NSCC-Usage(project id: 12001458)

Installing Anaconda, GCC, sox, zlib for my own

```
# login to external network node nscc04-ib0
ssh nscc04-ib0
wget "https://repo.anaconda.com/archive/Anaconda3-2021.05-Linux-x86_64.sh"
# installing anaconda3
./Anaconda3-2021.05-Linux-x86_64.sh
# new env for python3.6
conda create -n kaldi python=3.6
# installing gcc7.5
./contrib/download_prerequisites
./configure --prefix=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0 --
enable-checking=release --enable-languages=c,c++ --disable-multilib
make -j 4
make install
# config bashrc to include PATH of
/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/bin
# install sox
# install zlib
```

Installing Kaldi

Auto path for different node: /home/projects/12001458/haihuaxu/w2021/nscc-scripts/path.sh

fake script: /home/projects/12001458/haihuaxu/w2021/nscc-scripts/fack_2gpu.pbs

sending emails: ssh nscc04-ib0 && echo -e "\n ##LOG\n ERRORS: $errors \n @date `date`" | mail -s "jobs running failed" zhouhui6351233@gmail.com

For ASPIRE nodes(CPU)

```
# set path for CPU node
### path.sh
# module load cmake/3.14.4
export PATH="/home/app/intel/mkl/bin:$PATH"
module load python/2.7.12
export LD_LIBRARY_PATH=/home/app/intel/mkl/lib/intel64:$LD_LIBRARY_PATH
export CPATH=/home/app/intel/mkl/include:$CPATH
# export PATH="/home/projects/12001458/zengzp/ffmpeg/ffmpeg-4.1.4:$PATH"
# export
LD_LIBRARY_PATH=/home/projects/12001458/zengzp/ffmpeg/ffmpeg_build/lib:$LD_LIBRARY_PATH

# export PATH="/home/projects/12001458/zengzp/perl5/perls/bin:$PATH"
```

```
# export
LD_LIBRARY_PATH=/home/projects/12001458/zengzp/perl5/perls/lib:$LD_LIBRARY_PATH

export PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/sox-
14.4.2/bin:/home/projects/12001458/haihuaxu/libsndfile-
1.0.28:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/bin:$PATH
export LD_LIBRARY_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-
gnu/7.5.0:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib64:/home/projects/12001458/yizhoupeng/.local/usr/local/zlib-
1.2.11/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/lib:/lib64/:$LD_LIBRARY_PATH
export CPLUS_INCLUDE_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-
gnu/7.5.0/include:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-gnu/7.5.0/include-fixed
export C_INCLUDE_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-
gnu/7.5.0/include:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-gnu/7.5.0/include-fixed

# download kaldi
git clone https://github.com/kaldi-asr/kaldi.git
mv kaldi kaldi-2021
#login to normal node for faster build
qsub -I -q normal -l select=1:ncpus=24 -l walltime=2:00:00 -P 12001458
mv path.sh /home/projects/12001458/yizhoupeng/w2021/kaldi-2021
cd /home/projects/12001458/yizhoupeng/w2021/kaldi-2021
. path.sh
cd tools
make -j 24
# ALL done
# install srilm
rsync -P /home/hhx502/w2020/kaldi/tools/srilm.tgz
nscc:/home/projects/12001458/yizhoupeng/w2021/kaldi-2021/tools
./extra/install_srilm.sh
cd ../src
./configure --shared --mkl-root=/home/app/intel/mkl
make -j 24 clean depend
make -j 24
## Notice: env build the tools should be exactly the same as src
```

For DGX nodes (GPGPU)

```
# set path for DGX node
### path.sh # use gcc of dgx node Version 7.4.0
export PATH="/home/app/intel/mkl/bin:$PATH"
export LD_LIBRARY_PATH=/home/app/intel/mkl/lib/intel64:$LD_LIBRARY_PATH
```

```
export CPATH=/home/app/intel/mkl/include:$CPATH

export PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/sox-14.4.2/bin:/home/projects/12001458/haihuaxu/libsndfile-1.0.28:$PATH
# export LD_LIBRARY_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/lib/gcc/x86_64-pc-linux-gnu/7.5.0:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/lib64:/home/projects/12001458/yizhoupeng/.local/usr/local/zlib-1.2.11/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/lib:/lib64/:/usr/lib/x86_64-linux-gnu:$LD_LIBRARY_PATH
# export CPLUS_INCLUDE_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/lib/gcc/x86_64-pc-linux-gnu/7.5.0/include:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/lib/gcc/x86_64-pc-linux-gnu/7.5.0/include-fixed
# export C_INCLUDE_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/lib/gcc/x86_64-pc-linux-gnu/7.5.0/include:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/lib/gcc/x86_64-pc-linux-gnu/7.5.0/include-fixed
. path-dgx.sh

#### path-dgx.sh:
CUDAROOT=/home/projects/12001458/haihuaxu/cuda-10.1
# if need other versions of CUDA, you may need to download from NVIDIA and install by your self.
export PATH=$CUDAROOT/bin:$PATH
export LD_LIBRARY_PATH=$CUDAROOT/lib64:$LD_LIBRARY_PATH
export CUDA_HOME=$CUDAROOT
export CUDA_PATH=$CUDAROOT

git clone https://github.com/kaldi-asr/kaldi.git
mv kaldi kaldi-2021-dgx
# login to dgx node
qsub -I -q dgx-dev -l walltime=2:00:00 -P 12001458
cd /home/projects/12001458/yizhoupeng/w2021/kaldi-2021-dgx/tools
make -j 4
# DO NOT install srilm
cd ../src
./configure --shared --mkl-root=/home/app/intel/mkl --use-cuda --cudatk-dir=$CUDAROOT
make -j 24 clean depend
make -j 10
## Notice: env build the tools should be exactly the same as src
```

