

Redis에서 Physiological RDB 회복 기법 제안

진민화, 최원기, 박상현

연세대학교 컴퓨터과학과

e-mail : mhjin@yonsei.ac.kr, cwk1412@cs.yonsei.ac.kr, sanghyun@cs.yonsei.ac.kr

Physiological RDB Recovery Method in Redis

Minhwa Jin, Wonki Choi, Sanghyun Park

Dept. of Computer Science, Yonsei University

요약

Redis는 In-Memory 기반 Key-Value Store로서 Lists, Sets, Sorted Sets, Hashes와 같은 효율적인 자료구조를 제공한다. 그리고 분산 데이터 관리를 위한 Master-Slave Replication, Sharding 등을 제공하기 때문에 Twitter, Weibo, KakaoTalk과 같은 많은 기업에서 사용되고 있다. In-memory 기반의 Redis는 전원이 공급되지 않으면 데이터가 손실되는 문제점이 존재한다. 따라서 Redis에서는 데이터를 생성한 프로그램의 실행이 종료되더라도 데이터의 손실을 방지하는 특성인 영속성(Persistence)을 보장하는 기법을 제공한다. 먼저 Write/Update 연산 자체를 모두 Log파일의 마지막에 기록하는 AOF(Append Only File) 기법과 메인 메모리 내에 있는 데이터 전체를 주기적으로 이미지 파일 형태로 디스크에 저장하는 RDB(Snapshot) 기법을 이용한다. 그 중에서 RDB 기법은 메인 메모리에 존재하는 데이터의 크기가 클 경우 많은 CPU 점유율과 Disk I/O가 발생할 수 있기 때문에 Redis의 성능 저하로 이어질 수 있다는 문제점이 있다. 본 논문에서는 이러한 문제를 해결하기 위해 기존의 RDB 기법을 개선한 Physiological RDB 기법을 통하여 I/O 성능을 향상하는 방법에 대하여 제시하고자 한다.

1. 서론

기존의 데이터 관리를 위해 주로 사용하는 디스크 기반 데이터베이스는 데이터를 디스크에 저장을 하고 필요한 데이터의 대한 질의나 삽입, 삭제, 수정 등의 작업을 디스크에 접근하는 방식을 이용했다. 효율적인 디스크 I/O를 위한 여러 가지 인덱스와 자료구조들이 제시되었고, 기존의 HDD를 대신하여 Flash SSD를 이용해 I/O 향상을 이루었다. 하지만 컴퓨팅 성능의 발전 속도를 디스크와 기존의 디스크 기반 데이터베이스 시스템이 따라가지 못하게 되었고, 분석 데이터의 실시간 처리와 더 빠른 데이터 접근을 위해 디스크 I/O 없이 메모리에 데이터를 저장하고 관리를 하는 Memcached[1], Redis[2]와 같은 In-memory 기반 데이터베이스가 등장하게 되었다.

In-Memory 기반 데이터베이스는 메인 메모리에 설치되어 운영되는 데이터베이스 관리 시스템이다. 즉 데이터베이스의 데이터를 일부 또는 전부를 메인 메모리에서 관리를 하게 되어 기존의 디스크 기반 데이터베이스보다 좋은 성능을 나타낸다.

Redis는 기존의 Key-Value Store와 달리 Lists, Sets, Sorted Set 등과 같은 여러 가지 자료구조와 Master-Slave Replication과 Sharding 기능 등을 제공하기 때문에 대용량 분산 데이터 관리 시스템에서 사용되고 있다. 그러나 Redis에서 데이터 접근은 메모리에서 일어나지만 Redis 인스턴스가 종료되더라도 데이터의 손실을 막기 위해서 Persistence Storage를 이용한다[3]. Redis는 대표적인 두 가지 방법인 AOF(Append Only File), RDB(Snapshot) 기법을 이용하여 데이터의 손실이 있어도 데이터 복구가 가능함을 보장한다. 이 때 RDB 기법은 메모리에 있는 모든 데이터를 이미지 형태로 디스크에 기록하기 때문에 메모리상에서 관리하는 데이터가 많으면 I/O 비용이 증가하게 된다.

이에 본 연구는 기존 RDB 기법에 대해 알아보고 기존 RDB 기법을 개선한 Physiological RDB 기법을 제안하고 I/O 비용을 효과적으로 줄이는 방법에 대하여 다뤄보려 한다.

이 논문의 구조는 다음과 같다. 2장에서는 Redis가 구체적으로 어떠한 방식으로 영속성(Persistence)을 보장하는지 살펴보고, 구체적으로 RDB 파일의 구조에 대해 알아본다. 그리고 3장에서는 개선된 RDB 기법을 소개한다. 마지막 4장에서는 논문의 결론과 향후 연구 방향을 제시한다.

* 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2015R1A2A1A05001845).