ADL 3.1 Installation Memo

Lide Jiang, CIRA@NOAA/NESDIS/STAR

1. First download the ADL3.1 package:

https://jpss-adl-wiki.ssec.wisc.edu/mediawiki/index.php/Downloads

1) We can selectively download contents in disc1. For running OCC we only need:

adlSoftware.tgz

It's the main ADL software, the backbone.

adlTemplateDataLinks.tgz

It's only 3.0MB, so just download it, and you might need it if you want to develop new algorithms.

adlEdrViirsStaticInputDataLinks.tgz

It contains static inputs such as LUTs for EDR algorithms, including OCC.

adlSdrViirsStaticInputDataLinks.tgz

It contains static inputs such as LUTs for SDR algorithms, including ProAncTemporalInterpolation inputs.

2) For disc2:

adlEdrViirsOutputDataLinks.tgz

It contains truth outputs for EDR algorithms, including OCC.

adlSdrViirsOutputDataLinks.tgz

It contains truth outputs for SDR algorithms, including ProAncTemporalInterpolation truth outputs.

3) We don't need to download disc4, which contains the GridIP tiles. Instead, we download **adlTerEcoDataLinks.tgz** in disc3, which contains all the terrain eco tiles.

4) For disc5, we download **adlEdrViirsDynamicInputDataLinks_Part1.tgz**, which contains dynamic inputs such as SDRs for EDR algorithms, including OCC. Therefore, we don't need to download adlEdrViirsDynamicInputDataLinks_Part2.tgz in disc1.

5) We don't need to download disc6 which contains ADL Lite.

Extract all the .tgz downloaded, or subset the files in each .tgz to only those your need.

You can also download the respective .md5 file corresponding to each .tgz file so that you can check the md5sum of the files in that .tgz once they get extracted:

For example:

>tar xvf adlSoftware.tgz
>md5sum -c adlSoftware.md5 | grep -v OK

2. Set startup environment:

After all the needed files have been extracted, add the following to .bashrc located in the home direcotry:

export ADLINSTALL=/net/orbit260l/disk1/pub/ADL3.1
export ADL_SITE_ID=star
export ADL_DOMAIN=socd
export PATH=${ADLINSTALL}/support:$PATH
source ${ADLINSTALL}/ADL/build/envSetup.ksh

Close and reopen the terminal to activate the ADL build environment. For now just ignore any errors generated by this .bashrc.

3. Download the support scripts and install the COTs:

>mkdir -p ${ADLINSTALL}/support
>cd ${ADLINSTALL}/support
>wget -r -l1 -nH -nd --http-user=$MYUSER -A.sh --http-passwd=$MYPASSWRD
https://jpss.ssec.wisc.edu/adl/download/files/SSEC-Support/ADL3.1/scripts
*>chmod a+rwx **

where $MYUSER and $MYPASSWRD are the username and password that you created when you registered for this website.

We only need run the script that installs user COTs, since all other COTs that require root-access need to be installed by IT staff:

>cd ${ADLINSTALL}/support
>./installUserCots3.1.sh

Make sure to unset the environment variable $LD_RUN_PATH before installing the user COTs since it can interfere with shared library path:

>unset LD_RUN_PATH

Important Note: I had problem with ADL standalone programs generation later on, which I found was due to the interface incompatibility between the gcc 4.4 version ADL used for compilation and the gcc 4.1.2 (default gcc in our system) the lapack COT used. The issue was resolved after re-compiling lapack libraries using gcc4.4 (edit the make.inc under lapack dir).

Similar problem may arise for other COT library, but since OCC algorithm does not need those libraries, it can be left alone by only generating the OCC controller executable by going to the OCC controller directory and typing 'make program'.

4. Download and apply patches for ADL3.1:

>wget --http-user=$MYUSER -A.sh --http-passwd=$MYPASSWRD https://jpss.ssec.wisc.edu/adl/download/files//SSEC-Support/ADL3.1/config/ADL3.1.patch
>wget --http-user=$MYUSER -A.sh --http-passwd=$MYPASSWRD https://jpss.ssec.wisc.edu/adl/download/files//SSEC-Support/ADL3.1/config/envSetup.ksh

Copy the downloaded envSetup.ksh to $ADLINSTALL/ADL/build/ to overwrite the original file, and the downloaded ADL3.1.patch to $ADLINSTALL/ADL/. Go to $ADLINSTALL/ADL/ and apply the patch by:

>patch -p1 < ADL3.1.patch

Close and reopen your terminal to active the ADL build environment.

5. Building ADL

One important step not mentioned in the ADL installation guide must be followed to make sure the needed libraries can be found:

>export LD_LIBRARY_PATH= $LD_LIBRARY_PATH:$ADLINSTALL/local/lib:$ADLINSTALL/ADL/lib:$ADLINSTALL/COTS/xerces-c_2_8_0-x86_64-linux-gcc_3_4/lib

To build the ADL software just go to ADL/ and run the building script:

>buildAdl.ksh > buildAdl.log 2>&1 &

Also Make sure to unset the environment variabe $LD_RUN_PATH before building ADL.

The process encountered some errors after it built all sources and libraries and started to link them to generate executables. It had already produced all the utility programs we need, such as HDF5 packer and unpacker, binary editor, byte order converter, ancillary file temporal interpolator, etc. The only thing we needed which was not there yet was the OCC controller program. Just go to $ADLINSTALL/ADL/EDR/VIIRS/ocean/Color/Controller/ and type ‘make program’, and we are done.