Installing Espnet

```
#####PATH
export PATH="/home/app/intel/mkl/bin:$PATH"
export LD_LIBRARY_PATH=/home/app/intel/mkl/lib/intel64:$LD_LIBRARY_PATH
export CPATH=/home/app/intel/mkl/include:$CPATH
```

```
# export PATH="/home/projects/12001458/zengzp/ffmpeg/ffmpeg-4.1.4:$PATH"
# export
LD_LIBRARY_PATH=/home/projects/12001458/zengzp/ffmpeg/ffmpeg_build/lib:$LD_LIBRARY_PATH
# export PATH="/home/projects/12001458/zengzp/perl5/perls/bin:$PATH"
# export
LD_LIBRARY_PATH=/home/projects/12001458/zengzp/perl5/perls/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-
gnu/7.5.0:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib64:/home/projects/12001458/yizhoupeng/.local/usr/local/zlib-
1.2.11/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/lib:/lib64/:$LD_LIBRARY_PATH
export CPLUS_INCLUDE_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-
gnu/7.5.0/include:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-gnu/7.5.0/include-fixed:$CPLUS_INCLUDE_PATH
export C_INCLUDE_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-
gnu/7.5.0/include:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-gnu/7.5.0/include-fixed
export PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/sox-
14.4.2/bin:/home/projects/12001458/haihuaxu/libsndfile-
1.0.28:/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/bin:$PATH

conda create -n espnet python=3.7.9
qsub -I -q dgx-dev -l walltime=2:00:00 -P 12001458
cd tools
make TH_VERSION=1.7.1

# Then get back to ASPIRE nodes(nscc04)
# use anaconda base env, but still not working
pip install espnet
pip install mpi4py
pip install chainer==6.0.0

# I think the numpy version should be 1.20.1, 1.20.3 can not fit the Old CentOs.
```

