TAHOE Instllation Guide

Bibekananda Datta

April 2022

Installing on macOS:

TAHOE was installed and tested on macOS Catalina 10.15 with open-mpi 4.1.3 and gcc 11.2.0. It is recommended to install the parallel version. Sourceforge doesn't have current version TAHOE available yet and will be updated soon.

1. If you don't have X11 client on macOS, download and install XQuartz. Visit https://www.xquartz.org.

2. In case you don't have a zsh profile, open one using:

```
$ open ~/.zshrc
```

3. Save and close the .zshrc file. We will use this file later to save the environment variables and path related to TAHOE installation.

4. Download and install homebrew package manager using:

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/
   Homebrew/install/HEAD/install.sh)"
```

5. Change the read and write permission for homebrew using:

```
$ sudo chown -R $(whoami) $(brew --prefix)/*
```

6. Download and install X11 client for macOS, XQuartz:

```
$ brew install --cask xquartz
```

7. Install the following packages using homebrew:

```
$ brew install wget
$ brew install cmake
$ brew install automake
$ brew install open-mpi
```

8. Current stable version of open-mpi available via homebrew is 4.1.3 and depends on gcc 11.2.0. Please check the installed versions of gcc, g++, and gfortran from the following directory before proceeding to next step.

```
$ cd \usr\local\bin
```

9. Create symlink for `gcc`, `g++`, and `gfortran` compilers using (Please make sure to use the correct version of the compilers):

```
$ ln -s gcc-11 gcc
$ ln -s g++-11 g++
$ ln -s gfortran-11 gfortran
```

10. Log out of `zsh` terminal to activate the changes and log back in.

11. In your home directory, create a directory for downloading and installing tahoe and its dependencies:

```
$ mkdir tahoe && cd tahoe
```

12. Clone the ACCESS library using git. We will use `exodus` format. Install the Third-Party Libraries (TPL), build and install SEACAS using following:

```
$ git clone https://github.com/gsjaardema/seacas.git
$ cd seacas
$ ./install-tpl.sh
$ mkdir build && cd build
$ ../cmake-exodus
$ make && make install
```

13. Save the following environment variables to your `.zshrc` file. Please make sure to use the correct directory.

```
$ TAHOE_MAIN=$PWD
$ echo export TAHOE_MAIN=$TAHOE_MAIN >> ~/.zshrc
$ echo export ACCESS=$TAHOE_MAIN/seacas >> ~/.zshrc
$ echo export LD_LIBRARY_PATH=$ACCESS/lib >> ~/.zshrc
$ echo export CVS_RSH=ssh >> ~/.zshrc
$ source ~/.zshrc
```

14. Create symlinks for `exodus` libraries.

```
$ cd $ACCESS
$ ln -s include/ inc
$ cd lib
$ ln -s libexodus.dylib libexoIIv2c.dylib
$ ln -s libexodus.a libexoIIv2c.a
```

15. Log out of `zsh` terminal to activate the changes and log back in.

16. Download the latest version of `TAHOE-Installer` (`tahoe-manager`, installation macros and scripts) from Nguyen Lab OneDrive to `/Users/bibek/Downloads/TAHOE-Installer`.

17. Create a back-up directory for downloading all the default and optional modules for `TAHOE`.

```
$ cd TAHOE_MAIN
$ mkdir tahoe-backup && cd tahoe-backup
```

18. Set following variables for ease of usage:

```
$ TAHOE_DOWNLOAD=/Users/bibek/Downloads/TAHOE-Installer
$ TAHOE_BACKUP=$TAHOE_MAIN/tahoe-backup
```

19. Copy `tahoe-manager` to the `tahoe-backup` directory.

```
$ cp -i $TAHOE_DOWNLOAD/tahoe-manager $TAHOE_BACKUP
```

20. If the `tahoe-manager` isn't already an executable, convert it to an an executable file. If `tahoe-manager` is already an executable, you can skip this step.

```
$ sudo chmod 755 ./tahoe-manager
```

21. Run `tahoe-manager` to download the macros, default modules, and optional modulues:

```
$ ./tahoe-manager
```

22. Select connection type: `svn`, and enter sourceforge user name. It will start downloading the macros. Once the macros are downloaded, enter any architecture type to continue download.

23. For optional modules, select 0 (`CBLAS`), 6 (`ACCESS`), 8 (`benchmark_XML`), 10 (`contrib`), 11 (`development`), 12 (`development_benchmark_xml`), 13 (`f2c`), 14 (`metis`), 15 (`spooles`), 16 (`spoolesMPI`).

24. Copy the new macro files, `GNU-GCC-MPI-9.3.macros` and `GNU-GCC-9.3.macros`, to the downloaded `marcos` directory. This is a temporary step until the newer macros are available on sourceforge.

```
$ cp -i $TAHOE_DOWNLOAD/*.macros $TAHOE_BACKUP/macros
```

25. Clean all the settings before making a copy of the back-up directory for installation purpose.

```
$ ./tahoe-manager clean
```

26. Make a copy of the back-up directory for installation and rename it.

```
$ cd $TAHOE_MAIN
$ cp -r tahoe-backup tahoe-install
```

27. Go to the installation directory and run `tahoe-manager` to update the installation settings with correct architecture.

