LLVM: built-in scalable code clone detection based on semantic analysis

Institute for System Programming of the Russian Academy of Sciences

Sevak Sargsyan : sevaksargsyan@ispras.ru
Shamil Kurmangaleev : kursh@ispras.ru
Andrey Belevantsev : abel@ispras.ru

Considered Clone Types

1. **Identical code fragments except whitespaces, layout and comments.**

2. **Identical code fragments except identifiers, literals, types, layout and comments.**

3. **Copied fragments of code with further modifications. Statements can be changed, added or removed.**

Considered Clone Types : Examples

Original source

```
4: void sumProd(int n) {
5:     float sum = 0.0;
6:     float prod = 1.0;
7:     for (int i = 1; i<=n; i++) {
8:         sum = sum + i;
9:         prod = prod * i;
10:        foo(sum, prod);
11:    }
12: }
```

Clone Type 1

```
void sumProd(int n) {
  float sum = 0.0; //C1
  float prod = 1.0; // C2
  for (int i = 1; i <= n; i++) {
  ____    sum = sum + i;
  ____    prod = prod * i;
  ____    foo(sum, prod);
  }
}
```

Tabs and comments are added

Clone Type 2

```
void sumProd(int n) {
  int s = 0; //C1
  int p = 1; // C2
  for (int i = 1; i <= n; i++) {
  ____    s = s + i;
  ____    p = p * i;
  ____    foo(s, p);
  }
}
```

Tabs and comments are added

Variables names and types are changed

Clone Type 3

```
void sumProd(int n) {
  int s = 0; //C1
  int p = 1; // C2
  for (int i = 1; i <= n; i++) {
  ____    s = s + i * i;
  ____    foo(s, p);
  }
}
```

Tabs and comments are added

Variables names and types are changed

Instructions are deleted, modified

Code Clone Detection Applications

1. **Detection of semantically identical fragments of code.**
2. **Automatic refactoring.**
3. **Detection of semantic mistakes arising during incorrect copy-paste.**

Code clone detection approaches and restrictions

Textual (detects type 1 clones)

1. S. Ducasse, M. Rieger, S. Demeyer, A language independent approach for detecting duplicated code, in: Proceedings of the 15th International Conference on Software Maintenance.

Lexical (detects type 1,2 clones)

1. T.Kamiya, S.Kusumoto, K.Inoue, CCFinder : A multilinguistic token-based code clone detection system for large scale source code, IEEE Transactions on Software Engineering.

Syntactic (detects type 1,2 clones and type 3 with low accuracy)

1. I. Baxter, A. Yahin, L. Moura, M. Anna, Clone detection using abstract syntax trees, in: Proceedings of the 14th International Conference on Software.

Metrics based (detects type 1,2,3 clones with low accuracy)

1. N. Davey, P. Barson, S. Field, R. Frank, The development of a software clone detector, International Journal of Applied Software Technology.

Semantic (detects type 1,2,3 clones, but has big computational complexity)

1. M. Gabel, L. Jiang, Z. Su, Scalable detection of semantic clones, in: Proceedings of the 30th International Conference on Software Engineering, ICSE 2008

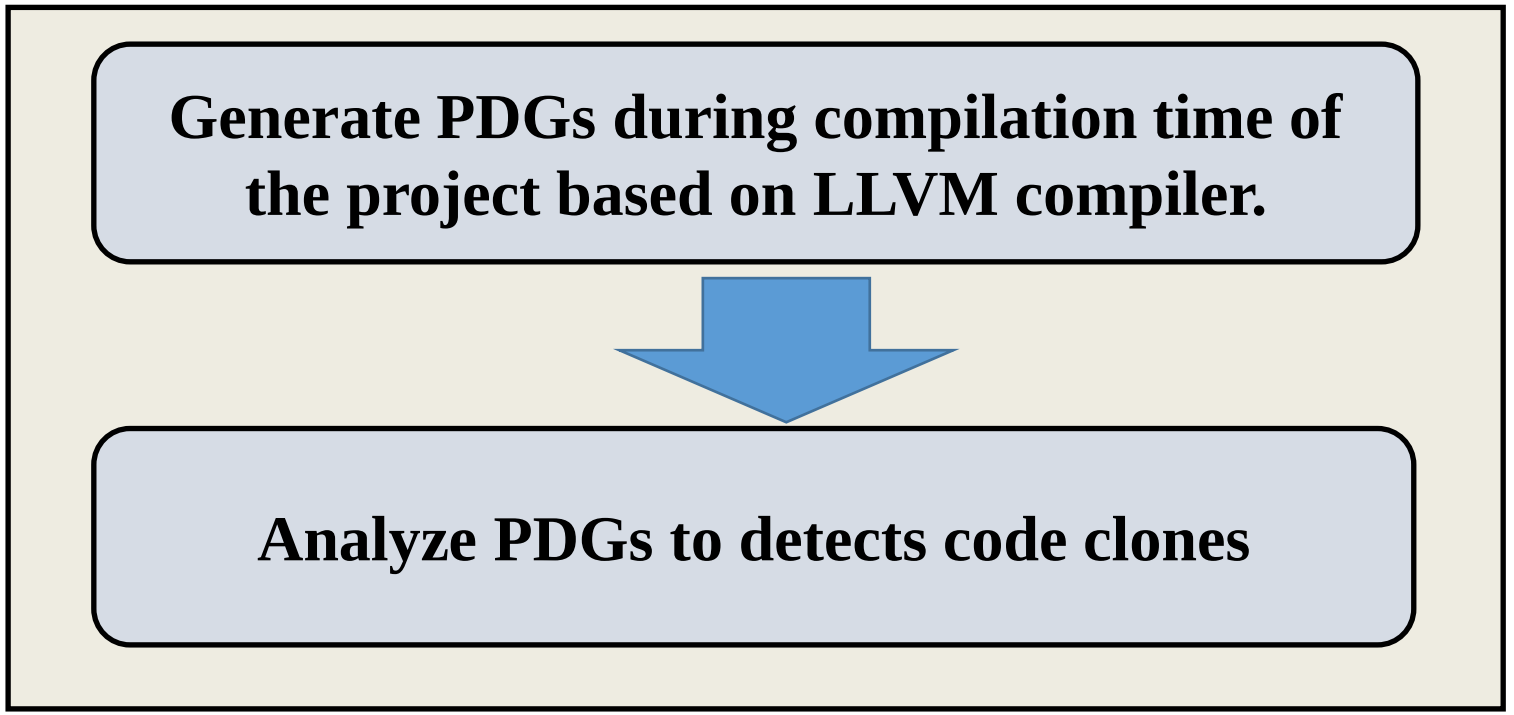
Formulation Of The Problem

Design code clone detection tool for C/C++ languages capable for large projects analysis.

