

# 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 <sup>1</sup> class provided by Oracle. It allows the developer to access low-level programming features. It is the equivalent to `unsafe` <sup>2</sup> 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 methods to allow the programmer to access otherwise impossible low-level details. For instance, *sun.misc.Unsafe* contains methods to do CAS operations, the base ground to develop lock-free data structures.

<sup>3</sup>

Our research goal is to find Similar [1]

It is possible to group methods in *sun.misc.Unsafe* by functionality. Table I shows all methods (without overloads) grouped by functionality.

Trend using unsafe methods. But what for? Certainly you can do everything without Unsafe API, so why using it?

Stackoverflow mining.

Measure error with boa.

Bugreport stackoverflow posts.

Overall, the main contributions of this paper are two-fold:

- We present a detailed study of how the Java *sun.misc.Unsafe* API is used and
- We contrast this information on why this API is used based on responses from Stackoverflow.

The rest of this paper is organized as follows: Section II presents related work. Section III explains the methodology and technologies used to get our results. Section IV shows the results we obtained and Section V concludes.

## II. RELATED WORK

rel work

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

<sup>1</sup><http://www.oracle.com/technetwork/java/faq-sun-packages-142232.html>

<sup>2</sup>[http://msdn.microsoft.com/en-us/en-en/library/chfa2zb8\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/en-en/library/chfa2zb8(v=vs.90).aspx)

<sup>3</sup><http://cr.openjdk.java.net/~psandoz/dv14-uk-paul-sandoz-unsafe-the-situation.pdf>

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*

## III. METHODOLOGY

methodology

We searched for Boa [3]

discouraged.

The complete API documentation and more extensive examples are available online <sup>4</sup>.

