


10 (funny) years of Apache Lucene hacking

Uwe Schindler

Apache Software Foundation / MARUM, Uni Bremen /
SD DataSolutions GmbH

 thetaph1 – <https://www.thetaphi.de>

My Background

- **Committer** and **PMC member** of **Apache Lucene and Solr** - main focus is on development of Lucene Core.
- Implemented fast numerical search and maintaining the new attribute-based text analysis API. Well known as *Generics and Sophisticated Backwards Compatibility* .
- **Elasticsearch** lover.
- Working at **MARUM, University of Bremen**
- Working as consultant and software architect at **SD DataSolutions GmbH** in Bremen, Germany.
- Maintaining **PANGAEA** (Data Publisher for Earth & Environmental Science) where I implemented the portal's geo-spatial retrieval functions with Apache Lucene Core and Elasticsearch.

10 year Overview

Heavy Committing era

Finite State era

Hotspot era



era

令和 era



Berlin Buzzwords 2010/2011: The ~~movie~~ soundtrack of the ~~movie~~ development

Heavy committing!

Simon Willnauer says...



“once there was a huge flow of commit email messages coming in on the mailing list... **about 50 or so...**”

Backwards compatibility

Long history in Lucene development was strict backwards compatibility!

*On new releases, **tests from previous version** were forked into current dev tree, **compiled against old JAR** file and then **ran against new JAR** file!*

Deprecations? Generics?

Lucene 3.0 was first version “minimum Java 5”

- Generics added (that cool feature in Java 5)

Lucene 2.9 was identical to Lucene 3.0, just with Java 1.4 and without generics!

Deprecations? C

Lucene 3.0 was first version “

- Generics added (that cool

Lucene 2.9 was identical to Lucene 2.8.1
Java 1.4 and without generics

```
* LUCENE-1257, LUCENE-1984, LUCENE-1985, LUCENE-2057, LUCENE-1833, LUCENE-2012,
LUCENE-1998: Port to Java 1.5:

- Add generics to public and internal APIs (see below).
- Replace new Integer(int), new Double(double),... by static valueOf() calls.
- Replace for-loops with Iterator by foreach loops.
- Replace StringBuffer with StringBuilder.
- Replace o.a.l.util.Parameter by Java 5 enums (see below).
- Add @Override annotations.
(Uwe Schindler, Robert Muir, Karl Wettin, Paul Elschot, Kay Kay, Shai Erera,
DM Smith)

* Generify Lucene API:

- TokenStream/AttributeSource: Now addAttribute()/getAttribute() return an
instance of the requested attribute interface and no cast needed anymore
(LUCENE-1855).
- NumericRangeQuery, NumericRangeFilter, and FieldCacheRangeFilter
now have Integer, Long, Float, Double as type param (LUCENE-1857).
- Document.getFields() returns List<Fieldable>.
- Query.extractTerms(Set<Term>)
- CharArraySet and stop word sets in core/contrib
- PriorityQueue (LUCENE-1935)
- TopDocCollector
- DisjunctionMaxQuery (LUCENE-1984)
- MultiTermQueryWrapperFilter
- CloseableThreadLocal
- MapOfSets
- o.a.l.util.cache package
- lot's of internal APIs of IndexWriter
(Uwe Schindler, Michael Busch, Kay Kay, Robert Muir, Adriano Crestani)

* LUCENE-1944, LUCENE-1856, LUCENE-1957, LUCENE-1960, LUCENE-1961,
LUCENE-1968, LUCENE-1970, LUCENE-1946, LUCENE-1971, LUCENE-1975,
LUCENE-1972, LUCENE-1978, LUCENE-944, LUCENE-1979, LUCENE-1973, LUCENE-2011:
Remove deprecated methods/constructors/classes:

- Remove all String/File directory paths in IndexReader /
IndexSearcher / IndexWriter.
```


Token Attributes

- First try to get rid of `Token` class
- ...but in a backwards compatible way!

next

```
public Token next(Token reusableToken)
    throws IOException
```

Deprecated. *The new [incrementToken\(\)](#) and [AttributeSource](#) APIs should be used instead.*

Returns the next token in the stream, or null at EOS. When possible, the input Token should be used as the returned Token (this gives fastest tokenization performance), but this is not required and a new Token may be returned. Callers may re-use a single Token instance for successive calls to this method.

This implicitly defines a "contract" between consumers (callers of this method) and producers (implementations of this method that are the source for tokens):

- A consumer must fully consume the previously returned [Token](#) before calling this method again.
- A producer must call [Token.clear\(\)](#) before setting the fields in it and returning it

Also, the producer must make no assumptions about a [Token](#) after it has been returned: the caller may arbitrarily change it. If the producer needs to hold onto the [Token](#) for subsequent calls, it must clone() it before storing it. Note that a [TokenFilter](#) is considered a consumer.

Parameters:

`reusableToken` - a [Token](#) that may or may not be used to return; this parameter should never be null (the callee is not required to check for null before using it, but it is a good idea to assert that it is not null.)

Returns:

next [Token](#) in the stream or null if end-of-stream was hit

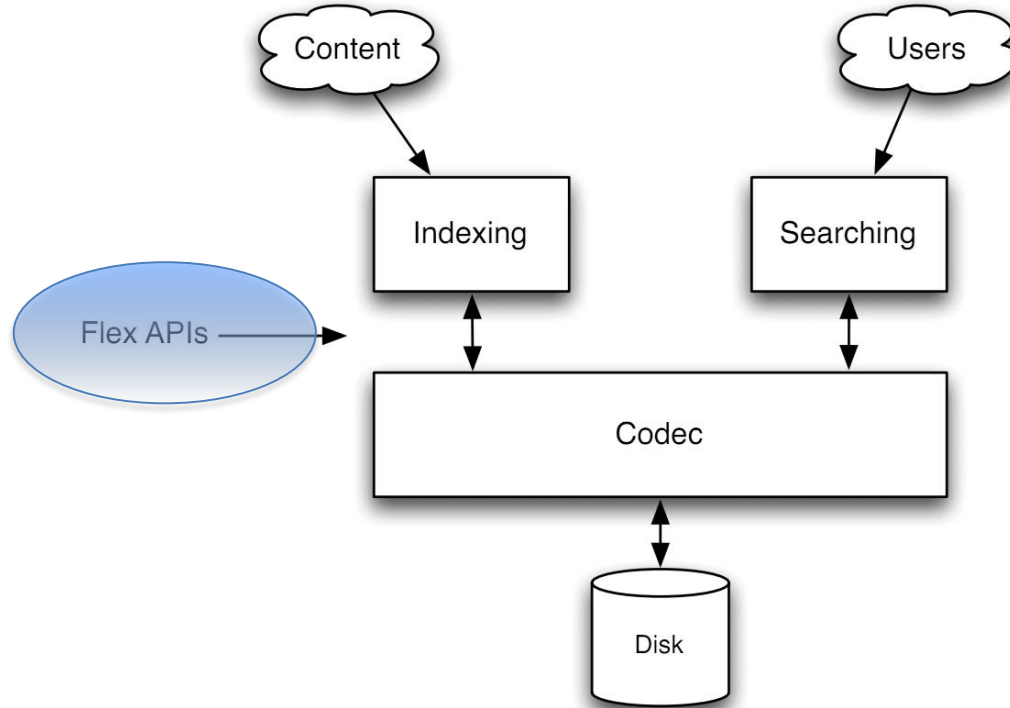
Throws:

[IOException](#)

