On Approximate Matching of Programs for Protecting Libre Software

Arnoldo Müller Takeshi Shinohara

Department of Artificial Intelligence
Kyushu Institute of Technology (Iizuka, Japan)

CASCON Oct 19, 2006

Outline

Outline

Libre Software

Protect Libre Software

- Libre software = (Free software $\cup$ Open Source Software)
- Licensing Violations.
  - FSF and gpl-violations.
  - Using “strings” (binutils).
- Source code of:
  - Pirate program: not available.
  - Libre program: available.
- Objective: Binary program matching.
  - Different Compilers/Obfuscators/Strings.

Libre Software

Protect Libre Software

- Libre software = (Free software ∪ Open Source Software)
- Licensing Violations.
  - FSF and gpl-violations.
  - Using “strings” (binutils).
- Source code of:
  - Pirate program: not available.
  - Libre program: available.
- Objective: Binary program matching.
  - Different Compilers/Obfuscators/Strings.

Libre Software

Protect Libre Software

- Libre software = (Free software ∪ Open Source Software)
- Licensing Violations.
  - FSF and gpl-violations.
  - Using “strings” (binutils).
- Source code of:
  - Pirate program: not available.
  - Libre program: available.
- Objective: Binary program matching.
  - Different Compilers/Obfuscators/Strings.

Outline

State of the art

No reliable techniques available that fit our problem setting

- “Strings” (binutils).
- Birthmarks (easy to obfuscate) [Tamada, 2005].
- Source-level slicing [Komondoor 2001].
- General obfuscation is impossible [Vadhan, 2001].
- We base our work on Compiler Validation Transformation [Engelen, 2004].

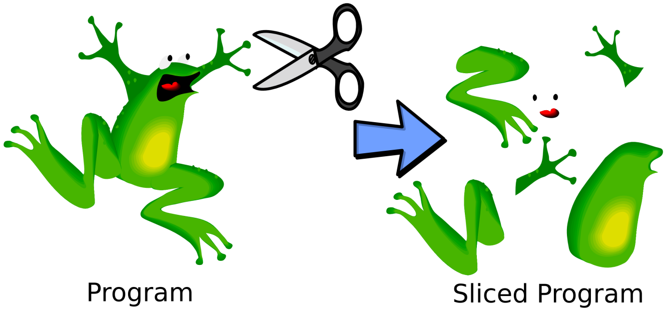
State of the art

No reliable techniques available that fit our problem setting

- “Strings” (binutils).
- Birthmarks (easy to obfuscate) [Tamada, 2005].
- Source-level slicing [Komondoor 2001].
- General obfuscation is impossible [Vadhan, 2001].
- We base our work on Compiler Validation Transformation [Engelen, 2004].

Matching of Programs

Achieved by slicing the programs and matching those slices

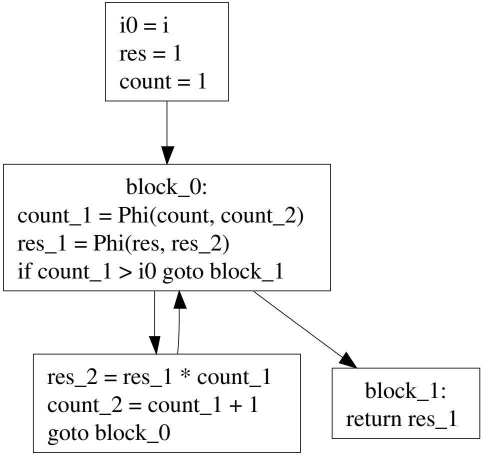


- Each slice is like a word of a document.

Outline

Slice Expansion

“Big” Slices can be used as program fingerprints.

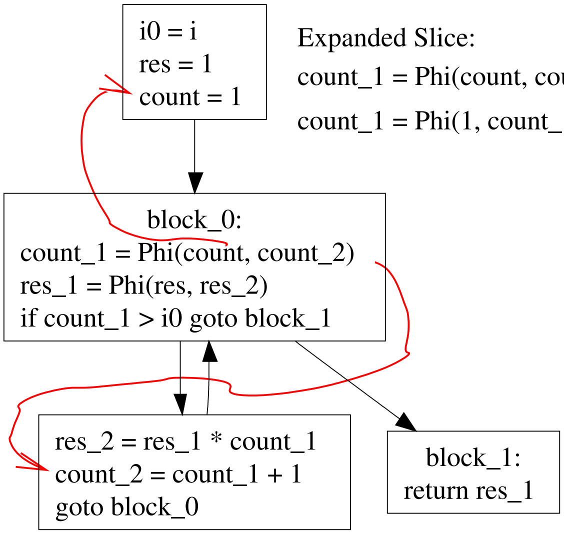


SSA representation

- SSA (Single Static Assignment).
- Slices: right hand side.
- References to other slices.
- Replace references with their contents.
    - Becomes bigger.
    - Useful fingerprint.

