VarexJ: A Variability-Aware Java Interpreter

Testing Configurable Systems

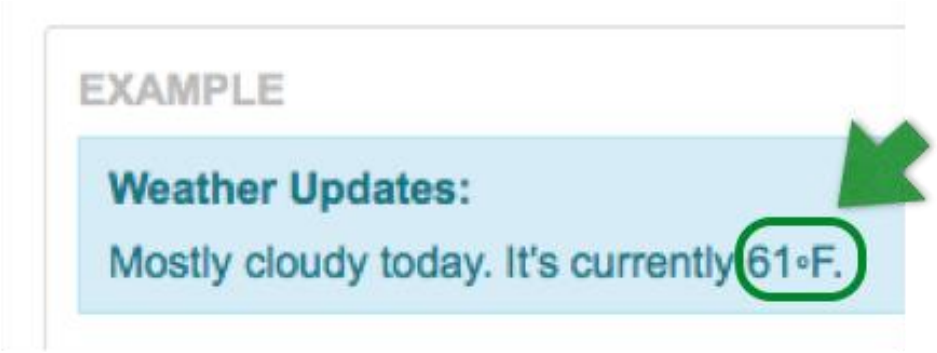
Jens Meinicke, Chu-Pan Wong, Christian Kästner

FOSD Meeting 2015

Feature Interaction

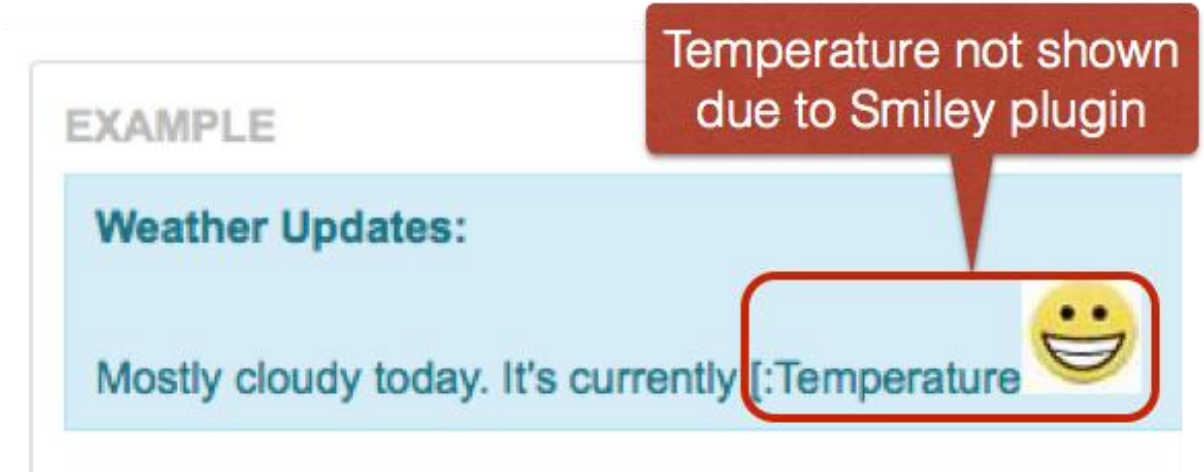
Weather

Smiley

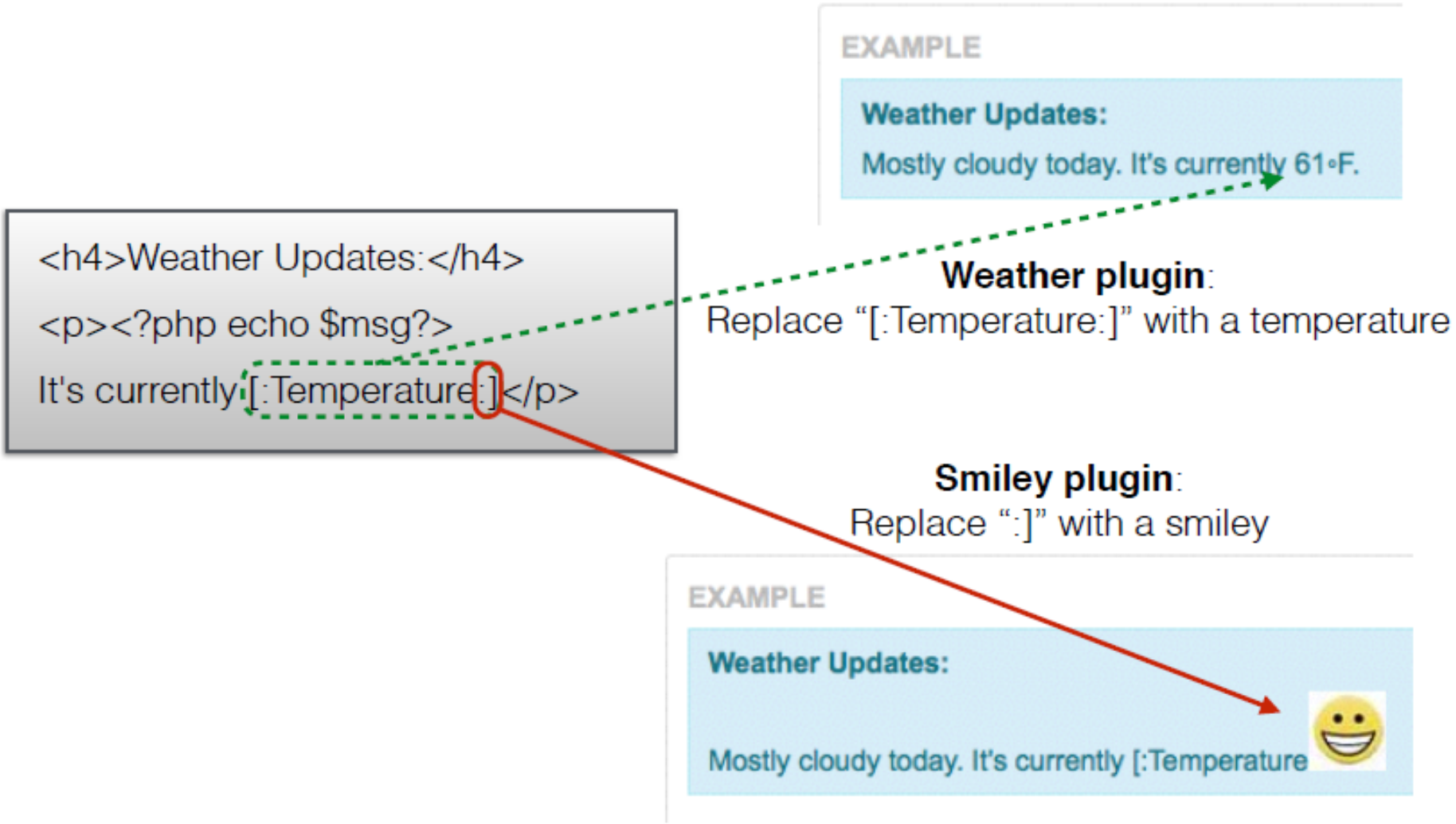


Weather

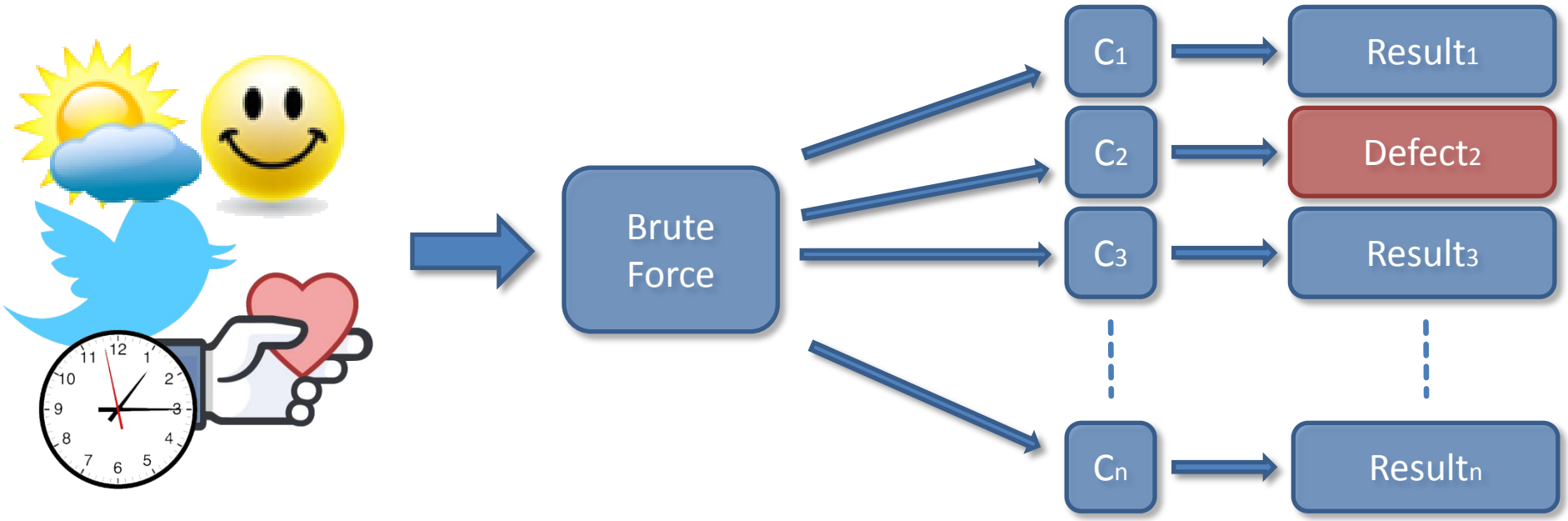
Smiley



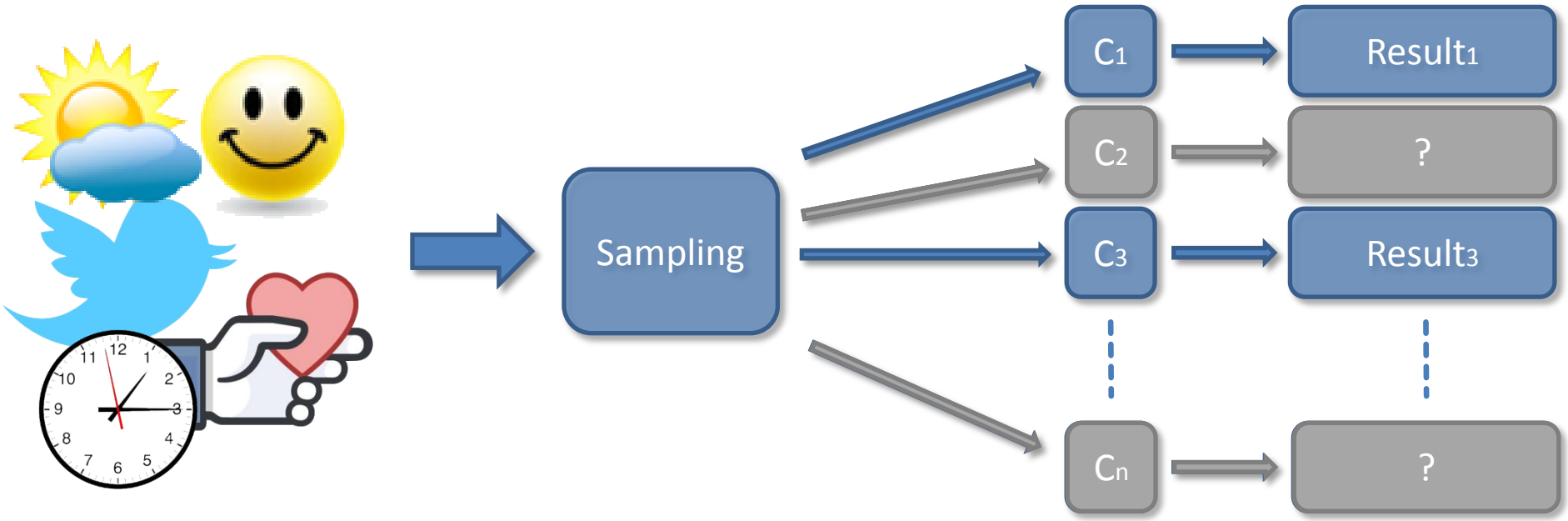
Feature Interaction

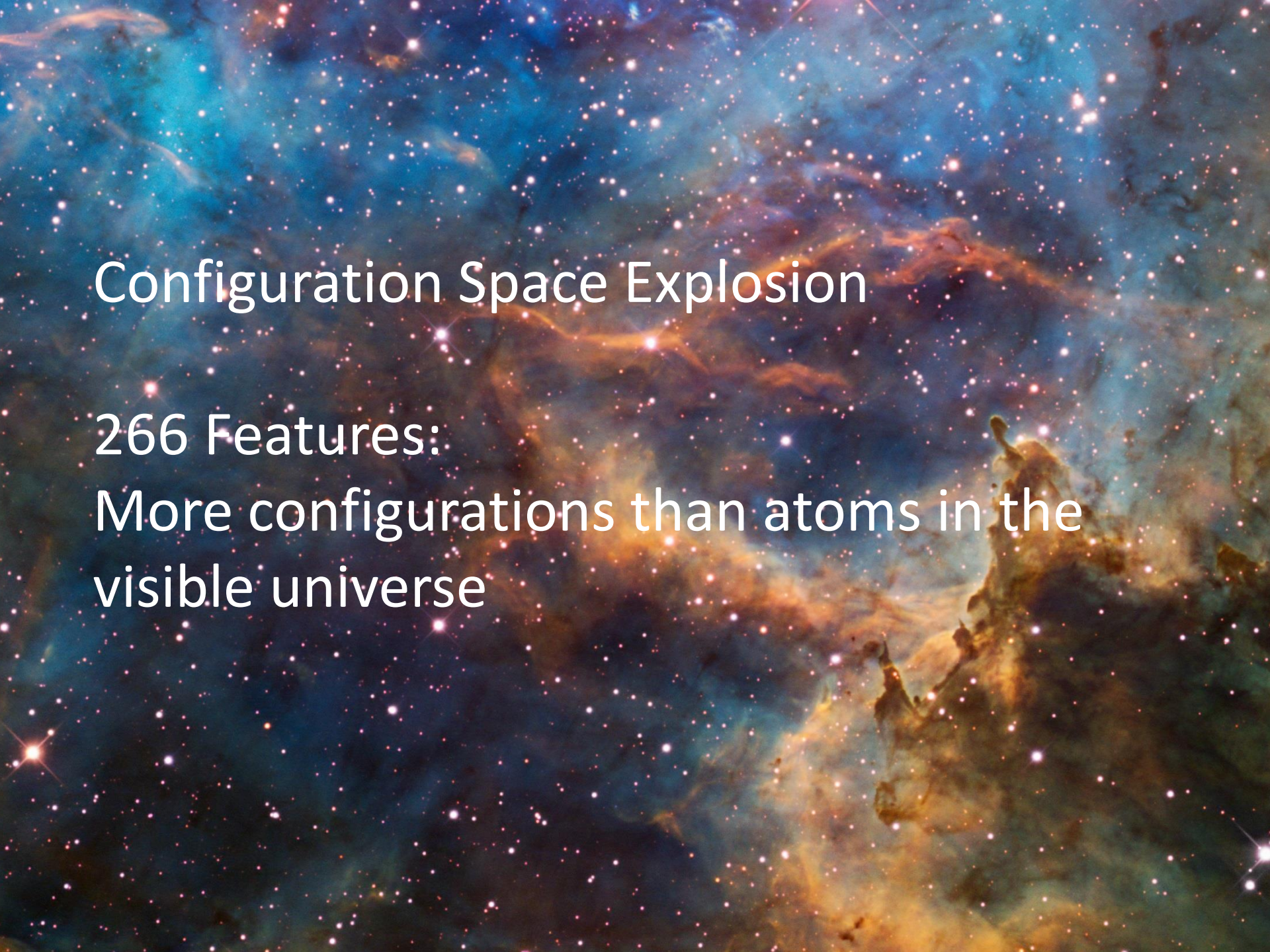


