Troubleshooting Redis

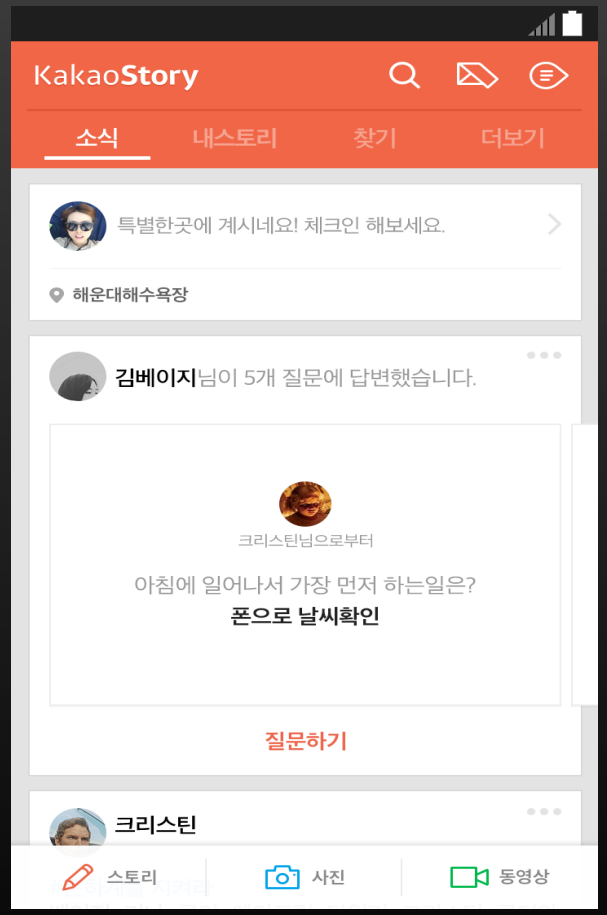
@charsyam

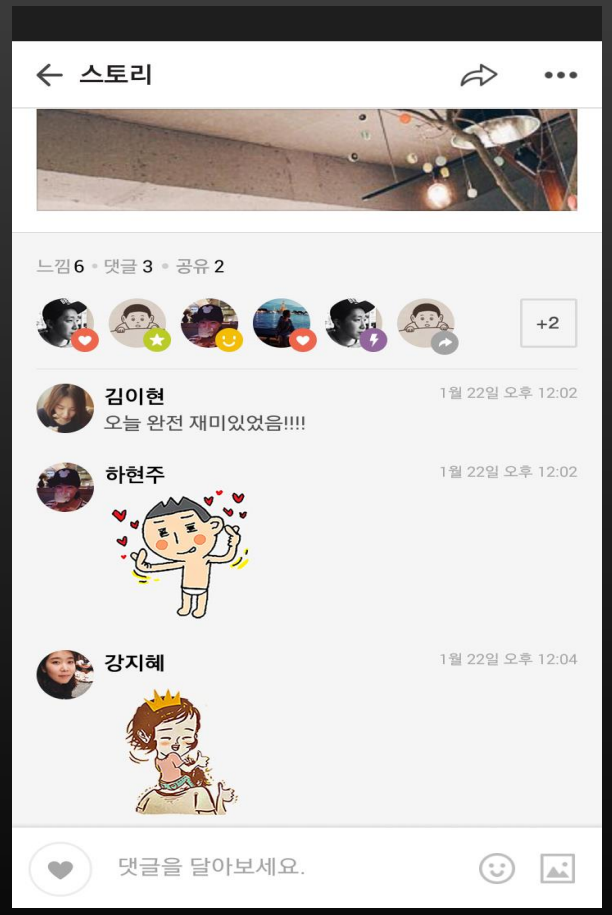
KAKAO

About me

- Senior Software Engineer in KAKAO
- Redis/Twemproxy contributor
- Redis-doc project merger.
- Apache Tajo commiter

kakaostory


KakaoStory
소식
내스토리
찾기
더보기
특별한곳에 계시네요! 체크인 해보세요.
해운대해수욕장
김베이지님이 5개 질문에 답변했습니다.
크리스틴님으로부터
아침에 일어나서 가장 먼저 하는일은?
폰으로 날씨확인
질문하기
크리스틴
스토리
사진
동영상


스토리
느낌 6 • 댓글 3 • 공유 2
+2
김이현
1월 22일 오후 12:02
오늘 완전 재미있었음!!!!
하현주
1월 22일 오후 12:02
강지혜
1월 22일 오후 12:04
댓글을 달아보세요.

kakaostory

DAU: 8M

MAU: 15M

kakaostory

420M API CALL COUNT

kakaostory Service Stack

- For Storage
  - MariaDB(Master/Slave for HA)
  - Hbase
  - cassandra
- For cache
  - Redis
  - Arcus
    - (Memcached variant, opensource, supporting collections)

Redis

5.2TB, 274 Servers

(Arcus: 3.3TB, 137 Servers)

Why Redis?

