Vcse-cdn2.0使用E3服务器的环境搭建

1.set proxy

declare -x http\_proxy="http://child-prc.intel.com:913"

declare -x https\_proxy="http://child-prc.intel.com:913"

或者：

export http\_proxy=http://proxy-prc.intel.com:911

export https\_proxy=https://proxy-prc.intel.com:911

2. install docker

apt-get install docker.io

3.Setup docker proxy on E3/E5 host

sudo mkdir -p /etc/systemd/system/docker.service.d

#

printf '[Service]\nEnvironment="HTTPS\_PROXY=http://child-prc.intel.com:913" "NO\_PROXY=hub.docker.intel.com,localhost"\n' | sudo tee /etc/systemd/system/docker.service.d/proxy.conf

4.restart docker

sudo systemctl daemon-reload

sudo systemctl restart docker

5. Build docker images

5.1

apt install cmake

apt install make

mkdir build (vcse-cdn目录下)

cd build

cmake ..

5.2

apt install m4

chmod 755 xcode-server/ffmpeg+vaapi+qsv/build.sh

cd /build/xcode-server/ffmpeg+vaapi+qsv/

make

chmod 755 cdn-server/nginx+rtmp/build.sh

cd /build/ cdn-server/nginx+rtmp/

make

6.Create docker network

docker network create -d bridge --subnet 192.168.31.0/24 --gateway 192.168.31.1 my\_bridge

7.push-stream

7.1

docker run -dit -v /home/common/:/media --device=/dev/dri:/dev/dri --network=my\_bridge --ip 192.168.31.30 --name push-nginx xeone3-ubuntu1804-ffmpeg:1.0 /bin/bash

##

7.2 modify nginx's listen port

docker exec -d push-nginx /bin/bash -c "sed -i '52s#80#410#1' /etc/nginx/nginx.conf"

docker exec -d push-nginx /bin/bash -c "sed -i '12s#1935#2410#1' /etc/nginx/nginx.conf"

7.3

docker exec -d push-nginx /bin/bash -c "nginx &"

8.receive docker

docker run -dit -v /home/common/:/media -p 410:410 -p 2410:2410 --network=my\_bridge --ip 192.168.31.33 --name receive-nginx xeon-ubuntu1804-nginx-rtmp:1.0 /bin/bash

8.Run transcoder server docker instance(pull stream)

Run below command on E3 server:

docker run -dit -v /home/common/:/media --device=/dev/dri:/dev/dri --network=my\_bridge --ip 192.168.31.31 --name xcoder xeone3-ubuntu1804-ffmpeg:1.0 /bin/bash

Run below command on E5 server:

docker run -it --network=my\_bridge --ip 192.168.31.31 --name xcoder -v /var/www/dash:/var/www/dash -v /var/www/hls:/var/www/hls -v /var/www/archive:/var/www/archive xeon-ubuntu1804-ffmpeg:1.0 /bin/bash

9.Run CDN server docker instance

Run below command on E3/E5 server:

docker run -dit -p 2420:2420 --network=my\_bridge --ip 192.168.31.42 --name cdn-nginx42 -v /var/www/dash:/var/www/dash -v /var/www/hls:/var/www/hls -v /var/www/archive:/var/www/archive xeon-ubuntu1804-nginx-rtmp:1.0 /bin/bash

docker exec -d cdn-nginx42 /bin/bash -c "sed -i '12s#1935#2420#1' /etc/nginx/nginx.conf"

Run:

docker exec -d cdn-nginx42 /bin/bash -c "nginx &"

ffmpeg -i rtmp://192.168.31.33:2410/stream/nature -s 4096\*2160 -c:a copy -c:v libsvt\_hevc -preset 10 -maxrate 4000000 -r 30 -f flv rtmp://192.168.31.42:2420/dash/4kStream -s 2048\*1024 -c:a copy -c:v libsvt\_hevc -preset 8 -maxrate 6000 -r 30 -f flv rtmp://192.168.31.42:2420/dash/2kStream -s 1920\*1080 -c:a copy -c:v libsvt\_hevc -preset 8 -maxrate 6000 -r 30 -f flv rtmp://192.168.31.43:2430/dash/1080Stream -s 1080\*720 -c:a copy -c:v libsvt\_hevc -preset 6 -maxrate 4000 -r 30 -f flv rtmp://192.168.31.43:2430/dash/720Stream -s 720\*480 -c:a copy -c:v libsvt\_hevc -preset 4 -maxrate 3000 -r 30 -f flv rtmp://192.168.31.44:2440/dash/480Stream -s 320\*240 -c:a copy -c:v libsvt\_hevc -preset 2 -maxrate 2000 -r 30 -f flv rtmp://192.168.31.44:2440/dash/240Stream

