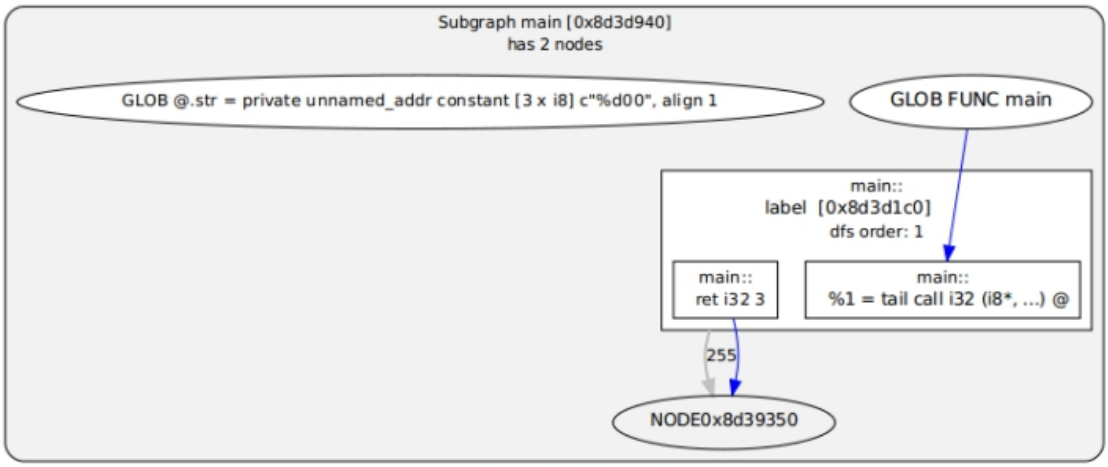
Week02 Report(22 Jul - 28 Jul) - Guangyao Xu

1. Work

1.1 generating llvm using dg tool

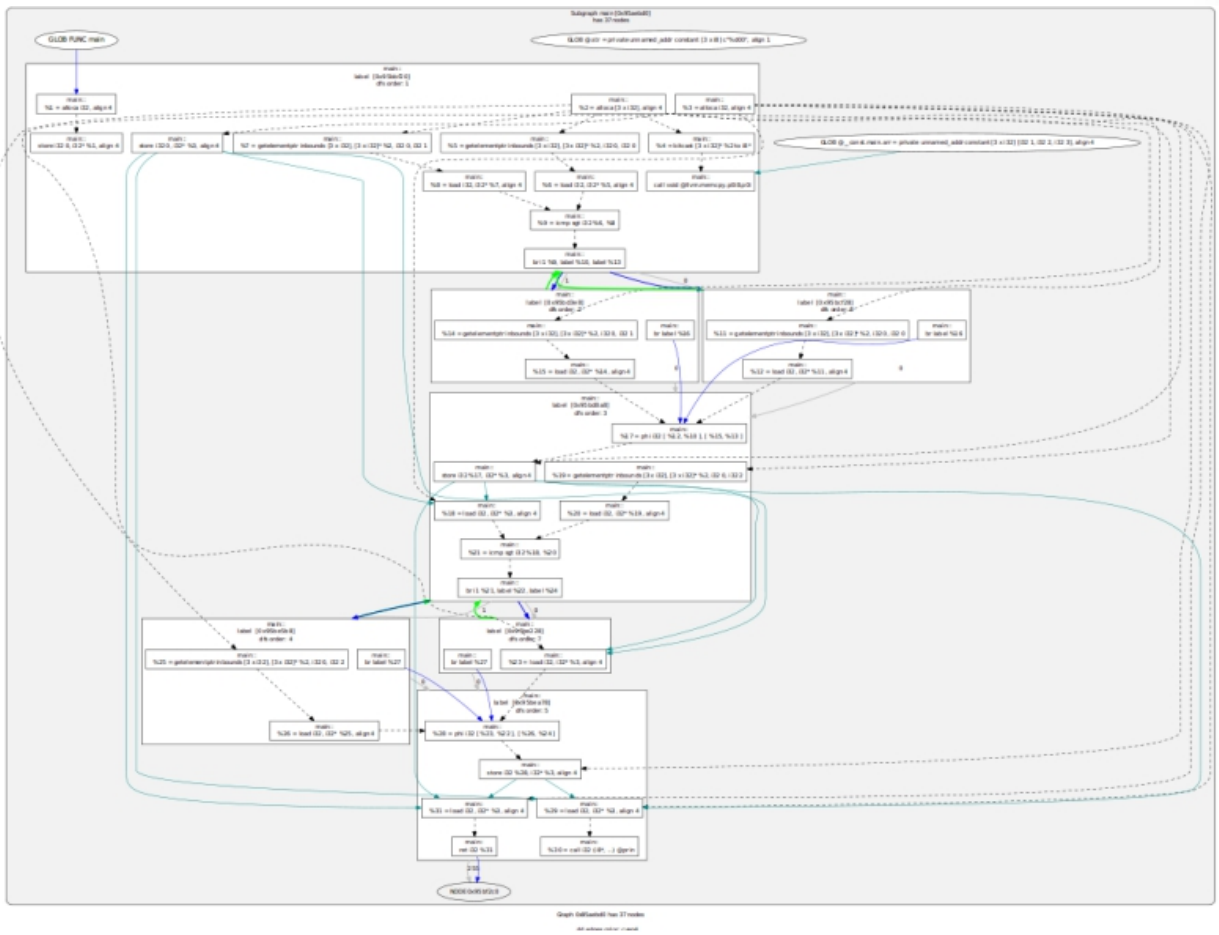
I use dg tool to extract .dot file from the C Program in my ubuntu 14.04, VMware, using the following command:

