RPC 프로토콜을 활용한 미디어 분석 엣지 컨테이너 원격 제어 시스템

오승택, *문재원, **금승우
한국전자기술연구원
stoh@keti.re.kr, *jwmoon@keti.re.kr, **swkum@keti.re.kr

Edge Container Remote Control System using RPC protocol

Seungtaek Oh, *Jaewon Moon, **Seungwoo Kum
Korea Electronics Technology Institute

요 약

고성능 컴퓨팅 기술과 딥 러닝 기술이 충분한 발전을 거쳐 인공지능 기술은 다양한 분야에서 실제로 적용되고 있다. 인공지능 플랫폼 기술이 사용자에게 적절하게 활용되기 위해서 엣지 컴퓨팅 기반의 마이크로 서비스 아키텍처(MSA)가 주목받고 있다. 이와 관련된 기술을 통해 클라우드 기반의 여러 인공지능 애플리케이션들이 엣지 장치에서 직접 처리가 가능하다면 비용적인 측면뿐 아니라 여러 관점에서 효율적이므로 엣지 컨테이너의 운용 기술에 대한 수요가 높아지고 있다. 이에 따라, 본 논문에서는 엣지 디바이스에 간단한 딥 러닝 서비스를 배포하고 운용할 수 있는 컨테이너를 구현하였다. 또한, REST 통신 방법 이외에 RPC 방식을 사용하여 원격 제어를 가능하게 하도록 구성하였으며, 여러 제어 기능들이 동작함을 확인하였다.

1. 서론

고성능 컴퓨팅과 딥 러닝 기술들이 발전함으로써 인공지능 기술은 어느 정도 상용화될 수준으로 상향되었다. 그에 따라 실제 사용자의 요구에 맞추어 현장에서 인공지능 기술이 알맞게 적용되기 위해서 성능적인 측면이 아닌, 인공지능 기술을 효과적으로 전달해 줄 수 있는 엣지 컴퓨팅 기술의 수요가 늘고 있다[1].

엣지 컴퓨팅은 클라우드 기반의 컴퓨팅 아키텍처에서 서비스 품질의 저하를 극복하기 위해 제안된 기술 중 하나이다. 기존의 엣지의 기능인 통신과 저장소 기능뿐만 아니라 분석과 같은 큰 연산 처리가 요구되는 과정도 기능에 포함된다. 따라서 클라우드 상에서의 모델 학습 및 추론 과정이 엣지 컴퓨팅을 통하여 CPU, 메모리, GPU 와 같은 컴퓨팅 자원을 가진 엣지 장치에서 자체적으로 처리가 가능하다. 이 근접성을 기반으로 클라우드와의 데이터 전송에 있어 지연을 줄일 수 있으며, 클라우드 서비스의 비용적 측면에서도 큰 이점이 있다. 단순히 센서의 데이터가 아닌 영상 스트리밍 데이터나 그 이상의 데이터가 클라우드상에서 분석되기 위해서는 통신을 위해 큰 네트워크 비용이 발생하지만, 엣지 장치에서 직접 분석된다면 이 비용을 크게 줄여 사용자에게 큰 이점을 주어 수요가 늘어날 것이다[2].

하지만 엣지 장치는 클라우드에 비해 컴퓨팅 자원이 제한적이고 확장성이 없기 때문에 딥 러닝 모델을 엣지 장치에 배포하기 위해서는 배포 전에 엣지 장치의 사양에 맞추어 모델을 자유롭게 운용하고 최적화할 수 있는 마이크로 서비스 아키텍처(MSA) 기능을 필요로 한다. 이를 위해 본 논문에서는 RESTful 프로토콜 방식의 통신을 통해 여러 엣지 노드 간의 모델 배포와 운용을 기반으로, 좀 더 효율적인 서비스와 디바이스 간 상호 통신을 위해 RPC 프로토콜을 적용한 시스템을 구현하였다.

2. 본론

2.1. 컨테이너 배포 환경

딥 러닝 모델을 배포 및 운용하는 환경을 만들기 위해 도커 컨테이너를 활용하여 마이크로 서비스 아키텍처를 구성하였다. 여러 형태의 엣지 노드에 배포하기 위해 그림 1 과 같이 CPU, GPU (2080Ti). Xavier 총 3 종 노드로 구성하였으며, Yolo v4 기반의 object detection, tracking 모델을 배포하는 컨테이너를 사용하였다.