```
$ cd tahoe-install
$ ./tahoe-manager update
```

28. Select a connection type, enter an user name, select an architecture. For parallel version of `TAHOE`, select `GNU-GCC-MPI-9.3` or for serial version of `TAHOE`, select `GNU-GCC-9.3`. Select the optional modules as before. If serial version of `TAHOE` is being installed, skip optional module 16 (`spoolesMPI`).

29. Build and compile `TAHOE:`

```
$ ./tahoe-manager build
```

30. Once `TAHOE` is compiled and built, add following environment variables and path to your `.zshrc` file:

```
$ echo export TAHOE_MOD=$TAHOE_MAIN/tahoe-install >> ~/.zshrc
$ echo export TAHOE_DIR=$TAHOE_MOD/tahoe >> ~/.zshrc
$ echo export PATH=$PATH:$TAHOE_MOD/bin >> ~/.zshrc
$ source ~/.zshrc
```

Installing on Ubuntu:

Parallel version of `TAHOE` was installed and tested on Ubuntu 20.04 via Windows Subsystem for Linux (WSL). Download the latest version of `tahoe-manager`, installation macros and scripts from Nguyen Lab OneDrive. Sourceforge doesn't have this version and will be updated soon.

1. Download the latest version of `TAHOE-Installer` (`tahoe-manager`, installation macros and scripts) from Nguyen Lab OneDrive to `/mnt/c/Users/bdatta1/Downloads`.

2. Go to your home directory, create a directory for downloading and installing tahoe modules and its dependencies:

```
$ cd
$ mkdir tahoe && cd tahoe
$ mkdir tahoe-backup && cd tahoe-backup
```

3. Set following variables for ease of usage. Make sure to use the correct directory.

```
$ TAHOE_DOWNLOAD=/mnt/c/Users/bdatta1/Downloads/TAHOE-Installer
$ TAHOE_MAIN=/home/bibek/TAHOE
$ TAHOE_BACKUP=/home/bibek/TAHOE/tahoe-backup
```

4. Copy `install_pre.sh` and `tahoe-manager` to the `tahoe-backup` and `tahoe` directory:

```
$ cp -i $TAHOE_DOWNLOAD/install_pre.sh $TAHOE_MAIN
$ cp -i $TAHOE_DOWNLOAD/tahoe-manager $TAHOE_BACKUP
```

5. Open `install_pre.sh` using a text editor (for example, VS Code) and change the installation directory variable, `$TAHOE_MAIN`, within the file. Save and close the file.

```
$ code install_pre.sh
```

6. Change the permission for `install_pre.sh` to make it an executable. Run the script to download and install pre-requisite packages for `TAHOE`.

```
$ chmod a+x tahoe_pre.sh
$ ./install_pre.sh
```

7. Log out of the bash shell to activate the change and log back in.

8. Set following variables for ease of usage:

```
$ TAHOE_DOWNLOAD=/mnt/c/Users/bdatta1/Downloads/TAHOE-Installer
$ TAHOE_BACKUP=/home/bibek/TAHOE/tahoe-backup
```

9. Go to `tahoe-backup` directory. Run `tahoe-manager` to download the macros, default modules, and optional modulues:

```
$ cd $TAHOE_BACKUP
$ ./tahoe-manager
```

10. Select connection type: `svn`, and enter sourceforge user name. It will start downloading the macros. Once the macros are downloaded, enter any architecture type to continue download.

11. For optional modules, select 0 (`CBLAS`), 6 (`ACCESS`), 8 (`benchmark_XML`), 10 (`contrib`), 11 (`development`), 12 (`development_benchmark_xml`), 13 (`f2c`), 14 (`metis`), 15 (`spooles`), 16 (`spoolesMPI`).

12. Copy the new macro files, `GNU-GCC-MPI-9.3.macros` and `GNU-GCC-9.3.macros`, to the downloaded `marcos` directory. This is a temporary step until the newer macros are available on sourceforge.

```
$ cp -i $TAHOE_DOWNLOAD/*.macros $TAHOE_BACKUP/macros
```

13. Clean all the settings before making a copy of the back-up directory for installation purpose.

```
$ ./tahoe-manager clean
```

14. Make a copy of the back-up directory for installation and rename it.

```
$ cd $TAHOE_MAIN
$ cp -r tahoe-backup tahoe-install
```

15. Go to the installation directory and run `tahoe-manager` to update the installation settings with correct architecture.

```
$ cd tahoe-install
$ ./tahoe-manager update
```

16. Select a connection type, enter an user name, select an architecture. For parallel version of `TAHOE`, select `GNU-GCC-MPI-9.3` or for serial version of `TAHOE`, select `GNU-GCC-9.3`. Select the optional modules as before. If serial version of `TAHOE` is being installed, skip optional module 16 (`spoolesMPI`).

17. Build and compile `TAHOE:`

```
$ ./tahoe-manager build
```

18. Once `TAHOE` is compiled and built, add following environment variables and path to your `.bashrc` file:

```
$ echo export TAHOE_MOD=$TAHOE_MAIN/Tahoe-Install >> ~/.bashrc
$ echo export TAHOE_DIR=$TAHOE_MOD/tahoe >> ~/.bashrc
$ echo export PATH="$TAHOE_MOD/bin:$PATH" >> ~/.bashrc
$ source ~/.bashrc
```