HDDL-S installation guide

Hardware: CoffeeLake, Intel(R) Xeon(R) E-2176G CPU @ 3.70GHz

OS: Ubuntu 16.04.3

1. **Install OpenVINO R5**

Download OpenVINO R5:

<https://software.intel.com/en-us/openvino-toolkit/choose-download/free-download-linux>

Install OpenVINO:

<https://software.intel.com/en-us/articles/OpenVINO-Install-Linux>

Note: below steps is the least

1) Install external software dependencies

### 2) Install the Intel® Distribution of OpenVINO™ toolkit core components

3) Set the environment variables

*source /opt/intel/computer\_vision\_sdk/bin/setupvars.sh*

### 4) Additional installation steps for Intel® Processor Graphics (GPU)

*cd /opt/intel/computer\_vision\_sdk/install\_dependencies*

*sudo ./install\_NEO\_OCL\_driver.sh*

Add OpenCL users to the video group:

*sudo usermod -a -G video USERNAME*

e.g. if the user running OpenCL host applications is foo, run: sudo usermod -a -G video foo

Install 4.14 kernel using install\_4\_14\_kernel.sh script and reboot into this kernel

*sudo ./install\_4\_14\_kernel.sh*

If you use 8th Generation Intel processor, you will need to add:

i915.alpha\_support=1

to the 4.14 kernel command line, in order to enable OpenCL functionality for this platform.

### 5) Additional Installation Steps for Intel® Vision Accelerator Design with Intel® Movidius™ VPUs

1. **Install OpenCl SDK**

*sudo apt-get install dkms*

*tar -xvf intel\_sdk\_for\_opencl\_2017\_7.0.0.2568\_x64.gz*

*cd intel\_sdk\_for\_opencl\_2017\_7.0.0.2568\_x64*

*./install\_GUI.sh*

1. **Install dependencies**

**3.1 dependency packages**

*sudo apt-get install libelf-dev*

*sudo apt-get install libusb-1.0-0-dev libudev-dev libssl-dev rpm cmake libboost-program-options1.58-dev libboost-thread1.58 libboost-filesystem1.58 git*

**3.2 Install json-c**

*git clone https://github.com/json-c/json-c.git*

*cd json-c*

*git checkout f8c632f579c71012f9aca81543b880a579f634fc*

*sudo apt-get install autoconf libtool*

*sh autogen.sh*

*./configure*

*make*

*sudo make install*

1. **Install gstreamer**

*sudo apt-get install gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-plugins-ugly gstreamer1.0-plugins-bad libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev*

*export PKG\_CONFIG\_PATH=$PKG\_CONFIG\_PATH:/opt/intel/mediasdk/lib64/pkgconfig*

1. **Install uWebSocket**

*sudo apt-get install libssl-dev*

*git clone* [*https://github.com/uNetworking/uWebSockets.git*](https://github.com/uNetworking/uWebSockets.git)

*cd uWebSocket && git checkout 1a126c8d4eb9127c2689ede9a8fdf8abcb62b981*

*make && sudo make install*

1. **Install OpenCV**

*sudo apt-get install libgtk2.0-dev pkg-config libgtkglext1-dev*

*export CPLUS\_INCLUDE\_PATH=/opt/intel/mediasdk/include:$CPLUS\_INCLUDE\_PATH*

*git clone https://github.com/opencv/opencv.git*

*cd opencv && git checkout 6ffc48769ac60d53c4bd1913eac15117c9b1c9f7*

*mkdir build && cd build*

*cmake -DWITH\_VA\_INTEL=ON -DWITH\_IPP=OFF -DWITH\_CUDA=OFF ..*

*note: make sure:*

*-- VA: YES*

*-- Intel VA-API/OpenCL: YES (OpenCL: /opt/intel/opencl)*

*If not, there is WA:*

*sudo ln -sf /opt/intel/opencl/SDK/include /opt/intel/opencl/include*

*sudo ln -sf /opt/intel/mediasdk/include/va /usr/local/include/va*

*make -j8*

*sudo make install*

*Note: OpenVINO has provided OpenCV libraries, but HDDL-S need VA support in OpenCV, so we must rebuild it.*

1. **Build HDDL-S software**

**7.1 download source code**

*git clone* [*git@gitlab-icv.inn.intel.com:hddl/s\_framework.git*](mailto:git@gitlab-icv.inn.intel.com:hddl/s_framework.git)

**7.2 build MSDK gst-plugin**

*sudo apt-get install libdrm-dev libudev-dev libgstreamer-plugins-bad1.0-dev libx11-xcb-dev libgles2-mesa-dev libgl1-mesa-dev*

*sudo ln -sf /opt/intel/mediasdk/lib64/libva.so.2 /usr/lib/libva.so*

*sudo ln -sf /opt/intel/mediasdk/lib64/libva-drm.so.2 /usr/lib/libva-drm.so*

*cd gstreamer-media-SDK && mkdir build && cd build && cmake ..*

*make*

*sudo cp build/lib/release/libgstmfx.so /usr/lib/x86\_64-linux-gnu/gstreamer-1.0/libgstmfx.so*

