Software Platform for Volunteered Computing using P2P

CS4089 Project

Midterm Evaluation

K C Sreevasthavan, Giridhar G Nair, Irfan T Naushad
Guided By: Dr. Priya Chandran

March 17, 2016

Outline

Introduction

- In this semester we have begun the implementation of our platform. In this presentation we will be giving an overview of the work done till now, and also what we hope to complete by the final evaluation.

Problem Statement

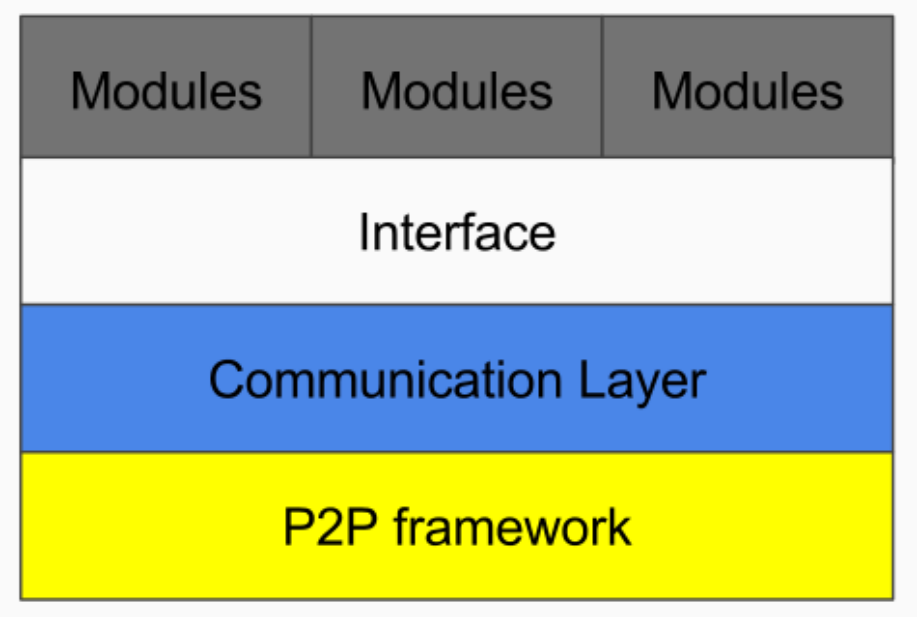
- To implement a software platform for computing that uses P2P for communication / data-sharing between volunteering nodes.

Literature Survey

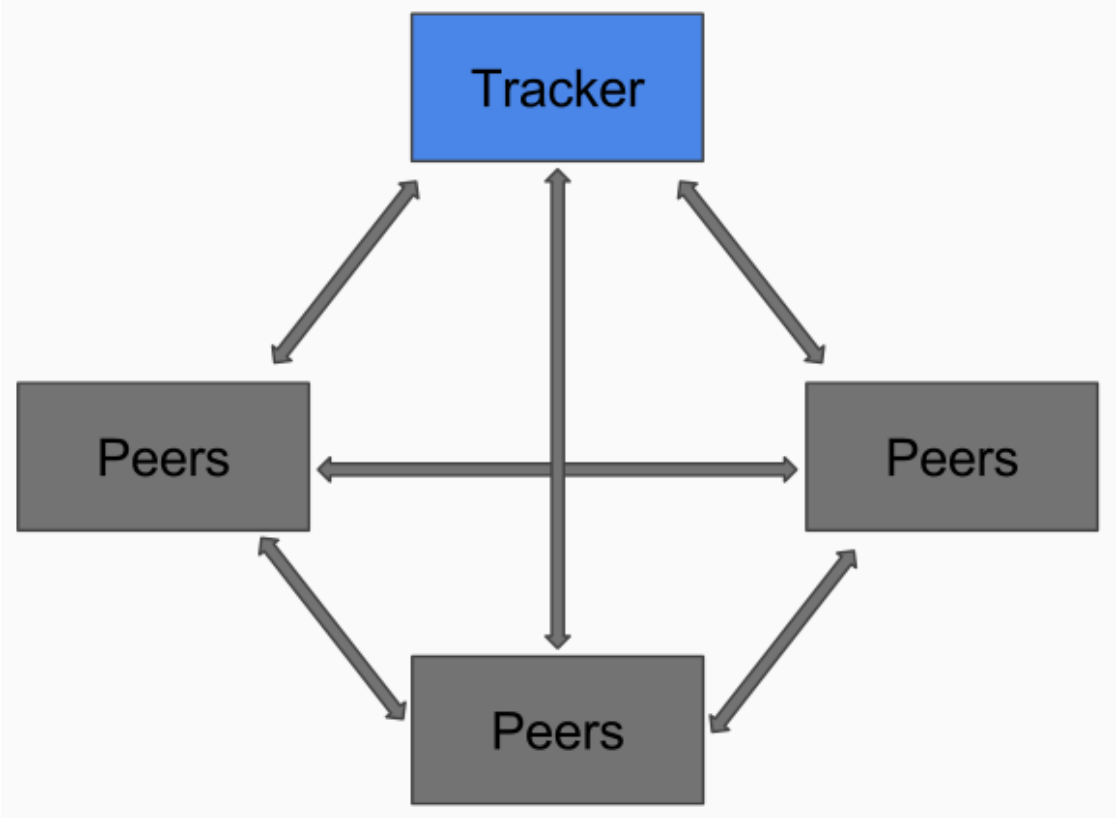
Various P2P frameworks were considered such as Kurin's Py-P2P[1], Sameul Chen's[2] and Twisted Networking Engine[3]. Official python documentation [4] is being used as reference for development.

