



# How to design GFS/BigTable/MapReduce

CS103 系统设计

Producer: 冯沁原

1.00



课程咨询



## Copyright Policy

All content included on the Site or third-party platforms as part of the class, such as text, graphics, logos, button icons, images, audio clips, video clips, live streams, digital downloads, data compilations, and software, is the property of BitTiger or its content suppliers and protected by copyright laws.

Any attempt to redistribute or resell BitTiger content will result in the appropriate legal action being taken.

We thank you in advance for respecting our copyrighted content.

For more info see <https://www.bittiger.io/termsofuse> and <https://www.bittiger.io/termsofservice>



## 版权声明

所有太阁官方网站以及在第三方平台课程中所产生的课程内容，如文本，图形，徽标，按钮图标，图像，音频剪辑，视频剪辑，直播流，数字下载，数据编辑和软件均属于太阁和合作伙伴所有并受版权法保护。

**对于任何尝试散播或转售太阁所属资料的行为，太阁将采取相应的法律行动。**

我们非常感谢您尊重我们的版权内容。

详情参阅，

<https://www.bittiger.io/termsofuse>

<https://www.bittiger.io/termsofservice>

# Interviewer: design search engine

# Three Musketeers of Google



MapReduce  
(名人)

BigTable  
( ? ? )

GFS  
(高富帅)

Read More

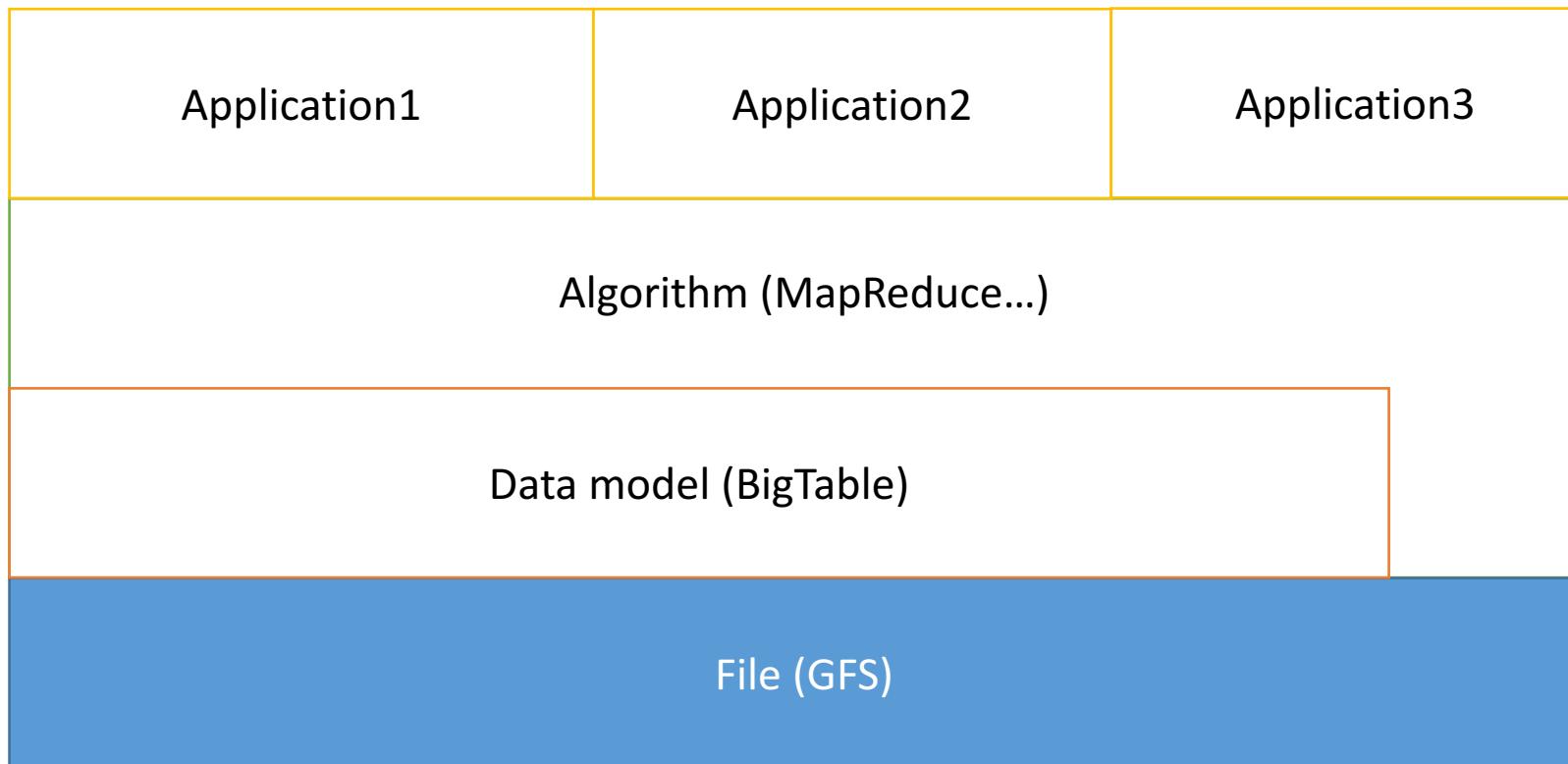
Novice, <http://url.cn/SVDkio>  
Novice, <http://url.cn/deN2uC>  
Novice, <http://url.cn/Vf6ABZ>  
Expert/Master, <http://url.cn/EZNxGe>  
Expert/Master, <http://url.cn/dxcRxl>



Do not recite the details of papers,  
but find a suitable solution under your scenario.

Interviewer: What is the layers of your search engine?

# Layers of system



Let's design with a bottom-up method

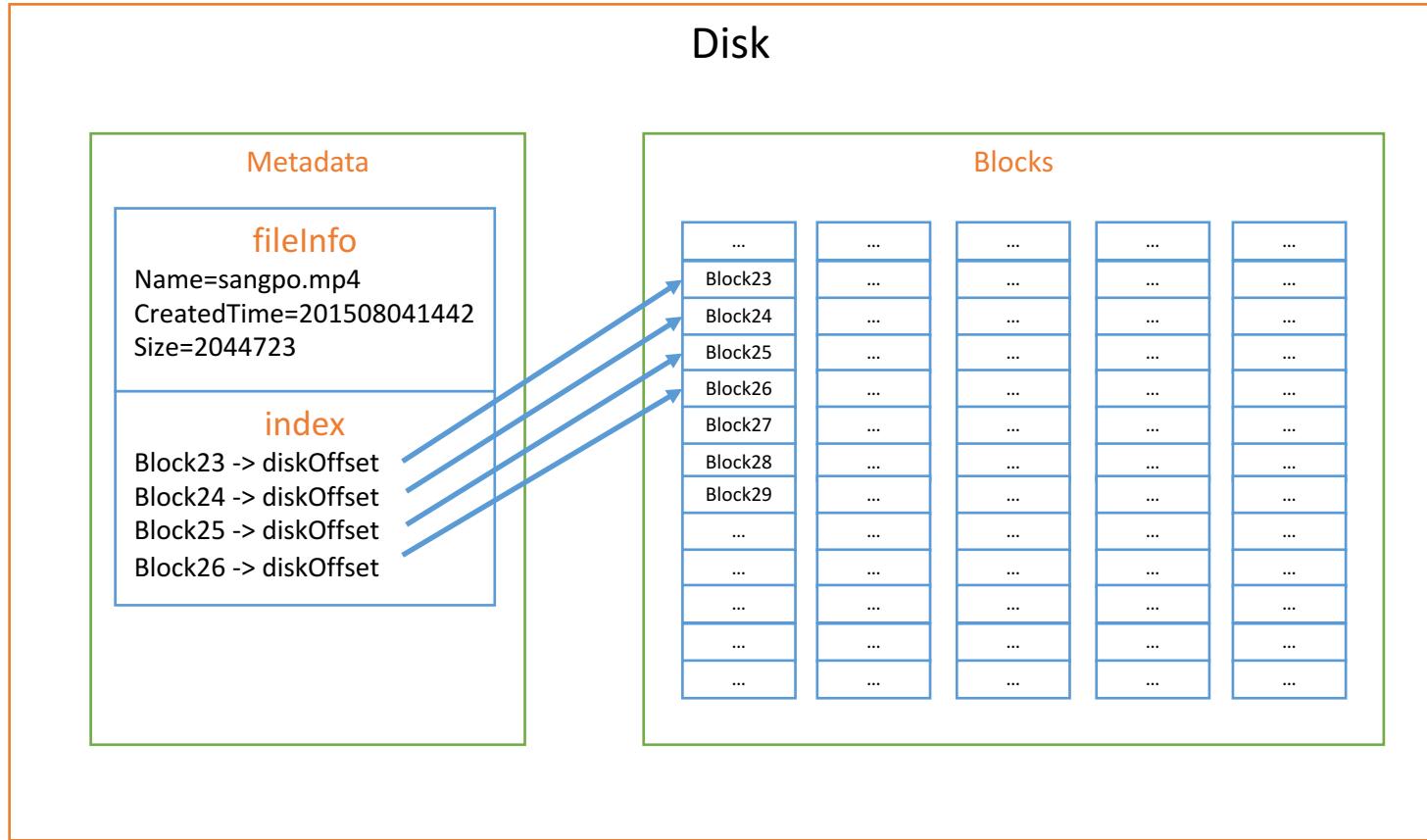
# Google File System

[Read More](#)

Expert/Master, <http://url.cn/dOLFCs>  
Expert/Master, <http://url.cn/eErkhn>  
Expert/Master, <http://url.cn/LqTko>

# Interviewer: How to save a file?

# How to save a file?

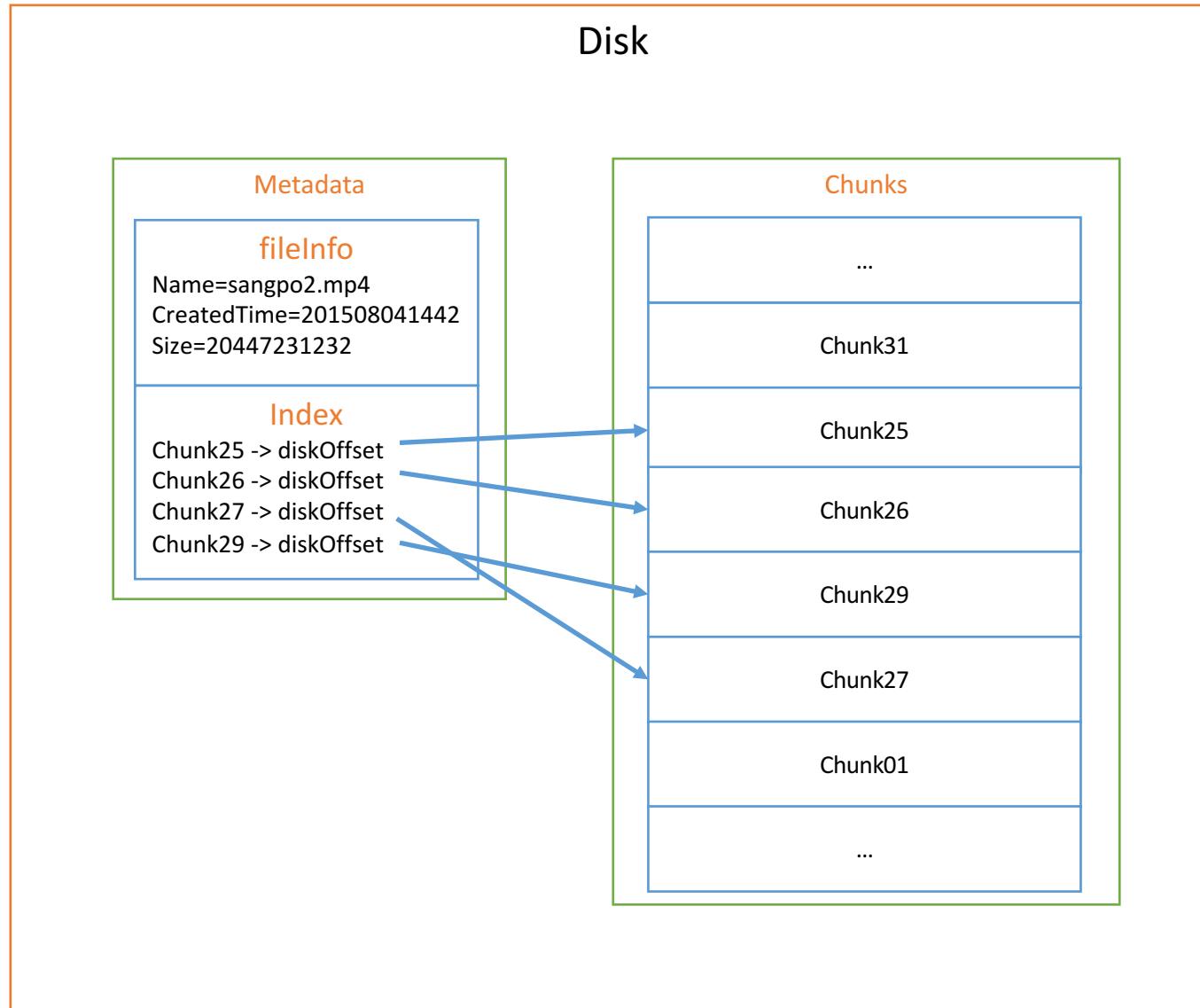


## Key point

- 1 block = 1024Byte

Interviewer: How to save a large file?

# How to save a large file?

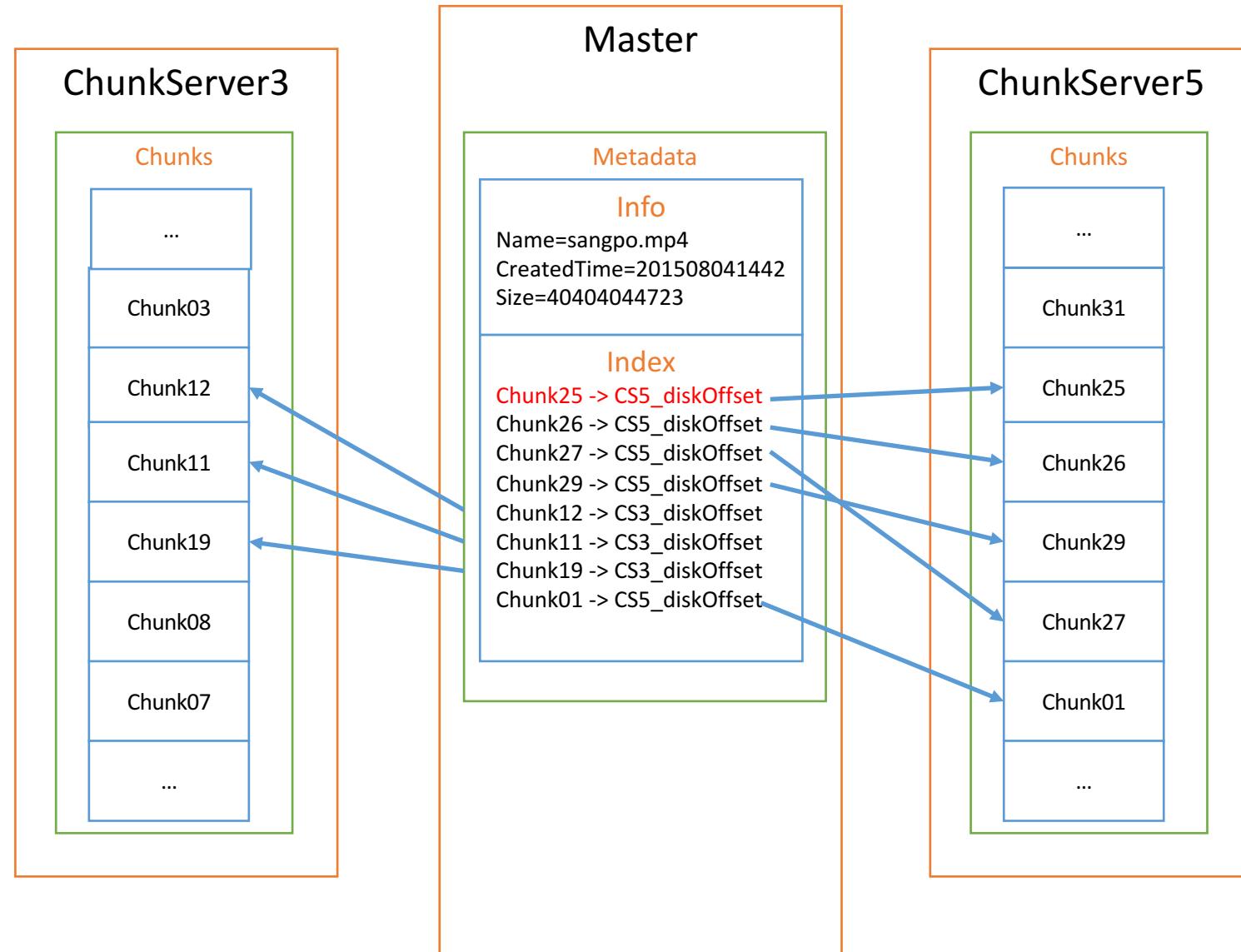


- Key point
  - 1 chunk = 64MB  
=  $64 * 1024$   
= 65,536 blocks
- Advantage
  - Reduce size of metadata
  - Reduce traffic
- Disadvantage
  - Waste space for small files