* 1. **build OpenVINO gst-plugin**

*sudo apt-get install libeigen3-dev libopenblas-dev liblapack-dev libdlib-dev*

*cd gstreamer\_plugin\_openVINO*

*make -j8*

*sudo make install*

Run below command to check if it has been installed successfully:

*gst-inspect-1.0 cvdlfilter*

*gst-inspect-1.0 resconvert*

*gst-inspect-1.0 wssink*

* 1. **Install hddlspipe**

*cd gstreamer\_pipeline*

*make && sudo make install*

1. **Setup HDDL-S Server**

*sudo apt-get install nodejs-legacy npm*

*npm config set proxy http://child-prc.intel.com:913*

*sudo npm install -g n*

*sudo n stable*

*cd s\_framework/hddls\_server\_controller\_receiver*

*npm install*

1. **Run HDDL-S Software Stack**

**9.1 Set environment variable**

Modify ~/.bashrc and add below command lines into it:

*-----------------------------------------------------------------------------------------------------------------------------*

*source /opt/intel/computer\_vision\_sdk/bin/setupvars.sh*

*export LD\_LIBRARY\_PATH=/usr/local/lib:/opt/intel/computer\_vision\_sdk\_2018.5.445/deployment\_tools/inference\_engine/external/hddl/lib:$LD\_LIBRARY\_PATH*

*export PKG\_CONFIG\_PATH=$PKG\_CONFIG\_PATH:/opt/intel/mediasdk/lib64/pkgconfig*

*export LD\_LIBRARY\_PATH=/opt/intel/mediasdk/lib64:/opt/intel/computer\_vision\_sdk/inference\_engine/lib/ubuntu\_16.04/intel64:/opt/intel/computer\_vision\_sdk\_2018.5.445/deployment\_tools/inference\_engine/external/omp/lib:/usr/lib/x86\_64-linux-gnu/gstreamer-1.0:$LD\_LIBRARY\_PATH*

*export HDDLS\_CVDL\_KERNEL\_PATH=/usr/lib/x86\_64-linux-gnu/libgstcvdl/kernels*

*export PATH=$PATH:/opt/intel/mediasdk/bin/*

---------------------------------------------------------------------------------------------

Run .bashrc

*source ~/.bashrc*

**9.2 Generate certificate**

Please generate all these certificates in one pc!!!

Please refer to hddls\_server\_controller\_receiver/certificate\_create\_explanation.md.

After generated

Copy "ca-crt.pem client1-crt.pem client1-key.pem" into 'controller/cert\_client\_8216\_8124'

Copy "ca-crt.pem client1-crt.pem client1-key.pem" into 'receiver/cert\_client\_8216\_8124'

Copy "ca-crt.pem server-crt.pem server-key.pem" into 'server/cert\_server\_8216\_8124'

Copy 'client1.crl' into 'cert\_server\_8216\_8124'

Copy 'server.crl' into 'cert\_client\_8216\_8124'

**9.3 Run HDDL-S server**

*cd s\_framework/* hddls\_server\_controller\_receiver */server*

*export HDDLS\_CVDL\_MODEL\_PATH=<hddl-s\_server\_folder>/models*

*node hddls-server.js*

**9.3.1 How to deploy customer models**

Step 1: implement libxxxalgo.so as customer guide

Step 2: copy model IR files into <*HDDLS\_CVDL\_MODEL\_PATH*>/<model\_name>

Step 3: register this customer models

registeralgo -a <model\_name>

Step 4: edit create\_xxx.json and add <model\_name> into algopipeline property

Step 5: controller\_client send create pipeline command with create\_xxx.json

**9.3.2 Some tips**

1. Dont't remove the files in hddls\_server\_controller\_receiver */server*/models

The hddlspipe will read models file from this directory.

2. Maybe it will fail to run h265 video stream, it was caused by cannot find libmfx\_hevcd\_hw64.so in /opt/intel/mediasdk/lib64/mfx/

It was mediasdk driver cfg issues.

There is a WA:

*#cd /opt/intel/mediasdk/lib64*

*#sudo ln -sf ../plugins mfx*

**9.4 Run HDDL-S clients**

**9.4.1 Run receiver client**

*cd s\_framework/*hddls\_server\_controller\_receiver*/receiver*

*node result\_receiver.js*

**9.4.2 Run controller client**

*cd s\_framework/*hddls\_server\_controller\_receiver*/controller*

*node controller\_cli.js*

*--Please type server ip and port firstly: <ip>:8445*

*--Please type model direcory:*

//Choose the models directory that will be updated into hddls-server

*Note:*

1. Command format:

*help commands that you can use.*

*c <create.json> create pipeslines*

*p <property.json> <pipe\_id> set pipeslines property*

*d <destroy.json> <pipe\_id> destroy pipeslines*

*pipe display pipes belonging to the very client*

*m upload model IR files*

*q exit client.*

2. Json file refer to: *controller*/json\_file