

Installation of WARP on the Willson cluster (TEV)

Vince Moens
vince.moens@epfl.ch

Ecole Polytechnique Fédérale de Lausanne (EPFL)

June 13, 2013

1 Introduction

For the purpose of this document, TEV refers to the Wilson Cluster.

The tarball that accompanies these instructions can be found at: <https://cdcvns.fnal.gov/redmine/documents/619>.

Installing WARP on TEV brings a few complications. These include compatibility issues with the python installed on TEV, compiler issues and super user rights. Instead of finding a solution in which one could bind the python package WARP, which has to be installed locally, into the native python of TEV by creating links and changing a lot of PATHS, we decided to just install a clean version of python in the local directory and use that distribution.

These instructions are for the installation of a non-parallel version of WARP. A parallel version of WARP will be installed later and a new set of instructions will be drafted.

Unless someone sent you this document, you should have found these instructions in a directory named `INSTRUCTIONS` in a tarball named `WARP_Install`.

2 Installation procedure

Place that tar file in your home directory on TEV. You can use `scp` to place it there.

```
scp -r WARP_Install.tar.gz username@tev.fnal.gov:/home/username/
```

Open up a terminal and go to your home directory. Once there expand your tarball using:

```
cd  
tar -xvf WARP_Install.tar.gz
```

You will now find a directory labeled `WARP_Install` in your home directory. Create a directory where you would want to install Python and WARP. For the purpose of this tutorial we will use the directory `Python` in your home directory.

```
cd  
mkdir Python
```

Next we want to make sure that every compilation of a python script from this point is done using the same compilers and your new python installation. There is two methods to do this. The first is the recommended one and automatically loads the new path upon opening a window. In order to do this check for a file named `.bash_profile` in your home directory.

```
cd  
ls -lisa
```

If it does not exist, create it by typing:

```
cd
nano .bash_profile
```

Then in the new terminal window insert the following data:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=/usr/local/gcc-4.6.2/bin:/home/USERNAME/Python/bin:$PATH
export PATH

export LD_LIBRARY_PATH=/usr/local/gcc-4.6.2/lib64:/usr/local/gcc-4.6.2/lib
```

Be sure to change the username on line 10 to your own username. Close nano by typing Ctrl-O, Enter and Ctrl-X.

If it does exist, append the file by opening it in a similar fashion as above using nano and append the file with:

```
# User specific environment and startup programs

PATH=/usr/local/gcc-4.6.2/bin:/home/USERNAME/Python/bin:$PATH
export PATH

export LD_LIBRARY_PATH=/usr/local/gcc-4.6.2/lib64:/usr/local/gcc-4.6.2/lib
```

Again, be sure to change the username on line 3 to your own username and close nano by typing Ctrl-O, Enter and Ctrl-X.

2.1 Installing Python

We will now install Python 2.7.3 locally into your home directory. Go into Python-2.7.3 in the directory WARP_Install.

```
cd WARP_Install/Python-2.7.3/
```

In order to install python, run the configure file, while specifying a prefix. In my case the prefix is /home/vmoens/Python.

```
./configure --prefix=/place/where/python/is/to/be/installed
make install
```

You can check whether the directories bin, include, lib and share can be found in the directory that you used as prefix above. Furthermore you should run

```
cd
which python
```

which checks which version of python you use. The output should refer to your install directory, for me it is /Python/bin/python.

2.2 Installing Numpy

Next you will install the numerical python package. In order to install it cd into the directory numpy in WARP_Install and run the setup.py script.

```
cd ~/WARP_Install/numpy/
python setup.py install --prefix=/place/where/python/is/to/be/installed
```

The path given to prefix should be the same as that for your python installation. Check your numpy installation by loading python and then numpy. Subsequently you should pass the command `numpy.__path__` to python. The path should refer to the python installation in your home directory.

```
cd
python
import numpy
numpy.__path__
```

(do not copy all 4 lines at once into your terminal. it will cause errors.)

I obtained the output: `['/home/vmoens/Python/lib/python2.7/site-packages/numpy']`. You can quit python by typing `quit()`.

2.3 Installing SciPy

Next you will install the scientific python package. In order to install it cd into the directory `scipy-0.12.0` in `WARP_Install` and run the `setup.py` script by first running `build` (no prefix) and then `install`.

