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.

B. Stackoverflow

Google search for sun.misc.unsafe site:stackoverflow.com returns about 1,360 results.

IV. Results

The Figure 1 shows the pie.

[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

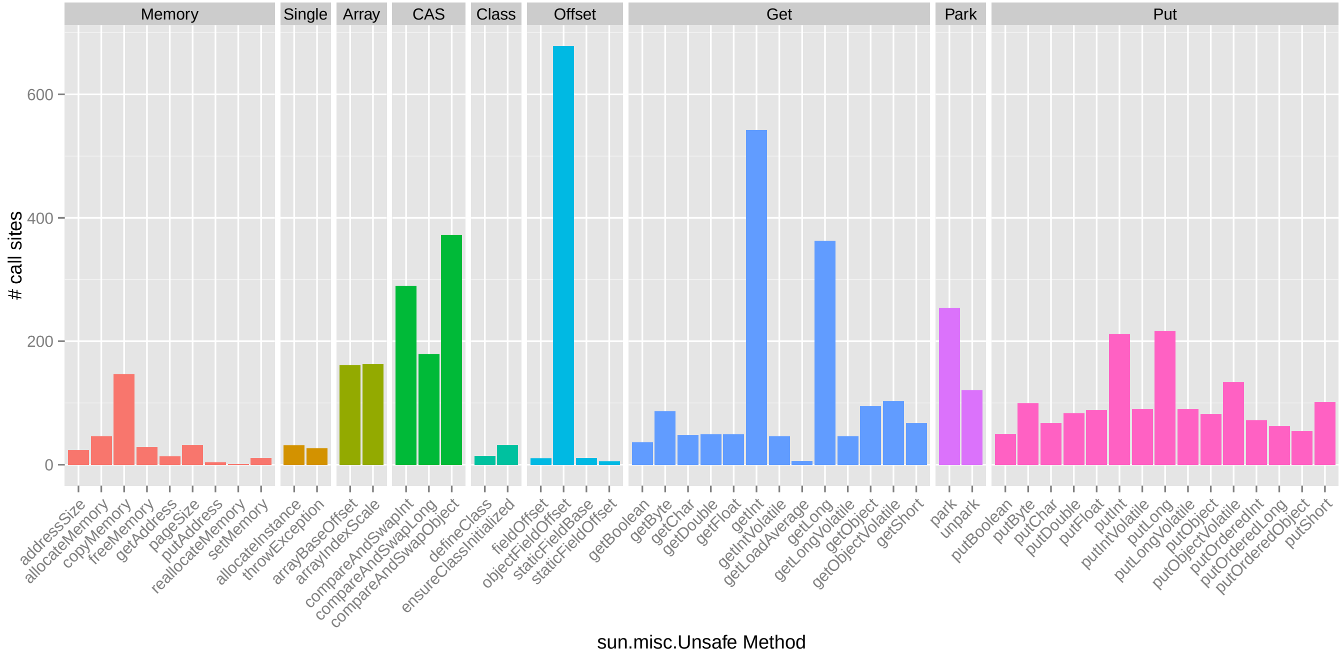

Fig. 3. sun.misc.Unsafe methods usage

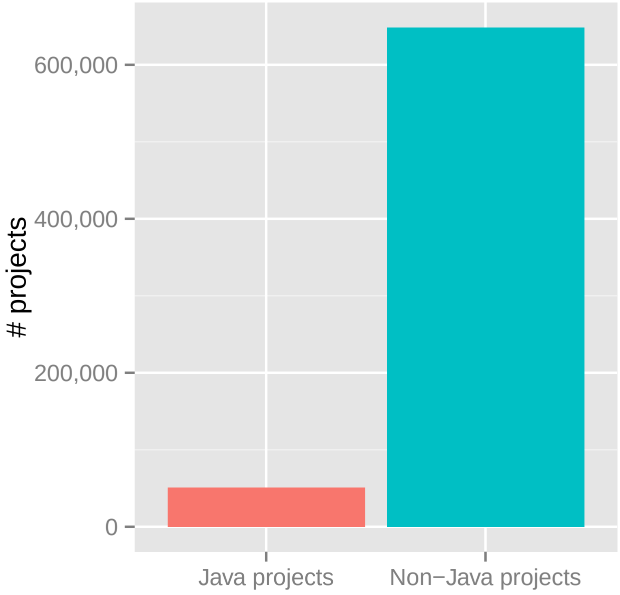

Fig. 1. # Java and non-Java projects

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.