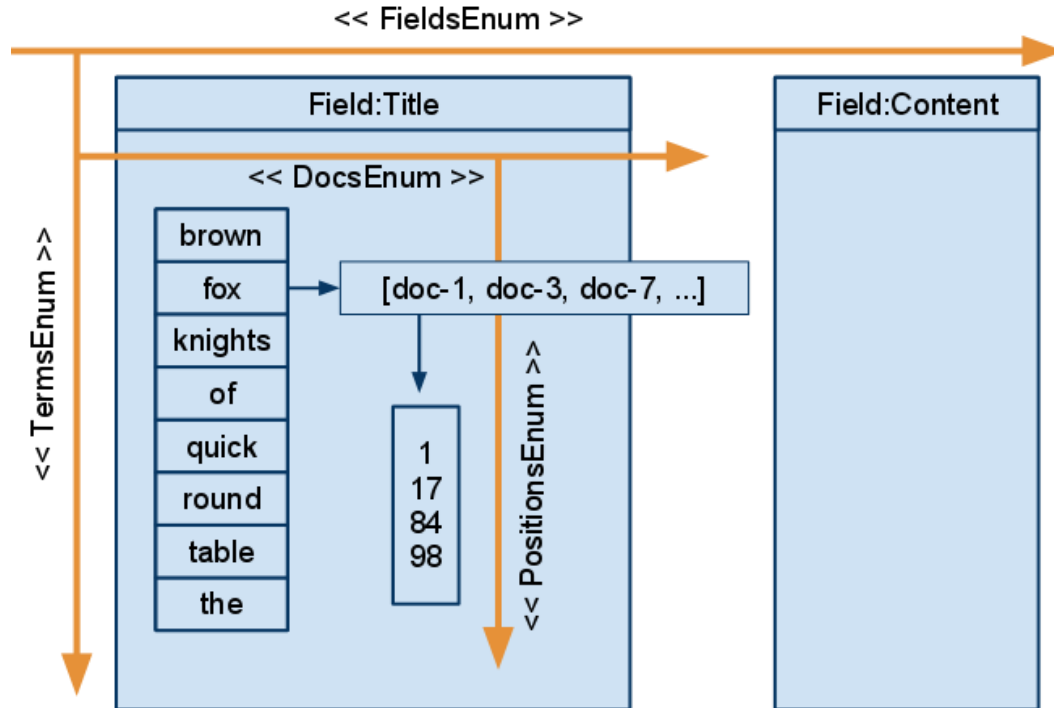
Lucene 4

- Start of development: 2010
- “Flexible indexing” (branch)
- Release: December 2012

Architecture



New 4-dimensional Enumeration-API

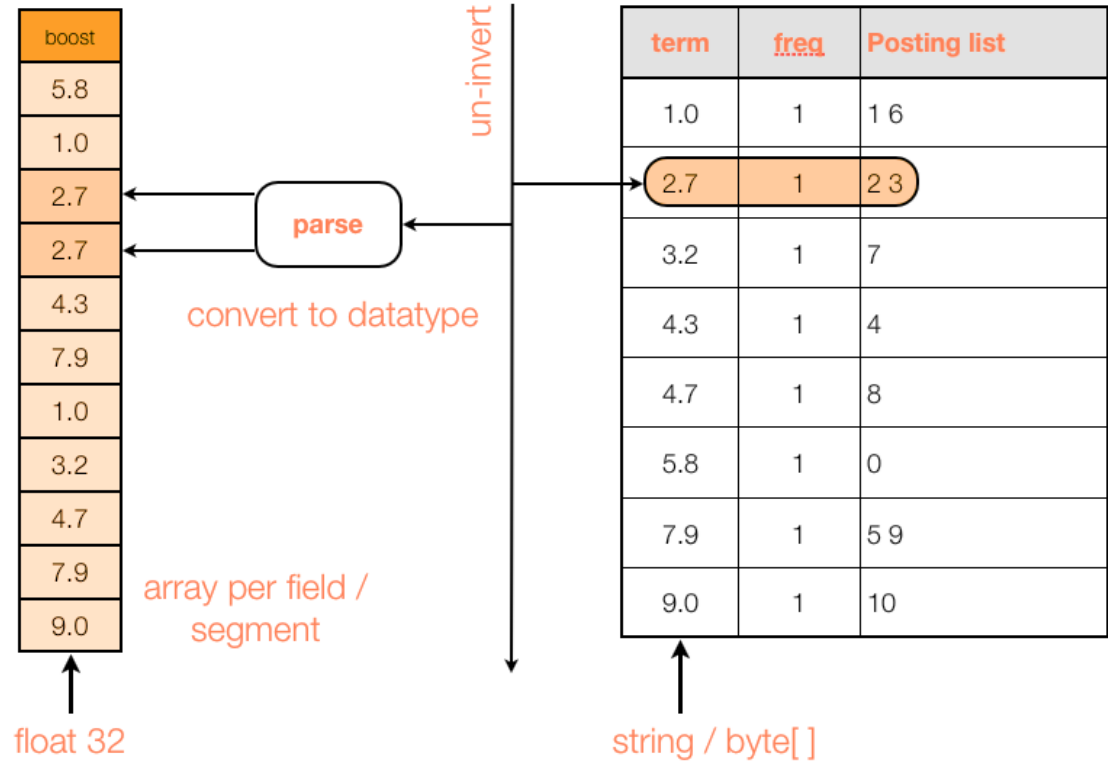


Fun moments

Surrogates dance

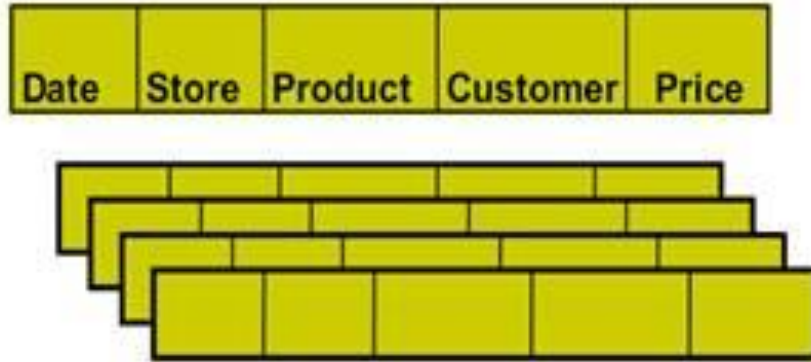


Die, FieldCache, die die die!

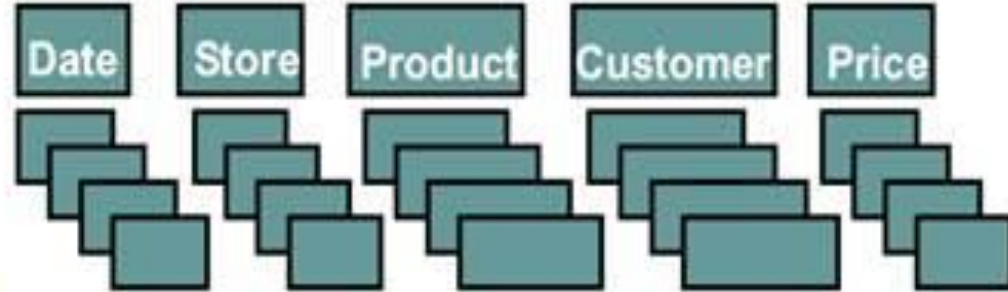


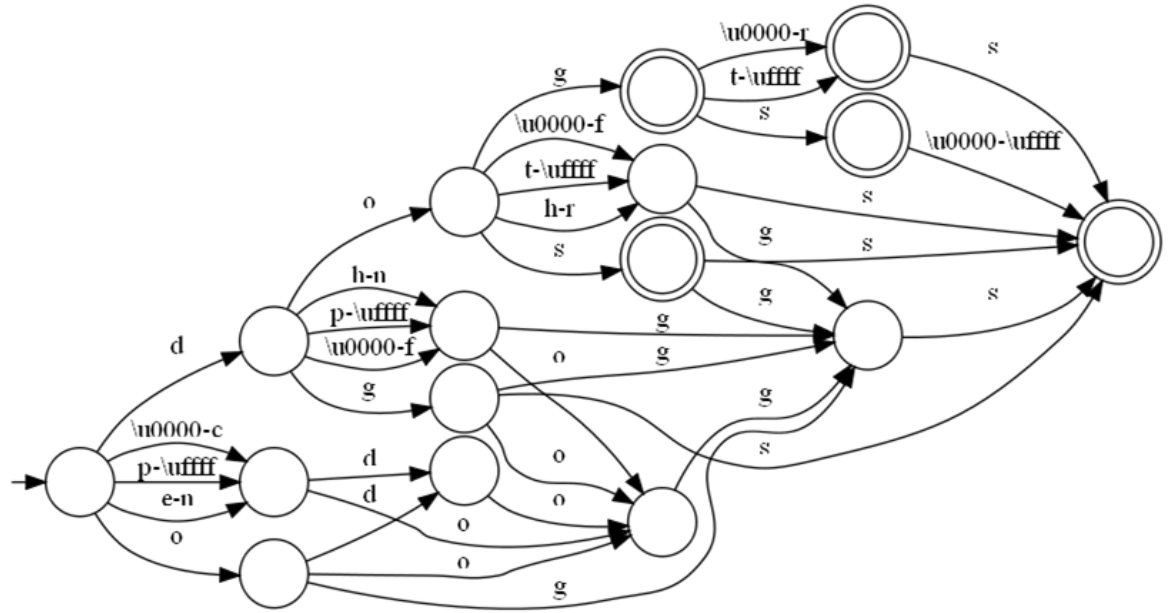
DocValues / Column Stride Fields

row-store



column-store





“Automata invasion”