Our Boa script starts

### A. 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.

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

## 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.

<sup>4</sup><https://bitbucket.org/acuarica/java-unsafe-analysis>

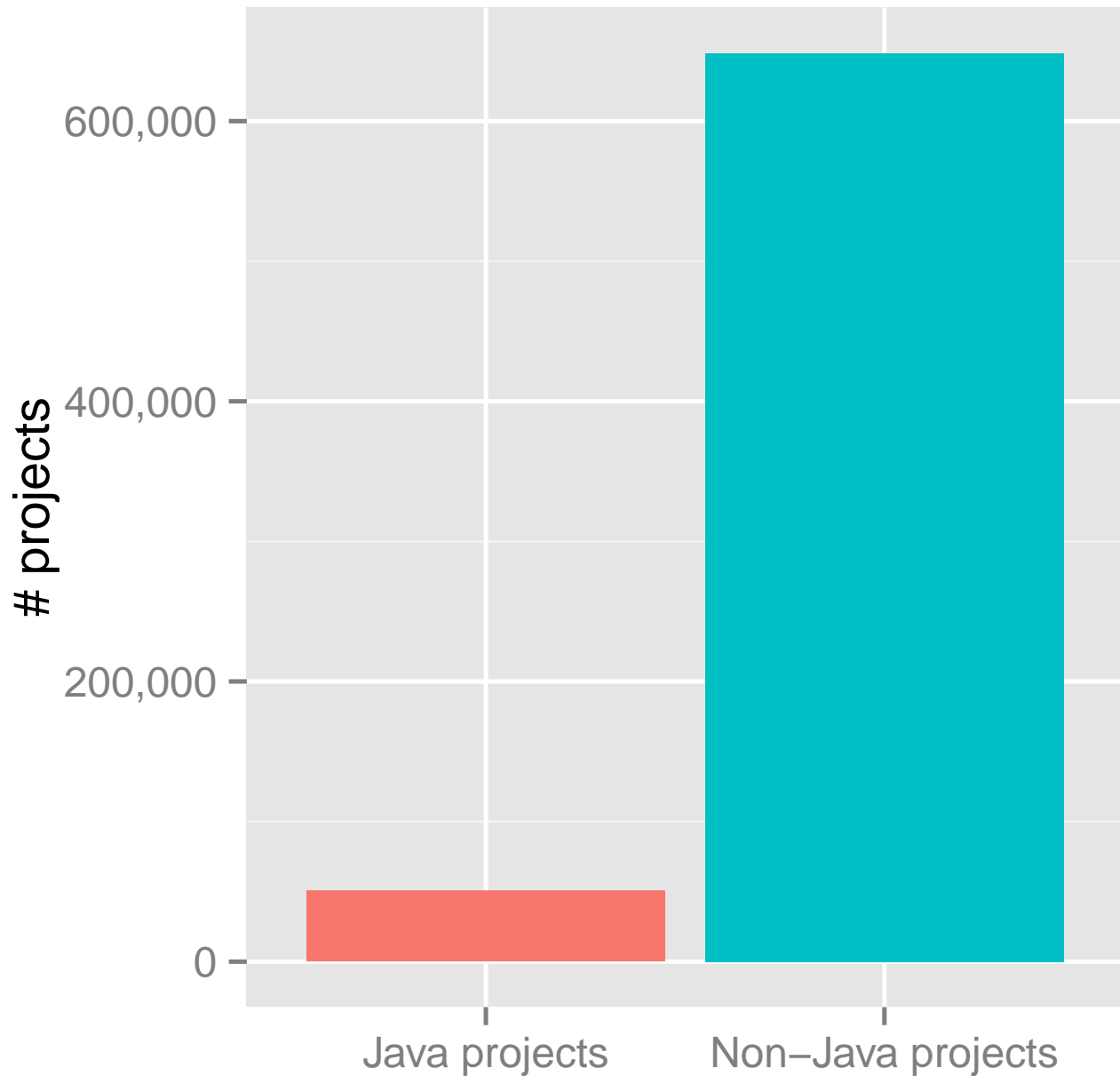


Fig. 1. Project using unsafe

## V. CONCLUSIONS

Although the current use of *sun.misc.Unsafe* seems low in SourceForge, it is important to notice the snapshot is from September 2013. It would be interesting to apply the same analysis but to the current GitHub source code database. Unfortunately at the moment we could not find any full dataset from GitHub.

We strongly believe that this study will help us to develop our language.

## ACKNOWLEDGMENTS

The first author was supported by Swiss National Science Foundation grant CRSII2\_136225.

## REFERENCES

- [1] R. Dyer, H. Rajan, H. A. Nguyen, and T. N. Nguyen, "Mining billions of AST nodes to study actual and potential usage of Java language features," in *36th International Conference on Software Engineering*, ser. ICSE'14, June 2014, pp. 779–790.
- [2] G. Gousios, "The ghortent 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>

