Detection of Code Cloning with Refactoring Technique

Ramandeep Kaur
M.Tech Student, Department of Information Technology
Chandigarh Engineering College, Landran, Punjab, India

Gurdeep Kaur
Assistant Professor, Department of Information Technology
Chandigarh Engineering College, Landran, Punjab, India

Parminder Singh
Assistant Professor, Department of Information Technology
Chandigarh Engineering College, Landran, Punjab, India

Abstract- The code cloning is the major issue in the industrial point of view. As the number of projects is increasing in this digital world; there is major challenge to verify the contents and identify the line of codes. In this paper we find the refactoring technique to analyze the code from the number of projects that was written in programming languages like C and Java. We identify the execution time and line of similar codes in the projects so that we can remove the bad code from the refactoring technique and improving the codes in the project. E.g. We analyze the two projects naming tureball and tracefirm ; those are written in the C and java language. We compare them with lines of code and find the similar code so that in the next time it would not be repeated. The execution that we find from the SPCP-mining tool and we said that code was effective because lines of code are not repeated and due to association rule we are obtaining improved code from the bad line of codes from the projects. This code cloning technique makes project better and highlighted bad line. The repeated or bad lines also being avoided in this case and the K mean algorithm has been effectively well on applied project cases.

Keywords- Code Cloning, Refactoring, LOC, source code, code fragment

I. INTRODUCTION

Code detection is most interesting area in the field of web mining and this is mostly used to detect the similar lines of code that is called as plagiarism. The reused and copied of code has been seen from many of projects. The similar code written in such a way that hide owns identity but functionality is same in the case which mentioned on earlier of projects. In software programs, we can find different kinds of replication or redundancy. There are numerous other factors such as performance improvement and coding style because of which large software systems may contain a significant proportion of duplicated code [1].

A. Clone Detection Tool:

A clone detector tool must try to find pieces of code of high similarity in a system’s source code or text. The main problem is that, it is not known initially which code fragments may be repeated [1]. Thus the detector really should compare every possible code of fragment with every other possible fragment . Such a comparison is expensive from a computational point of view and thus, several measures are used to reduce the domain of comparison before performing the actual comparisons.[2]

1. Reliability: The source code is more reliable when we remove the similarity of lines of code or when we remove unwanted parts to show more precious code in the software code that defines the reliability of the project. The source code mapped with other code from the same project but one condition is that it is fully independent; then we represent that it is reliable in other terms of software engineering.

2. *Source code:* The source code is written in editor to process or executed by the compiler defined by any programming language. The source is better when the uninteresting code should be removed and user gets the efficient program. Source code is partitioned into a number of modules and we say it can be represented as

fragments; the set of disjoint fragments called source units. The source unit includes file, class, function, block and sequencing of Lines of codes (LOC).

3. Maintenance: The code cloning is the process to remove whitespaces, commas and removal of comments. This method is to providing good management of lines of code and the project that was developed in the different programming languages having free from error code and removing repeated lines so at the end projects needs to be less maintenance of software/project.

B. Code Detection Process:

The clone detection in any software and web mining research area defined the process; the process that divided into number modules as we called process states and namely preprocessing, transformation, match detection, formatting, filtering and some aggregation process then finally go to clone pairs as for detection similarities. The preprocessing similar divide the clones as we did in this paper for detecting similarities then transform the code into appropriate intermediate representation. This transformation removes the unwanted free white space and commas.

Here, in the match detection we add some code in the project and then detect that whether it match from the lines of other project. This process is also called refactoring. Here we replace aggregation with refactoring. Source coordinates of each clone pair obtained in the comparison phase are mapped to their positions in the original source files. Identifying and ranking clones according to refactoring need can help us minimize refactoring effort and cost, because we are able to focus on just those clones that are important to be refactored, and we can leave the clones where refactoring is less important or unnecessary [8].

