

Redis 기능의 변화

- 2.8 부터 최근까지

글로벌 오픈 프론티어 4기 파트, 강대명

발표자 소개



- 강대명
- Redis Contributor
- RedisConf 2016 Speaker

오늘 할 이야기들

- Redis 3.x 부터 최근까지(2.8도 조금)
- 가볍게 살펴보고 깊게 들어가지 않습니다.

버전	추가된 기능
2.8	HyperLogLog
3.0	Redis Cluster
3.2	GeoHash
4.0	Redis Module

HyperLogLog #1

- From 2.8
- 원리는 논문을 참고하세요. 키워드로 검색하면 바로 나옵니다.
- 화장품 광고의 타겟 두 그룹
 - A: 20대 후반 여성(50만명)
 - B: 서울의 여성(100만명)
- A+B의 Unique 한 광고 타겟은 몇 명일까?
 - 50만명 + 100만명 = 150만명?
 - 유니크한 유저를 다시 구해야 할까요?

HyperLogLog #2

- 유니크 유저는 어떻게 구해야 하나요?
 - A그룹을 로딩해서, B그룹에 존재하는지 일일이 찾아야 합니다.
 - 유저 수가 클 수록 시간이 많이 걸리게 됩니다.
- 그렇다면 HyperLogLog는...
 - 확률적 자료구조
 - 정확도가 떨어지는 대신, 메모리를 적게 먹고, 속도가 빠릅니다.
 - 즉 유니크한 원소의 정확한 값이 아닌 근사치 값을 보여줍니다.
 - 버킷 사이즈에 따라서 달라지는데, 15%에서 20%까지의 차이가 있을 수 있습니다.
 - 이러면 써야되? 라는 생각이 드실 수도 있는데, 들어가는 비용이 정말 작아서...

HyperLogLog #3

- PFADD, PFCOUNT, PFMERGE 의 세 가지 명령이 있습니다.

명령어	내용
PFADD	HyperLogLog 를 생성한다.
PFCOUNT	HyperLogLog 내의 Unique한 값을 보여준다.
PFMERGE	HyperLogLog 그룹을 하나로 합쳐준다.

HyperLogLog # 4

- PFADD pf1 a b c d e f g
- PFCOUNT pf1
 - 7
- PFADD pf1 a b c z
- PFCOUNT pf1
 - 8
- PFADD pf2 e1 e2 e3 e4 a b
- PFCOUNT pf2
 - 6
- PFMERGE pf3 pf1 pf2
- PFCOUNT pf3
 - 12

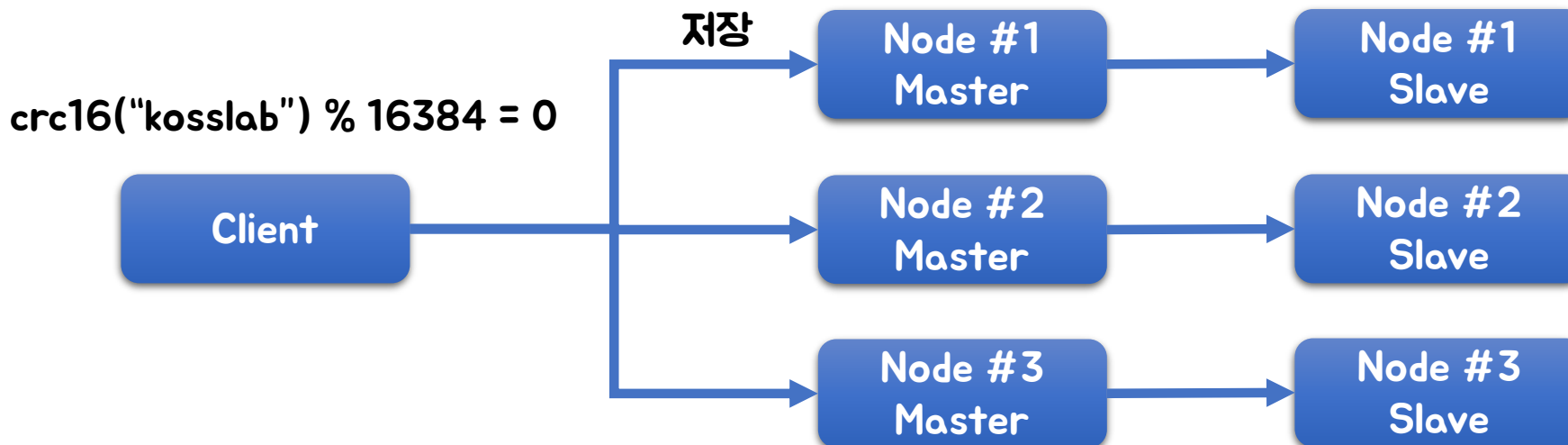
Redis Cluster #1

- From 3.0
- 모두가 기다리던 Redis Cluster
 - 그런데 결과적으로는 좋아하는 곳도 있고,
 - 아닌곳도 많습니다.

