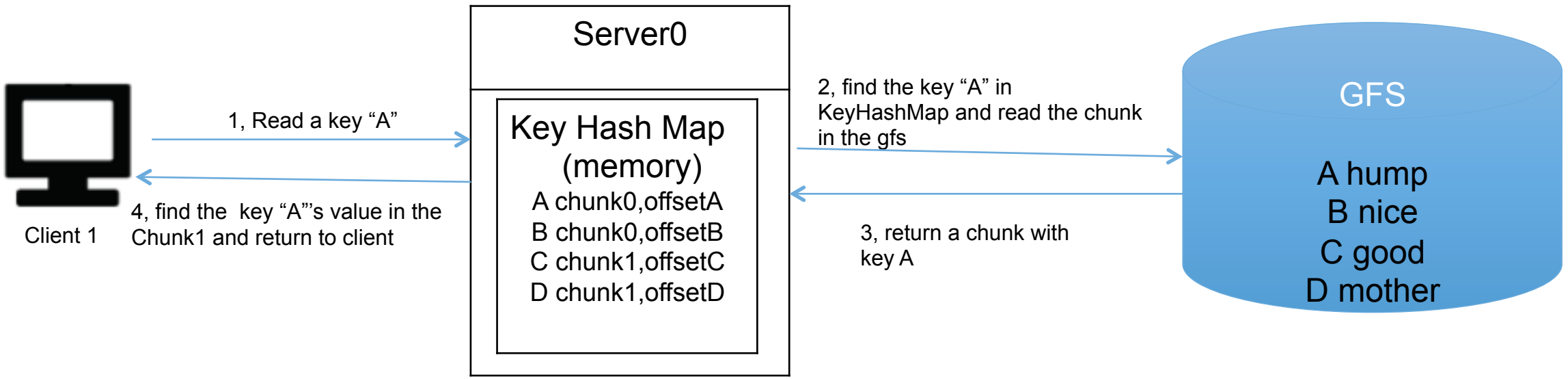
Interviewer: How do we write/ read in tablet server in detail?

回顾Look Up Service

如果只有读的过程，没有写的过程

client怎么和server沟通去lookup一个key返回对应value

回顾Look Up Service



Why we need to read a chunk not just a key?

- Server will cache the chunk, we don’t need disk operation for every read
- GFS support read a chunk but not one byte

Tablet Server?

We need to read Key:A and Return: Apple. (Similar to Lookup Service).
We need to Write Key:A, Value:Apple. Return: Done

Question?

If we have write operation, how do we write ?

For the write, we have two ways

Question which one is better?

1. Modify the key in the GFS directly.
2. Put it in the memory temporarily and modify it later.

Put it in the memory temporarily and modify it later.

Question:

- What if memory is not enough?
  - What if Server crashes?

Put it in the memory temporarily and modify it later.

- What if memory is not enough?
  - Short Answer: Store them in disk when memory is not enough.
- What if Server crashes?
  - Short Answer:写log. write ahead log.

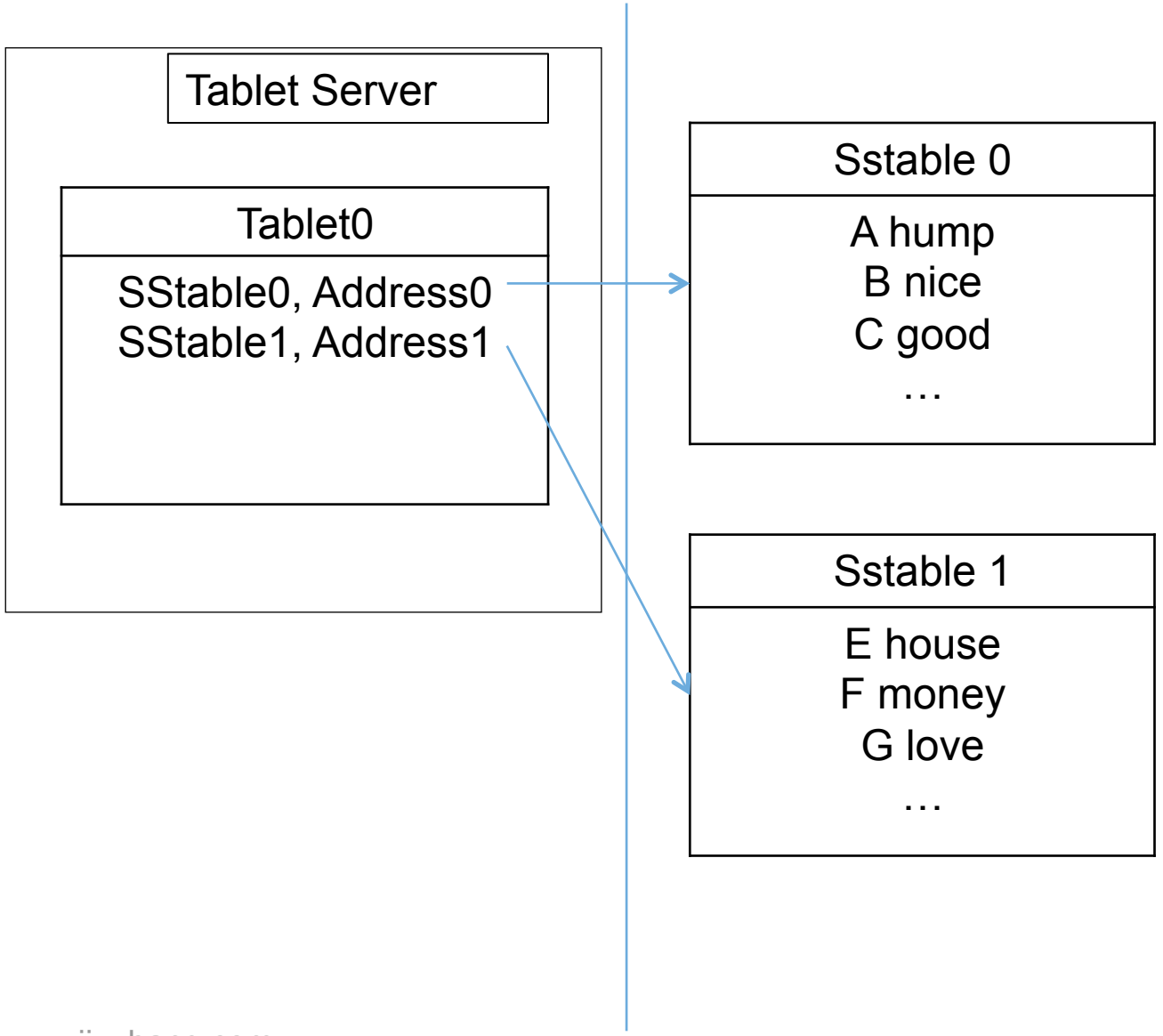
Question:
Could you explain Tablet Server Write Process in detail?

