Modeling Time in Java Programs for Automatic Error Detection

Giovanni Liva, Muhammad Taimoor Khan,
Francesco Spegni, Luca Spalazzi, Andreas Bollin,
Martin Pinzger

Problem

Automatic verification of software could save companies from huge losses. There are many examples where a single bug has cost companies millions of dollars although, many of these bugs could have been prevented [1].

[1] Robert N Charette. 2005. Why software fails. IEEE spectrum 42, 9 (2005), 1–36.

Solution

Time Domain

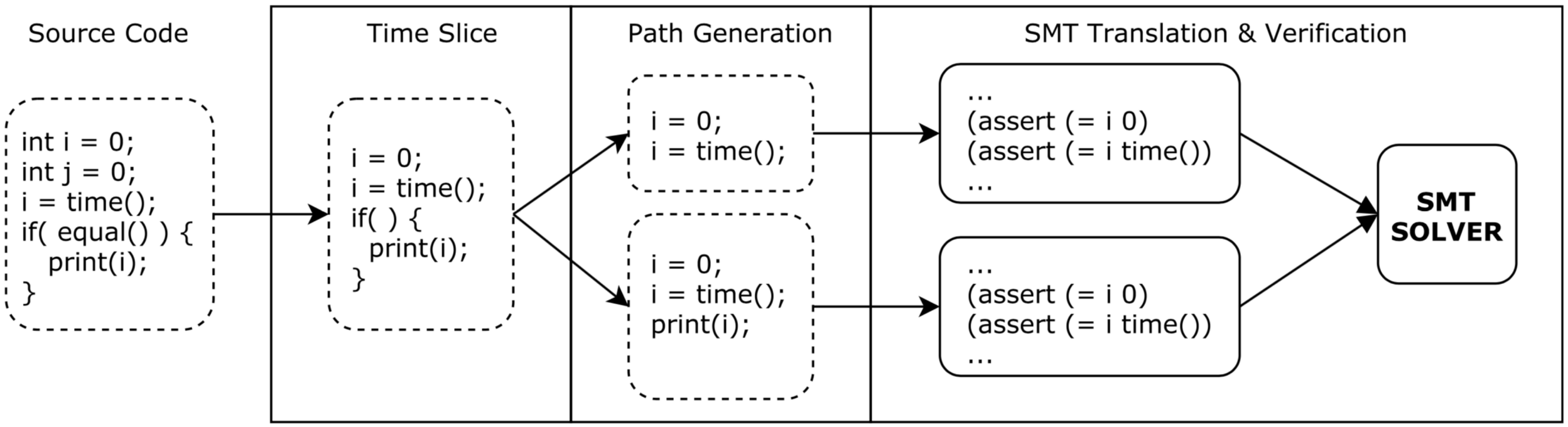
```
long t0 = System.nanoTime();
long t1 = System.nanoTime();
```

1. `t1 – t0 < 0`
2. `t1 < t0`

Time Domain

```
1  public void poll(long timeout) {
2    long now = time.milliseconds();
3    long deadline = now + timeout;
4    while (now <= deadline) {
5      if (coordinatorUnknown()) {
6        ensureCoordinatorReady();
7        now = time.milliseconds();
8      }
9      if (needRejoin()) {
10       ensureActiveGroup();
11       now = time.milliseconds();
12     }
13     pollHeartbeat(now);
14     long remaining = Math.max(0,
15         deadline - now);
16     client.poll(Math.min(remaining,
           timeToNextHeartbeat(now)));
17     now = time.milliseconds();
18   }
19 }
```

Approach


Source Code
int i = 0;
int j = 0;
i = time();
if(equal()) {
print(i);
}
Time Slice
i = 0;
i = time();
if( ) {
print(i);
}
Path Generation
i = 0;
i = time();
i = 0;
i = time();
print(i);
SMT Translation & Verification
...
(assert (= i 0)
(assert (= i time())
...
...
(assert (= i 0)
(assert (= i time())
...
SMT
SOLVER

