

A Criterion for Filtering Code Clone Related Bugs

Yasuhiro Hayase[†], Yii Yong Lee, Katsuro Inoue[†]
Osaka University

[†]{y-hayase,inoue}@ist.osaka-u.ac.jp

Background

- Code Clone

- ▶ a code fragment occurring more than once in identical or similar form into a software system
- ▶ introduced in the source program because of various reasons such as reusing code by 'copy-and-paste'

- Clone Pair

a pair of code fragment that are identical or similar each other

Clone-Related Bugs

- Clone-Related Bug

- ▶ Clones are often modified after copy-and-paste
- ▶ Faults are possibly introduced through the modification.

- CP-Miner (Zhenmen Li *et al.*, 2007)

- ▶ Detecting clone-related bugs

- Find and present inconsistent renaming of identifier between clones

- ▶ Problem

- False-positives

➔ Propose a new criterion to filter out the false-positives

Brief Summary of CP-Miner

- Map identifier appearances between a clone pair
- Create renaming table between a clone pair
- Compute **UnchangedRatio (UR)**
 - ▶ Smaller UR except for 0 means that the renaming is suspicious.

Clone C1	Clone C2
<pre>... = b; b++; for (p=0; p<10; p++) { x += p; }</pre>	<pre>... = c; c++; for (q=0; q<10; q++) { y += p; }</pre>



ID in C1	ID in C2	count	UR
a	a	2	1
b	c	2	0
p	p	1	0.25
	q	3	
x	x	1	0.20
	y	2	
	z	2	

Leave unchanged

Our Approach: New Filtering Criterion

- **x** seems intentionally renamed to different symbols
- Criterion **Conflict (CF)**
 - ▶ true if the identifier mapped into two or more other identifiers
 - ▶ false otherwise

ID in C1	ID in C2	count	UR	CF
a	a	2	1	false
b	c	2	0	false
p	p	1	0.25	false
	q	3		
x	x	1	0.20	true
	y	2		
	z	2		

Implementation

- The filter using **Conflict** is implemented into clone-related bug detection system.

- ▶ The system uses CCFinder

The screenshot displays the CCFinder tool interface. On the left, the 'Candidate List' table shows various file entries with their IDs and unchanged ratios. On the right, the 'Source Code' window shows the implementation of the filter in kernel/prtrace.c, with specific code blocks highlighted in blue.

Base File Name	Rel File Name	ID	UnchangedRatio
linux-2.6.6/arch...	linux-2.6.6/arch...	current	0.16666667
linux-2.6.6/arch...	linux-2.6.6/arch...	err	0.16666667
linux-2.6.6/arch...	linux-2.6.6/arch...	eflags	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	current	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	i	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	i	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	err	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	err	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	err	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	err	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	err	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	err	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	err	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	prom_phys_total	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	prom_phys_total	0.2
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25
linux-2.6.6/arch...	linux-2.6.6/arch...	IEEE754_CLASS_I...	0.25

```
534         audit_syscall_exit(current, reg)
535     }
536 }
537
538     if ((test_thread_flag(TIF_SYSCALL_TRACE))
539         return;
540     if (!(current->ptrace & PT_PTRACED))
541         return;
542     /* the 0x80 provides a way for the tracing parent to distinguish
543        between a syscall stop and SIGTRAP delivery */
544     ptrace_notify(SIGTRAP | ((current->ptrace & PT_TRACESYSGID)
545                            ? 0x80 : 0));
546
547     /*
548     * this isn't the same as continuing with a signal, but it will do
549     * for normal use.  strace only continues with a signal if the
550     * stopping signal is not SIGTRAP.  -brl
551     */
552     if (current->exit_code) {
553         send_sig(current->exit_code, current, 1);
554         current->exit_code = 0;
555     }
556 }
```

Base File ID	Rel File ID	Frequency	UnchangedRatio
current	current	1	0.167
current	tsk	5	0.167