Work Done

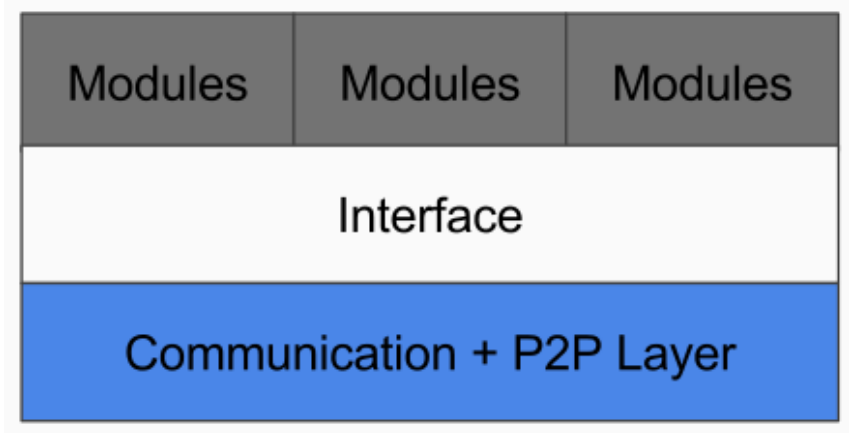
- The design of the platform we arrived in the previous semester.



- Intercommunication between the nodes.



- We have revised the design of the platform.



- Integrated Communication and P2P layer into one single layer
- Tracker nearly finalised. Handles Peerlist and modulelist manipulations, Account creation and login.

- Message format finalised.

```
# Message format
-------------------------------------------------------------------------

# Authentication - login message
{"type":"login","uname":"username", "pwd":"password", "port":"port", "status":"active/offline/busy"}

#Authentication - signup message
{"type":"signup", "uname":"username", "pwd":"password", "port":"port"}

# File transfer - peerlist download request
{"type":"get", "content":"peerlist"}

# File transfer - modlist download request
{"type":"get", "content":"modlist"}

# sending the details of the modules present
{"type":"modlist", "modlist":(set of module names) }

# Module request message
{"type":"get", "content":"module", "modname":"modulename"}

# Check peer status message
{"type":"status", "uname": "username", "status":"active/offline/busy"}

# computation / volunteer request message
{"type":"request", "title":"title of the project", "info":"description about the proj",
 "modname":"name of required module"}

# connection close
{"type":"quit"}

# Response Codes
-------------------------------------------------------------------------

# login response message
{"type":"rlogin", "content":"yes/no"}

# signup response message
{"type":"rsignup", "content":"yes/no/exists"}

# file transfer response message
{"type":"rget", "response":"yes/no"}

# computation / volunteer request message response
{"type":"response", "reply":"yes/no"}
```

Future Work

- Completion of networking layer (communication + P2P layer) of the platform
- Implementation of Interface layer

References I

[1] Kurin's Py-P2P `https://github.com/kurin/py-p2p`, Toby Burress.

[2] P2Python `https://github.com/samuelchen/P2Python`, Samuel Chen.

[3] Twisted Event Driven Networking Engine `https://twistedmatrix.com/trac/`, Twisted Matrix Labs.

[4] Python 3.5.1 Tutorial `https://docs.python.org/3/tutorial/` Python Software Foundation.