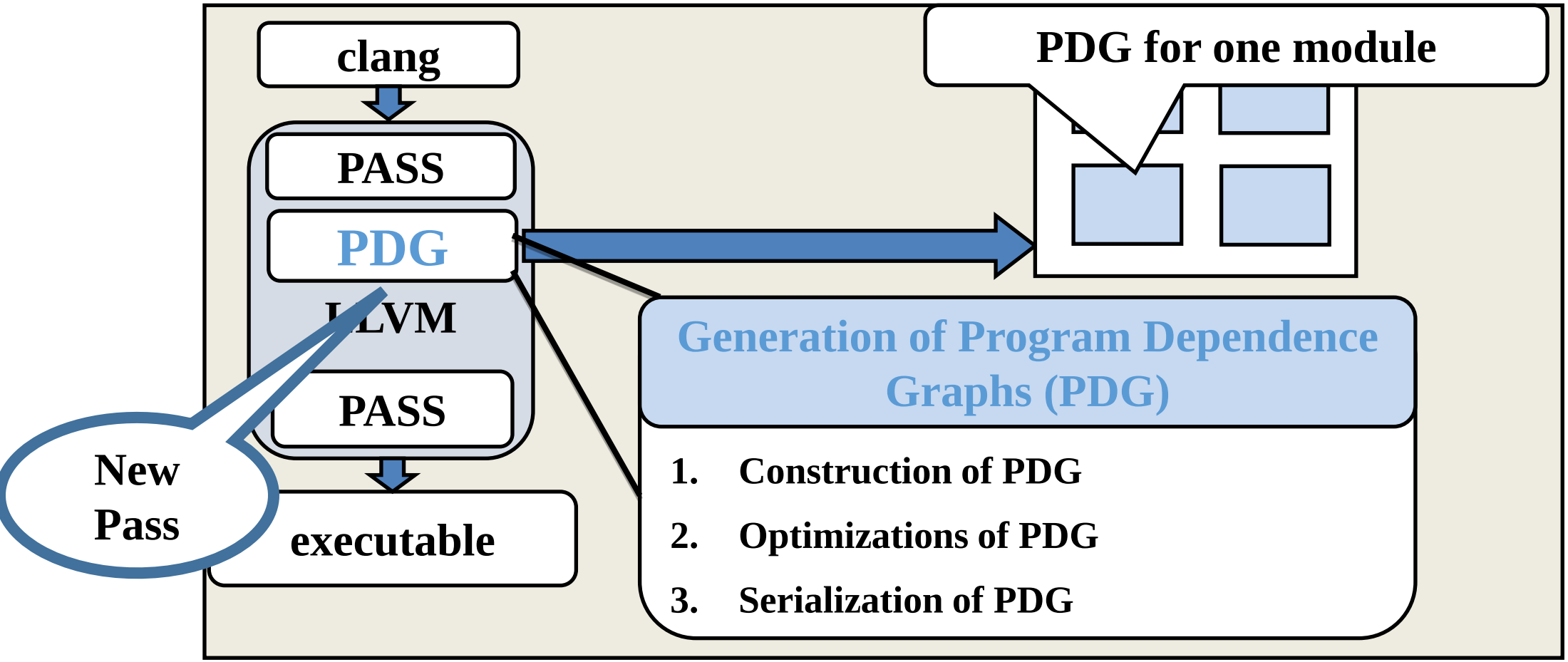
Requirements :

- **Semantic based (based on Program Dependence Graph)**
- **High accuracy**
- **Scalable (analyze up to million lines of source code)**
- **Detect clones within number of projects**