**Interviewer: How to save an extra-large file?**

# How to save an extra-large file?

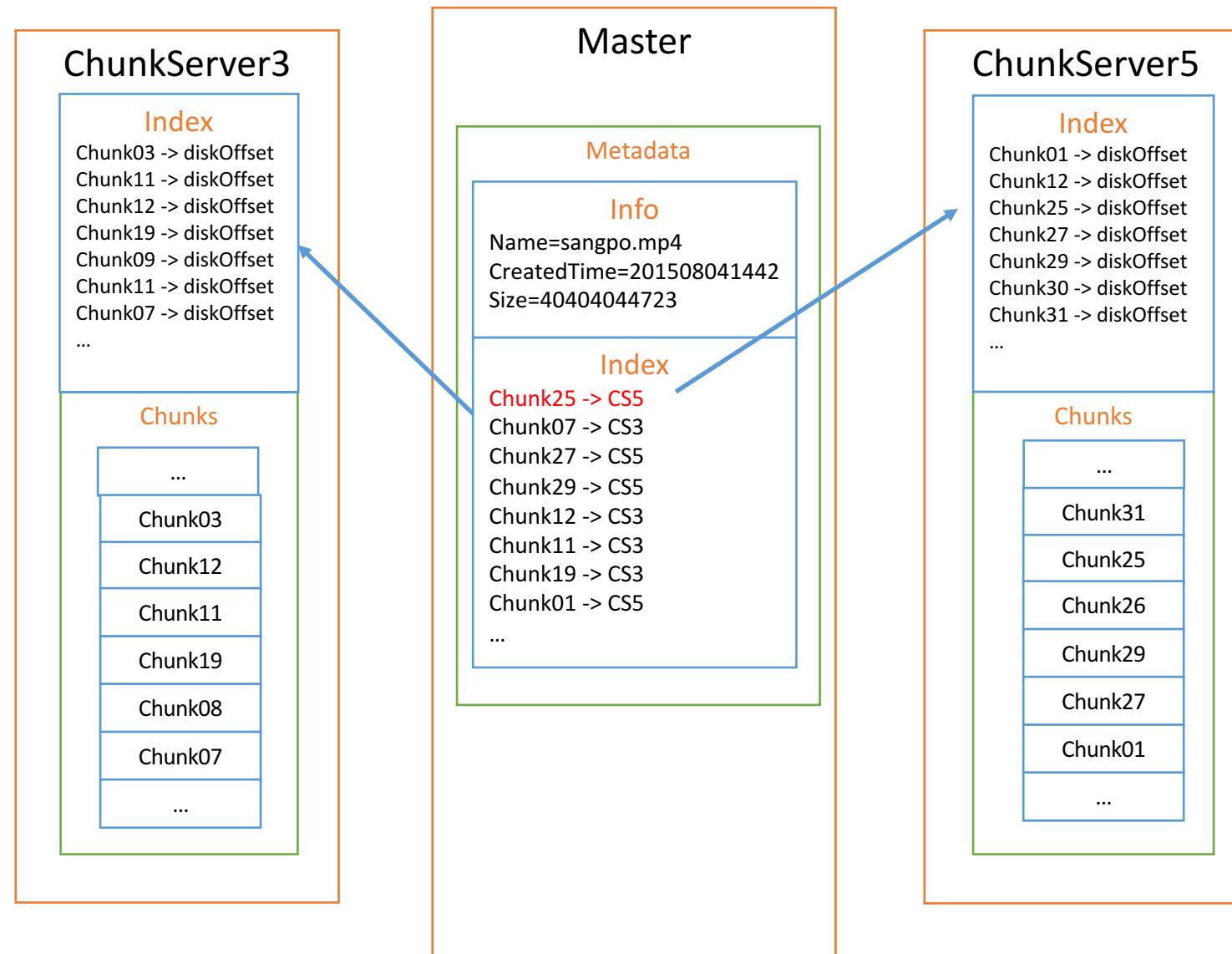


- Key point
  - One master + many ChunkServers
- Disadvantage
  - Any change of the diskOffset of a chunk in a ChunkServer need to notify the Master



Interviewer: How to reduce traffic and storage of master?

# How to reduce traffic and storage of master?

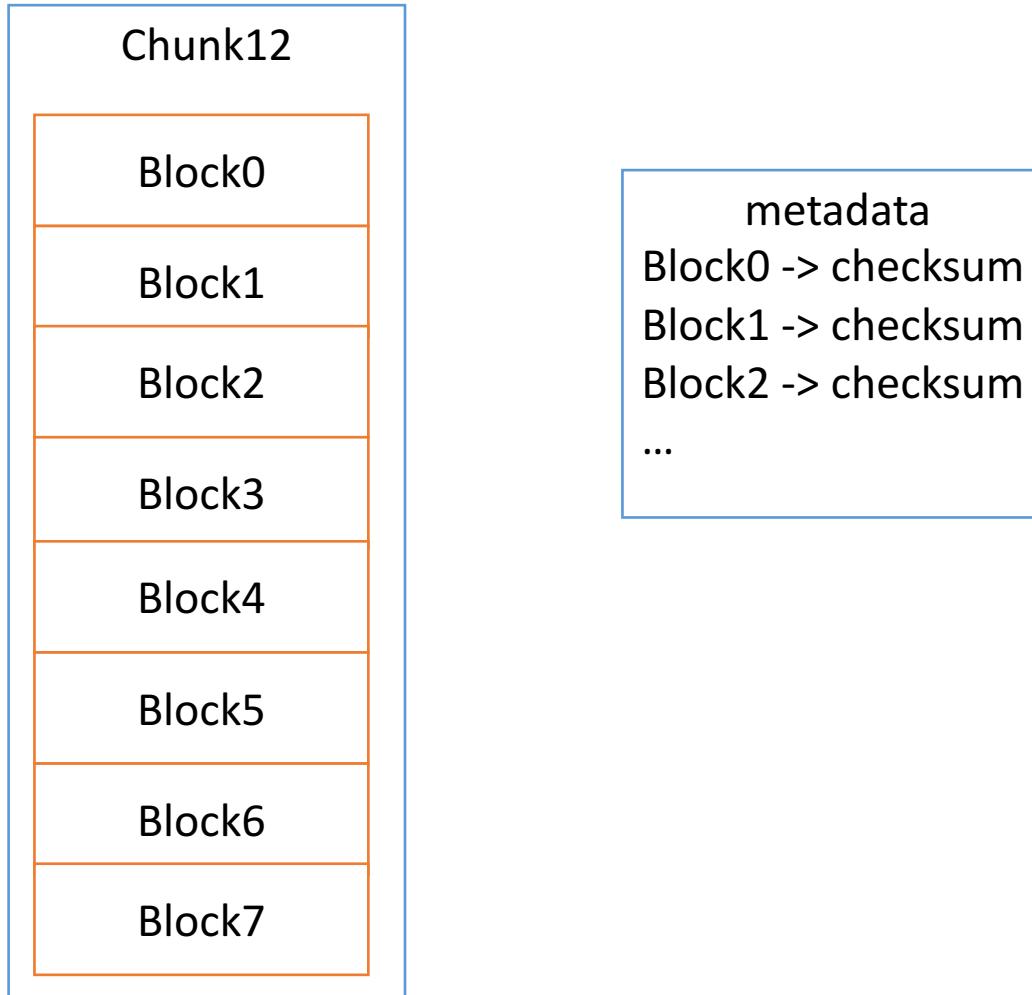


- Key point
  - The master don't record the diskOffset of a chunk
- Advantage
  - Reduce the size of metadata in master
  - Reduce the traffic between master and ChunkServer



**Interviewer: How to identify whether a  
chunk on the disk is broken?**

# How to identify whether a chunk on the disk is broken?



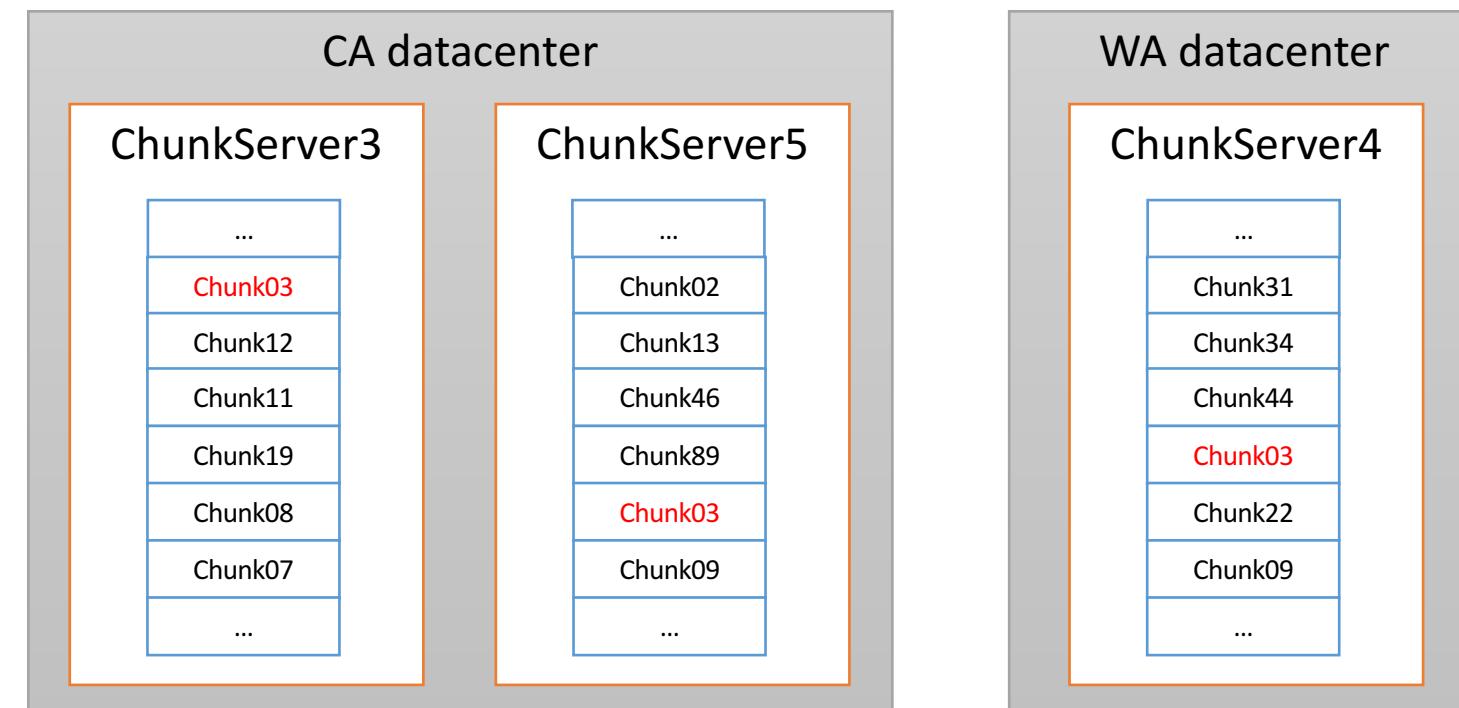
- Key point

- 1 chunk = a list of blocks
- 1 block = 64KB
- Each block has a **checksum**
- 1 checksum = 32bit
- The size of checksum of 1T file
  - $1\text{T}/64\text{KB} \times 32\text{bit} = 64\text{MB}$
- It will compare its **checksum** when it reads a block



Interviewer: How to avoid data loss when  
a ChunkServer is down/fail?

# How to avoid the loss of data when a ChunkServer is down/fail?



- **Key point**

- Replicate the Chunks
- How many replications?
  - 3
- How to select ChunkServers of a chunk?
  - Server with below-average disk space utilization
  - Limited number of “recent” creation
  - Across racks: 2+1

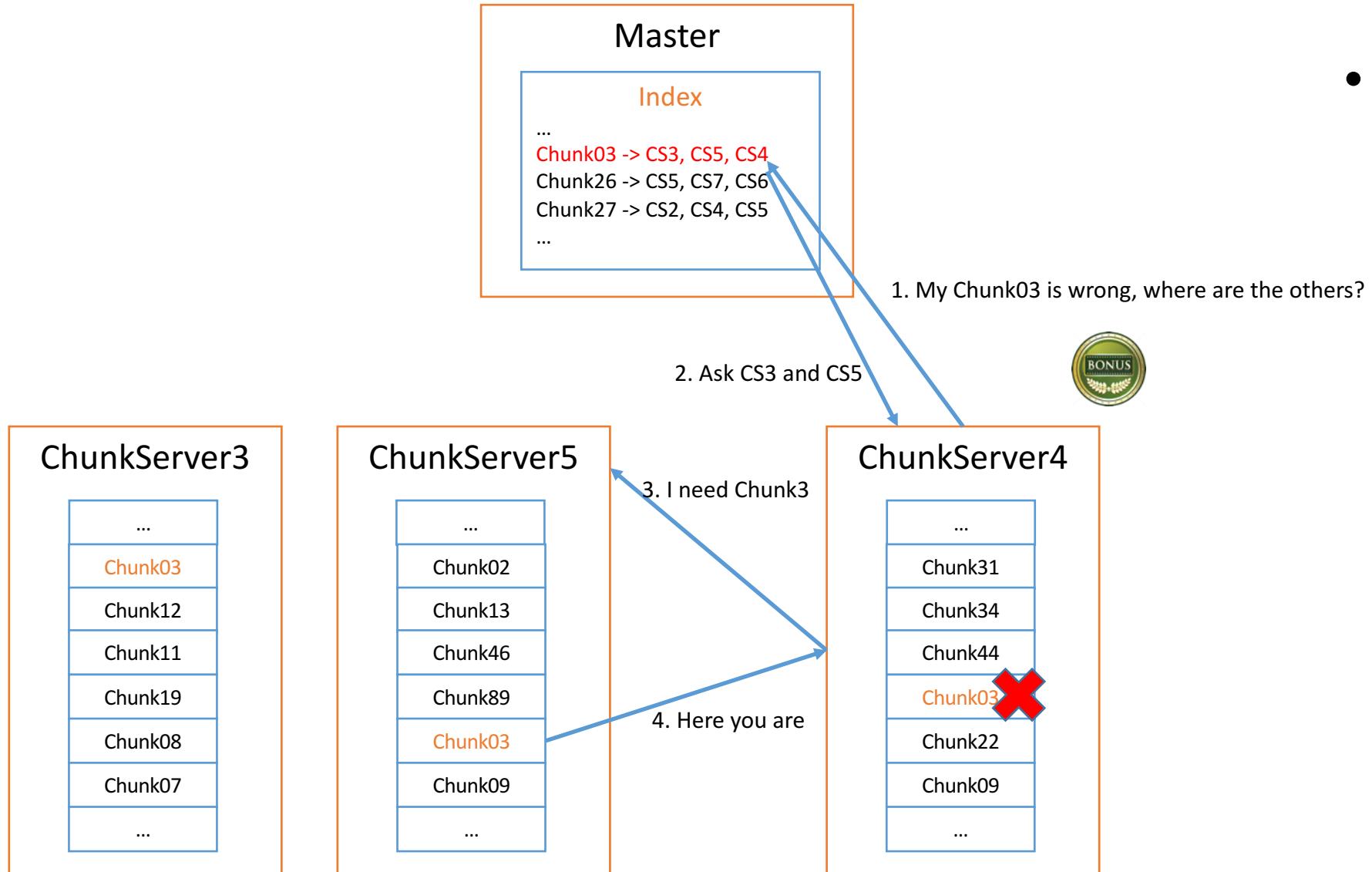


[Read More](#)

Expert/Master, <http://url.cn/d1BrZU>

**Interviewer: How to recover when  
a chunk is broken?**

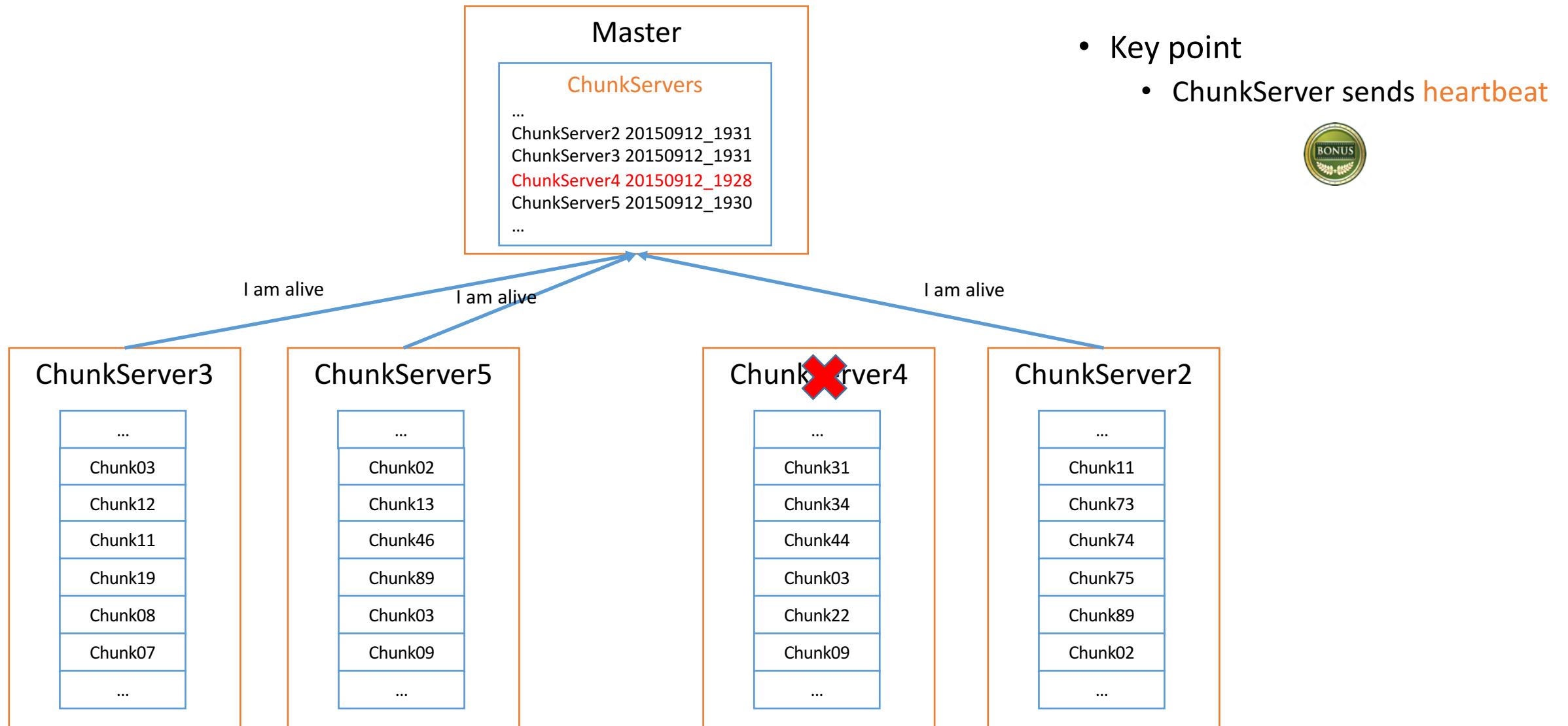
# How to recover when a chunk is broken?