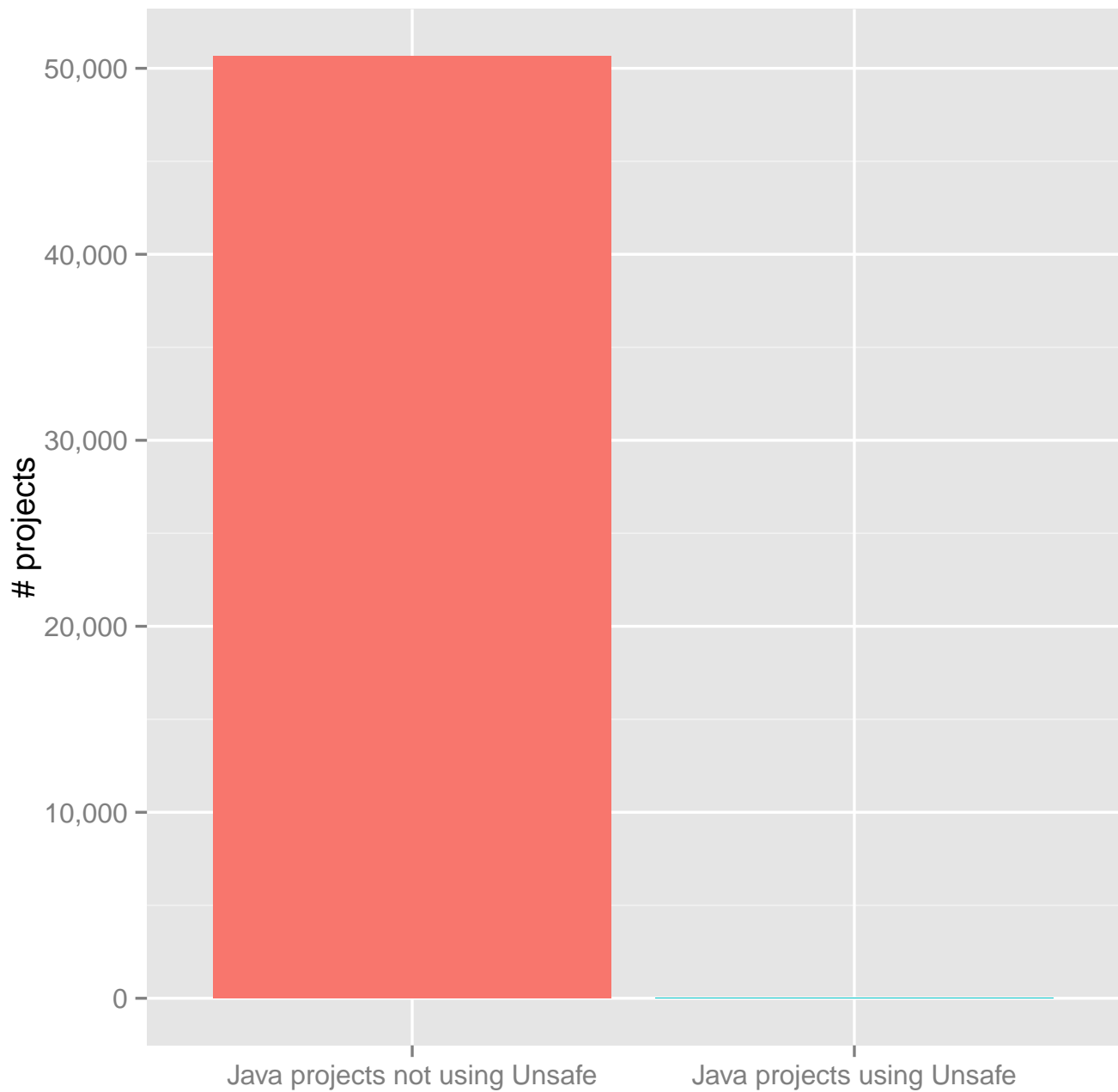


Fig. 2. Project using unsafe

- [3] 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.

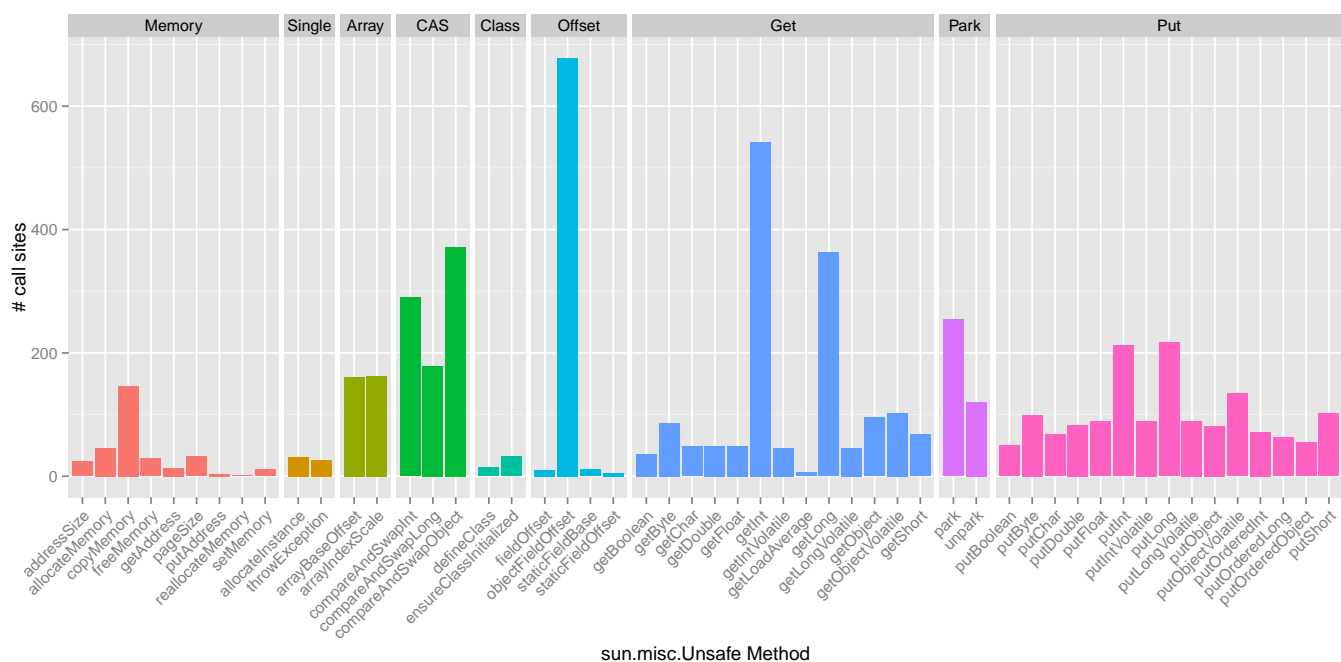


Fig. 3. `sun.misc.Unsafe` methods usage