Approach :: Time Slice

```
1  now = time.milliseconds();
2  deadline = now + timeout;
3  while (now <= deadline) {
4    if () {
5      now = time.milliseconds();
6    }
7    if () {
8      now = time.milliseconds();
9    }
10   pollHeartbeat(now);
11   remaining = Math.max(0, deadline - now);
12   client.poll(Math.min(remaining,
         timeToNextHeartbeat(now)));
13   now = time.milliseconds();
14 }
```

Approach :: Path Generation

```
now = time.milliseconds();                    ①
deadline = now + timeout;
while (now <= deadline) {
 if () {
  now = time.milliseconds();                  ②
 }
 if () {
  now = time.milliseconds();                  ③
 }
 pollHeartbeat(now);
 remaining = Math.max(0, deadline-now);
 client.poll(Math.min(remaining,             ④
      timeToNextHeartbeat(now)));
 now = time.milliseconds();
}
```

Original

```
now = time.milliseconds();                    ①
deadline = now + timeout;
while (now <= deadline) {
 pollHeartbeat(now);                          ④
 ...
}
```

Path a

```
now = time.milliseconds();                    ①
deadline = now + timeout;
while (now <= deadline) {
 now = time.milliseconds();                   ②
 pollHeartbeat(now);                          ④
 ...
}
```

Path b

```
now = time.milliseconds();                    ①
deadline = now + timeout;
while (now <= deadline) {
 now = time.milliseconds();                   ③
 pollHeartbeat(now);                          ④
 ...
}
```

Path c

```
now = time.milliseconds();                    ①
deadline = now + timeout;
while (now <= deadline) {
 now = time.milliseconds();                   ②
 now = time.milliseconds();                   ③
 pollHeartbeat(now);                          ④
 ...
}
```

Path d

Approach :: SMT Translation

- Z3-Opt – Linear Optimization over SMT Formulas
- SMT-Lib 2.0 Model constructed iteratively
- Check values only when a variable is read

```
deadline = now + timeout;
```

```
(declare-const deadline Int)
(assert (= deadline (+ now timeout)))
(assert (<= deadline over_max_val))
```

```
now = time.milliseconds();
```

```
(declare-fun milliseconds () Int)
(assert (and (>= milliseconds 0) (<= milliseconds max_val)))
(assert (= now milliseconds))
```

Evaluation

- **RQ1**: What is the precision of our approach in detecting time related errors in source code?

- **RQ2** : What is the run time that our approach requires for producing the results?