- Key point
  - Ask master for help

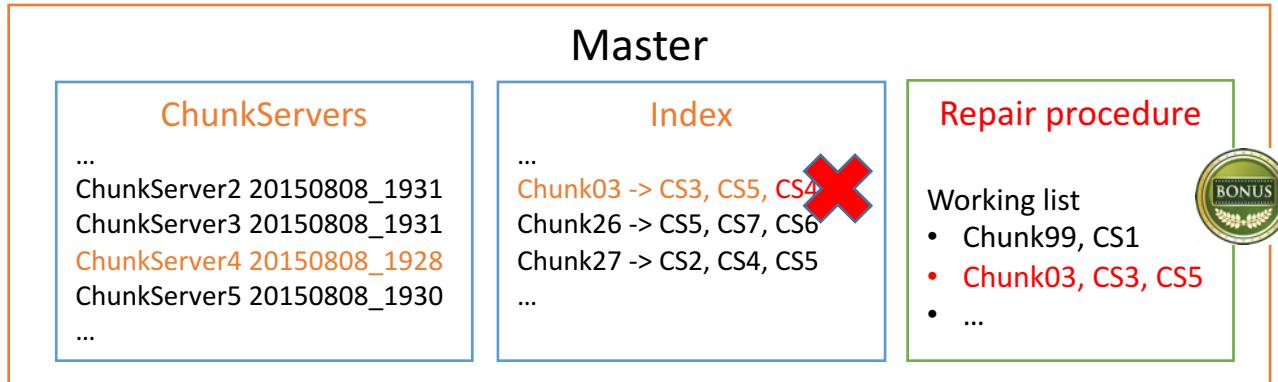
Interviewer: How to find whether a  
ChunkServer is down?

# How to find whether a ChunkServer is down?

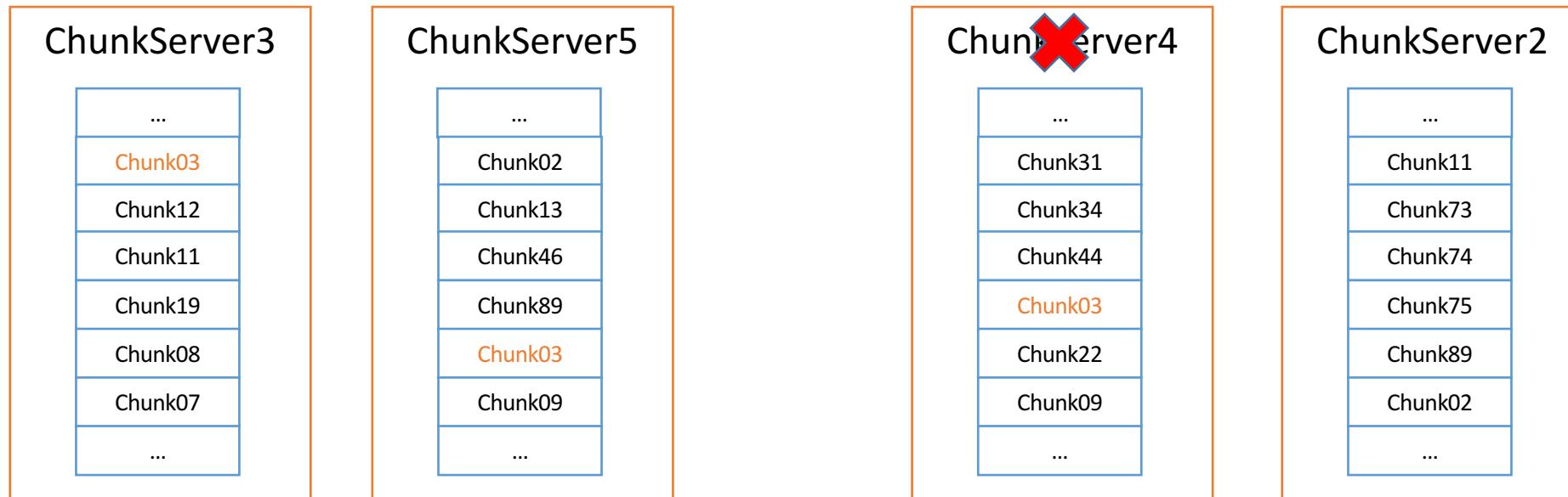


Interviewer: How to recover the data  
when a ChunkServer is down?

# How to recover data when a ChunkServer is down?

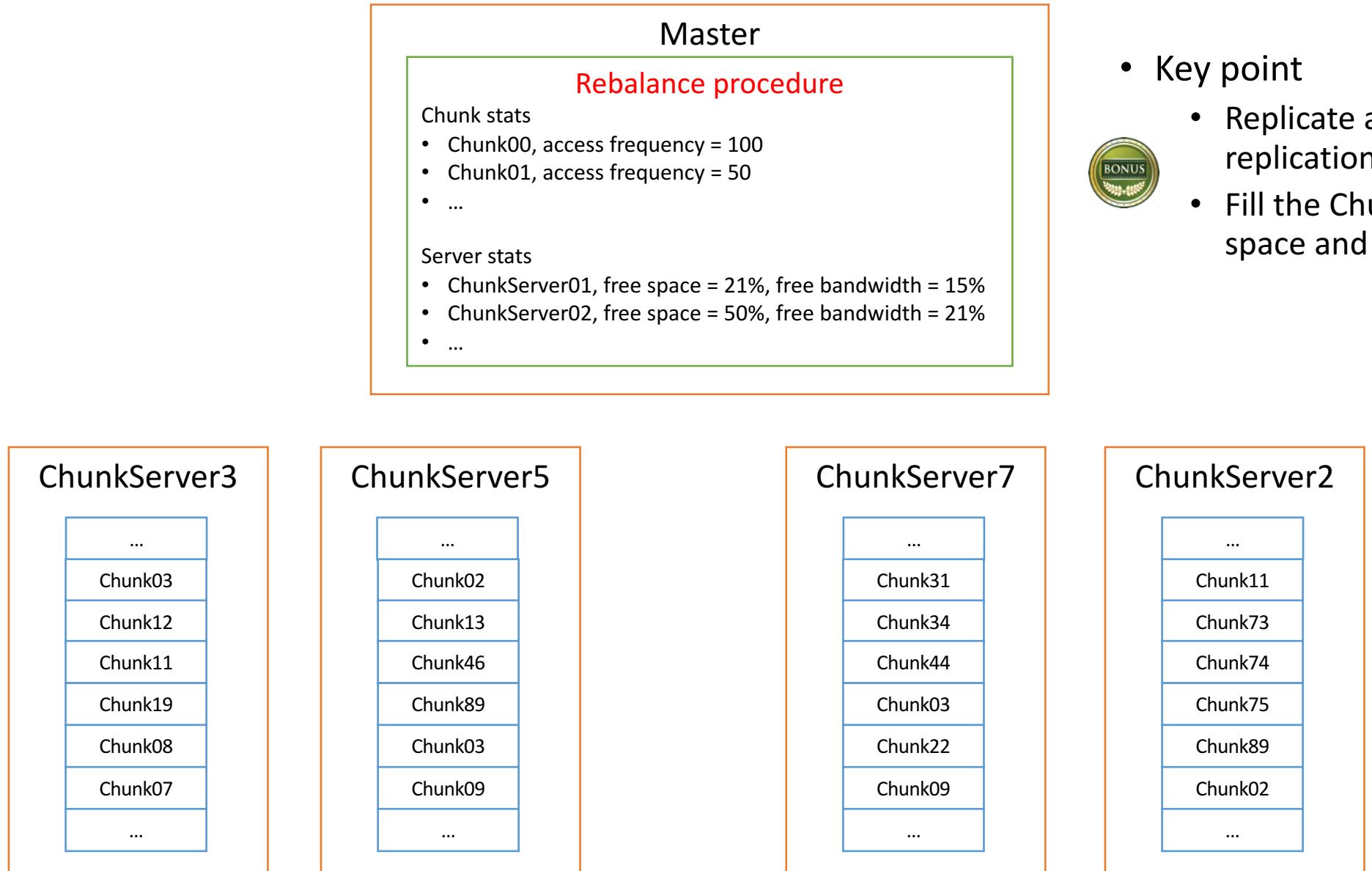


- Key point
  - Repair priority is based on the number of replications



# Interviewer: How to avoid hot spot?

# How to avoid hot spot?



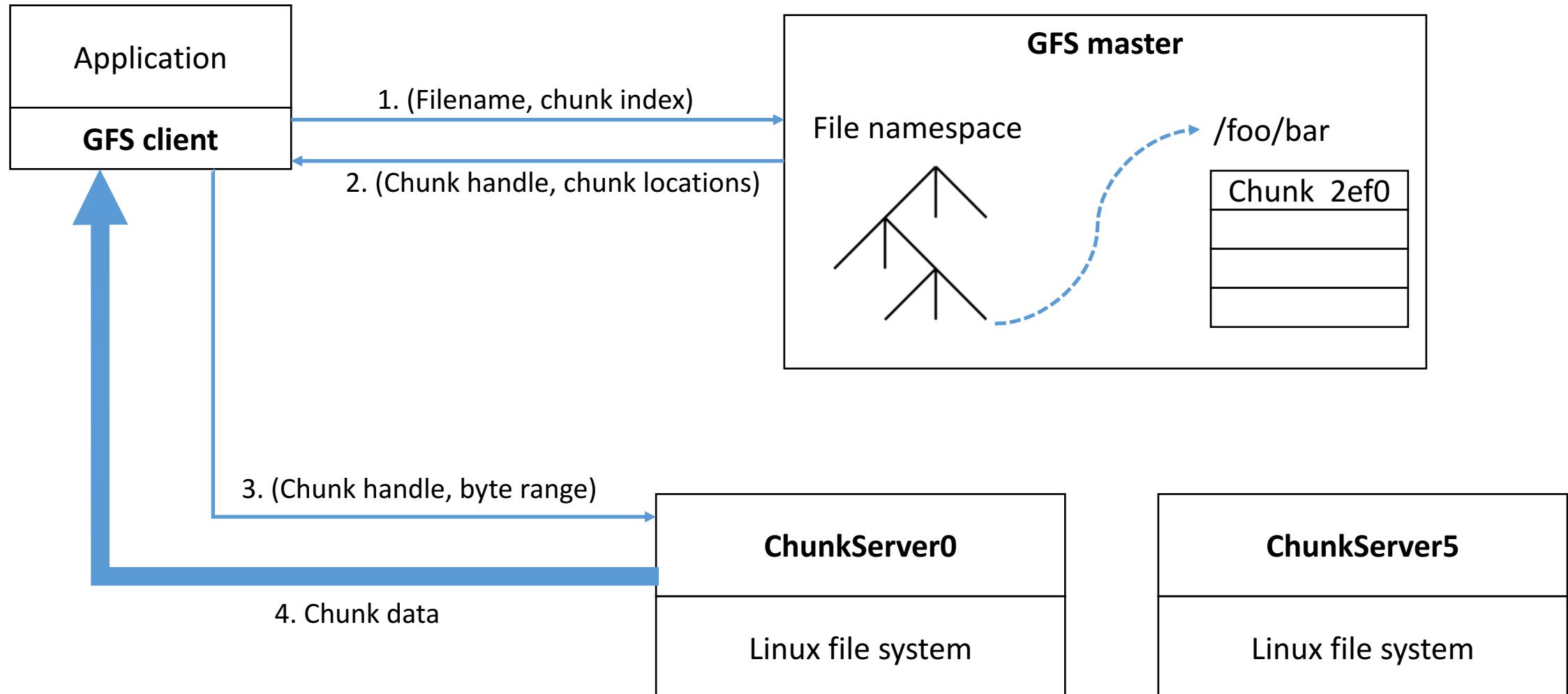
- Key point

- Replicate a chunk into more replications when it is busy
- Fill the ChunkServer with more space and more bandwidth



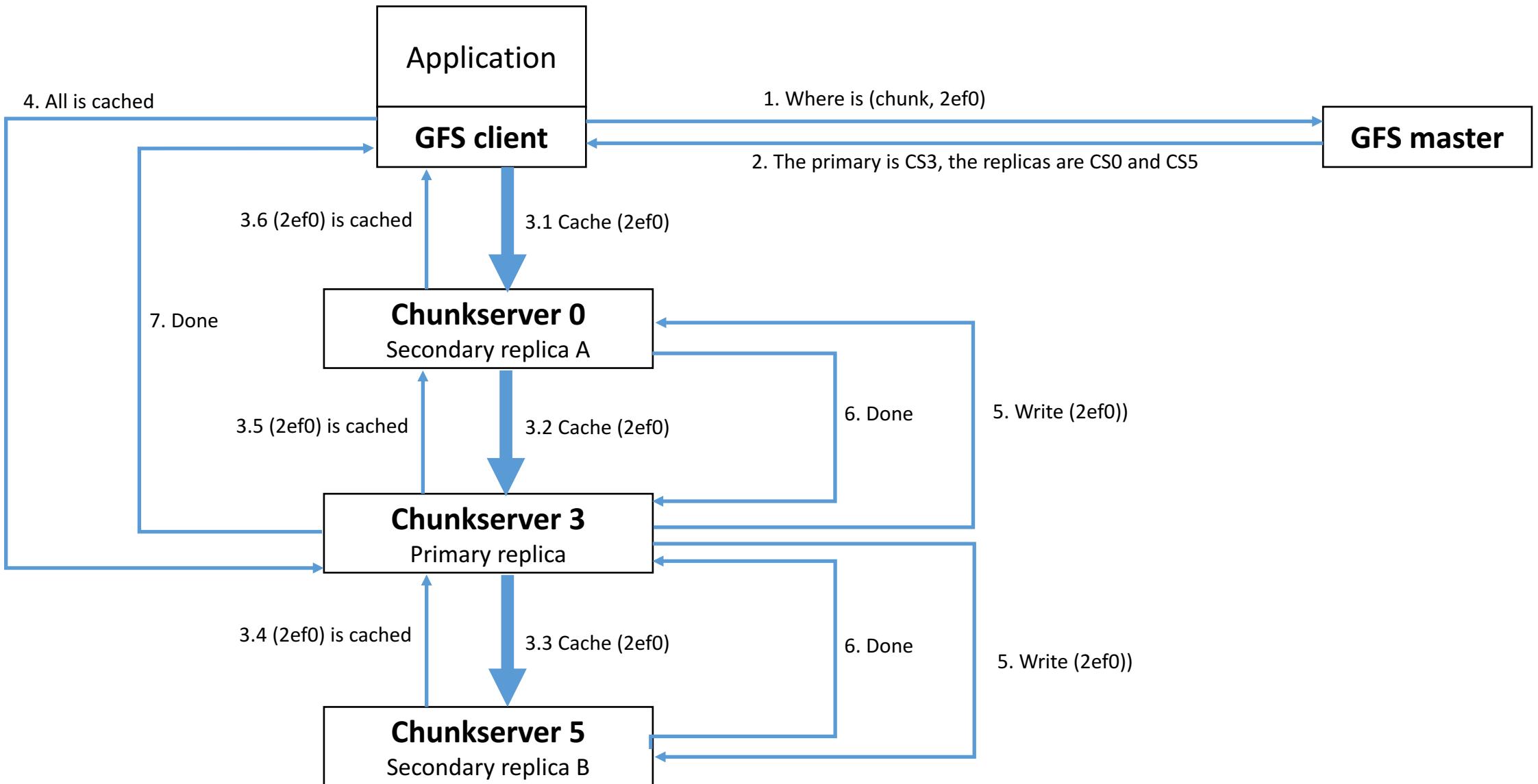
**Interviewer: How to read from a file?**