```
cd ~/WARP_Install/scipy/
python setup.py build
python setup.py install --prefix=/place/where/python/is/to/be/installed
```

The path given to prefix should be the same as that for your python installation. Check your scipy installation by loading python and then scipy. Subsequently you should pass the command `scipy.__path__` to python. The path should refer to the python installation in your home directory.

```
cd
python
import scipy
scipy.__path__
```

(do not copy all 4 lines at once into your terminal. it will cause errors.)

I obtained the output: `['/home/vmoens/Python/lib/python2.7/site-packages/scipy']`. You can quit python by typing `quit()`.

2.4 Installing iPython

iPython is a more freindly interface for python. It allows syntax coloring and simplifies certain commands such as `quit()` becomes `quit`. It furthermore allows you to call directories and change paths from inside python in a simple fashion. This package is not required but recommended. Install it using the `setup.py` file:

```
$ tar -xzf ipython.tar.gz
$ cd ipython
$ python setup.py install
```

From now on you may start python by typing `ipython` instead of `python`.

2.5 Installing Forthon

Having installed python and numpy in your home directory, you now have a fully functioning python distribution for science. We still need warp. Therefore we now install Forthon. This is a bidning between Fortean and Python.

Start of by going into the Forthon directory and executing the `setup.py` file.

```
cd ~/WARP_Install/Forthon-0.8.11
python setup.py install --prefix=/place/where/python/is/to/be/installed
```

Upon termination of the script you can check the installation by checking the path of the Forthon installation:

```
cd
python
import Forthon
Forthon.__path__
```

The path should again point to your install directory.

For me it points to: `['/home/vmoens/Python/lib/python2.7/site-packages/Forthon']`. Terminate Python with `quit()`. Additionally you can type `which Forthon` to check for the path of the Forthon package that is used.

Given a correct directory for Forthon, you have successfully installed Forthon and can now start installing WARP itself.

2.6 Installing WARP

This is probably the trickiest part. First you must decide on whether you wish to install a single, a parallel or both versions of WARP. The difference is apparent in how you handle the compilation of `Makefile.Forthon` or `Makefile.Forthon.pympi`. I will first explain the single version and then the parallel version. You may compile both in the same directory. The installations will not interfere with each other. The difference is in how you call warp once it is installed.

2.6.1 Single Installation Configuration

Call the warp directory and go into `pywarp90`. Here you will have to make a file called `WarpC.so` which will be placed in the `scripts` directory.

```
cd ~/WARP_Install/warp/pywarp90/
make -f Makefile.Forthon
```

Upon successful compilation, you should find a file called `WarpC.so` in the directory `scripts` which is located in the parent directory. Additionally a folder labeled `build` will appear in your `pywarp` folder.

2.6.2 Parallel Installation Configuration

In order to install a parallel version of WARP, you will have to first find the install directories of openmpi. On tev they can be found at `/usr/local/openmpi/`. You will then have to `cd` into the `pywarp` directory:

```
cd ~/WARP_Install/warp/pywarp90/
```

In the `WARP_Install` directory, which accompanies this script, all necessary modifications have already been made and you may thus skip to the making of `Makefile.Forthon.pympi`.

In case you use an updated version of WARP which you can download from `warp.lbl.gov`, you will have to make these changes again. First replace line 2 of `Makefile.Forthon.pympi`, which defines `FARGS`, with:

```
FARGS = --farg "-DMPIPARALLEL -I/usr/local/openmpi/include -L/usr/local/openmpi/lib/"
```

Now you should create a new file in `pywarp90` called `setup.local.py`, which reads:

```
if parallel:
    library_dirs = library_dirs + ['/usr/local/openmpi/lib/']
    libraries = fcompiler.libs + ['mpi', 'mpi_f77']
```

Once you have modified the first file and created the setup file, or have decided to use the files present in the WARP install directory, you may make `Makefile.Forthon.pympi` using the command:

```
make -f Makefile.Forthon.pympi
```

Upon successful compilation, you should find a file called `WarpCparallel.so` in the directory `scripts` which is located in the parent directory. Additionally a folder labeled `buildparallel` will appear in your `pywarp` folder.

For the parallel installation you will also require `pyMPI`. You may install the newest version by obtaining it from `git`:

```
cd ~/WARP_Install/
git clone http://portal.nersc.gov/project/warp/git/pyMPI.git
or alternatively you may use the version included in the WARP_Install directory. You may then install it via:
./configure --prefix=/place/to/install
make install
```

Now call the directory `scripts` and execute the `setup.py` file.