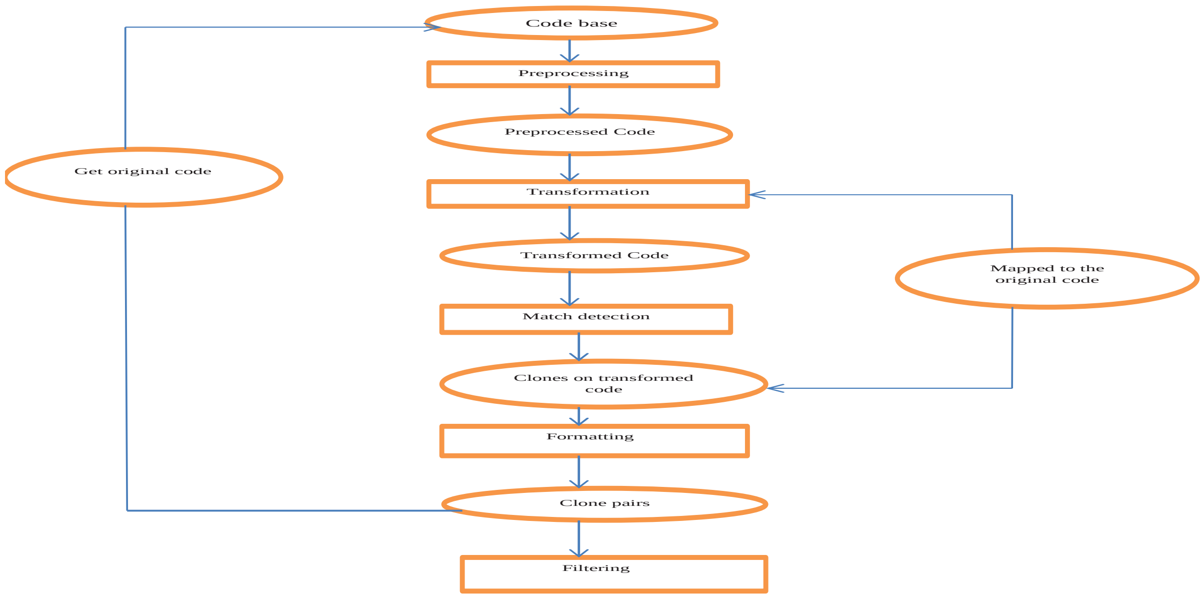

Figure 1: Generic Clone detection process [2]

II. RELATED WORK

The author [1] described the code cloning process and detects from the software's from high similarities in the source code. There are number of clone detection techniques and their corresponding tools, and therefore, a comparison of these tools is worth in order to pick the right tool for a particular purpose of interest. Then further authors compared four clone detection tools was syntax tree based tool, two token based tools CP-Miner, CCFinderX, and PMD a string matching based tool. Clone detector tool must try to find pieces of code of high similarity in a system's source code or text. The main problem is that, it is not known initially which code fragments may be repeated [2]. Initially, Dataset is in XML format. Initially, convert the project into dataset. A dataset stores information about nodes, hierarchies on these nodes, and edges between those nodes. The key indicates the name of the attribute and is always a string, whereas the value of the attribute can be a string, integer, or float. The attributes encode information about the nodes and edges. The Author's expect to report on more advanced clone detection and removal methods in the near future. This method of clone detection can also be implemented to more complex applications such as web based applications. The Author's [3] was to produce a better input for the generic pipeline model by processing smaller part of source code files before focusing on the large chunk of source codes in a single pipeline. They implemented and applied the proposed approach with the support of a tool called Java Code Clone Detector. The result obtained shows an improvement in the rate of code clone detection and overall runtime performance as compared to the existing generic pipeline model. Code reuse [4] is the concept of copying and pasting the code in multiple places in the same software or different software without modification. Clone detection is live problem in an active search area with plenty of work on detecting and removing clones from software. In the last few decades numerous code clone detection technique and tools have been proposed for capturing duplicated redundant code, which is also known as software clone. Code clone detection [5] is an enabling technology for plenty of applications, each having different requirements to a clone detector. In this author's presented a generic pipeline mode lf the code clone detection process. In JCCD source units are represented by subtrees of an AST. First, such a tree reflects the syntactical structure of a document at a certain level of detail. Hence, we are able to take syntactical information into account, while irrelevant information like whitespaces or comments need not be represented in the AST. Second, it is less sensitive to code restructuring. Thus, it is less sensitive to minor code modifications.