# How to read from a file?

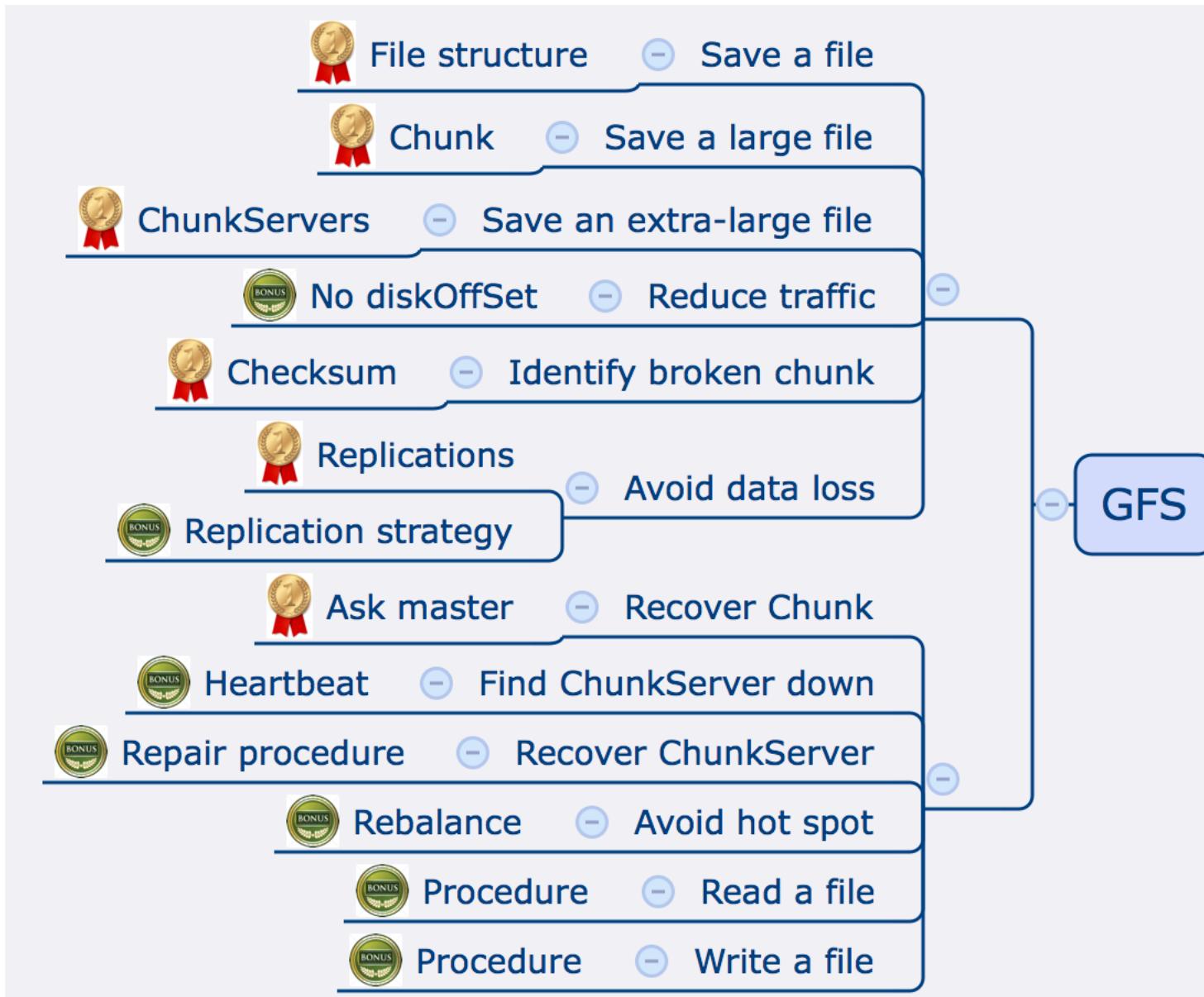


**Interviewer: How to write into a file?**

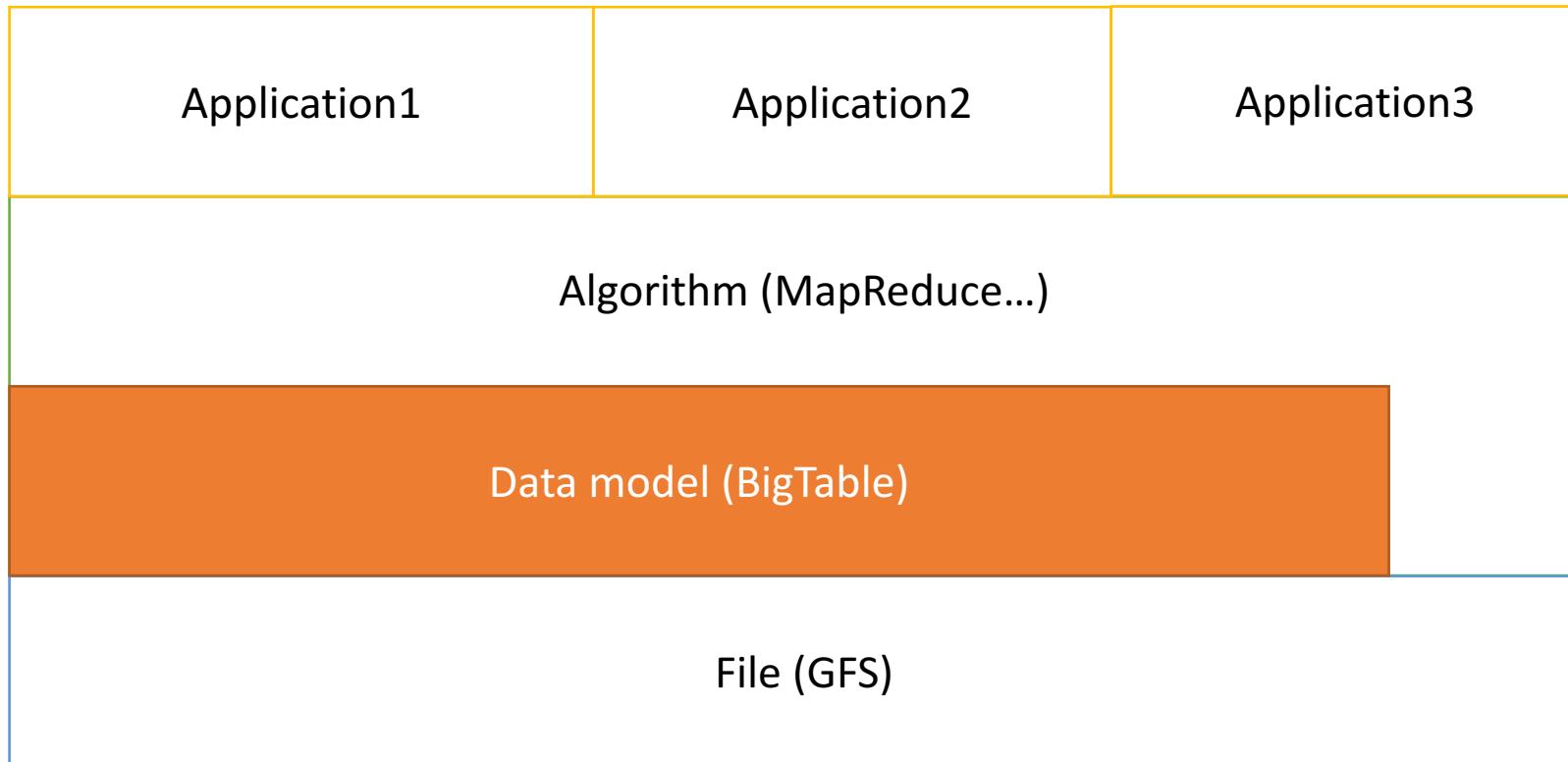
# How to write into a file?



# GFS Summary (6+7)



# Layers of system



我又不是因為想  
得到你們的承認  
才當英雄的

是因為  
我想當才當的



# BigTable

[Read More](#)

Expert/Master, <http://url.cn/eJjmEp>

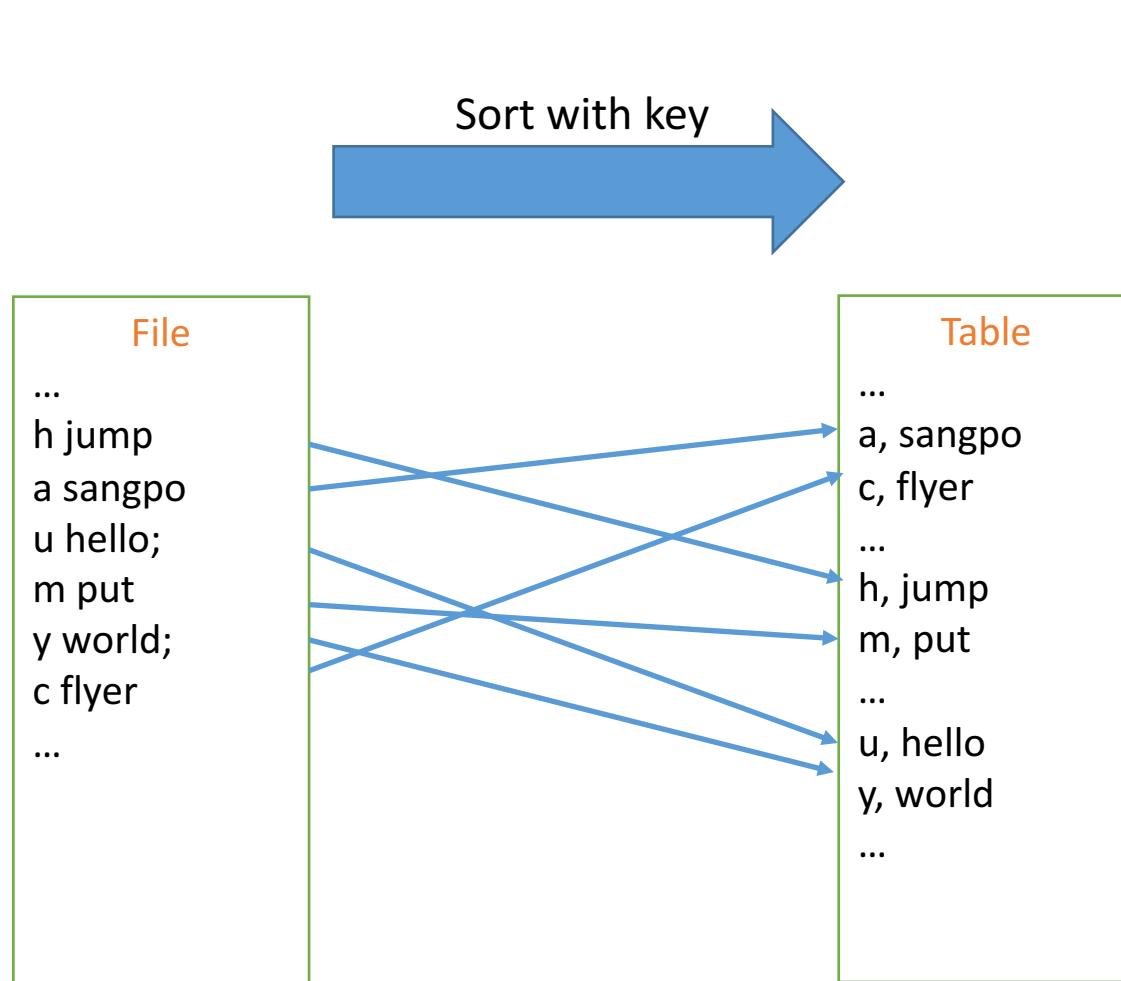
Expert/Master, <http://url.cn/VRwOSW>

Expert/Master, <http://url.cn/W6Xua6>

Expert/Master, <http://url.cn/2aslLW>

Interviewer: How to support lookup  
and range query on a file?

# How to support lookup and range query on a file?



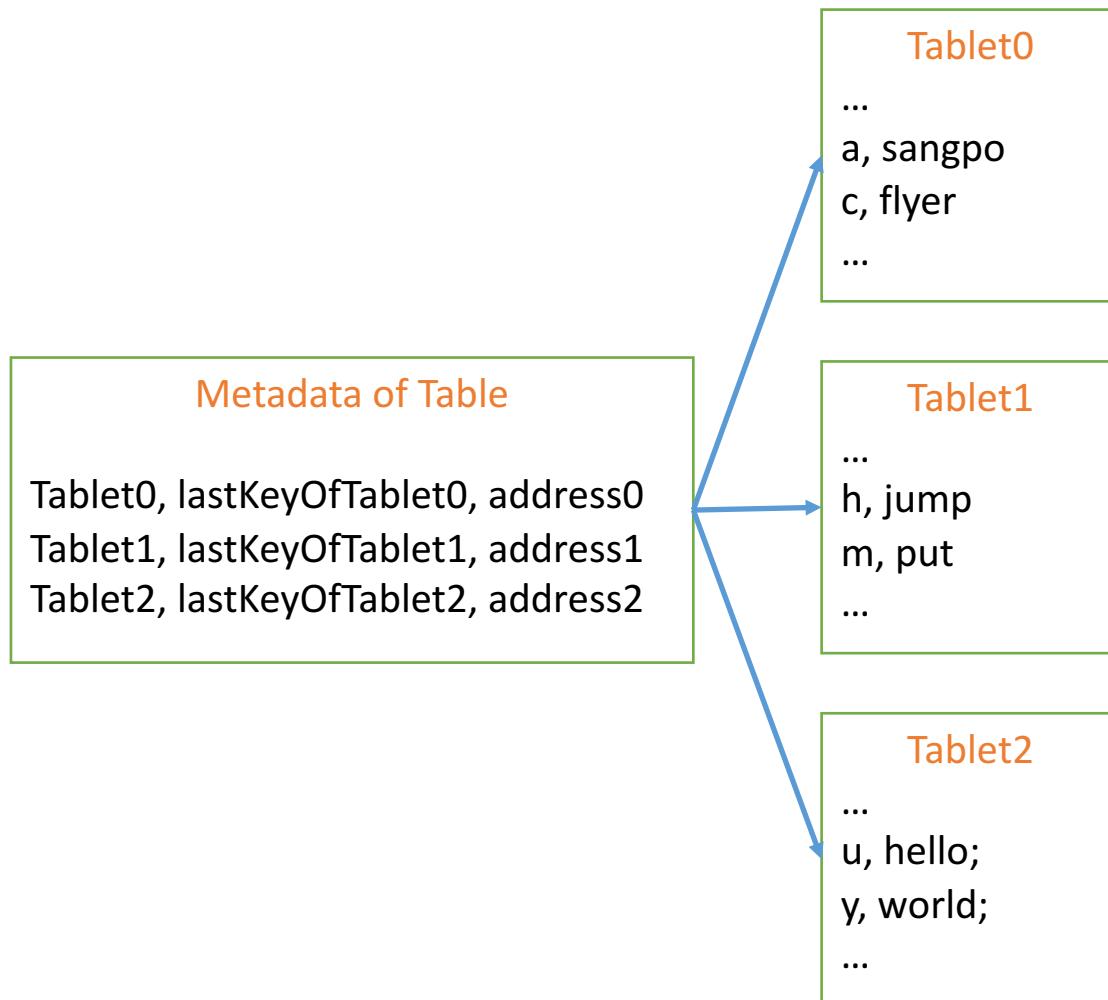
- Key point



- Table = A list of sorted <key, value>

Interviewer: How to save a large table?