Finite State Era

```

graph TD
    5((5)) --- 4((4))
    5 --- 5((5))
    5 --- 6((6))
    4 --- 42((42))
    4 --- 44((44))
    5 --- 52((52))
    6 --- 63((63))
    6 --- 64((64))
    42 --- 421((421))
    42 --- 423((423))
    44 --- 445((445))
    44 --- 446((446))
    44 --- 448((448))
    52 --- 521((521))
    52 --- 522((522))
    63 --- 632((632))
    63 --- 633((633))
    63 --- 634((634))
    64 --- 641((641))
    64 --- 642((642))
    64 --- 644((644))
    style 423 stroke-width:4px
    style 44 stroke-width:4px
    style 63 stroke-width:4px
    style 641 stroke-width:4px
    style 642 stroke-width:4px
  
```

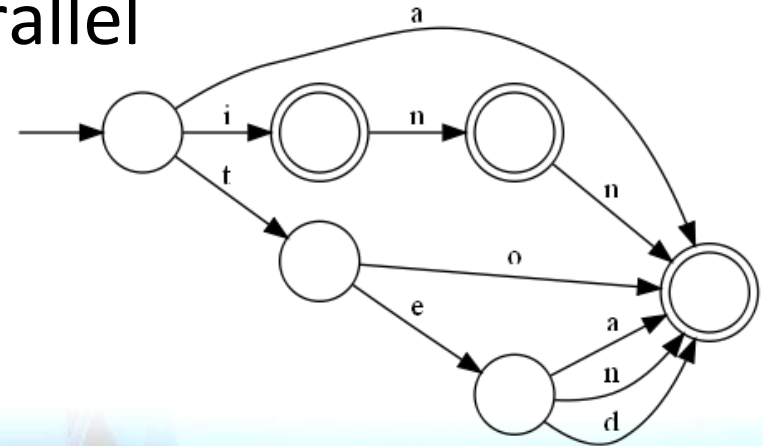
Regex, Wildcard, Fuzzy

- Without constant prefix, exhaustive
 - Regex: (http|ftp)://foo.com
 - Wildcard: ?oo?ar
 - Fuzzy: foobar~
- Re-implemented as automata queries
 - Just parsers that produce a DFA
 - Improved performance and scalability
 - (http|ftp)://foo.com examines 2 terms.



Automaton Queries

- Only explore subtrees that can lead to an accept state of some finite state machine.
- AutomatonQuery traverses the term dictionary and the state machine in parallel



Lucene's FST Implementation

Downside: Generics Policeman does **not** approve!



@UweSays "I looked at the code, it was
ununderstandable why this thing was generified"

10



Lucene's FST Implementation

Downside: Generics Policeman does **not** approve!



@UweSays "I looked at the code, it was
ununderstandable why this thing was generified"

10



How to crush Java?

Hotspot era

Remember 2011?

Chronology: Friday, July 29, 2011

22

Java 7 paralyzes Lucene and Solr

www.h-online.com/open/news/item/Java-7-paralyzes-Lucene

THE H
O P E N The H open source security In association with

Last 7 days News Archive Features Forums Newsletter RSS

29 July 2011, 12:58 < previous | next >

Java 7 paralyzes Lucene and Solr

The hotspot compiler in the recently [released](#) Java 7 has a defective optimiser that can cause flawed loops, according to a [warning](#) published by the Apache Software Foundation. As a result, the Java Virtual Machine can crash, and calculations can produce incorrect results.



A number of Apache projects are affected, including every published version of [Lucene](#) and [Solr](#). The Apache developers say that the indexing of documents on Solr causes Java to crash. Loops in Lucene can also be incorrectly compiled, thereby corrupting the indexes. In particular, the trunk version of Lucene with the [pulsing codec](#) is affected.

The bugs were discovered only five days before Java 7 was published; Oracle says it will correct them in the second service release of Java 7 [at the latest](#); the first update to Java 7 was reserved solely for security fixes, but the issue may prompt Oracle to change that plan. Until then though, users of Lucene and Solr should refrain from using the new version of Java or at least use the JVM option `-XX:-UseLoopPredicate` to disable the optimisation and prevent the index from being damaged.

The Apache developers say that users of Java 6 could also be affected. However, the flaws only occur in Java 6 when the JVM is used with the options `-XX:+OptimizeStringConcat` or `-XX:+AggressiveOpts` which activate normally disabled Hotspot optimisations.

Oracle has registered the flaws under [7070134](#), [7044738](#) and [7068051](#). The first one causes JVM to crash when Martin Porter's [stemmer algorithm](#) is used, which traces English words back to their stems. This flaw currently is of "low priority" while the others are "medium".

(djwm)

Chronology:
July 29, 2011

Java 7 paralyzes Lucene and Solr

www.h-online.com/open/news/item/Java-7-paralyses-Lucene-and-Solr

THE H
O P E N The H open source security In association with

Last 7 days News Archive Features Forums Newsletter RSS

29 July 2011, 12:58 < previous | next >

Java 7 paralyzes Lucene and Solr

The hotspot compiler in the recently [released](#) Java 7 has a defective optimiser that can cause flawed loops, according to a [warning](#) published by the Apache Software Foundation. As a result, the Java Virtual Machine can crash, and calculations can produce incorrect results.



A number of Apache projects are affected, including every published version of [Lucene](#) and [Solr](#). The Apache developers say that the indexing of documents on Solr causes Java to crash. Loops in Lucene can also be incorrectly compiled, thereby corrupting the indexes. In particular, the trunk version of Lucene with the [pulsing codec](#) is affected.

The bugs were discovered only five days before Java 7 was published; Oracle says it will correct them in the second service release of Java 7 [at the latest](#); the first update to Java 7 was reserved solely for security fixes, but the issue may prompt Oracle to change that plan. Until then though, users of Lucene and Solr should refrain from using the new version of Java or at least use the JVM option `-XX:-UseLoopPredicate` to disable the optimisation and prevent the index from being damaged.

The Apache developers say that users of Java 6 could also be affected. However, the flaws only occur in Java 6 when the JVM is used with the options `-XX:+OptimizeStringConcat` or `-XX:+AggressiveOpts` which activate normally disabled Hotspot optimisations.

Oracle has registered the flaws under [7070134](#), [7044738](#) and [7068051](#). The first one causes JVM to crash when Martin Porter's [stemmer algorithm](#) is used, which traces English words back to their stems. This flaw currently is of "low priority" while the others are "medium".

(djwm)

Java 7 Could Cause Bugs in

jaxenter.com/apache-warn-java-7-causes-bugs-in-some-apache-pr

jaxenter

NEWS VIDEOS BOOKS EVENTS JAVA TECH JOURNAL

HOME NEWS JAVA EDITORS PICK ECLIPSE ANDROID ARCHITECTURE CLOUD

Find out what's new in JBoss AS7 and OpenShift, in the new issue of Java Tech Journal!