III. FRAMEWORK AND IMPLEMENTATION

SPCP-Miner [7] detects method genealogies and clone genealogies considering all the revisions of the subject system. Detecting the genealogy for a particular method involves identifying each instance of that method in each of the revisions where the method was alive. By detecting the genealogy of a method, we can determine how it changed during evolution [9]. The following figure represents the code fragment and detection is represented in figure 2 .

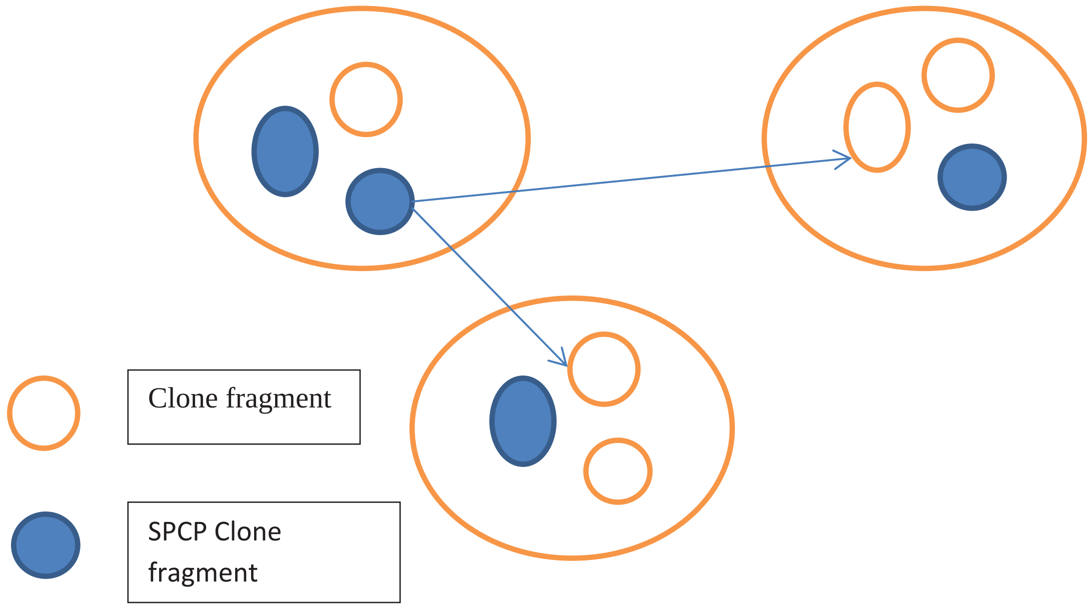

Figure 2: Clone fragment and non-fragment process [8]

The SPCP miner tool fragment and non-fragment the code in the project; the project which we have written in JAVA and C with little modification and added some similar lines of code then used miner tool to detect the similar lines. The SVN repository adds in the project and then it matches. If any match then we remove these similar lines and also to make project better.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this graph the relationship is present between the group-id and lines of code. This graph shows the detected cloned lines that is total number of lines of code which increases with the group-id and if the code is similar then chances to replace the code so that in next time less cost in software maintenance. Here, in this figure 3, line 2 having number of lines of code and it was analyzed that more detection and experimented on the line2 as compared to other lines. The total number of lines of code in the project has 1200 and we see that number of changes in case of line2 and line4. The number of groups taken as lines of code in different groups as 14 groups and find that we need to work on middle groups because higher the modification and changes desired by the developer.