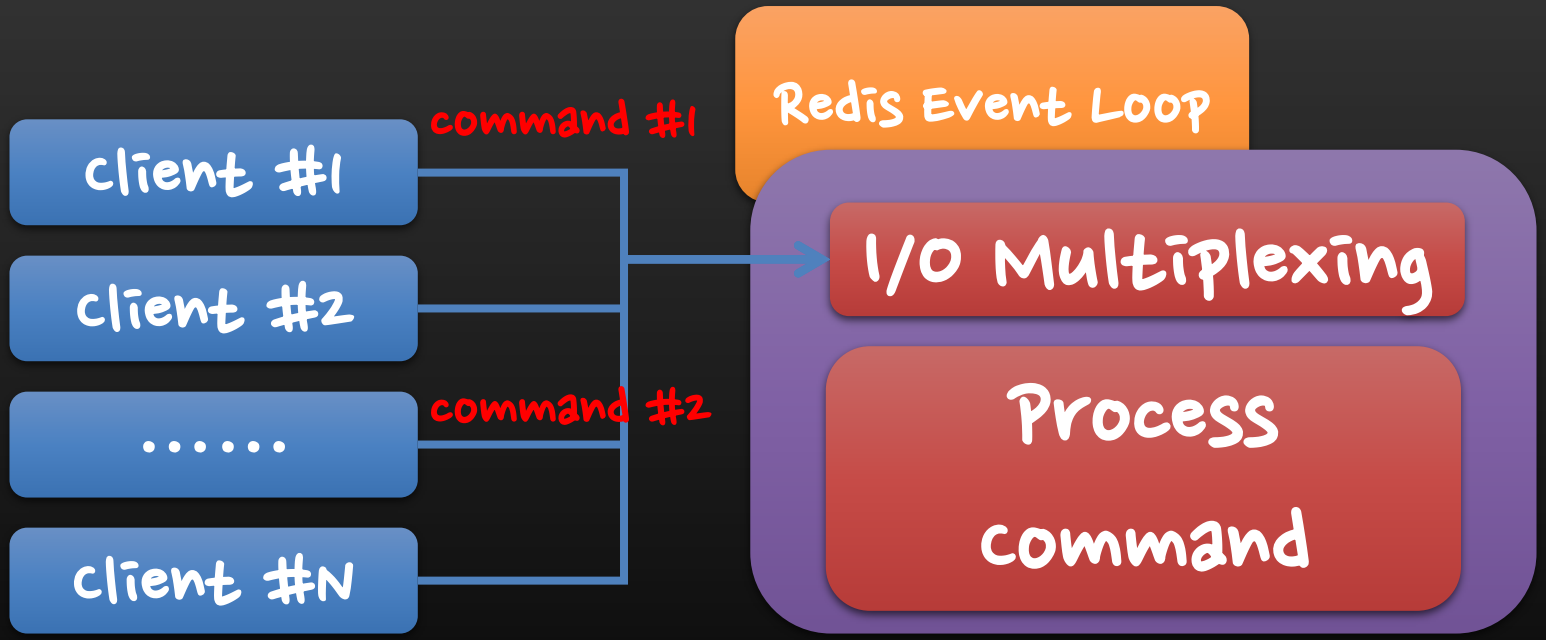
- As lookaside cache for service data
  - Example)
    - User Profile Information
    - Feeds
    - Activities
    - Friends
    - Notifications

Agenda

- Single Threaded
- Memory Fragmentation
- Redis Troubleshooting cases
- Redis Monitoring
- Redis HA

Single Threaded

Redis Event Loop



only one command
at once

Long-time Spending operations

KEYS

FlushAll/FlushDB

LUA Script

MULTI/EXEC

Delete collections

why slow?

O(n)

KEYS - Iterating all keys

```
di = dictGetSafeIterator(c->db->dict);
allkeys = (pattern[0] == '*' && pattern[1] == '\0');
while((de = dictNext(di)) != NULL) {
    ......
    stringmatchlen(pattern,plen,key,sdslen(key),0)
}
```

FlushAll - Deleting all items

```
for (i = 0; i < ht->size && ht->used > 0; i++) {
    dictEntry *he, *nextHe;
    if ((he = ht->table[i]) == NULL) continue;
    while(he) {
        nextHe = he->next;
        dictFreekey(d, he);
        dictFreeval(d, he);
        zfree(he);
        ht->used--;
        he = nextHe;
    }
}
```

How slow?

FlushAll

command	Item count	Time
flushall	1,000,000	1000ms(1 second)

Delete collections

	Item count	Time
list	1,000,000	1000ms(1 second)
set	1,000,000	1000ms(1 second)
Sorted set	1,000,000	1000ms(1 second)
hash	1,000,000	1000ms(1 second)

You can use Xscan commands from 2.8.x

Using Multiple Instances in a Physical Server (can use more CPUs)

Fork for creating RDB, AOF Rewrite

Maximum 2x Memory

Disk IO

CPU Load/usage

CPU 4 core, 32G Memory

more Reliable

Set CPU Affinity using taskset

Divide NIC Interrupt CPU and Redis Process CPU

Memory Fragmentation

Memory Fragmentation #1

used_memory RSS

M [redacted]	368,969	50,000	1,664	688.52K	958.32K	33,589	11.00G	4.60G	5.33G	2.6.16
M [redacted]	415,589	50,000	1,665	741.75K	1.25M	36,500	11.00G	5.28G	6.43G	2.6.16
M [redacted]	441,285	50,000	1,664	772.79K	1.19M	37,960	11.00G	5.61G	6.79G	2.6.16
M [redacted]	427,980	50,000	1,664	769.71K	1.27M	37,820	11.00G	5.44G	6.59G	2.6.16

M [redacted]	6,596,618	50,000	565	2.29M	4.80M	81,985	8.00G	2.66G	12.	2.8.6
M [redacted]	6,191,293	50,000	558	2.22M	5.84M	79,424	8.00G	2.55G	12.	2.8.6
M [redacted]	6,109,388	50,000	564	2.23M	5.10M	79,260	8.00G	2.56G	12.	2.8.6
M [redacted]	5,683,407	50,000	545	2.09M	4.78M	73,412	8.00G	2.44G	12.	2.8.6

Memory Fragmentation #2

							used_memory	RSS	
4,023,378	50,000	1,615	1.41M	12.38M	70,287	12.00G	8.07G	9.60G	2.8.24
3,983,225	50,000	1,616	1.85M	16.59M	72,485	12.00G	7.99G	9.50G	2.8.24
3,887,425	50,000	1,615	1.41M	12.51M	69,209	12.00G	7.77G	9.21G	2.8.24
4,282,243	50,000	1,615	1.53M	13.94M	76,217	12.00G	8.61G	10.30G	2.8.24
3,035,938	50,000	1,615	1.17M	10.02M	58,153	12.00G	6.10G	7.06G	2.8.24
3,989,537	50,000	1,615	1.45M	13.18M	71,869	12.00G	7.99G	9.49G	2.8.24