讲怎么写之前，需要知道 Tablet Server里面怎么存储的

SStable

Sorted Strings Table

SStable



Key point

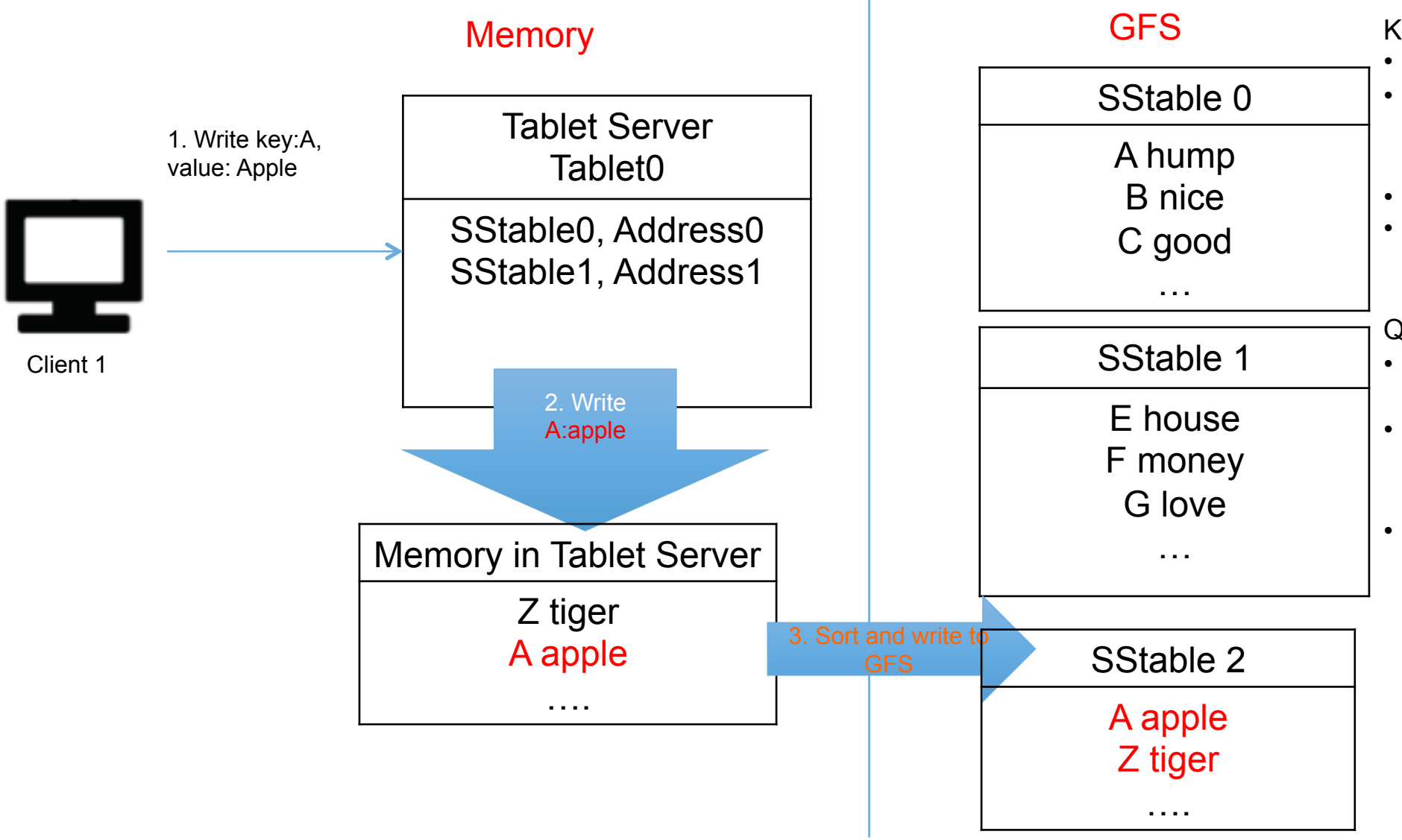
- SStable
- A table = a list of tablets
- A tablet = a list of SSTables
- A SSTable = a list of sorted <key, value>
- Sstable = Sorted Strings Table

Question?

- SStable为什么要用排好序?
  - Answer: 方便查找
- 一个Tablet怎么生成Sstable的?

Question:
一个Tablet怎么生成Sstable的?

一个Tablet 怎么生成Sstable



Key:

- Key先写到内存里面
- 当内存达到一个值，把内存里面的数据按照key排序后，以sstable形式写入GFS
- Sstable 是在GFS里面
- Tablet 上面存Sstable的地址

Question:

- 为什么要把Sstable存GFS不存在local disk上面?
- Answer:
  - Replica
  - Failture and Recovery
- 读的时候sstable应该去哪一个读?

读的时候应该去哪个SStable读?

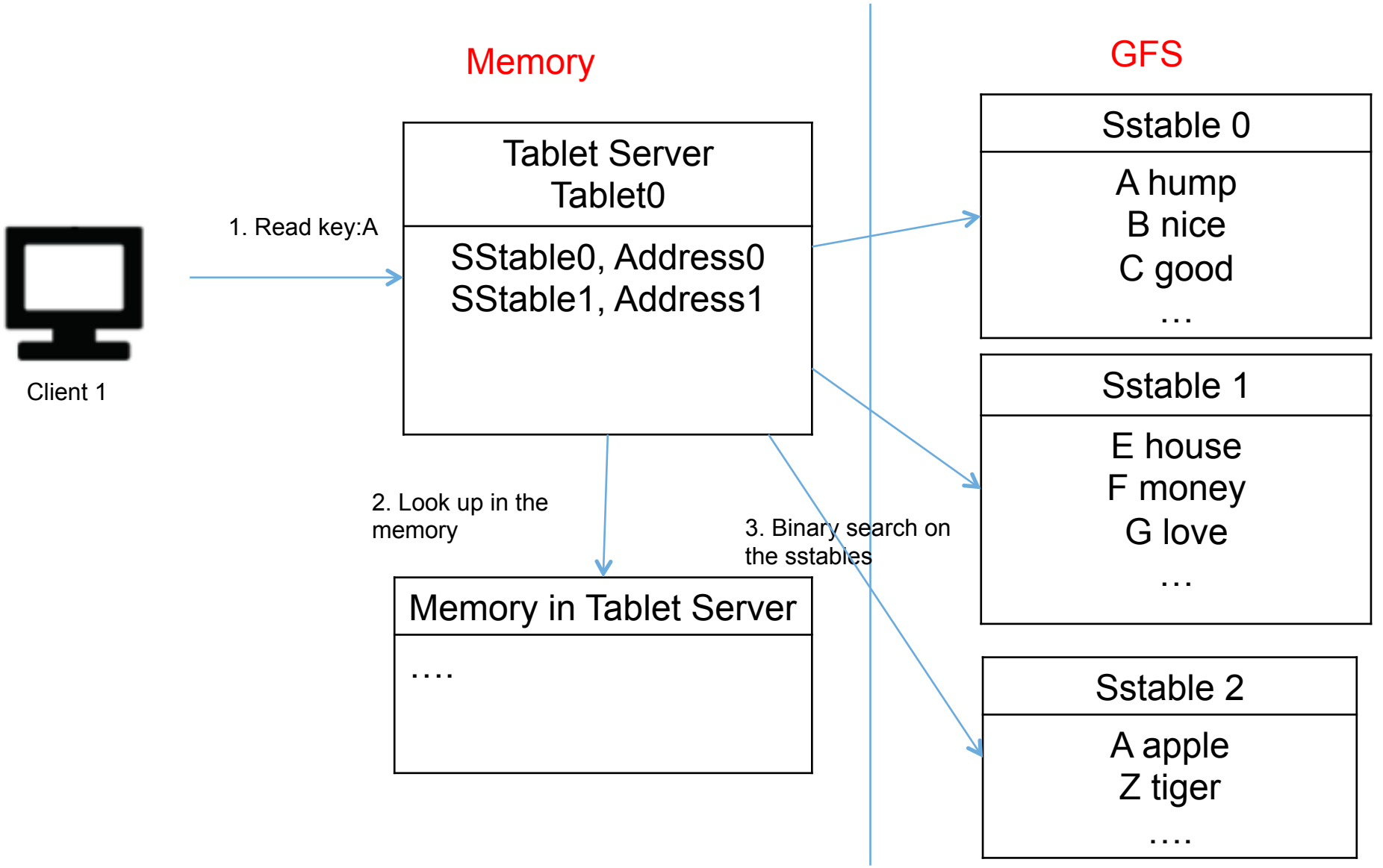
SStable 0 和 SStable2 都有相同的key

读一部分? Vs 全部都读?

SStable 0
A hump B nice C good …

SStable 2
A apple Z tiger ….

读的时候应该去哪个Sstable读?