2410

ffmpeg -re -stream\_loop 500 -i Nature.mp4 -c:v copy -an -f flv rtmp://192.168.31.33/stream/nature/

ffmpeg -re -stream\_loop -1 -i Nature.mp4 -c:v copy -an -f flv rtmp://192.168.31.33/stream/nature

ffmpeg -stream\_loop -1 -hwaccel vaapi -hwaccel\_device /dev/dri/renderD128 -hwaccel\_output\_format vaapi -i Nature.mp4 -c:v hevc\_vaapi -f flv rtmp://192.168.31.33:2410/stream/nature -abr\_pipeline

ffmpeg -hwaccel vaapi -hwaccel\_device /dev/dri/renderD128 -hwaccel\_output\_format vaapi -i Nature.mp4 -c:v h264\_vaflv rtmp://192.168.31.50:2500/stream/nature -abr\_pipeline

ffmpeg -hwaccel vaapi -hwaccel\_output\_format vaapi -i Nature.mp4 -c:v hevc\_vaapi -f flv rtmp://192.168.31.33:2410/stream/nature -abr\_pipeline

ffmpeg -hwaccel vaapi -hwaccel\_device /dev/dri/renderD128 -hwaccel\_output\_format vaapi -i rtmp://192.168.31.33:2410/live/nature -c:v h264\_vaapi -f flv rtmp://192.168.31.32/hls/nature -c:v h264\_vaapi -f flv rtmp://192.168.31.32/dash/nature -abr\_pipeline

ffmpeg -hwaccel vaapi -hwaccel\_device /dev/dri/renderD128 -hwaccel\_output\_format vaapi -i rtmp://192.168.31.33:2410/stream/nature -c:v h264\_vaapi -f flv rtmp://192.168.31.42:2420/hls/nature -c:v h264\_vaapi -f flv rtmp://192.168.31.42:2420/dash/nature -abr\_pipeline

Met issues:

### Setup docker proxy on E5 host

root@unassigned-hostname:~/wangjing/OVC# systemctl daemon-reload

root@unassigned-hostname:~/wangjing/OVC# sudo systemctl restart docker

Failed to restart docker.service: Unit docker.service not found.

Issue fix:

apt install docker.io

Docker -v

### Build docker images

On E5 Server, run below command to build docker images:

``` sh

mkdir build

cd build

cmake ..

cd xcode-server/ffmpeg

make

Ctest (with GUI 3/4 pass, one vpx fail; without GUI all fail for without dri device)

### Create docker network

Run below command on E5 server:

```

docker network create -d bridge --subnet 192.168.31.0/24 --gateway 192.168.31.1 my\_bridge

```

### Run transcoder server docker instance

Run below command on E5 server:

```

docker run -it --device=/dev/dri:/dev/dri --network=my\_bridge --ip 192.168.31.31 --name xcoder xeon-ubuntu1804-ffmpeg:1.0 /bin/bash

Run:

####ffmpeg -i rtmp://10.67.117.70/live/nature -c:v libsvt\_hevc -f flv rtmp://192.168.31.32/hls/nature -c:v libsvt\_hevc -f flv rtmp://192.168.31.32/dash/nature

```

[Need Update to Guide]:

With GUI:

docker run -it --device=/dev/dri:/dev/dri --network=my\_bridge --ip 192.168.31.31 --name xcoder xeon-ubuntu1804-ffmpeg:1.0 /bin/bash

Without GUI：

docker run -it --network=my\_bridge --ip 192.168.31.31 --name xcoder xeon-ubuntu1804-ffmpeg:1.0 /bin/bash

### Run CDN server docker instance

Run below command on E5 server:

```

docker run -it --device=/dev/dri:/dev/dri -p 80:80 --network=my\_bridge --ip 192.168.31.32 --name nginx xeon-ubuntu1804-nginx-rtmp:1.0 /bin/bash

Run:

nginx &

```

[Need Update to Guide]:

With GUI:

docker run -it --device=/dev/dri:/dev/dri -p 80:80 --network=my\_bridge --ip 192.168.31.32 --name nginx xeon-ubuntu1804-nginx-rtmp:1.0 /bin/bash

Without GUI:

docker run -it -p 80:80 --network=my\_bridge --ip 192.168.31.32 --name nginx xeon-ubuntu1804-nginx-rtmp:1.0 /bin/bash

[Update] can set two port: one is for rtmp; another is for http which can be configure in nginx.conf

### Install FFmpeg on Streaming Server

Run below command on streaming server:

```

sudo apt-get install -y ffmpeg

```

### Install VLC on client

Run below command on client:

```

sudo apt-get install -y vlc

```

## Sample Execution

### Streaming

Run below command on streaming server:

```

ffmpeg -re -stream\_loop 500 -i Nature.mp4 -c:v copy -an -f flv rtmp://10.67.117.70/live/nature

```

### Live play

Run below command on client:

```

vlc http://10.67.116.179/hls/nature/index.m3u8

vlc http://10.67.116.179/dash/nature/index.mpd

Add Docker Command Line:

[root@localhost ~]# docker ps

[root@localhost ~]# docker start Name

[root@localhost ~]# docker stop Name

[root@localhost ~]# docker restart Name

[root@localhost ~]# docker rm $(docker ps -a -q)