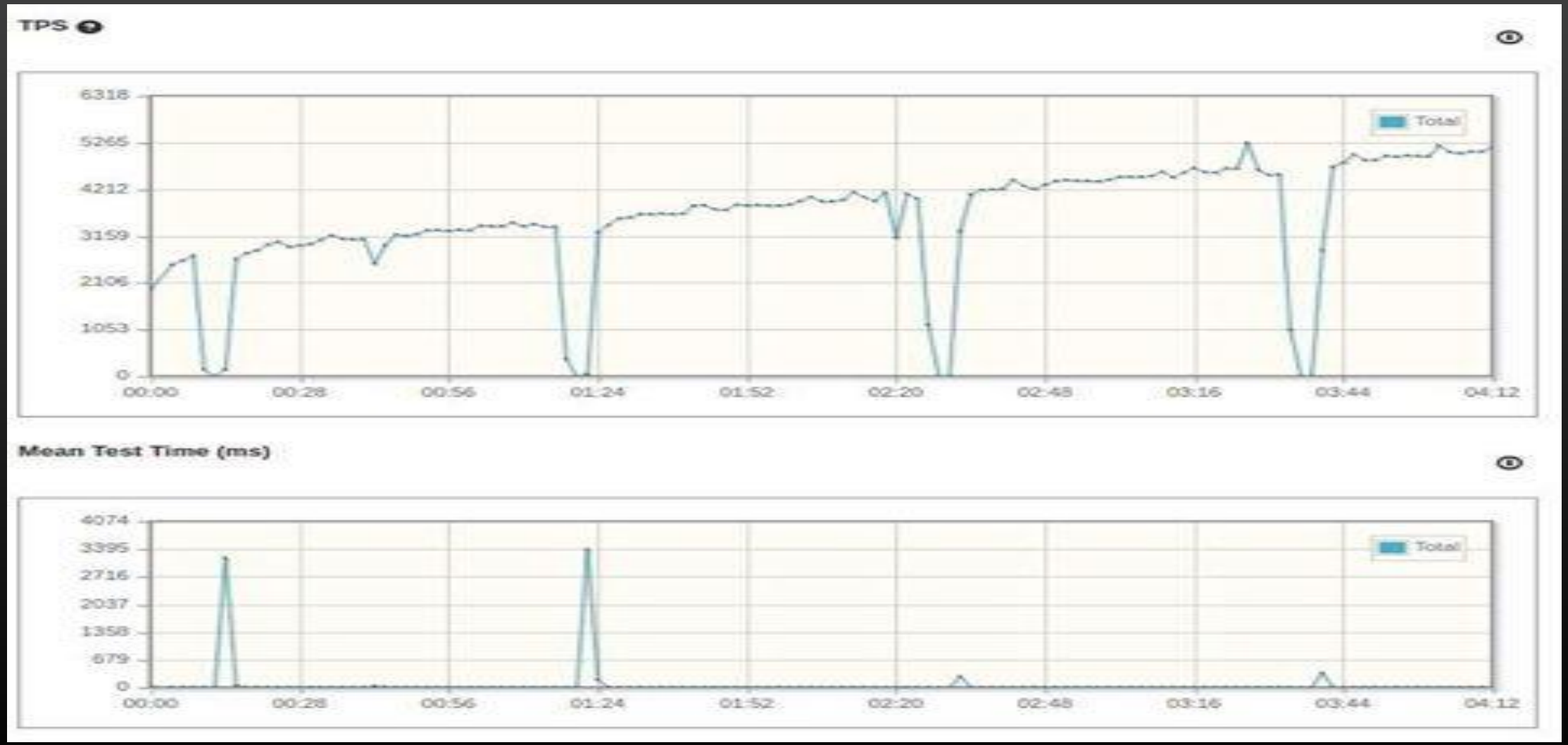
Starting to use Arcus at this case

Redis Troubleshooting cases

Problem #1

KEYS

Performance Spike



INFO all

```
# commandstats
cmdstat_psetex:calls=2326667,usec=9322929,usec_per_call=4.01
......
cmdstat_pexpire:calls=3695333,usec=10068580,usec_per_call=2.72
cmdstat_keys:calls=249,usec=1000314022,usec_per_call=4017325.50
cmdstat_ping:calls=27005,usec=30027,usec_per_call=1.11
......
```

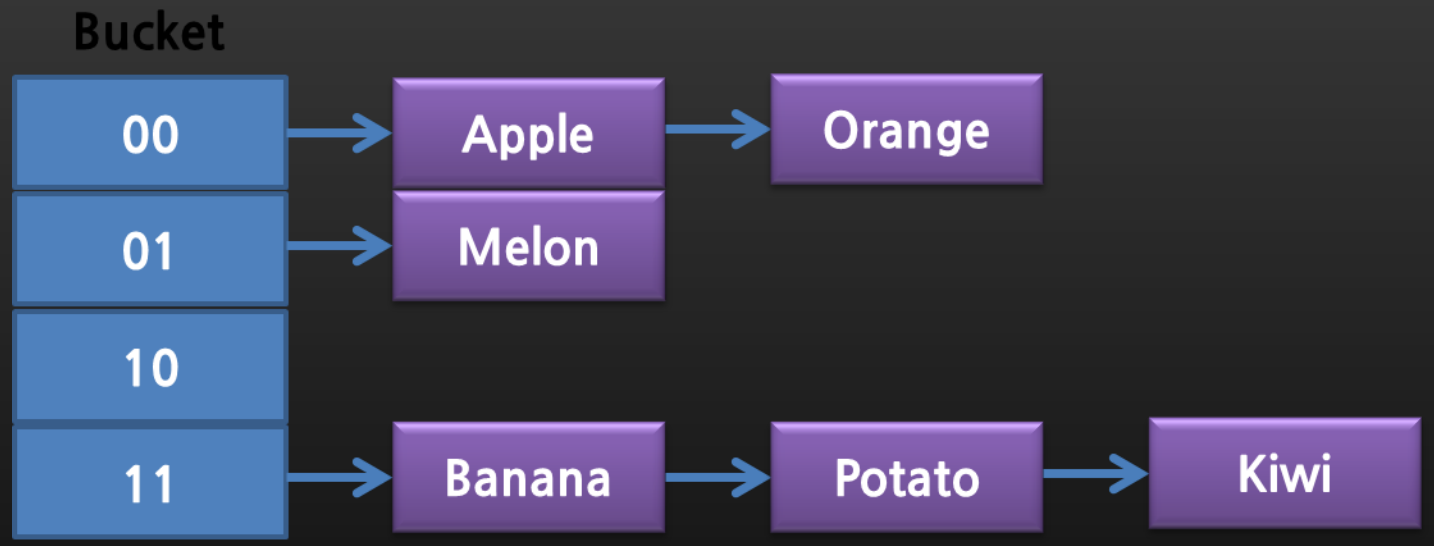
Slowlog get 10

```
redis 127.0.0.1:6379> slowlog get 2
1) 1) (integer) 14
   2) (integer) 1309448221
   3) (integer) 15
   4) 1) "ping"
2) 1) (integer) 13
   2) (integer) 1309448128
   3) (integer) 30
   4) 1) "slowlog"
      2) "get"
      3) "100"
```

