TVB#2.1: Packaging TVB for the modern world

INCF GSOC 2018 - project proposal

Student contact details

Full Name: Umar Haruna Abdullahi

Email: umarbrowser20@gmail.com

Location (Country and city): Nigeria, Gombe.

Blogs: https://medium.com/@pycoder_boy , http://www.umarbrowser.co.vu

Hangouts: umarbrowser20@gmail.com

Github: umarbrowser

Time Zone: West Africa Time Zone UTC+01:00

Phone Number: +2347066399174, +2347050906454

IRC: pyc0d3r

Project details

Project synopsis / summary

Packaging and distribution of the the virtual brain (TVB) to the scientific community using the most used scientific software distributors and virtual machines such as packaging to conda, Packaging to neurodebian a linux system packed with neuroscience software, Develop a TVB native launcher for some operating systems, Develop a vagrantfile script that will install TVB on virtual machine, Develop a script for running TVB on docker, And finally if time allow, Create an Amazon Web Image (AMI) to run TVB on the cloud.

What is the project about? Why is it important?

The Virtual Brain (TVB) is a framework for the simulation of the dynamics of large scale brain networks with biologically realistic connectivity, Because of it absence in the most used scientific distribution for windows, Mac and linux, It is too hard for many people in the scientific community to install and use it.

The virtual brain only distribute its packages in form of zip package or source using git and recently using python pypi, If TVB Distributes it’s package using the most used distribution method on scientific community like conda, Neurodebian etc, At the end this project I will deliver all the project deliverables.

Project in details

How will you handle the project? Detailed description of your planned approach.

I have plenty of time that will allow me to work up to 50 Hours per week, I encounter some issues while trying to migrate from Fedora to Debian and right now i fix them.

Meanwhile, i will keep in touch to my mentors on every piece of code I write. However, if there is something wrong or not in the right place they can guide me and I will make the necessary changes as quick as possible. Every week I will send a report on my blog and I intend to complete the project on time scheduled (12 weeks) including the bugs fixing, documentation and final report for submission.

Project deliverables

Minimal set of deliverables

1. Packaging TVB to Anaconda.
2. Develop a Native launcher for TVB Distributions
3. Packaging and Distribution for NeuroDebian .
4. Develop a Vagrantfile script for virtual machines.
5. Develop a Script(Dockerfile) for building a docker image

6. Code Cleanup, Final Documentation.

Additional ‘if time allows’ deliverables

1. Develop a script for building a Amazon Web Image.
2. Continue to contribute/maintain all the mentions above project and some The Virtual Brain Projects using code and Documentations(Plan After GSoC 2018)

Detailed timeline

Community Bonding	April 23 - May 14
Getting completely comfortable with existing codebase, research, code documentation	
Packaging to Anaconda (#1)	May 14 - May 28
Develop a Native Launcher for TVB Distributions (#2)	May 28 - June 11
GSoC coding phase 1 evaluation	June 11 - June 15
NeuroDebian Packaging (#3)	June 11 - July 2
Develop a Script for Vagrantfile (#4)	July 2 - July 9
Develop a Script for Docker Image (#5)	July 9 - July 16
GSoC Coding phase 2 evaluation	July 9 - July 13
Develop a Script for Amazon web image (#6)	July 16 - July 23
Code Cleanup, Final Documentation (#7)	July 23
Project submission	August 6

Community Bounding and Reviewing Code base of TVB

- Remain in constant touch with my mentor(s) and community.

- Explore and read wiki articles of The Virtual Brain. This would, for example, help me in better understanding diffs, permalinks, ordinary article URLs, etc.