Testing Configurable Systems

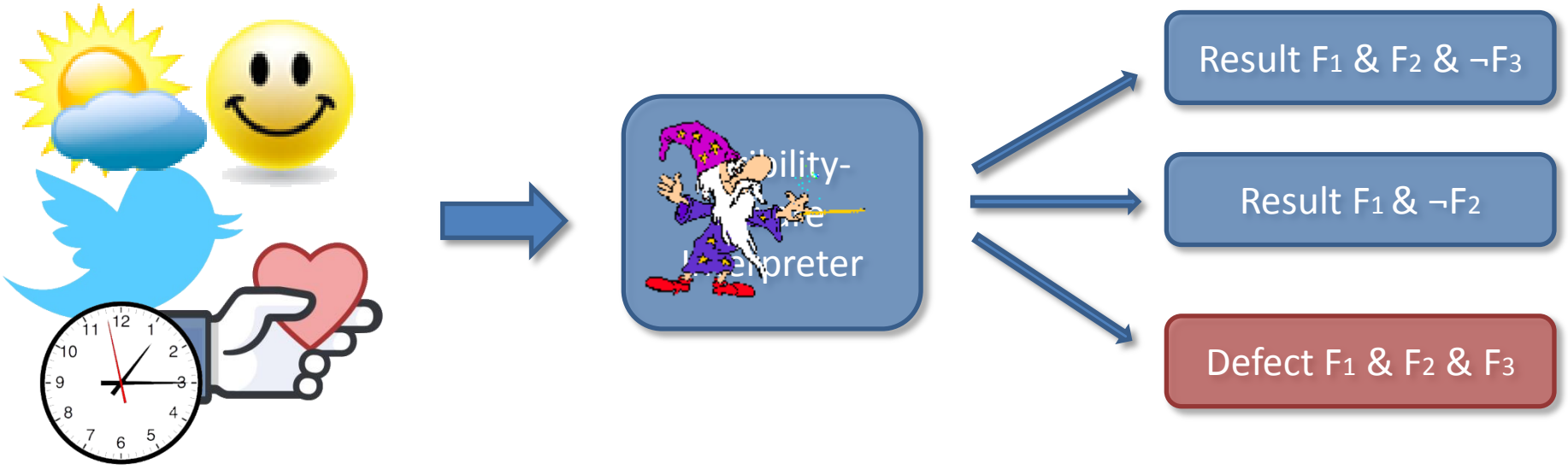


Testing Configurable Systems




Configuration Space Explosion
266 Features:
More configurations than atoms in the visible universe

Testing Configurable Systems



[CRE12, NCN14, M14]

VarexJ

Variability-Aware **ex**ecution for **J**ava

Based on the interpreter of Java Pathfinder

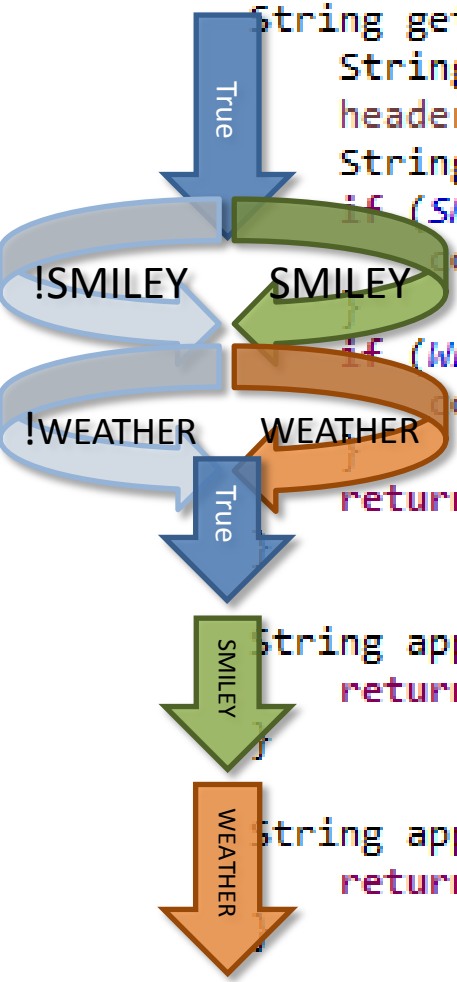
Share as much redundant instructions as possible

