Kubernetes를 활용한 영상 기반 멤버 검증 어플리케이션의 분산 배치 기법

*김영기 **금승우
전자부품연구원
*youngkee.kim@keti.re.kr

Kubernetes Microservices for Video-based Member Verification Application

*Kim, Young-kee **Kum, Seung-woo
Korea Electronics Technology Institute

요약

중앙 집중형 구조로 인터넷을 통해 온디맨드 컴퓨팅 리소스를 제공하는 클라우드 컴퓨팅 기술이 범용화 됨에 따라, 다양하고 높은 성능의 컴퓨팅 자원을 사용하는 어플리케이션이 늘고 있다. 하지만 특정 어플리케이션은 인터넷을 이용한 중앙 집중형 구조인 클라우드 컴퓨팅 자원을 사용하는 경우 서비스 품질에 영향을 받을 수 있다. 본 연구는 영상 기반 멤버 검증 어플리케이션의 운용에 있어 영상 데이터의 방대한 크기에 따른 지연시간, 네트워크 병목현상 및 영상에 포함된 얼굴 이미지로 인한 개인신상정보 관련 문제 등을 완화하기 위한 마이크로서비스화 및 분산 배치 기법을 보인다. 또한 이 멤버 검증 어플리케이션의 분산 배치 기법을 적용하여 Docker 컨테이너 단위 마이크로서비스의 배포, 스케일링, 운영을 자동화하기 위한 오픈소스 플랫폼인 Kubernetes를 활용하여 구현함으로써 검증하였다.

1. 서론

디바이스의 컴퓨팅 리소스 성능이 향상됨과 동시에 클라우드 컴퓨팅 기술의 적용이 범용화 되면서, 기계학습 모델의 연산 등 높은 성능의 컴퓨팅 자원을 필요로 하는 서비스를 제공하는 다양한 어플리케이션이 등장할 수 있게 되었다.

그러나 이미지, 영상 등 방대한 크기의 데이터를 처리/연산하는 어플리케이션의 경우, 고성능 컴퓨팅 자원의 사용만을 위해 중앙 집중형 구조인 클라우드 컴퓨팅 기술을 도입하는 것은 영상 데이터의 방대한 크기로 인한 긴 지연시간 및 네트워크 병목현상, 영상 데이터에 포함 될 수 있는 개인정보 관련 문제 등을 야기할 수 있다. 상기의 문제점을 완화하기 위한 방안으로, 다양한 단말 기기에서 발생하는 데이터를 클라우드와 같은 중앙 집중식 데이터센터로 보내지 않고 데이터가 발생한 현장 혹은 근거리에서 실시간 처리하는 방식으로 데이터 흐름 가속화를 지원하는 컴퓨팅 기술인 에지 컴퓨팅 기술에 대한 연구가 활발하게 진행되고 있다[1].

본 연구는 이러한 흐름에 따라 영상 데이터를 입력으로 받아 기계학습 모델 추론을 통해 분석한 후 결과를 도출해내는 영상 기반 멤버 검증 어플리케이션의 마이크로서비스화 및 배치 기법을 제시하며, 상대적으로 제한된 컴퓨팅 리소스 환경인 에지 컴퓨팅 기술 적용에 적합한 가볍고 빠른 동작의 장점을 가진 Docker 컨테이너[2]와, 이 컨테이너들의 배포 및 관리, 컴퓨팅 리소스의 모니터링에 용이한 Kubernetes 플랫폼[3]을 활용하여 실제 구현한 예를 함께 제시한다.

* 교신저자(Corresponding Author)
김영기 : 전자부품연구원
※ 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 1711075689, AI 어플리케이션을 지원하는 IoT 연동 분산 Edge 클라우드 기술 개발)

2. 본론

영상 기반 멤버 검증 서비스 어플리케이션은 카메라에서 들어오는 입력 영상 스트림으로부터 전처리를 거쳐 프레임을 얻어내고, 해당 프레임에 포함된 얼굴 이미지를 잘라낸 다음, 그 얼굴 이미지를 이용해 해당 얼굴의 특징 지도(feature map)를 얻어낸다. 그리고 이 특징 지도를 이용해 특정 그룹에 포함된 사람의 얼굴인지 아닌지를 최종적으로 판단한다. 이 과정에서 기계학습 추론 모델과 의존성 패키지들을 위한 저장 공간, 합리적인 추론 연산 시간을 보장하기 위한 GPU(Graphics Processing Unit) 등의 컴퓨팅 리소스가 요구되므로 이를 위해 클라우드 상에 배포하는 것이 가장 손쉬운 방법일 수 있다.