```
cd ../scripts
python setup.py install --prefix=/place/where/python/is/to/be/installed
```

again you should use the same prefix directory as for your other installations. You may check the successful installation of WARP by launching Python, loading WARP and checking its path.

```
cd
python
import warp
warp.__path__
```

I obtain the following result `['/home/vmoens/Python/lib/python2.7/site-packages/warp']`. Terminate python using `quit()`.

2.7 Installing PyGist

PyGist is the graphical interface of WARP. It is the most tricky to install since it has hardcoded directories in the install code that require super user privileges and need to be changed.

Using an editor of your liking you will need to adapt the code. For the purpose of this example I will use VIM, since it is a terminal integrated editor that supports syntax highlighting.

Start of by calling the `pygist` directory and opening the `setup.py` file in your editor.

```
cd ~/WARP_Install/pygist
vim setup.py
:set number
:syntax on
```

Now go to line 445 and replace `'/home/vmoens/Python/bin'` with the `bin` directory in the directory you have used above for prefix. Do the same on line 452. You may edit in vim by typing `i` and leave the editing mode by pressing `Esc`. You may go to a line by typing `:` and then the number. For example `:445`.

After having changed the lines, leave the editor (type `:wq`) and configure the install script. After this you may run the install script as usual.

```
cd ~/WARP_Install/pygist
python setup.py config
python setup.py install --prefix=/place/where/python/is/to/be/installed
```

You may again check the installation by calling any directory except the install directories, running Python and importing `gist`. Now check the path in which `gist` is installed.

```
cd
python
import gist
gist.__path__
```

I obtain `['/home/vmoens/Python/lib/python2.7/site-packages/gist']`. You may again quit python by typing `quit()`.

You have now successfully installed WARP on TEV. If you wish you may now delete that `WARP_Install` directory and its tarball.

3 Usage of WARP on TEV

4 Reinstall & Update

Updates and reinstalls are easily possible. In order to reinstall a part of the software from a install directory that you have already used, you will have to ensure that all folders labeled `build` and `buildparallel` have been removed from your installation directory. Once this is completed, you may proceed as you did during the installation above. Updates versions of all the packages are either available from `warp.lbl.gov` or from the respective python repositories. As a rule of thumb, you may always just reinstall what is in a subsection in the instructions above. The different subsections in the installation instructions should be independent of eachother. Only for the installation of the `warp` directory, will you have to go through the compilation of the makefiles and the installation of the scripts directory.

5 Acknowledgements

At the points I would like to thank the following people in helping me troubleshoot the issues with installing WARP.

5.0.0.1 David P. Grote David is one of the developers of WARP over at Lawrence Berkley National Laboratory. He was incredibly helpfull in debugging all the error codes from TEV and even created an account on TEV for himself in order to help with the installation of WARP.

5.0.0.2 Eric G. Stern Eric works at the Scientific Computing Division at Fermilab. He helped out with local issues and the debugging of some of the error messages on TEV.

5.0.0.3 Moses Chung Moses works at the APC Experimental Beam Physics Department and has used WARP for his PhD thesis. He was helpfull in gathering information from friends, former students and his own work in order to help with the installation.

5.0.0.4 tev-admin@fnal.gov The admin guys of tev were incredibly useful in answering questions concerning the architecture of tev and the packages that are installed.

5.0.0.5 Alexander Valishev Alexander organized the accounts for David, which ultimately lead to david being able to solve the issue.

5.0.0.6 Giulio Stancari Giulio was very helpfull by providing his old installation files from a previous version that provided some comparisons and giving helpful tips.

For further questions concernign the installation of WARP, please don't hesitate to ask me by sending me an email to `vmoens@fnal.gov`.