Share redundant data

https://github.com/meinicke/VarexJ

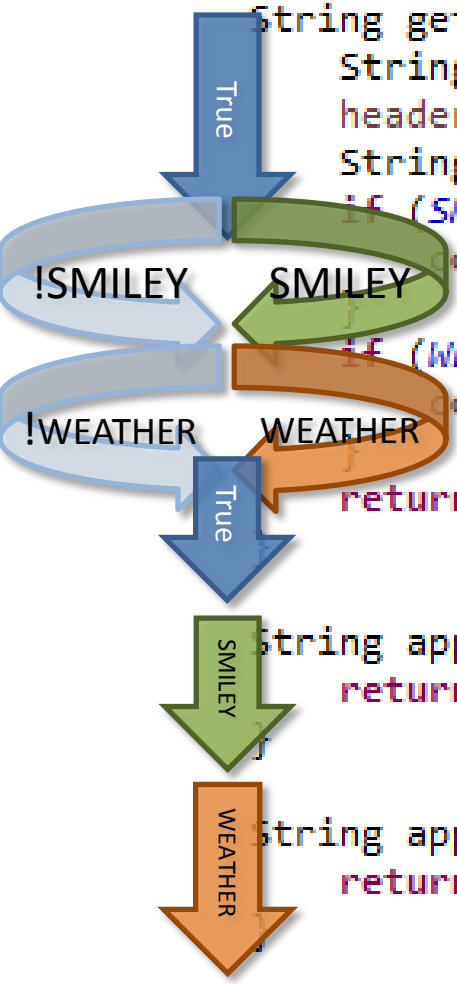
```
content = Choice(SMILEY,
  “<p>…[:Temperature<img=“smiley.jpg”></p>”,
  Choice(WEATHER,
    “<p>…10°C>”,
    “<p>…[:Temperature:]>”))
```

```
@Conditional
static boolean SMILEY = true;
@Conditional
static boolean WEATHER = true;

String getHTML() {
    String header = "<h4>Weather Updates:</h4>";
    header = header + '\n';
    String content = "<p>It's currently [:Temperature:].</p>";
    if (SMILEY) {
        content = applySmiley(content);
    }
    if (WEATHER) {
        content = applyWeather(content);
    }
    return header + content;
}

String applySmiley(String content) {
    return content.replaceAll("\\:\\]", "<img src=\"smiley.jpg\">");
}

String applyWeather(String content) {
    return content.replaceAll("\\[:Temperature\\:\\]", "10°C");
}
```



```
content = Choice(SMILEY,
  “<p>…[:Temperature<img=“smiley.jpg”></p>”,
  Choice(WEATHER,
    “<p>…10°C>”,
    “<p>…[:Temperature:]>”))
```

```
@Conditional
static boolean SMILEY = true;
@Conditional
static boolean WEATHER = true;

String getHTML() {
    String header = "<h4>Weather Updates:</h4>";
    header = header + '\n';
    String content = "<p>It's currently [:Temperature:].</p>";
    if (SMILEY) {
        content = applySmiley(content);
    }
    if (WEATHER) {
        content = applyWeather(content);
    }
    return
```

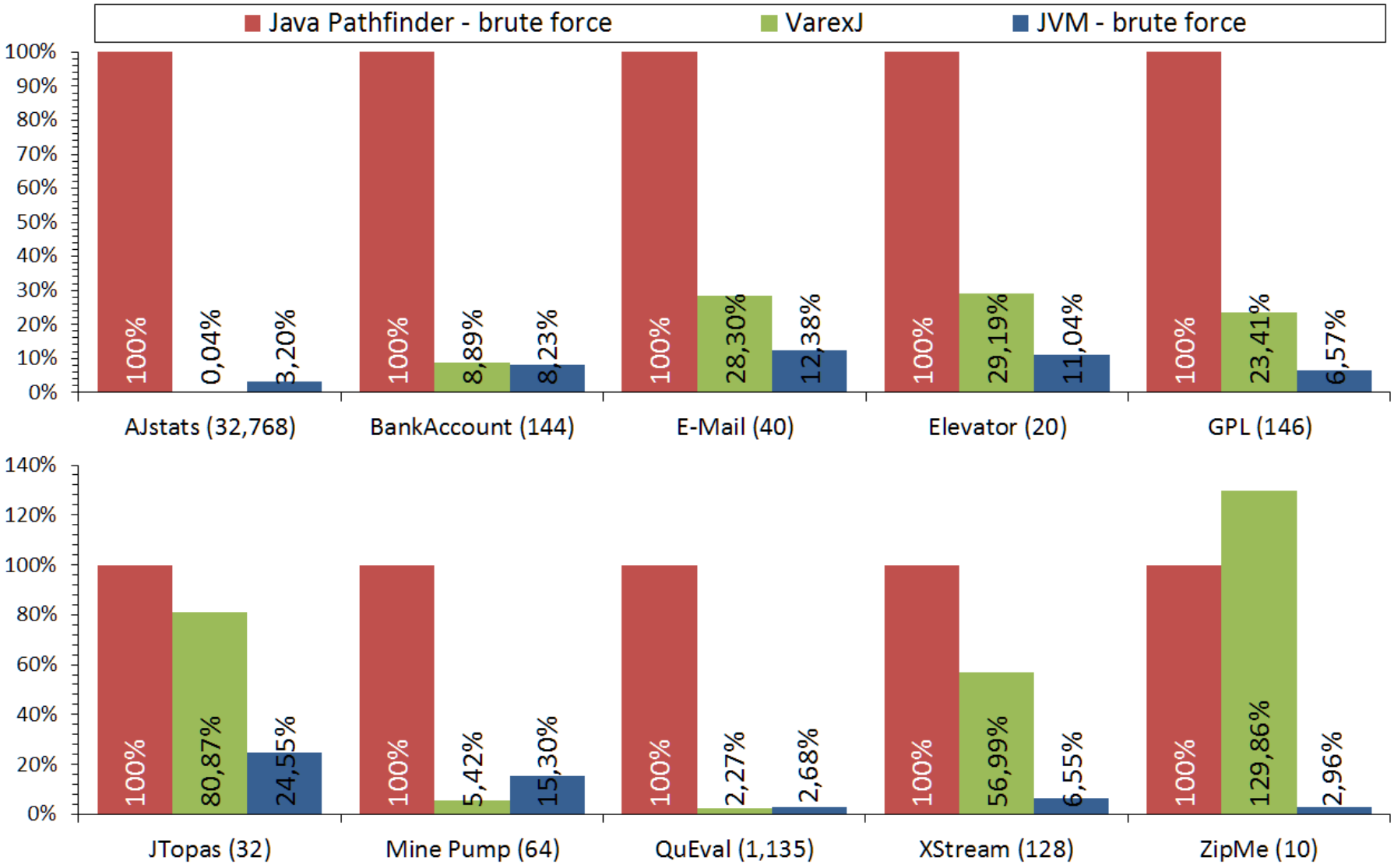



```
{<h4>Weather Updates:</h4>
<p>It's currently [:Temperature<img src="smiley.jpg">.</p>} : SMILEY
{<h4>Weather Updates:</h4>
<p>It's currently [:Temperature:].</p>} : !SMILEY & !WEATHER
{<h4>Weather Updates:</h4>
<p>It's currently 10°C.</p>} : !SMILEY & WEATHER
```

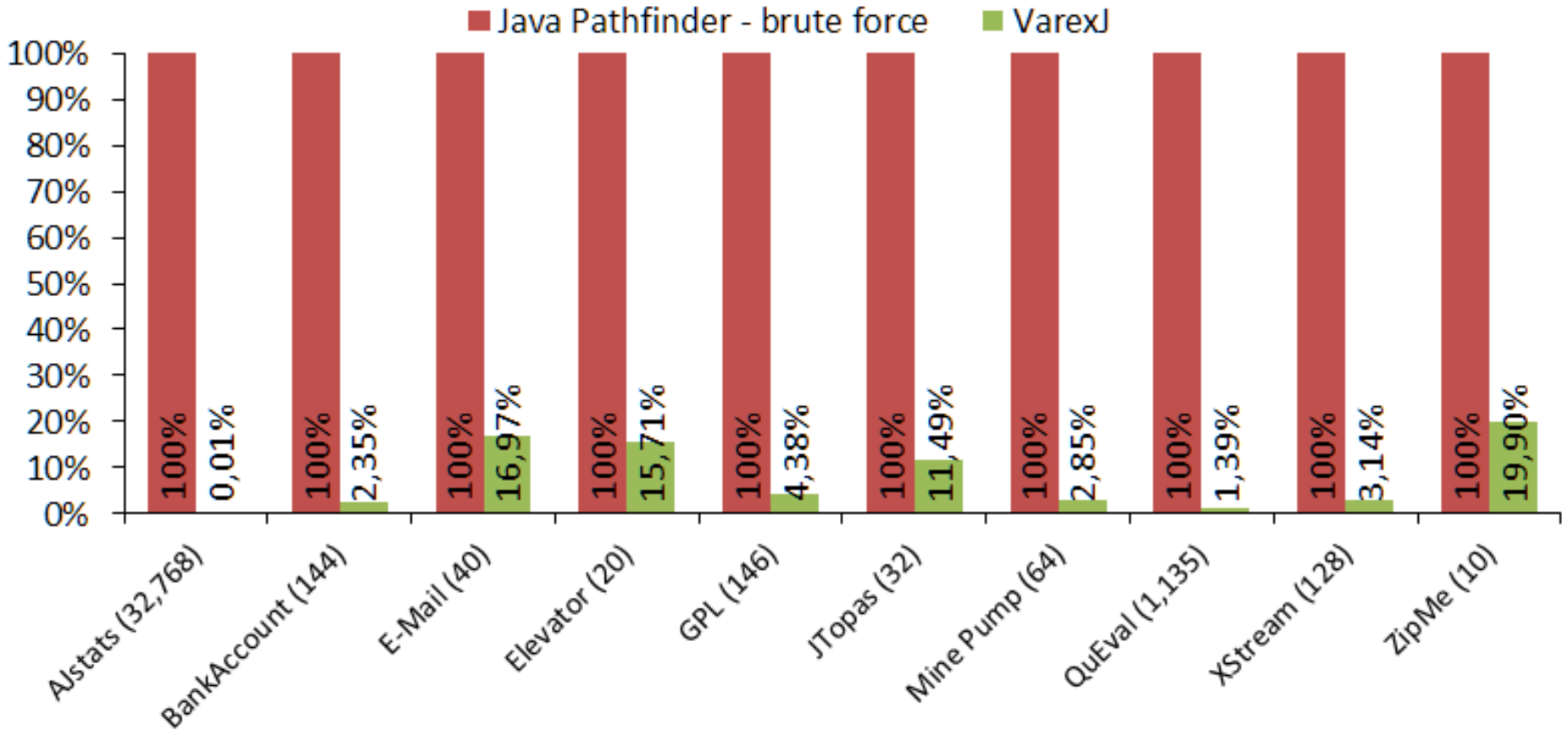
```
String app
    return
}

String applyWeather(String content) {
    return content.replaceAll("\\[:Temperature\\:\\]", "10°C");
}
```

Does it Scale?