그림 1. 컨테이너 배포 환경

2.2. RPC (Remote Procedure Call)

여러 엣지 노드 간의 데이터 전송 및 모델 배포를 위해서 여러 통신 방식들이 존재한다. 배포 환경에서 기본적인 통신 방법인 REST 방식을 통해 서비스를 구성하였으나, 확장성 및 생산성과 효율적인 유지보수를 위해 RPC 프로토콜을 지원하는 MSA 시스템을 추가로 구성하였다.

그림 2 와 같이 RPC 프로토콜은 프로세스 간 통신을 위해 사용하는 IPC (Inter Process Communication) 방법의 한 종류로, 원격지의 프로세스에 접근하여 프로시저 또는 함수를 호출하여 사용하는 방법이다. 그 중 gRPC 는 Google 사의 범용 RPC 인터페이스를 크로스 플랫폼, 오픈소스화하여 개발한 높은 성능의 범용 RPC 프레임워크로 분산 네트워크 환경에서 조금 더 쉽게 구현할 수 있도록 등장하였다.

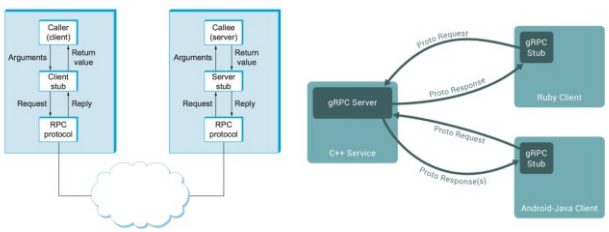

그림 2 . RPC 통신 과정[3] 및 gRPC 구조[4]

gRPC 의 장점으로는 높은 생산성과 효율적인 유지 보수를 가능하게 한다. IDL(Identity Definition Language) 만 정의하면 높은 성능을 보장하는 서비스와 메시지에 대한 소스 코드가 다양한 언어에 맞게 자동 생성되기 때문에 클라이언트, 서버 간 사용 언어에 의존하지 않아 의사소통 비용이 감소한다. 또한, 내부적으로 HTTP/2 를 사용하여 메시지 압축률이 높으며 양방향 스트리밍 통신이 가능하다.

기능	gRPC	HTTP with JSON
프로토콜	HTTP/2, 빠름	HTTP, 느림
Payload Data	Protobuf	JSON
점검 형식	엄격함 (명확한 데이터 타입 정의)	느슨함
브라우저 지원	X (BloomRPC 같은 tool 필요)	O
스트리밍	양방향	단일 방향

그림 3. gRPC 와 HTTP API 간 비교[5]

하지만, 계약 우선 접근 방식을 따라 통신하고자 하는 데이터 타입을 명확하게 정의해야 하며, 점검 형식이 매우 엄격하다. 또한, 브라우저와 서버 간의 gRPC 통신이 지원되지 않아, grpc-gateway 를 통해 데이터 직렬화 방식인 Protocol Buffer 형태로 데이터를 변환한 뒤에 사용할 수 있다. 바이너리 형태의 Data 는 BloomRPC 와 같은 tool 을 통해 gRPC 의 테스트를 진행해야 한다. 그럼에도 이러한 단점들을 극복할 만큼 넓은 확장성과 다양한 장점들을 가진 gRPC 프로토콜을 통해 엣지 노드에서의 제어가 가능한 서비스를 구현하였다.

3. 실험

엣지에 배포하는 딥 러닝 모델의 운용 능력을 검증하기 위해서 스마트 팜 환경에서의 돼지 영상 데이터를 사용하여 돼지를 detection 하고, tracking 하는 컨테이너를 구성하였다. 그림 4 와 같이 유선 공유기를 통해 GPU 가 없는 PC 로부터 GPU 가 장착된 PC 노드에 분석을 요청하여 그 결과를 전달받는 형태로 RPC 프로토콜 통신 환경을 구축하였다. 딥 러닝 모델로는 Darknet 기반의 YOLOv4[6]와 DeepSORT[7]를 사용하였으며, gRPC 기반의 API 로는 입력 영상의 주소 입/출력, 결과 영상의 주소 입/출력, 연산(분석) 총 5 개의 항목으로 구성하였다.