Slice Example

Replacements only performed once



Expanded Slice:

count_1 = Phi(count, count_2)

count_1 = Phi(1, count_1 + 1)

Equivalence of Slices

Similarity of slices can be calculated by “tree edit distance”

- Syntactical equivalence is not enough.
- Parts of the slices are ignored.
    - Strings, variable names.
- Distance Matching.
    - Between two slices A,B.
    - Integer indicates how different A,B are.
    - Tree edit distance.

Equivalence of Slices

Similarity of slices can be calculated by “tree edit distance”

- Syntactical equivalence is not enough.
- Parts of the slices are ignored.
    - Strings, variable names.
- Distance Matching.
    - Between two slices A,B.
    - Integer indicates how different A,B are.
    - Tree edit distance.

Outline

Furia

An approximate semantic matcher for Java

- Works on Java Bytecode.
- Verified our ideas. It works!
- But it is slow.
- License Violation:
  - Trovador 3000 lines.
  - Uses “jmusic”.
  - Was Obfuscated.
  - Matched against 369 Programs.

Matching: trovador (JDK 1.5 + ZKM Obfs.)

App Name	**Score**
jmusic	*0.202*
ChordAssist	0.189
pmd	0.077
skink	0.075
dynamicjava	0.064
catchxsl	0.059
j80	0.057
mockrunner	0.040

- With jikes 85%

Furia
An approximate semantic matcher for Java

- Works on Java Bytecode.
- Verified our ideas. It works!
- But it is slow.
- License Violation:
  - Trovador 3000 lines.
  - Uses “jmusic”.
  - Was Obfuscated.
  - Matched against 369 Programs.

Matching: trovador (JDK 1.5 + ZKM Obfs.)

App Name	**Score**
jmusic	*0.202*
ChordAssist	0.189
pmd	0.077
skink	0.075
dynamicjava	0.064
catchxsl	0.059
j80	0.057
mockrunner	0.040

- With jikes 85%

Furia
An approximate semantic matcher for Java

- Works on Java Bytecode.
- Verified our ideas. It works!
- But it is slow.
- License Violation:
    - Trovador 3000 lines.
    - Uses “jmusic”.
    - Was Obfuscated.
    - Matched against 369 Programs.

Matching: trovador (JDK 1.5 + ZKM Obfs.)

App Name	**Score**
jmusic	*0.202*
ChordAssist	0.189
pmd	0.077
skink	0.075
dynamicjava	0.064
catchxsl	0.059
j80	0.057
mockrunner	0.040

- With jikes 85%

Outline

What's next?

Create data for testing and improve speed. Learn slice changes

- Performance improvement.
- Database/corpora for measuring precision.
- Expression normalization.
- Learning techniques.
- Match/ranking refinements.
- Other architectures to byte-code.

Summary

Results seem promising. Further experimentation is required

- A very simple and new technique has been proposed.
    - Slice + Expansion + Distance.
- Speed issues must be solved.
- The technique works reasonably well.
    - Even with control flow obfuscation.

Summary

Results seem promising. Further experimentation is required

- A very simple and new technique has been proposed.
    - Slice + Expansion + Distance.
- Speed issues must be solved.
- The technique works reasonably well.
    - Even with control flow obfuscation.

Summary

Results seem promising. Further experimentation is required

- A very simple and new technique has been proposed.
    - Slice + Expansion + Distance.
- Speed issues must be solved.
- The technique works reasonably well.
    - Even with control flow obfuscation.

Questions / Comments

- Thank you!
- You can contact me at:
    - arnoldoMuller@gmail.com
    - arnoldo@daisy.ai.kyutech.ac.jp

Questions/Comments 1

Q: question A: answer C: comment

- Q: Can you use this for patent infringement detection?
    - A: No, this doesn't match an algorithm, but parts or snapshots of
    - methods.
- C: Pointer to a research released in Germany on source-level matching
    - A: Need to check it, thank u!

