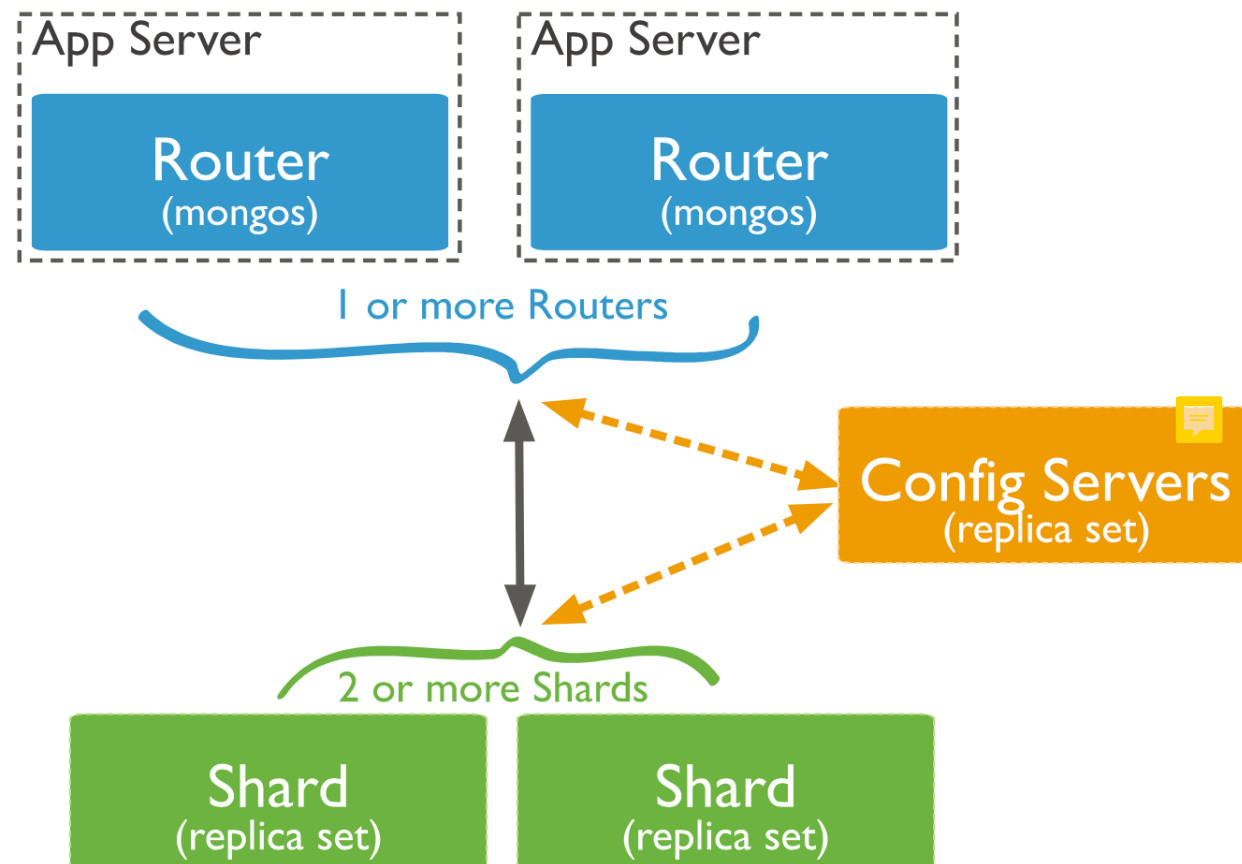


SHARDING - 分片

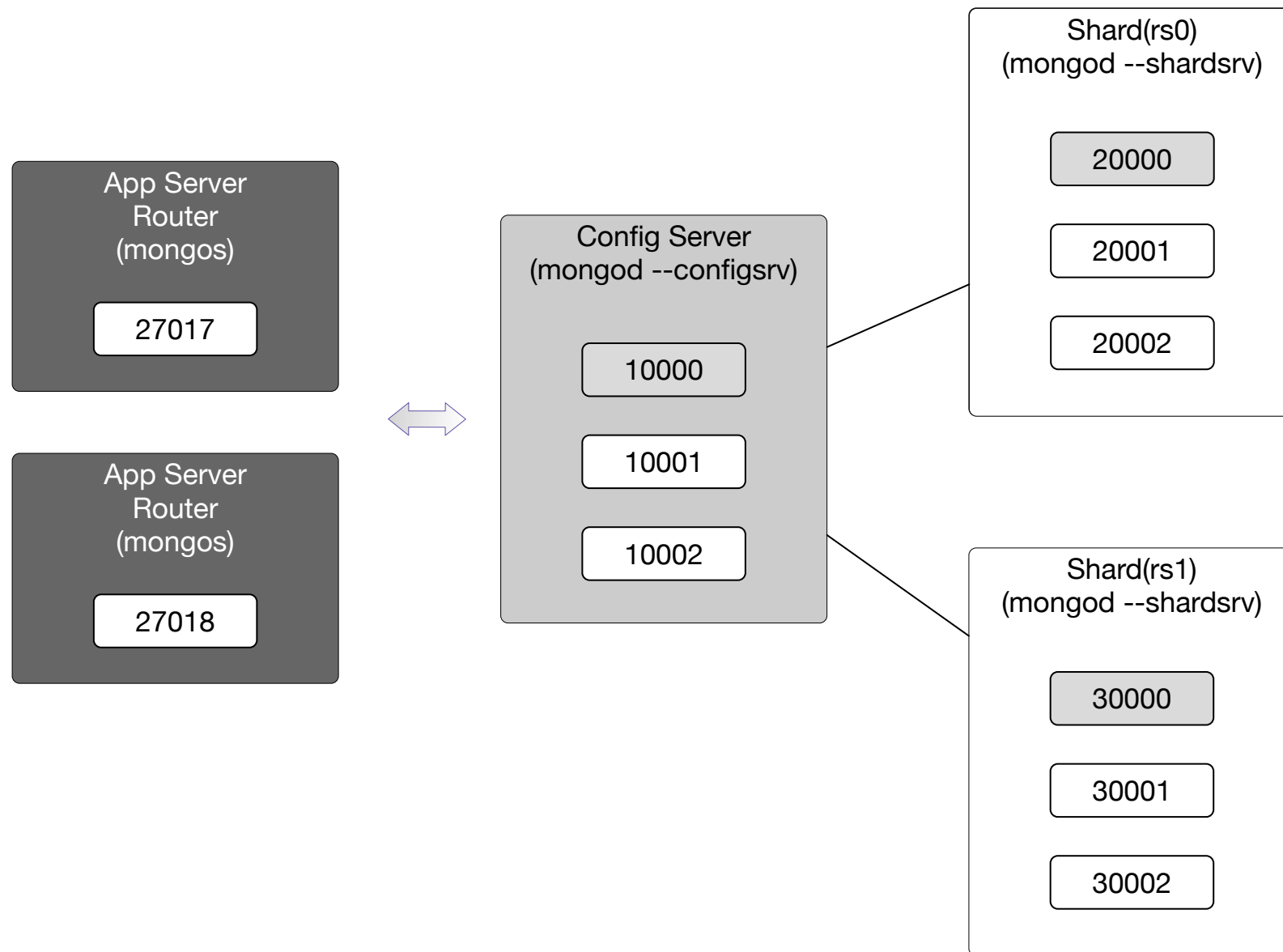
朱克剛

架構

將資料分散到別的主機 (Shard)
Shard與Config都必須使用覆寫集架構



實作



第 1 步：設定SHARD主機RS0

```
$ mongod --port 20000 --dbpath data/rs0/0 --replSet rs0 --shardsvr
```

```
$ mongod --port 20001 --dbpath data/rs0/1 --replSet rs0 --shardsvr
```

```
$ mongod --port 20002 --dbpath data/rs0/2 --replSet rs0 --shardsvr
```

都啟動後使用 mongo 連進 port 20000 那一台設定覆寫集，請參考「覆寫」投影片

第2步：設定SHARD主機RS1

```
$ mongod --port 30000 --dbpath data/rs1/0 --replSet rs1 --shardsvr
```

```
$ mongod --port 30001 --dbpath data/rs1/1 --replSet rs1 --shardsvr
```

```
$ mongod --port 30002 --dbpath data/rs1/2 --replSet rs1 --shardsvr
```