/llvm-dg-dump -no-cfg -no-use -no-data ~/UIUC/sum3.bc > ~/UIUC/sum3-cd.dot



Graph 0x8d3d940 has 2 nodes

dd edges color: cyan4
use edges color: black, dashed
cd edges color: blue
cfg edges color: gray



1.2 reading paper about methods to computing the similarity of graphs

I read a paper 'A Method to Evaluate CFG Comparison Algorithms' which comparing the efficency and time cost of four methods when calculate the similarity of graphs. And in this paper, the author claims that the edit distance method is the best methos not only in efficency but also in time cost. So I decided to choose Edit Distance as our method to calculate the similarity of graphs.

More details about this paper can be seen on this website: http://cfgsim.cs.arizona.edu/qsic14.pdf

1.3 finding tools to evaluate the edit distance between graphs

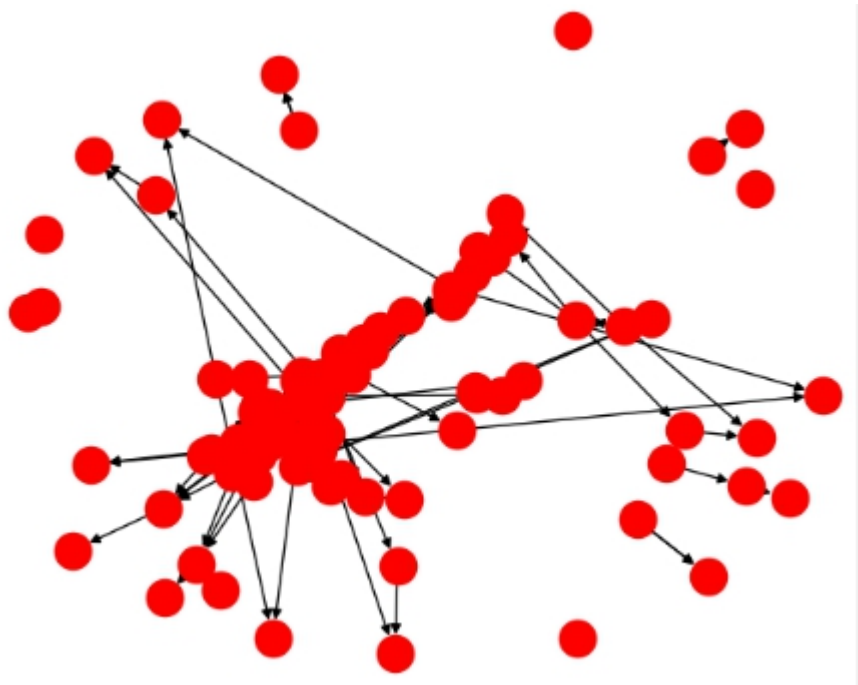
After deciding the algorithm to calculate the similarity of graphs, I started to find existing tools to help me calculate the similarity of graphs. And I found those tools: GMatch4py(a open source tool on github), Networkx(a popular tool for graph computing), Pydot(a python module to transfer dot file to graph structure)……