Questions/Comments 2

Q: question A: answer C: comment

- Q: What other transformations can an obfuscator do?
  - A: An obfuscator can add or remove or replace instructions.
  - Replacing can be undone by a term rewriting rule.
  - Removing instructions requires static analysis.
  - Add instructions (that modify slices) adds garbage into a phi
  - instruction.
    - Not a problem! if some subset slice matching function is defined
  - Using the best Obfuscator we could get.

Questions/Comments 3

Q: question A: answer C: comment

- C: Maybe you should not release this so the obfuscator developers will not try to attack your techniques
- Q: How will you enforce this?
  - A: This is not my job, this is the FSF's and gpl-violations group's
  - job.
- C: I have seen problems when enforcing these things. Linksys
- example. (Comment from a linux kernel developer)
  - You have to buy the router in order to complain

Questions/Comments 4

Q: question A: answer C: comment

- C: What you are trying to do is very hard (Formal specification
- expert)
  - Recommendation: Use clustering.

  fa
- Q: Formal methods won't help you.
- C: Another reference from a German researcher on source level matching
  - Not yet checked :)

Interesting Paper

Running programs on graphics cards

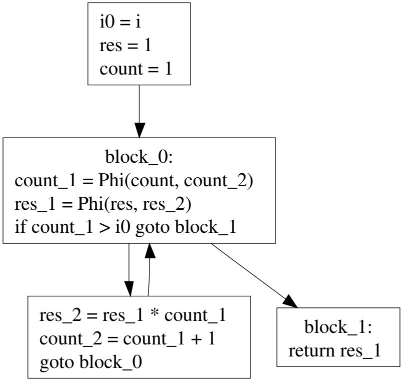
- Control flow graph is transformed
    - Simplified
    - To conform with gpu constrains
- It was a workshop so the paper is not published in the proceedings

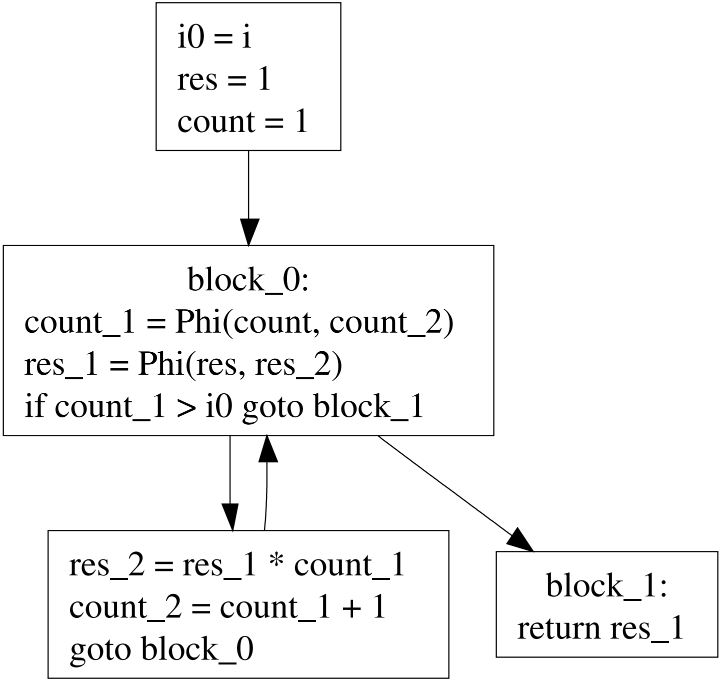
Outline

SSA (Single Static Assignment)

One assignment per variable. "Phi" is a selection function

```
f(int i){
 int res,count = 1;
 while(count <= i){
  res = res*count;
  count++;
 }
 return res;
}
```





```
i0 = i
res = 1
count = 1
```

```
block_0:
count_1 = Phi(count, count_2)
res_1 = Phi(res, res_2)
if count_1 > i0 goto block_1
```

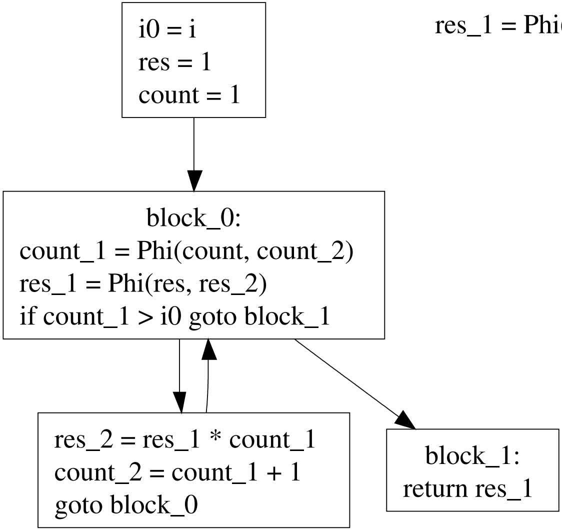
```
res_2 = res_1 * count_1
count_2 = count_1 + 1
goto block_0
```

```
block_1:
return res_1
```