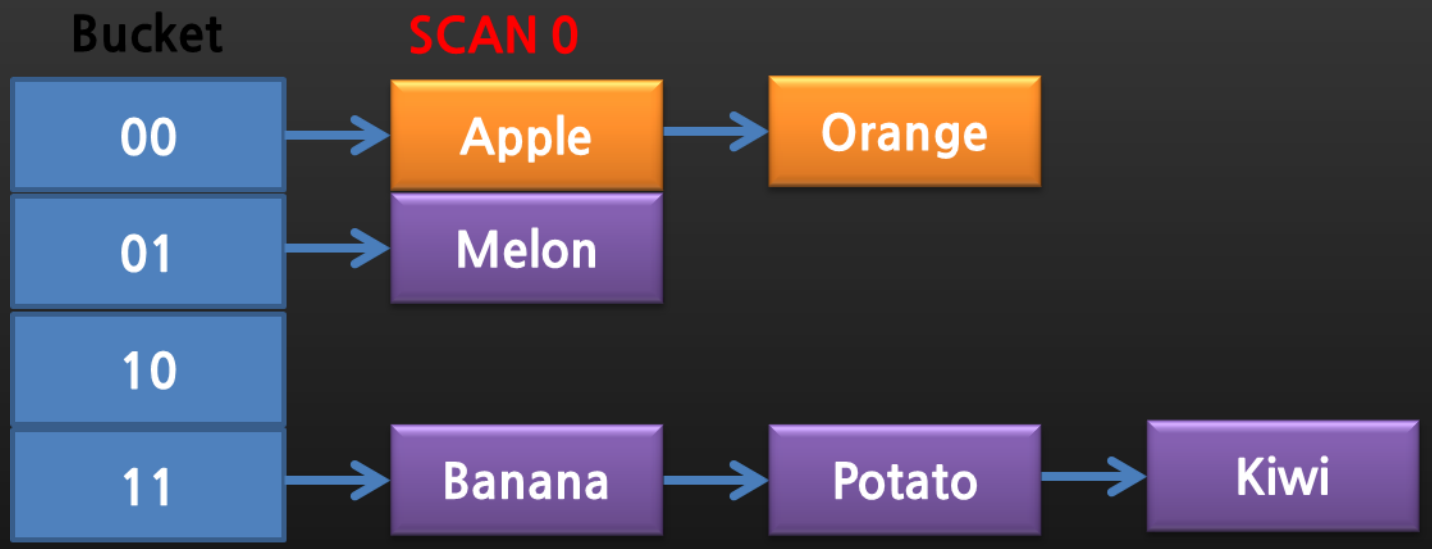
rename KEYS command

using Scan

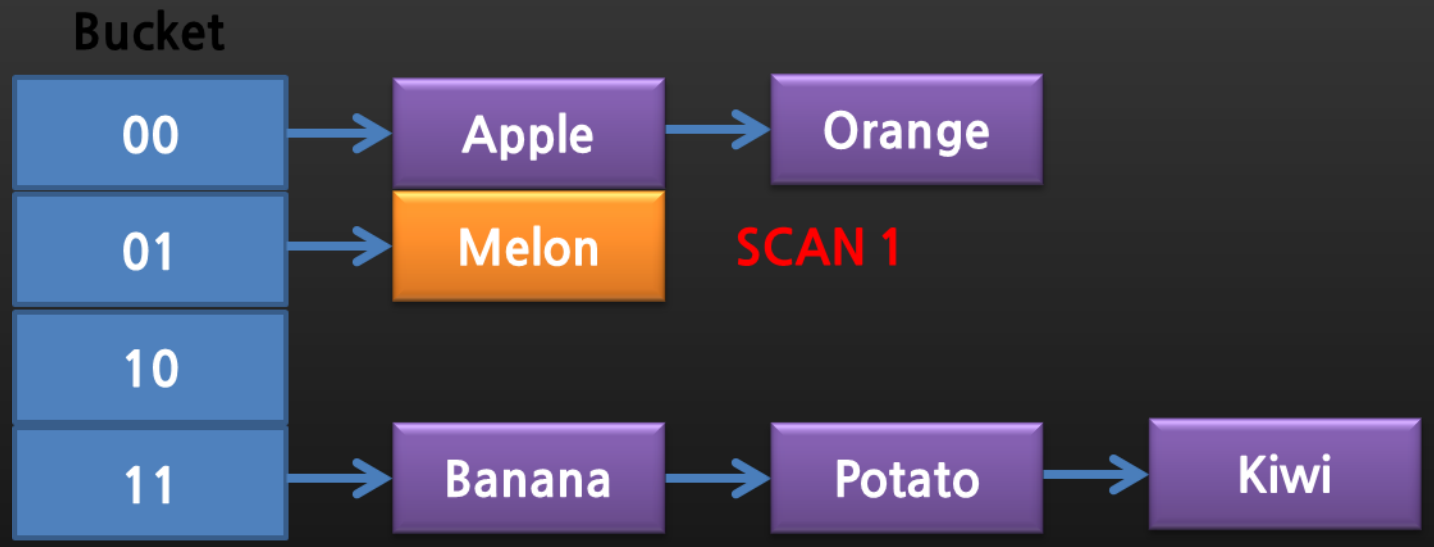
Redis Dict Structure



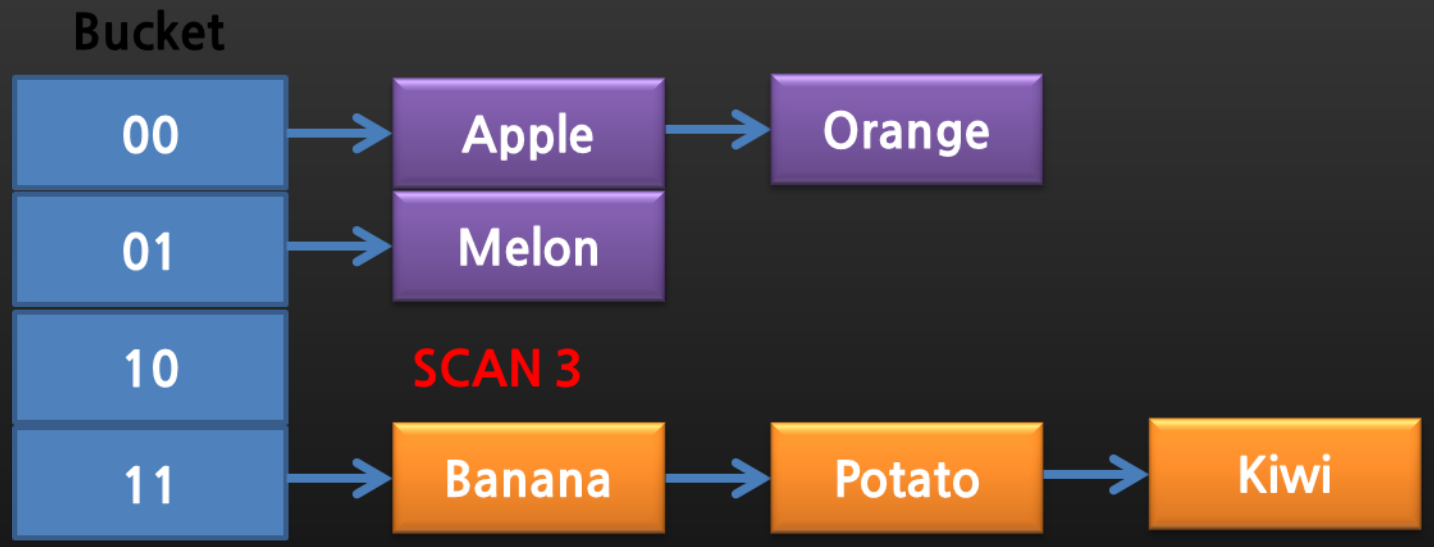
Scan #1


Bucket
SCAN 0
00
01
10
11
Apple
Orange
Melon
Banana
Potato
Kiwi

Scan #2


Bucket
00
01
10
11
Apple
Orange
Melon
SCAN 1
Banana
Potato
Kiwi

Scan #3


Bucket
00
01
10
11
Apple
Orange
Melon
SCAN 3
Banana
Potato
Kiwi

Problem #2
All write commands Fail

“MISCONF Redis is configured to save RDB snapshots, but is currently not able to persist on disk. Commands that may modify the data set are disabled. Please check Redis logs for details about the error.”

Reason

```
if (((server.stop_writes_on_bgsave_err &&
        server.saveparamslen > 0 &&
        server.lastbgsave_status == C_ERR) ||
        server.aof_last_write_status == C_ERR) &&
      server.masterhost == NULL &&
      (c->cmd->flags & CMD_WRITE ||
       c->cmd->proc == pingcommand))
{
      ...
}
```

config set stop-writes-on-bgsave-error no

Problem #3
Using Default Option

Redis as cache

SAVE 900 1

SAVE 300 10

SAVE 60 10000

Heavy Disk IO
High CPU Load
with creating RDB

config set SAVE “”

Problem #4
Using Swap Memory

Redis using 28G
on single 32G machine

Migrate or Restart

Monitor Redis Server and keep within bounds

Problem #5
Simultaneous AOF Rewrite

A 256GB Single Machine

Simultaneous AOF Rewrite

Stop all AOF Rewrites

Turn off Automatic AOF Rewrite

Config set auto-aof-rewrite-percentage 0

Manually Run AOF Rewrite

Problem #6
Replication is Broken with Network Line Failure

All redis replication
are broken
by Network line failure

What Happens if network is recovered

Replication

All slaves automatically try to reconnect to master.