And I will use GMatch4py to calculate the approximated Edit Distance, use Networkx to generate the input data of the computing and Pydot to extract data from dot file.

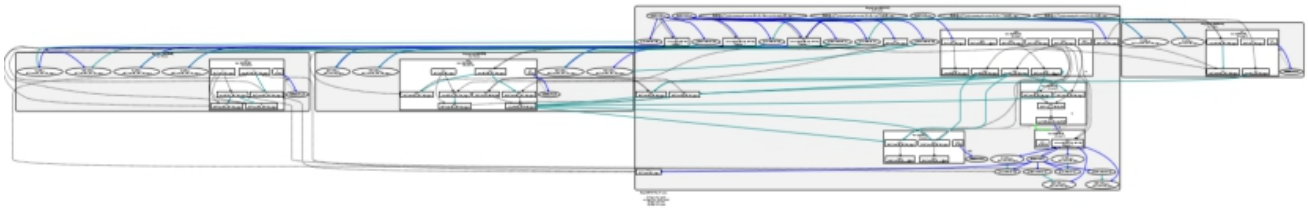
1.4 coding to extract the subgraph structures from dot file and tansfer it to standard networkx data structure

While programming, I encountered a problem that the program crushed when I tried to read dot file straightly from dot file using networkx's read_dot method. Then I found the answer on the stackoverlfow that networkx currently not support the nested structure of dot file, but the dot file generated by dg tool is nested. So I have to transfer the subgraph stucture to non-subgraph structure. I use pydot to finis this job. First, I traverse all subgraphs in dot file and push their nodes into one set. Then, I use DFS to traverse all edges in those subgraphs and put them into one graph. What's more, to increase the robust of program, I initialized all nodes with a attribution 'label'(if node doesn't have original value, the value will be '')

The extracted graph from the dot file transfered to a graph in data structure of networkx and looked like this:



The graph in original dot file looks like this:



1.5 displacing the original algorithms of edit distance which is NP problem with a new algorithm which is quadratic time

The Graph Edit Distance algorithm is NP problem, and its time cost will be huge. So I read papers about how to improve the efficency of Edit Distance algorithm. Then, I found some algorithms in this paper:

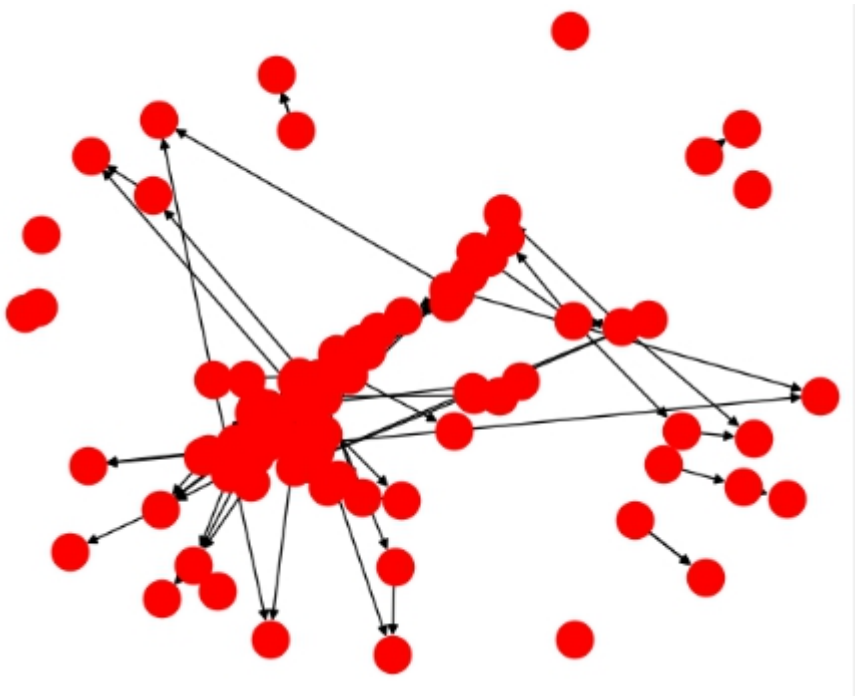
Fischer, A., Riesen, K., & Bunke, H. (2017). Improved quadratic time approximation of graph edit distance by combining Hausdorff matching and greedy assignment. Pattern Recognition Letters, 87, 55-62.