Key:

- 先在memory里面查找
- 按照时间顺序倒着把sstable都会读一遍
- 最近的sstable最先查询
- 读sstable的时候用binary search查找key

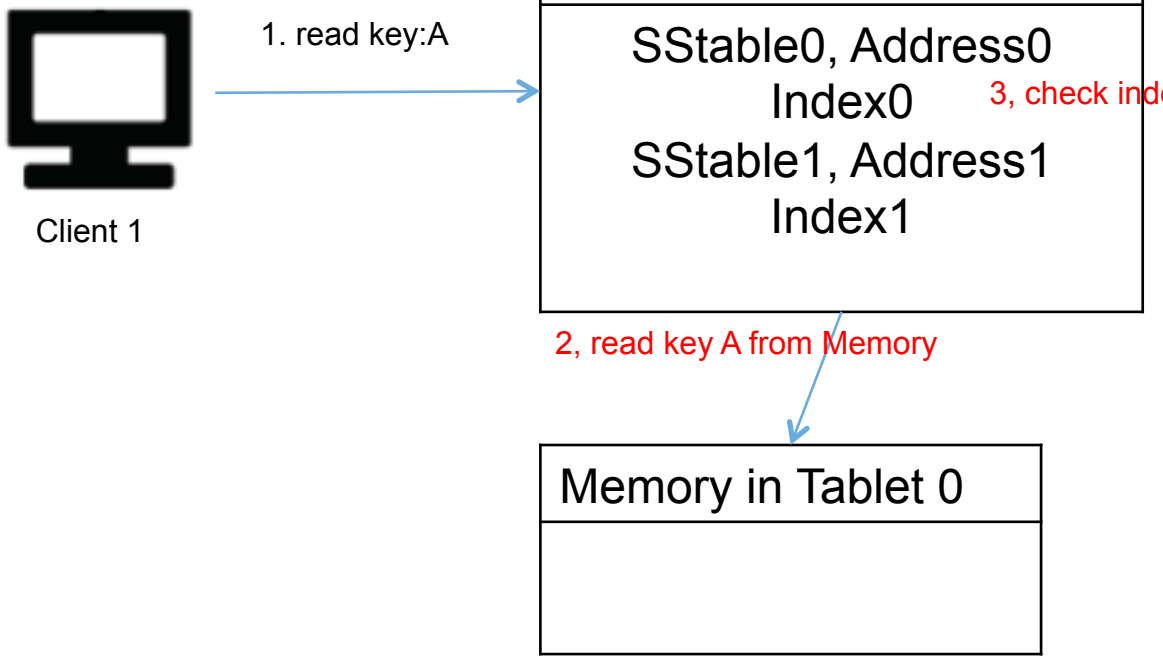
Question:

- 单个sstable每次都二分查找是不是特别慢?

Interview: How to read from sstable faster?

Build Index

Interview: How to read from a Tablet Server

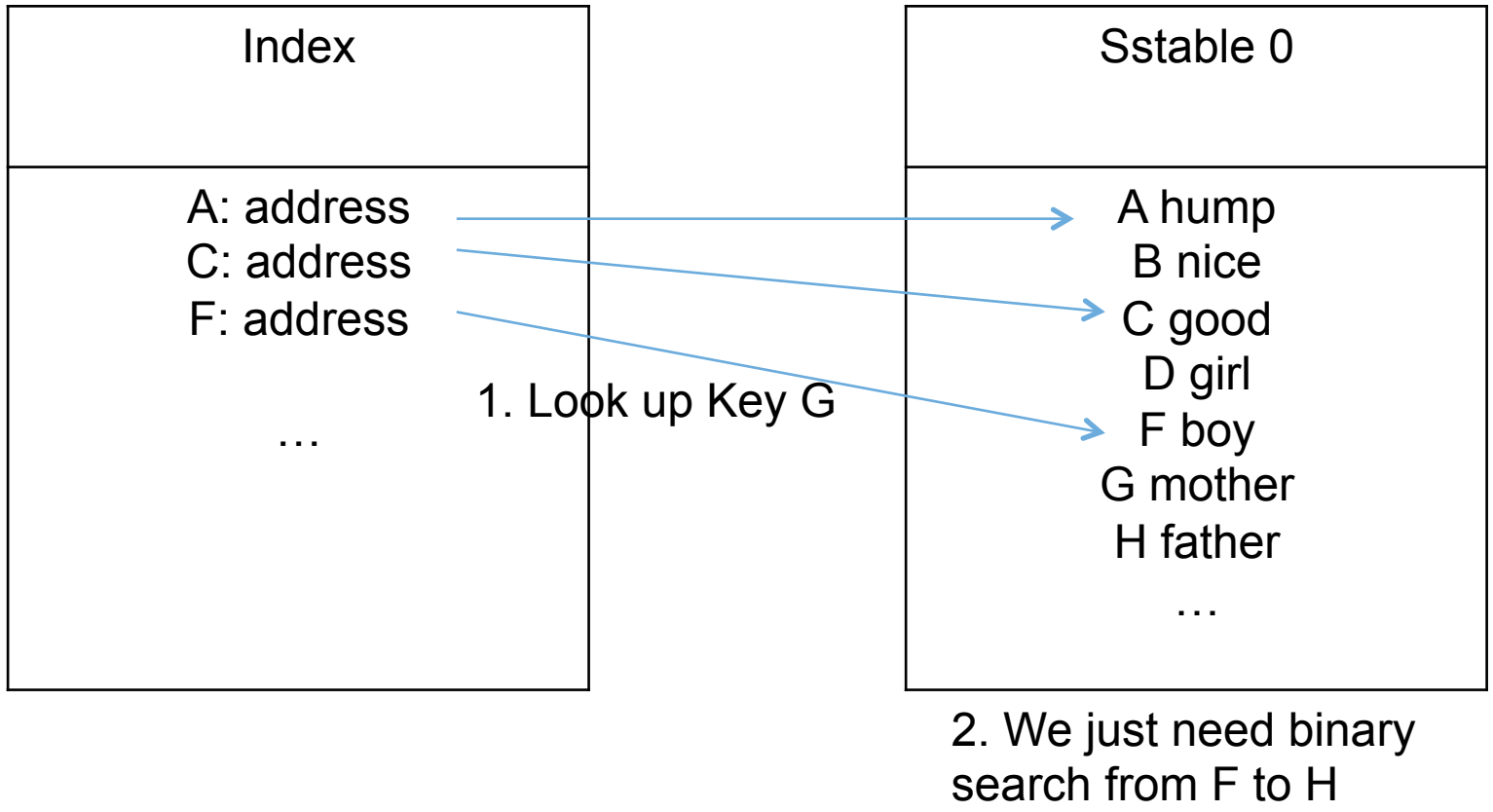


- Key
  - Build index for each sstable

Interview: How to build index?

One easy way to build index

One easy way to build index



Key

- 把一些Key放入内存作为Index
- Index有效减少磁盘读写次数
- Look up Key G，We just need binary search from F to H

Question?