July 29, 2011 RELATED NEWS BOOKS VIDEOS EVENTS

Apache Code Affected by Java 7

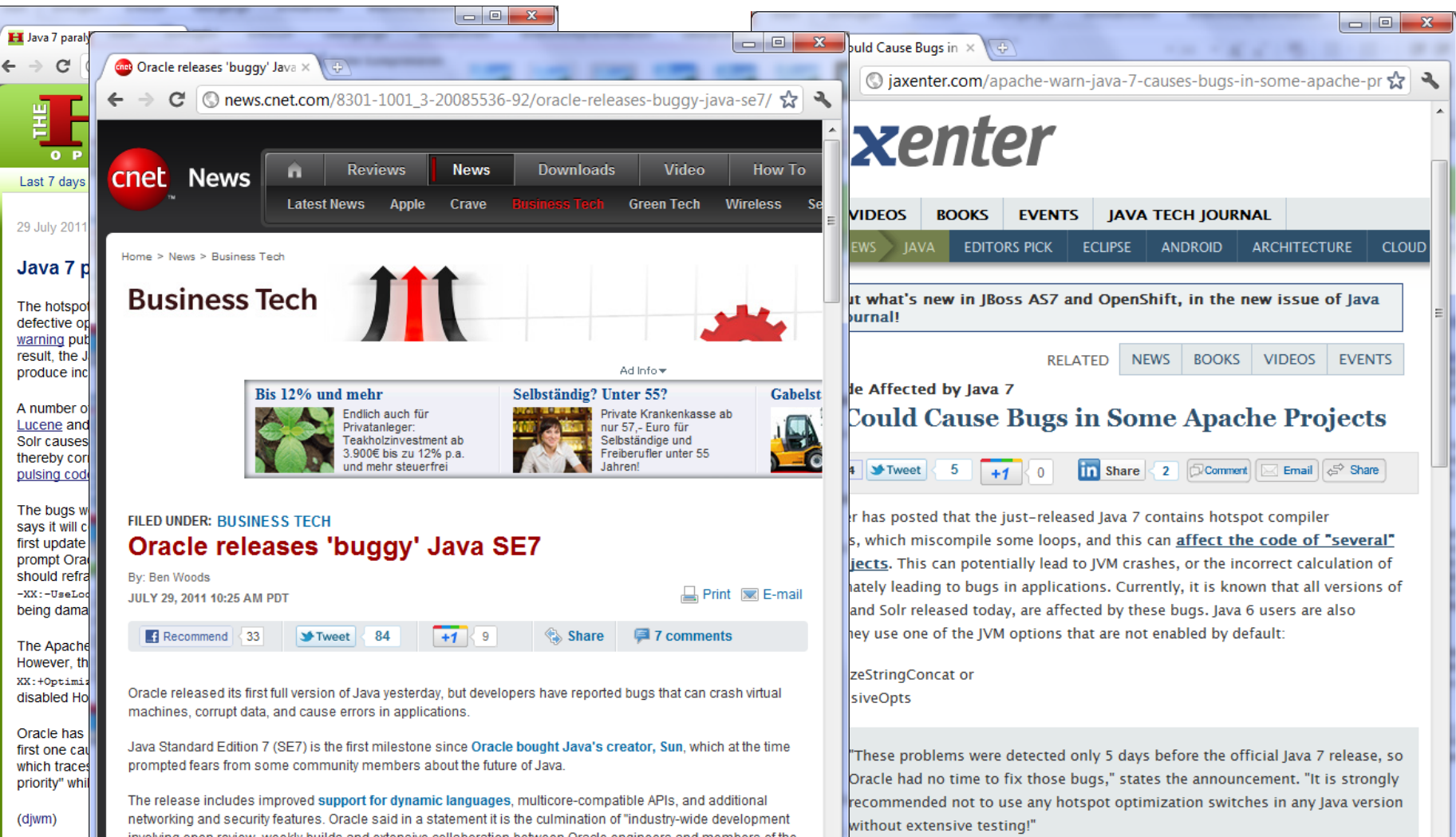
Java 7 Could Cause Bugs in Some Apache Projects

Share 4 Tweet 5 +1 0 in Share 2 Comment Email Share

Uwe Schindler has posted that the just-released Java 7 contains hotspot compiler optimisations, which miscompile some loops, and this can [affect the code of "several" Apache projects](#). This can potentially lead to JVM crashes, or the incorrect calculation of results, ultimately leading to bugs in applications. Currently, it is known that all versions of Lucene Core and Solr released today, are affected by these bugs. Java 6 users are also affected, if they use one of the JVM options that are not enabled by default:

-XX:+OptimizeStringConcat or
-XX:+AggressiveOpts

"These problems were detected only 5 days before the official Java 7 release, so Oracle had no time to fix those bugs," states the announcement. "It is strongly recommended not to use any hotspot optimization switches in any Java version without extensive testing!"



Home > News > Business Tech

Business Tech



Advertisement section with three items: 'Bis 12% und mehr', 'Selbständig? Unter 55?', and 'Gabelstapler'.

FILED UNDER: BUSINESS TECH

Oracle releases 'buggy' Java SE7

By: Ben Woods

JULY 29, 2011 10:25 AM PDT

Print E-mail

Social sharing buttons: Recommend, Tweet, +1, Share, 7 comments.

Oracle released its first full version of Java yesterday, but developers have reported bugs that can crash virtual machines, corrupt data, and cause errors in applications.

Java Standard Edition 7 (SE7) is the first milestone since Oracle bought Java's creator, Sun, which at the time prompted fears from some community members about the future of Java.

The release includes improved support for dynamic languages, multicore-compatible APIs, and additional networking and security features. Oracle said in a statement it is the culmination of "industry-wide development involving open review, weekly builds and extensive collaboration between Oracle engineers and members of the

What's new in JBoss AS7 and OpenShift, in the new issue of Java Journal!

RELATED NEWS BOOKS VIDEOS EVENTS

Affected by Java 7

Could Cause Bugs in Some Apache Projects

Social sharing buttons: Tweet, +1, 0, Share, Comment, Email, Share.

Someone has posted that the just-released Java 7 contains hotspot compiler bugs, which miscompile some loops, and this can affect the code of "several" projects. This can potentially lead to JVM crashes, or the incorrect calculation of dates leading to bugs in applications. Currently, it is known that all versions of Lucene and Solr released today, are affected by these bugs. Java 6 users are also affected if they use one of the JVM options that are not enabled by default:

StringConcat or
StringOptions

"These problems were detected only 5 days before the official Java 7 release, so Oracle had no time to fix those bugs," states the announcement. "It is strongly recommended not to use any hotspot optimization switches in any Java version without extensive testing!"

Home > News > Business Tech

Business Tech



Ad Info

Bis 12% und mehr



Endlich auch für
Privatanleger:
Teakholzinvestment ab
3.900€ bis zu 12% p.a.
und mehr steuerfrei

Selbständig? Unter 55?



Private Krankenkasse ab
nur 57,- Euro für
Selbständige und
Freiberufler unter 55
Jahren!

FILED UNDER: BUSINESS TECH

Oracle releases 'buggy' Java SE7

By: Ben Woods

JULY 29, 2011 10:25 AM PDT



f Recommend 33

t Tweet 84

+1 9

Share

7 comments

Oracle released its first full version of Java yesterday, but developers have reported bugs that can crash v
machines, corrupt data, and cause errors in applications.

Java Standard Edition 7 (SE7) is the first milestone since [Oracle bought Java's creator, Sun](#), which at th
prompted fears from some community members about the future of Java.

The release includes improved [support for dynamic languages](#), multicore-compatible APIs, and addition
networking and security features. Oracle said in a statement it is the culmination of "industry-wide develo
involving open review, weekly builds and extensive collaboration between Oracle engineers and member

InfoWorld

Sign in

CHANNELS

Application Development

Applications

Cloud Computing

Data Center



News

Blogs

Test Center

Technologies

Tech Watch

White Pa

[InfoWorld Home](#) / [InfoWorld Tech Watch](#) / Apache and Oracle warn of serious Java 7 compiler...


The First Word on Tech

INFOWORLD TECH WATCH

JULY 29, 2011

Apache and Oracle warn of serious Java 7 compiler bugs

The newly released Java upgrade suffers hotspot-compiler problems that affect Lucene and Solr

By [Ted Samson](#) | [InfoWorld](#)

Follow @tsamson_JW



4 Comments

Gefällt mir

It looks like a few bugs have crashed [Oracle's Java 7 release party](#) that can wreak havoc on Apache Project applications. The news likely will come as a disappointment to fans of Java, who've





PRODUCTS

SUPPORT & SERVICES

WHY LUCID?

BLOG

DEVZONE

DOWNLOADS

ABOUT US

Sign Up or Log In

Home . Blog

Search Lucene

Categories

apache
ApacheCon
Books
BoostingTermQuery
Droids
ecommerce
Enterprise Search
Events
functions
Hadoop
Libraries
Lucene
Lucene Connector Framework
Lucid Imagination
Lucid Imagination Solutions

LucidGaze
LucidWorks
Lucy
Mahout
ManifoldCF
NoSQL
nutch
Open Relevance

Don't Use Java 7, For Anything

July 28, 2011

Posted by *hossman*

Java 7 GA was **released today**, but as noted by Uwe Schindler, there are some very frightening bugs in HotSpot Loop optimizations that are enabled by default. In the best case scenario, these bugs cause the JVM to crash. In the worst case scenario, they cause incorrect execution of loops.

