[양식 4-0] 과제 중간보고서

[캡스톤디자인 과제 중간보고서]

과 제 명	Microservice Architecture에서 L/B, autoscaling algorithm에 따른 성능 평가

1. 과제 개요

가. 과제 선정 배경 및 필요성

Microservice Architecture로 서버를 개발할 때, 다양한 load balancing, autoscaling 방법들이 존재한다. 이때, 프로그램의 연산량, 동기/비동기 처리, I/O, 동시 처리 등 프로그램의 특성에 따라 효율이 프로그래밍 언어나, L/B 방법에 따라 다르다. 따라서, 본 연구를 통해 Microservice Architecture로 서버를 개발할 때 L/B 방법 및 프로그래밍 언어 선택에 있어 선택의 기준을 제시하도록 한다.

나. 과제 주요내용

- Variables (server): I/O 연산량, 동시 요청 수, 컨테이너의 크기, microservice의 수
- Variables (L/B): thread isolation, L/B rule, programming language

위 내용들에 대해 지연 시간, 자원 사용량, 최대 동시 처리 가능 요청 수들을 측정한다.

다. 최종결과물의 목표

상황에 따른 가장 최적화된 프로그래밍 언어 및 L/B 설정 값들을 알 수 있다.

2. 과제 수행방법

변수를 바꿔가며 Apache JMeter로 성능을 측정하고, 최대 연결 가능한 노드 수와 지연시간을 기록한다.

3. 진행내용

가. 과제진행 내용

- Netflix Zuul, Eureka을 Docker compose로 환경 구성 완료
- 위 환경에서 Node.js 서버에 대한 1차 성능 측정

나. 진행내용의 주요특징 및 설명

- Netflix에서 제작한 API registry인 eureka와 api gateway인 Zuul은 Java Spring으로 작성되었으며 Netflix Ribbon과 함께 다양한 L/B 방법과 설정을 지원한다.
- 연결된 노드 수에 따라 비례하여 성능이 하락하지 않고, 일정 구간에서 갑자기 성능이 저하되는 구간이 존재한다.
- 현재 설정한 Load Balancing 기법으로는 트래픽이 급증한 경우 Load balancing이 제대로 되지 않아 Hystrix 및 Ribbon을 사용하여 트래픽이 잘 분산될 수 있도록 추가 설정이 필요하다.

4. 향후계획

- 다른 programming langugage도 소스 코드를 작성하여 테스트 한다.
- 현재 L/B 세팅에 Hystrix, Ribbon을 추가하여 L/B 성능을 높인 뒤, 설정 값을 바꾸면서 성능을 측정한다.

팀 학생대표 성명 : 차 수환 (인)