

gRPC 기반 양방향 영상 스트리밍 애플리케이션 개발 및 구현

정민균¹

¹단국대학교 컴퓨터공학과
32194061@dankook.ac.kr

Abstract

2024학년도 단국대학교 컴퓨터공학과의 고급프로그래밍실습 과제에 대한 보고서이다. 개발 시 많이 쓰이는 플랫폼/프레임워크/기법 중 메세지 큐, REST API, gRPC, MySQL, MongoDB/Neo4j의 위 5가지 기술을 각각 이용해 간단한 애플리케이션 개발이 목표이다. 본 보고서는 5가지 기술 중 gRPC를 이용한 어플리케이션에 대한 내용으로 gRPC와 Python을 이용해 양방향 영상 스트리밍 애플리케이션(영상 통화)을 구현해보았다.

1. Introduction

COVID-19의 여파로 대면 수업, 업무가 불가능해지자 Zoom, MS teams, Discord 등 다양한 화상 채팅 플랫폼이 등장했다. 뿐만 아니라, 카카오톡, 라인 등과 같은 메신저 플랫폼 역시 다수의 사람들과 동시에 영상통화가 가능해졌다. 다수의 사용자가 영상통화 혹은 화상 회의를 하기 위해서는 각자의 카메라에 비춰지는 영상을 실시간으로 송출하고 서버에서 수신한 각각의 영상을 다른 사용자에게 전송해야한다. 본 프로젝트에서는 gRPC를 이용해 실시간 영상 통화 애플리케이션 기능을 구현해본다.

2. gRPC 이용 방안

gRPC의 HTTP/2 기반 스트리밍을 활용하여 양방향 데이터 전송 및 처리한다. video_concat.proto 파일에서 정의된 VideoProcessingService를 기반으로 서버-클라이언트 간 인터페이스가 동작한다. 1개의 서버를 두고 4명의 클라이언트가 각자의 영상 프레임을 gRPC를 이용해 'localhost:50051'로 전송한다. 서버는 접속한 클라이언트를 입력받은 client id 순서로 4개의 영상을 2x2 형태로 영상을 붙여 1개의 영상으로 만들어 각 클라이언트들에게 합성한 영상을 전송한다.

- 클라이언트는 client id와 video path를 argument로 받음.
- 각 클라이언트에 argument로 넘어온 영상을 localhost:50051에 jpg 인코딩하여 스트리밍.
- 서버는 클라이언트가 연결되면 스트리밍한 영상을, 연결되지 않으면 검정색 화면을 리스트에 저장함.
- 4개의 클라이언트를 2x2로 concat하여 모든 클라이언트에게 합성된 영상을 전송함.

2.1. Server

주요 메서드로는 StreamFrames과 _concat_frames이 있다. StreamFrames은 클라이언트가 프레임 데이터를 서버로 전송하고, 서버는 합성된 프레임을 다시 클라이언트로 스트리밍하는

역할을 수행한다. StreamFrames 메서드는 양방향 스트리밍 방식으로 구현돼있다. 클라이언트가 지속적으로 프레임을 전송하면 서버는 수신한 클라이언트의 데이터를 처리(2x2 합성)하여 다시 클라이언트로 스트리밍한다.

2.2. Client

grpc.insecure_channel('localhost:50051')로 서버와의 연결을 설정한다. stream_video 메서드의 파라미터로 client의 id 와 송출할 영상을 받는다. StreamFrames 메서드를 사용해 비디오 데이터를 지속적으로 전송한다. response_iterator = stub.StreamFrames(frame_generator()) 받은 응답값을 디코딩한 다음 client id로 이루어진 폴더에 jpg로 저장한다.

3. 설치 및 실행 방법

패키지 업데이트 및 설치와 tmux를 통해 server와 client를 실행하는 스크립트를 실행한다. 모든 서버와 클라이언트를 실행하는 스크립트이다.

- sudo apt-get update
- sudo apt-get install git
- git clone https://github.com/mingun0112/dku_grpc.git or unzip dku_grpc.zip
- cd ./dku_grpc
- sh ./start_grpc.sh

3.1. 실행 방법

서버와 클라이언트를 각각 실행하는 방법에 대한 내용이다.

