Mining Ultra-Large-Scale Software Repositories and StackOverflow to study *sun.misc.Unsafe* API usage patterns in Java applications

Luis Mastrangelo*, Matthias Hauswirth* and Nate Nystrom*
*Faculty of Informatics, University of Lugano

***Abstract*—**
We analyse source code repositories to answer the following question: How and how much the Unsafe API is used in Java projects?

Our aim is to devise if the Unsafe API is used extensively so it is worth to create a new language to improve programmer productivity.

I. Introduction

sun.misc.Unsafe is an undocumented [1] class provided by Oracle. It allows the developer to access low-level programming features. It is the equivalent to `unsafe` [2] in C#.

There is a trend in the late years to use *sun.misc.Unsafe*. The main reason to use *sun.misc.Unsafe* is *performance*. Because *sun.misc.Unsafe* provides groups of methods to allow the programmer to access otherwise impossible low-level access.

Group	Methods
Array	arrayBaseOffset arrayIndexScale
CAS	compareAndSwapInt compareAndSwapLong compareAndSwapObject
Class	defineAnonymousClass defineClass ensureClassInitialized
Get	getBoolean getByte getChar getDouble getFloat getInt getIntVolatile getLoadAverage getLong getLongVolatile getObject getObjectVolatile getShort getBooleanVolatile getDoubleVolatile getFloatVolatile getByteVolatile getCharVolatile getShortVolatile
Memory	addressSize allocateMemory copyMemory freeMemory getAddress pageSize putAddress reallocateMemory setMemory
Offset	fieldOffset objectFieldOffset staticFieldBase staticFieldOffset
Park	park unpark
Put	putBoolean putByte putChar putDouble putFloat putInt putIntVolatile putLong putLongVolatile putObject putObjectVolatile putOrderedInt putOrderedLong putOrderedObject putShort putCharVolatile putOrderedInt putBooleanVolatile putShortVolatile putFloatVolatile putByteVolatile putDoubleVolatile
Single	allocateInstance throwException
Monitor	monitorEnter monitorExit tryMonitorEnter

TABLE I
FUNCTIONAL GROUPS OF *sun.misc.Unsafe*

Our research goal is to find

Trend using unsafe methods. But what for? Certainly you can do everithing without Unsafe API, so why using it? The key is performance.

We searched for Boa [1]

[1]http://www.oracle.com/technetwork/java/faq-sun-packages-142232.html
[2]http://msdn.microsoft.com/en-us/en-en/library/chfa2zb8(v=vs.90).aspx

discouraged.

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: This demo file is intended to serve as a "starter file" for IEEE conference papers produced under LaTeX using IEEEtran.cls version 1.7 and later.

Stackoverflow mining.

Measure error with boa.

B. Reflection

What happens with other uses such as reflection? It is not detected but it uses Unsafe. There should be a way to measure this kind of use.

Mine github, maybe parsing html.

Look for problematic uses of the API, and some use patterns.

Bugreport stackoverflow posts.

II. Related Work

rel work

GHTorrent [2] provide GitHub quering but for metadata, not source code mining, true?

III. Methodology

methodology

The complete API documentation and more extensive examples are available online [3].

Our Boa script starts

IV. Results

The Figure 2 shows the pie.

The figure 3 shows how many times a method is called. Grouped by functional group.

The most called is `objectFieldOffset`. Because the result is then used by many other calls to Unsafe.

V. Conclusions

The conclusion goes here.

And yes, it is used,

Acknowledgments

The first author was supported by Swiss National Science Foundation grant CRSII2_136225.