Slave of no one

Problem #7
Replication Failure

Permission

Memory Allocation Fail

sysctl vm.overcommit_memory=1

Replication Failure with OutputBufferSize

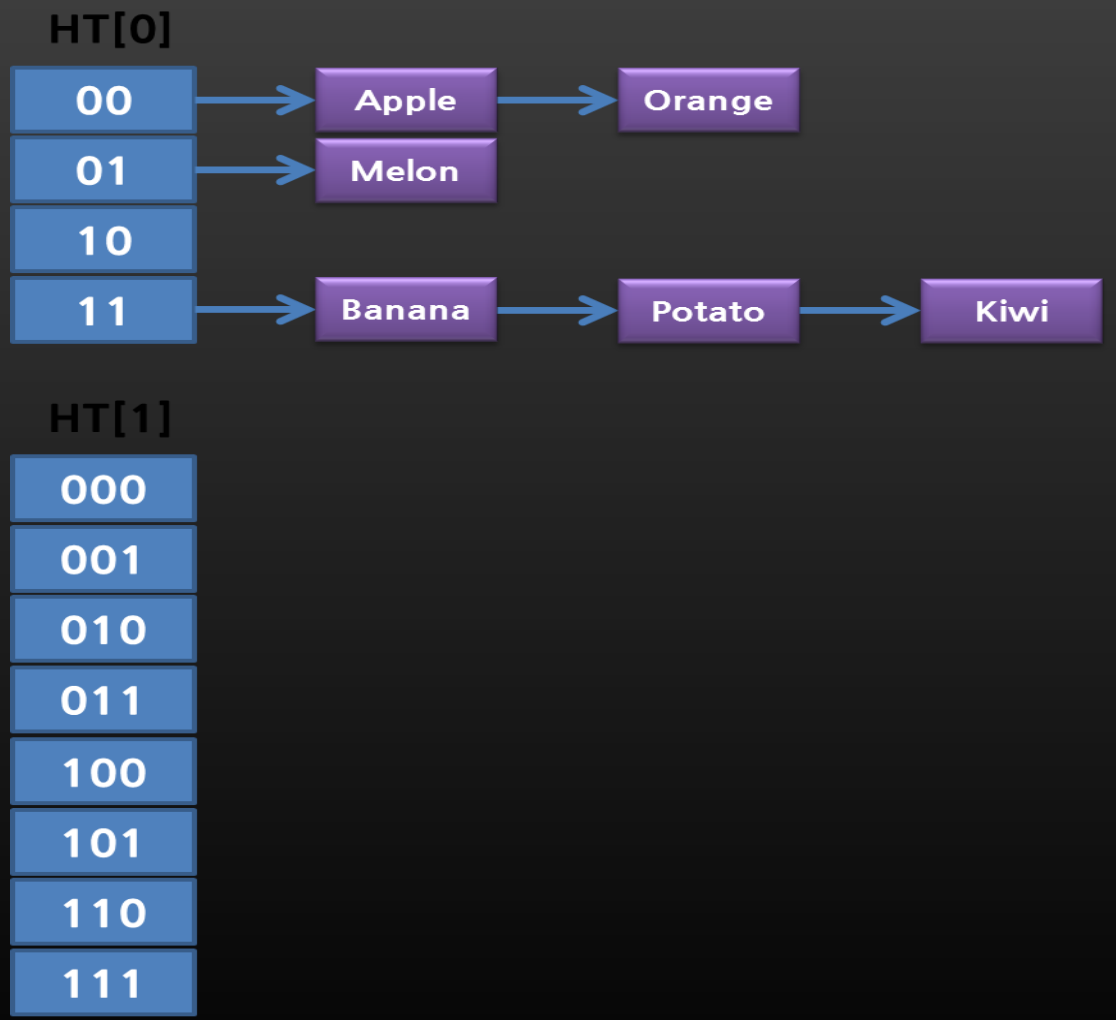
Hard Limit

Soft Limit

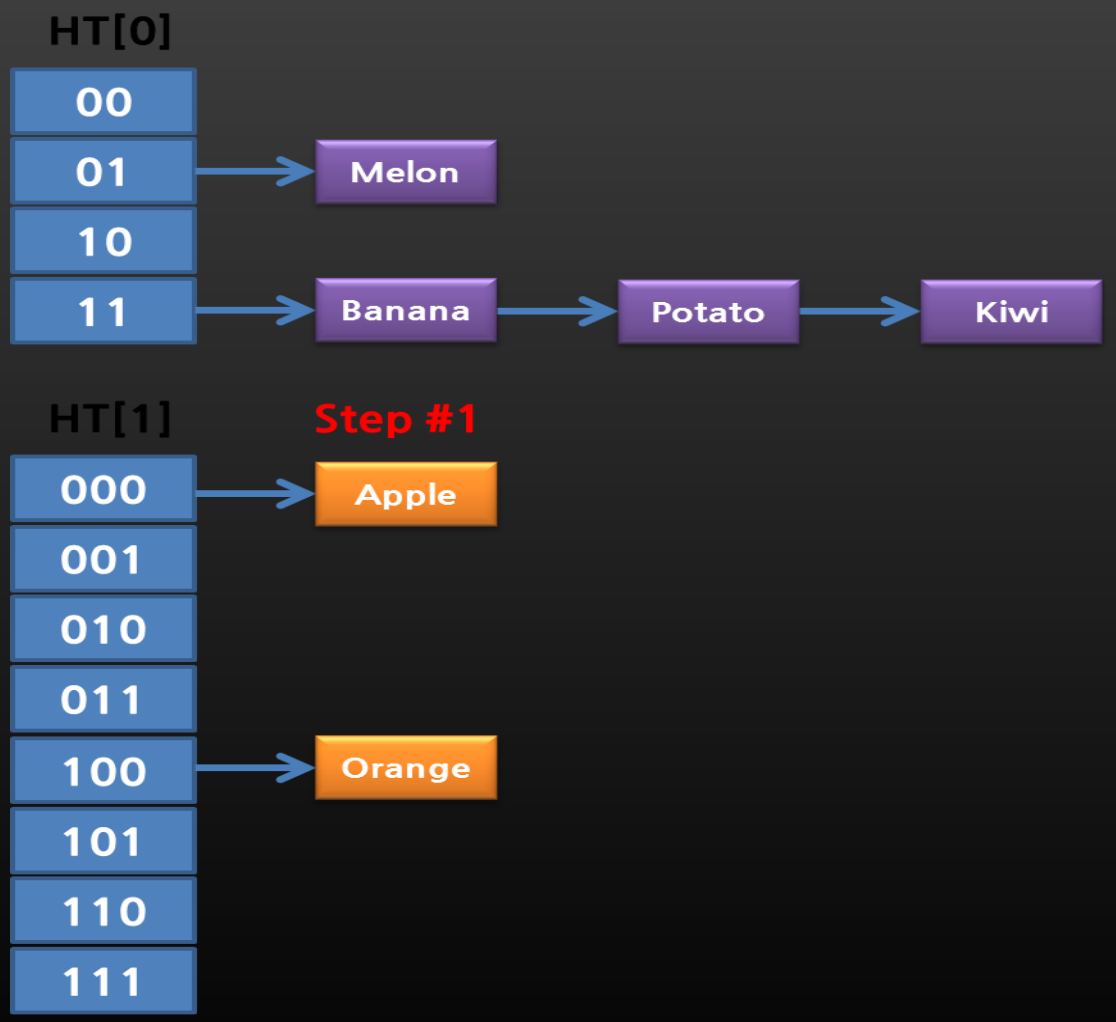
```
config set client-output-buffer-limit
"slave 1024mb 1024mb 60"
```