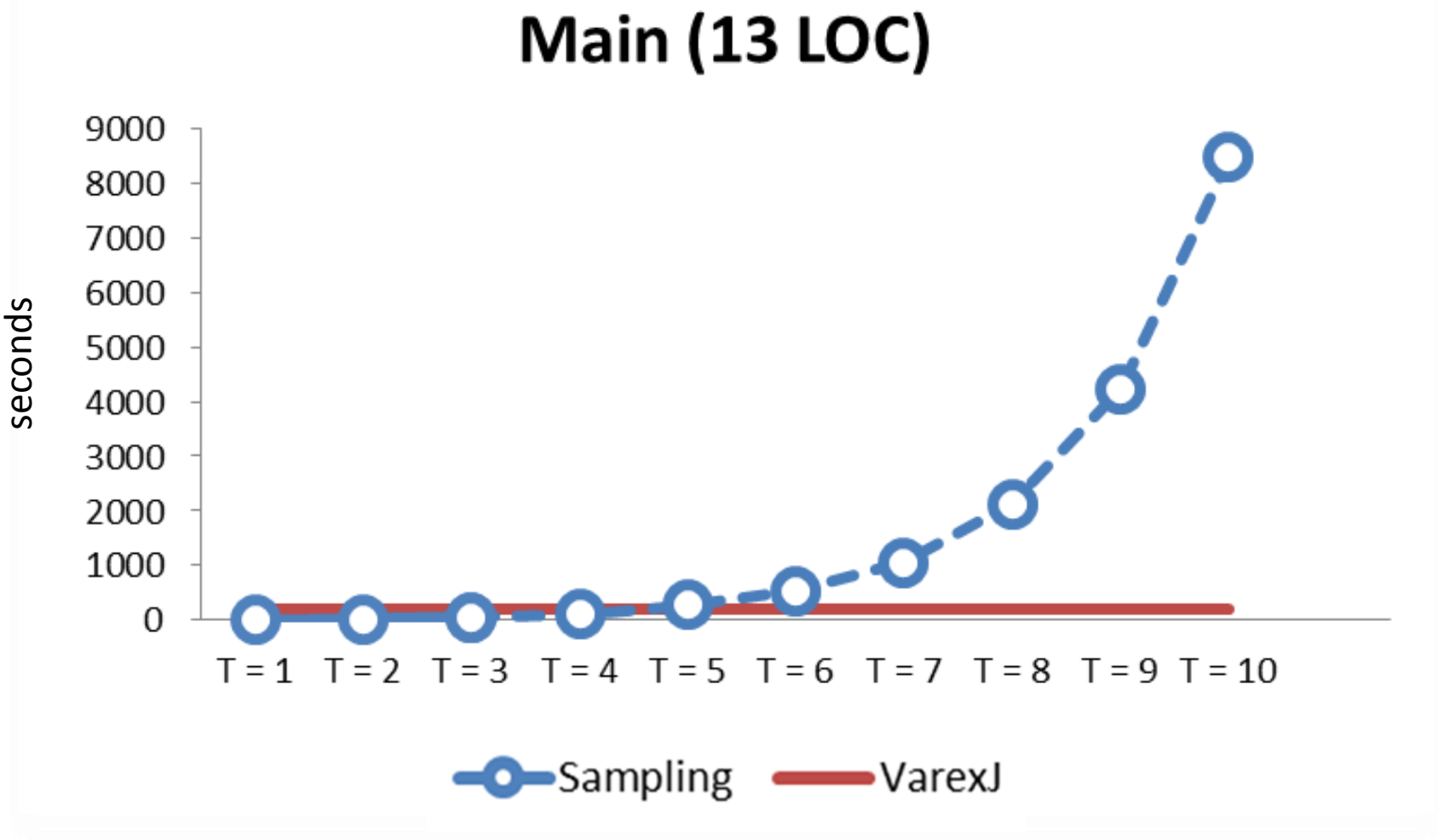
Execution Time



Sharing of Instructions

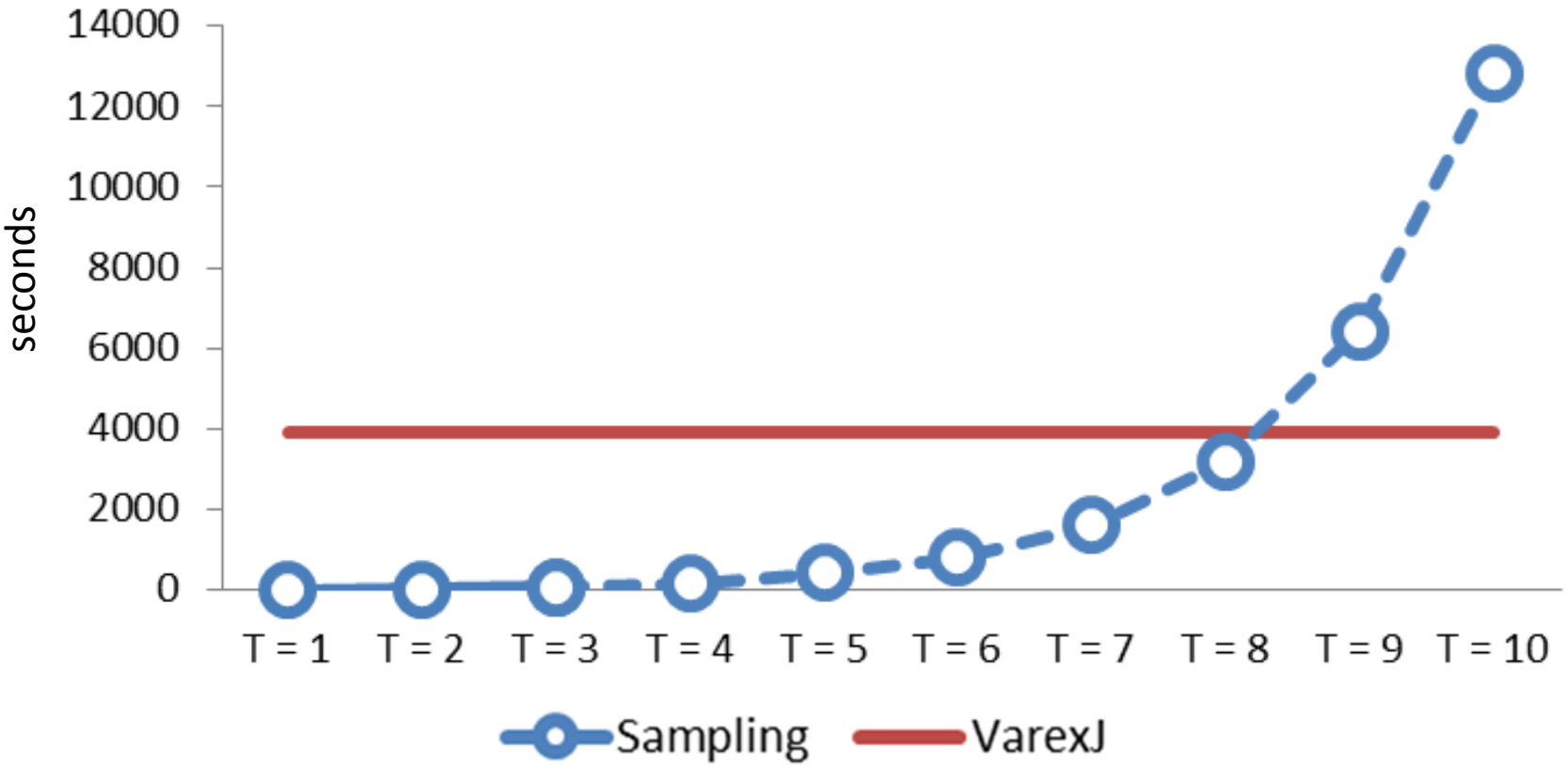


checkstyle


Main (13 LOC)
seconds
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
T = 1
T = 2
T = 3
T = 4
T = 5
T = 6
T = 7
T = 8
T = 9
T = 10
Sampling
VarexJ