# How to save a large table?



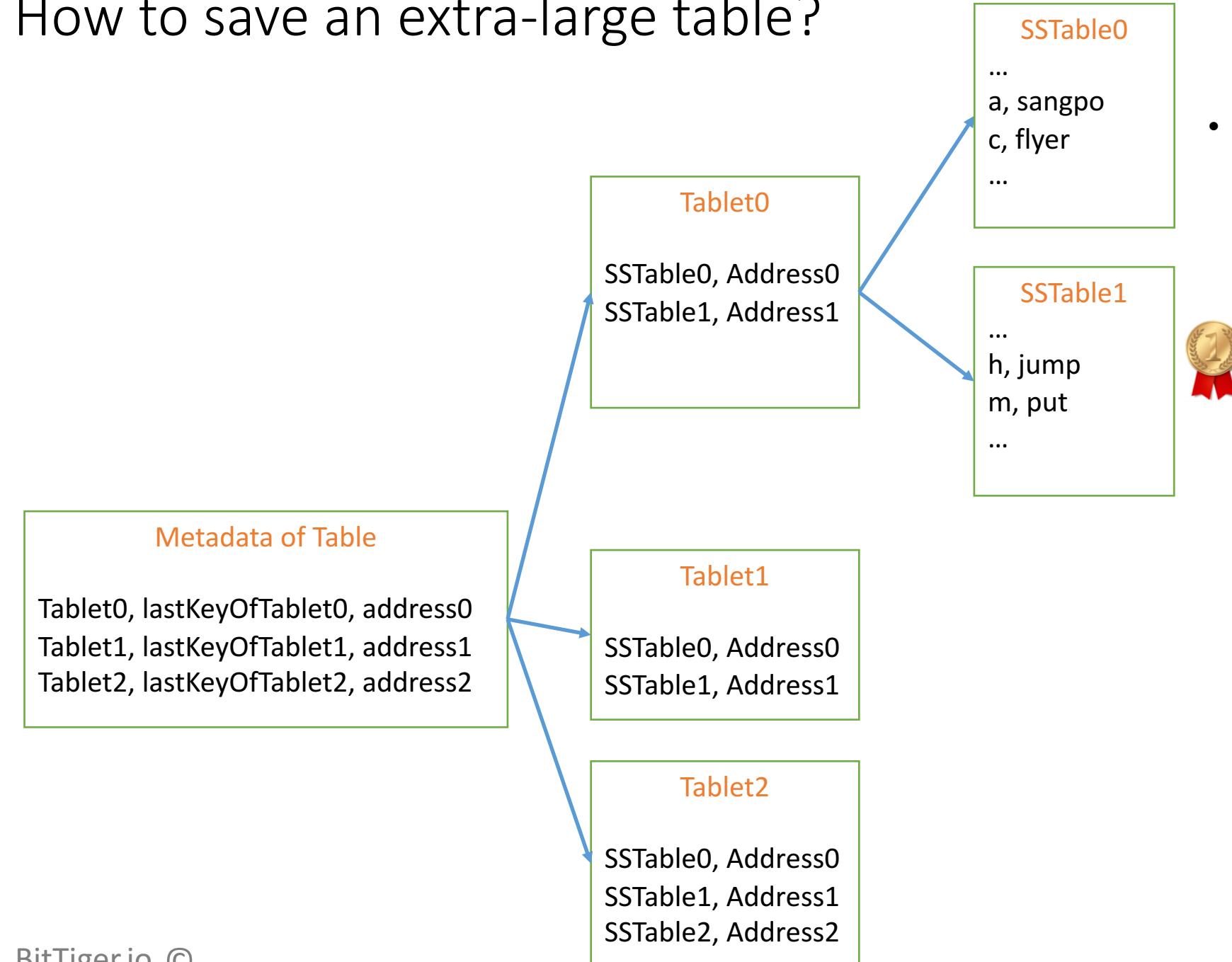
- Key point



- A table = a list of tablets
- A tablet = a list of sorted <key value>

**Interviewer: How to save an extra-large table?**

# How to save an extra-large table?



- Key point

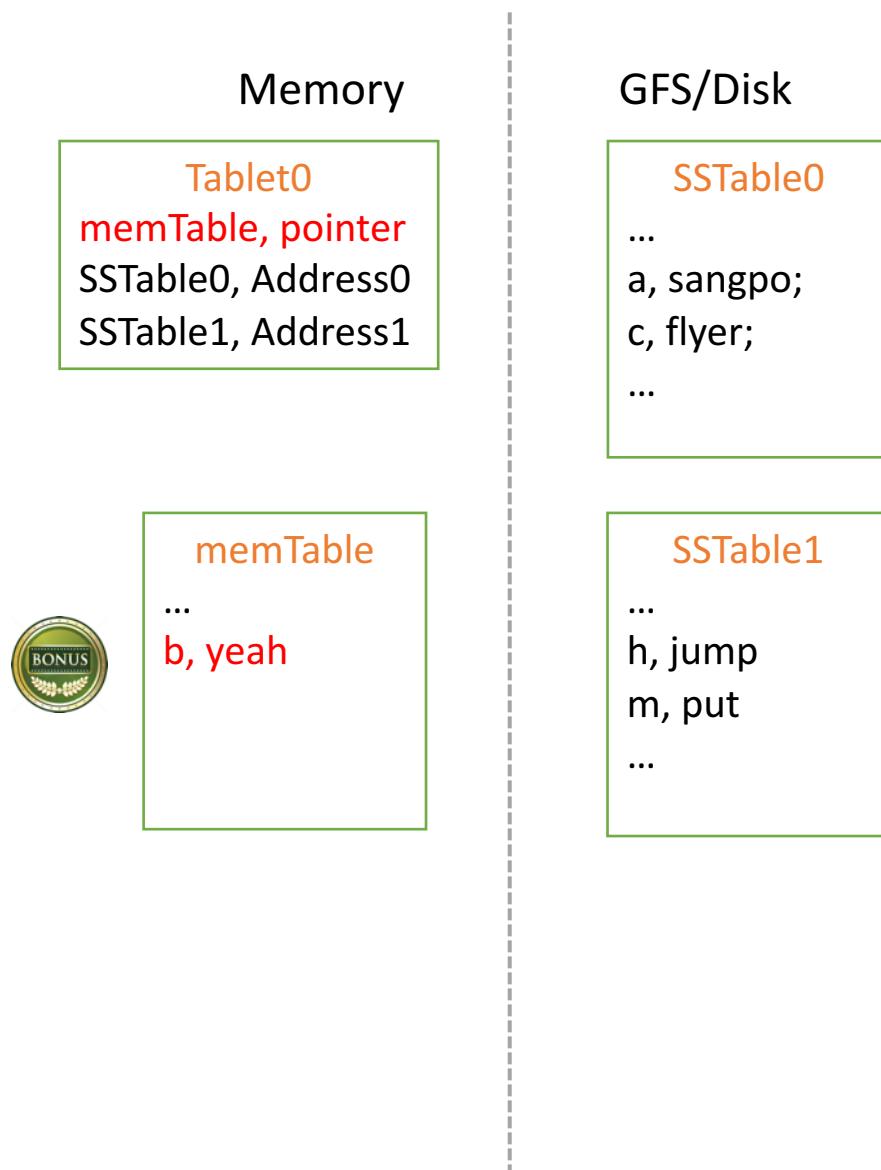
- A table = a list of tablets
- A tablet = a list of SSTables
- A SSTable = a list of sorted <key, value>



**Interviewer: How to write into a table?**

# How to write into a table?

Task: add <b, yeah>

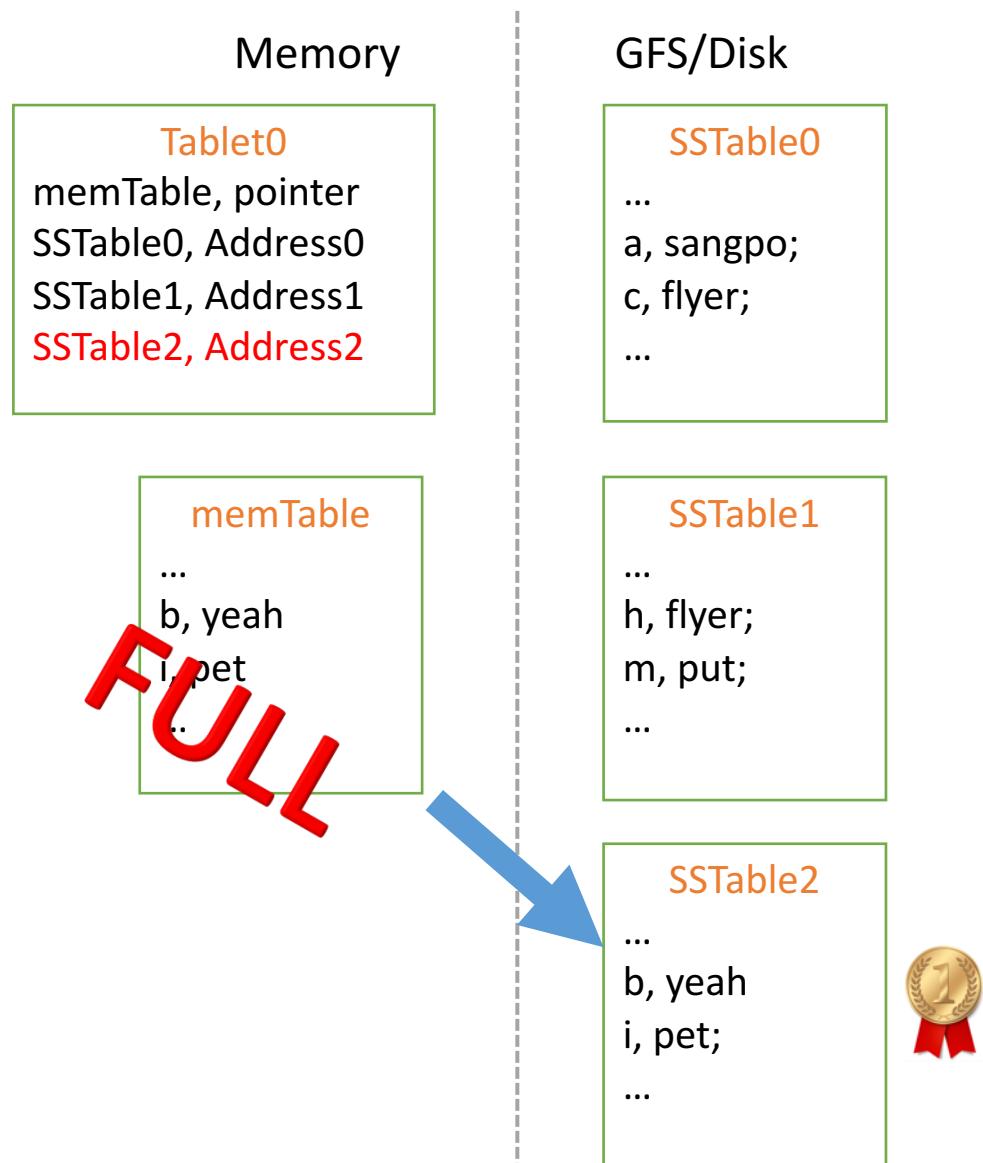


- **Key point**

- Write into a memTable to increase speed
- A tablet = **memTable** + a list of SSTables

Interviewer: What if memTable  
is too large?

# What if memTable is too large?



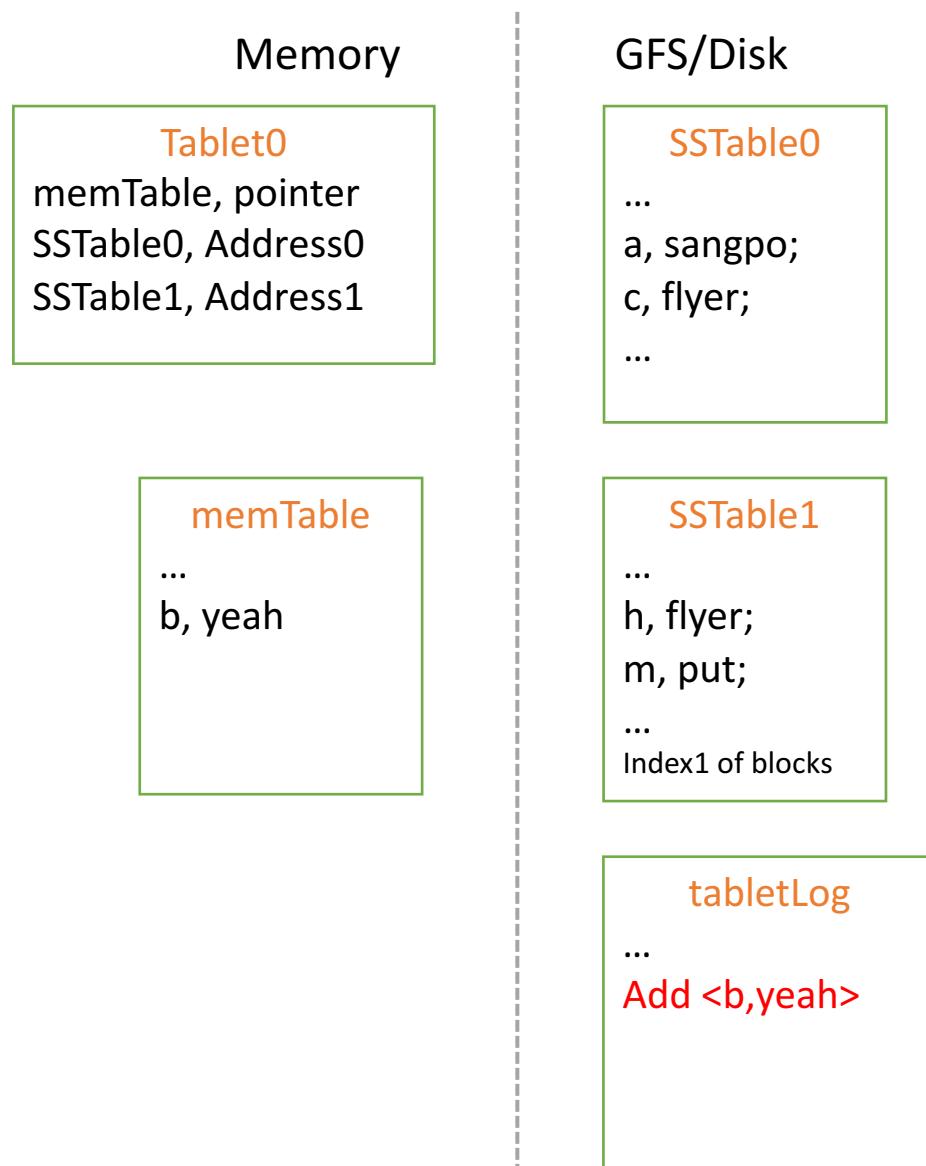
- Key point

- Copy into a new SSTable in GFS/Disk when memTable is full

# Interviewer: How to avoid system failure?

# How to avoid system failure?

Task: add <b, yeah>



- Key point

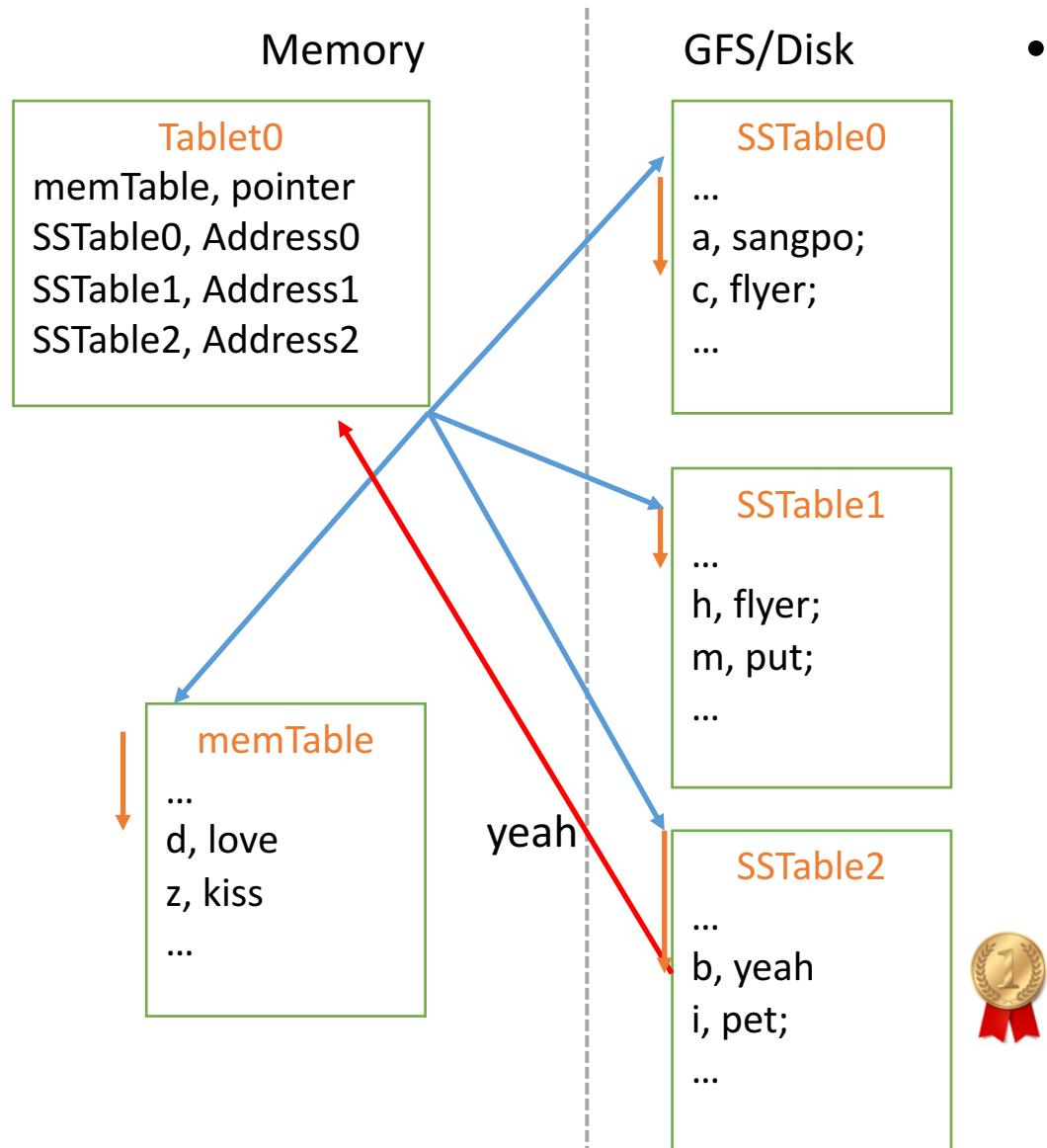
- A tablet = memTable + a list of SSTables + log



# Interviewer: How to read data?

# How to read data?

Task: lookup(b)



- **Key point**

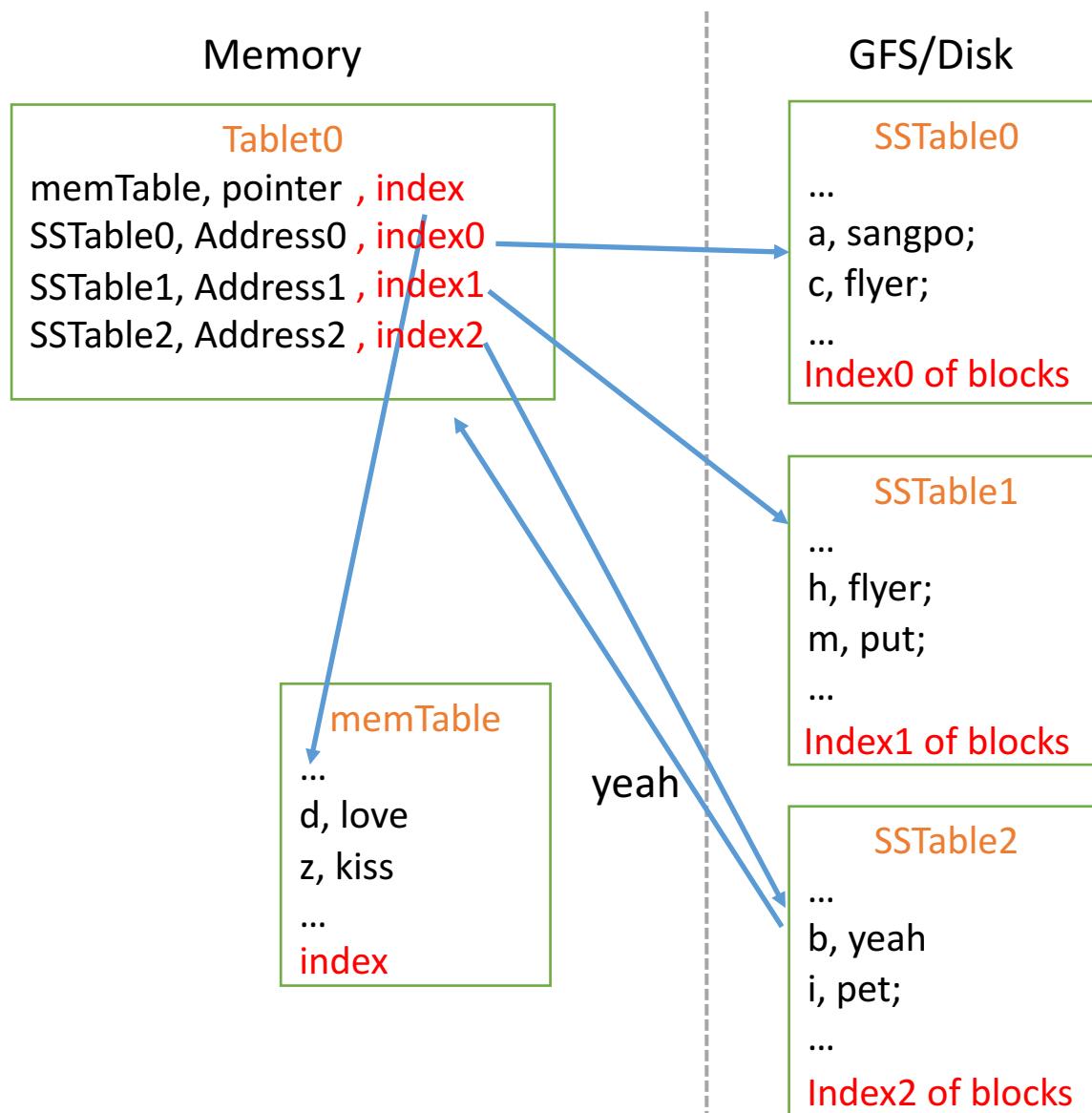
- Due to our write strategy
  - Data is ordered inside a SSTable
  - Data is not ordered among SSTables
- A read need to ask all SSTables and memTable
- Each SSTable needs to use Disk search to find the element



Interviewer: How to read data fast?