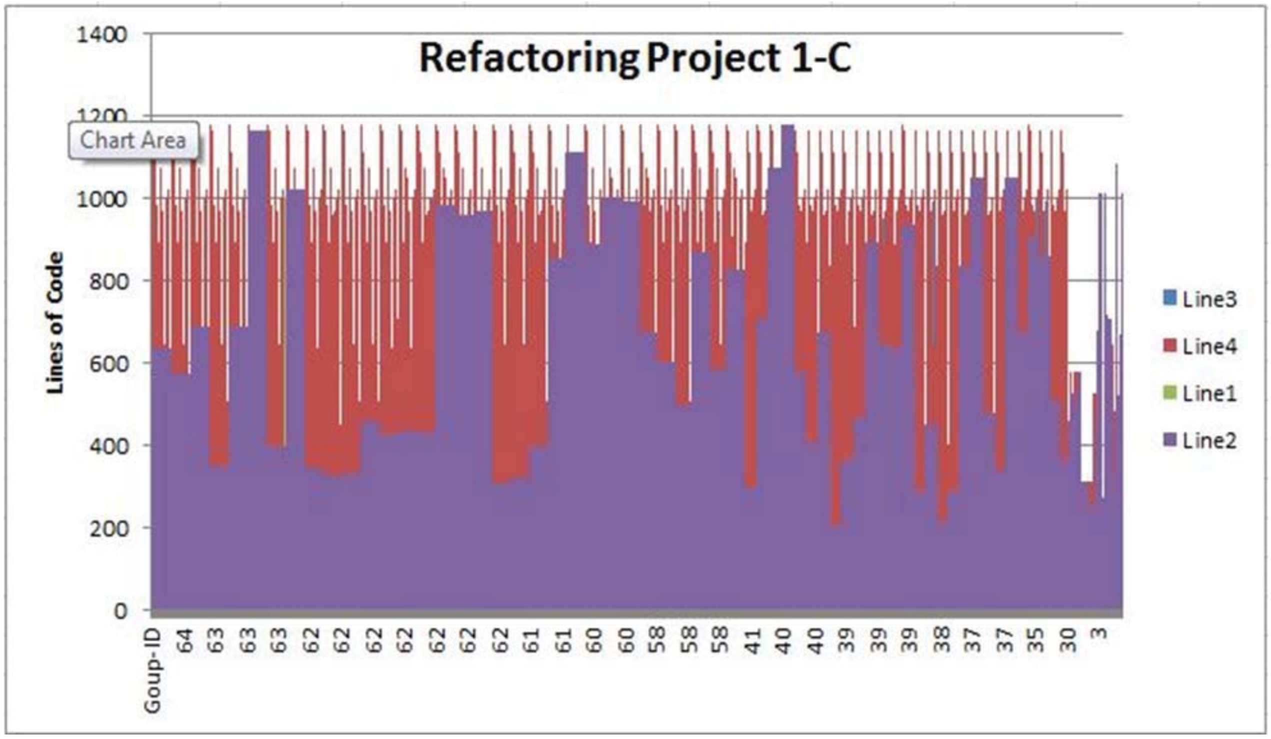

Figure 3: Project written in C

In this graph the relationship is present between the group-id and lines of code. This graph shows the detected cloned line that is total number of lines of code which increases with the group-id decreased.