1. (Two weeks) Conda packaging Already started on Jira[1]

I will start packaging to conda, I have already install all the necessary files and tools for building and packaging to conda, Some of the packages like tvb-gdist, tvb-library, tvb-data are already documented on jira[1] and conda-forge[2], Right now i try and build some packages using:

- bld.bat packaging details for windows kind operating system
- build.sh packaging details for Linux and OSX operating systems
- meta.yaml build instruction contain some information of the package like library, licences details that will be included

At the end of this project i will test by install using:

$ conda config --add channels
$ conda-forge conda install tvb-framework

Right now there was some 3rd party libraries for tvb-framework with has some problems those are:

★ formencode — we use version 1.3, and only an older version is there
★ cfflib — can only be installed on linux. And Mac it is not compatible

A direction here would be to package ourselves these two(2) libraries, but we need to answer first the main question:

- Are we allowed to do it, considering that we are not the authors? I already email and submit an issue[3] to cfflib author and he gives me permission to package it to Anaconda.
- Try to replace these libraries in TVB code. e.g. formencode for sure we can make optional (just like the TVB developers do with allenSDK)

2. (Two weeks) Developing a native launcher for TVB Distributions

TVB is only being launch using .bat on windows and so on, why not develop a standard and default launching format e.g .EXE on Windows with the help of 3rd party tool, I will use Cx_freeze with is a set of scripts and modules for freezing python scripts into executables, I decide not to use Pyinstaller because there was trouble getting it to include certain egg dependencies, It couldn't handle conditional imports as well as needed and looking through the bundled archive was difficult, On windows it requires pywin32 (so it can't be used with virtualenv), I obtain a BitRock InstallBuilder[9] open source license, I will use it to build a cool Standalone (One File not in zip) setup installer for TVB, BitRock InstallBuilder is a development tool for building cross-platform installers. At the end of the project i will install and test.

3. (Three weeks) Neurodebian packaging - already documented on jira[4]

Neuro Debian packaging which is already started and documented on jira[4] currently now I sign up as a debian maintainer with his role is uploading a packages to mentors.debian[5] and search for Sponsor.

I will start building the .deb packages and other debian supported packages like .tar.xz .tar.gz, I try to build some[6] Just for Testing,

NeuroDebian Packaging consists of:

- Neuro Debian Software (tvb-framework) packaging
- Neuro Debian Datasets (tvb-data, etc) packaging.

Neuro Debian Software (tvb-framework) packaging is a process of binding the relevant files and components to build a customized application for a pkgs, with will be install using ***$ sudo apt-get install ###pkgname###*** or ***$ gdebi ###pkgname###*** and i will build the package for the support of I386, amd64.

Neuro Debian Datasets(tvb-data, include its libraries) packaging the goal of these packages is to provide some interesting and relatively large, datasets to demonstrate various data analysis challenges in python, I will package it and will be installed alongside with tvb-framework, Currently now i contact the neurodebian developers and they told me the resolved some licensing issues with the virtual brain a few years back this will be very helpful, At the end of this project i will test by adding the neurodebian repository.

```
$ wget -O- http://neuro.debian.net/lists/buster.au.full | sudo tee
/etc/apt/sources.list.d/neurodebian.sources.list
$ su apt-key adv --recv-keys --keyserver hkp://pool.sks-keyservers.net:80 0xA5D32F012649A5A9

$ su apt-get update && apt-get install tvb-framework
```

4. (One week) Developing a script(Vagrantfile) for Virtual machine.

The primary function of the Vagrantfile is to describe the type of machine required for a project, and how to configure and provision these machines, I developed a simple vagrant script with will download a puppetlabs/ubuntu-14.04-32-nocm [7] with can be run using: ***$ vagrant up*** command and you can view and edit it using virtualbox i encountered some issues while trying to run vagrant base image Thanks to mentors for for guiding me on Reviewing my Proposal, they told me TVB don't need GUI to run that is the (1) below is only require:

1. with is generate using this command: e.g ***config.vm.box = "ubuntu/trusty64"*** (because its only support Terminal or ssh)
2. Using the personal generated Vagrantfile of the user, this will be generated when user has a running virtual machine on virtualbox only needs to run this command: e.g ***vagrant package --base=<name of the virtual image running on virtualbox> --output=<outputname>.box*** (it's only remain small fix and write the script and upload it to vagrant cloud or any hosting address)
3. using the Graphical user interface(GUI) this one with Ubuntu 64 and can be run using e.g ***config.vm.box = "chad-thompson/ubuntu-trusty64-gui"*** just continue to write and download our tvb-tools ether using python pip(working), Anaconda(to continue), or Neurodebian(to start) by adding the neurodebian repository to Ubuntu ***/etc/apt/sources.list.***

At the end of this project i will deliver a full and completed vagrantfile script with will install a Debian and the virtual brain inside, you can run it by visiting the expose port address on your host browser, After i finished, i will add the script repository to Travis CI.