Bottom Line: **Don't use Java 7** for anything (unless maybe you know you don't have any loops in your java code)

*From: Uwe Schindler**Date: Thu, 28 Jul 2011 23:13:36 +0200**Subject: [WARNING] Index corruption and crashes in Apache Lucene Core / Apache Solr with Java 7*

*Hello Apache Lucene & Apache Solr users,
Hello users of other Java-based Apache projects,*

Oracle released Java 7 today. Unfortunately it contains hotspot compiler optimizations, which miscompile some loops. This can affect code of several Apache projects. Sometimes JVMs only crash, but in several cases, results calculated can be incorrect, leading to bugs in applications (see Hotspot bugs 7070134 [1], 7044738 [2], 7068051 [3]).

Apache Lucene Core and Apache Solr are two Apache projects, which are affected by these bugs, namely all versions released until today. Solr users

Recent Posts

- ▶ Multivalued geolocation fields in Solr
- ▶ Monitoring Apache Solr and LucidWorks with Zabbix
- ▶ Lucene in Barcelona, in Action
- ▶ SF Bay Lucene/Solr Meetup Attracts 100 Attendees (and a special appearance by Doug Cutting!)
- ▶ Announcing LucidWorks 2.0, the search platform for Apache Solr/Lucene
- ▶ Some more European Search in Action
- ▶ Lucene goes from Enterprise Search to search platform
- ▶ SF Bay Area Lucene/Solr Meetup: 9/22 6:30PM (<http://bit.ly/r19aZx>)
- ▶ Happy Anniversary, Lucene! 10 years at the ASF

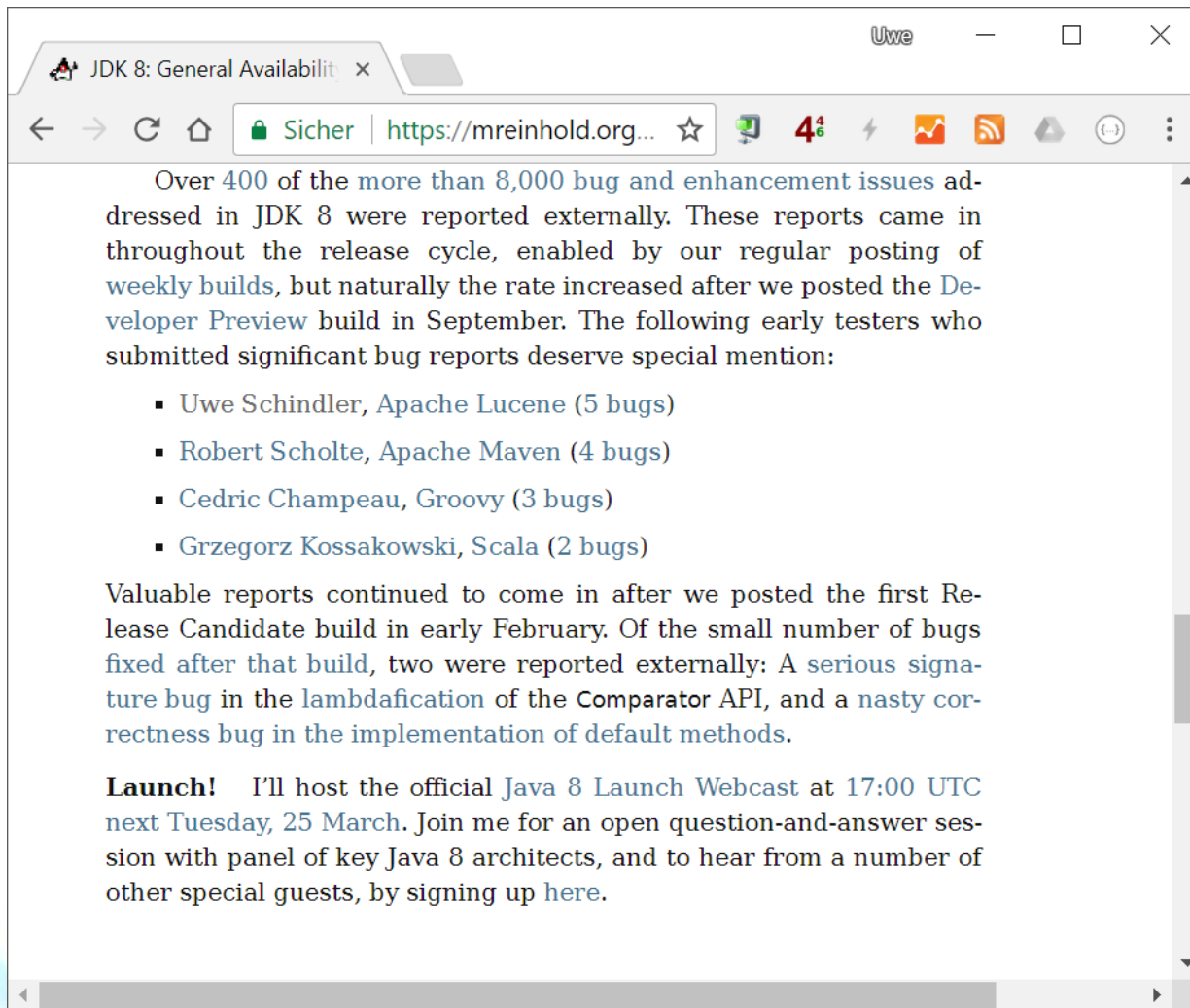
Reaction

Oracle (*Rory O'Donnell*) contacted Lucene PMC.

Weekly preview builds.

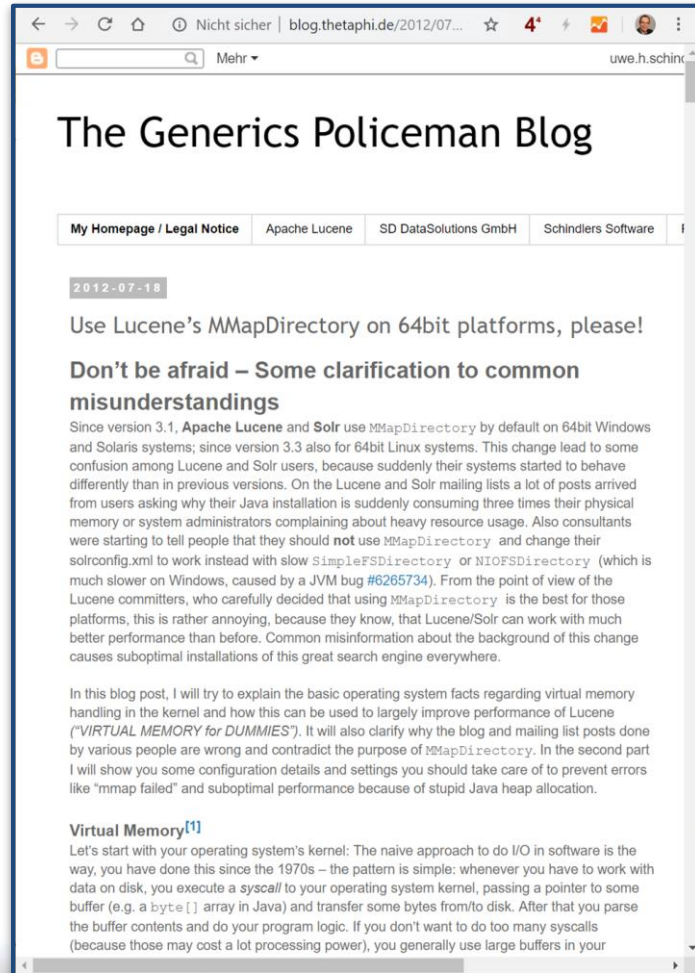
Other Open Source projects started to test with preview builds of Java 8 – and later **Java 9...**

Easy and fast bug reporting!



Java makes it hard!