- 20 Open Source Java Projects:
  - 125,130 Classes
  - 939,861 Methods

https://github.com/rtse-project/automatic-error-detection

Evaluation :: RQ1

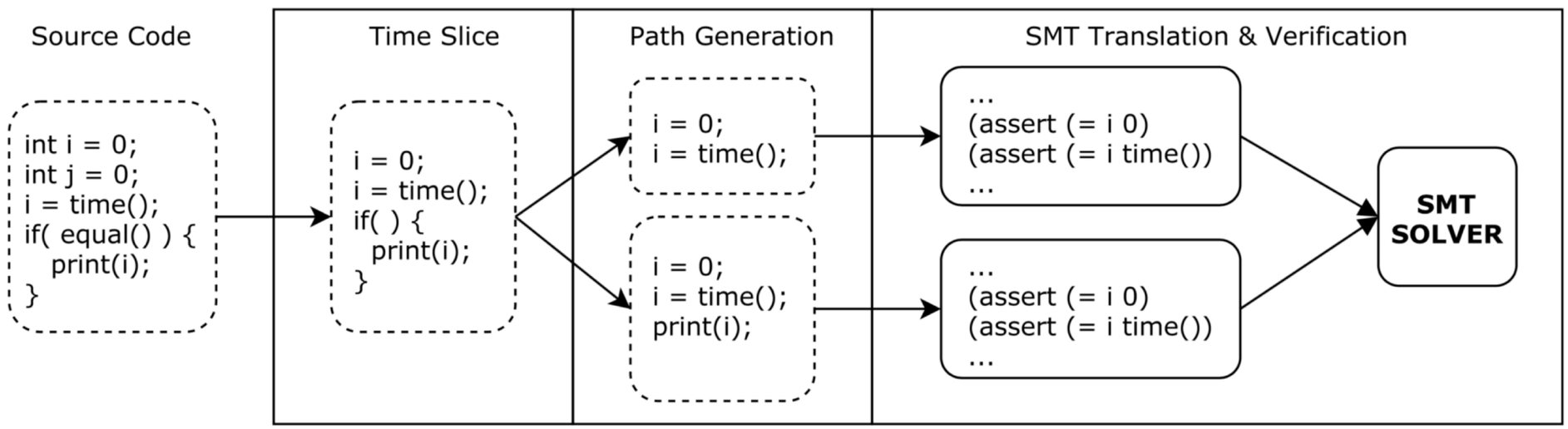
Name	%SLOC	#Detected	#TP	#FP	Precision
activemq	9.24%	16	13	3	81.25%
atmosphere	10.81%	1	1	0	
aws-sdk-java	124.07%	1	1		
camel	8.26%	4			
flume	11.39%				
hado					82.35%
				2	91.67%
			13	1	92.86%
	.9%	13	13	0	100.00%
neo4j	5.20%	5	4	1	80.00%
sling	13.40%	21	20	1	95.24%
…	…	…	…	…	…
Overall	**28.30%**	**146**	**134**	**12**	**91.78%**

Setting a negative initialDelayTime is an error so an exception should be thrown indicating this so the value can be fixed

Evaluation :: RQ2

Name	Parsing [s]	Detectig [s]	ms/method
activemq	226.5	17.3	5.92
atmosphere	22.3	2.3	6.06
aws-sdk-java	4373.5	141.9	22.00
camel	1333.1	28.2	4.84
flume	20.1	2.7	3.44
hadoop	542.3	56.4	6.03
hazelcast	266.3	15.0	4.82
hbase	923.6	90.4	7.98
jetty	158.7	10.2	6.78
kafka	55.0	7.1	4.55
neo4j	247.6	16.8	4.38
sling	268.1	14.2	7.64
…	…	…	…
Overall	**9,355.6 s**	**485.8 s**	**7.02 ms**

Summary



Evaluation :: RQ1

Name	%SLOC	#Detected	#TP	#FP	Precision
activemq	9.24%	16	13	3	81.25%
atmosphere	10.81%	1	1	0	100.00%
aws-sdk-java	124.07%	1	1	0	100.00%
camel	8.26%	4	4	0	100.00%
flume	11.39%	3	3	0	100.00%
hadoop	9.56%	27	26	1	96.30%
hazelcast	5.64%	17	14	3	82.35%
hbase	145.92%	24	22	2	91.67%
jetty	11.03%	14	13	1	92.86%
kafka	9.10%	13	13	0	100.00%
neo4j	5.20%	5	4	1	80.00%
sling	13.40%	21	20	1	95.24%
…	…	…	…	…	…
Overall	**28.30%**	**146**	**134**	**12**	**91.78%**

Evaluation :: RQ2

Name	Parsing [s]	Detectig [s]	ms/method
activemq	226.5	17.3	5.92
atmosphere	22.3	2.3	6.06
aws-sdk-java	4373.5	141.9	22.00
camel	1333.1	28.2	4.84
flume	20.1	2.7	3.44
hadoop	542.3	56.4	6.03
hazelcast	266.3	15.0	4.82
hbase	923.6	90.4	7.98
jetty	158.7	10.2	6.78
kafka	55.0	7.1	4.55
neo4j	247.6	16.8	4.38
sling	268.1	14.2	7.64
…	…	…	…
Overall	**9,355.6 s**	**485.8 s**	**7.02 ms**

Modeling Time in Java Programs for Automatic Error Detection

Giovanni Liva, Muhammad Taimoor Khan,
Francesco Spegni, Luca Spalazzi, Andreas Bollin,
Martin Pinzger

