

Managing Redis - Persistence

🕒 생성일	@2023년 1월 6일 오전 2:21
🏷 태그	

Redis persistence

레디스가 디스크에 데이터를 작성하는 법

지속성(Persistence)은 SSD와 같은 내구성있는 스토리지에 데이터를 쓰는 것을 의미한다.

Redis 지속성 옵션

- RDB(Redis DataBase): 지정된 간격으로 데이터 세트의 특정 시점 스냅샷 수행
- AOF(Append Only File): 모든 쓰기 작업 기록. 이 기록은 서버 시작 시 다시 재생되어 원본 데이터 세트를 재구성 가능.
- 지속성 없음: 지속성 비활성화. 캐싱할 때 가끔 사용
- RDB + AOF

RDB advantages

- 백업에 적합하다.
- 원거리 데이터 센터 또는 AWS S3로 전송할 수 있는 단일 압축 파일이므로 재해 복구에 매우 유용하다.
- Redis 부모 프로세스가 지속성을 가지기 위해 수행해야 하는 유일한 작업이 나머지 모든 작업을 수행할 자식을 fork하는 것이므로 Redis 성능을 최대화한다.
- AOF에 비해 큰 데이터 세트로 더 빠르게 재시작할 수 있다.
- 복제본에서 RDB는 다시 시작 및 장애 조치 후 부분 재동기화를 지원합니다.

RDB disadvantages

- Redis가 작동이 멈춘 경우 데이터 손실 가능성을 최소화해야하는 경우는 RDB는 좋지 않다. 일정 간격으로 스냅샷을 생성하는데, 데이터를 그사이에 잃을 수도 있기 때문

- RDB는 하위 프로세스를 사용하여 디스크에서 지속되기 위해 자주 fork해야한다. 이 fork는 최대 1초동안 클라이언트 서비스를 중지할 수 있다. AOF는 fork가 필요하긴 한데 빈번하지 않아서 더 낫다.

AOF advantages

- 훨씬 내구성을 가질 수 있음. fsync를 사용하지 않거나, 매초, 매쿼리, 매초 기본 정책으로 사용하면 쓰기성능이 좋다. fsync는 백그라운드 쓰레드를 사용하여 수행된다.
- AOF로그는 추가 전용 로그이므로 정전 시 검색이나 손상 문제가 없다. 어떤 이유로 로그가 절반만 작성된 명령으로 끝나도 Redis-check-aof 도구를 사용해 쉽게 수정 가능하다.
- Redis에서 AOF가 너무 커지면 백그라운드에서 자동으로 다시 작성할 수 있다. 그리고 이 재작성은 완전히 안전하다.
- 이해하기 쉽고 구문 분석하기 쉬운 형식의 모든 작업의 로그가 하나씩 포함되어 있다.

AOF disadvantages

- 동일한 데이터 세트에 대해서 RDB파일보다 크다.
- 정확한 fsync 정책에 따라 RDB보다 느릴 수 있다.

Redis < 7.0

- AOF는 재작성 중에 데이터베이스에 대한 쓰기가 있는 경우 많은 메모리를 사용할 수 있다.
- 재작성 중에 도착하는 모든 쓰기 명령은 디스크에 두 번 기록된다.
- Redis는 재작성 종료 시 이러한 쓰기 명령을 새 AOF 파일에 대한 쓰기 및 fsync를 중지할 수 있다.

Ok, so what should I use?

RDBMS가 제공할 수 있는 수준의 데이터 안전을 원하는 경우 RDB와 AOF 모두 사용해야 한다.

데이터에 많은 관심을 가지고 있지만 조금의 데이터 손실을 감수할 수 있는 경우 RDB만 사용가능하다.

AOF만 사용할 때, RDB 스냅샷을 사용하는 것은 피하는 것이 좋다. 데이터베이스 백업을 수행하고, 더 빠른 재시작을 하고, AOF 엔진에 버그가 있는 경우에 좋은 아이디어이기 때문이다.



이거 무슨 말인지 모르겠다.

Snapshotting

Redis는 데이터 세트의 스냅샷을 dump.rdb라는 바이너리 파일로 디스크에 저장한다. 자동으로 변경된 횟수나 지정된 시간마다 데이터 세트를 저장하도록 설정할 수 있다.

How it works

1. fork로 자식 프로세스와 부모 프로세스가 생김
2. 자식 프로세스가 데이터 세트를 임시 RDB 파일에 쓴다.
3. 새 RDB 파일 작성을 완료하면 이전 파일 대체

Append-only file

데이터 세트를 변경하는 명령 (ex. SET)을 수신할 때마다 AOF에 추가한다.

