

Managing Redis - Replication

🕒 생성일	@2022년 12월 28일 오후 8:08
☰ 태그	

Replication

| Redis가 복제를 통해 고가용성 및 장애 조치를 지원하는 방법

Redis 복제의 기반에는 사용 및 구성이 간편한 리더 팔로워 복제가 있다. 팔로워는 리더와의 연결이 끊어질 때마다 정확한 복제본이 되려고 시도한다.

세가지 메커니즘을 활용한다.

1. 리더는 팔로워에게 명령 스트림을 전송하여 팔로워를 업데이트된 상태로 유지한다.
2. 연결이 끊어지면 재연결 후 재동기화를 진행한다.
3. 부분 재동기화가 불가능하면 전체 재동기화를 한다. 부분 재동기화보다 더 많은 리소스가 사용된다.

팔로워는 비동기적으로 리더를 따라간다. 리더는 팔로워가 동기화를 했던 안했던 상관없이 기다리지 않고 운영된다.

클라이언트는 WAIT 명령어를 사용해 특정 데이터의 동기식 복제를 할 수 있다. 하지만 100 퍼센트의 일관성을 가진 동기화는 하지 않는다. 하지만 WAIT 명령어를 사용하면 쓰기 손실 확률을 크게 줄여준다.

Important facts about Redis replication

- 비동기식 복제를 한다.
- 리더와 팔로워는 1:N관계이다.
- 팔로워는 다른 팔로워와 연결할 수 있다.
- 복제는 리더와 팔로워에서 차단되지 않는다.
- 복제는 확장성, 읽기 전용 쿼리, 데이터 안전, 고가용성 향상을 위해 사용 가능하다.
- 마스터가 전체 데이터 세트를 디스크에 쓰는 비용을 피하기 위해 복제를 사용할 수 있다.

Safety of replication when master has persistence turned off

리더와 팔로워는 웬만하면 켜두는 것이 좋다. 인스턴스 자동 재시작을 비활성화 하자.

How Redis replication works

모든 Redis 마스터에는 복제 ID가 있다. 그리고 각 리더는 팔로워로 전송되도록 생성한 복제 스트림의 오프셋을 사용하여 데이터 세트를 수정하는 새로운 변경 사항으로 복제본의 상태를 업데이트한다. 팔로워가 연결되지 않아도 오프셋이 증가하므로 `Replication ID, offset` 형태를 가지고 있다. 이 형태는 리더 데이터 세트의 정확한 버전을 식별한다.

팔로워가 리더에 연결되면 필요한 부분만 식별해서 동기화 하지만, 백로그가 충분하지 않거나 알지 못하는 기록을 참조한 경우 전체 재동기화를 진행한다.

Replication ID explained

복제 ID는 기본적으로 데이터 세트의 주어진 기록을 표시한다. 리더 인스턴스가 띄워지거나 팔로워가 리더가 되는 순간마다 새 복제 ID가 생겨난다. 리더에 연결된 팔로워는 복제 ID를 상속한다. 동일한 ID와 동일한 데이터를 가지고 있지만 다른 시간으로 흘러간다.

Redis 인스턴스가 두 개의 복제 ID를 가지는 이유는 팔로워가 리더로 승격하기 때문이다. 장애 조치 후 승격된 팔로워는 이전 리더의 복제 ID이기 때문에 이전 복제 ID를 계속 기억해야 한다. 이것으로 부분동기화를 진행한다.

Diskless replication

전체 재동기화를 위해서는 디스크에 RDB 파일을 생성하고, 디스크에서 동일한 RDB를 로드해서 팔로워에 공급해야 한다. 느린 디스크(HDD)의 경우 리더에게 굉장히 부담스러운 작업이 될 수 있다.

Configuration

기본 Redis 복제를 구성하는 방법은 `replicaof 192.168.1.1 6379` 를 추가하는 것이다. 리더 IP와 포트를 넣으면 된다. (또는 REPLICAOF 명령어 사용)

디스크 없는 복제도 활성화할 수 있다.

Read-only replica

읽기 전용 지원을 할 수 있다. 읽기 전용 팔로워는 쓰기 명령을 거부하므로 실수로 팔로워에 쓰기 요청을 할 수 없다. 웬만하면 쓰기 가능한 팔로워는 사용하지 마라. 리더와 팔로워간 불일치가 발생할 수 있다.

Setting a replica to authenticate to a master

리더에 requirepass를 통한 비밀번호가 있는 경우 모든 동기화 작업에서 해당 비밀번호를 사용하도록 복제본을 구성하는 것은 간단하다. redis-cli를 치고 다음 명령어 작성

```
config set masterauth <password>
```

영구적으로 설정하려면 다음과 같이 config 파일에 작성

```
masterauth <password>
```

Allow writes only with N attached replicas

최소 N개의 팔로워가 연결되어 있을 때만 쓰기 쿼리를 수행 가능하도록 리더를 설정할 수 있다. 왜냐하면 비동기식 복제이기 때문에 항상 팔로워가 데이터 손실이 없는지 보장할 수 없기 때문이다. 이는 최소한의 안전 매커니즘이다.

조건이 충족되지 않으면 마스터는 대신 오류로 응답하고 쓰기가 허용되지 않는다.

How Redis replication deals with expires on keys

Redis가 만료되면 키의 TTL이 제한된다. 이러한 기능을 구현하기 위해서는 Redis는 동기화된 시계를 갖는 리더나 팔로워의 기능에 의존할 수 없다. 다음과 같은 기술로 운영된다.

1. 팔로워는 키를 만료할 수 없고, 리더가 키를 만료하기를 기다린다. 키가 만료되면 팔로워에게 DEL 명령을 보낸다.
2. 만약 리더가 죽어서 팔로워가 키를 계속 가지고 있다면, 일관성을 해치지 않는 읽기 작업에 대해서만 키가 존재하지 않는다고 보고한다. 이러한 방식으로 삭제되어야 할 키를 논리적으로 클라이언트에게 리턴하지 않는다.
3. Lua스크립트 실행중에는 키만료가 수행되지 않는다.

팔로워가 리더로 승격되면 키가 독립적으로 만료되고, 이전 리더의 도움이 필요없다.

Configuring replication in Docker and NAT

Docker 또는 포트 포워딩을 사용하는 다른 유형의 컨테이너 또는 네트워크 주소 변환이 사용되는 경우 Redis 복제는 특히 Redis Sentinel 또는 마스터 INFO 또는 ROLE 명령 출력이 스캔되어 복제본의 주소를 검색하는 다른 시스템을 사용하는 경우 약간의 추가 주의가 필요하다.

문제는 ROLE 명령과 INFO 출력의 복제 섹션이 마스터 인스턴스로 발행될 때 복제본이 마스터에 연결하는 데 사용하는 IP 주소를 갖는 것으로 표시된다는 것이다. 추가로 설정해주자.

The INFO and ROLE command

- INFO: 명령이 복제 인수와 함께 INFO 복제로 호출되면 관련된 정보만 표시
- ROLE: 마스터 및 복제본의 복제 상태를 복제 오프셋, 연결된 복제본 목록을 제공

Partial sync after restarts and failovers

Maxmemory on replicas