These improved algorithms can decrease the time cost to quadratic time cost, which is awesome. So I decided to use these improved algorithms rather than the original algorithm.

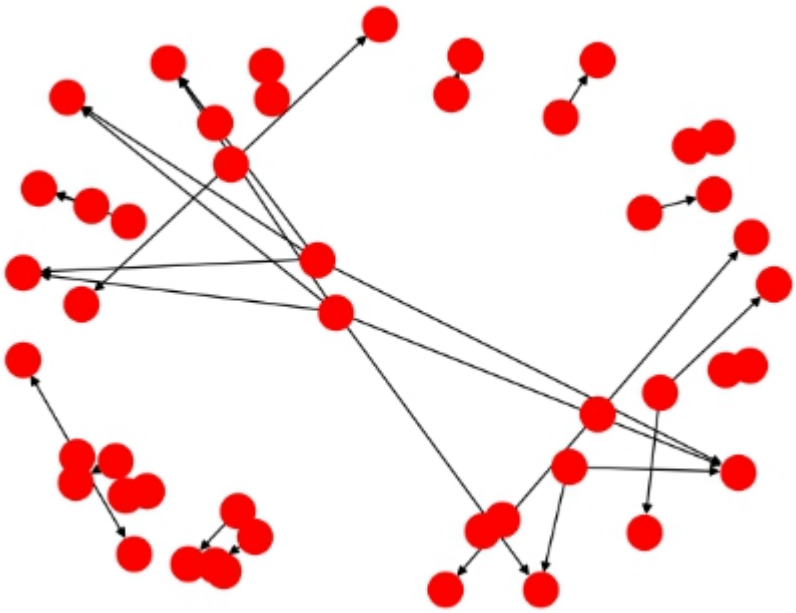
1.6 extracting CFG, DD, CD seperately from the original graph

Because the CFG,DD,CD can be totally different, and their meanings is also different in program analysis. So wrote a program to extract these three graphs seperately from the original huge graph.

Original Graph:



Extracted DD graph:



Plus, I delete nodes whose degree is zero. Because these nodes will influence the accuracy of the algorithm.

1.7 discussing with Angello Astorga about some detail of the program

Friday, I discuss with Angello Astorga about some detail of this research and solutions to decrease the time cost of my program.

1.8 Simplely summarize the process of calculating similarity

1).Using LLVM tool (llvm-dg-dump in dg) to generate dot file containing CFG, CD, DD graphs.

2). extract subgraph from dot file

2.1) using pydot to transfer dot file to graph data structure in pydot

2.2) using pydot to search all of the subgraph in the dot file (dot file generated by llvm-dg-dump containing many subgraphs which cannot be straightly transfer to data structure in networkx)

2.3) extract three different graphs in dot file , containing DD, CD, CFG

2.4) transfer the three graphs to standard networkx graphs

3). Using optimized edit distance to evaluate the similarity between two graph

4). combine three results of the similarity and get the finial result of the similarity evaluation.

2. problems

The problems that I stiil have while doing this research, and will be resolved in future work.

2.1 Which improved Edit Distance methods is the best?

2.2 Whether should we take the label of nodes into consideration when we compare the similarity of graphs?

2.3 How to combine the similarity of CFG,DD,CD into one similarity? How to choose the weight of each graph?

2.4 Are there any other factors taking into consideration to help to improve the accuracy of similarity?