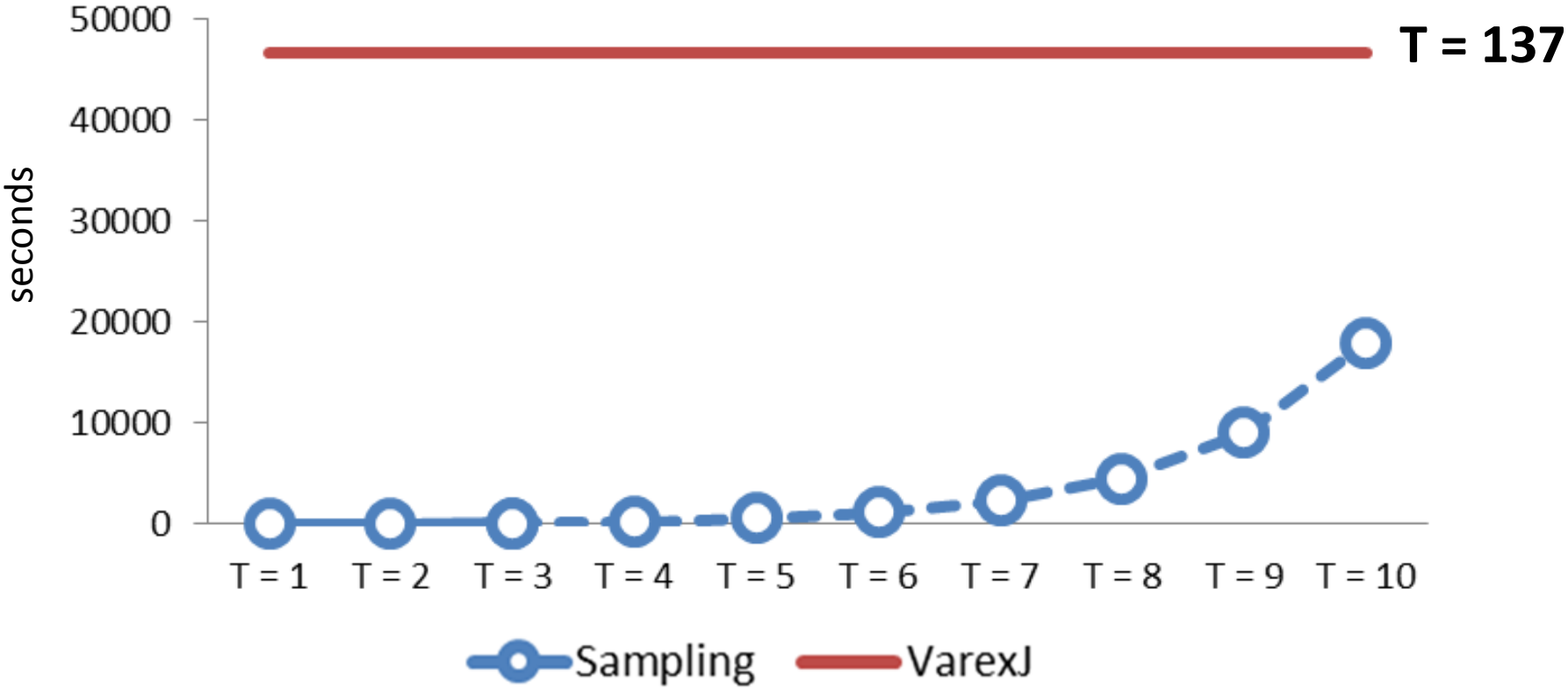
checkstyle

RandomSearch (153 LOC)



checkstyle

Checker (641 LOC)



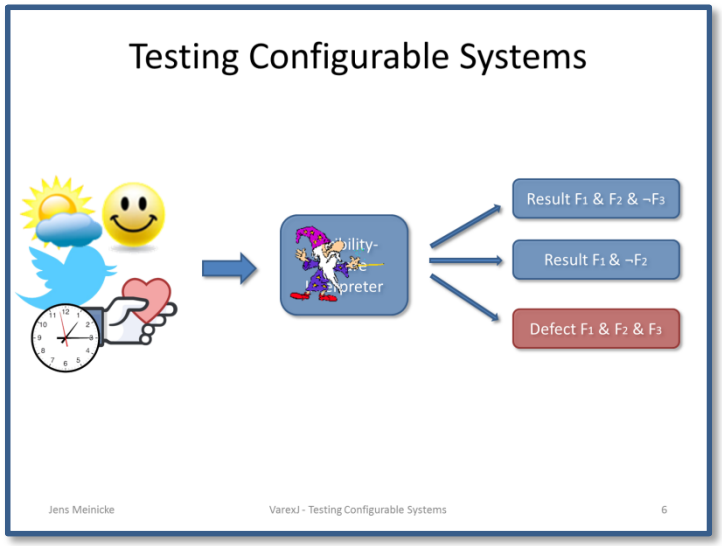
References

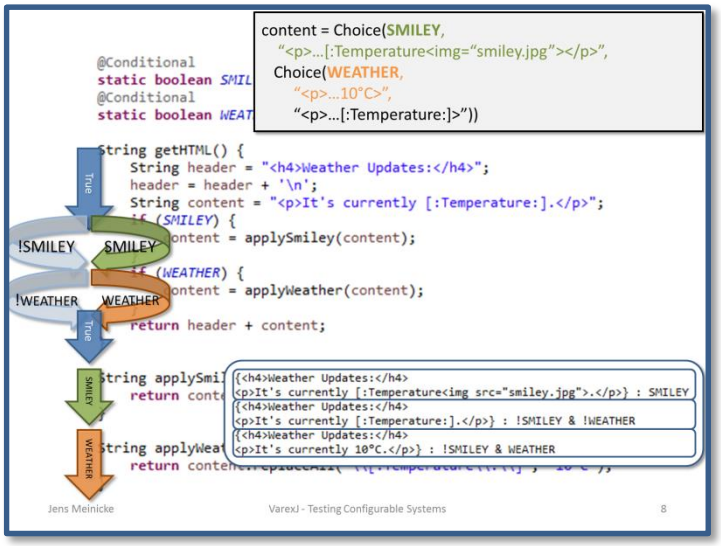
Jens Meinicke. **VarexJ: A Variability-Aware Interpreter for Java Applications**. Master's thesis, University of Magdeburg, Germany, December 2014.

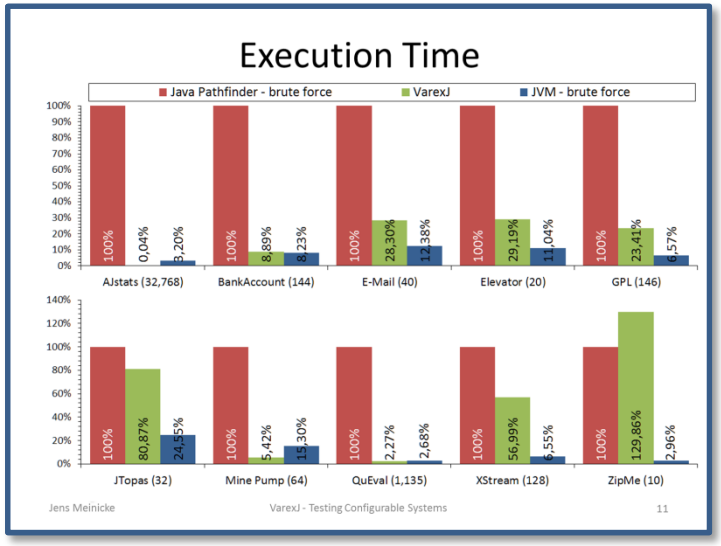
Hung Viet Nguyen, Christian Kästner, and Tien N. Nguyen. **Exploring Variability-Aware Execution for Testing Plugin-Based Web Applications**. In Proceedings of the International Conference on Software Engineering (ICSE), pages 907–918. ACM, 2014.

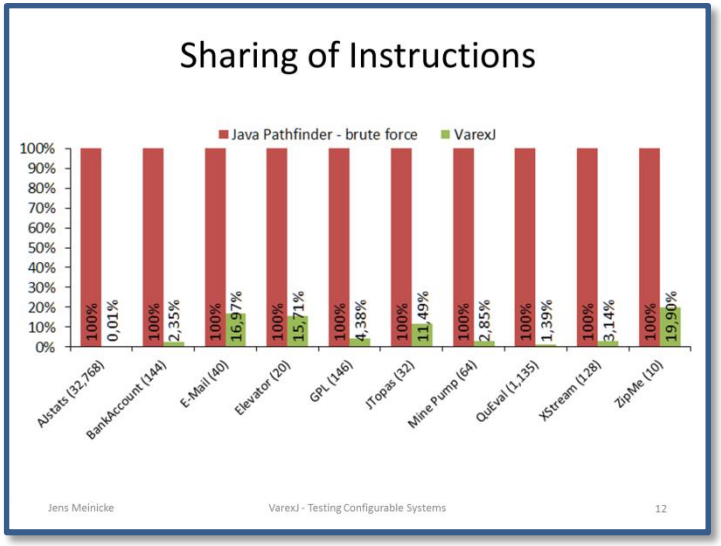
Christian Kästner, Alexander von Rhein, Sebastian Erdweg, Jonas Pusch, Sven Apel, Tillmann Rendel, and Klaus Ostermann. **Toward Variability-Aware Testing**. In Proceedings of the International SPLC Workshop Feature-Oriented Software Development (FOSD), pages 1–8. ACM, 2012b.

So far…


Testing Configurable Systems
Result F1 & F2 & ¬F3
Result F1 & ¬F2
Defect F1 & F2 & F3
Jens Meinicke
VarexJ - Testing Configurable Systems
6


content = Choice(SMILEY,
" ...[:Temperature<img="smiley.jpg"> ",
Choice(WEATHER,
" ...10°C>",
" ...[:Temperature:]>"))
@Conditional
static boolean SMIL
@Conditional
static boolean WEAT
String getHTML() {
String header = "<h4>Weather Updates:</h4>";
header = header + '\n';
String content = " It's currently [:Temperature:]. ";
if (SMILEY) {
ontent = applySmiley(content);
if (WEATHER) {
ontent = applyWeather(content);
return header + content;
True
!SMILEY
SMILEY
!WEATHER
WEATHER
True
{<h4>Weather Updates:</h4>
It's currently [:Temperature<img src="smiley.jpg">. } : SMILEY
{<h4>Weather Updates:</h4>
It's currently [:Temperature:]. } : !SMILEY & !WEATHER
{<h4>Weather Updates:</h4>
It's currently 10°C. } : !SMILEY & WEATHER
Jens Meinicke
VarexJ - Testing Configurable Systems
8


Execution Time
Java Pathfinder - brute force
VarexJ
JVM - brute force
AJstats (32,768)
BankAccount (144)
E-Mail (40)
Elevator (20)
GPL (146)
JTopas (32)
Mine Pump (64)
QuEval (1,135)
XStream (128)
ZipMe (10)
Jens Meinicke
VarexJ - Testing Configurable Systems
11


