

Redis 캐시 적용 및 성능 개선 보고서

작성일: 2025-05-08

작성자: 김종완 (2팀)

1. 요약 (Executive Summary)

도입 목적

급증하는 사용자 트래픽과 30만 건 이상의 상품 데이터 증가로 인한 데이터베이스 부하 심화 문제를 해결하고, 응답 성능을 개선하기 위해 Redis 캐시 도입 효과를 정량적으로 분석함.

테스트 개요

- 시나리오:** 사용자 수(VU) 10명부터 100명까지 점진적으로 증가
- 대상 데이터:**
 - 상품 데이터: 300,000건
 - 판매 통계 데이터: 50,000건
- 테스트 대상 API:**
 - 상품 목록 조회 API: `/products?offset=0&limit=10000`
 - 인기 상품 조회 API: `/products/popular?limit=1000`

결론 요약

Redis 캐시 적용 후 두 API 모두 다음과 같은 효과를 보였음:

- 평균 응답 시간 최대 **67% 감소**
- 초당 처리량(TPS) 최대 **3배 증가**
- SLA(300ms 이하 응답) 만족 비율: **51% → 85% (API #1 기준)**

2. 문제 배경

현황 및 과제

- 전체 상품 수 30만 건 이상, 데이터 지속 증가
- 트래픽 증가로 인한 DB 부하 및 쿼리 처리 지연

API별 문제점

- 상품 목록 조회:** offset/limit 기반의 조회 구조로 요청 시마다 쿼리 수행. limit가 커질수록 성능 저하 심화
- 인기 상품 조회:** 상품 테이블과 통계 테이블 간 Join으로 인한 연산 부담. 실시간 통계 필요 시 성능 저하 우려

3. 캐시 적용 설계

캐시 Key 구조

- 상품 목록 조회: `products:list:{offset}:{limit}`
- 인기 상품 목록 조회: `products:rank:{limit}`

TTL 및 무효화 정책

- 상품 목록: TTL 1분, 상품 정보 변경 시 캐시 삭제
- 인기 상품: 매일 새벽 1시 통계 갱신 스케줄러 실행 시 갱신

적용 방식

- Spring Cache 또는 RedisTemplate + AOP 기반 적용

- Cache miss 발생 시 DB 조회 후 캐시 저장

4. 성능 테스트 개요

테스트 환경

- Redis: Standalone
- DB: 300,000건의 상품이 삽입된 MySQL 샘플 DB
- 테스트 도구: K6

측정 지표

- 평균 응답 시간
- TPS (초당 처리량)
- SLA (응답 시간 기준)
- DB 쿼리 수

5. 성능 비교 요약

항목	상품 목록 조회 (API #1)	인기 상품 조회 (API #2)
성공률	72.46% → 96.03% (+23.57%)	99.78% → 99.97% (+0.19%)
평균 응답시간(ms)	604ms → 250ms (-59%)	177ms → 59ms (-67%)
p95 응답시간(ms)	1.23s → 738ms (-40%)	368ms → 186ms (-49%)
SLA 만족률	51% → 85% (+34%)	10% → 83% (+73%)
TPS	77.7 → 186.6 (2.4배)	263 → 791 (3배)
최대 응답시간	1.96s → 9.61s (일부 outlier)	776ms → 1.01s (일부 outlier)

6. 분석 인사이트

API #1 (상품 목록 조회)

DB 의존도가 높아 SLA 미달 요청이 다수 발생했으며, Redis 적용 후 응답 속도는 절반 이하로 감소하고 TPS는 2.4배 증가함.

API #2 (인기 상품 조회)

본래 성능이 양호하였으나, 캐시 적용 후 처리량 3배 증가 및 속도 안정성 확보. 트래픽 급증 상황 대응력 향상됨.

7. 결론 및 제안

결론

본 테스트는 Redis 캐시의 성능 개선 효과를 확인하기 위해 요청량과 데이터 크기를 인위적으로 증폭한 테스트 환경에서 수행함.

API 요청 조건을 일반적인 수준으로 조정한 경우, 캐시 없이도 충분히 빠른 응답 시간 확보 가능:

- 상품 목록 조회 (offset ≤ 100): 평균 70ms
- 인기 상품 조회 (3건 요청 시): 평균 30ms

그러나 TPS 증가 등 **처리량 확보** 측면에서는 Redis 캐시 도입 효과가 뚜렷하게 나타남.

제안

- **현 시점에서는 즉각적인 Redis 캐시 도입은 필요 없음**
- offset 제한 설정 및 SLA 목표 수립 후 지속적인 모니터링 권장

예) 상품 목록 조회 SLA 70ms, 인기 상품 조회 SLA 30ms

- **향후 오류 응답률이 10% 이상 발생할 경우 Redis 캐시 도입을 적극 검토**