

# **Study and Development of Call Graphs for Open Source Projects**

**CS590BD: Big Data Analytics and Apps**

**Instructor: Prof. Yugyung Lee**

**Project Group: Independent Research**

**Team Members**

**Rashmi Tripathi – 41**

**Megha Nagabhushan- 24**

# Project Objectives

## Significance

Developers reuse the existing python open source projects to develop their own new projects. Today, there are a lot of open source projects available like on Github etc. These projects contain huge lines of code. Determining the flow of code in such projects becomes challenging for the developer. By using call graphs, the developer will be able to know the functionality and various dependencies like environmental, functional, library etc. in an open source project.

This project will help the developer to know the dependencies, functionalities and the flow of control for open source project using virtual reality.

## Features

Below are the features relevant to the project:

1. Extraction of source code from open source repositories like Github.
2. Creation of evidence list for the features for easier analysis.
3. Analysis of code to identify the dependency with the help of evidence list.
4. Study of TensorFlow big data framework
5. Study of possible Machine learning Algorithms and data mining tools.
6. Loading the extracted information in TensorFlow
7. Proposing a model by developing an algorithm for the solution.
8. Creating a feature index using the existing code base.
9. Predict the features of the other new codebases.
10. Implementation of algorithm and indexing in python using possible mining tools.
11. Integrating all modules to Web interface which internally has indexing, implementation of algorithm and database.
12. Using virtual reality showcase the both static and dynamic call graph.

# Approach

## Analytic Tools

1. Using Pycallgraph library:  
Pycallgraph is an existing library that generates call graph for a Python module. The library can be modified to accommodate open source projects with many modules.
2. Using inspect module of Python:  
The inspect module provided by Python can be used to know the frame information for each method. The frame information can be logged. The logged information can be used to generate a good visualization for the project.

### Expected Inputs/Outputs:

For any organization, it will be interesting to adapt to a system which can train people on open source projects quickly. The combination of virtual reality world and call graphs can be proven best for the same. Users should have an application where they can observe these call graphs for any open source project in virtual reality world. With the motion of their hand like may be a swipe action , the application will reflect all the call graphs for given user program to human eyes. User should be able to see the various entry points, middle ones and the exit ones. Based upon the performance and the shortest path between an entry and an exit point User can easily learn and start making decision on what can be the best program for any particular goal.

Input	Output
Static	Here are possible static call graphs for TensorFlow module will be projected with all entry points and exit points.
Dynamic execution of sample TensorFlow Program	Here call graph of all execution points will be shown.

### Algorithms

Work in progress

## Related Work

### Open Source Projects

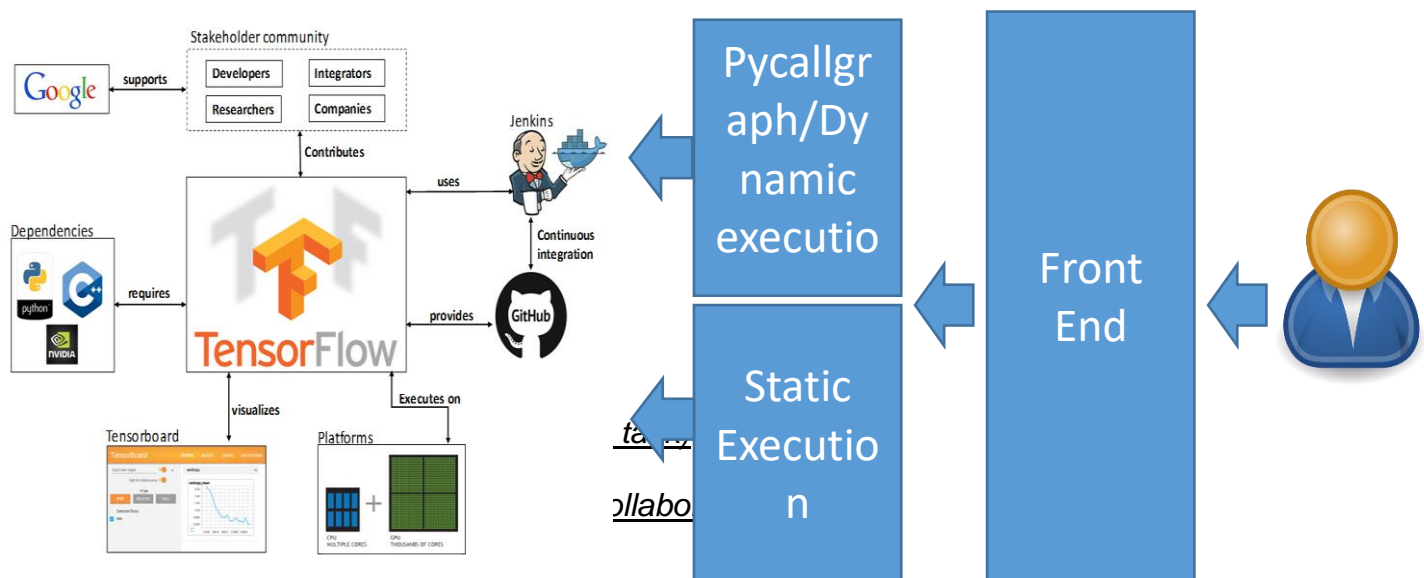
TensorFlow <https://github.com/tensorflow/tensorflow>