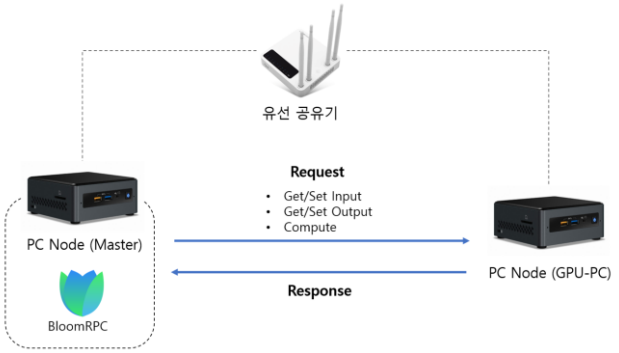

그림 4. RPC 를 활용한 통신 환경

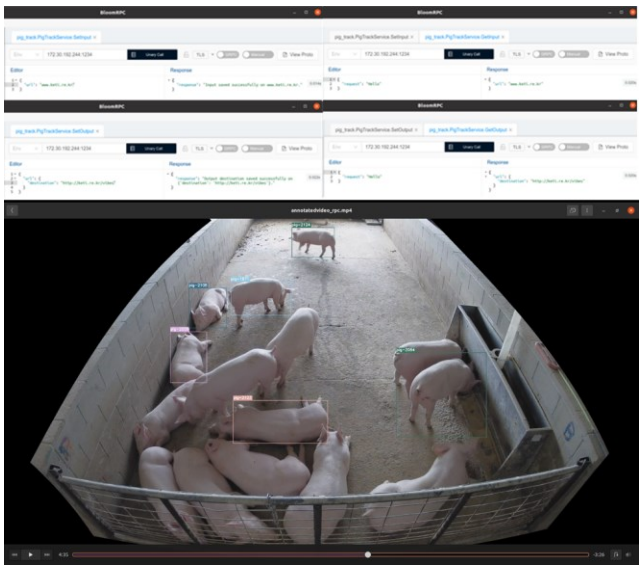

그림 5. gRPC 프로토콜을 통한 컨테이너 제어 결과

결과적으로 그림 5 와 같이 BloomRPC 라는 클라이언트 시각화 도구를 활용하여 입력 영상 및 결과 영상의 주소 입/출력과 (Get/Set input, Get/Set output), 분석 (Compute) 총 5 개의 API 를 통해 위와 같은 결과를 확인했으며, gRPC 프로토콜을 사용하여 엣지 컨테이너를 원격에서 적절하게 제어함을 확인하였다.

4. 결론

본 논문에서는 엣지 컴퓨팅을 위한 다양한 프로토콜 방식 중 gRPC 를 통해 엣지 컨테이너를 운용하는 서비스를 구현하였다. 엣지에 배포하는 딥 러닝 모델의 운용 능력을 검증하기 위해서 스마트 팜 환경에서의 돼지 영상 데이터를 사용하여 돼지를 detection 하고, tracking 하는 컨테이너를 구성하였으며, 여러 API 를 통해 적절하게 운용할 수 있음을 확인하였다. 또한, 확장성과 범용성이 뛰어난 gRPC 방식을 사용하여 REST 방식의 단점인 통신의 단방향성 및 지연 문제를 극복하여 통신의 효율성을 높일 수 있었다. 이후에는 더 많은 엣지와 기능을 추가하는 등 여러 작업을 통해 서비스를 확장할 수 있을 것으로 기대된다.

Acknowledgement

본 연구는 산업통상자원부와 한국산업기술진흥원의 “국제공동기술개발사업”의 지원을 받아 수행된 연구결과임. (과제번호: P0011948)

참고문헌

[1] H. Li et al., “Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing”, *IEEE Network*, 2018

[2] P. Mendki, “Docker container based analytics at IoT edge Video analytics usecase”, *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2018

[3] RPC: https://book.systemsapproach.org/e2e/rpc.html

[4] gRPC: https://grpc.io/docs/guides/

[5] gRPC vs. HTTP API: https://docs.microsoft.com/ko-kr/aspnet/core/grpc/comparison?view=aspnetcore-6.0

[6] A. Bochkovskiy et al., “YOLOv4: Optimal Speed and Accuracy of Object Detection”, *CVPR*, 2020

[7] N. Wojke et al., “Simple online and realtime tracking with a deep association metric”, *ICIP*, 2017