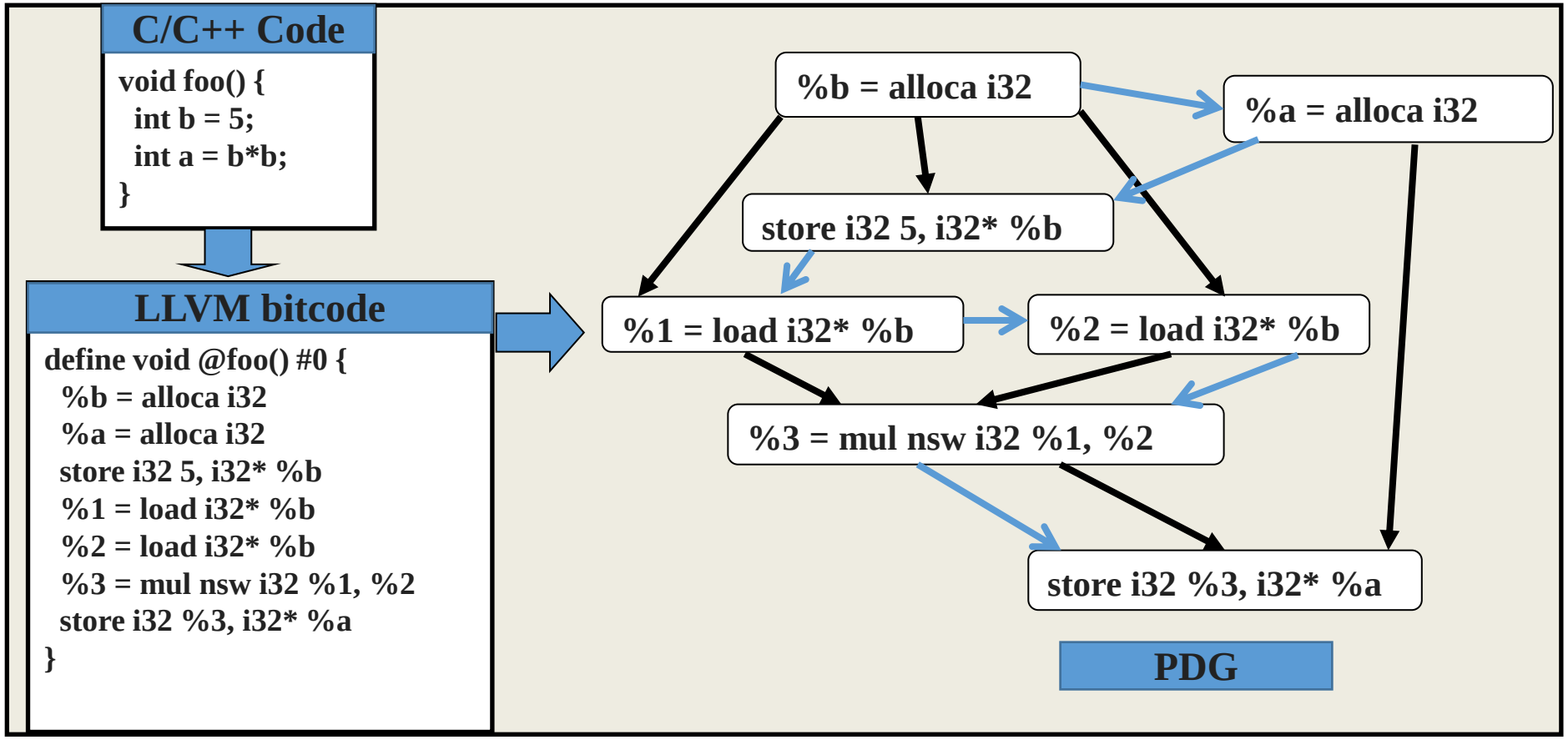
Architecture



Architecture : PDGs' generation



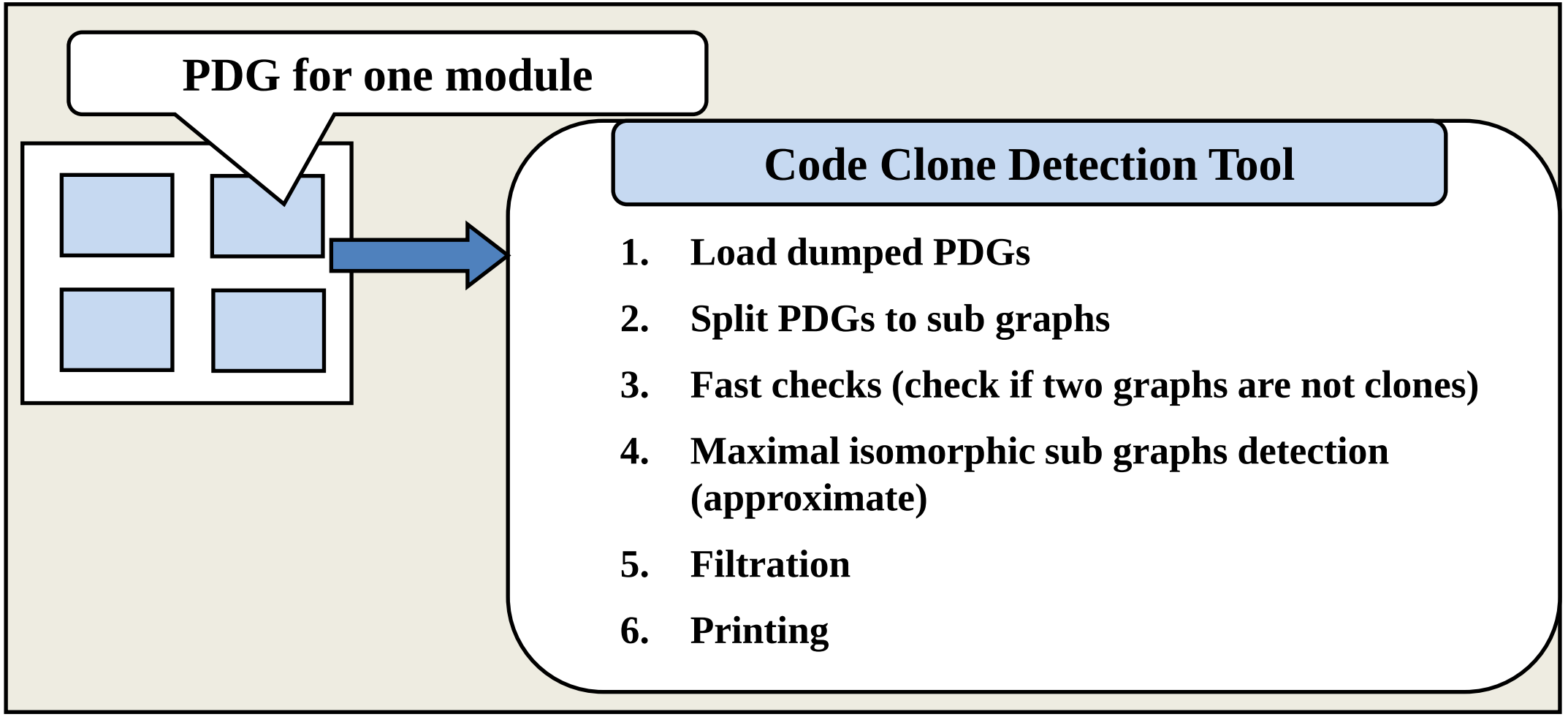
Example of Program Dependence Graph



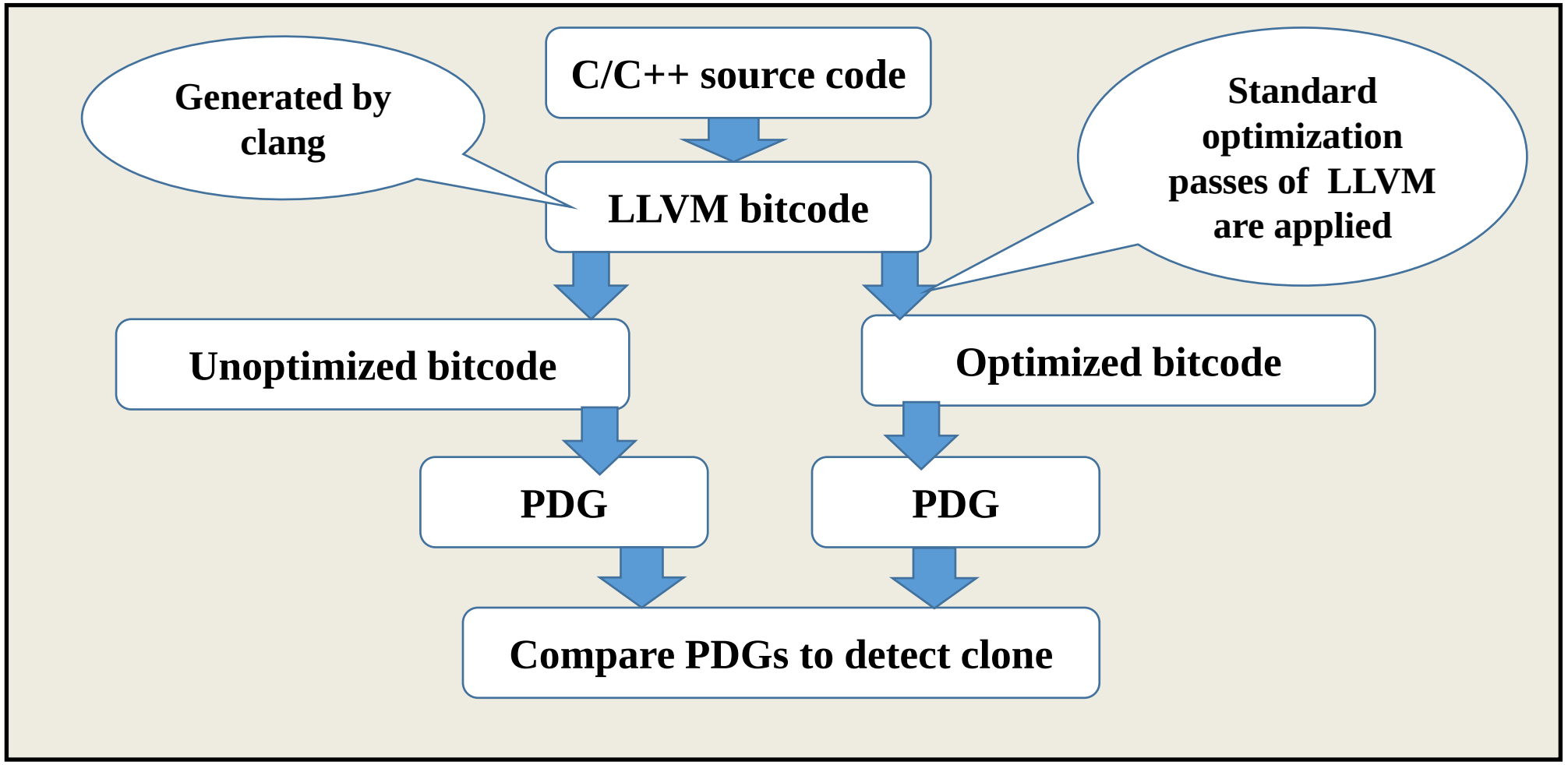
Edges with blue color are control dependences