그러나 영상 기반 멤버 검증 서비스 어플리케이션이 컴퓨팅 리소스의 보장을 위해 클라우드 상에 배포될 경우, 영상 데이터의 근원지인 카메라로부터의 물리적인 거리가 멀어지게 됨에 따라 방대한 크기로 인한 지연시간, 네트워크 대역폭 제약 등의 문제는 물론, 사용자의 얼굴 이미지가 포함된 영상 데이터가 그대로 클라우드 상의 서버로 전달되어 개인정보 관련 문제가 발생할 수 있다.

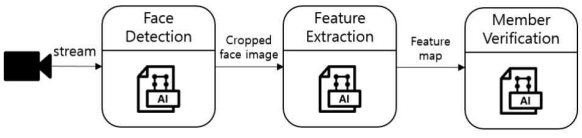

그림 1. 영상 기반 멤버 검증 어플리케이션 구성

그림 1 은 상기의 문제점을 완화하기 위해 에지 컴퓨팅 기술을 반영하여 영상 기반 멤버 검증 어플리케이션을 마이크로서비스 단위로 구분한 구조이다.

카메라로부터 영상 스트림을 직접 받아 프레임을 얻어내고 얼굴 이

미지를 찾아내는 얼굴 감지(Face Detection) 마이크로서비스는 영상 스트림을 직접 받아야하기 때문에 영상 데이터의 근원지인 카메라에 물리적으로 가까운 에지노드에 배포될 수 있도록 한다. 또한 프레임 전체를 연산하는 AI 모델을 사용하므로 고성능 컴퓨팅 리소스인 GPU를 활용할 수 있는 에지노드가 필요하다.

얼굴 감지 마이크로서비스로부터 찾아낸 얼굴 이미지를 입력으로 받아 특징 지도(feature map)를 뽑아내는 기계학습 모델의 추론을 수행하는 특징 추출(Feature Extractor) 마이크로서비스는 얼굴 감지 마이크로서비스 보다는 요구되는 컴퓨팅 리소스의 사양은 낮지만, 기본적으로 이미지를 처리하는 연산이기에 고성능 컴퓨팅 리소스인 GPU를 활용할 수 있는 에지노드에 배포되도록 한다.

특징 추출 마이크로서비스에서 뽑아낸 특징 지도를 입력으로 받아 기계학습 모델의 추론을 통해 최종 멤버 여부를 판단하는 멤버 검증(Member Verification) 마이크로서비스는 입력으로 받는 특징 지도의 크기가 상대적으로 작고 연산 모델 또한 상대적으로 단순하여, 다른 마이크로서비스들과는 다르게 GPU를 활용할 수 있는 에지노드에 반드시 배포될 필요는 없다.

제안된 마이크로서비스 구조는 영상 데이터의 근원지에 보다 가까운 물리적 장치에 얼굴 감지 마이크로서비스를 배포하여 영상 데이터의 크기로 인한 지연시간 및 네트워크 병목현상을 완화할 수 있도록 하고, 특징 추출 마이크로서비스의 출력인 특징 지도(feature map) 부터는 영상에 포함된 얼굴 이미지 복원이 어려우므로 특징 추출 마이크로서비스를 적절한 장치에 배포하여 개인정보 관련 문제를 완화할 수 있도록 한다.

3. 구현

앞서 설명한 에지 컴퓨팅 기술을 반영한 마이크로서비스 단위 구성에 따라 Docker 컨테이너와 Kubernetes 플랫폼을 활용한 영상 기반 멤버 검증 어플리케이션의 구현 및 배포를 진행하였다.