Sharing of Instructions
Java Pathfinder - brute force
VarexJ
AJstats (32,768)
BankAccount (144)
E-Mail (40)
Elevator (20)
GPL (146)
JTopas (32)
Mine Pump (64)
QuEval (1,135)
XStream (128)
ZipMe (10)
Jens Meinicke
VarexJ - Testing Configurable Systems
12

VarexJ: A Variability-Aware Java Interpreter

Understanding Feature Interactions

Jens Meinicke, Chu-Pan Wong, Christian Kästner

FOSD Meeting 2015

Analyzing Variability

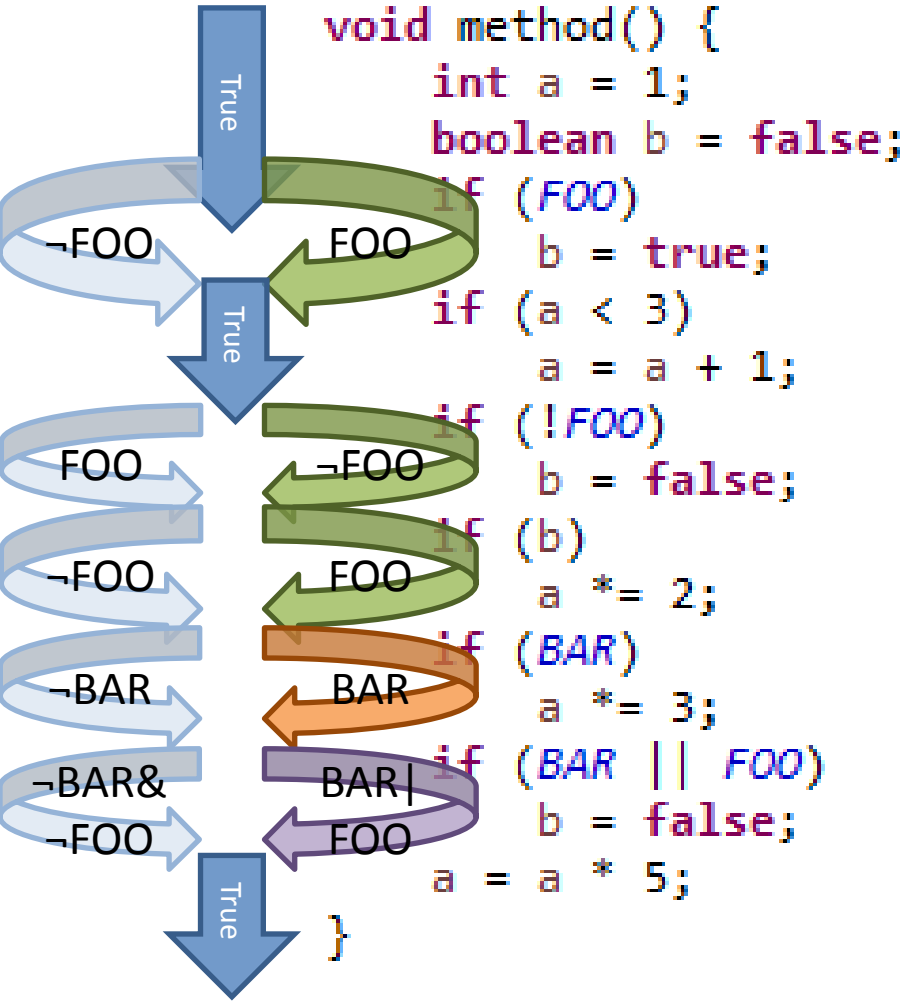
Where do features interact?

Why do features interact?

Are they supposed to interact?

Is the interaction accidental?

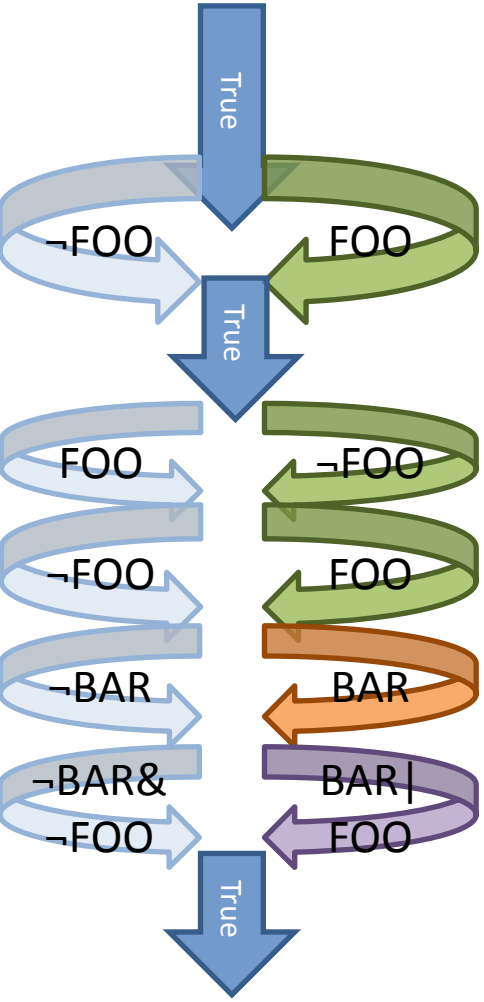
Variability-Aware Execution




```
void method() {
    int a = 1;
    boolean b = false;
    if (FOO)
        b = true;
    if (a < 3)
        a = a + 1;
    if (!FOO)
        b = false;
    if (b)
        a *= 2;
    if (BAR)
        a *= 3;
    if (BAR || FOO)
        b = false;
    a = a * 5;
}
```

a = 1 b = false
a = 1 b = Chc(**FOO**, **true**, false)
a = **2** b = Chc(FOO, true, false)
a = 2 b = Chc(FOO, true, false)
a = Chc(FOO, **4**, 2) b = Chc(FOO, true, false)
a = Chc(FOO, Chc(**BAR, 12**, 4), Chc(**BAR, 6**, 2)) b = Chc(FOO, true, false)
a = Chc(FOO, Chc(BAR,2, 4), Chc(BAR, 6, 2)) b = **false**
a = Chc(FOO,Chc(BAR,**60**,**20**),Chc(BAR,**30**, **10**)) b = false

Variability in Executions




```
void method() {
    int a = 1;
    boolean b = false;
    if (FOO)
        b = true;
    if (a < 3)
        a = a + 1;
    if (!FOO)
        b = false;
    if (b)
        a *= 2;
    if (BAR)
        a *= 3;
    if (BAR || FOO)
        b = false;
    a = a * 5;
}
```

Max features: 2
!FOO & BAR | FOO

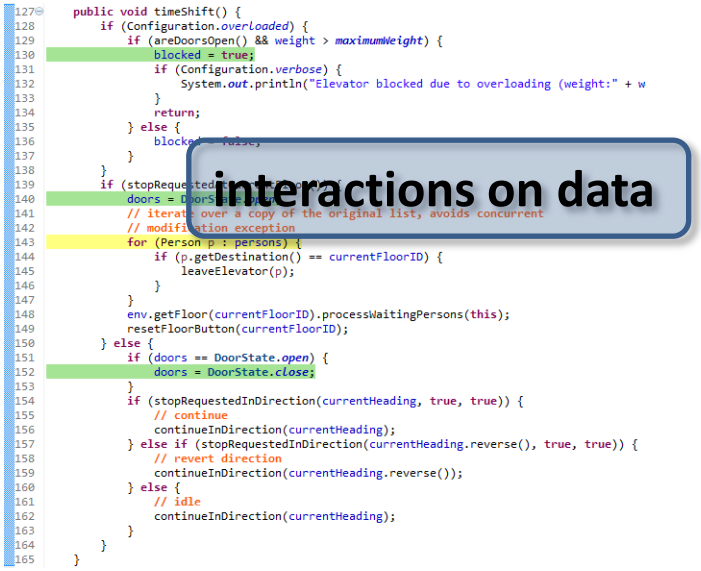
Variability in Data

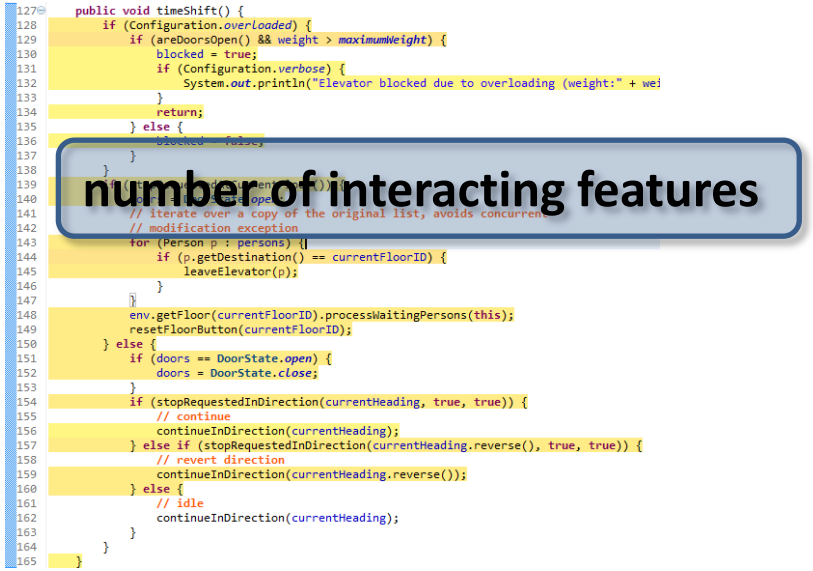
```
void method() {
    int a = 1;
    boolean b = false;
    if (FOO)
        b = true;
    if (a < 3)
        a = a + 1;
    if (!FOO)
        b = false;
    if (b)
        a *= 2;
    if (BAR)
        a *= 3;
    if (BAR || FOO)
        b = false;
    a = a * 5;
}
```