- Why not store all key?
  - Memory is not enough

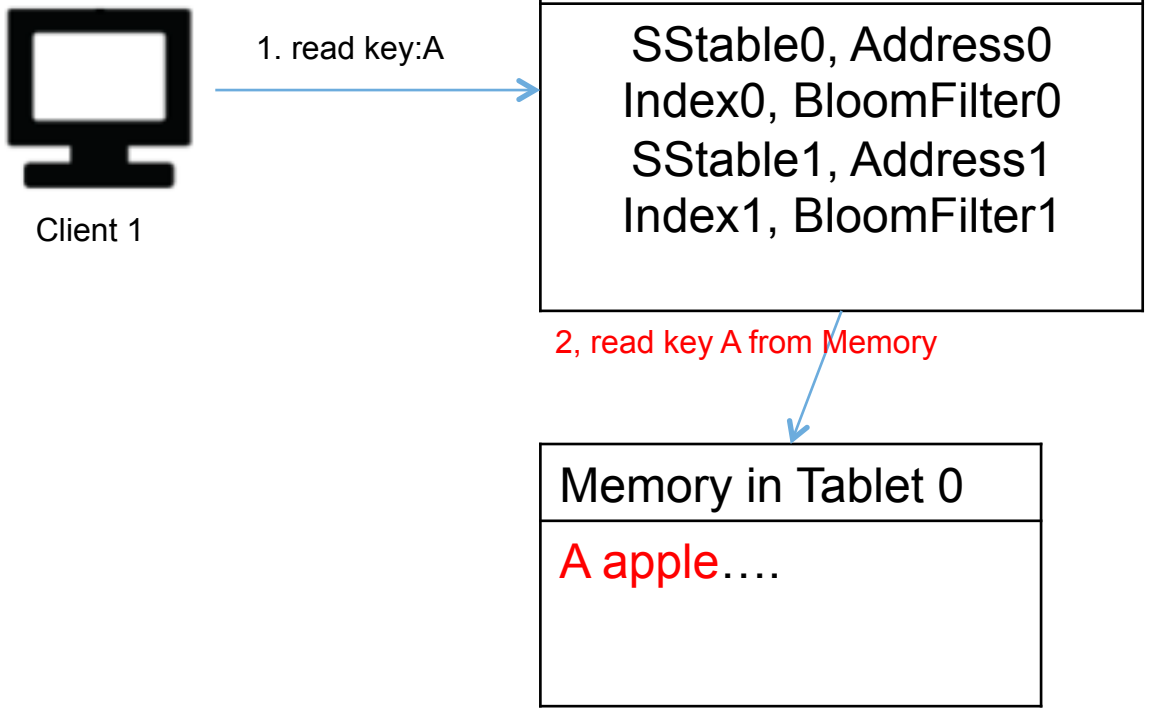
Intersection of Two Arrays ii Follow Up

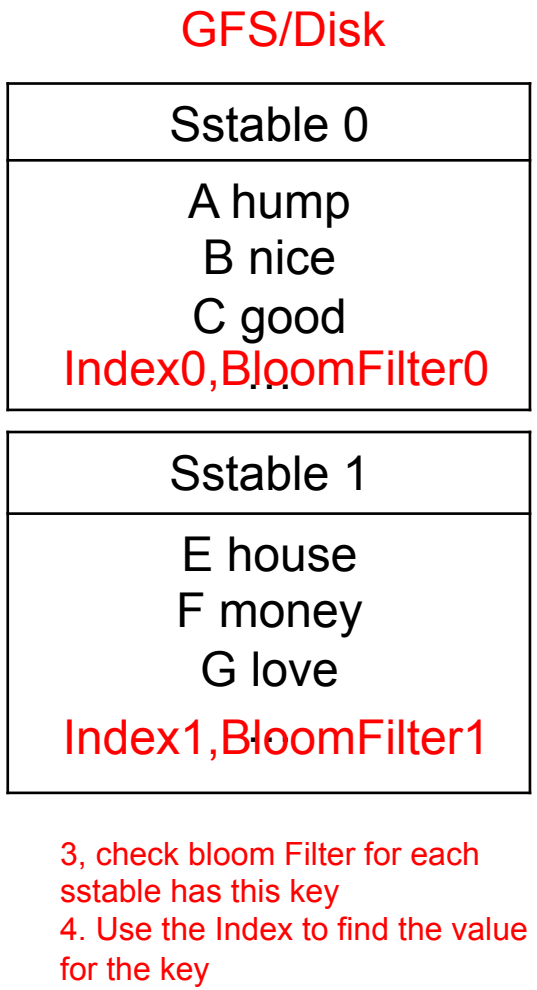
http://www.lintcode.com/en/problem/intersection-of-two-arrays-ii/

这其实可以拓展成一道算法题

Interview: How to read from sstable even **faster**?

Interview: How to read from a Tablet Server





Key

- Build bloom filter for each sstable
- Bloom filter 可以检查key在不在sstable里面

Interview: How to build bloom filter?

Interview: How to build bloom filter?

1. 初始化话一个bool数组全为0=false, 1=true

值	0	0	0	0	0	0	0	0	0	0
下标	0	1	2	3	4	5	6	7	8	9

2. 把所有的key填入数组里面

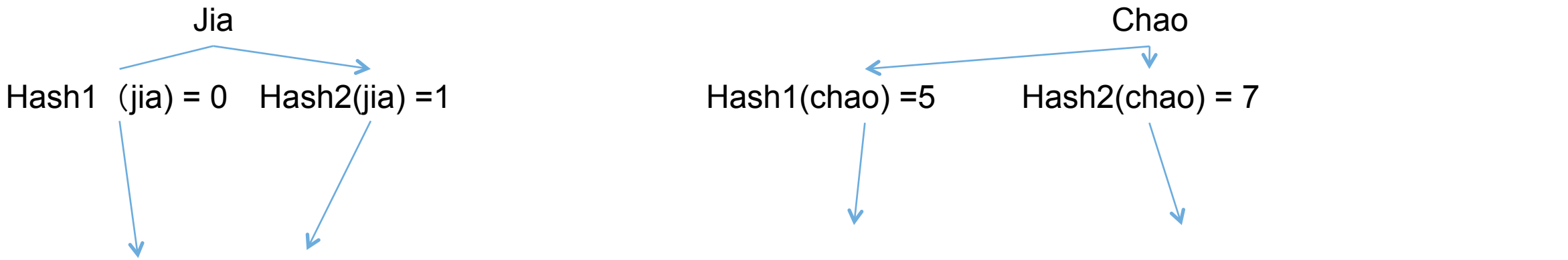
值	0	1	1	0	0	1	0	1	0	0
下标	0	1	2	3	4	5	6	7	8	9

3. 这个数组就是我们build好的bloom filter

Interview: How to look up in bloom filter?

Interview: How to build bloom filter?

- 对于建立好的bloom filter, 查找jia 和 chao，看是否在里面？

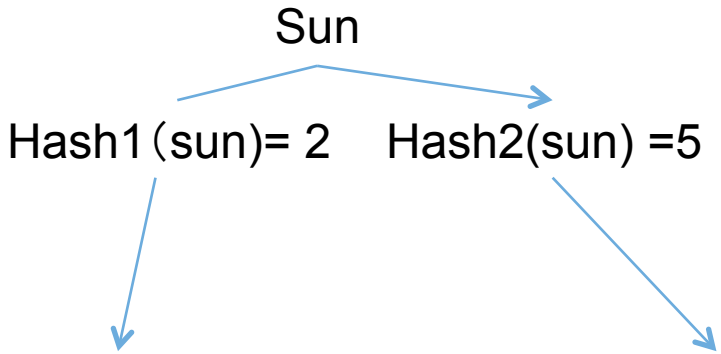


值	0	1	1	0	0	1	0	1	0	0
下标	0	1	2	3	4	5	6	7	8	9

sun

问题?

- Question: 一个key Sun 检查到在bloom filter 里面
- Answer:
  - Step 2: Index.
  - Bloom Filter 只是帮助过滤大部分情况。