Formal Time Semantics

- Return Time: `long now = System.currentTimeMillis();`

$$\frac{System.currentTimeMillis() \in M_r^t}{\langle (V^t, M_r^t, \dots), now = System.currentTimeMillis() \rangle \vdash (\{now\} \cup V^t, \dots)}$$

- Explicit Timeout: `th.join(now);`

$$\frac{th \in java.lang.Thread \wedge Thread.join(long) \in M_t^t}{\langle (V^t, \_, M_t^t, \_), th.join(now) \rangle \vdash (\{now\} \cup V^t, \_, M_t^t, \_)}$$

Name	#Methods	#T. Methods	#Paths	SLOC	% SLOC	#Detected	#TP	#FP	Precision
activemq	41,212	12,583 (30.5%)	16,447	38,430	9.24%	16	13	3	81.250%
Activiti	15,358	6,034 (39.3%)	7,885	15,798	11.31%	0	0	0	-
airavata	70,843	39,858 (56.3%)	70,015	646,626	90.87%	0	0	0	-
alluxio	24,859	13,570 (54.6%)	19,706	84,268	36.03%	0	0	0	-
atmosphere	4,043	1,626 (40.2%)	2,237	3,875	10.81%	1	1	0	100.000%
aws-sdk-java	205,202	150,932 (73.6%)	247,855	2,227,270	124.07%	1	1	0	100.000%
beam	20,477	7,832 (38.2%)	9,489	8,125	3.85%	0	0	0	-
camel	114,938	34,760 (30.2%)	44,960	87,985	8.26%	4	4	0	100.000%
elastic-job	2,493	637 (25.6%)	783	559	2.12%	0	0	0	-
flume	6,627	2,429 (36.7%)	3,614	9,771	11.39%	3	3	0	100.000%
hadoop	99,343	40,173 (40.4%)	54,819	121,523	9.56%	27	26	1	96.296%
hazelcast	58,405	20,741 (35.5%)	25,488	36,626	5.64%	17	14	3	82.353%
hbase	127,061	81,747 (64.3%)	111,069	175,268	145.92%	24	22	2	91.667%
jetty.project	24,907	8,057 (32.3%)	12,779	37,794	11.03%	14	13	1	92.857%
kafka	13,669	5,158 (37.7%)	7,196	13,616	9.10%	13	13	0	100.000%
lens	8,063	3,917 (48.6%)	5,265	10,450	10.50%	0	0	0	-
nanohttpd	710	205 (28.9%)	294	659	8.75%	0	0	0	-
neo4j	60,378	18,595 (30.8%)	24,435	35,425	5.20%	5	4	1	80.000%
sling	36,969	15,489 (41.9%)	23,111	57,342	13.40%	21	20	1	95.238%
twitter4j	4,304	1,875 (43.6%)	2,561	12,663	39.01%	0	0	0	-
Overall	939,861	466,218 (49.6%)	690,008	3,624,073	28.30%	146	134	12	**91.781%**

```
(declare-const max_val () Int)
(declare-const over_max_val () Int)
(declare-const min_val () Int)
(declare-fun milliseconds () Int)
(assert (= max_val 92233720368547758O7))
(assert (= over_max_val 92233720368547758O8))
(assert (= min_val (- 92233720368547758O8)))
(assert (and (>= milliseconds 0) (<= milliseconds max_val)))
(declare-const timeout Int)
(assert (<= min_val timeout))
(assert (>= over_max_val timeout))
(declare-const now Int)
(assert (= now milliseconds))
(assert (<= now over_max_val))
(assert (>= now min_val))
(declare-const deadline Int)
(assert (= deadline (+ now timeout)))
(assert (<= deadline over_max_val))
(assert (>= deadline min_val))
(push)
(maximize now)
(check-sat)
(pop)
(push)
(minimize now)
(check-sat)
(pop)
(push)
(maximize deadline)
(check-sat)
(pop)
(push)
(minimize deadline)
(check-sat)
(pop)
...
```