이를 위하여 독립된 Kubernetes 클러스터를 구성하였으며, 해당 클러스터는 1개의 master 노드와 4개의 worker 노드로 구성된다. 각 노드들은 i5 CPU, 16GB memory 등의 컴퓨팅 리소스를 공통으로 가지며 worker 노드는 GPU가 없는 노드, Nvidia GTX-1660ti/1080ti, RTX-2080ti 등 컴퓨팅 리소스에 차등을 두어 구성했다.

얼굴 감지, 특징 추출, 멤버 검증 마이크로서비스 모두 각각의 기계학습 추론 모델 연산이 동반되므로 각 마이크로서비스 컨테이너는 기계학습 연산 라이브러리인 Tensorflow를 포함하고 있으며, 이 중 얼굴 감지 마이크로서비스 컨테이너는 영상 데이터의 전처리를 위한 OpenCV 라이브러리 또한 포함한다. 아래의 그림 2 는 얼굴 감지 마이크로서비스 컨테이너의 구성을 보인다.

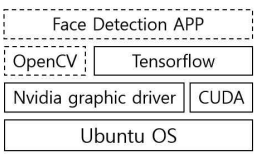

그림 2. 얼굴 감지(Face Detection) 컨테이너 구성

영상 데이터를 직접 입력으로 받아 처리하고 기계학습 추론 모델의 연산이 무거운 얼굴 감지(Face Detection) 마이크로서비스와 감지한 얼굴 이미지를 연산하여 특징 지도를 출력하는 특징 추출 마이크로서비스는 합리적인 추론 연산 시간을 보장하기 위해 고성능 컴퓨팅 리소스인 GPU를 보유한 디바이스에 배포되어야 한다. 이를 위해 클러스터 상의 물리적 노드 중 GPU를 보유한 노드를 식별, 작동 가능하도록 하는 Nvidia device plugin for Kubernetes[4]를 클러스터에 적용하였다. 아래 그림 3 은 GPU 리소스 명세를 포함한 얼굴 감지 마이크로서비스의 Kubernetes 배포용 명세 내용이다.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: face_detection
spec:
  template:
    spec:
      containers:
      - name: face_detection_app
        image: keticmr.mynetgear.com/face_detection
        resources:
          limits:
            nvidia.com/gpu: 1 # require GPU
```

그림 3. 얼굴 감지(Face Detection) deployment 명세

상대적으로 작은 크기의 데이터인 특징 지도를 입력으로 받고 기계학습 추론 모델의 연산 또한 가벼운 멤버 검증(Member Verification) 마이크로서비스는 GPU 리소스가 필수적이지 않기 때문에 GPU 리소스 관련 명세는 필요하지 않다.

상기의 마이크로서비스 구성이 클러스터에 배포되었으며, 최대 3.5GB의 이미지 크기를 가지는 컨테이너에 대하여 최대 200sec의 배포 지연시간이 확인되었다.

4. 결론

본 논문에서는 카메라로부터 영상 데이터를 입력으로 받아 영상에 포함된 사람의 얼굴이 허가된 그룹의 멤버인지 여부를 판단하는 영상 분석 어플리케이션을 제한된 컴퓨팅 리소스 환경에서 영상 데이터의 특성과 기계학습 추론의 연산 부하에 따른 지연시간 및 네트워크 병목현상, 개인정보 문제 완화를 목적으로 에지 컴퓨팅 기술을 적용한 마이크로서비스 단위로 구성하는 기법을 제시하였다. 또한 제시한 구조를 기반으로 Docker 컨테이너, Kubernetes 플랫폼을 활용하여 구현, 배포를 진행하였다.

향후에는 개선된 부분들에 대한 정량적 수치를 측정하기 위한 연구와 함께 마이크로서비스를 특정 장치와 물리적으로 가까운 위치에 배포할 수 있는 방안에 대한 연구를 진행할 수 있을 것이다.

5. 참고 문헌

[1] TTA, 에지 컴퓨팅(Edge Computing), http://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=165974-5

[2] Docker, https://docs.docker.com

[3] Kubernetes, https://kubernetes.io

[4] Nvidia device plugin for Kubernetes, https://github.com/NVIDIA/k8s-device-plugin