Dynamic Taint Tracking for Modern Java Virtual Machines

Supplemental

Table 1: Functional Benchmarks. For each benchmark group (Group), we report the total number of tests in the group (#), and a description of the type of functionality exercised by tests in that group (Description).

Group	#	Description
Array Access	74	Loads and stores elements from arrays of various types, lengths, and shapes. Checks taint tag propagation to and from array elements and indices.
Array Length	5	Creates arrays of various types, lengths, and shapes. Checks taint tag propagation to and from array lengths.
Array Reflection	75	Uses `java.lang.reflect.Array` to create arrays, load array elements, and store array elements. Checks taint tag propagation to and from array elements, indices, and lengths.
Assignment	12	Assigns values of various types to local variables. Checks taint tag propagation to and from local variables.
Boxed Type	8	Creates various boxed primitive type (e.g., `java.lang.Integer`) instances. Checks taint tag propagation through the boxing and unboxing of primitive values.
Class Reflection	7	Checks the correctness of class' attributes inspected using reflection.
Collection	6	Checks taint tag propagation through operations on common JCL collections (e.g., `java.util.ArrayList`).
Conditional	4	Checks taint tag propagation in the presence of conditional branches.
Constructor Reflection	88	Invokes various constructors using reflection. Checks taint tag propagation through the constructors' arguments.
Field	6	Loads and stores values from fields of various types. Checks taint tag propagation to and from the fields.
Field Reflection	176	Loads and stores values from fields of various types using reflection. Checks taint tag propagation to and from the field.
Jdk Unsafe	322	Uses `jdk.internal.misc.Unsafe` to load and store array elements and fields of various types using various access semantics. Checks taint tag propagation to and from the accessed values.
Lambda	7	Invokes various lambda expressions. Checks taint tag propagation through the expressions' arguments and return values.
Loop	4	Checks taint tag propagation in the presence of various looping constructs.
Method Call	14	Invokes various methods. Checks taint tag propagation through the methods' arguments and return values.
Method Handle	40	Uses `java.lang.invoke.MethodHandle` instances to invoke various methods. Checks taint tag propagation through the methods' arguments and return values.
Method Handle 9+	13	Uses `java.lang.invoke.MethodHandle` instances created using methods added to the JCL in Java 9 to invoke various methods. Checks taint tag propagation through the methods' arguments and return values.
Method Reflection	213	Invokes various methods using reflection. Checks taint tag propagation through the methods' arguments and return values.
Record Type	68	Creates record type instances with various types of components. Checks taint tag propagation through the records' components and methods.
Static Initializer	2	Checks taint tag taint tag propagation through static method calls defined in classes that have not yet been initialized.
String	26	Checks taint tag propagation through operations on `java.lang.String`.
String Builder Concat	37	Checks taint tag propagation through pre-Java 9 string concatenation which uses `java.lang.StringBuilder`.
String Indy Concat	37	Checks taint tag propagation through Java 9+ string concatenation which uses `invokedynamic`.
Sun Unsafe	270	Uses `sun.misc.Unsafe` to load and store array elements and fields of various types using various access semantics. Checks taint tag propagation to and from the accessed values.
Throwable	5	Catches explicitly thrown exceptions. Checks taint tag propagation from the thrown exception to the caught exception.
Var Handle	1932	Uses `java.lang.invoke.VarHandle` instances to load and store array elements and fields of various types using various access semantics. Checks taint tag propagation to and from the accessed values.

Table 2: Execution Time and Peak Memory Usage Statistical Tests. We report the two-tailed, asymptotic p-value (p) and the Vargha-Delaney $\hat{A}_{12}$ statistic ($\hat{A}_{12}$) for Mann-Whitney U tests comparing between the execution time (left) and peak memory usage (right) of Galette against MirrorTaint and Phosphor. Values that are statistically significantly ($p < 0.0167$) greater than or less than Galette's are colored green and red, respectively.

	Execution Time				Peak Memory Usage			
	MirrorTaint		Phosphor		MirrorTaint		Phosphor	
	p	$\hat{A}_{12}$	p	$\hat{A}_{12}$	p	$\hat{A}_{12}$	p	$\hat{A}_{12}$
avrora	$2.562 \cdot 10^{-34}$	1.000	$1.125 \cdot 10^{-33}$	0.995	$2.562 \cdot 10^{-34}$	1.000	$5.928 \cdot 10^{-34}$	0.997
batik	$2.562 \cdot 10^{-34}$	1.000	—	—	$2.562 \cdot 10^{-34}$	1.000	—	—
biojava	$2.561 \cdot 10^{-34}$	1.000	$2.560 \cdot 10^{-34}$	1.000	$2.562 \cdot 10^{-34}$	1.000	$2.562 \cdot 10^{-34}$	1.000
eclipse	—	—	—	—	—	—	—	—
fop	—	—	$5.930 \cdot 10^{-22}$	0.894	—	—	$2.562 \cdot 10^{-34}$	1.000
graphchi	—	—	$3.672 \cdot 10^{-34}$	0.999	—	—	$2.562 \cdot 10^{-34}$	1.000
h2	$2.560 \cdot 10^{-34}$	1.000	$9.947 \cdot 10^{-31}$	0.972	$6.592 \cdot 10^{-16}$	0.831	$1.046 \cdot 10^{-32}$	0.988
h2o	—	—	—	—	—	—	—	—
jme	$2.562 \cdot 10^{-34}$	1.000	$1.021 \cdot 10^{-5}$	0.681	$2.562 \cdot 10^{-34}$	1.000	$5.236 \cdot 10^{-20}$	0.875
jython	—	—	—	—	—	—	—	—
luindex	—	—	$1.968 \cdot 10^{-13}$	0.801	—	—	$4.530 \cdot 10^{-34}$	0.998
lusearch	$2.561 \cdot 10^{-34}$	1.000	$1.617 \cdot 10^{-7}$	0.714	$5.535 \cdot 10^{-21}$	0.885	$2.562 \cdot 10^{-34}$	1.000
pmd	—	—	—	—	—	—	—	—
spring	—	—	$1.753 \cdot 10^{-33}$	0.994	—	—	$2.562 \cdot 10^{-34}$	1.000
sunflow	$2.561 \cdot 10^{-34}$	1.000	$2.064 \cdot 10^{-25}$	0.926	$2.562 \cdot 10^{-34}$	1.000	$6.770 \cdot 10^{-1}$	0.517
tomcat	$2.561 \cdot 10^{-34}$	1.000	—	—	$2.562 \cdot 10^{-34}$	1.000	—	—
tradebeans	—	—	—	—	—	—	—	—
tradesoap	—	—	—	—	—	—	—	—
xalan	—	—	$7.212 \cdot 10^{-2}$	0.574	—	—	$2.562 \cdot 10^{-34}$	1.000
zxing	$2.560 \cdot 10^{-34}$	1.000	—	—	$2.562 \cdot 10^{-34}$	1.000	—	—