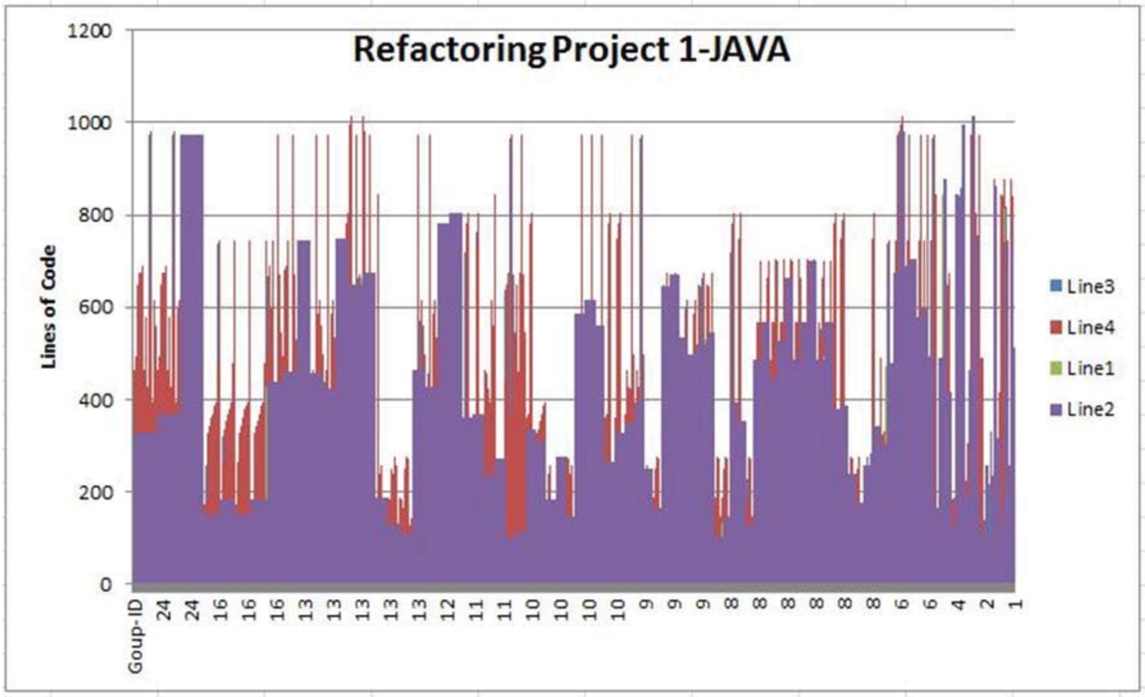

Figure 4: Project written in JAVA

In this graph the relationship is present between the group-id and lines of code. This graph shows the detected cloned line that is total number of lines of code which increases with the group-id decreased.

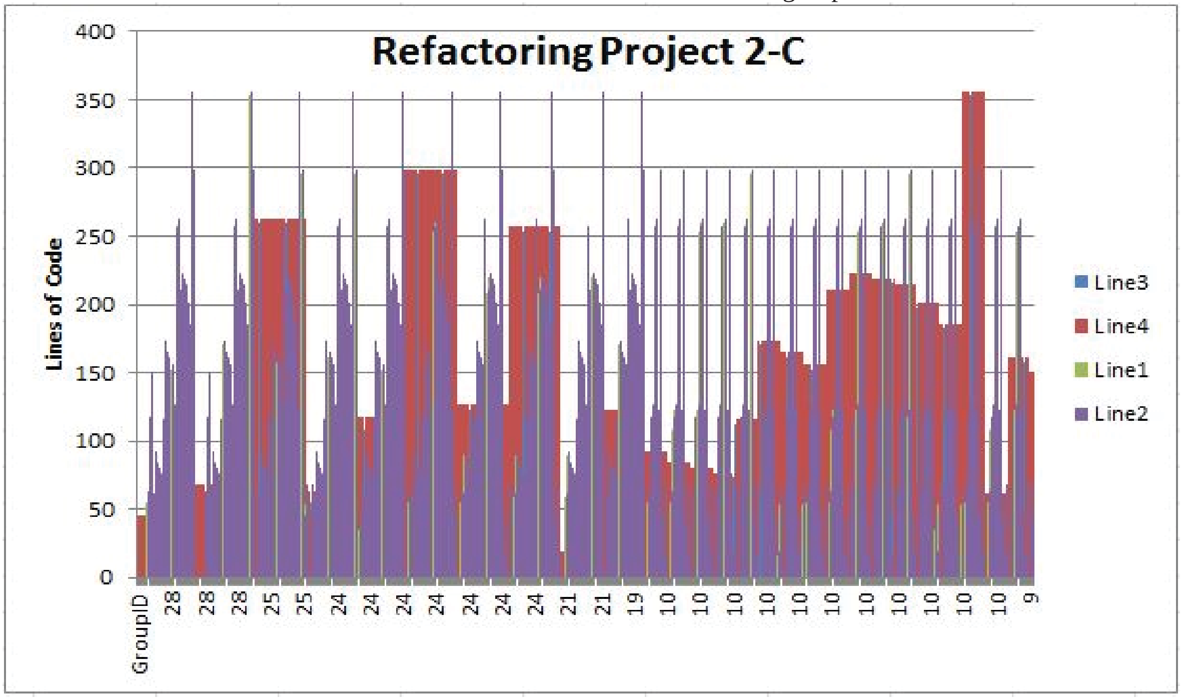

Figure 5: Project C-2

In this graph the relationship is present between the Group-id and lines of code. This graph shows the detected cloned line that is total number of lines of code which increases with the group id decreased.