- shell을 두 개 연다.
- python3 ./server.py
- python3 client.py {client_id} {video_path}
python3 client.py client4 ./video/4.mp4

3.2. 설치 방법 사진

```

 mingun@DESKTOP-UP9LUKD: ~/submit/dku_grpc
mingun@DESKTOP-UP9LUKD: ~/submit$ sudo apt-get update
[sudo] password for mingun:
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 257 kB in 2s (126 kB/s)
Reading package lists... Done
mingun@DESKTOP-UP9LUKD: ~/submit$ sudo apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.11).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
mingun@DESKTOP-UP9LUKD: ~/submit$ git clone https://github.com/mingun0112/dku_grpc.git
Cloning into 'dku_grpc'...
remote: Enumerating objects: 44, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 44 (delta 7), reused 12 (delta 3), pack-reused 25 (from 1)
Receiving objects: 100% (44/44), 66.21 MiB | 11.59 MiB/s, done.
Resolving deltas: 100% (9/9), done.
mingun@DESKTOP-UP9LUKD: ~/submit$ cd dku_grpc/
mingun@DESKTOP-UP9LUKD: ~/submit/dku_grpc$
```

```

mingun@DESKTOP-UP9LUKD: ~/submit/dku_grpc$ sh ./start_grpc.sh
Updating system packages...
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Installing Python and pip...
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04.1).
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tmux is already the newest version (3.2a-4ubuntu0.2).
tmux set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Defaulting to user installation because normal site-packages is not writable
Requirement already satisfied: grpcio in /home/mingun/.local/lib/python3.10/site-packages (from -r requirements.txt (line 1)) (1.68.1)
Requirement already satisfied: grpcio-tools in /home/mingun/.local/lib/python3.10/site-packages (from -r requirements.txt (line 2)) (1.68.1)
Requirement already satisfied: opencv-python in /home/mingun/.local/lib/python3.10/site-packages (from -r requirements.txt (line 3)) (4.10.0.84)
Requirement already satisfied: numpy in /home/mingun/.local/lib/python3.10/site-packages (from -r requirements.txt (line 4)) (2.1.3)
Requirement already satisfied: protobuf<6.0dev,>=5.26.1 in /home/mingun/.local/lib/python3.10/site-packages (from grpcio-tools->-r requirements.txt (line 2)) (5.29.1)
Requirement already satisfied: setuptools in /home/mingun/.local/lib/python3.10/site-packages (from grpcio-tools->-r requirements.txt (line 2)) (75.6.0)
duplicate session: dku_grpc
Starting server...
Starting client...
Starting client2...
Starting client3...
Starting client4...

```

Figure 1: git에서 레포지토리를 클론한 후 디렉토리에 접근하여 패키지를 업데이트 및 설치하고 start_grpc.sh 스크립트를 실행하는 것에 대한 내용

3.3. 실행 결과 사진

```
python3 server.py
mininum@DESKTOP-UP9LUKD:~/submit/dku_grpc$ python3 server.py
Server started on port 9006
Connected clients: ['client1']
Disconnected clients: []
Connected clients: ['client2', 'client3', 'client4']
Connected clients: ['client3', 'client4']
Connected clients: ['client1', 'client2', 'client3']
Connected clients: ['client1', 'client2', 'client3', 'client4']
Disconnected clients: []
mininum@DESKTOP-UP9LUKD:~/submit/dku_grpc$
```

```
Received result 417
Received result 418
Received result 419
Received result 420
Received result 420
Received result 421
Received result 422
Received result 423
Received result 424
Received result 425
Received result 426
Received result 427
Received result 428
Received result 429
Received result 430
Received result 431
Received result 432
mininum@DESKTOP-UP9LUKD:~/submit/dku_grpc$
```

```
Received result 420
Received result 421
Received result 422
Received result 423
Received result 424
Received result 425
Received result 426
Received result 427
Received result 428
Received result 429
Received result 430
Received result 431
Received result 432
mininum@DESKTOP-UP9LUKD:~/submit/dku_grpc$
```

```
Received result 420
Received result 421
Received result 422
Received result 423
Received result 424
Received result 425
Received result 426
Received result 427
Received result 428
Received result 429
Received result 430
Received result 431
Received result 432
mininum@DESKTOP-UP9LUKD:~/submit/dku_grpc$
```

```
Received result 417
Received result 418
Received result 419
Received result 420
Received result 421
Received result 422
Received result 423
Received result 424
Received result 425
Received result 426
Received result 427
Received result 428
Received result 429
Received result 430
Received result 431
Received result 432
mininum@DESKTOP-UP9LUKD:~/submit/dku_grpc$
```

```
Received result 420
Received result 421
Received result 422
Received result 423
Received result 424
Received result 425
Received result 426
Received result 427
Received result 428
Received result 429
Received result 430
Received result 431
Received result 432
mininum@DESKTOP-UP9LUKD:~/submit/dku_grpc$
```