# How to read data fast?

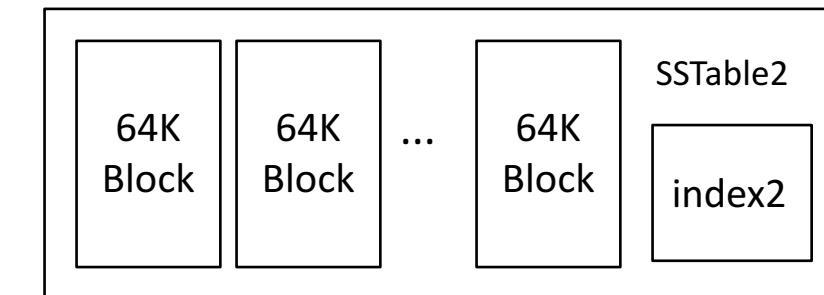
Task: `lookup(b)?`



- Key point

- A SSTable = a list of sorted `<key, value>`  
= a list of 64K blocks + index  

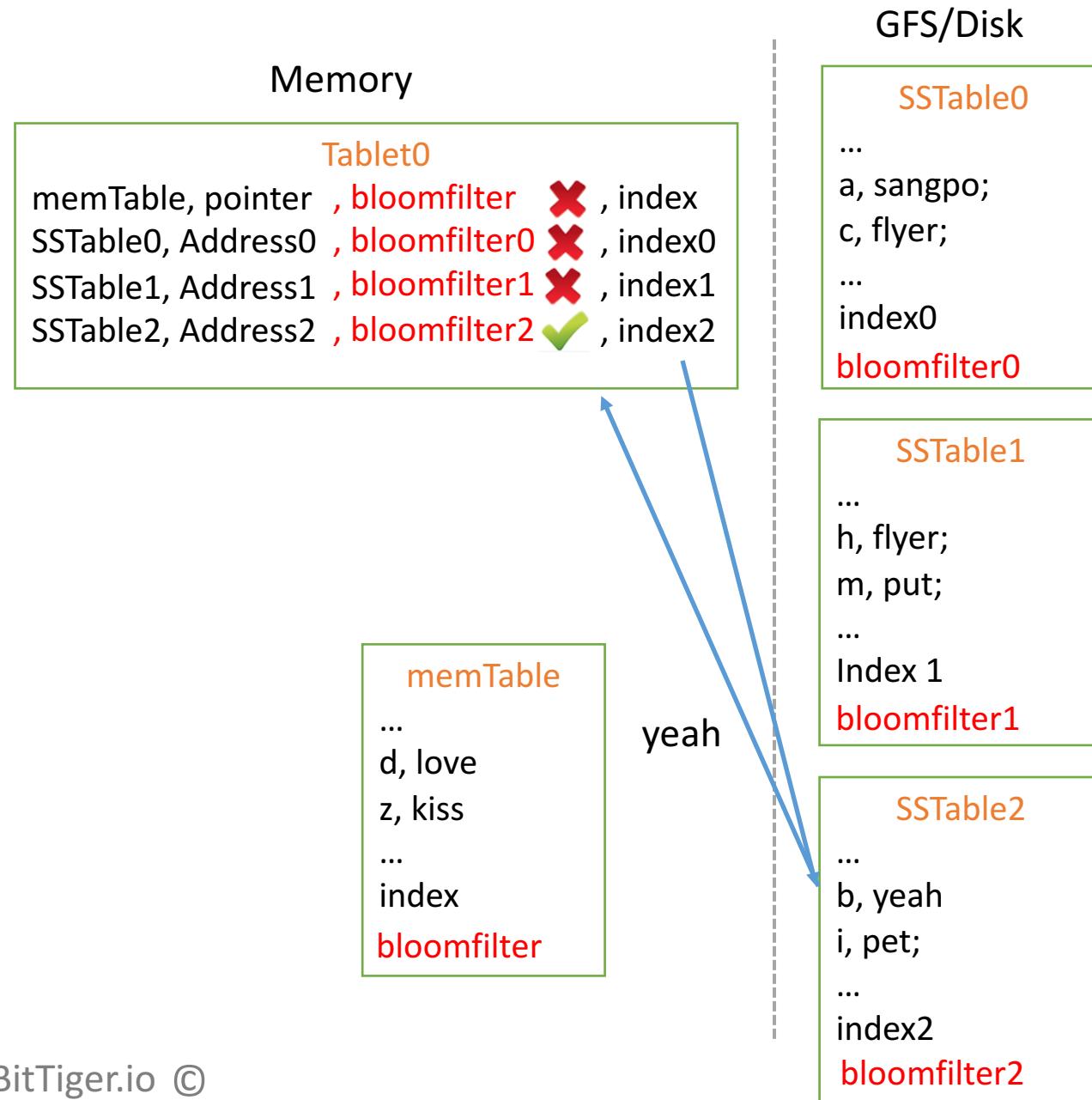
- Indices are pre-loaded into memory
- Read fast by
  - Use indices to identify the place in disk



# Interviewer: Read faster?

# How to read data faster?

Task: `lookup(b)?`

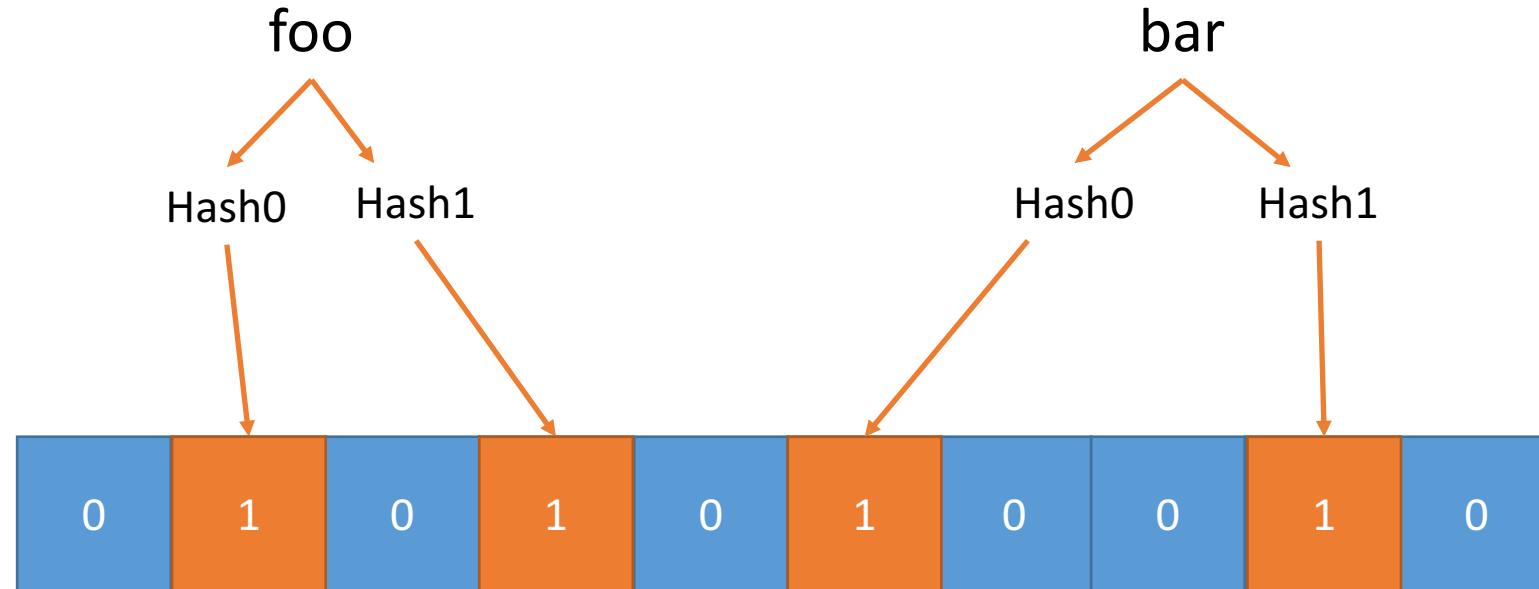


- Key point

- A SSTable = a list of sorted `<key, value>`  
= a list of 64K blocks + index + **bloomfilter**
- Bloomfilters are pre-loaded into memory
- Read faster by
  - Check the existence of the element with bloomfilters



# How to build bloom filter?



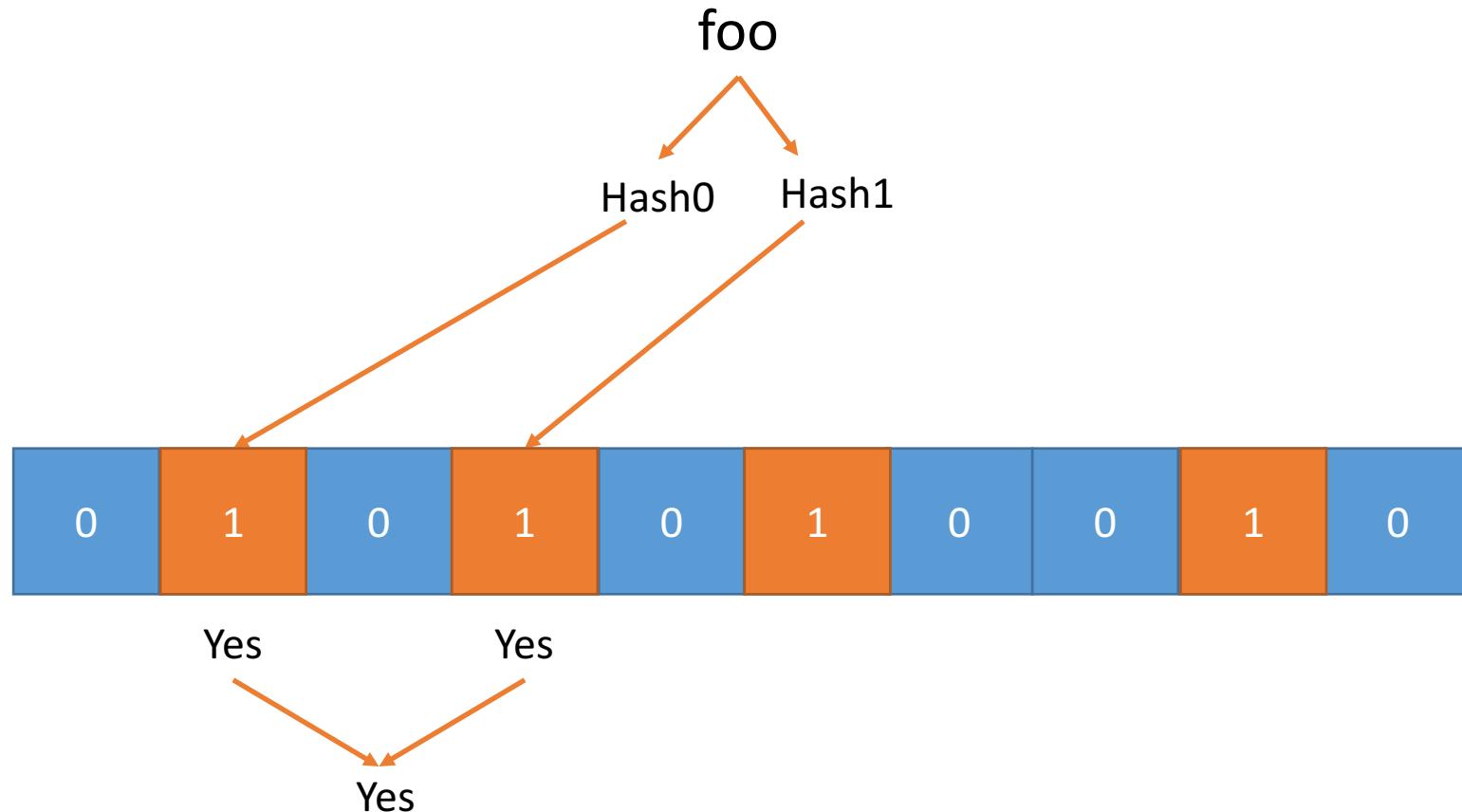
[Read More](#)

Novice, <http://url.cn/Qu6dab>

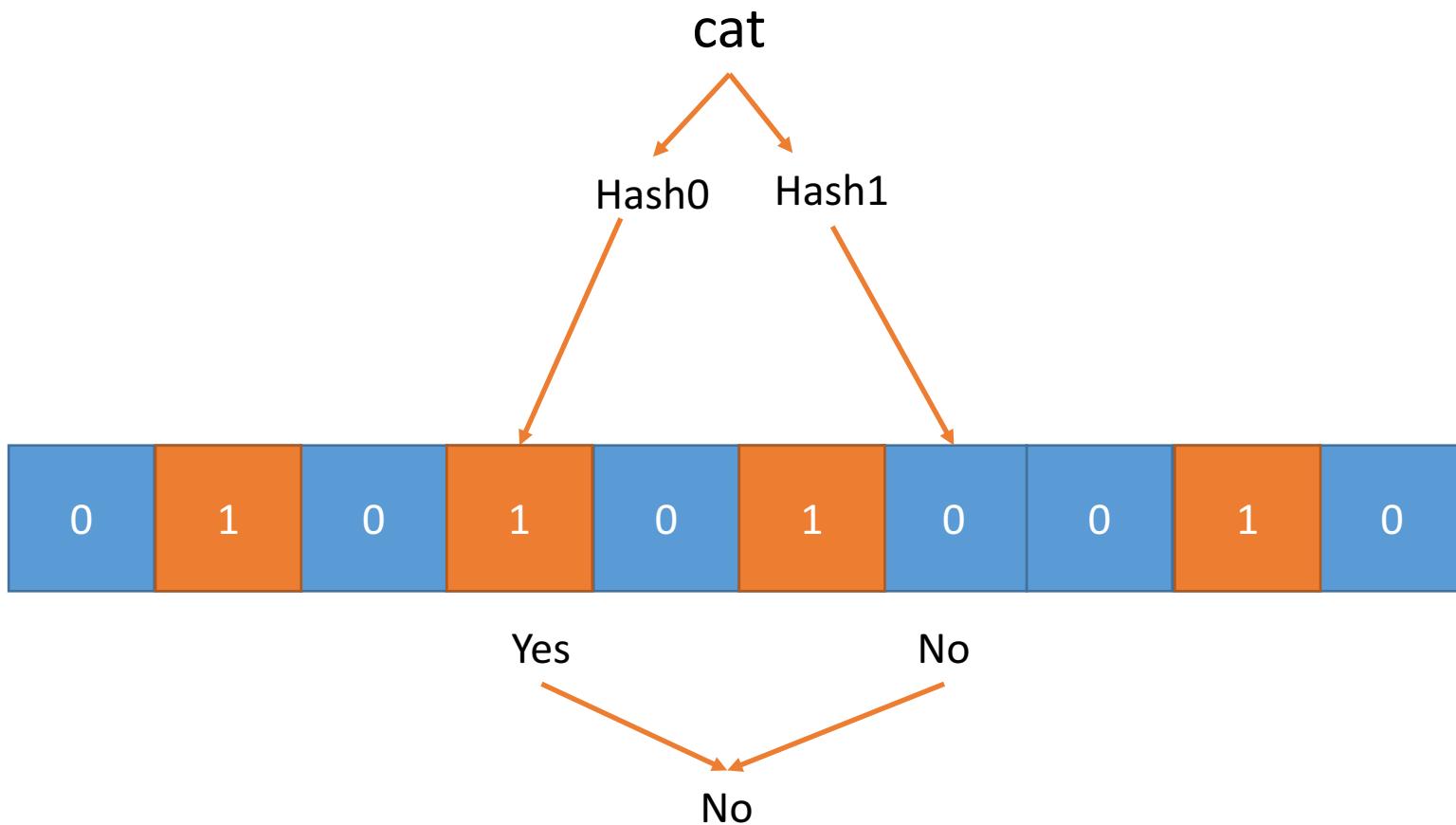
Expert/Master, <http://url.cn/aPCYvo>

Expert/Master, <http://url.cn/TkF35M>

# Lookup “foo”

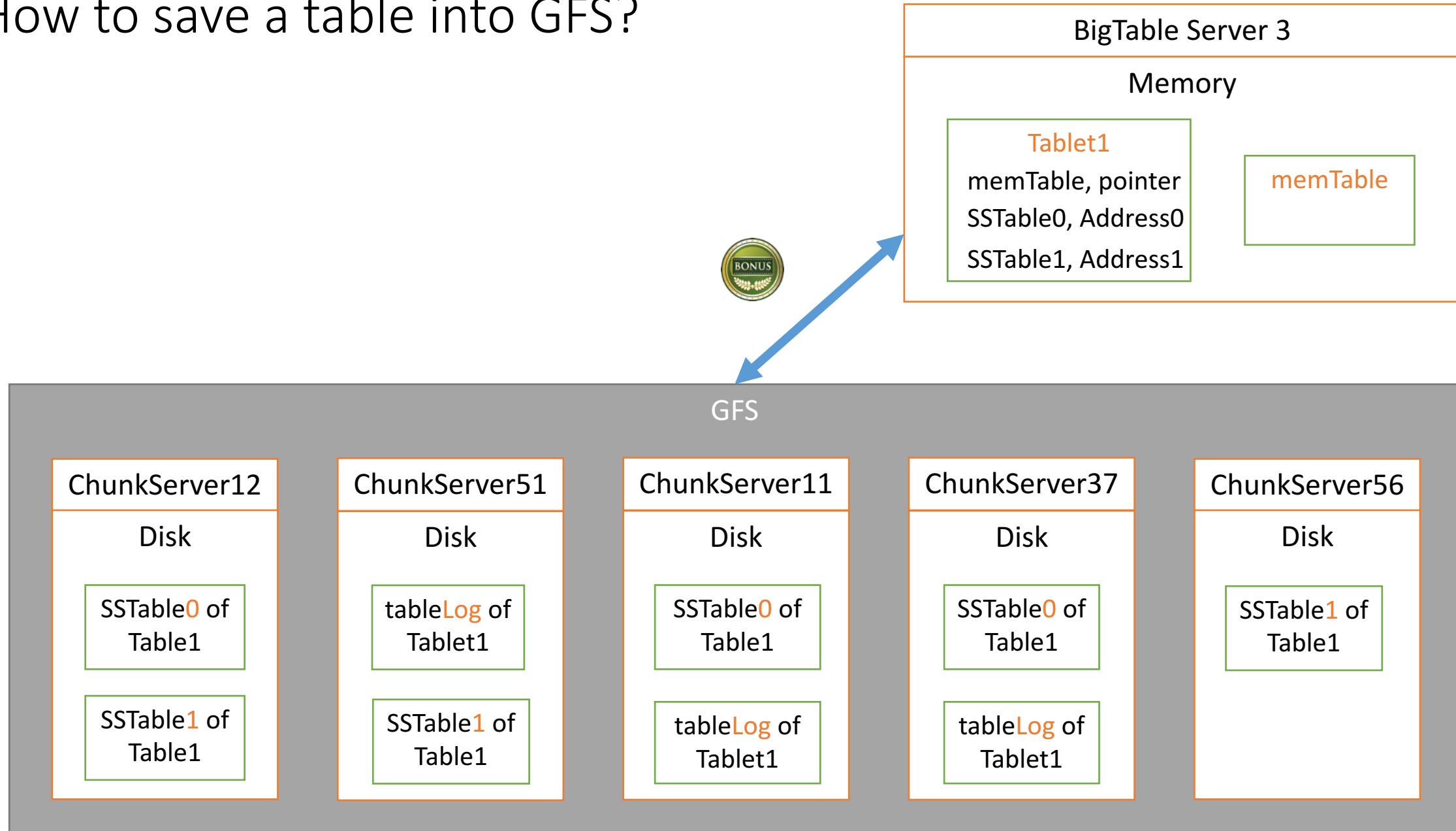


# Lookup “cat”



Interviewer: How to save a table into GFS?