Problem #8
Hash Table Expansion

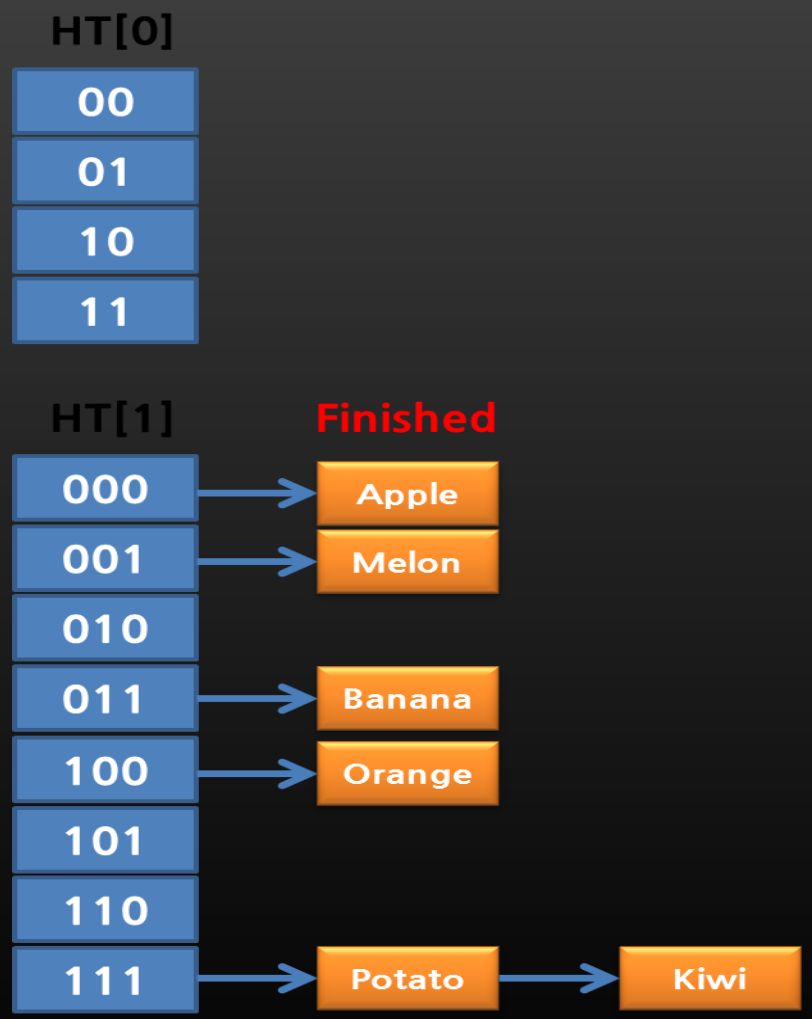
Redis Dict - Hash Table Expansion #1



Redis Dict - Hash Table Expansion #2



Redis Dict - Hash Table Expansion #3



Grows by twice

Maxmemory
and
freeMemoryIfNeeded

1 Billion items

1,000,000,000 * 4 = 4G

Maxmemory = 16G

used_memory = 12G

Hash Table Expansion
is needed.

4G * 2 = 8G.

You need 20G(12G + 8G)

20G > 16G(maxmemory)

Need a feature that can Set Initial size of Hash Table

(Not Supported)