Memory mapping the index



The screenshot shows a web browser window with the address bar displaying "blog.thetaphi.de/2012/07...". The page title is "The Generics Policeman Blog". Below the title is a navigation bar with links: "My Homepage / Legal Notice", "Apache Lucene", "SD DataSolutions GmbH", and "Schindlers Software". The main content area shows a date "2012-07-18" and a title "Use Lucene's MMapDirectory on 64bit platforms, please!". Below the title is a section "Don't be afraid – Some clarification to common misunderstandings". The text discusses the change in Lucene and Solr versions 3.1 and 3.3, where MMapDirectory is used by default on 64bit Windows and Solaris systems. It mentions that this change led to confusion among users, as their systems started to behave differently than in previous versions. The text also mentions that Lucene/Solr can work with much better performance than before, but common misinformation about the background of this change causes suboptimal installations of this great search engine everywhere. The text continues with "In this blog post, I will try to explain the basic operating system facts regarding virtual memory handling in the kernel and how this can be used to largely improve performance of Lucene ('VIRTUAL MEMORY for DUMMIES'). It will also clarify why the blog and mailing list posts done by various people are wrong and contradict the purpose of MMapDirectory. In the second part I will show you some configuration details and settings you should take care of to prevent errors like 'mmap failed' and suboptimal performance because of stupid Java heap allocation." Below this is a section "Virtual Memory^[1]" which starts with "Let's start with your operating system's kernel: The naive approach to do I/O in software is the way, you have done this since the 1970s – the pattern is simple: whenever you have to work with data on disk, you execute a `syscall` to your operating system kernel, passing a pointer to some buffer (e.g. a `byte[]` array in Java) and transfer some bytes from/to disk. After that you parse the buffer contents and do your program logic. If you don't want to do too many syscalls (because those may cost a lot processing power), you generally use large buffers in your".

Why?



Uwe Says


@UweSays

Folge ich

The garbage collector cannot see that our 30 bytes object instance "sits" on something like 300 Gigabytes of virtual memory and disk


JDK BUG SYSTEM

Dashboards ▾ More ▾ Search 🔍 ? Log In

 **JDK / JDK-4724038**
(fs) Add unmap method to MappedByteBuffer

Export ▾

Details

Type:  Enhancement

Status: **OPEN**

Priority:  P4

Resolution: Unresolved


Affects Version/s: 1.4.0, 5.0, 7

Fix Version/s: None

Component/s: [core-libs](#)

People

Assignee:  Andrew Haley

Reporter:  Nathanael Thompson
(Inactive)

Votes:

Interesting Detail

In Java 9,
`sun.misc.Unsafe`
got a new method:

```
void invokeCleaner (ByteBuffer b)
```

Testing, testing, testing + static code analysis



**BERLIN
BUZZWORDS**
2016 JUNE 05-07



Dawid Weiss: More Challenges for JVM!

**RANDOMIZE YOUR TESTS AND
IT WILL BLOW YOUR SOCKS
OFF! *)**

Randomization everywhere

- Input data, iteration counts, arguments.
 - Random, constraint-bound, shuffled
- Software components.
 - If multiple implementations exist: Field, Directory abstraction, IndexSearcher...
- Environment.
 - Locale, Timezone,...
 - *JVM (!), operating system*
- Exceptional triggers.
 - I/O problems, network problems (using mocks or runtime engineering)

RandomizedRunner's goals

Compatibility

with JUnit (and tools). At 99%, relax contracts when useful.

Built-in randomization

including reporting/ stack augmentations.

Test isolation

by tracking spawned threads. Timeouts. Terminations.

Utilities

@Repeat, @Seed, @Nightly, @TestGroup, @TestFactories...

```
if ("hiho".equals(  
    myString.toLowerCase()  
))
```

```
(GregorianCalendar) Calendar.getInstance()
```

```
    } catch (Exception e) {  
        // Eclipse autogenerated  
        e.printStackTrace();  
    }
```

```
new InputStreamReader(is) 32
```



```
if ("hiho".equals  
myString.toLowerCase()  
) )
```

```
(GregorianCalendar
```

```
} ca  
//  
e.  
}
```


new Input

A Cellphone's Missing Dot Kills Two People, Puts Three More in Jail


GIZMODO

Jesus Diaz
Filed to: LOCALIZATION PROBLEMS 4/21/08 10:05am

287,992 1 ★



The life of 20-year-old Emine, and her 24-year-old husband Ramazan Çalçoban was pretty much the normal life of any couple in a separation process. After deciding to split up, the two kept having bitter arguments over the cellphone, sending text messages to each other until one day Ramazan wrote "you change the topic every time you run out of arguments." That day, the lack of a single dot over a letter—product of a faulty localization of the cellphone's typing system—caused a chain of events that ended in a violent blood bath (Warning: offensive language ahead.)



The surreal mistake happened because Ramazan's sent a message and Emine's cellphone didn't have an specific character from the Turkish alphabet: the letter "ı" or closed i. While "i" is available in all phones in Turkey—where this happened—the closed i apparently doesn't exist in most of the terminals in that country.

```
if ("hiho".equals(  
    myString.toLowerCase()  
))
```

```
(GregorianCalendar) Calendar.getInstance()
```

```
    } catch (Exception e) {  
        // Eclipse autogenerated  
        e.printStackTrace();  
    }
```

```
new InputStreamReader(is) 32
```

```
if ("hiho".equals(
```

```
mySt
```

```
))
```

```
(Gre
```

```
ar.getInstance()
```

```
der(is)
```

32



Forbidden-APIs

- Keep Lucene free of „unsafe“ APIs:
 - Locale sensitive calls using system default
 - Use of platform's default charset
 - Same applies to timezones and default use of non-Gregorian calendars (e.g., Thai).
- Other no-gos:
 - Printing to System.out/err
 - Creating threads without name

Forbidden-APIs

- Used by other projects, too:
 - **Elasticsearch** (with Gradle)
- Available through Maven Central
 - Ant plugin, Maven Mojo, Gradle plugin
 - Various signature files included: “unsafe”, JDK deprecated methods, `System.out`

Policeman's Forbidden Apis

[**https://github.com/policeman-tools/forbidden-apis**](https://github.com/policeman-tools/forbidden-apis)



24/7 randomized testing using many JVMs (-settings)

