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 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.

[3]

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 everithing 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 constrast 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?

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 [4].

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 1 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.

[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

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

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

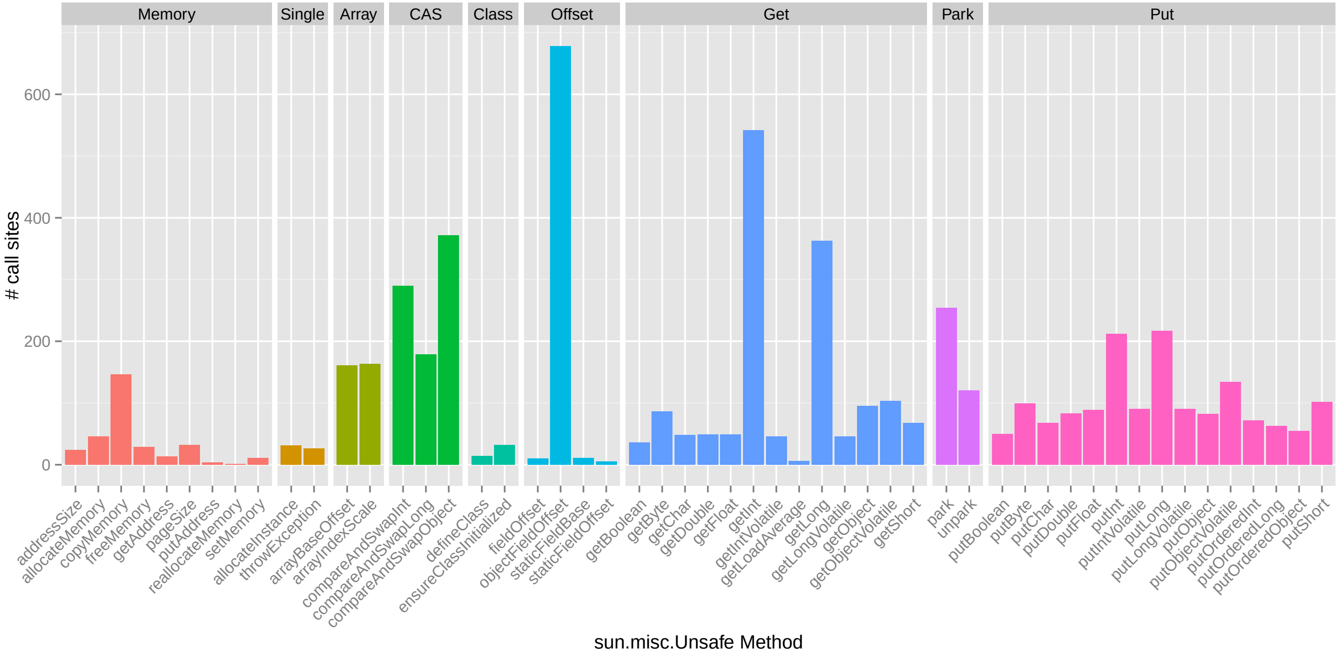

Fig. 3. sun.misc.Unsafe methods usage

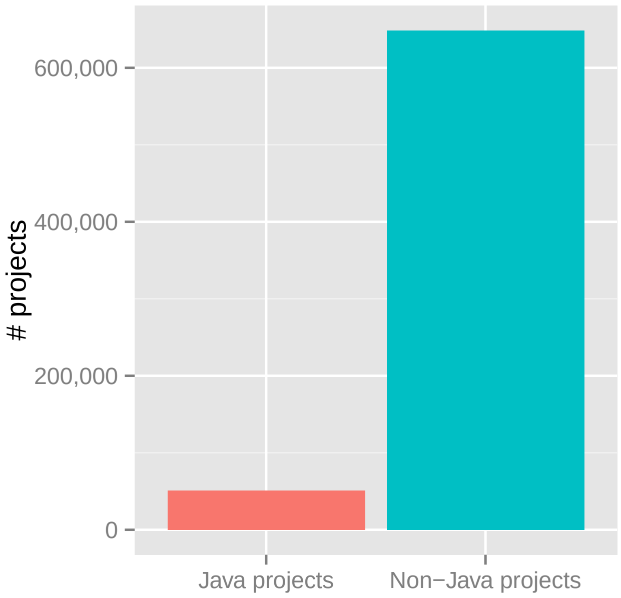

Fig. 1. # Java and non-Java projects

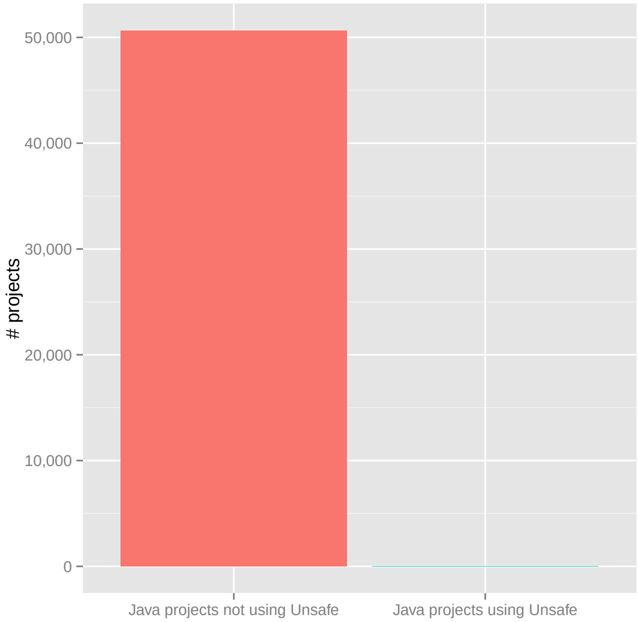

Fig. 2. Project using unsafe

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 GuitHub.

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

#	Name	Description
1	adtools	Amiga Development Tools (adtools)
2	amino	Concurrent Building Block
3	amock	Java Mock libarary for static method
4	android	Android on PXA270
5	aojunit	An aspect-oriented extension to JUnit
6	archaiosjava	Scalable and fast libraries for Java
7	beanlib	Java Bean Library
8	caloriecount	Track what you eat
9	cegcc	CeGCC - Cross development for Pocket PC
10	cgnu	CGNU (Clean GNU)
11	classreach	Identifies unused Java classes and methods
12	clipc	Library for IPC
13	concutest	Tools to test concurrent Java programs easier
14	ec	ec-gin Europe China Grid InterNetworking
15	essence	Essence Java Framework
16	essentialbudget	Essential Budget
17	glassbox	Troubleshooting and monitoring agent
18	grinder	Load testing framework
19	high	Highly Scalable Java
20	ikvm	JVM for .NET Framework and Mono
21	jadoth	abstraction utils and frameworks