Pycallgraph <https://github.com/gak/pycallgraph>

## Application Specification

### System Specification

## Software Architecture



## Feature Specification

Following are the features as part of this project implementation:

1. Extraction of source code from open source repositories like Github.
2. Creation of evidence list for the features for easier analysis.
3. Analysis of code to identify the dependency with the help of evidence list.
4. Study of TensorFlow big data framework
5. Study of possible Machine learning Algorithms and data mining tools.
6. Loading the extracted information in TensorFlow
7. Proposing a model by developing an algorithm for the solution.
8. Creating a feature index using the existing code base.
9. Predict the features of the other new codebases.
10. Implementation of algorithm and indexing in python using possible mining tools.
11. Integrating all modules to Web interface which internally has indexing, implementation of algorithm and database.
12. Using virtual reality showcase the both static and dynamic call graph.

### Operation Specification (Input/output, exceptions)

Input	Output
Static	Here are possible static call graphs for TensorFlow module will be projected with all entry points and exit points.
Dynamic execution of sample TensorFlow Program	Here call graph of all execution points will be shown.

### Existing Applications/Services Used

TensorFlow <https://github.com/tensorflow/tensorflow>

Pycallgraph <https://github.com/gak/pycallgraph>

## Documentation & Implementation

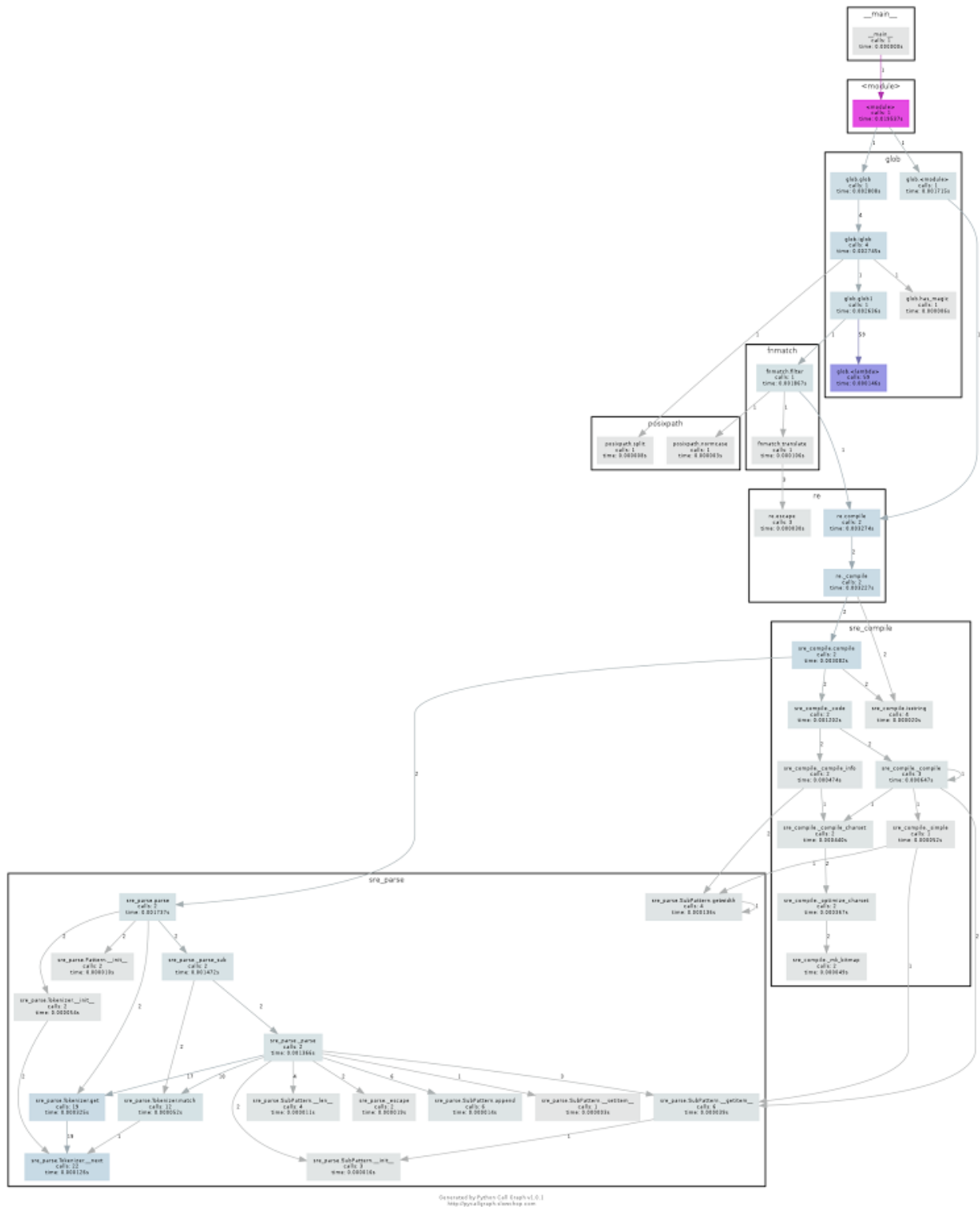
### Screenshots of sample python Program dynamic call graph using PycallGraph