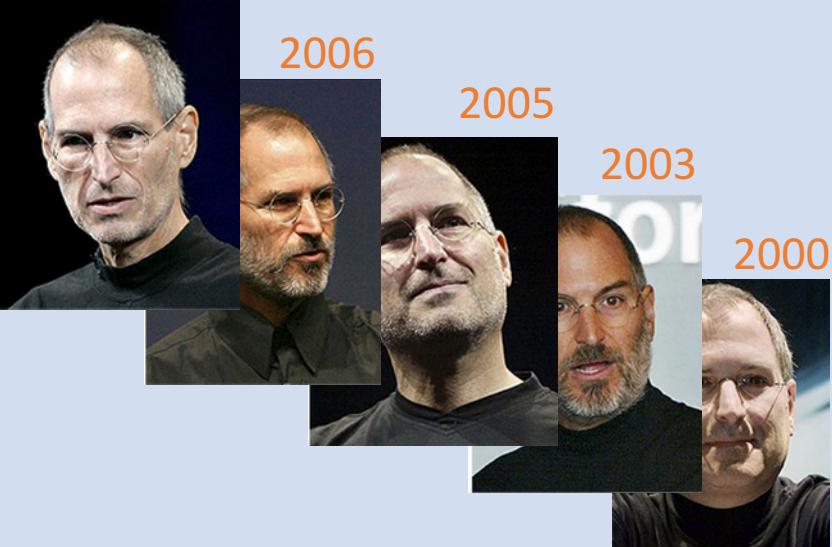
# How to save a table into GFS?



Interviewer: What is the logical view of table?

# What is the logical view of table?



| Name   | Photo   | Body                                      |                          |
|--------|---|---|--------------------------|
|        |   | Weight                                    | Height                   |
| Sangpo |   |   |                          |
| Steve  |  | 110lb, 2009<br>120lb, 2008<br>150lb, 1995 | 6'2", 2011<br>5'7", 1987 |
| Tiger  |   |   |                          |

Interviewer: How to transform logic view into physical storage?

# How to transform logic view into physical storage?

Logic view of Table

| Name   | Photo   | Body                                      |                          |
|--------|---|---|--------------------------|
|        |   | Weight                                    | Height                   |
| Sangpo |   |   |                          |
| Steve  |  | 110lb, 2009<br>120lb, 2008<br>150lb, 1995 | 6'2", 2011<br>5'7", 1987 |
| Tiger  |   |   |                          |

Physical storage of <Key, value>



Table

...

Steve;Body:Height;2011 -> 6'2"

Steve;Body:Height;1987 -> 5'7"

Steve;Photo;2009 -> 2009.JPG

Steve;Photo;2006 -> 2006.JPG

Steve;Photo;2005 -> 2005.JPG

...

Interviewer: What is the architecture of BigTable?

# BigTable architecture



Assign tablets

Master

Process metadata  
& balance load

Tablet Server

Tablet

Tablet

Tablet Server

Tablet

Tablet

read/write

Tablet Server

Tablet

Tablet

Client

Client Library

open()

Serve data from tablets

Cluster Scheduling  
System

Monitor & handles failover

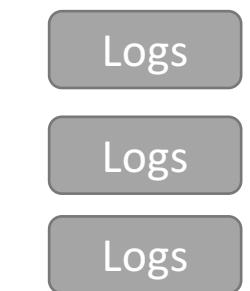
Read More

Expert, <http://url.cn/Td9onD>

GFS



SSTables



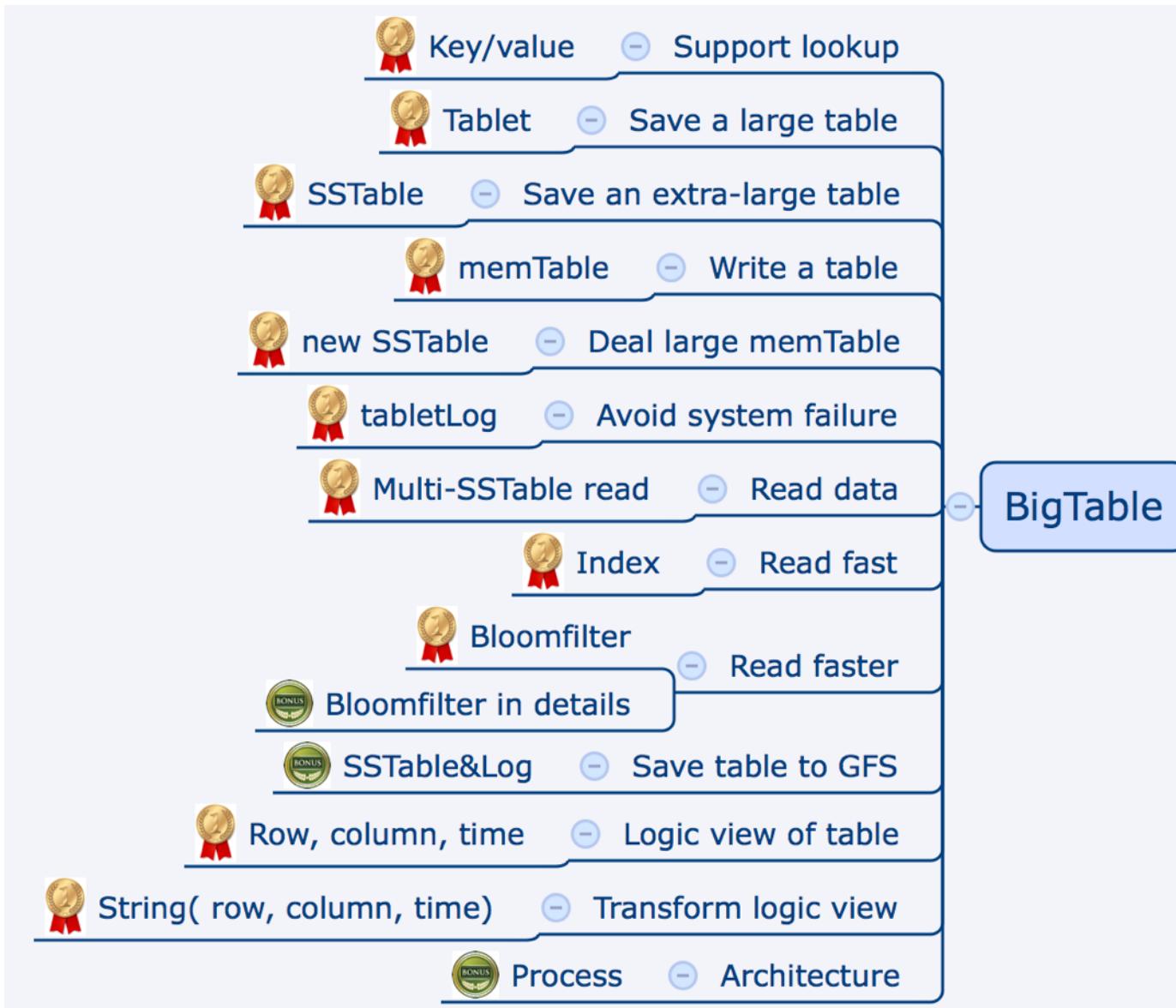
Tablet logs

Replicas

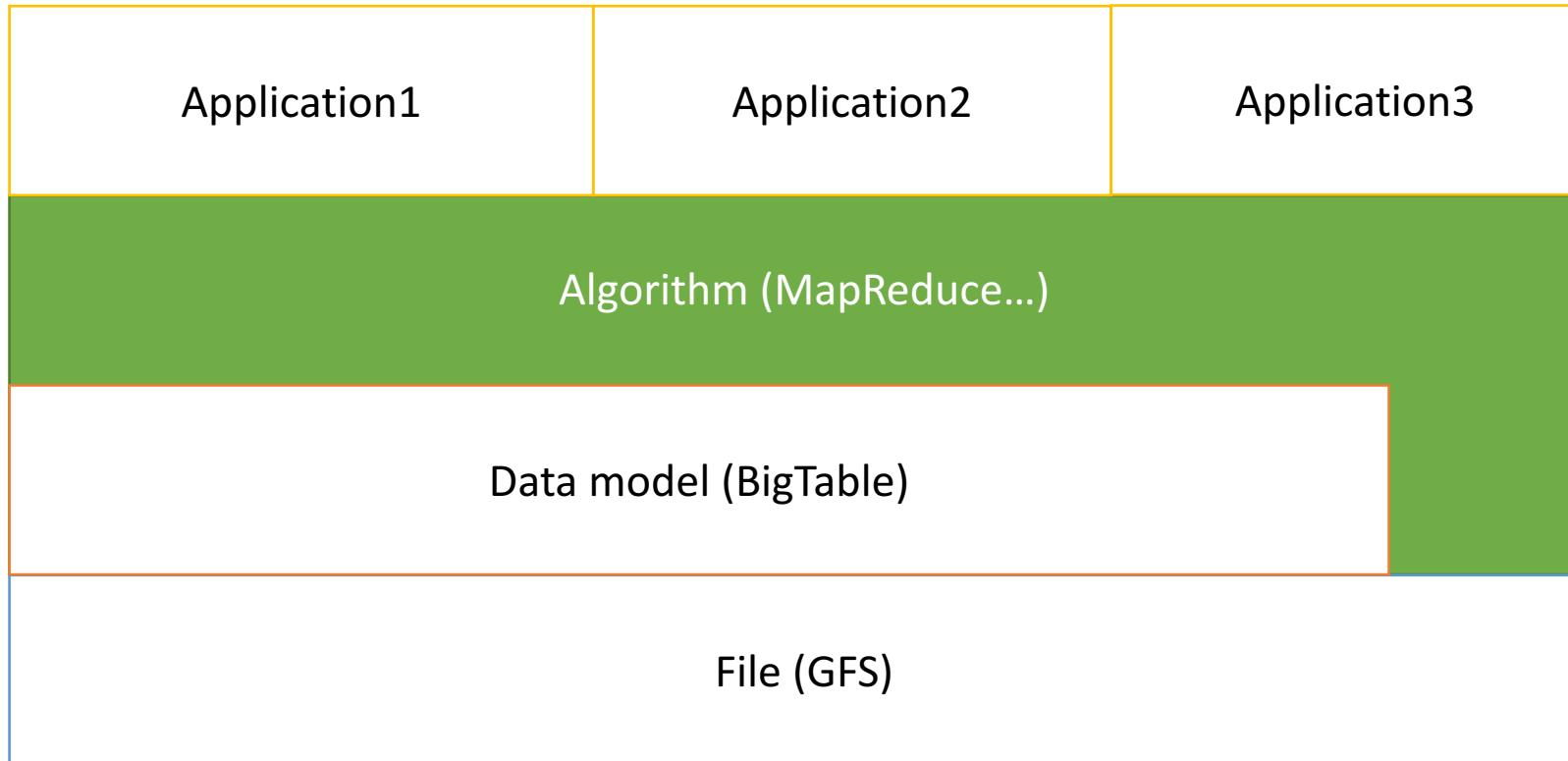
Chubby

Serve table metadata  
& provide lock  
& elect chubby master

# BigTable summary (11+3)



# Layers of system



# MapReduce

[Read More](#)

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>

For 1TB <key, value> data, how to compute

- appearances of words
- inverted index
- anagrams

# Interviewer: What is Map and Reduce?

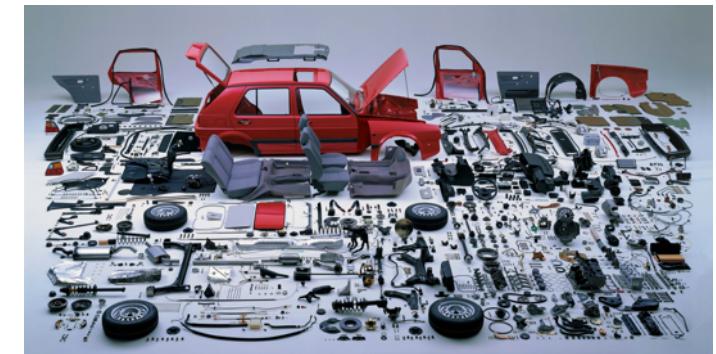
# What is map?