都啟動後使用 `mongo` 連進 `port 30000` 那一台設定覆寫集，請參考「覆寫」投影片

第3步：設定CONFIG主機

```
$ mongod --port 10000 --dbpath data/cfg/0 --replSet cfg --configsvr
```

```
$ mongod --port 10001 --dbpath data/cfg/1 --replSet cfg --configsvr
```

```
$ mongod --port 10002 --dbpath data/cfg/2 --replSet cfg --configsvr
```

都啟動後使用 `mongo` 連進 `port 10000` 那一台設定覆寫集，請參考「覆寫」投影片

第4步：設定ROUTER主機

```
$ mongos --port 27017 --configdb cfg/localhost:10000
```

```
$ mongos --port 27018 --configdb cfg/localhost:10000
```

第5步：在CONFIG中加入SHARD主機

1. 使用 `mongo` 連進任何一台 Router 主機
2. 將 Shard 主機中的 PRIMARY 加到 Config 主機。只要 PRIMARY 即可，Configure 會自動加入覆寫集中的其他節點。

```
sh.addShard("rs0/localhost:20000")  
sh.addShard("rs1/localhost:30000")
```

3. 若要再增加 Shard (例如 rs2, rs3, ...)，再執行第上一步指令即可

第6步：測試看看

使用 mongo 或是 python...等連進 Router 主機，下任何查詢、新增、修改、刪除等指令看看

```
use opendata  
db.AQI.find()
```

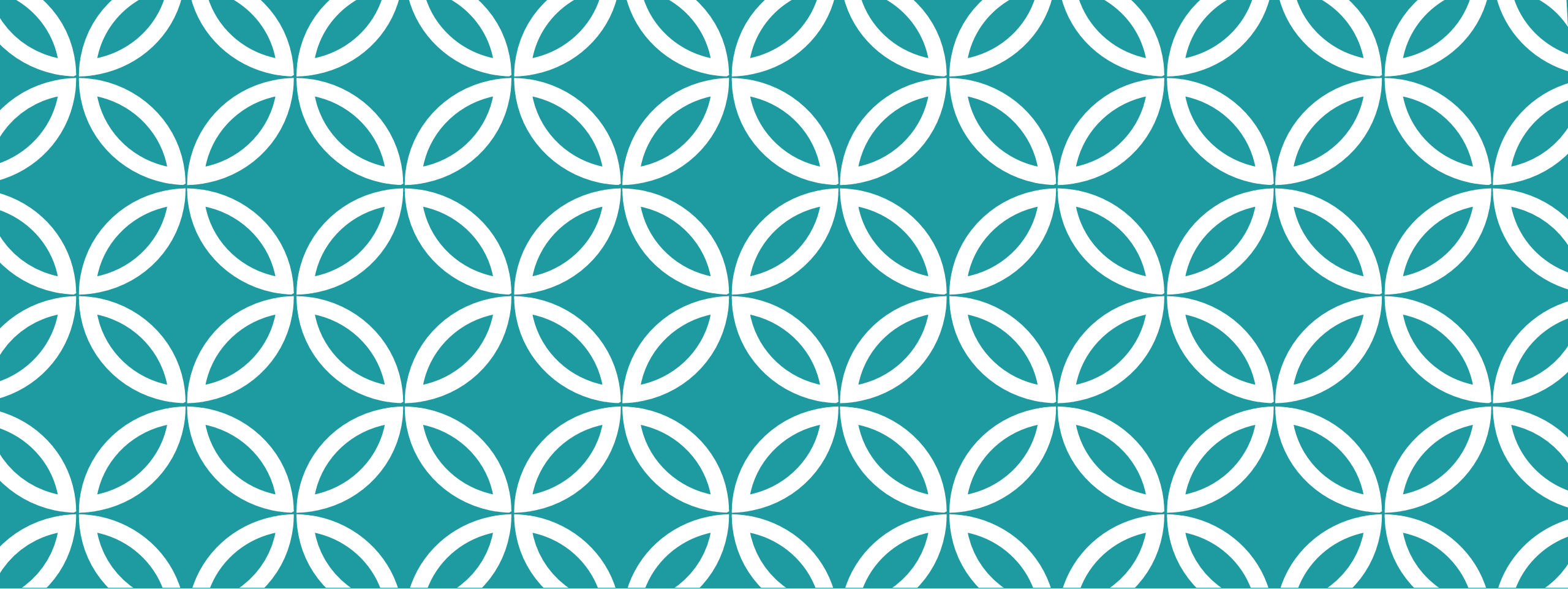


: D

分片了嗎？

當然還沒！

現在資料只會存到 Shard(rs0)，不論資料量再多都不會存到 Shard(rs1)，因為我們還沒設定片鍵，所以 MongoDB 不知道如何分片。



片鍵選擇與設定

設定CHUNK大小

use config

為了立即看到分片
效果，設定2MB

```
db.settings.insertOne( { _id: 'chunksize', value: 2 } )
```