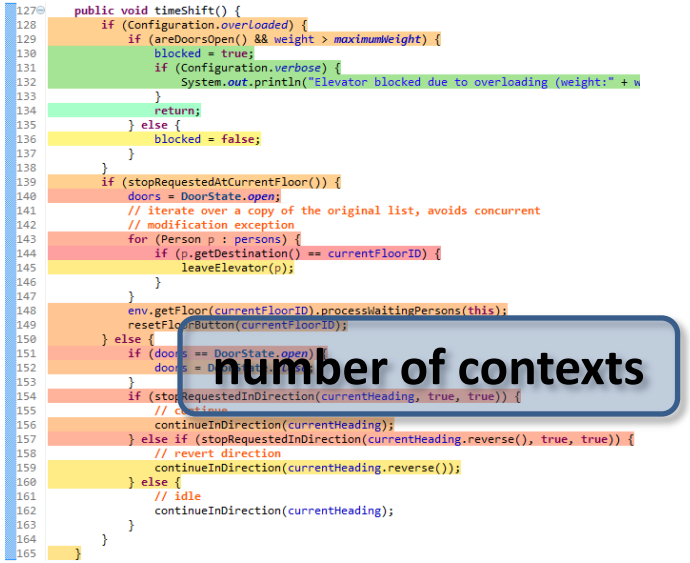
a = 1 b = false
a = 1 b = Chc(**FOO**, **true**, false)
a = **2** b = Chc(FOO, true, false)
a = 2 b = Chc(FOO, true, false)
a = Chc(FOO, **4**, 2) b = Chc(FOO, true, false)
a = Chc(FOO, Chc(**BAR, 12**, 4), Chc(**BAR, 6**, 2))
R, 6, 2))
R,**30**, **10**))

Interaction: 2
a (BAR):
2 : {!FOO: [2] ; FOO: [4] }
->
4 : {!FOO & !BAR: [2] ; FOO & !BAR: [4] ;
!FOO & BAR: [6] ; FOO & BAR: [12] }

Some Views on Elevator







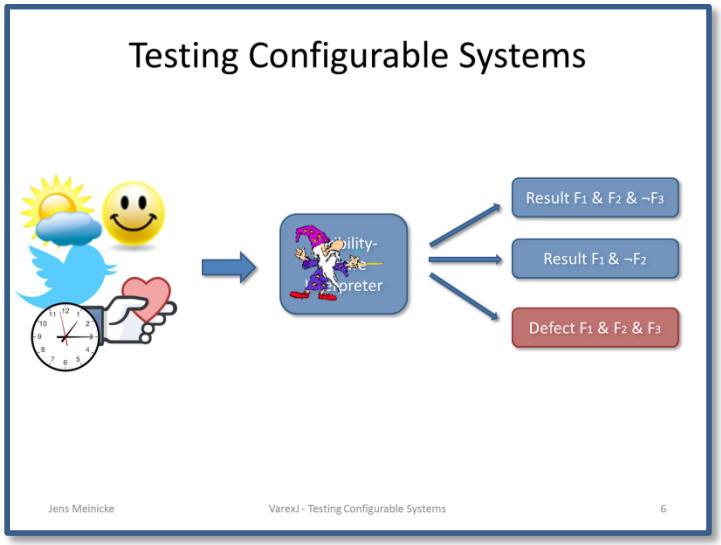
Big Questions

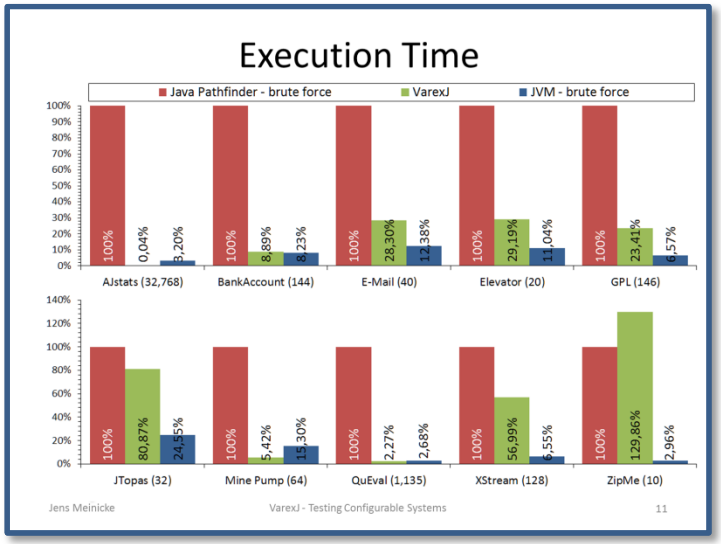
How do features interact in real world systems?

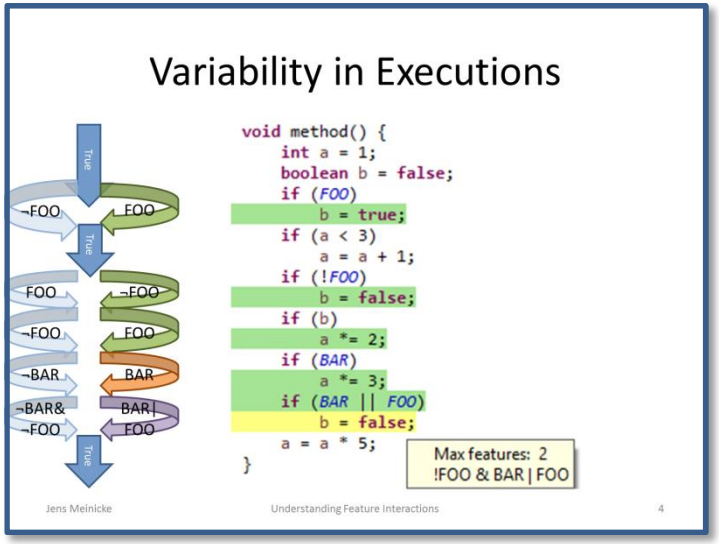
What pattern cause (high) feature interactions?

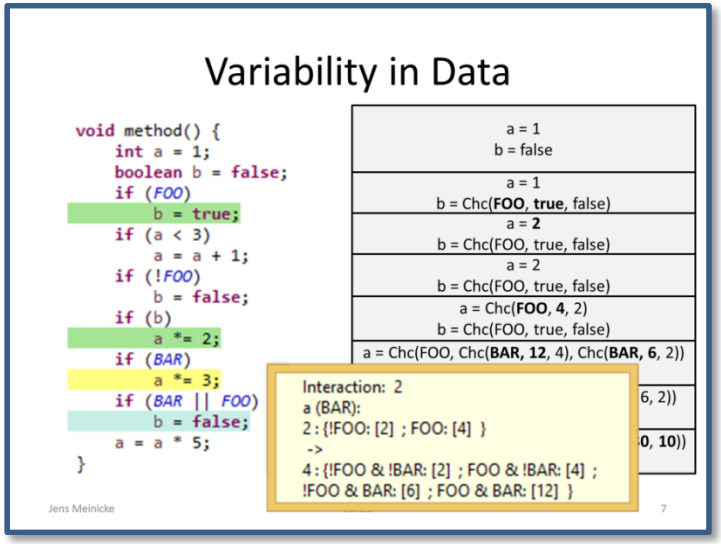
How do dangerous and unpredictable feature interactions appear?

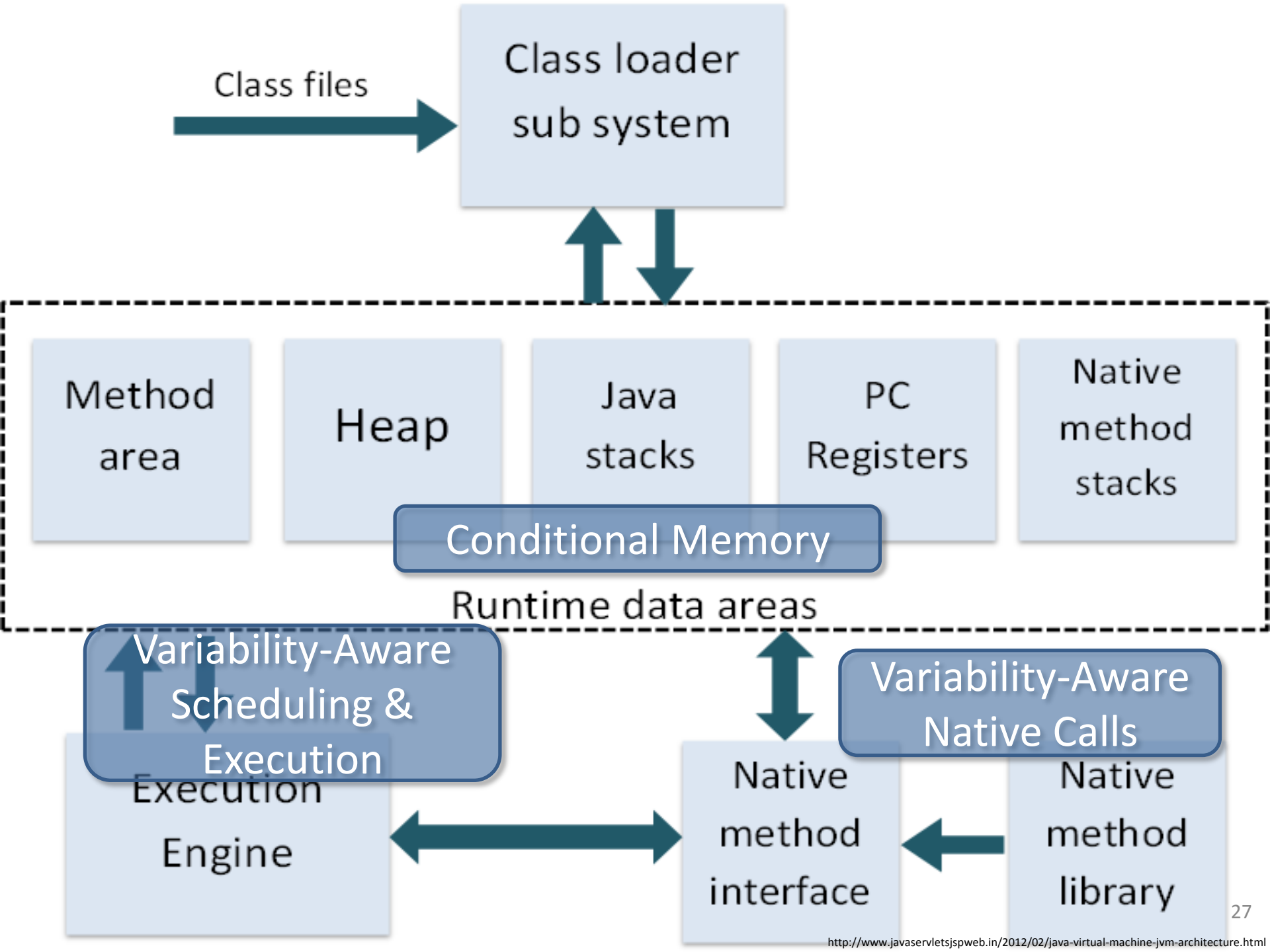
Summary


Testing Configurable Systems
Result F1 & F2 & ¬F3
Result F1 & ¬F2
Defect F1 & F2 & F3
Jens Meinicke
VarexJ - Testing Configurable Systems
6


