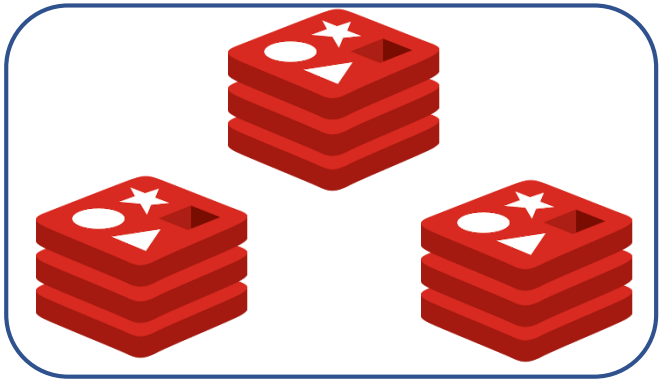
ADDB Cluster Demo

연세대학교 컴퓨터과학과 박상현
2018년 10월

과제명: IoT 환경을 위한 고성능 플래시 메모리 스토리지 기반 인메모리 분산 DBMS 연구개발
과제번호: 2017-0-00477

과학기술정보통신부 Ministry of Science and ICT
연세대학교 YONSEI UNIVERSITY
IITP 정보통신기술진흥센터 Institute for Information & communications Technology Promotion

Redis Cluster를 활용



- 데이터를 최대한 균등하게 분산해서 저장하기 위한 Scale-out 방식
- Load Balancing, Fault Tolerance 이점
- **Cluster** 당 여러 **Nodes**로 구성
- 각 **Node**는 **Slot 범위**를 할당 받음
- 명령어 처리 과정
  - 입력 데이터 Key가 들어왔을 때 , 적절한 Slot을 지정 (Key에 따른 Slot 배정)
  - 이 Slot을 할당 받은 Node에서 데이터를 처리함

  즉, Key → Slot → Node → 처리

How to setup?

* 설치 환경
  - CentOS
  - Redis 4.0.1 (v3.0.0 부터 Cluster 지원)
  - rvm 1.29.3
  - ruby 2.4.2

* Redis Cluster 설치 및 설정
  1. redis gem 설치 (+ rvm, ruby)
  2. conf 파일 설정
  3. Cluster mode 실행

1. redis gem 설치

* 설치 목적

  - **redis-trib.rb** 사용
  - ruby로 짜여진 파일로, Redis Cluster를 쉽게 관리할 수 있음

1. rvm 설치

```
$ curl -sSL https://rvm.io/mpapis.asc | gpg --import -
$ curl -L get.rvm.io | bash -s stable
$ source /path/to/userHome/.rvm/scripts/rvm
$ rvm reload
```

```
hwan@hwan:~/redis/redis-stable$ rvm -v
rvm 1.29.3 (latest) by Michal Papis, Piotr Kuczynski, Wayne E. Seguin [https://rvm.io]
```

2. ruby 설치

```
$ rvm install 2.4.2
```

```
hwan@hwan:~/redis/redis-stable$ ruby -v
ruby 2.4.2p198 (2017-09-14 revision 59899) [x86_64-linux]
```

3. redis gem설치

```
$ gem install redis
```

2. conf 파일 설정

- local에서 3개의 Node(port 8000, 8001, 8002)를 사용한다고 가정
- 각 Node마다의 설정을 담은 **redis.conf** 파일을 만들고, 각 conf 파일 사용하여 server를 실행해야 함

1. 각 conf 파일을 담은 디렉토리 생성

```
$ cd /path/to/redis/
$ mkdir 8000 8001 8002
$ cp redis.conf 8000 & cp redis.conf 8001 & cp redis.conf 8002
$ touch nodes0.conf & touch nodes1.conf & touch nodes2.conf
```

2. 각 conf 파일 수정

8000/redis.conf

port **8000**
cluster-enabled yes
cluster-config-file nodes**0**.conf
cluster-node-timeout 3000
appendonly yes
dir /path/to/**8000**
pid /path/to/pid**8000**/redis_**8000**.pid
logfile /path/to/log**8000**/redis_**8000**.log