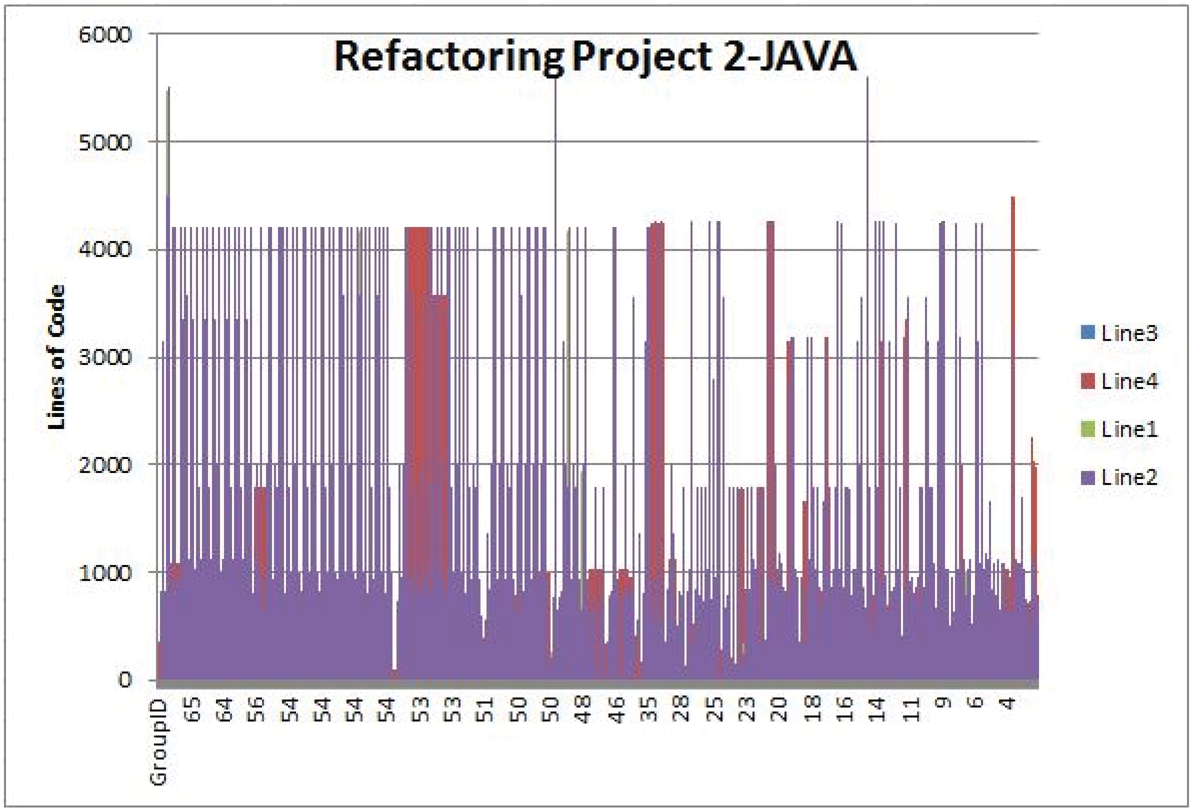

Figure 6 : Project Java-2

In this graph the relationship is present between the Group-id and lines of code. This graph shows the detected cloned line that is total number of lines of code which increases with the group id decreased.