Log rewriting

AOF는 데이터의 개수에 상관 없이 명령의 개수에 비례해서 점점 커진다. 이로 인해 다시 쓰기는 완전히 안전하다. 그리고 서비스를 중단하지 않고 백그라운드에서 AOF를 다시 빌드할 수 있다.

How durable is the append only file?

AOF가 얼마나 내구성이 있는지는 Redis가 디스크에서 데이터를 fsync 하는 횟수를 구성할 수 있다. 다음과 같은 세가지 옵션이 있다.

- 항상: 새 명령이 AOF에 추가할 때마다 fsync한다. 매우 느리지만 안전하다.
- 매초: 충분이 빠르고 장애가 발생하면 1초의 데이터가 손실될 수 있다.
- fsync 안함: 데이터를 운영체제에 맞기기 때문에 가장 빠르지만 덜 안전하다.

What should I do if my AOF gets truncated?

AOF가 충돌하거나 잘렸을 수 있다. 이 경우 마지막 올바른 명령은 버린다. 충돌하거나 잘렸을 경우

What should I do if my AOF gets corrupted

AOF가 중단되었을 때 Redis는 에러를 뱉으며 정지한다. 파일을 수동으로 복구할 수 있는지 확인하고, 수동으로 수정하면 된다. 수동으로 복구할 수 없는 경우, 복구 가능하지만 유효하

지 않은 부분부터 AOF 끝부분이 폐기될 수 있으므로 데이터 손실이 일어날 수 있다.

How it works

로그 재작성은 스냅샷에 사용되는 복사트릭을 사용한다.

1. 부모 프로세스를 fork해서 자식 프로세스를 생성한다.
2. 자식 프로세스는 AOF파일을 기반으로 재작성을 시작한다.
3. 부모는 새로운 버퍼(7버전 이하) 또는 새로운 AOF(7버전 이상)에 새롭게 들어오는 명령들의 로그를 작성한다.
4. 자식 프로세스가 작성을 마치면 부모프로세스가 새로 추가한 로그들을 기반으로 지속적으로 작성한다.

How I can switch to AOF, if I'm currently using dump.rdb snapshots?

dump.rdb파일을 AOF에 적용할 수 있다.

Interactions between AOF and RDB persistence

AOF와 RDB는 같이 실행되지 않게 하는 것이 좋다. 동시에 진행하면 많은 디스크 I/O를 사용하기 때문이다. 스냅샷을 시작하면 AOF는 잠시 멈추고 스냅샷 작업이 완료되면 재작성이 시작된다. AOF와 RDB가 모두 활성화된 상태에서 Redis가 재시작하면 AOF파일로 데이터 세트를 재구성한다.

Backing up Redis data

DB는 항상 백업해야한다. 데이터베이스가 실행되는 동안 RDB파일을 복사할 수 있으므로 데이터 백업에 매우 유리하다.

- 한 디렉토리에 RDB파일의 시간별 스냅샷을 생성하고 다른 디렉토리에 일일 스냅샷을 생성하는 서버에서 cron작업을 생성한다.
- cron 스크립트가 실행될 때마다 find명령을 호출해 너무 오래된 스냅샷을 주기적으로 삭제완료 했는지 확인해야한다.
- 이 RDB 스냅샷을 하루에 한번 이상 다른 곳으로 전송해서 보존해야한다.

Backing up AOF persistence

AOF만 활성화된 상태에서도 백업 수행 가능하다. AOF파일이 저장된 곳을 복사하고 tar 파일로 압축하기만 하면 된다. 하지만 재작성 중에 이작업을 수행하면 유효하지 않은 백업이 생성될 수 있기 때문에 백업중에 AOF 다시 쓰기를 비활성화 해야한다.

AOF가 비활성화되는 시간을 최소화 하려면 추가적인 설정으로 최소화 할 수 있다.

Disaster recovery

재해 복구를 위해 위의 방법으로 백업한 데이터를 외부 데이터로 전송 가능하다. 이러면 Redis 인스턴스가 망가져도 데이터가 보호된다.

S3에 업로드하는 방법과 SCP(SSH의 일부)를 이용해 스냅샷을 원거리 서버로 전송하는 방법이 있다.

이 방법이 잘못 구현되어있으면 쉽게 실패할 수 있다. 최소한 전송이 완료된 후 파일크기와 SHA1 다이제스트(VPS를 사용하는 경우)를 확인할 수 있는지 무조건 확인해라. 만약 백업 전송이 안되는 경우 경고를 할 수 있는 시스템도 필요하다.