Edges with black color are data dependences

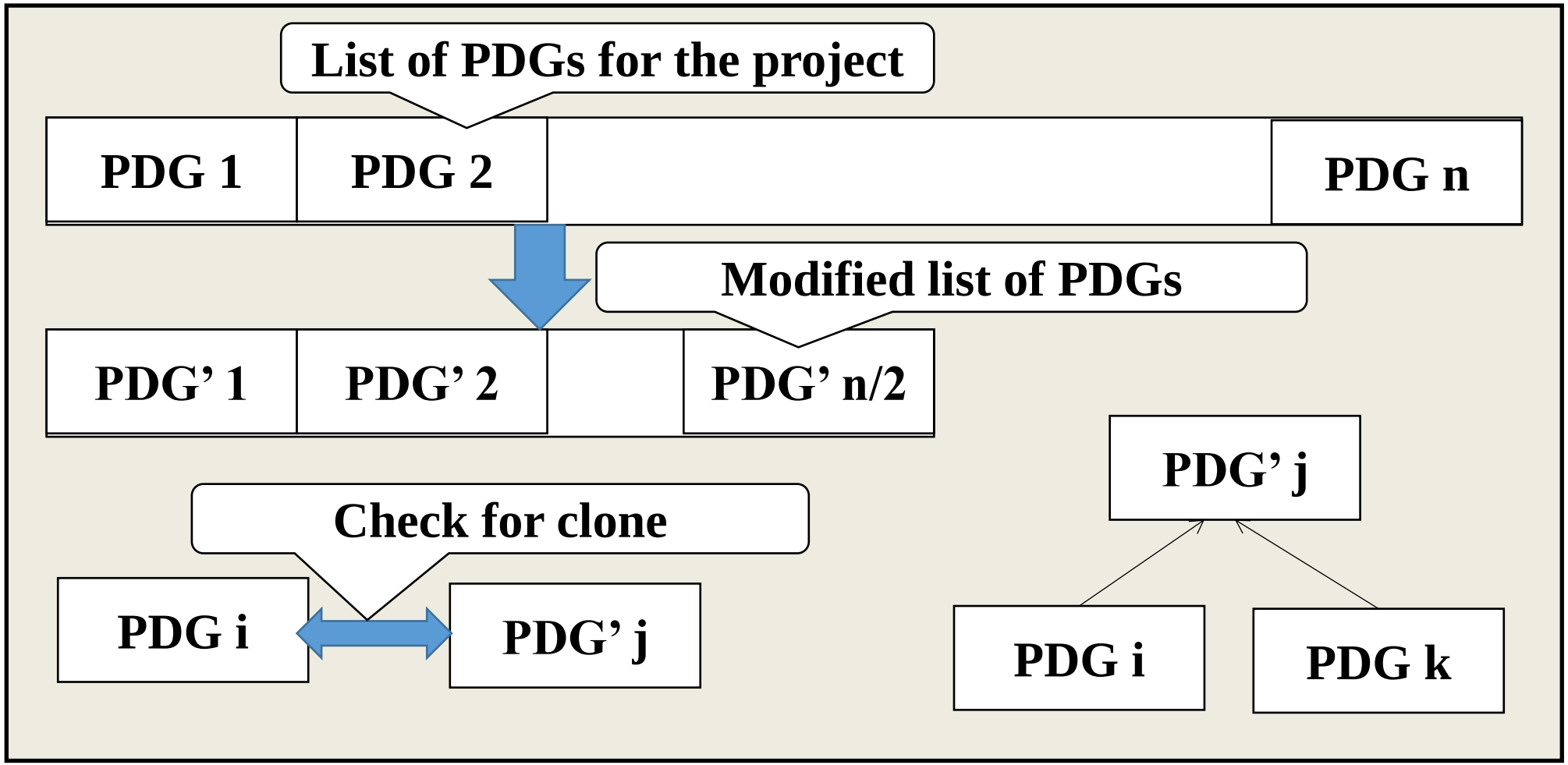
Architecture : PDGs' analyzes



Automatic clones generation for testing : LLVM optimizations



Automatic clones generation for testing : PDGs' marge



Advantages

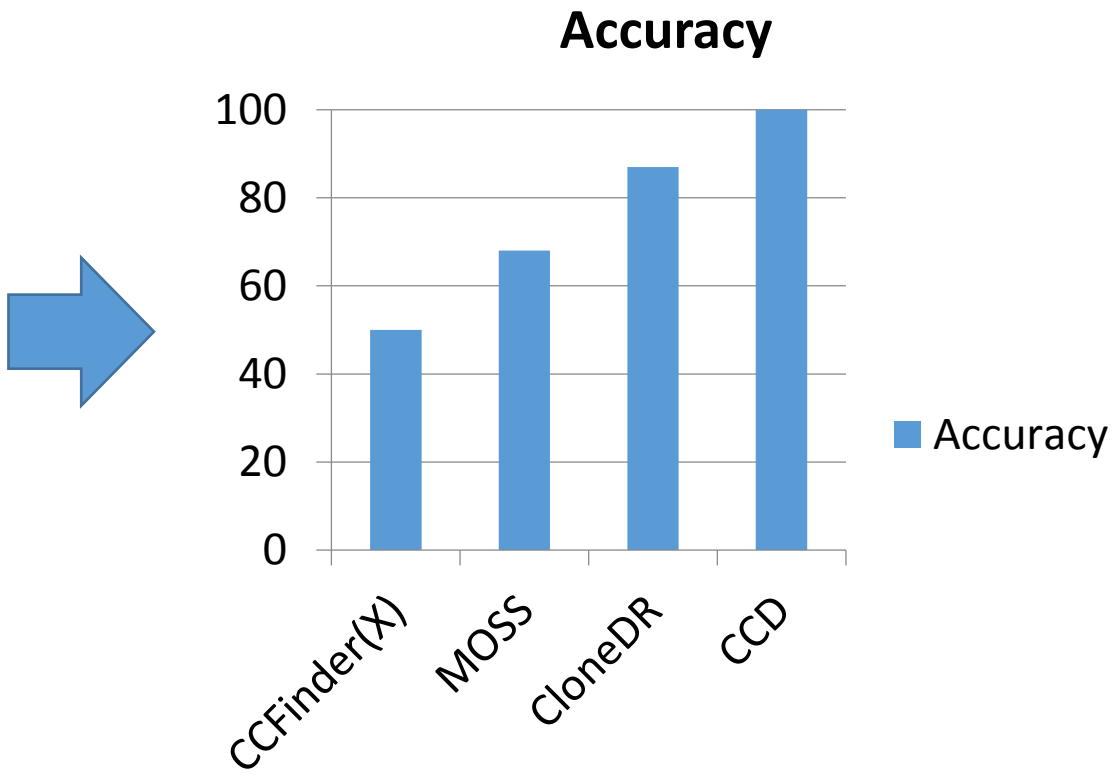
1. Compile-time very fast generation of PDGs.
2. No need of extra analysis for dependencies between compilation modules.
3. High accuracy (above 90 %).
4. Scalable to analyze million lines of source code (C/C++).
5. Possibility to detect clones within list of projects.
6. Possibility for parallel run.
7. Opportunity of automatic clones generation for testing.