[3]https://bitbucket.org/acuarica/java-unsafe-analysis

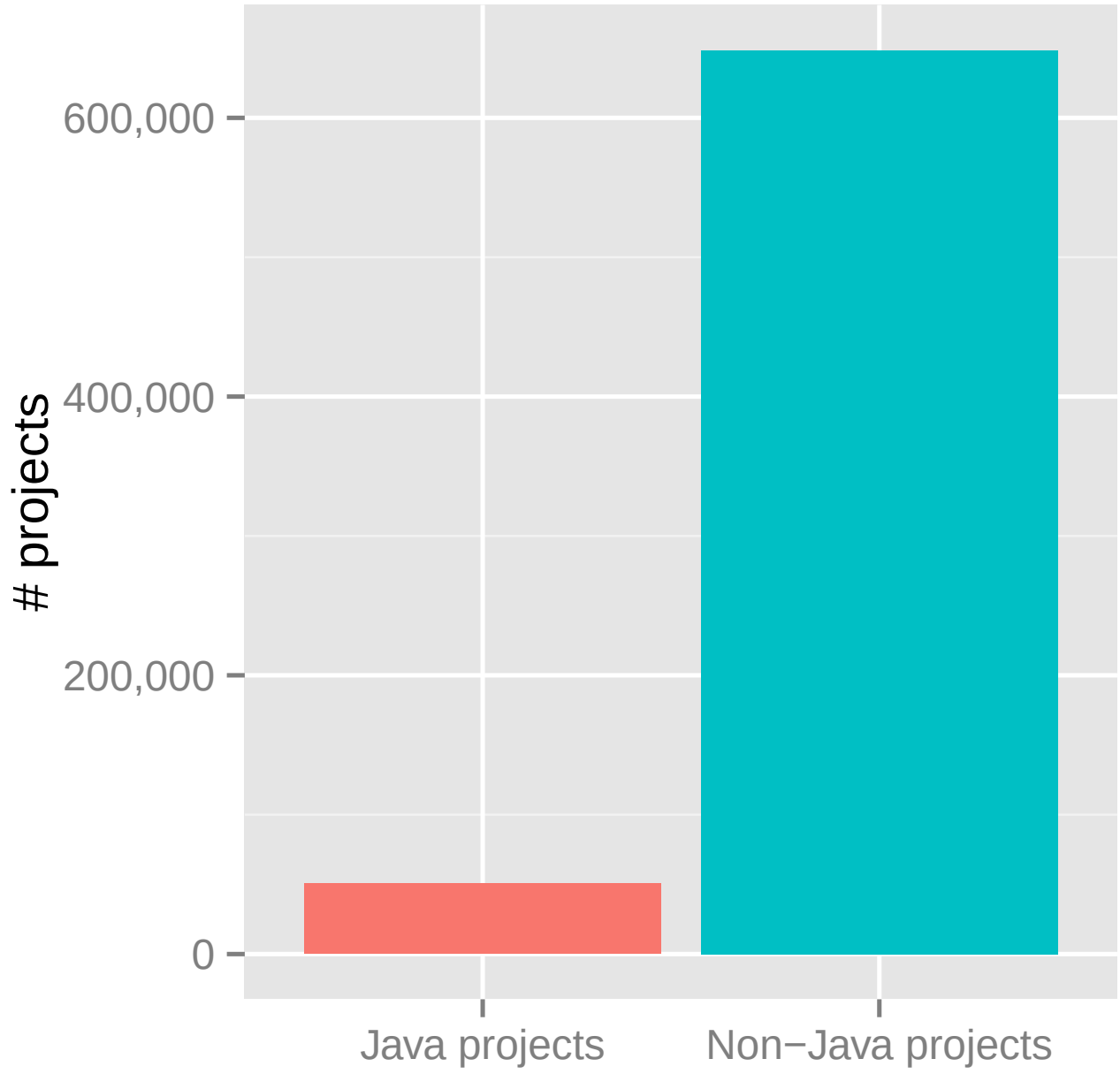

Fig. 1. Project using unsafe

REFERENCES

[1] R. Dyer, H. A. Nguyen, H. Rajan, and T. N. Nguyen, "Boa: A language and infrastructure for analyzing ultra-large-scale software repositories," in *Proceedings of the 35th International Conference on Software Engineering*, ser. ICSE'13, May 2013, pp. 422–431.

[2] G. Gousios, "The ghtorrent dataset and tool suite," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, ser. MSR '13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 233–236. [Online]. Available: http://dl.acm.org/citation.cfm?id=2487085.2487132