Figure 2: 월)Server에 client의 connect와 disconnect를 표시한다. 우) 4개의 client가 각각에 할당된 영상을 스트리밍한다.



Figure 3: Client 1이 전송한 프레임 중 하나이다.

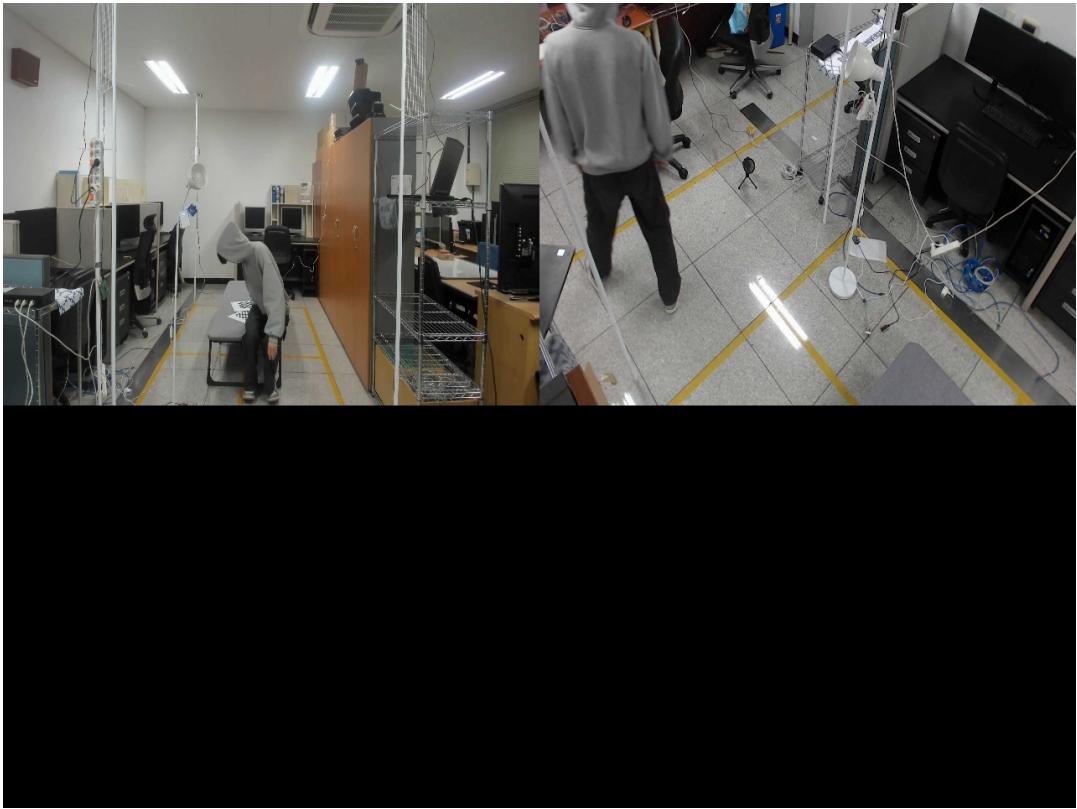


Figure 4: Client 1이 수신한 프레임 중 하나이다. 아직 client 3과 client 4가 연결되지 않은 것을 확인할 수 있다.



Figure 5: Client 1이 수신한 프레임 중 하나이다. 모든 client가 연결된 상태로 싱크가 다른 것은 각자 시간을 두고 하나씩 연결되도록 환경을 구성했기 때문이다.