值	0	1	1	0	0	1	0	1	0	0
下标	0	1	2	3	4	5	6	7	8	9

sun

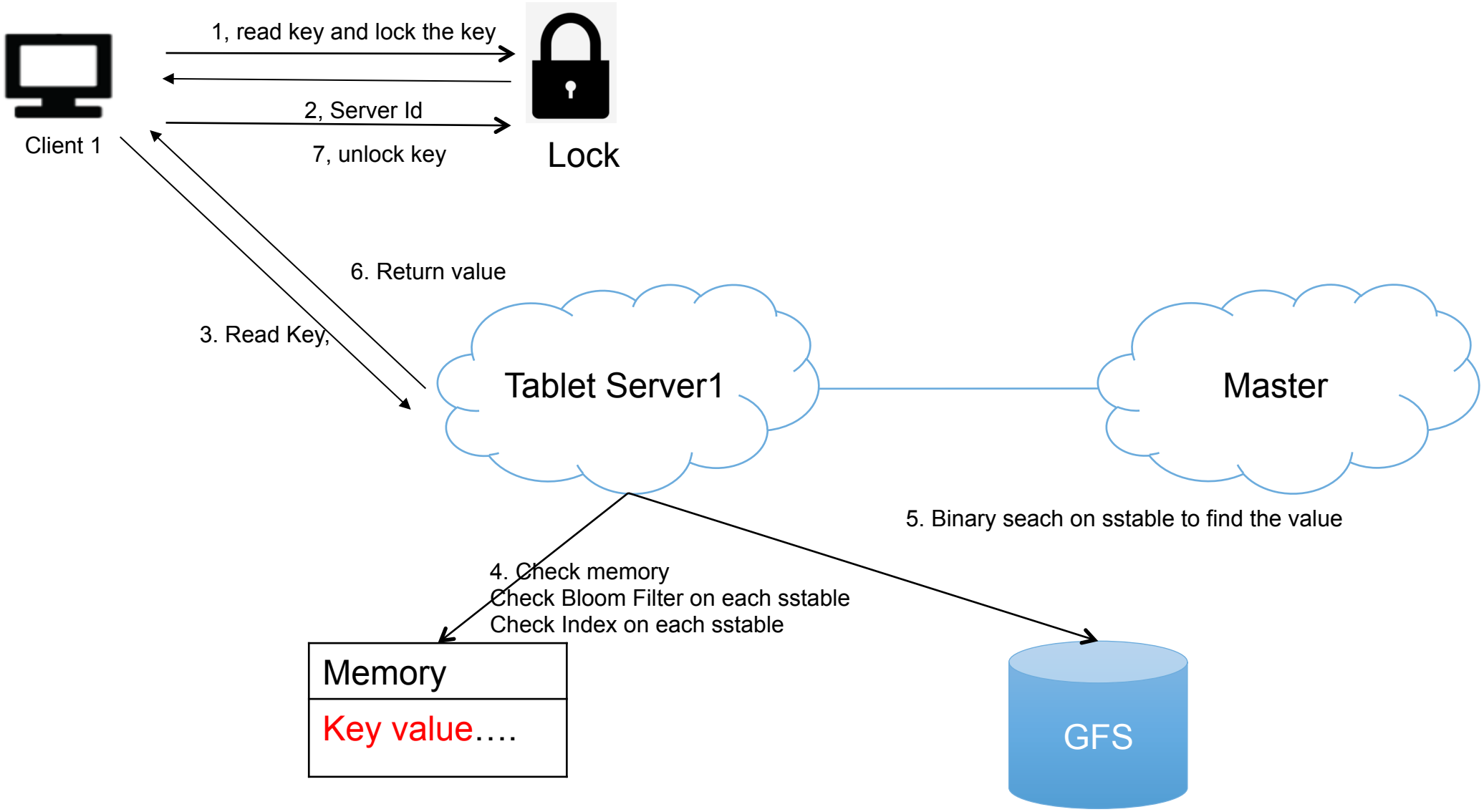
Bloom Filter

可以高效帮助我们查找key是否在sstable里面

Bloomfilter里面能够找到key的话，接下来我们再用index去查找

参考阅读:http://bit.ly/1sUPuwk

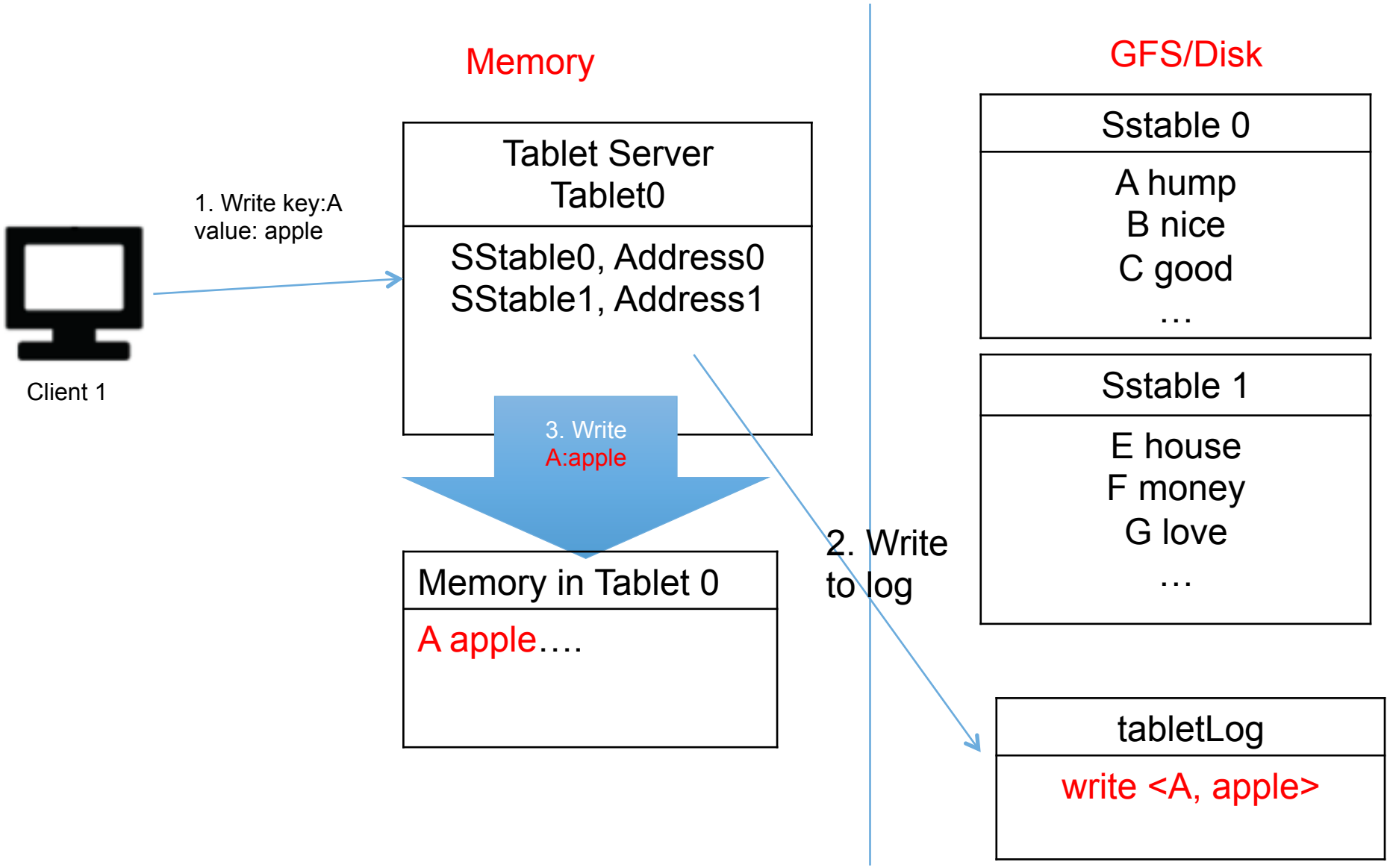
Summary of Read



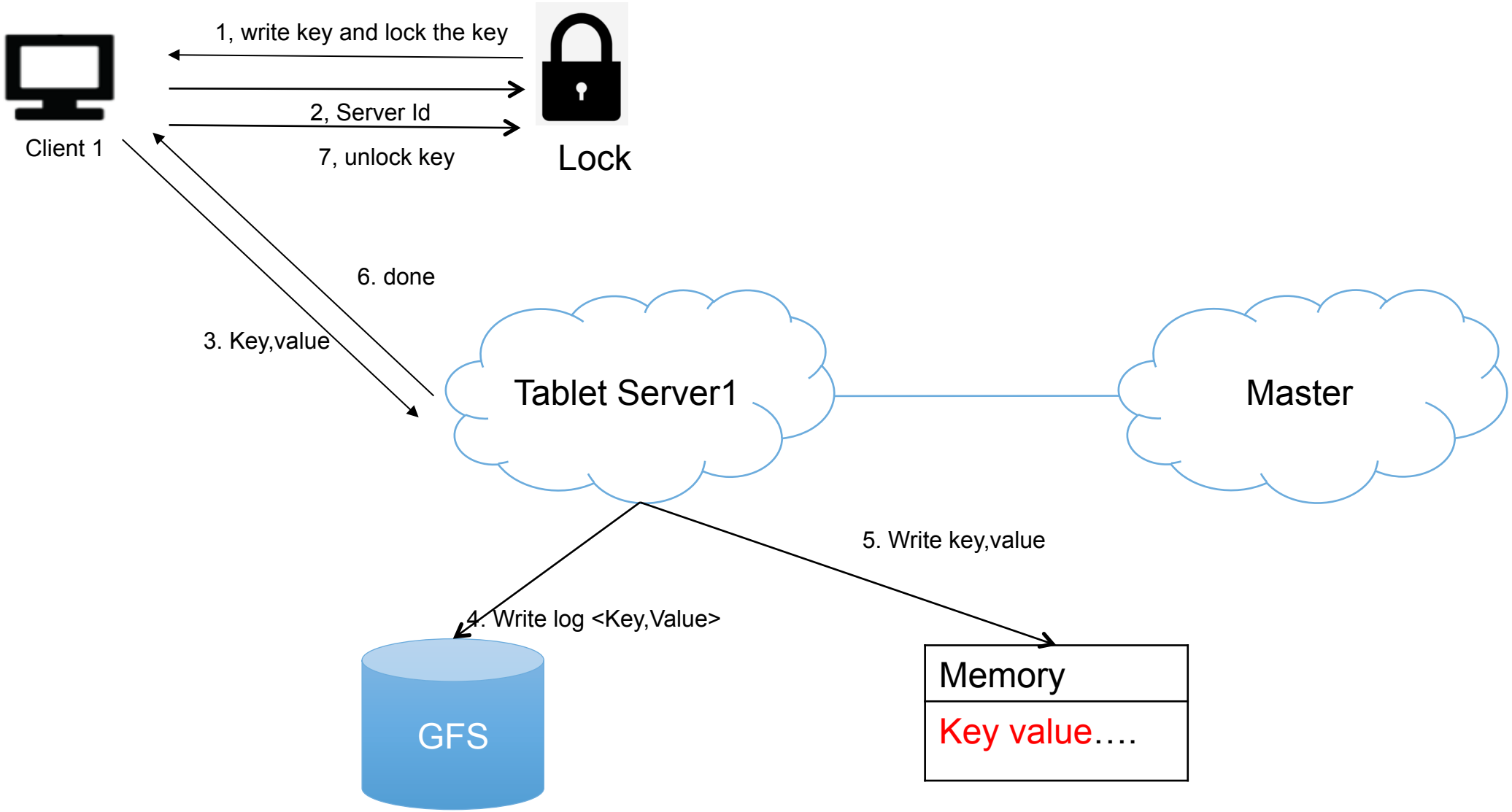
Interviewer: What if Server crashes when we write to memory?