Results : comparison of tools

All tests are clones. One original file was modified to obtain all 3 types of clones [1].

Test Name	CCFinder(X)	MOSS	CloneDR	CCD
copy00.cpp	yes	yes	yes	yes
copy01.cpp	yes	yes	yes	yes
copy02.cpp	yes	yes	yes	yes
copy03.cpp	yes	yes	yes	yes
copy04.cpp	yes	yes	yes	yes
copy05.cpp	yes	yes	yes	yes
copy06.cpp	no	no	yes	yes
copy07.cpp	no	yes	yes	yes
copy08.cpp	no	no	no	yes
copy09.cpp	no	no	yes	yes
copy10.cpp	no	no	yes	yes
copy11.cpp	no	no	no	yes
copy12.cpp	no	yes	yes	yes
copy13.cpp	no	yes	yes	yes
copy14.cpp	yes	yes	yes	yes
copy15.cpp	yes	yes	yes	yes

1. Chanchal K. Roy : Comparison and evaluation of code clone detection techniques and tools : A qualitative approach

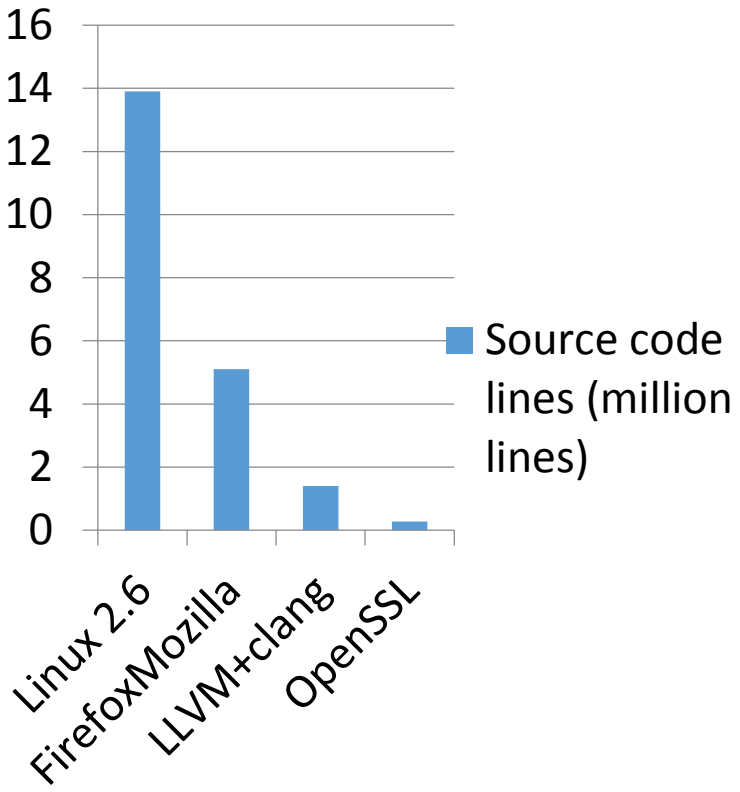


yes – test was detected as clone with original code.
no – test was not detected

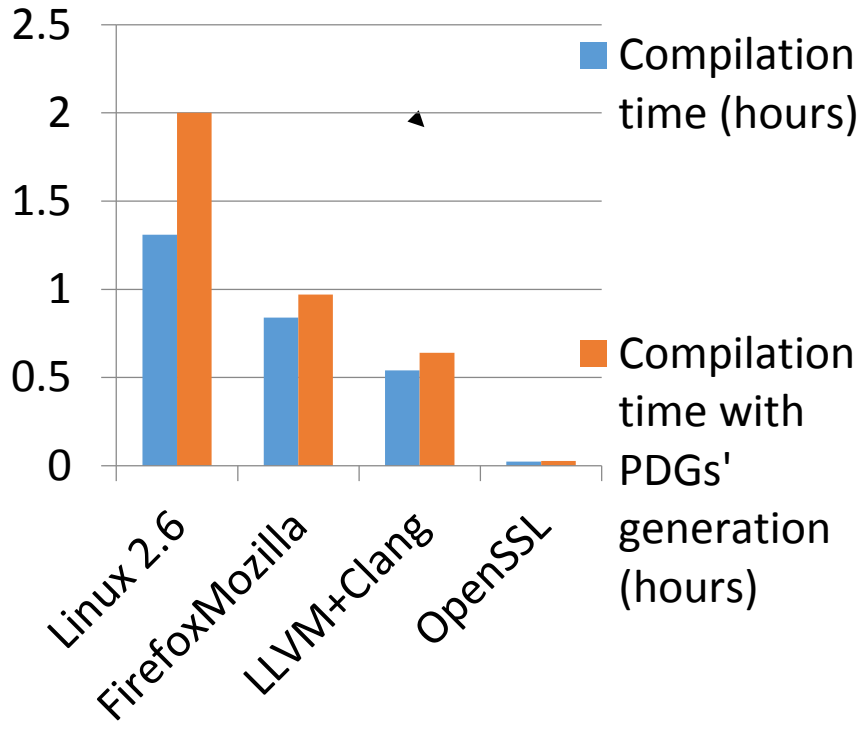
Results : PDGs' generation

Intel core i3, 8GB Ram.