Outline

Slice Expansion Example

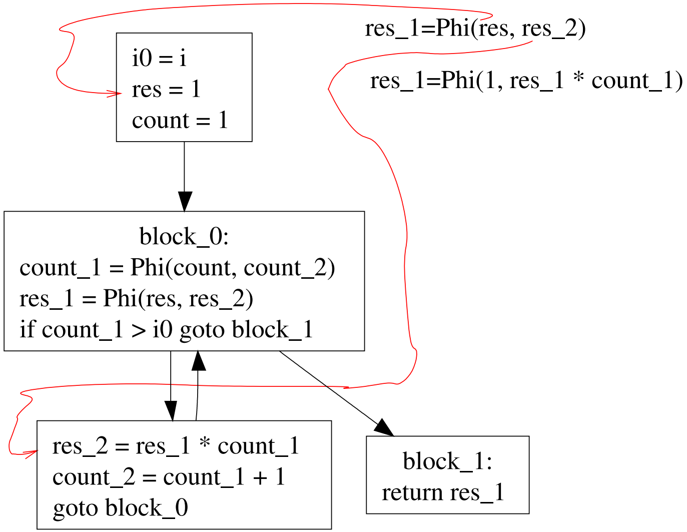
Slice res_1 to be expanded



res_1 = Phi(res, res_2)