Input



Map



Output

# What is reduce?



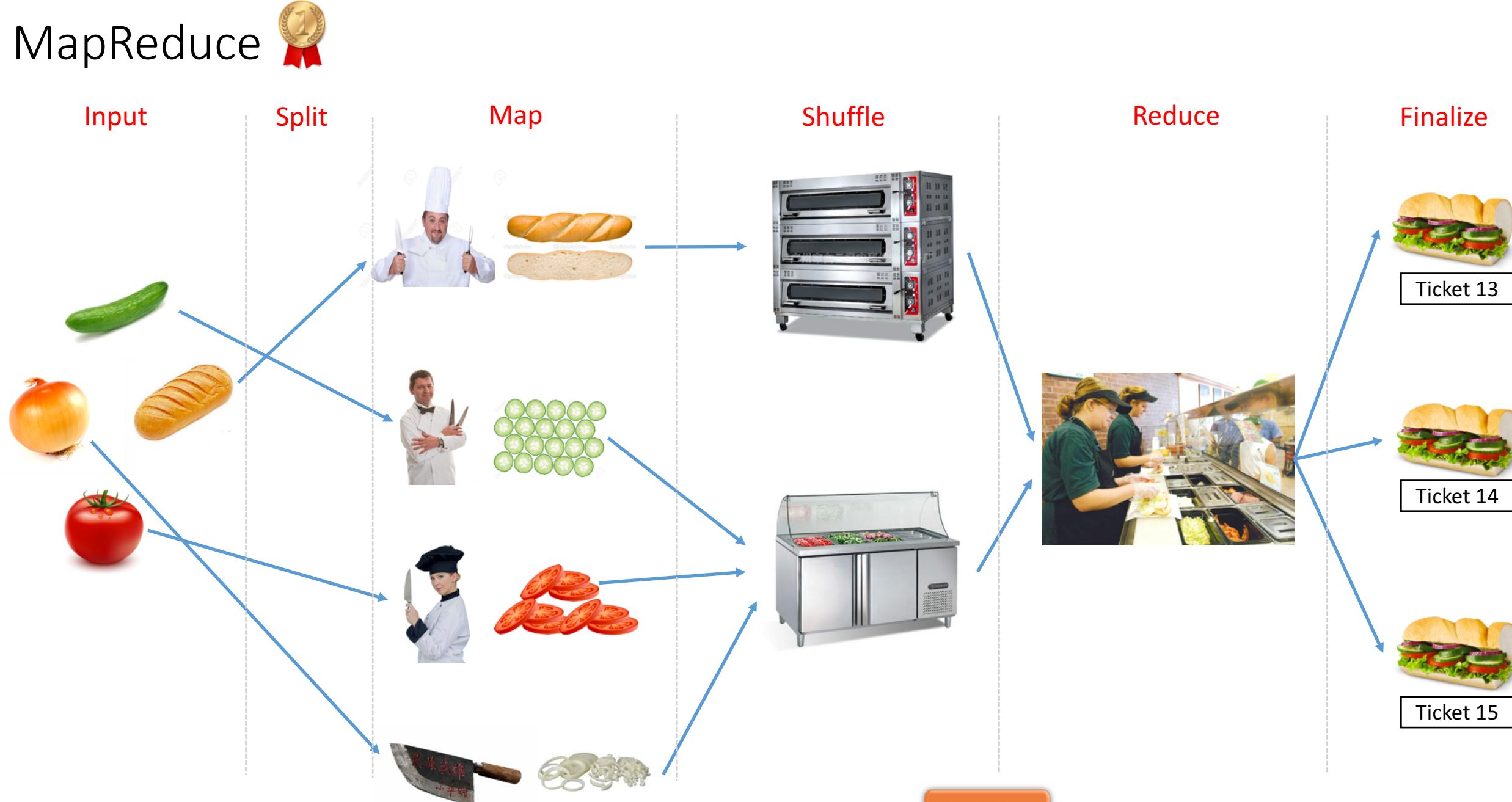
Input



Reduce



Output



Read More

Novice, <http://url.cn/Sd0btu>

# Interviewer: Calculate word appearance with MapReduce

# Calculate each word's number of appearance



Input

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

Split

0: Deer Bear River

1: Car Car River

2: Deer Car Bear

Map

Deer, 1  
Bear, 1  
River, 1

Car, 1  
Car, 1  
River, 1

Deer, 1  
Car, 1  
Bear, 1

Shuffle

Bear, 1  
Bear, 1

Car, 1  
Car, 1  
Car, 1

Deer, 1  
Deer, 1

River, 1  
River, 1

Reduce

Bear, 2

Car, 3

Deer, 2

River, 2

Finalize

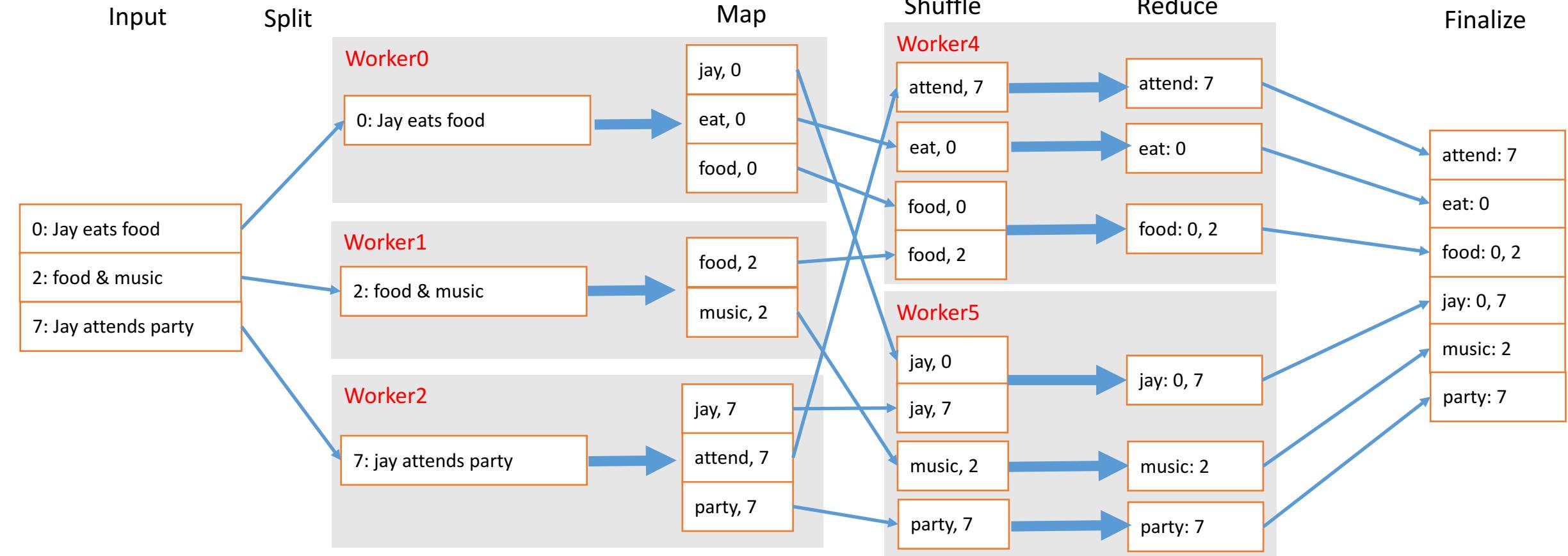
Bear, 2  
Car, 3  
Deer, 2  
River, 2

**Map( string key, string value)**  
 //key: the id of a line  
 //value: the content of the line  
 for each word in value:  
**OutputTemp( word, 1);**

**Reduce( string key, list valueList)**  
 //key: the name of a word  
 //valueList: the appearances of this word  
 int sum = 0;  
 for value in valueList:  
 sum+=value;  
**OutputFinal( key,sum);**

Interviewer: Build inverted index  
with MapReduce?

# Build inverted index



```

Map( string key, string value )
//key: the id of a document
//value: the content of the document
for each word in value:
    OutputTemp( word, key );

```

[Read More](#)

```

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

```

Interviewer: Calculate anagrams  
with MapReduce?

Given: ["bit", "ibt", "tiger", "regit"],

Return:

[ ["bit", "ibt"], ["tiger", "regit"] ]

Anagram(bit) = { <b,1>, <i,1>, <t,1> }

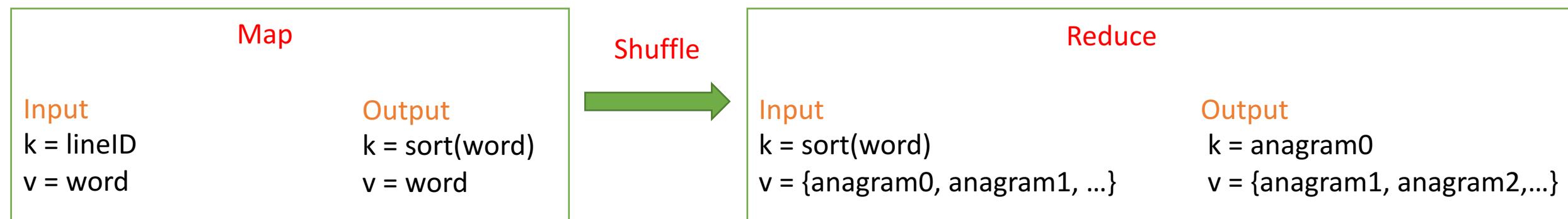
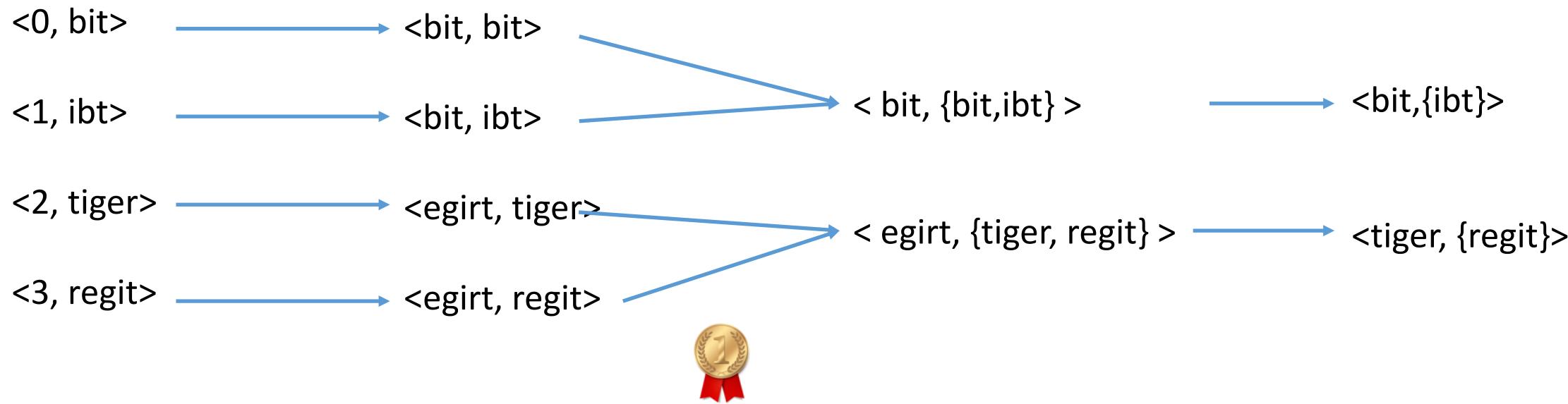
Anagram(ibt) = { <b,1>, <i,1>, <t,1> }

Anagram(tiger) = { <e,1>, <g,1>, <i,1>, <r,1>, <t,1> }

Anagram(regit) = { <e,1>, <g,1>, <i,1>, <r,1>, <t,1> }

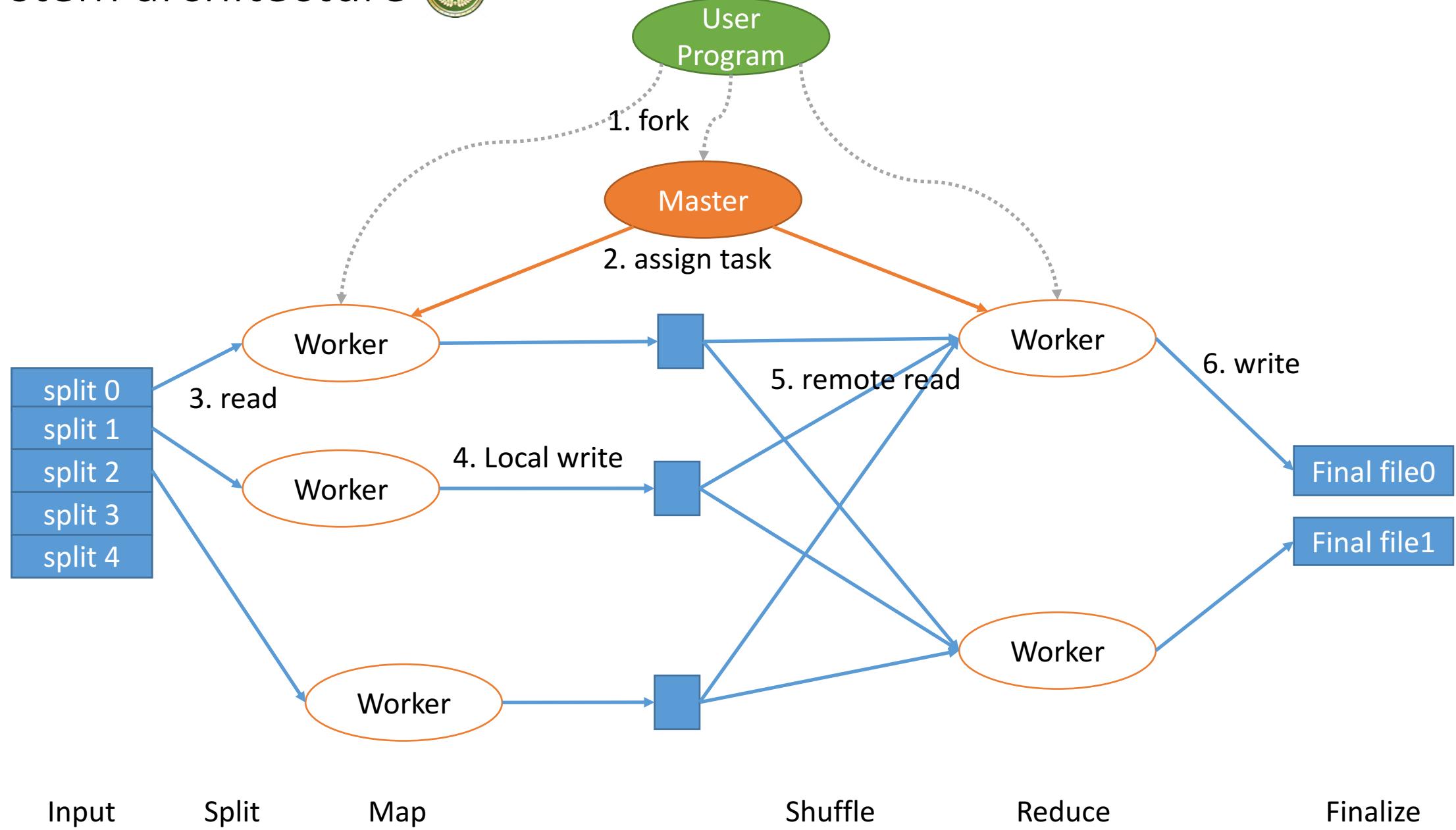
[Read More](#)

Novice/Expert, <http://url.cn/cGRS1I>

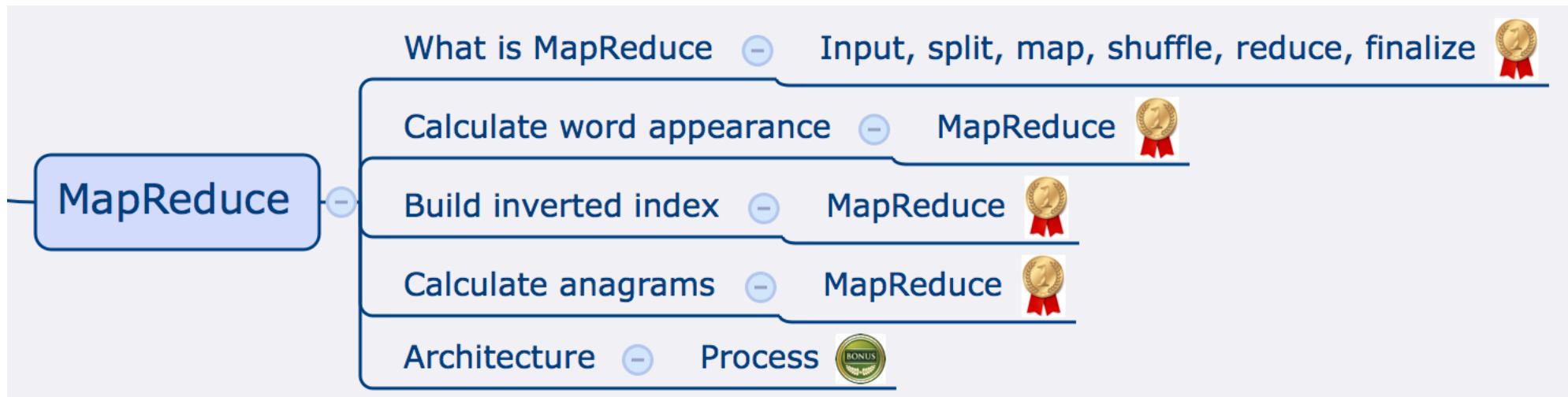


**Interviewer: What is the  
architecture of MapReduce?**

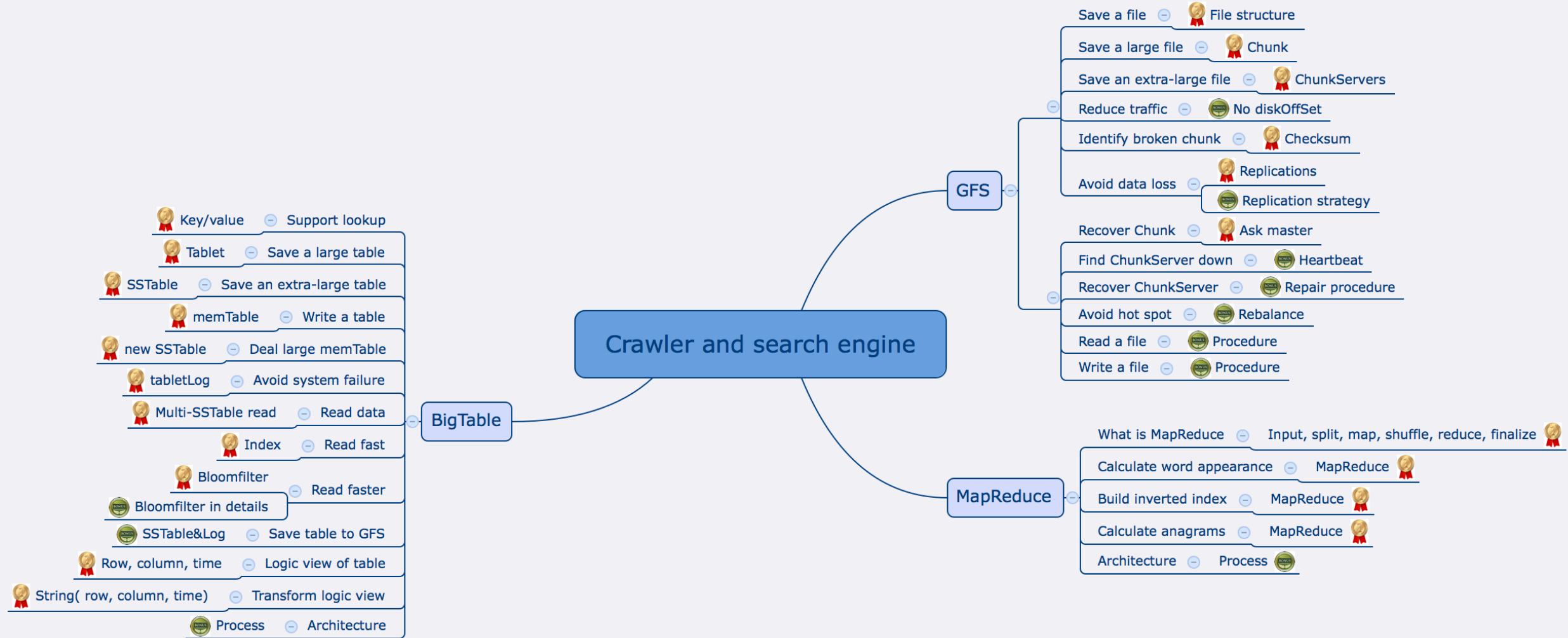
# System architecture



# MapReduce summary (4+1)



# Summary



## MR. Problem

Map is **disassemble**;

Reduce is **assemble**.

Our life is a sequence of  
**assemble** and **disassemble**.

You may wonder why!

Is it true?

But how to solve my job interview,  
such a real problem?

**Disassemble** MR. problem again and again,  
Until you can **assemble** the courage to face it.

Wish you, my friend,  
**Assemble** all the pieces to win your job, hohoho!