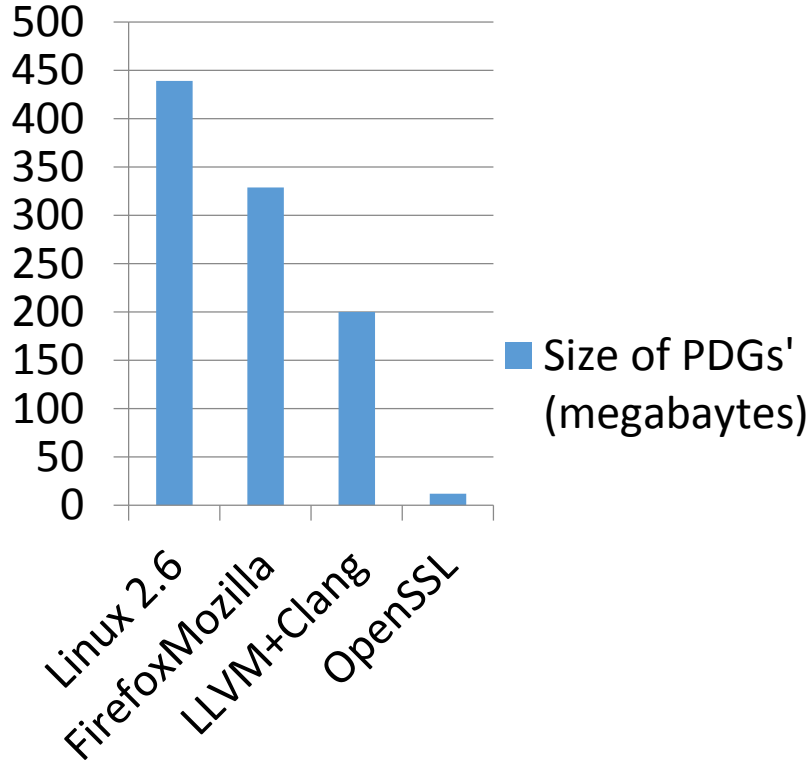
Source code lines



PDGs' generation time

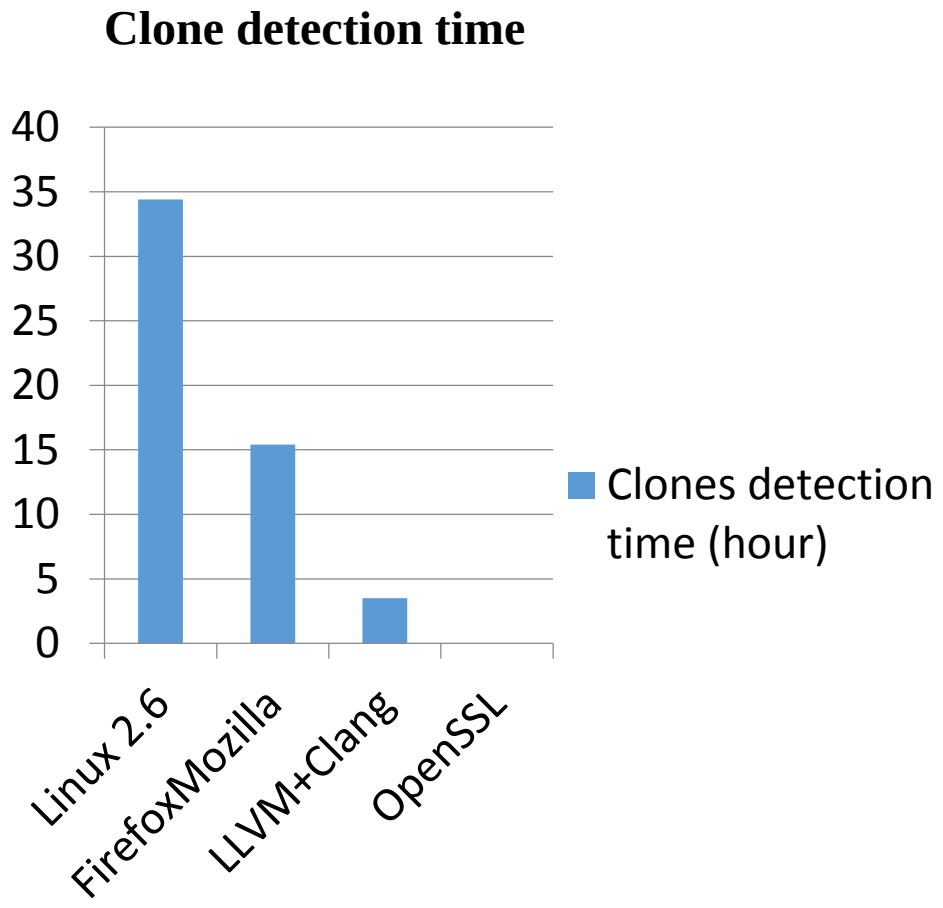


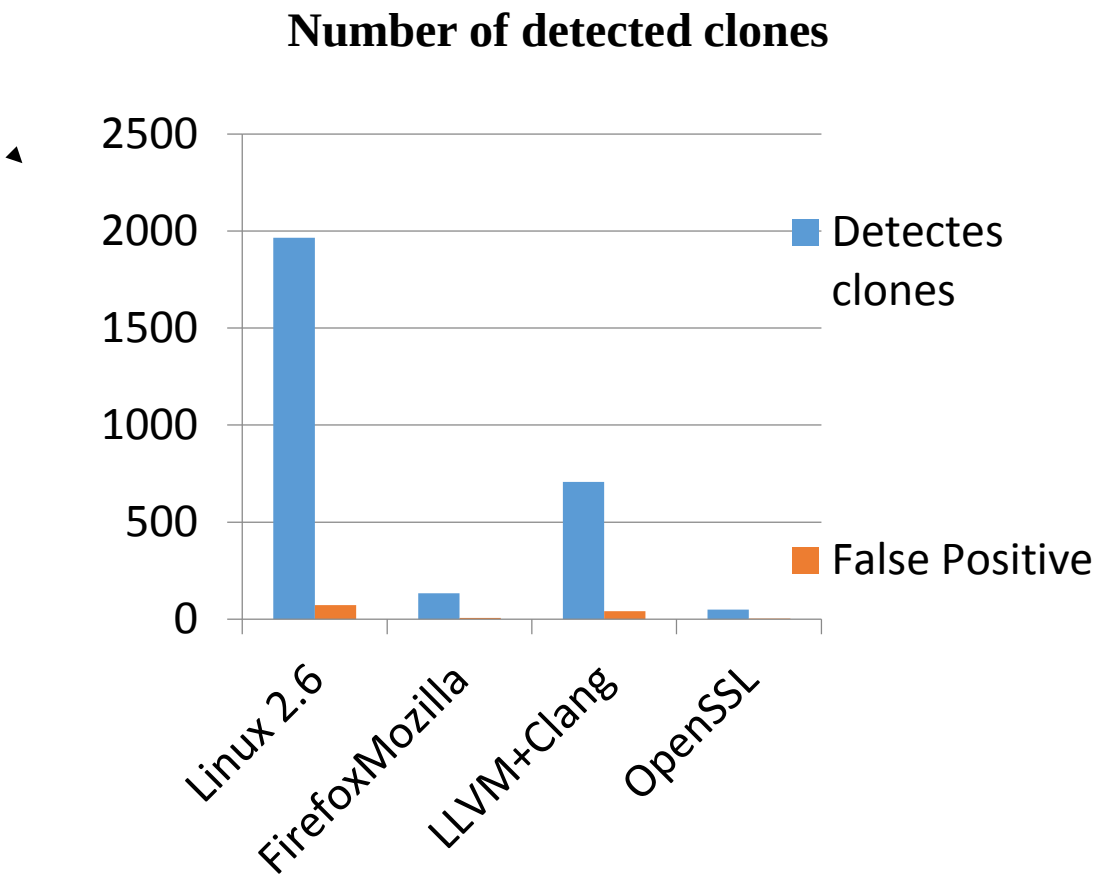