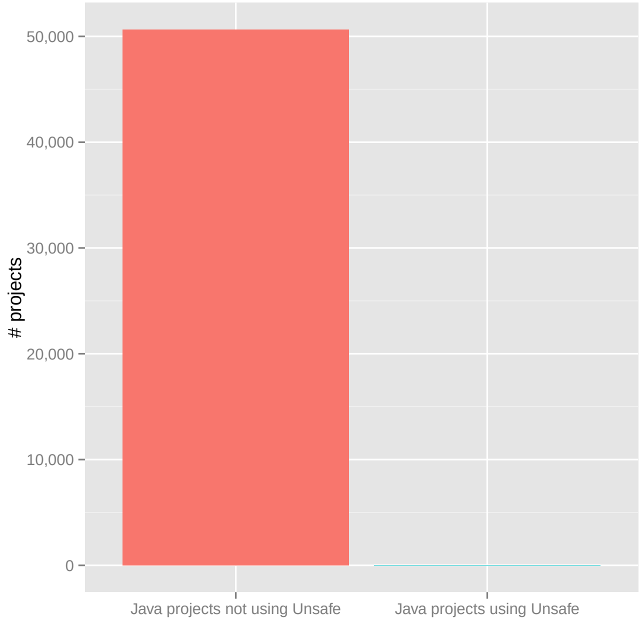

Fig. 2. Project using unsafe

A. Stackoverflow

Searching for the term: ”unsafe java” on stackoverflow returns 1,241 results. While searching for only the term ”sun.misc.unsafe” returns 318 results.

#	Name	Description	# Rev	# AST	Lifetime	# Call sites	# SMU literal
1	adtools	Amiga Development Tools (adtools)	466	14911 k	6 years	421	
2	amino	Concurrent Building Block	691	255 k	4 years	53	
3	amock	Java Mock libarary for static method	3	13 k	21 days	14	
4	android	Android on PXA270	146	4836 k	7 months	77	
5	aojunit	An aspect-oriented extension to JUnit	5	1 k	1 day	1	
6	archaiosjava	Scalable and fast libraries for Java	17	11 k	14 days	102	
7	beanlib	Java Bean Library	854	119 k	6 years	4	
8	caloriecount	Track what you eat	202	352 k	5 months	10	
9	cegcc	CeGCC - Cross development for Pocket PC	1449	2203 k	4 years	101	
10	cgnu	CGNU (Clean GNU)	60	2135 k	1 month	101	
11	classreach	Identifies unused Java classes and methods	69	196 k	2 years	10	
12	clipc	Library for IPC	278	228 k	6 months	10	
13	concutest	Tools to test concurrent Java programs	14	550 k	2 years	185	
14	ec	ec-gin Europe China Grid InterNetworking	9	635 k	2 month	10	1
15	essence	Essence Java Framework	293	157 k	2 years	75	
16	essentialbudget	Essential Budget	55	60 k	4 months	20	2
17	glassbox	Troubleshooting and monitoring agent	458	99 k	4 years	1	
18	grinder	Load testing framework	4334	770 k	11 years	6	
19	high	Highly Scalable Java	78	37 k	2 years	37	
20	hlv	Collection of high level view plugins for eclipse	278	33 k	7 months		4
21	ikvm	JVM for .NET Framework and Mono	3980	531 k	10 years	123	
22	jadoth	abstraction utils and frameworks	2922	619 k	3 years	949	
23	janetdev	Ja.NET - Java Development Tools for .NET	366	10034 k	2 years	280	
24	janux	Java directly on the Linux Kernel	25	564 k	1 month	10	
25	java	Lightweight Java Game Library	3841	571 k	11 years	6	
26	javapathfinder	Verifies Java bytecode programs	4038	9952 k	6 years		4
27	javapayload	Payloads to be used for post-exploitation	92	74 k	2 years	28	1
28	jaxlib	Platform independent Java library	3208	5405 k	11 years	42	3
29	jigcell	Computational biology problem solving	5286	3573 k	8 years		3
30	jikesrvm	The Jikes Research Virtual Machine (RVM)	16068	9026 k	10 years	32	16
31	jnode	JNode: new Java Operating System	11972	44401 k	10 years	2104	
32	jon	Java Object Notation	118	29 k	8 months	3	
33	jprovocateur	RAD for Ajax applications in Java	934	197 k	2 year	10	2
34	junitrecorder	Record test cases	18	34 k	3 months	1	
35	katta	Lucene in the cloud	478	169 k	1 year	31	
36	l2next	L2 Private Server code	22	39 k	1 month	26	
37	lockss	Lots of Copies Keep Stuff Safe	23048	11551 k	11 years		2
38	neurogrid	P2P Bookmark Organiser	738	337 k	5 years	2	
39	osfree	osFree operating system	1124	119 k	5 years	96	
40	ps2toolchain	Toolchain for the Playstation 2's	8	4298 k	1 day	202	
41	simulaeco	Semester project	66	136 k	4 months	10	2
42	snarej	Snare's Not A Risc os Emulator in Java	82	111 k	27 days	19	
43	statewalker	Graph traversing library	432	477 k	3 years	36	2
44	takatuka	TakaTuka Java Virtual Machine	2637	1176 k	3 years	107	
45	timelord	A tool for estimating and tracking time	546	697 k	2 year	40	
46	ucl	A final year project by UCL students	70	1639 k	3 months		1
47	vcb	Component Based Development tool	2446	602 k	3 years	11	
48	x10	Experimental language for DARPA/HPCS	25432	12292 k	9 years	279	
49	xbeedriver	Driver for the ZigBee network	6	119 k	3 days	10	2

TABLE II
JAVA PROJECTS USING *sun.misc.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.

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.