Write Log

Interviewer: What if Server crashes?



Summary of write



Summary

- Bigtable
  - Tablet
  - SStable
  - How to read
    - Index
    - Bloom filter
  - How to write
    - Memtable
    - Write Log

Key Point

- Index
- Bloom filter
- Sstable

参考阅读

Bigtable

- http://www.cse.buffalo.edu/~mpetropo/CSE736-SP10/slides/seminar100409b1.pdf
- http://www.cs.colostate.edu/~cs435/slides/week11-B.pdf
- http://read.seas.harvard.edu/cs261/2011/bigtable.html
- http://the-paper-trail.org/blog/bigtable-googles-distributed-data-store/
- http://courses.cs.washington.edu/courses/csep552/13sp/lectures/6/bigtable.pdf

DevelDb+LSM Tree

- http://www.xuebuyuan.com/1537388.html

Map Reduce

Map Reduce

Why Map Reduce?

Distributed System is bulid for fast computing

大数据职位面试敲门砖

学会MapReduce可以找大数据工作

Interviewer: Count the word frequency of a web page?

Google 面试真题

http://www.lintcode.com/en/problem/word-count/

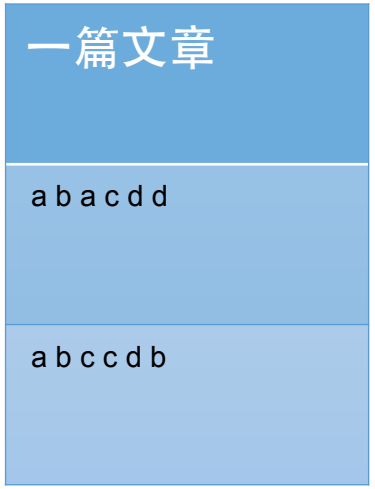
http://www.jiuzhang.com/solutions/word-count/

常见土方法一 For循环

方法一 For循环

伪代码

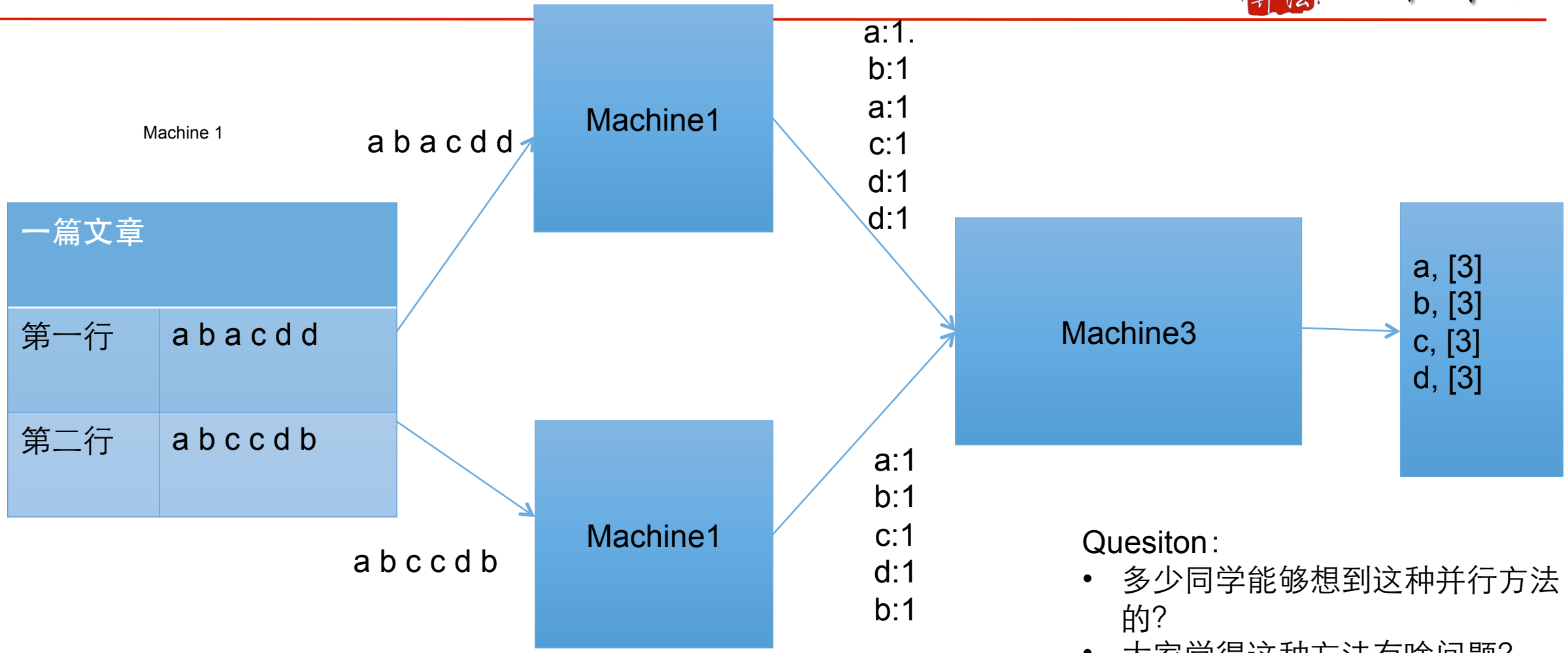
- HashMap<String,int> wordcount;
- for each word in webpage
  - wordcount[word]++



- Question?
  - 多少同学能够想到这种方法?
  - 如果你有多台机器呢?

常见土方法二 多台机器For循环

方法二 多台机器For循环



Quesiton:

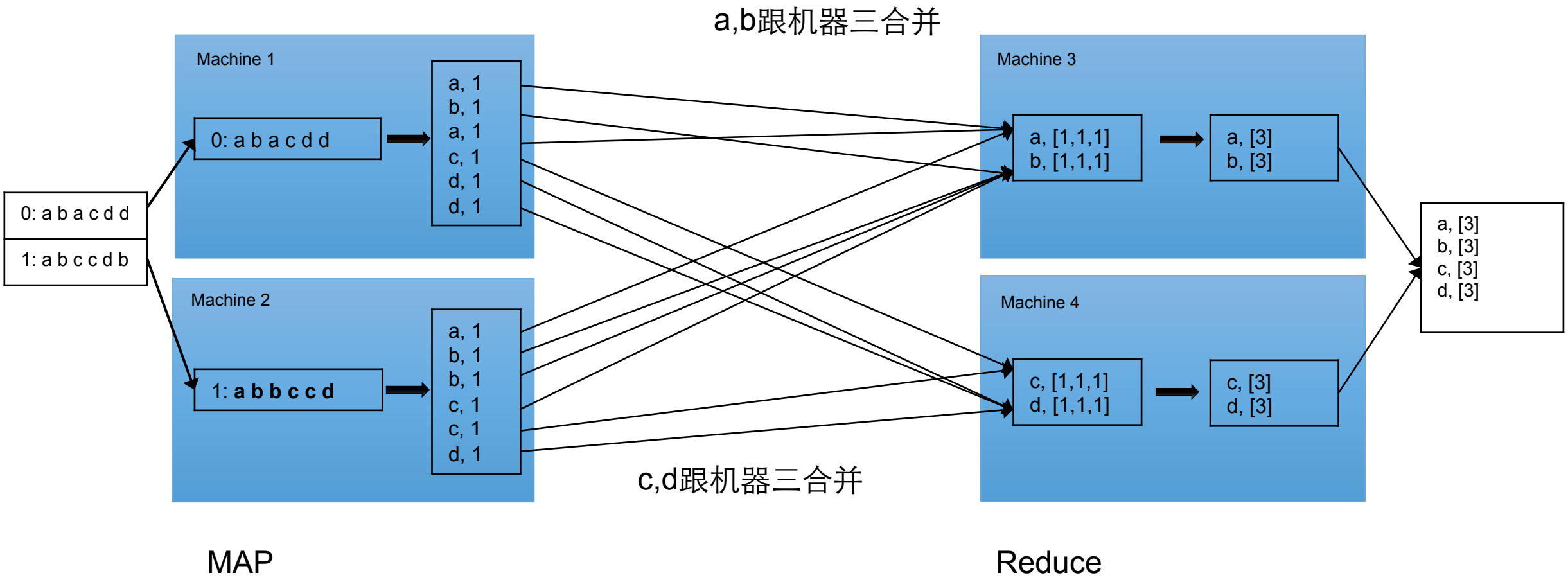
- 多少同学能够想到这种并行方法的?
- 大家觉得这种方法有啥问题?