Redis Cluster #2

- 동작원리

- 미리 16384 개의 Hash Slot을 정의해두고 주어진 노드들에게 할당한다.
 - Node #1 : 0 ~ 5461
 - Node #2 : 5462 ~ 10922
 - Node #3 : 10923 ~ 16384
- $\text{crc16}(\text{key}) \% 16384 = 0$ 면, 0이 속하는 노드에 저장됨.

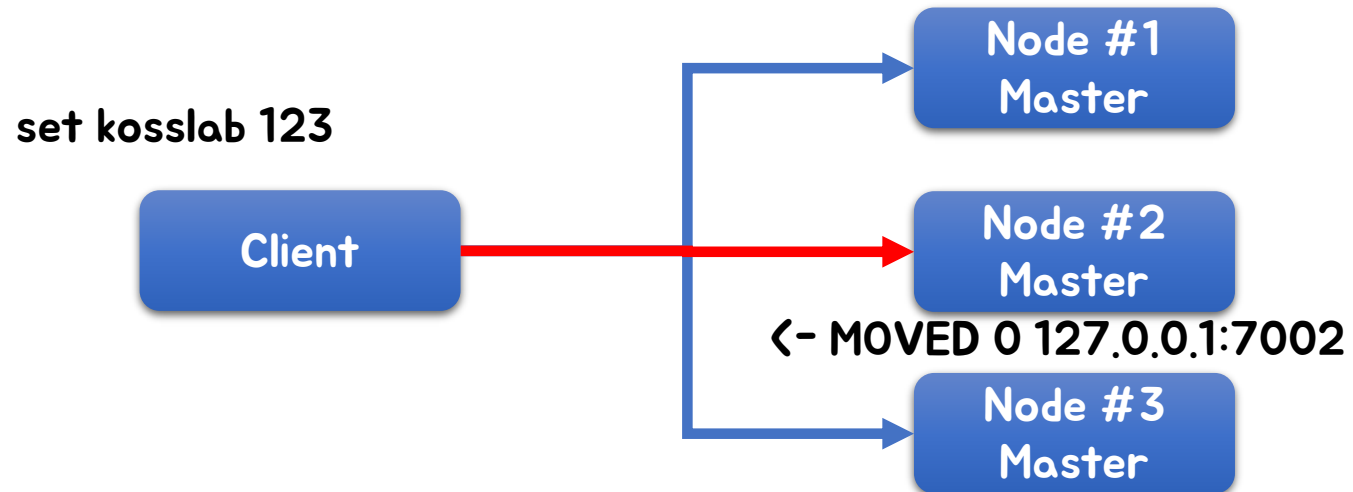


Redis Cluster #3

- Slave 노드는 자신의 Master의 내용만 들고 있음.
- 즉 Node #2의 Master, Slave 가 모두 장애가 났을 때는 그냥 데이터 유실.
- Slot의 재분배를 통해서 데이터의 이동이 가능함.
 - Migration 중인 Slot은 잠시 사용 불가.