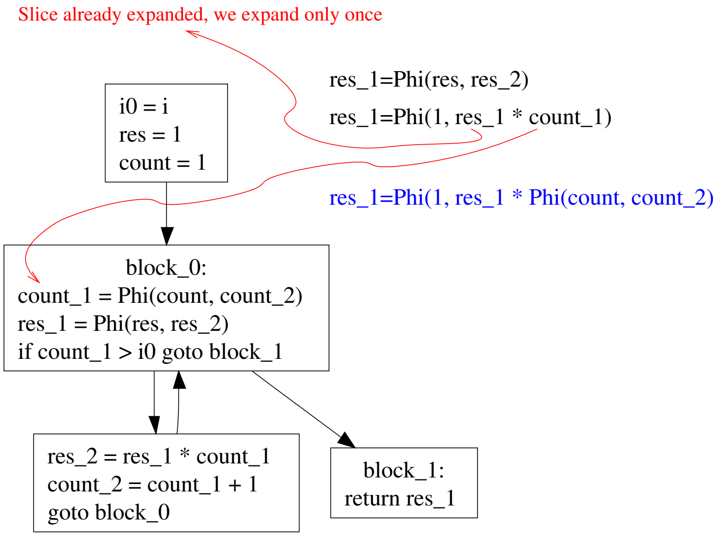
Slice Expansion Example

Slice res_1 to be expanded



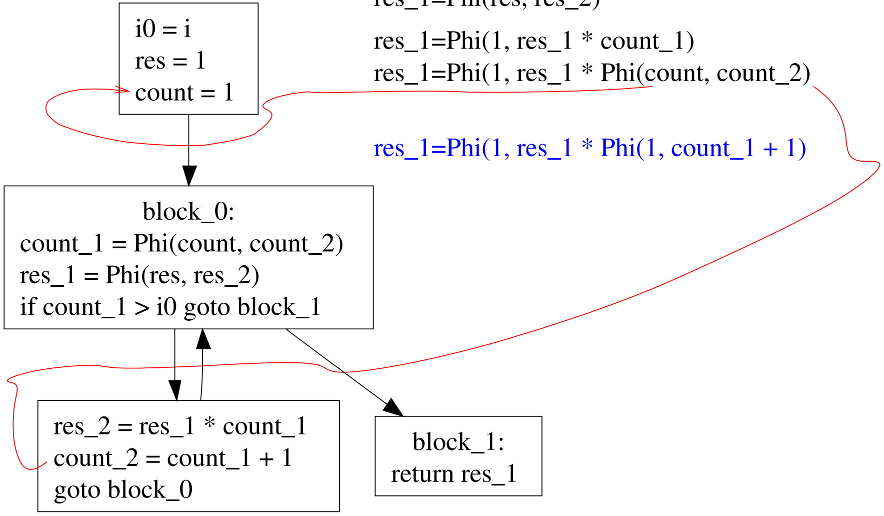
Slice Expansion Example

Slice res_1 to be expanded


Slice already expanded, we expand only once
res_1=Phi(res, res_2)
res_1=Phi(1, res_1 * count_1)
res_1=Phi(1, res_1 * Phi(count, count_2)
i0 = i
res = 1
count = 1
block_0:
count_1 = Phi(count, count_2)
res_1 = Phi(res, res_2)
if count_1 > i0 goto block_1
res_2 = res_1 * count_1
count_2 = count_1 + 1
goto block_0
block_1:
return res_1

Slice Expansion Example

Slice res_1 to be expanded



res_1=Phi(res, res_2)

res_1=Phi(1, res_1 * count_1)
res_1=Phi(1, res_1 * Phi(count, count_2)

res_1=Phi(1, res_1 * Phi(1, count_1 + 1)

Outline

toList Procedure

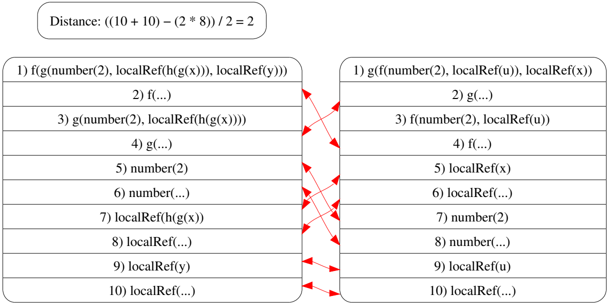
- all the subexpressions that can be created from a slice
- Adds a parameterless copy per each subexpression

```
toList(sum(localRef(3), num(2)) ) =
[sum(localRef(3), num(2)), sum(), localRef(3)
localRef(), num(2), num()]
```

dmatch Procedure

$$dmatch : E \times E \longrightarrow \mathbb{N}$$

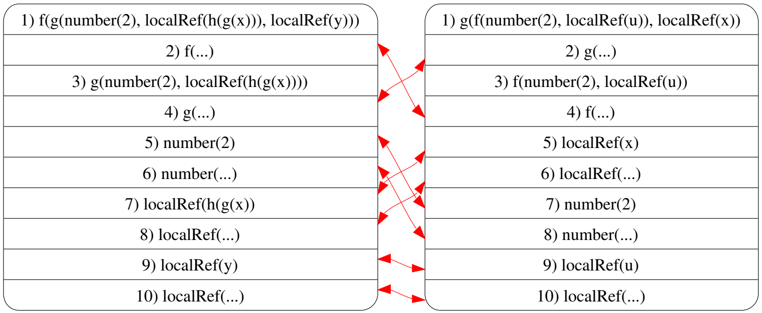
$$dmatch(e1, e2) = \frac{(slength(e1)+slength(e2))-(2*|toList(e1)\cap toList(e2)|)}{2}$$



Distance: ((10 + 10) – (2 * 8)) / 2 = 2

1) f(g(number(2), localRef(h(g(x))), localRef(y)))	1) g(f(number(2), localRef(u)), localRef(x))
2) f(...)	2) g(...)
3) g(number(2), localRef(h(g(x))))	3) f(number(2), localRef(u))
4) g(...)	4) f(...)
5) number(2)	5) localRef(x)
6) number(...)	6) localRef(...)
7) localRef(h(g(x))	7) number(2)
8) localRef(...)	8) number(...)
9) localRef(y)	9) localRef(u)
10) localRef(...)	10) localRef(...)

Distance: ((10 + 10) − (2 * 8)) / 2 = 2



Outline

dmatch Procedure

Effects of changing *ignore_slices_lower_than*

Database 18 Apps jdk 1.5

slice_cut_threshold=30
ignore_slices_lower_than=4
maximum_acceptable_distance=1
Matching: jfreechart (Jikes 1.22)

App Name	**Score**
jfreechart	*0.828*
freesudoku	0.227
htmlparser	0.188
jgnash	0.157
checkstyle	0.115
freemind	0.109
pdfbox	0.100
findbugs	0.084
triplea	0.079
jmusic	0.076
jasperreports	0.076
schemaspy	0.057
ireport	0.049
yale	0.033
azureus	0.028

slice_cut_threshold=30
ignore_slices_lower_than=15
maximum_acceptable_distance=1
Matching: jfreechart (Jikes 1.22)

