

프로젝트 명세서

웹 어플리케이션 성능 향상을 위한 캐시 적용

목차

1. 프로젝트 개요.....	3
2. 과제	4
3. 심화 과제	6
4. 산출물 제출.....	7

1. 프로젝트 개요

TOP SSAFY 팀'은 많은 고생 끝에 기획했던 '맛 집 추천 서비스'를 성공적으로 구현 할 수 있었습니다. 기능 테스트를 꼼꼼하게 진행했기에 일반 사용자를 대상으로 실 서비스를 할 때에도 별다른 걱정이 없었습니다. 입 소문이 나며 달콤한 꿈까지 꿀 수 있었습니다. 그렇지만 사용자 수가 눈에 띄게 증가함에 따라 웹 어플리케이션에 많은 부하가 걸리게 되었고 답답한 서비스 응답시간 때문에 고객의 신뢰를 잃어 서비스 중단이라는 안타까운 길을 걷게 되었습니다.

사용자 요청이 증가함에 따라 웹 어플리케이션 서버에는 많은 부하가 발생하게 됩니다. 개발자 1명이 테스트 할 때에는 시스템 구조나 프로그래밍의 비효율성이 큰 문제가 되지 않지만 점점 더 많은 요청을 처리해야 할 때에는 치명적인 문제를 야기할 수 있습니다. 첫술에 모든 문제를 풀어낼 수 있는 해안을 가질 수 는 없습니다. 그렇지만 적은 노력으로 가장 큰 효과를 얻을 수 있는 '캐시'에 대한 개념을 알고 프로젝트에 적용 할 수만 있다면 그 첫술로는 최고의 선택이라 말 할 수 있을 것 같습니다.

2. 과제

1) 캐시에 관한 학습

공통 프로젝트에 캐시 솔루션을 적용하기 전 일반적인 캐시의 개념을 학습 할 필요가 있습니다. 캐시는 왜 쓰는지? 그 기초 원리는 무엇인지? 핵심 개념을 이해해야만 캐시를 프로젝트에 도입 할 때에도 어떤 서비스에 캐시를 적용 할지 또 위험 요소는 없는지 체크할 수 있는 능동적인 활용이 가능하기 때문입니다.

캐시는 어려운 개념이 아니며 대부분의 프로그램에 적용되어있습니다. 예를 들어 브라우저는 페이지 요청마다 네트워크를 통해 화면에 표시할 이미지를 가져오지 않습니다. 보통은 로컬 파일 저장 장치에 캐싱하여 두고 이를 바로 사용하게 됩니다. 이렇듯 상대적으로 느린 장치의 입출력을 통해 사용 되는 정보를 보다 빠른 장치에 임시 저장하여 데이터로 활용함으로써 전체적인 성능 향상을 도모하는 메커니즘이 캐시의 핵심입니다. 아래의 참고 링크 뿐만 아니라 스스로 캐시의 개념에 대해 설명 할 수 있을 때까지 집중력을 가지고 학습하셨으면 합니다.

- 왜 캐시를 사용하는가?

<https://charsyam.wordpress.com/2016/07/27/%EC%9E%85-%EA%B0%9C%EB%B0%9C-%EC%99%9C-cache%EB%A5%BC-%EC%82%AC%EC%9A%A9%ED%95%98%EB%8A%94%EA%B0%80/>