Size of dumped PDG



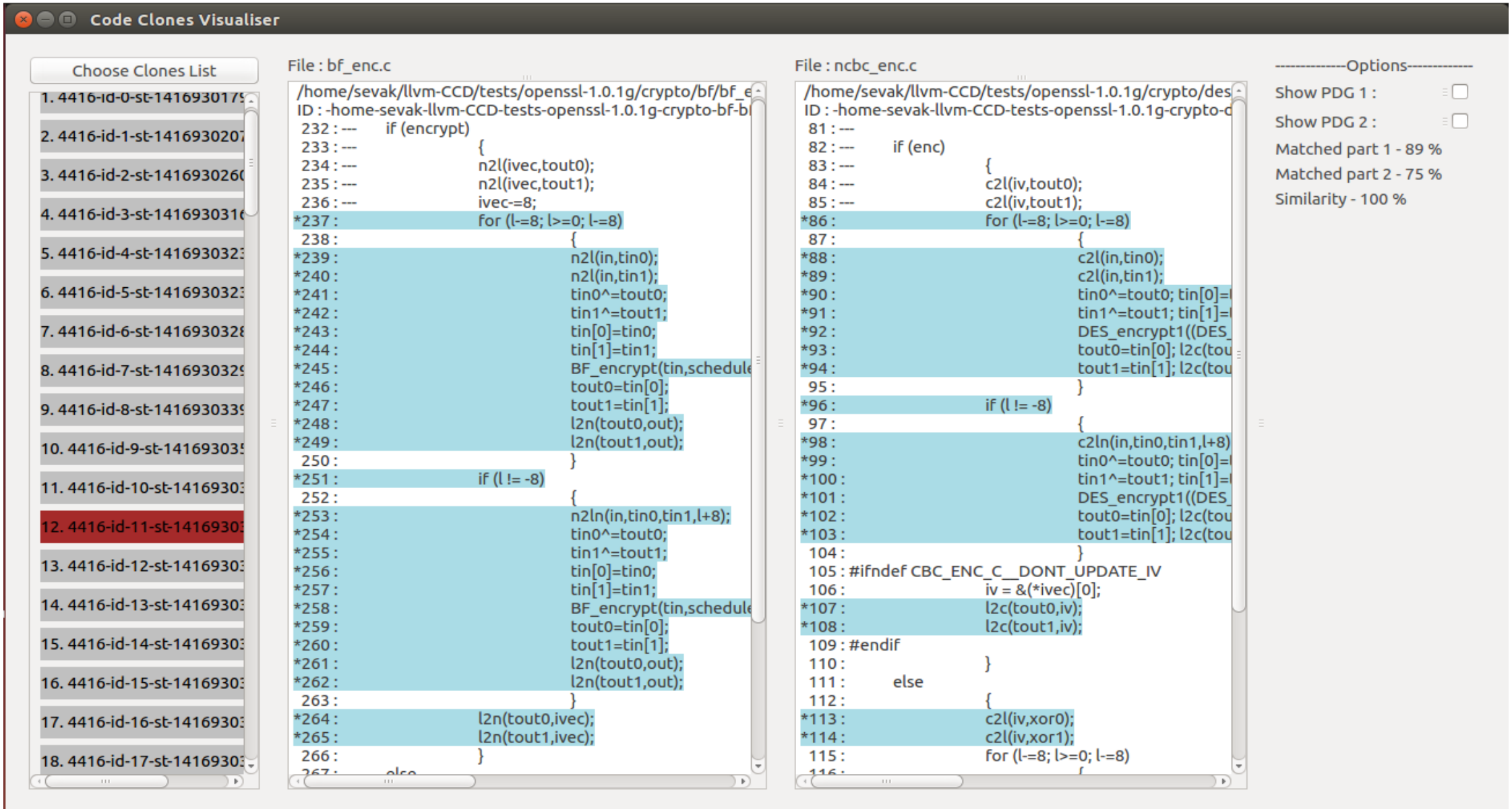
Results : clones detection

Similarity level higher 95%, minimal clone length 25.
Intel core i3, 8GB Ram.