https://github.com/antirez/redis/pull/2812

Redis Monitoring

Monitoring is important as much as Management

Redis Monitoring Metrics

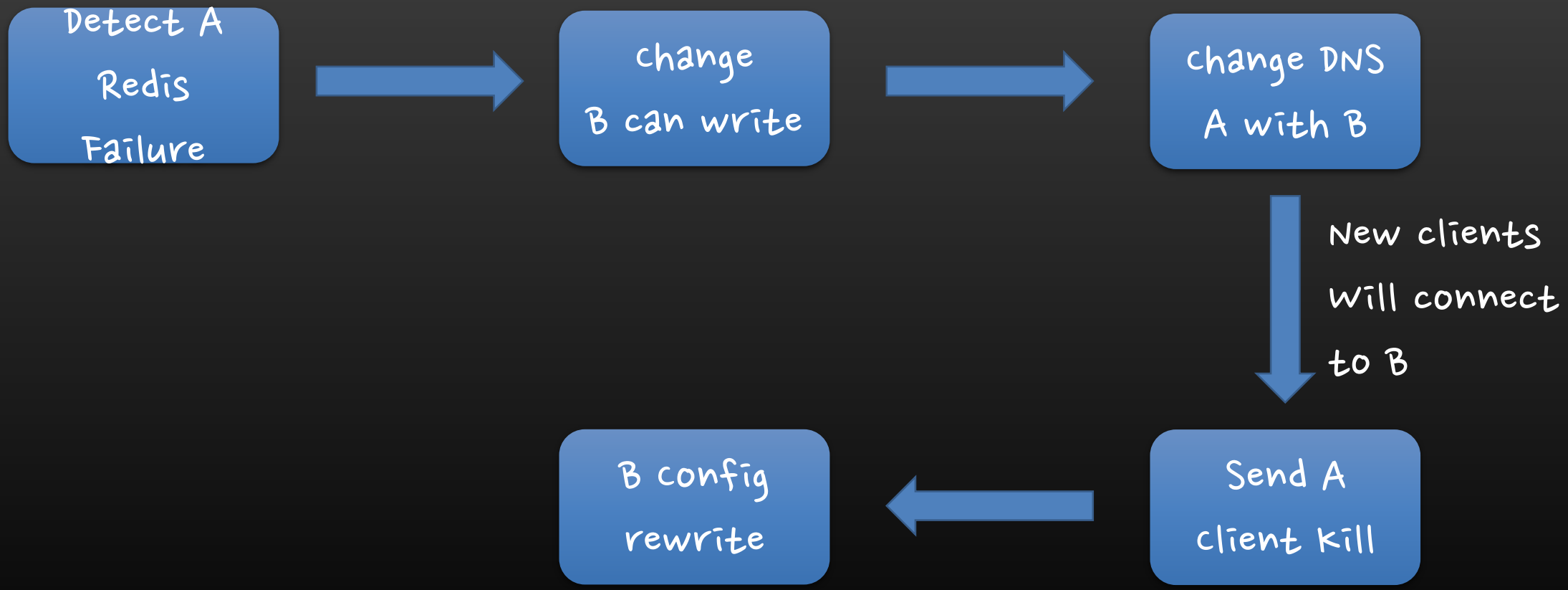
Factor	System or Redis Info
CPU usage, Load	System
Network Inbound/outbound	System
Client connections Maxclient setting	Info
Key size Processed commands	Redis
Memory usage, RSS(very Important)	Redis
Disk usage, IO	System
Expired keys, Evicted keys	Redis

Redis HA

Using DNS for Failover

Private Internal DNS Server with TTL 0

DNS HA Flow


Detect A Redis Failure
change B can write
change DNS A with B
New clients will connect to B
Send A client kill
B config rewrite

JVM

add -Dsun.net.inetaddr.ttl=0

twemproxy
using 0.4.1

using coordinator

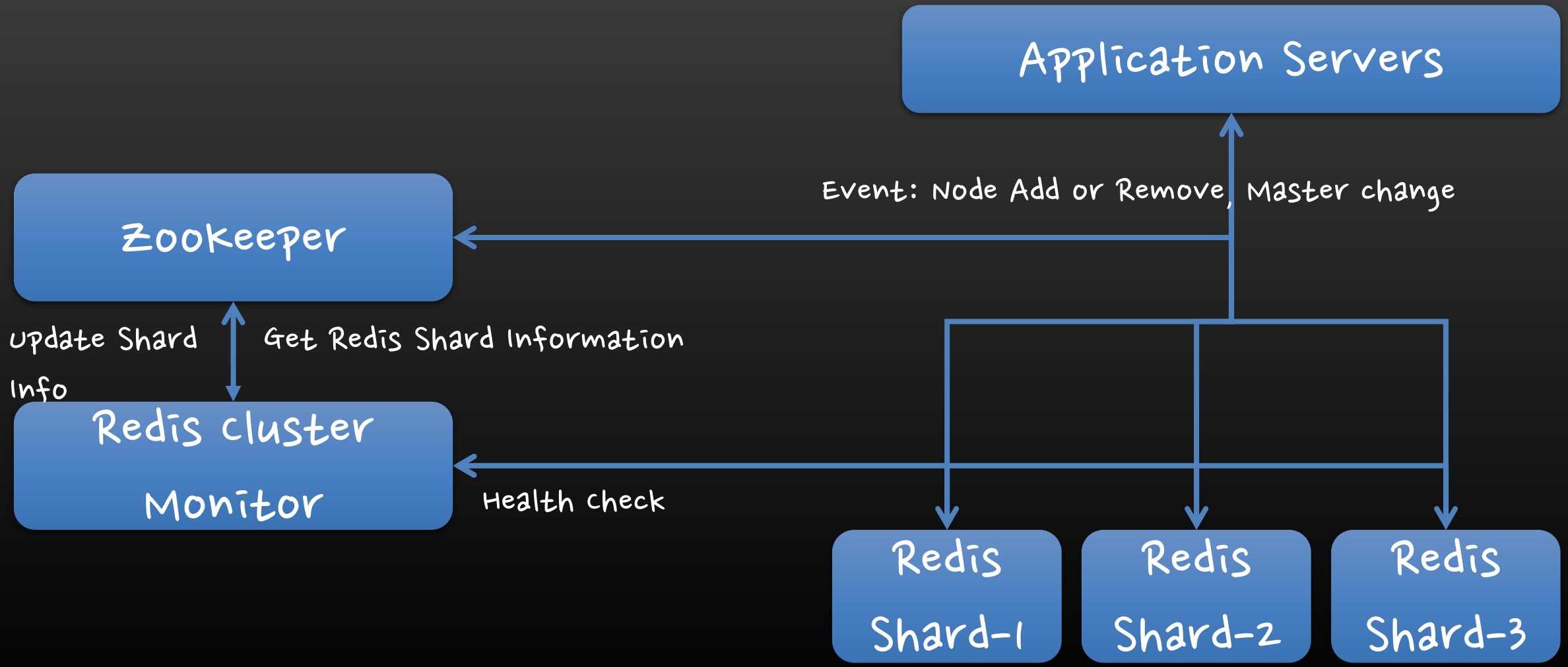
Zookeeper

Zookeeper with Redis Information

Data

{"master":"172.17.50.117:6379","slaves":["172.17.50.116:6379"]}

Zookeeper with Redis



Summary

- Redis is Single Threaded
- creating RDB or AOF Rewrite is expensive
- Don't use KEYS command.
- Don't use default redis configuration.
- Monitoring is very importatnt.

Thanks