Execution Time
Java Pathfinder - brute force
VarexJ
JVM - brute force
AJstats (32,768)
BankAccount (144)
E-Mail (40)
Elevator (20)
GPL (146)
JTopas (32)
Mine Pump (64)
QuEval (1,135)
XStream (128)
ZipMe (10)
Jens Meinicke
VarexJ - Testing Configurable Systems
11


Variability in Executions
void method() {
int a = 1;
boolean b = false;
if (FOO)
b = true;
if (a < 3)
a = a + 1;
if (!FOO)
b = false;
if (b)
a *= 2;
if (BAR)
a *= 3;
if (BAR || FOO)
b = false;
a = a * 5;
}
Max features: 2
!FOO & BAR | FOO
Jens Meinicke
Understanding Feature Interactions
4


Variability in Data
void method() {
int a = 1;
boolean b = false;
if (FOO)
b = true;
if (a < 3)
a = a + 1;
if (!FOO)
b = false;
if (b)
a *= 2;
if (BAR)
a *= 3;
if (BAR || FOO)
b = false;
a = a * 5;
}
a = 1
b = false
a = 1
b = Chc(FOO, true, false)
a = 2
b = Chc(FOO, true, false)
a = 2
b = Chc(FOO, true, false)
a = Chc(FOO, 4, 2)
b = Chc(FOO, true, false)
a = Chc(FOO, Chc(BAR, 12, 4), Chc(BAR, 6, 2))
Interaction: 2
a (BAR):
2 : {!FOO: [2] ; FOO: [4] }
->
4 : {!FOO & !BAR: [2] ; FOO & !BAR: [4] ;
!FOO & BAR: [6] ; FOO & BAR: [12] }
Jens Meinicke
7

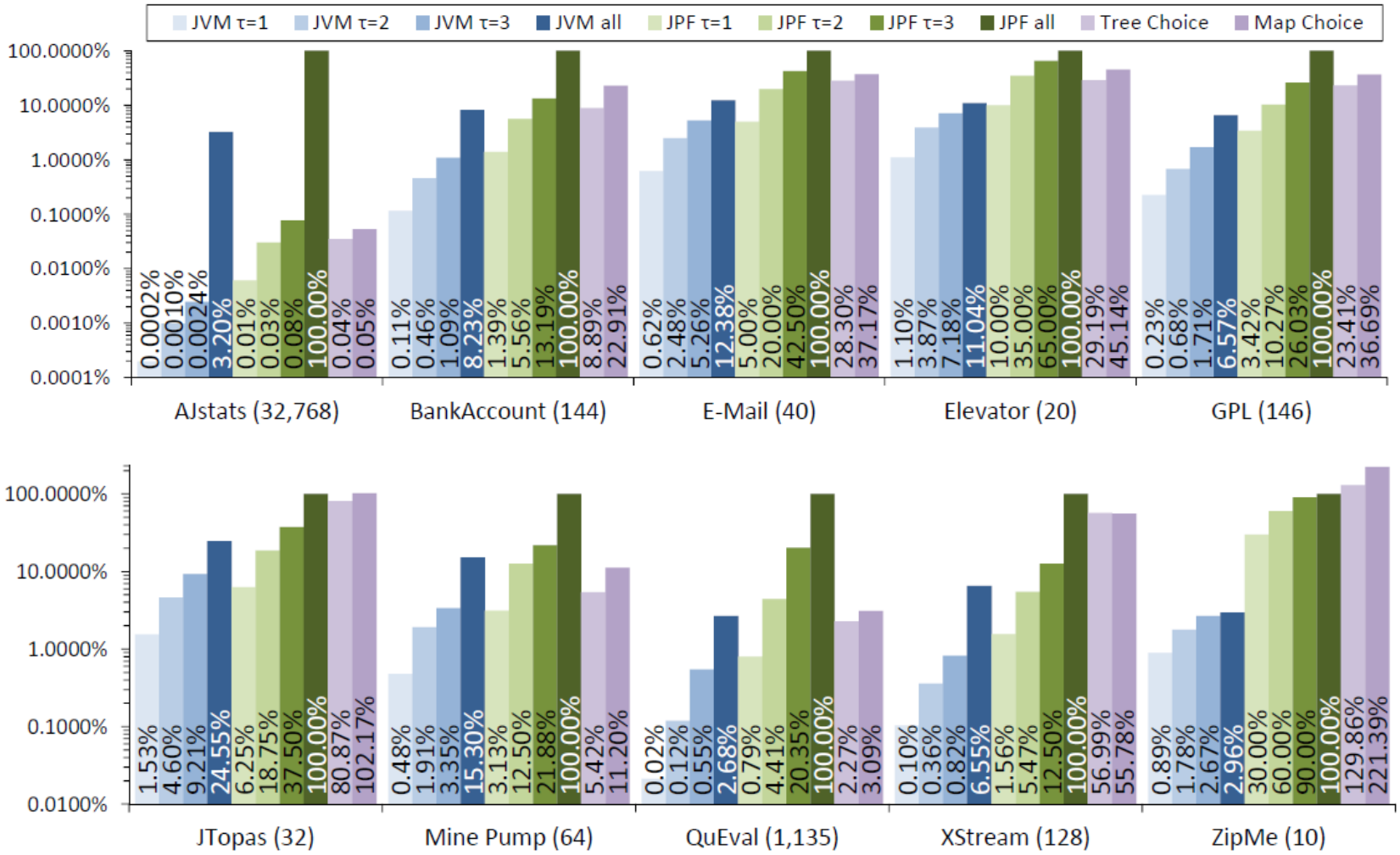


http://www.javaservletsjspweb.in/2012/02/java-virtual-machine-jvm-architecture.html

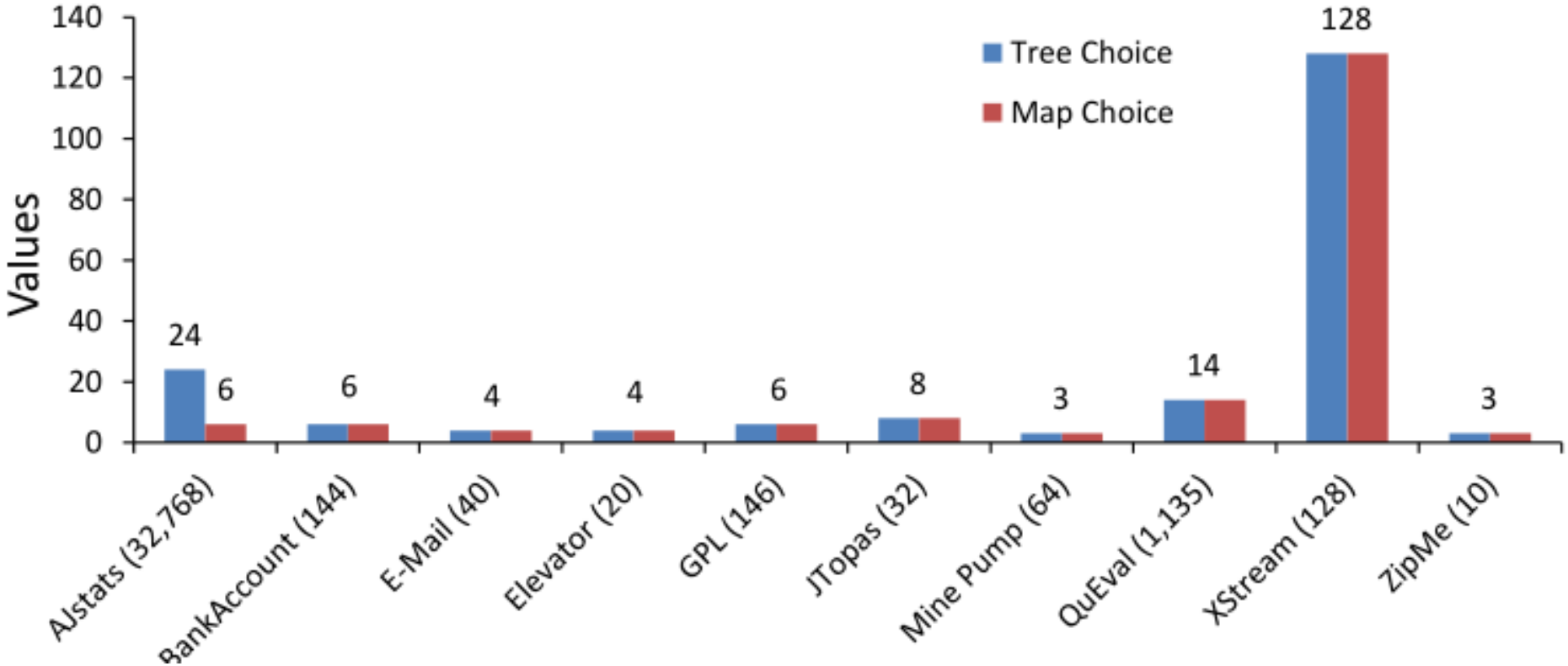
Example Programs

System	Instructions	Features	Products
AJstats	35235	20	32768
BankAccount	4217	10	144
E-Mail	2075	9	40
Elevator	4816	6	20
GPL	2971	22	146
JTopas	14528	5	32
Mine Pump	1687	7	64
QuEval	18294	30	1135
XStream	38234	7	128
ZipMe	12901	8	10

Time for Sampling



Interactions



Sharing of Instrucions