App Name	**Score**
jfreechart	*0.739*
freesudoku	0.009
jgnash	0.008
jmusic	0.001
jasperreports	0.001
ireport	0.001
checkstyle	0.001
findbugs	0.001
yale	0.001
azureus	0.000

Matching control flow obfuscated Programs

Database 18 Apps jdk 1.5

slice_cut_threshold=30
ignore_slices_lower_than=15
maximum_acceptable_distance=1
Flow obfuscate String decryption: off
Matching: freemind
JDK 1.5 + ZKM (full)

App Name	**Score**
freemind	*0.518*
checkstyle	0.020
jgnash	0.012
jfreechart	0.006
ireport	0.004
triplea	0.004
htmlparser	0.002
jmusic	0.002
azureus	0.001
findbugs	0.001
pdfbox	0.001
yale	0.001
jacksum	0.000

slice_cut_threshold=30
ignore_slices_lower_than=15
maximum_acceptable_distance=1
Flow obfuscate String decryption: on
Matching: freemind
JDK 1.5 + ZKM (full)

App Name	**Score**
freemind	*0.122*
checkstyle	0.013
jgnash	0.012
ireport	0.006
htmlparser	0.003
pdfbox	0.003
findbugs	0.002
azureus	0.001
jasperreports	0.001
jmusic	0.001
yale	0.001
jacksum	0.000

Smoke Screen Obfuscator

Database of 18 Apps jdk 1.5

slice_cut_threshold=30
ignore_slices_lower_than=15
maximum_acceptable_distance=3
Matching: jacksum
JDK 1.5 + smoke screen (full options)

App Name	**Score**
jacksum	*0.804*
azureus	0.086
checkstyle	0.017
jgnash	0.012
jasperreports	0.011
findbugs	0.009
htmlparser	0.007
ireport	0.006
pdfbox	0.006
triplea	0.005
yale	0.004
jfreechart	0.003
schemaspy	0.003
jmemorize	0.003
smallexample	0.003
jmusic	0.002
freemind	0.001
freesudoku	0.000

Matching freesudoku and jmusic

Database of 269 App (Different compilers)

slice_cut_threshold=30
ignore_slices_lower_than=15
maximum_acceptable_distance=1
Matching: freesudoku (JDK 1.5)
		slice_cut_threshold=30
		ignore_slices_lower_than=15
		maximum_acceptable_distance=1
		Matching: jmusic (JDK 1.5 + ZKM (full))	

App Name	**Score**	**App Name**	**Score**
freesudoku	*0.900*	*jmusic*	*0.085*
JAMonAll_020106	0.040	jquery-2006-Jan-07-dist	0.030
nachocalendar-0.23	0.015	jreversepro-1.4.1-bin	0.028
jwebunit-1.2	0.013	coinjema-0.4	0.025
jin-2.13.1-unix	0.009	mobup_client_0.3.2	0.015
ejb3unit-1.0-alpha2	0.009	iHTbot-0.5.1b2	0.012
siscweb-bin-0.32	0.009	jmsn-0.9.9b2	0.011
matharcade-1.2	0.007	fitdecorator-beta0.2	0.009
HTCommunicator_0.1	0.005	jopt_csp_1-0	0.008
transform-2.1	0.005	etl-1.0-full	0.008
polliwog-bin-stable-0.5	0.001	regexSearch-1_2	0.007
esper-0.7.0	0.001	jwp_v1.0_beta4_bin	0.007
Furthur175	0.001	cap4j-0.1.2-beta	0.005
cayenne-1.2M10	0.000	freemind	0.005

Effects of changing *maximum_acceptable_distance*

Database of 363 App (Various Compilers)

slice_cut_threshold=30		slice_cut_threshold=30	
ignore_slices_lower_than=15		ignore_slices_lower_than=15	
maximum_acceptable_distance=1		maximum_acceptable_distance=3	
Matching: freesudoku (JDK 1.5 ZKM full)		Matching: freesudoku (JDK 1.5 ZKM full)	
App Name	**Score**	**App Name**	**Score**
freesudoku	0.108	freesudoku	**0.31**
DocSearcher-3.88	0.018	DocSearcher-3.88	0.020
jnetstream	0.018	jnetstream	0.020
BlinkenApplet0.7	0.017	jgames-0.9.2	0.020
jgames-0.9.2	0.015	ocl4javaLib_2.1.7	0.010