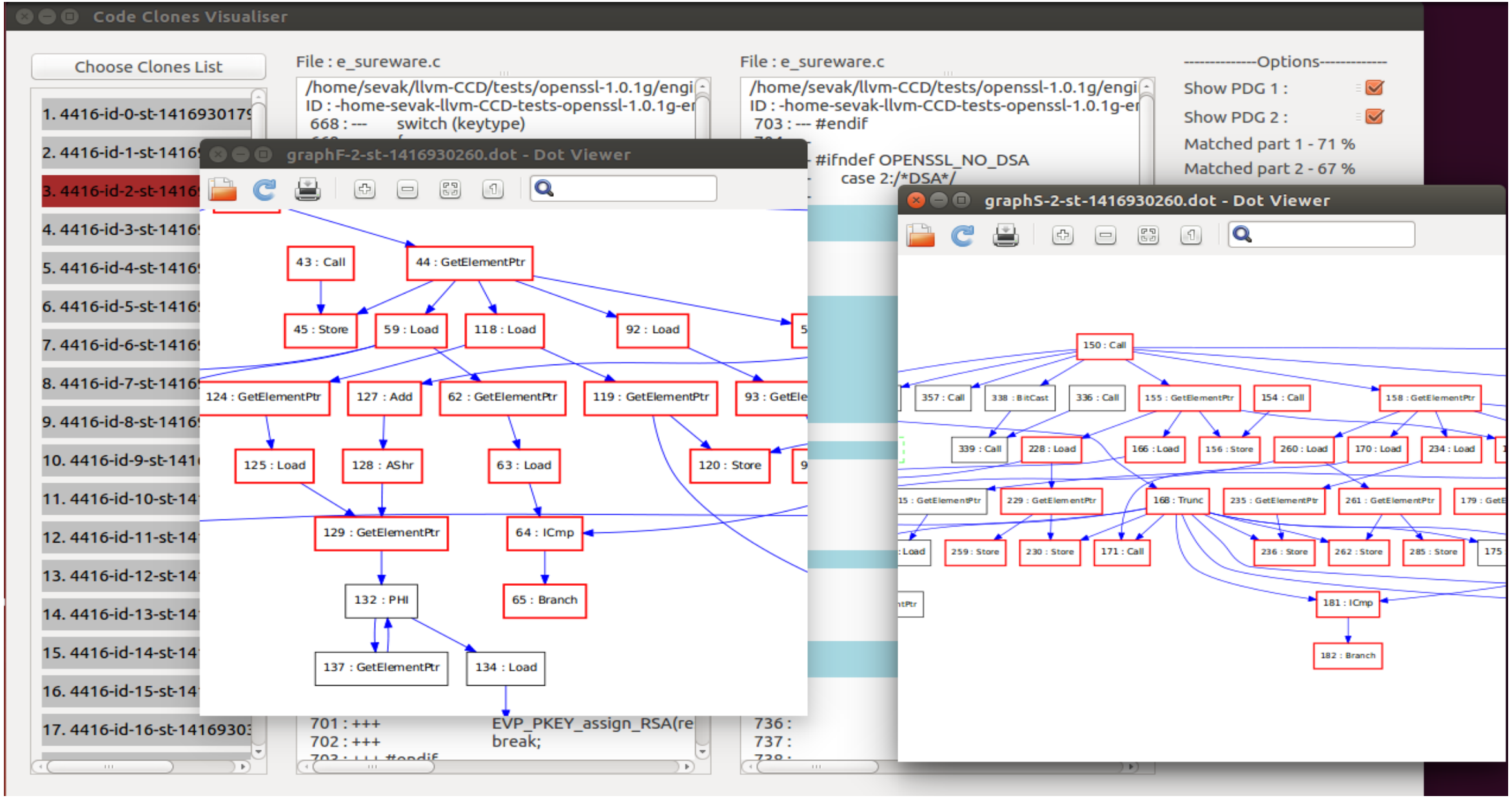




Results


Code Clones Visualiser
Choose Clones List
1. 4416-id-0-st-1416930179
2. 4416-id-1-st-141693020
3. 4416-id-2-st-1416930260
4. 4416-id-3-st-141693031
5. 4416-id-4-st-141693032
6. 4416-id-5-st-141693032
7. 4416-id-6-st-141693032
8. 4416-id-7-st-141693032
9. 4416-id-8-st-141693033
10. 4416-id-9-st-14169303
11. 4416-id-10-st-1416930
12. 4416-id-11-st-1416930
13. 4416-id-12-st-1416930
14. 4416-id-13-st-1416930
15. 4416-id-14-st-1416930
16. 4416-id-15-st-1416930
17. 4416-id-16-st-1416930
18. 4416-id-17-st-1416930
File : bf_enc.c
/home/sevak/llvm-CCD/tests/openssl-1.0.1g/crypto/bf/bf_e
ID : -home-sevak-llvm-CCD-tests-openssl-1.0.1g-crypto-bf-b
232 : --- if (encrypt)
233 : --- {
234 : --- n2l(ivec,tout0);
235 : --- n2l(ivec,tout1);
236 : --- ivec-=8;
*237 : for (l-=8; l>=0; l-=8)
238 : {
*239 : n2l(in,tin0);
*240 : n2l(in,tin1);
*241 : tin0^=tout0;
*242 : tin1^=tout1;
*243 : tin[0]=tin0;
*244 : tin[1]=tin1;
*245 : BF_encrypt(tin,schedule
*246 : tout0=tin[0];
*247 : tout1=tin[1];
*248 : l2n(tout0,out);
*249 : l2n(tout1,out);
250 : }
*251 : if (l != -8)
252 : {
*253 : n2ln(in,tin0,tin1,l+8);
*254 : tin0^=tout0;
*255 : tin1^=tout1;
*256 : tin[0]=tin0;
*257 : tin[1]=tin1;
*258 : BF_encrypt(tin,schedule
*259 : tout0=tin[0];
*260 : tout1=tin[1];
*261 : l2n(tout0,out);
*262 : l2n(tout1,out);
263 : }
*264 : l2n(tout0,ivec);
*265 : l2n(tout1,ivec);
266 : }
File : ncbc_enc.c
/home/sevak/llvm-CCD/tests/openssl-1.0.1g/crypto/des
ID : -home-sevak-llvm-CCD-tests-openssl-1.0.1g-crypto-c
81 : ---
82 : --- if (enc)
83 : --- {
84 : --- c2l(iv,tout0);
85 : --- c2l(iv,tout1);
*86 : for (l-=8; l>=0; l-=8)
87 : {
*88 : c2l(in,tin0);
*89 : c2l(in,tin1);
*90 : tin0^=tout0; tin[0]=
*91 : tin1^=tout1; tin[1]=
*92 : DES_encrypt1((DES_
*93 : tout0=tin[0]; l2c(tou
*94 : tout1=tin[1]; l2c(tou
95 : }
*96 : if (l != -8)
97 : {
*98 : c2ln(in,tin0,tin1,l+8)
*99 : tin0^=tout0; tin[0]=
*100 : tin1^=tout1; tin[1]=
*101 : DES_encrypt1((DES_
*102 : tout0=tin[0]; l2c(tou
*103 : tout1=tin[1]; l2c(tou
104 : }
105 : #ifndef CBC_ENC_C__DONT_UPDATE_IV
106 : iv = &(*ivec)[0];
*107 : l2c(tout0,iv);
*108 : l2c(tout1,iv);
109 : #endif
110 : }
111 : else
112 : {
*113 : c2l(iv,xor0);
*114 : c2l(iv,xor1);
115 : for (l-=8; l>=0; l-=8)
----------Options----------
Show PDG 1 :
Show PDG 2 :
Matched part 1 - 89 %
Matched part 2 - 75 %
Similarity - 100 %

Results


Code Clones Visualiser
Choose Clones List
1. 4416-id-0-st-1416930179
2. 4416-id-1-st-1416
3. 4416-id-2-st-1416
4. 4416-id-3-st-1416
5. 4416-id-4-st-1416
6. 4416-id-5-st-1416
7. 4416-id-6-st-1416
8. 4416-id-7-st-1416
9. 4416-id-8-st-1416
10. 4416-id-9-st-141
11. 4416-id-10-st-14
12. 4416-id-11-st-14
13. 4416-id-12-st-14
14. 4416-id-13-st-14
15. 4416-id-14-st-14
16. 4416-id-15-st-14
17. 4416-id-16-st-1416930
File : e_sureware.c
/home/sevak/llvm-CCD/tests/openssl-1.0.1g/engi
ID : -home-sevak-llvm-CCD-tests-openssl-1.0.1g-er
668 : --- switch (keytype)
701 : +++ EVP_PKEY_assign_RSA(re
702 : +++ break;
File : e_sureware.c
/home/sevak/llvm-CCD/tests/openssl-1.0.1g/engi
ID : -home-sevak-llvm-CCD-tests-openssl-1.0.1g-er
703 : --- #endif
#ifndef OPENSSL_NO_DSA
case 2:/*DSA*/
736 :
737 :
----------Options----------
Show PDG 1 :
Show PDG 2 :
Matched part 1 - 71 %
Matched part 2 - 67 %
graphF-2-st-1416930260.dot - Dot Viewer
43 : Call
44 : GetElementPtr
45 : Store
59 : Load
118 : Load
92 : Load
124 : GetElementPtr
127 : Add
62 : GetElementPtr
119 : GetElementPtr
93 : GetEle
125 : Load
128 : AShr
63 : Load
120 : Store
129 : GetElementPtr
64 : ICmp
132 : PHI
65 : Branch
137 : GetElementPtr
134 : Load
graphS-2-st-1416930260.dot - Dot Viewer
150 : Call
357 : Call
338 : BitCast
336 : Call
155 : GetElementPtr
154 : Call
158 : GetElementPtr
339 : Call
228 : Load
166 : Load
156 : Store
260 : Load
170 : Load
234 : Load
229 : GetElementPtr
168 : Trunc
235 : GetElementPtr
261 : GetElementPtr
259 : Store
230 : Store
171 : Call
236 : Store
262 : Store
285 : Store
181 : ICmp
182 : Branch

ISP RAS

Thank You.