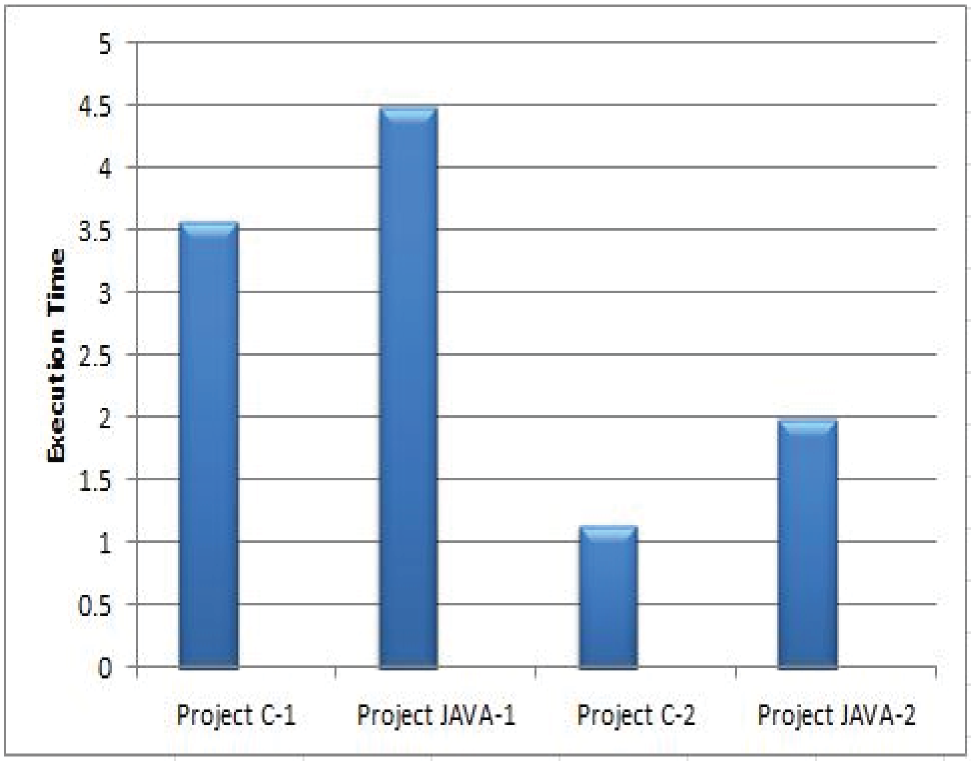

Figure 7: Execution time of different no. of projects

This graph shows the execution time taken by different number of projects that is Project C-1, Project JAVA-1, Project C-2 and Project JAVA-2. The execution time is more in Project JAVA-1 and less in JAVA-2 because we assume that it depends on the lines of code and Mining tool that has been used by the researcher.

V. CONCLUSION

This SPCP miner tool finds a significant amount of code clones. Identification and subsequent unification of simple clones is useful in software maintenance. Our main goal is to identify the total number of lines of code and from which how many of similar lines so that we remove these similar lines and to improve the code efficiency. In future this technique can be further extended to find more number of code clones on the basis of which the potential clones are found so that more accurate results can be obtained and the output can be integrated with other clone detection approaches to confirm whether the detected potential clones are actually clones or not.

REFERENCES

[1] Shahid Ahmad Wani, Shilpa Dang, "A Comparative Study of Clone Detection Tools", ijarcsms, 2015, pp.37-42.

[2] Deepak Sethi, Manisha Sehrawat, Bharat Bhushan Naib,"Detection of code clones using Datasets", IJARCSSE, 2012, pp.263-268.

[3] Al-Fahim Mubarak-Ali, Sharifah Mashita Syed-Mohamad1, Shahida Sulaiman,"Enhancing Generic Pipeline Model for Code Clone Detection Using Divide and Conquer Approach",2012.

[4] Harpreet Kaur, Rupinder kaur,"A Review: Clone Detection in Web Application Using Clone Metrics", IJCST,2014, pp.164-171.

[5] Benjamin Biegel, Stephan Diehl, "Highly Configurable And Extensible Code Clone Detection",WCRE,2010.

[6] M. Mondal, C. K. Roy, and K. A. Schneider, “Connectivity of Cochanged Method Groups: A Case Study on Open Source Systems”, Proc. CASCON, 2012, pp. 205-219.

[7] SPCP Miner. https://homepage.usask.ca/mam815/spcpminer/index.php

[8] M. Mondal, C. K. Roy, and K. A. Schneider, “Automatic Identification of Important Clones for Refactoring and Tracking”, Proc. SCAM, 2014, pp. 11 – 20.

[9] M. Mondal, C. K. Roy, and K. A. Schneider, “Automatic Ranking of Clones for Refactoring through Mining Association Rules”, Proc. CSMRWCRE, 2014, pp. 114 - 123.

[10] M. Mondal, C. K. Roy, and K. A. Schneider, “Comparative Stability of Cloned and Non-cloned Code: An Empirical Study”, Proc. SAC, 2012, pp. 1227 – 1234.