22	janetdev	Ja.NET - Java Development Tools for .NET
23	janux	Java directly on the Linux Kernel
24	java	Lightweight Java Game Library
25	javapayload	Payloads to be used for post-exploitation
26	jaxlib	Platform independent Java library
27	jikesrvm	The Jikes Research Virtual Machine (RVM)
28	jnode	JNode: new Java Operating System
29	jon	Java Object Notation
30	jprovocateur	RAD for Ajax applications in Java
31	junitrecorder	Record test cases
32	katta	Lucene in the cloud
33	l2next	L2 Private Server code
34	neurogrid	P2P Bookmark Organiser
35	osfree	osFree operating system
36	ps2toolchain	Toolchain for the Playstation 2's
37	simulaeco	Semester project
38	snarej	Snare's Not A Risc os Emulator in Java
39	statewalker	Graph traversing library
40	takatuka	TakaTuka Java Virtual Machine
41	timelord	A tool for estimating and tracking time
42	vcb	Component Based Development tool
43	x10	Experimental language for DARPA/HPCS
44	xbeedriver	Driver for the ZigBee network

TABLE II
JAVA PROJECTS USING *sun.misc.Unsafe*

	id	name	description
1	ec	ec-gin	Europe China Gri
2	essentialbudget	Essential Budget	Essential Budget i
3	hlv	hlv	hlv is a collection
4	javapathfinder	JavaPathFinder	PLEASE NOTE -
5	javapayload	JavaPayload	A collection of pu
6	jaxlib	JaXLib	JaXLib is a platfo
7	jigcell	JigCell	Computational bi
8	jikesrvm	Jikes RVM	The Jikes Researc
9	jprovocateur	JProvocateur	RAD platform for
10	lockss	LOCKSS (Lots of Copies Keep Stuff Safe)	The LOCKSS(tm)
11	simulaeco	simulaeco	Trata-se de um tra
12	statewalker	statewalker	Graph traversing l
13	ucl	ucl-solrcloud	A final year proje
14	xbeedriver	XBeeDriver	This is a Java imp

TABLE III
PROJECTS USING LITERAL UNSAFE

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 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

[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.