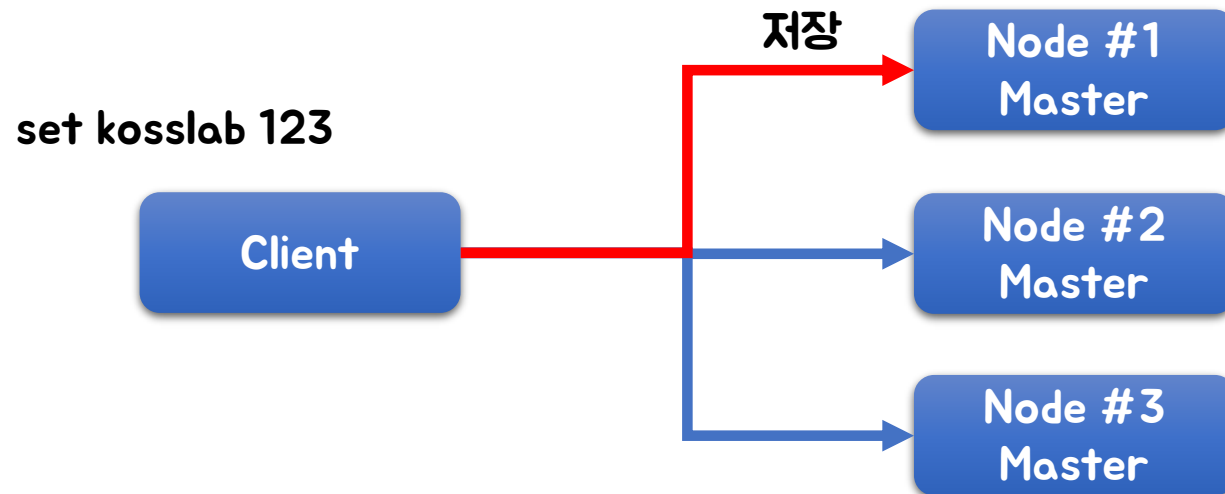
Redis Cluster #4

- Redirection



Redis Cluster #5

- **No Redirection**



Redis Cluster #6 – no cli cluster mode

- cli> set kosslab 123
 - -> (error) MOVED 0 127.0.0.1:7002
- cli> set openfrontier 123
 - -> (error) MOVED 13000 located at 127.0.0.1:7000
- cli> get kosslab
 - -> (error) MOVED 0 127.0.0.1:7002
- cli> get openfrontier
 - -> (error) MOVED 0 127.0.0.1:7000

Redis Cluster #6 – cli cluster mode

- cli:7001> set kosslab 123
 - -> Redirected to slot [0] located at 127.0.0.1:7002
 - OK
- cli:7002> set openfrontier 123
 - -> Redirected to slot [0] located at 127.0.0.1:7000
 - OK
- cli:7000> get kosslab
 - Redirected to slot [0] located at 127.0.0.1:7002
 - "123"
- cli:7002> get openfrontier
 - Redirected to slot [0] located at 127.0.0.1:7000
 - "123"

GeoHash #1

- From 3.2
- Redis 도 GeoHash가 지원됩니다.
- 우버 같은 서비스를 만든다면?
 - 각 우버드라이버의 위치 정보가 존재(지속적 업데이트)
 - 우버사용자의 위치가 존재
 - 현재 사용자의 위치에서 반경 K 킬로미터에 있는 우버드라이버의 리스트를 가져온다면?

GeoHash #2

명령어	내용
GEOADD	GEO Location 정보를 추가한다.
GEOHASH	해당 위치 정보에 대한 해시값을 출력한다.
GEOPOS	해당 유저의 지역정보를 출력한다.
GEODIST	두 지역간의 거리를 출력한다.
GEORADIUS	특정 지역 반경에 존재하는 객체를 조회한다.
GEORADIUSBYMEMBER	특정 객체 반경에 존재하는 객체를 조회한다.

GeoHash #3

- **cli> GEOADD coex 127.0590198 37.5119854 bugerking 127.0588857 37.5127556 coex_sb 127.0601088 37.5153427 bong_sb**
- **cli> GEORADIUSBYMEMBER coex bugerking 1 km**
 - 1) "bugerking"
 - 2) "coex_sb"
 - 3) "bong_sb"
- **cli> GEORADIUSBYMEMBER coex bugerking 100 m**
 - 1) "bugerking"
 - 2) "coex_sb"
- **cli> GEORADIUSBYMEMBER coex bugerking 10 m**
 - 1) "bugerking"

Redis Module #1

- From 4.0
- Redis는 lua script를 지원함.
- lua script 사용의 단점
 - 성능이 느리다.(네이티브로 돌아왔으면 좋겠다.)
- 그러면서 다양한 기능을 커버했으면 좋겠다.
 - Redis PR의 대부분은 기능 추가
 - 기존에 잘 받아들여지지 않음.(가벼운(?) 서버를 지향.)

Redis Module #2

- Redis Module
 - Dynamic Loading Library(.so)
 - 플러그인 방식으로 Redis Command를 동적으로 등록할 수 있다.
 - 기존 명령을 바꿀 수는 없지만, 내부적으로 사용이 가능
 - 새로운 명령을 등록함으로써, 자료구조를 추가할 수 있다.

Redis Module #3

- RedisLabs Module Repository
 - <https://github.com/RedisLabsModules>
 - bloomfilter, Full-Text Search 같은 모듈이 이미 만들어져 있음

Redis Module # 4

- 모듈 작성시 주의 사항
 - Redis는 Single Threaded 이므로, 모듈에서 긴 시간 처리하는 경우 Redis 자체가 블럭되어 버림.
 - 모듈에서 장애가 날 경우, Redis 서버 자체가 종료됨.
 - Master/Slave 형태일 경우는, 장애를 대비해서 Slave 들도 해당 모듈을 다 로드해야 함.
- Redis 코드 자체가 익숙할 경우는 모듈 개발이 어렵지는 않음.

Thanks you!