合并的时候是Bottle Neck

合并是否也可以并行?

方法三 多台机器Map Reduce

方法三：Map Reduce



多台机器Map Reduce

Map

- 机器1，2 只负责把文章拆分为一个一个的单词

Reduce

- 机器3，4各负责一部分word的合并

Map Reduce

Map

把文章拆分单词的过程

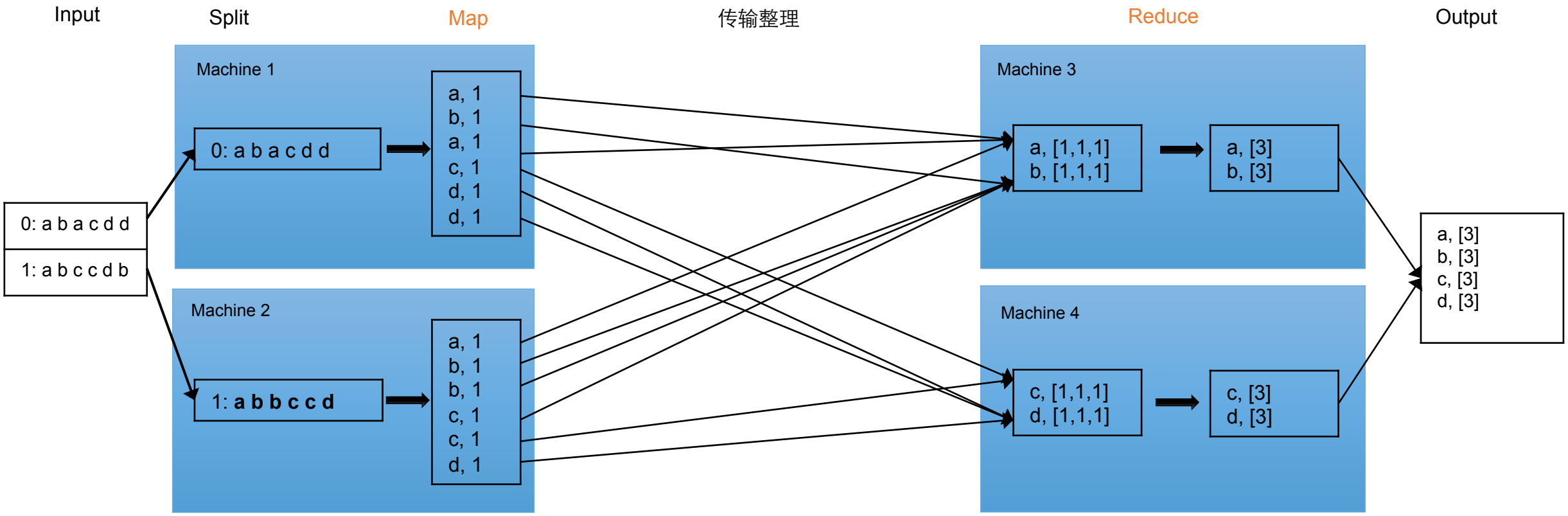
Reduce

把单词次数合并在一起的过程

Map Reduce Steps

- Map Reduce 是一套实现分布式运算的框架
- Step1 Input
- Step2 Split
- Step3 Map
- Step4 传输整理
- Step5 Reduce
- Step6 Output

Word Count



我们要实现什么代码呢?

我们要实现什么呢?

Map 函数 和 Reduce 函数

Map Reduce Steps

- Map Reduce 是一套实现分布式运算的框架
- Step1 Input
- Step2 Split
- Step3 Map 实现怎么把文章切分成单词
- Step4 传输整理
- Step5 Reduce 实现怎么把单词统一在一起
- Step6 Output

Map Reduce 函数接口是什么？

他们的输入和输出必须是Key Value 形式

Map 输入: key:文章存储地址, Value: 文章内容

Reduce 输入： key:map输出的key, value: map输出的value

Google面试真题实战

http://www.lintcode.com/en/problem/word-count/

http://www.jiuzhang.com/solutions/word-count/

Map Reduce Steps

- Map Reduce 是一套实现分布式运算的框架
- Step1 Input 设定好输入文件
- Step2 Split 系统帮我们把文件尽量平分到每个机器
- Step3 Map 实现代码
- Step4 传输整理 系统帮我们整理
- Step5 Reduce 实现代码
- Step6 Output 设定输出文件

Map Reduce Steps

- Question1?
- Map 多少台机器？ Reduce 多少台机器?
  - 全由自己决定。 一般1000map, 1000reduce规模

- Question2?（加分）
- 机器越多就越好么?
  - Advantage:
    - 机器越多，那么每台机器处理的就越少，总处理数据就越快
  - Disadvantage:
    - 启动机器的时间相应也变长了。

- Question3?（加分）
  - 如果不考虑启动时间，Reduce 的机器是越多就一定越快么?
    - Key的数目就是reduce的上限

Apple Interviewer: Build inverted index with MapReduce?

http://www.lintcode.com/en/problem/inverted-index-map-reduce/#

http://www.jiuzhang.com/solutions/inverted-index-map-reduce/

Read More:
Novice/Expert, http://url.cn/fsZ927

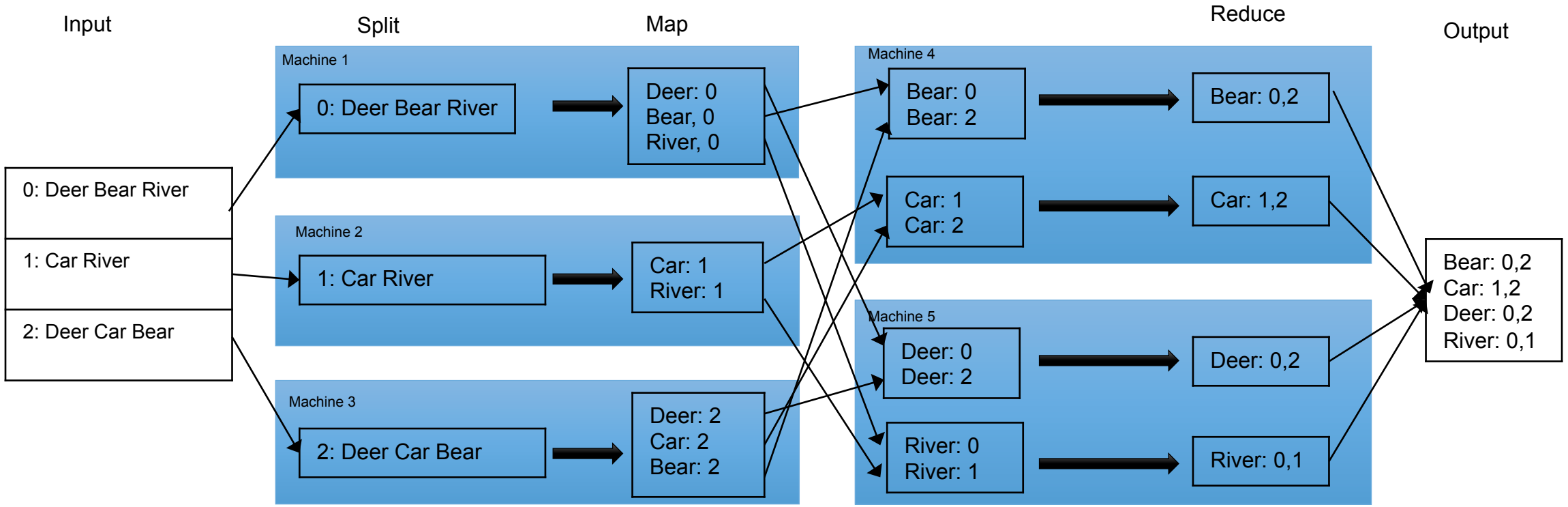
Input

0: Deer Bear River
1: Car River
2: Deer Car Bear

Output

Bear: 0,2
Car: 1,2
Deer: 0,2
River: 0,1

Build inverted index with MapReduce?




```
//key: the id of a doc
//value: the content of the line
Map( string key, string value)
    for each word in value:
        Output( word, key);
```