- sizeInMB: 1~1024 MB
- 預設為 64MB (MongoDB 6 128MB)

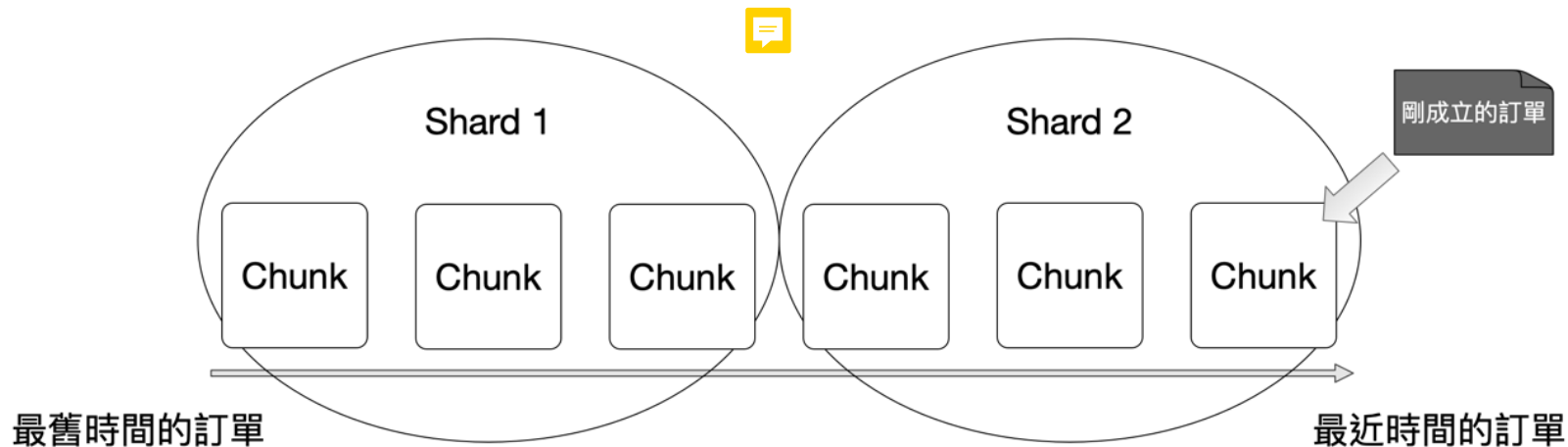
第1步：增加資料 shard_simple.py

```
import pymongo

client = pymongo.MongoClient(['localhost:27017', 'localhost:27018'])
db = client.test

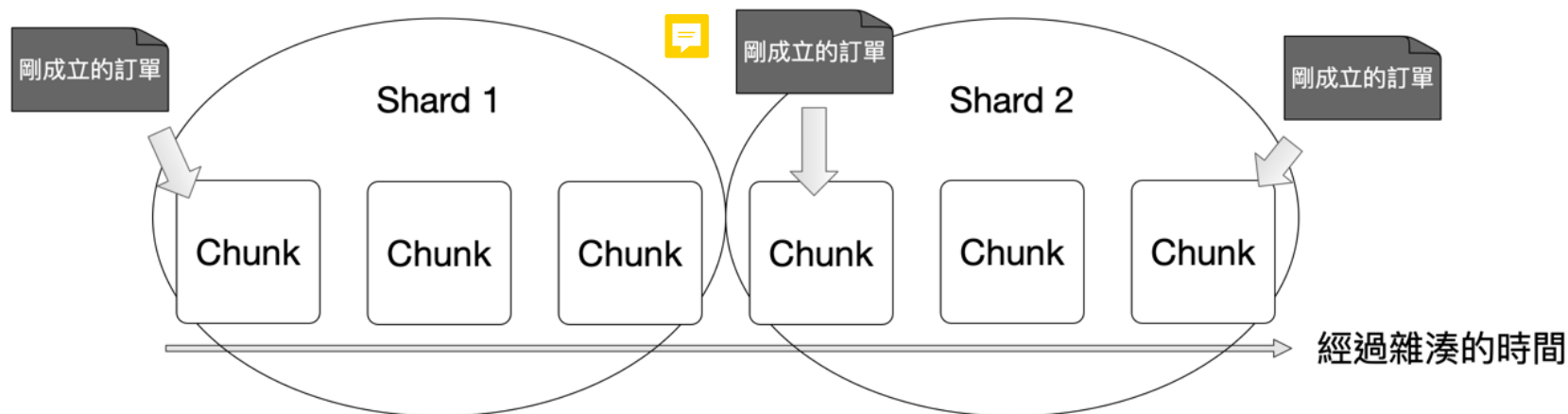
num = 480000
data = [13, 2, 5, 7, 7, 9, 9, 13, 14, 17, 13, 2, 9, 10]
for n in data:
    db.d.insert_one(
        {
            'lv': None,
            'n': n,
            'junk': '_' * num
        }
    )
```