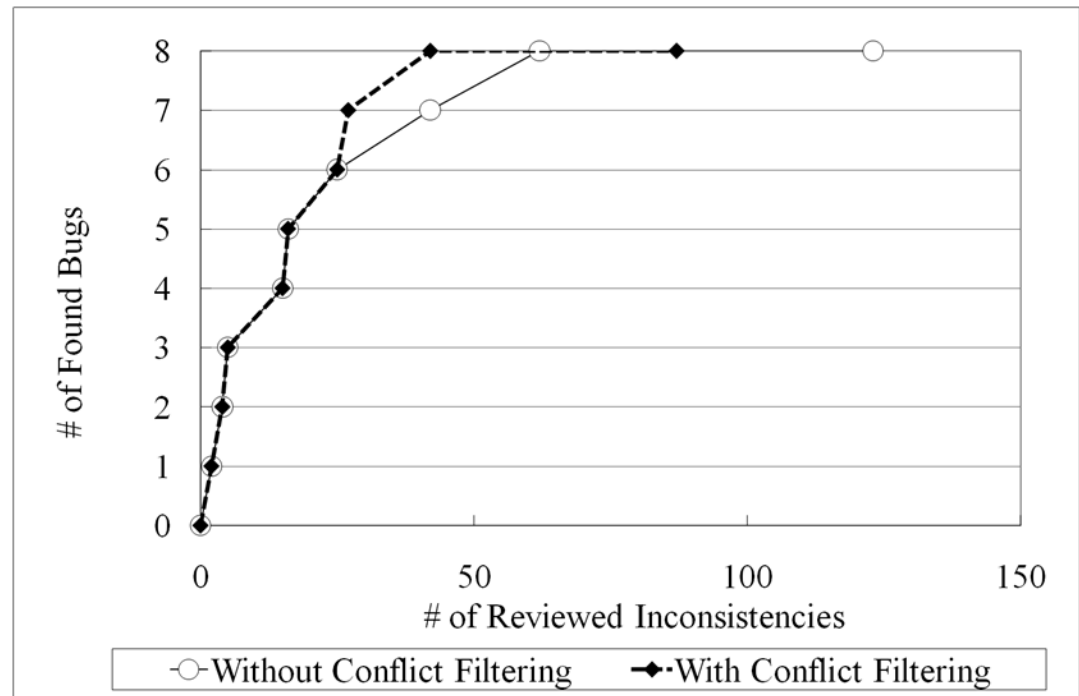
```
304 }
305
306 asmlinkage void do_syscall_trace(void)
307 {
308     struct task_struct *tsk = current;
309
310     if ((test_thread_flag(TIF_SYSCALL_TRACE))
311         return;
312     if (!(tsk->ptrace & PT_PTRACED))
313         return;
314     /* the 0x80 provides a way for the tracing parent to distinguish
315        between a syscall stop and SIGTRAP delivery */
316     ptrace_notify(SIGTRAP | ((current->ptrace & PT_TRACESYSGID)
317                            ? 0x80 : 0));
318
319     /*
320     * this isn't the same as continuing with a signal, but it will do
321     * for normal use.  strace only continues with a signal if the
322     * stopping signal is not SIGTRAP.  -brl
323     */
324     if (tsk->exit_code) {
325         send_sig(tsk->exit_code, tsk, 1);
326         tsk->exit_code = 0;
327     }
328 }
329 }
```

Evaluation

- An experiment was performed for evaluation
 - ▶ Target: arch module of Linux 2.6.6
 - ▶ Inconsistencies whose $\text{UnchangedRatio} \leq 0.4$ are reviewed in ascent order of UnchangedRatio

- Result

- ▶ Reduce 27% of inconsistencies
- ▶ Filter removes NO true-faults



Conclusion

- We proposed a criterion for filtering false-candidates detected by CP-Miner
 - ▶ The criterion recognizes the identifiers renamed to two or more names different from the original name
- The filter using the criterion is implemented into a clone-related bug detection system
- The filter reduces 27% of inconsistencies