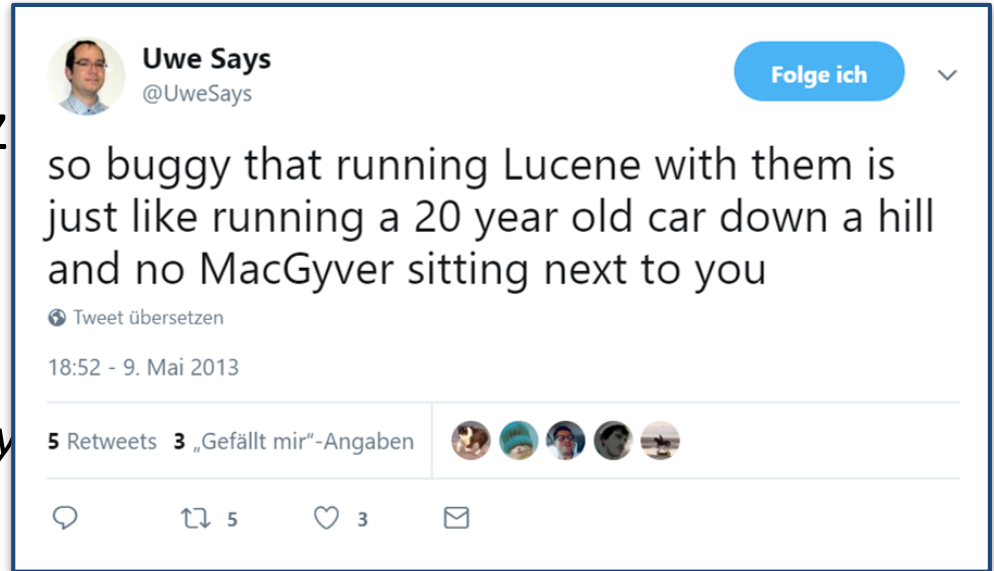
POLICEMAN JENKINS

Missing parts

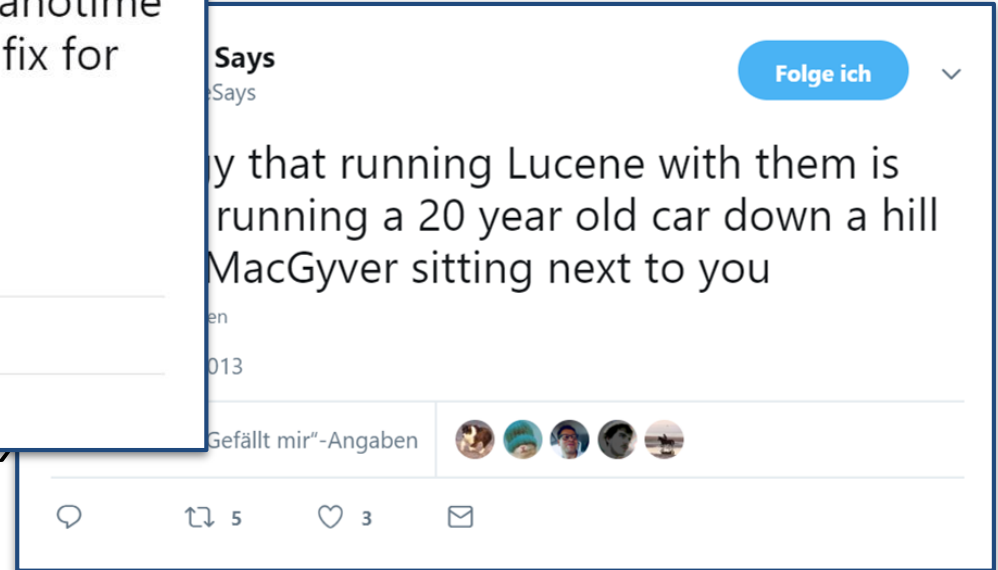
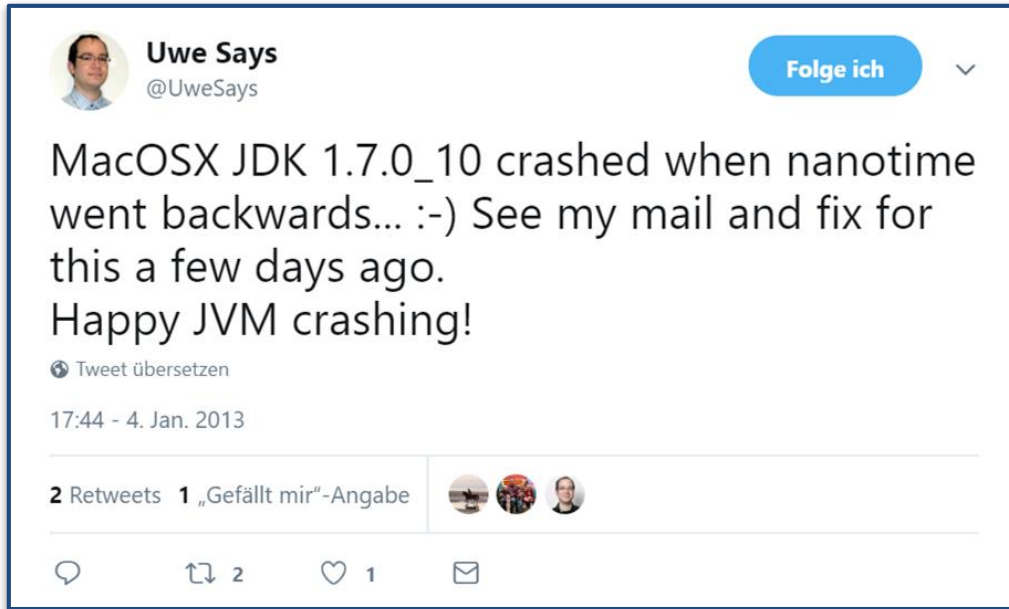
- JVM randomization
- JVM settings randomization
 - Garbage collector
 - Bitness: 32 / 64 bits
 - Server / Client VM
 - Compressed OOPs (*ordinary object pointer*)
- Platform
 - Linux, Windows, MacOS X, Solaris

Missing parts

- JVM randomization
- JVM settings randomiz
 - Garbage collector
 - Bitness: 32 / 64 bits
 - Server / Client VM
 - Compressed OOPs (*ordinary*)
- Platform
 - Linux, Windows, MacOS X, Solaris



parts



- Platform
 - Linux, Windows, MacOS X, Solaris



PANGAEA.
 Data Publisher for Earth & Environmental Science

- Zurück zur Übersicht
- Status
- Änderungen
- Arbeitsbereich
- GroovyScriptTrigger Log
- Java Warnungen
- Embeddable Build Status

Build-Verlauf

Trend

suchen x

- #24234
15.06.2019 15:31
Java: 64bit/jdk-13-ea+shipilev-fastdebug -
XX:+UseCompressedOops -
XX:+UseConcMarkSweepGC
- #24233
15.06.2019 11:31
Java: 64bit/jdk-11.0.2 -XX:-UseCompressedOops -
XX:+UseG1GC
- #24232
15.06.2019 06:31
Java: 64bit/jdk-13-ea+shipilev-fastdebug -
XX:+UseCompressedOops -XX:+UseSerialGC
- #24231

Projekt Lucene-Solr-master-Linux

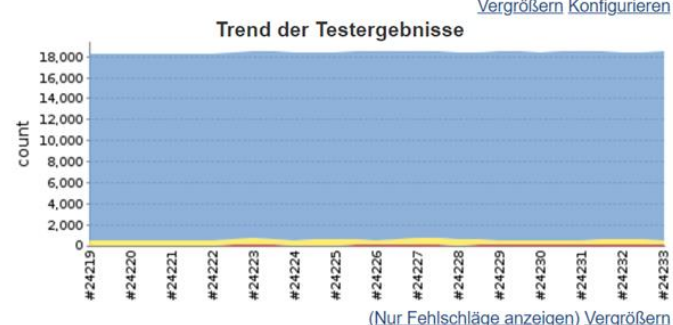
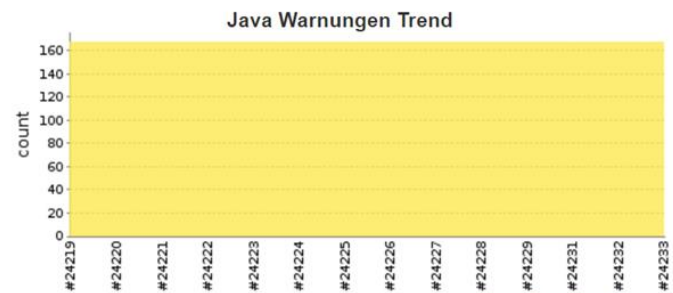
- Arbeitsbereich
- Letzte Änderungen
- Letztes Testergebnis (1 fehlgeschlagener Test / ±0)

Nachgelagerte Projekte

- Lucene-Solr-7.x-Linux
- Lucene-Solr-8.1-Linux
- Lucene-Solr-8.x-Linux

Permalinks

- Letzter Build (#24234), vor 1 Stunde 10 Minuten
- Letzter stabiler Build (#24228), vor 1 Tag 4 Stunden
- Letzter erfolgreicher Build (#24233), vor 5 Stunden 10 Minuten
- Letzter fehlgeschlagener Build (#24230), vor 20 Stunden
- Letzter instabiler Build (#24233), vor 5 Stunden 10 Minuten
- Letzter erfolgloser Build (#24233), vor 5 Stunden 10 Minuten
- Neuester abgeschlossener Build (#24233), vor 5 Stunden 10 Minuten



(Nur Fehlschläge anzeigen) Vergrößern

Status

 Änderungen

 Arbeitsbereich

 GroovyScriptTrigger Log

Java Warnungen

Embeddable Build Status

Arbeitsbereich

 Letzte Änderungen

 Letztes Testergebnis (1 fehlgeschlagener Test / ±0)

Nachgelagerte Projekte

- [Lucene-Solr-7.x-Linux](#)