片鍵選擇



片鍵必須是索引，所以建立片鍵前要先建立索引


片鍵不能亂選



好的片鍵

資料有效分散

最常用的查詢條件需在片鍵中，包含索引前綴

- 例如片鍵為 { a: 1, b: 1, c: 1 } 時，查詢條件為
- { a: <value>, b: <value>, c: <value> } 
- { a: <value>, b: <value> }
- { a: <value> }

思考：預設的 _id 適不適合拿來做片鍵？為什麼？ 

第2步：建立索引

設定片鍵的欄位一定要為索引，資料按照索引排序後放到 chunk 中，然後將所有的chunks平均分配到每個 Shard。這裡設定 n 為索引。

```
db.d.createIndex({n: 1})
```

```
data = [13, 2, 5, 7, 7, 9, 9, 13, 14, 17, 13, 2, 9, 10]
```



```
data = [2, 2, 5, 7, 7, 9, 9, 9, 10, 13, 13, 13, 14, 17]
```


第3步：設定片鍵

```
sh.enableSharding("test")  
sh.shardCollection("test.d", {n:1})
```

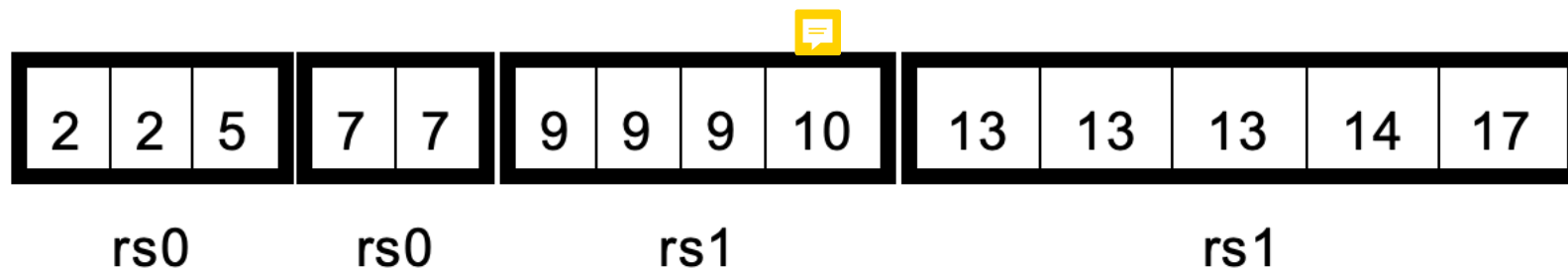
第4步：檢查看看

檢查看看，若資料超過 `chunk` 大小，`mongodb` 會自動建立第二個 `chunk`，並且自動平衡每一個 `shard` 中的 `chunk` 數量

使用 `sh.status()` 查看分片狀況

列出完整資訊

- `mongos> sh.status('verbose')`



JUMBO



執行 `shard_jumbo.py`

重選片鍵

此指令為同步執行，因此需等控制權回來才代表分片結束

```
test> db.adminCommand({
  reshardCollection: 'test.d',
  key: { 'n': 1 }
})
```

查詢剩餘時間

```
test> db.getSiblingDB("admin").aggregate([
  { $currentOp: { allUsers: true, localOps: false } },
  {
    $match: {
      type: "op",
      "originatingCommand.reshardCollection": "test.d"
    }
  }
])
```