Running exp

```
##PATH path.sh
dgx_sign=0

if grep -q "dgx" <<<"`uname -n`";then
  dgx_sign=1;

else
  dgx_sign=0;
fi || dgx_sign=0
```

```
# load common path...
# export PATH="/home/projects/12001458/yizhoupeng/anaconda3/bin:$PATH"
# export PYTHONHOME=/home/projects/12001458/yizhoupeng/anaconda3

# export PYTHONPATH=/home/projects/12001458/yizhoupeng/anaconda3

#ffmpeg
# export PATH="/home/projects/12001458/zengzp/ffmpeg/ffmpeg-4.1.4:$PATH"
# export
LD_LIBRARY_PATH=/home/projects/12001458/zengzp/ffmpeg/ffmpeg_build/lib:$LD_LIBRARY_PATH

#perl5
export PATH="/home/projects/12001458/zengzp/perl5/perls/bin:$PATH"
export LD_LIBRARY_PATH=/home/projects/12001458/zengzp/perl5/perls/lib:$LD_LIBRARY_PATH

MAIN_ROOT=/home/projects/12001458/yizhoupeng/w2021/espnet-2021

export PATH=$MAIN_ROOT/utils:$MAIN_ROOT/espnet/bin:$PATH
# conda activate espnet

export PATH="/home/projects/12001458/yizhoupeng/anaconda3/envs/espnet/bin:$PATH"

if [ $dgx_sign -le 0 ]; then

#    export LD_LIBRARY_PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/gcc-
7.5.0/lib/gcc/x86_64-pc-linux-
gnu/7.5.0:/home/projects/12001458/$izhoupeng/.local/usr/local/gcc-
7.5.0/lib64:/home/projects/12001458/yizhoupeng/.local/usr/local/zlib-
1.2.11/lib:$LD_LIBRARY_PATH

    export LD_LIBRARY_PATH=/lib:/lib64/:$LD_LIBRARY_PATH


    export PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/sox-
14.4.2/bin:/home/projects/12001458/haihuaxu/libsndfile-
1.0.28:/home/projects$12001458/yizhoupeng/.local/usr/local/gcc-7.5.0/bin:$PATH



        # locate libdl.so.2
    export
PATH="/home/app/intel/xe2017/compilers_and_libraries/linux/mpi/intel64/bin:$PATH"

    export
LD_LIBRARY_PATH=/home/app/intel/xe2017/compilers_and_libraries/linux/mpi/intel64/lib:/u
sr/mpi/gcc/openmpi-3.1.1rc1/lib64:$LD_LIBRARY_PATH
```

```
    export
CPATH=/home/app/intel/xe2017/compilers_and_libraries/linux/mpi/intel64/include:$CPATH



    export KALDI_ROOT=/home/projects/12001458/yizhoupeng/w2021/kaldi-2021
    [ -f $KALDI_ROOT/tools/env.sh ] && . $KALDI_ROOT/tools/env.sh

    export PATH=$PWD/utils/:$KALDI_ROOT/tools/openfst/bin:$PWD:$PATH



    export
PATH=$PWD/utils/:$KALDI_ROOT/tools/openfst/bin:$KALDI_ROOT/tools/kaldi_lm:$KALDI_ROOT/t
ools/srilm/bin/i686-m64:$PATH
    export PATH=$KALDI_ROOT/tools/sph2pipe_v2.5:$KALDI_ROOT/tools/sctk/bin:$PATH



    export
LD_LIBRARY_PATH=$KALDI_ROOT/src/lib:$KALDI_ROOT/tools/openfst/lib:$KALDI_ROOT/tools/spe
ex/lib:$LD_LIBRARY_PATH

    [ ! -f $KALDI_ROOT/tools/config/common_path.sh ] && echo >&2 "The standard file
$KALDI_ROOT/tools/config/common_path.sh is not present -> Exit!"
&& exit 1
    . $KALDI_ROOT/tools/config/common_path.sh
    export LC_ALL=C
else
        # load path for dgx path...
  # sox tool on dgx platform
    export PATH=/home/projects/12001458/yizhoupeng/.local/usr/local/sox-
14.4.2/bin:$PATH
    export KALDI_ROOT=/home/projects/12001458/yizhoupeng/w2021/kaldi-2021-dgx

    [ -f $KALDI_ROOT/tools/env.sh ] && . $KALDI_ROOT/tools/env.sh
    export PATH=$PWD/utils/:$KALDI_ROOT/tools/openfst/bin:$PWD:$PATH



    export PATH=$PWD/utils/:$KALDI_ROOT/tools/openfst/bin:$PATH
    export PATH=$KALDI_ROOT/tools/sctk/bin:$PATH



    export
LD_LIBRARY_PATH=$KALDI_ROOT/src/lib:$KALDI_ROOT/tools/openfst/lib:$KALDI_ROOT/tools/spe
ex/lib:$LD_LIBRARY_PATH

    [ ! -f $KALDI_ROOT/tools/config/common_path.sh ] && echo >&2 "The standard file
$KALDI_ROOT/tools/config/common_path.sh is not present -> Exit!"
```

```
&& exit 1

    . $KALDI_ROOT/tools/config/common_path.sh
    export LC_ALL=C


        #intel mkl

    export PATH="/home/app/intel/mkl/bin:$PATH"
    export LD_LIBRARY_PATH=/home/app/intel/mkl/lib/intel64:$LD_LIBRARY_PATH

    export CPATH=/home/app/intel/mkl/include:$CPATH


    export PATH="/home/projects/12001458/haihuaxu/libsndfile-1.0.28/bin:$PATH"

    export LD_LIBRARY_PATH="/home/projects/12001458/haihuaxu/libsndfile-
1.0.28/lib:$LD_LIBRARY_PATH"

        #mpi  TODO #error using mpi4py

    export
PATH="/home/app/intel/xe2017/compilers_and_libraries/linux/mpi/intel64/bin:$PATH"

    export
LD_LIBRARY_PATH=/home/app/intel/xe2017/compilers_and_libraries/linux/mpi/intel64/lib:$L
D_LIBRARY_PATH
    export
CPATH=/home/app/intel/xe2017/compilers_and_libraries/linux/mpi/intel64/include:$CPATH

# cuda

    CUDAROOT=/home/projects/12001458/haihuaxu/cuda-10.1

    export PATH=$CUDAROOT/bin:$PATH
    export LD_LIBRARY_PATH=$CUDAROOT/lib64:$LD_LIBRARY_PATH

    export CUDA_HOME=$CUDAROOT

    export CUDA_PATH=$CUDAROOT

fi
```