#### Input Program:

```
import glob
# glob supports Unix style pathname extensions
python_files = glob.glob('*.py')
for file_name in sorted(python_files):
    print '    -----' + file_name

    with open(file_name) as f:
        for line in f:
            print '        ' + line.rstrip()

print
```



## Project Management

Project Timelines, Members, Task Responsibility:

← → ↻ GitHub, Inc. [US] | [https://github.com/rashmitripathi/DeepLearning\\_VirtualReality\\_BigData\\_Project/projects#boards](https://github.com/rashmitripathi/DeepLearning_VirtualReality_BigData_Project/projects#boards)

This repository Search Pull requests Issues Gist ToDo

rashmitripathi / DeepLearning\_VirtualReality\_BigData\_Project

<> Code 4 Issues 4 Pull requests 0 Boards Reports Projects 1 Wiki

View Repos (1/1) Show one Labels Milestones Assignees Epics

New Issues 0 Icebox 0 Backlog 0 In Progress 1 Review/QA 0 Done 3

DeepLearning\_VirtualReality... #4  
Study of TensorFlow big data framework  
Increment 1

DeepLearning\_VirtualReality... #1  
Extraction of source code from open source repositories like Github.  
Increment 1

DeepLearning\_VirtualReality... #2  
Analysis of code to identify the dependency with the help of evidence list  
Increment 1

DeepLearning\_VirtualReality... #3  
Creation of evidence list for the features for easier analysis  
Increment 1

Work completed:

No open issues in Increment 1.



## Work to be completed

In next increment we need to come up with algorithms needed for static graph and dynamic graph analysis.

## Challenges

Existing libraries create call graphs that are huge and less incomprehensible. We have to modify the existing libraries so that they provide better visualization by using virtual reality techniques.

Also, the existing libraries create call graph for just one python module.

## References

1. K.T. Stolee, S. Elbaum, and D. Dobos, "Solving the Search for Source Code," ACM Trans. Software Eng. and Methodology, vol. 23, no. 3, 2014, article 26.
2. Cain, Harold W., Barton P. Miller, and Brian JN Wylie. "A callgraph-based search strategy for automated performance diagnosis." European Conference on Parallel Processing. Springer Berlin Heidelberg, 2000.
3. T. Menzies and T. Zimmermann, "Software Analytics: So What?," IEEE Software, vol. 30, no. 4, 2013, pp. 31–37.
4. Herbsleb, James, Christian Kästner, and Christopher Bogart. "Intelligently Transparent Software Ecosystems." IEEE Software 33.1 (2016): 89-96.
5. Witten, Ian H., et al. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.
6. Krishnan, Malathy. Feature-based analysis of open source using big data analytics. Diss. Faculty of the University Of Missouri-Kansas City in partial fulfillment Of the requirements for the degree MASTER OF SCIENCE By MALATHY KRISHNAN BE, Anna University, 2015.