5. (One weeks) Developing a script(Dockerfile) for building a Docker image.

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers, I developed a simple Dockerfile[8] you can access it by opening ***http://localhost:8888*** in your browser, or ***http://<DOCKER-MACHINE-IP>:8888*** if you are using a Docker on Virtual Machine, I would separate the task in two stages:

1) Build a simple docker image with TVB inside (and all its dependencies): I will create a docker image that have the complete TVB installed.
2) Use This Image from python(Anaconda): use ***continuumio/miniconda*** image, because its base on python and is installed into the /opt/conda folder and ensures that the default user has the conda command in their path.

After i finished then i will upload the script to github and i will add the script repository to Travis CI, Then i will upload the developed image to[18] then user can use those command to download the complete virtual brain docker image and use it:

$ docker pull tvb/the-virtual-brain-image

$ docker run -i -t tvb/the-virtual-brain-image /bin/bash

6. (One week) Develop a script for building a Amazon Web Image

An amazon machine image (AMI) [10] provides the information required to launch an instance, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need.

If time allow i will develop a (Amazon EBS-backed Linux AMI) with tvb inside (and all its dependencies) i will test it by running using ssh.

7. (Two week) Documentation

Code Cleanup, Final documentation on tvb-documentation and Report

- Wrap-up phase

- Edit documentations on wiki and on code

- Final Testings

- Community Reviews

- **Submission to Google**

Plan for communication with mentors

I plan to contact my mentors on Hangouts call, chat and mail.

Candidate extra details

Motivation

I find the entire TVB project and also this particular project extremely interesting and It will bring me to participate more on open source community, I consider this is a good chance for me to improve my current skills and develop some new ones and it will help me to familiarize with the open source community. Also, after completing this project I want to continue the work in this organisation.

Match - tell us about something you've worked on in the past that would make you a good candidate for this project.

1. Test and Packaged one of TVB package to Anaconda[12] just for learning
2. Github pull request of Brain Imaging Pipelines under INCF for fixing small bug[13]
3. Pull request of fixing Missing documentation tvb-recon[14]
4. Developed a simple Neurodebian package .deb [6]
5. Developed a simple vagrantfile [7]
6. Developed a simple Dockerfile[8]
7. My Blender Developer profile [15]
8. My Fedora Wiki page[16]

Is this the only project that you will apply for?

Yes this is the only project i will apply for.

Working time and commitments - will you be working full time on this?

Yes i will be working full time(up to 50 hours per week).

Do you have any other plans for the summer

No i don’t have any other plans on this summer.

Past experience

I have a passion to open source, I have experience on some programming languages and graphic design, must of them using open source software like blender, gimp, inkscape for about 5-6 years and some programming languages like bash, C/C++ etc and i am running linux that why i was creating some simple Bash Script to automate my simples task I was running virtual machine to run different OS on my single machine.

I am about to finish government science and technical college gombe, and now i am pursuing to study computer science at Gombe state university.

Curriculum Vitae

Here it is[17]

Links

1. https://req.thevirtualbrain.org/browse/TVB-2301
2. https://github.com/conda-forge/staged-recipes/pull/4211
3. https://github.com/LTS5/cfflib/issues/6
4. https://req.thevirtualbrain.org/browse/TVB-1074
5. https://mentors.debian.net/
6. https://github.com/umarbrowser/Neuropackage
7. https://github.com/umarbrowser/vagrantTvb
8. https://github.com/umarbrowser/dockerImageTvb
9. https://installbuilder.bitrock.com/index.html
10. https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/AMIs.html
11. http://docs.thevirtualbrain.org/
12. https://anaconda.org/pyc0d3r/tvb-data
13. https://github.com/INCF/BrainImagingPipelines/issues/77
14. https://github.com/the-virtual-brain/tvb-recon/pull/44
15. https://developer.blender.org/p/pyc0d3r/
16. https://fedoraproject.org/wiki/User:Pyc0d3r
17. https://docs.google.com/document/d/1Duuc8DBGRpaf1bD96YWTEbCO2yIPlb7TGGnOJpCZPJw/edit?usp=sharing
18. https://hub.docker.com