8001/redis.conf

port **8001**
cluster-enabled yes
cluster-config-file nodes**1**.conf
cluster-node-timeout 3000
appendonly yes
dir /path/to/**8001**
pid /path/to/pid**8001**/redis_**8001**.pid
logfile /path/to/log**8001**/redis_**8001**.log

8002/redis.conf

port **8002**
cluster-enabled yes
cluster-config-file nodes**2**.conf
cluster-node-timeout 3000
appendonly yes
dir /path/to/**8002**
pid /path/to/pid**8002**/redis_**8002**.pid
logfile /path/to/log**8002**/redis_**8002**.log

3. Cluster mode 실행

* server 실행

$ cd /path/to/redis
$./src/redis-server 8000/redis.conf // 각 Node에 대한 redis server 실행
$./src/redis-server 8001/redis.conf // redis.conf 에서 daemonize yes 하여 단일 bash에서 실행 가능 (default no)
$./src/redis-server 8002/redis.conf

```
# By default Redis
# Note that Redis
daemonize yes
```

* Cluster 시작

$**./src/redis-trib.rb create 127.0.0.1:8000 127.0.0.1:8001 127.0.0.1:8002** // node에 대한cluster 구축
>> Creating cluster
…
Can I set the above configuration? (type 'yes' to accept): **yes**

redis-trib.rb에 의해 cluster 시작 말고도 생성된 cluster에 node 추가/제거, slot 이전 등 기능 수행 가능

* Cluster용 Client 실행

$./src/redis-cli **-c -p 8000** // 8000번 노드로의 클러스터 모드 client 접속
-c : cluster mode

```
hwan@beluga:~/addb$ ./src/redis-cli -c -p 8000
127.0.0.1:8000>
```

3. Cluster mode 실행

* Cluster 정보 확인

127.0.0.1:8000> cluster info

```
127.0.0.1:8000> cluster info
cluster_state:ok
cluster_slots_assigned:16384
cluster_slots_ok:16384
cluster_slots_pfail:0
cluster_slots_fail:0
cluster_known_nodes:3
cluster_size:3
cluster_current_epoch:11
cluster_my_epoch:9
cluster_stats_messages_ping_sent:413081
cluster_stats_messages_pong_sent:399768
cluster_stats_messages_sent:812849
cluster_stats_messages_ping_received:399768
cluster_stats_messages_pong_received:413080
cluster_stats_messages_received:812848
```

* Cluster slot 정보 확인

127.0.0.1:8000> cluster slots

```
127.0.0.1:8000> cluster slots
1) 1) (integer) 0
   2) (integer) 5460
   3) 1) "127.0.0.1"
      2) (integer) 8000
      3) "386c58b505f056f45c4c7988dedba37d553ba9ac"
2) 1) (integer) 5461
   2) (integer) 10922
   3) 1) "127.0.0.1"
      2) (integer) 8001
      3) "bcc0ac63ecc82a55047b10a2bb5cef3475d85542"
3) 1) (integer) 10923
   2) (integer) 16383
   3) 1) "127.0.0.1"
      2) (integer) 8002
      3) "e5cf92d0c27a1cab7c1cda03afc4caa781dec76c"
```

node 번호

slot 시작 번호

slot 끝 번호

host

port 번호

node ID

3. Cluster mode 실행

* Cluster 작동 확인

-
```
127.0.0.1:8000> set foo bar
-> Redirected to slot [12182] located at 127.0.0.1:8002
OK
127.0.0.1:8002>
```

key foo 에 대한 slot이 12182이며, slot [12182]는 node 8002에 할당 되어 있음

→ 따라서 node 8002로 redirect하여 명령어를 수행함

-
```
127.0.0.1:8002> get foo
"bar"
```

그러므로 node 8002에서 올바른 값을 찾을 수 있으며,

-
```
hwan@beluga:~/addb$ ./src/redis-cli -c -p 8000
127.0.0.1:8000> get foo
-> Redirected to slot [12182] located at 127.0.0.1:8002
"bar"
127.0.0.1:8002>
```

node 8000에서 get foo를 해도 마찬가지로 redirect 됨