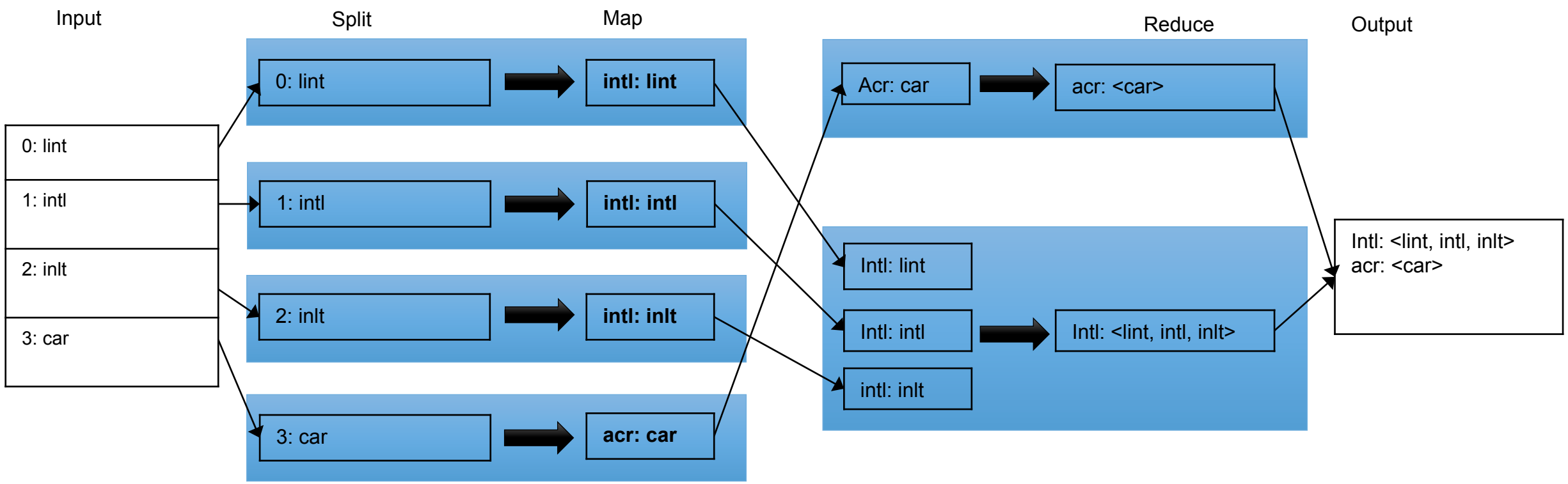
```
//key: the name of a word
//valueList: the appearances of this word in documents
Reduce( string key, list valueList )
    List sumList;
    for value in valueList:
        sumList.append(value);
    OutputFinal( key, sumList );
```

Apple Interviewer: Anagram - Map Reduce

http://www.lintcode.com/en/problem/anagram-map-reduce/

http://www.jiuzhang.com/solutions/anagram-map-reduce/

Anagram - Map Reduce



Map

Input
k = lineID
v = word

Output
k = sort(word)
v = word

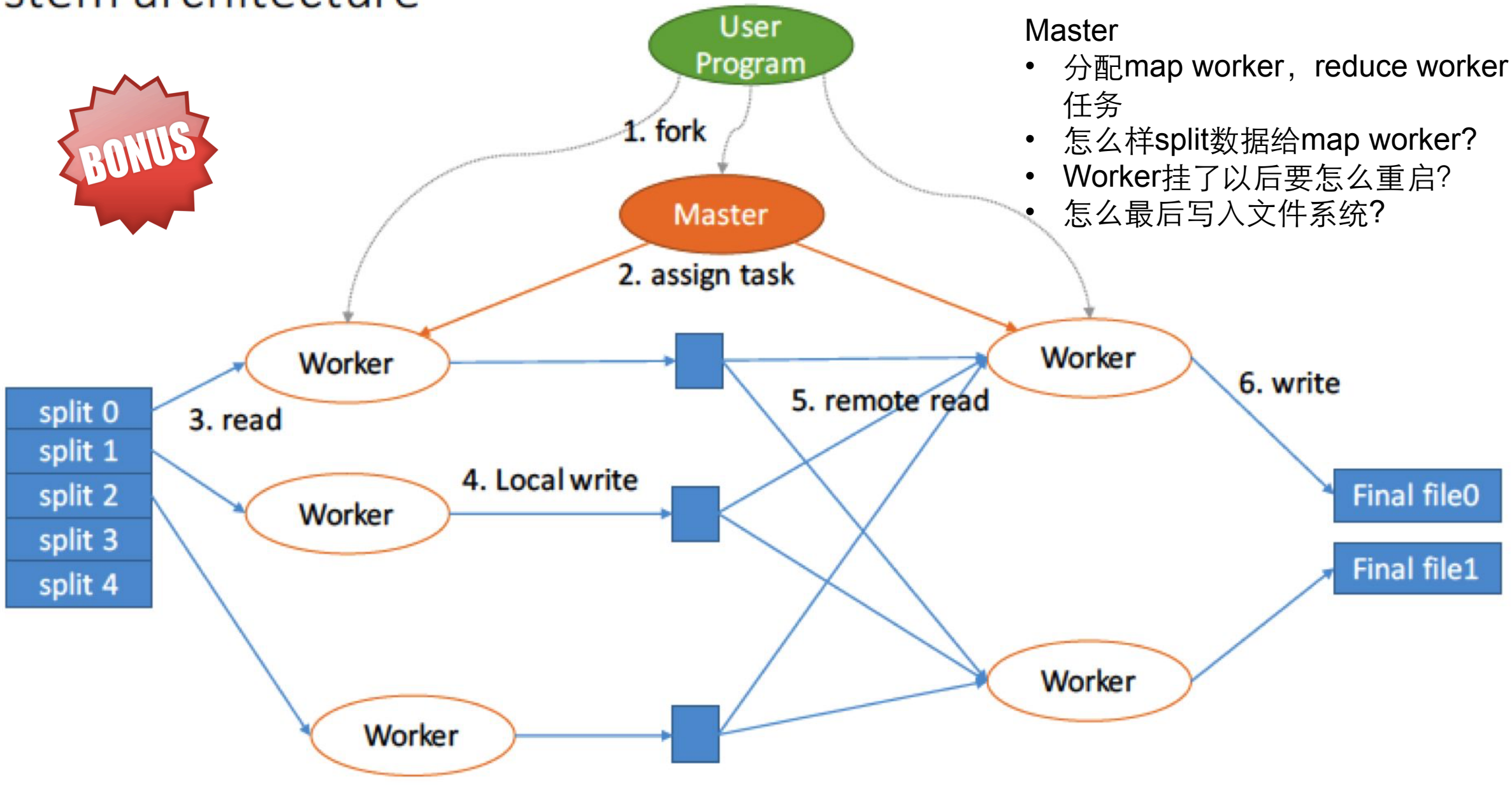
Reduce

Input
k = sort(word)
v = {anagram0, anagram1, ...}

Output
k = anagram0
v = {anagram1, anagram2,...}

Interviewer:
Design a MapReduce system

System architecture



Master

- 分配map worker，reduce worker任务
- 怎么样split数据给map worker?
- Worker挂了以后要怎么重启?
- 怎么最后写入文件系统?

Where does MapReduce input / output store?

GFS(Sstable)

MapReduce FrameWork

- Map Reduce Solve Problem
  - Words Count
  - Inverted index
  - Anagrams
  - Top K Frequency (http://bit.ly/25D8Q7I)
  - PageRank (http://bit.ly/1TOwoyV)
- Map Reduce Step
  - Step1 Input
  - Step2 Split
  - Step3 Map
  - Step4 传输
  - Step5 Reduce
  - Step6 Output
- Map Reduce System
  - Master and Worker
- More
  - 大数据班尽请期待.....

相关阅读资料

- Novice, http://url.cn/YM1tHI
- Expert, http://url.cn/b41Qzf
- Expert, http://url.cn/1VO6Qa
- Expert, http://url.cn/ccvLOr
- Expert/Master, http://url.cn/SuVoAP
- Expert/Master, http://url.cn/SJCoso
- Master, http://url.cn/Z3OOVZ

Summary of Today

- Bigtable
  - Index
  - Bloom Filter
  - Sstable 读和写
- Map Reduce
  - Word Count
  - Inverted Index

、

、