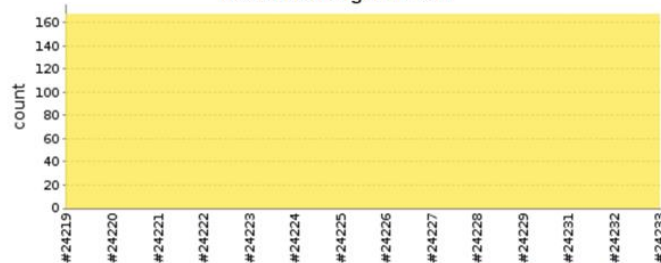
- Lucene-Solr-8.1-Linux

- Lucene-Solr-8.x-Linux

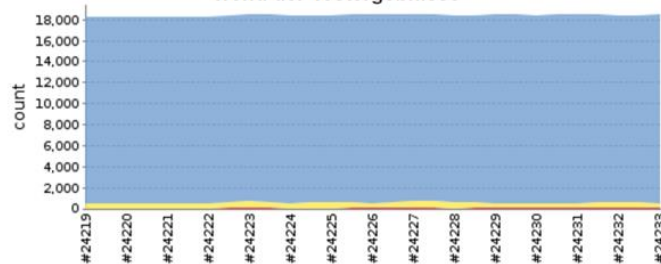
Permalinks

- Letzter Build (#24234), vor 1 Stunde 10 Minuten
- Letzter stabiler Build (#24228), vor 1 Tag 4 Stunden
- Letzter erfolgreicher Build (#24233), vor 5 Stunden 10 Minuten
- Letzter fehlgeschlagener Build (#24230), vor 20 Stunden
- Letzter instabiler Build (#24233), vor 5 Stunden 10 Minuten
- Letzter erfolgloser Build (#24233), vor 5 Stunden 10 Minuten
- Neuester abgeschlossener Build (#24233), vor 5 Stunden 10 Minuten

Java Warnungen Trend

[Vergrößern Konfigurieren](#)

Trend der Testergebnisse



(Nur Fehlschläge anzeigen) Vergrößern

☀ Build-Verlauf

Trend =

suchen

#24234

15.06.2019 15:31

```
Java: 64bit/jdk-13-ea+shipilev-fastdebug -
XX:+UseCompressedOops -
XX:+UseConcMarkSweepGC
```

#24233

15.06.2019 11:31

```
Java: 64bit/jdk-11.0.2 -XX:-UseCompressedOops -XX:+UseG1GC
```

#24232

15.06.2019 06:31

```
Java: 64bit/jdk-13-ea+shipilev-fastdebug -
XX:+UseCompressedOops -XX:+UseSerialGC
```

#24231

Lucene-Solr-master-Linux [Jenkins x]

https://jenkins.thetaphi.de/view/Lucene-Solr/job/Lucene-Solr-master-Linux/

Suchen

Anmelden

Jenkins > Lucene-Solr > Lucene-Solr-master-Linux

AUTO-AKTUALISIERUNG EINSCHALTEN

ScriptTrigger

Groovy Expression evaluation to true. (10s)

[EnvInject] - Loading node environment variables.

[EnvInject] - Preparing an environment for the build.

[EnvInject] - Keeping Jenkins system variables.

[EnvInject] - Keeping Jenkins build variables.

[EnvInject] - Evaluation of Groovy script content:

return evaluate(new java.io.File(JENKINS_HOME, "scripts/linux-random-java8.groovy"))

Using Java: 32bit/jdk-9-ea+102 -server -XX:+UseG1GC -XX:-CompactStrings

[EnvInject] - Injecting contributions.

Building on master in workspace /var/lib/jenkins/workspace/Lucene-Solr-trunk-Linux

Fetching changes from the remote Git repository

Cleaning workspace

Checking out Revision 9332b1602cc0f7312fc22a3d088c549299015691 (refs/remotes/origin/master)

No emails were triggered.

[description-setter] Description set: Java: 32bit/jdk-9-ea+102 -server -XX:+UseG1GC -XX:-CompactStrings

[Lucene-Solr-trunk-Linux] \$ /var/lib/jenkins/tools/hudson.tasks.Ant_AntInstallation/ANT_1.8.2/bin/ant "-Dargs=-server -XX:+UseG1GC -XX:-CompactStrings" jenkins-hourly

Buildfile: /home/jenkins/workspace/Lucene-Solr-trunk-Linux/build.xml

jenkins-hourly:

-print-java-info:

[java-info] java version "9-ea"

[java-info] Java(TM) SE Runtime Environment (9-ea+102-2016-01-21-001243.javare.4316.nc, Oracle Corporation)

[java-info] Java HotSpot(TM) Server VM (9-ea+102-2016-01-21-001243.javare.4316.nc, Oracle Corporation)

[java-info] Test args: [-server -XX:+UseG1GC -XX:-CompactStrings]

clean:

#24232

15.06.2019 06:31

Java: 64bit/jdk-13-ea+shipilev-fastdebug -XX:+UseCompressedOops -XX:+UseSerialGC

• [Letzter erfolgreicher Build \(#24233\), vor 6 Stunden 10 Minuten](#)

• [Neuester abgeschlossener Build \(#24233\), vor 5 Stunden 10 Minuten](#)

(Nur Fehlschläge anzeigen) Vergrößern

GAEA.

for Earth & Environmental Science

Lucene-Solr

← → ↺

Police

Jenkins

↑ Zur

🔍 Sta

📁 Änd

📁 Arb

📁 Gro

🔥 Jav

🛡 Em

🔍 suche

#242

15.06

Java

XX:+

XX:+

#242

15.06

Java

XX:+UseG1

#24232

15.06.2019

Java: 64bit/

XX:+UseCo

#24231

```
[junit4]
[junit4] Suite: org.apache.lucene.search.TestSimilarityProvider
[junit4] Completed [25/410] on J1 in 0.05s, 1 test
[junit4]
[junit4] JVM J0: stdout was not empty, see: /home/jenkins/workspace/Lucene-Solr-trunk-
Linux/lucene/build/core/test/temp/junit4-J0-20160130_093254_626.sysout
[junit4] >>> JVM J0 emitted unexpected output (verbatim) ----
[junit4] #
[junit4] # A fatal error has been detected by the Java Runtime Environment:
[junit4] #
[junit4] # SIGSEGV (0xb) at pc=0xef297ffb, pid=18795, tid=19014
[junit4] #
[junit4] # JRE version: Java(TM) SE Runtime Environment (9.0+102) (build 9-ea+102-2016-01-21-
001243.javare.4316.nc)
[junit4] # Java VM: Java HotSpot(TM) Server VM (9-ea+102-2016-01-21-001243.javare.4316.nc,
mixed mode, tiered, g1 gc, linux-x86)
[junit4] # Problematic frame:
[junit4] # J 5236 C2
org.apache.lucene.codecs.compressing.CompressingTermVectorsWriter.flushOffsets([I)V (820 bytes)
@ 0xef297ffb [0xef297d20+0x000002db]
[junit4] #
[junit4] # No core dump will be written. Core dumps have been disabled. To enable core
dumping, try "ulimit -c unlimited" before starting Java again
[junit4] #
[junit4] # An error report file with more information is saved as:
[junit4] # /home/jenkins/workspace/Lucene-Solr-trunk-
Linux/lucene/build/core/test/J0/hs_err_pid18795.log
[junit4] Could not load hsdisk-1386.so; library not loadable; PrintAssembly is disabled
[junit4] [thread 19028 also had an error]
[junit4] [thread 19016 also had an error]
[junit4] #
[junit4] # If you would like to submit a bug report, please visit:
[junit4] # http://bugreport.java.com/bugreport/crash.jsp
[junit4] #
[junit4] <<< JVM J0: EOF ----
[junit4] Suite: org.apache.lucene.index.TestIndexWriterMerging
[junit4] Completed [26/410] on J2 in 3.89s, 6 tests
[junit4]
```

GAEA

for Earth & Environmental Science

Lucene 8

令和 era

“The” Change

- **New result collection engine**
 - Allows short circuit if total count is not needed
- Works for combinations of many query types!

How does it work?

- Add some information about **maximum TF** and **norm** to **posting list** blocks (e.g., 64 postings or larger)
- **Multi-Level:** same stats for block of blocks!
- Stored in already existing “Skip List”

How does it work?

**Faster top-k document retrieval using
block-max indexes. SIGIR '11**

- Proceedings of the 34th international ACM
SIGIR conference on Research and
development in Information Retrieval,
- Pages 993-1002,

- <https://doi.org/10.1145/2009916.2010048>

How does it work?

- Add some information about **maximum TF** and **norm** to **posting list** blocks (e.g., 64 postings or larger)
- **Multi-Level:** same stats for block of blocks!
- Stored in already existing “Skip List”



THANK YOU!

Questions?



SD DataSolutions GmbH

Wätjenstr. 49

28213 Bremen, Germany

+49 421 40889785-0

<https://www.sd-datasolutions.de>