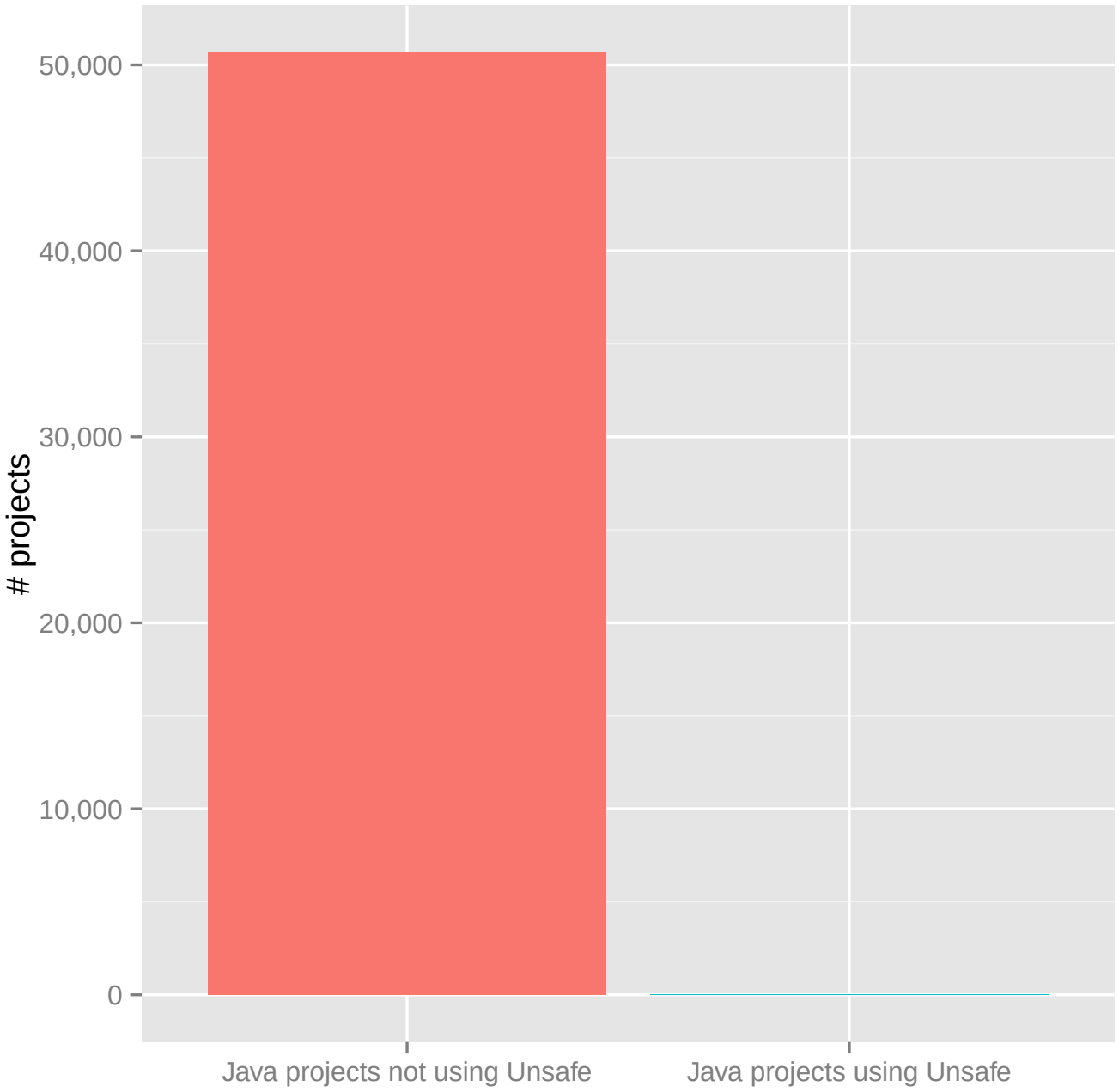

Fig. 2. Project using unsafe

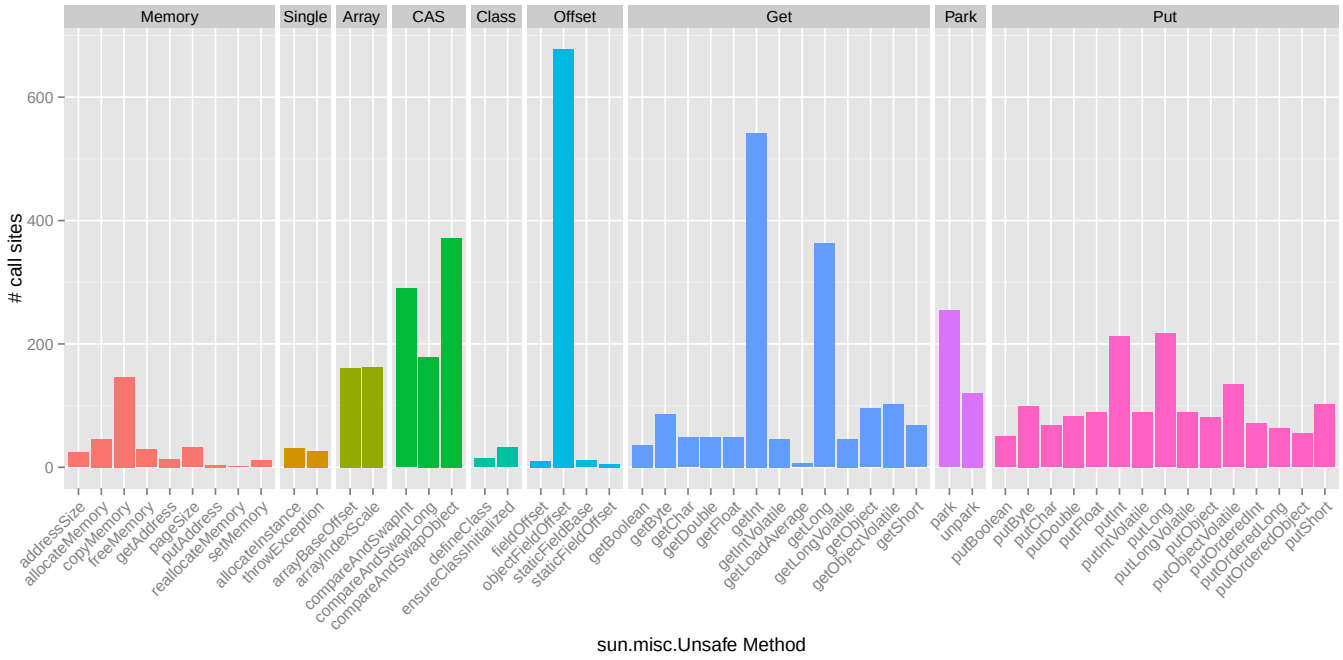

Fig. 3. sun.misc.Unsafe methods usage