- 캐시 메모리

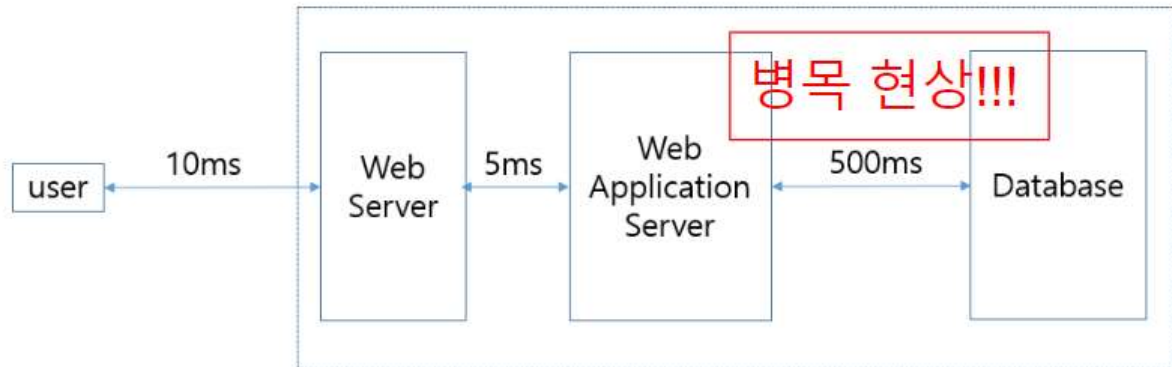
<https://namu.wiki/w/%EC%BA%90%EC%8B%9C%20%EB%A9%94%EB%AA%A8%EB%A6%AC>

- 웹 어플리케이션 전반에 걸친 캐싱 정리

<https://opentutorials.org/course/697/3839>

2) 공통 프로젝트 시스템 아키텍처 구성도

공통 프로젝트 시스템 아키텍처를 그리고 사용자가 많아지면서 http 요청이 기하 급수적으로 증가했을 경우 가장 큰 병목이 예상되는 구간을 표시합니다. 그리고 선정 이유를 기재합니다. 아래 그림은 일반적인 웹 서비스의 구조로 참고해주시기 바랍니다.



3) 캐시 솔루션 'Redis' 적용

공통 프로젝트 backend 어플리케이션에 직접 프로그래밍하여 간단하게 구현 할 수도 있으나 오픈 소스로 현업에서도 많이 사용되고 있는 redis를 활용합니다. 아래 링크는 backend 프레임워크에서의 redis 사용법을 포함하고 있습니다.

- Redis ? : <https://engkimbs.tistory.com/869>
- 윈도우 Redis 설치: <https://github.com/microsoftarchive/redis/releases/tag/win-3.2.100>
- Spring 연동 : <https://yonguri.tistory.com/82>
- Django 연동 : <https://lee-seul.github.io/django/2019/05/02/django-cache-framework.html>
- Node.js 연동 : <https://bcho.tistory.com/1098>

2)번 과제에서 선정한 병목 예상 서비스 중 rest api 하나를 선정하여 redis 캐싱을 적용하도록 합니다.

3. 심화 과제

- 1) JMeter 등의 성능 측정 도구로 캐시 적용 전후의 응답 시간 또는 TPS 등 성능 지표를 측정한 후 캡처본을 제출하도록 합니다.

4. 산출물 제출

1) 캐시에 관한 개인 학습 내용 정리

일반적인 캐시의 정의 및 장단점에 대해 학습하신 내용을 정리합니다. 분량 제한은 없으나 내가 이해한 내용을 정리하도록 합니다.

2) 공통 프로젝트 시스템 아키텍처 구성도

공통 프로젝트 시스템 아키텍처를 그리고 병목이 예상되는 구간을 표시하고 선정 이유를 기재합니다.

3) 캐시가 적용된 공통 프로젝트 소스

Backend 프로젝트에 캐시를 적용합니다. 캐시 대상으로 가장 적합하다고 판단되는 하나의 서비스를 선정하여 적용합니다. 캐시 적용으로 인해 변경된 소스 파일 내용을 문서에 기재하여 제출합니다.

4) [심화 과제(선택)] JMeter 등의 성능 측정 도구를 활용한 캐시 적용 서비스 성능 비교

심화 과제 결과물은 성능 측정 도구로 캐시 적용 전후의 응답 시간 또는 TPS 등 성능 지표를 측정 한 후 캡처본을 제출하도록 합니다.