Non-Intrusive Performance Bugs in Real-World Applications

Matsui, Donovan et Nistor, Adrian

Overview

In today's multi-core oriented world of computing, the speed of individual processors is no longer increasing at a predictable rate. This forces software developers to focus more than ever on code efficiency and performance bug avoidance. While software systems increase in complexity, it is becoming more and more difficult to locate and fix bugs affecting performance. Research must be done to discover better performance bug profiling techniques and design patching tools for developers facing these challenges.

What do all of these widely-used applications have in common?

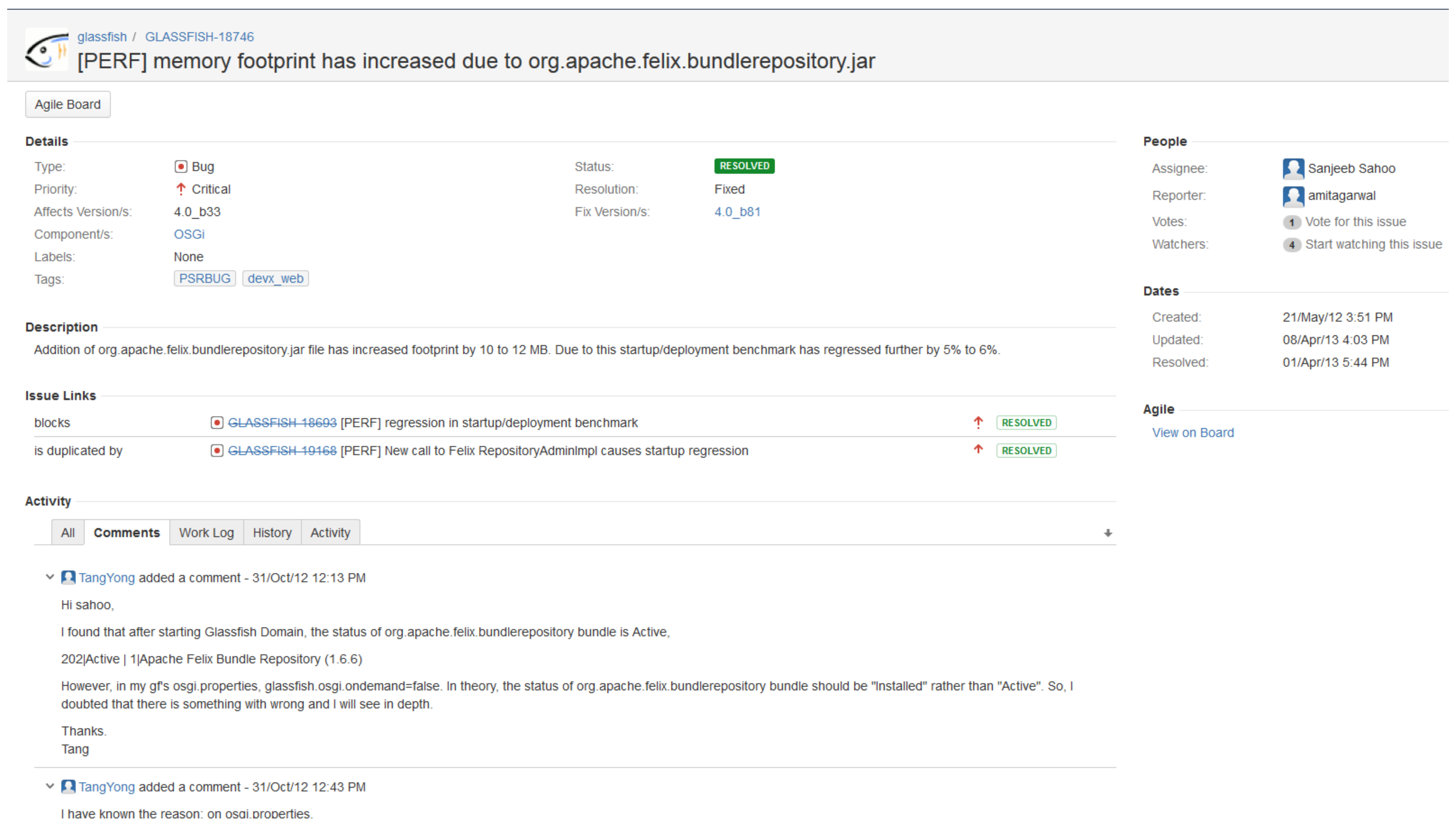
guava-libraries

eclipse

GCC

Apache

...Performance bugs!



glassfish / GLASSFISH-18746

[PERF] memory footprint has increased due to org.apache.felix.bundlerepository.jar

Agile Board

Details

Type:	Bug	Status:	RESOLVED
Priority:	Critical	Resolution:	Fixed
Affects Version/s:	4.0_b33	Fix Version/s:	4.0_b81
Component/s:	OSGi		
Labels:	None		
Tags:	PSRBUG devx_web		

Description

Addition of org.apache.felix.bundlerepository.jar file has increased footprint by 10 to 12 MB. Due to this startup/deployment benchmark has regressed further by 5% to 6%.

Issue Links

blocks	~~GLASSFISH-18693~~ [PERF] regression in startup/deployment benchmark	RESOLVED
is duplicated by	~~GLASSFISH-19168~~ [PERF] New call to Felix RepositoryAdminImpl causes startup regression	RESOLVED

Activity

All | Comments | Work Log | History | Activity

TangYong added a comment - 31/Oct/12 12:13 PM

Hi sahoo,

I found that after starting Glassfish Domain, the status of org.apache.felix.bundlerepository bundle is Active,

202|Active | 1|Apache Felix Bundle Repository (1.6.6)

However, in my gf's osgi.properties, glassfish.osgi.ondemand=false. In theory, the status of org.apache.felix.bundlerepository bundle should be "Installed" rather than "Active". So, I doubted that there is something with wrong and I will see in depth.

Thanks.
Tang

TangYong added a comment - 31/Oct/12 12:43 PM

I have known the reason: on osgi.properties.

People

Assignee:	Sanjeeb Sahoo
Reporter:	amitagarwal
Votes:	1 Vote for this issue
Watchers:	4 Start watching this issue

Dates

Created:	21/May/12 3:51 PM
Updated:	08/Apr/13 4:03 PM
Resolved:	01/Apr/13 5:44 PM

Agile

View on Board

Research Process

1. *Mine developer forums* for bugs which meet the profile. Ideal bugs are reproduceable, testable, and fixable, however, most are not ideal.

To the right is an example posting from the forums of Sun Microsystems' open-source serve application, Glassfish.

2. *Extract all pertinent information*: generally a summary of the bug and what caused it, e.g.) The developer(s)' misunderstanding of API features, failure to use the correct data structures, not foreseeing how a patch can cause regressions, etc. It is important to note any suspect methods and classes, any interesting comments made between the bug assignee(s) and bug reporter, and any related bugs/patterns across other bugs. If a link to the fixed revision (patch) is provided, this is included because it is very helpful when it comes to reproducing the bug and narrowing down the offending code.

An example summary is provided below.

An Apache Commons Collections bug demonstration and patch reported and written by Adrian Nistor.

Example performance bug

A.K.A. a programming error that slows down program execution.

```
//***TEST***
import java.util.ArrayList;
import java.util.List;

import org.apache.commons.collections.ListUtils;

public class Test {
    public static void main(String[] args) {
        int largeNumber = 300000;
        List<Integer> one = new ArrayList<Integer>();
        for(int i = 0; i < largeNumber; i++) {
            one.add(i+5000);
        }
        List<Integer> two = new ArrayList<Integer>();
        for(int i = 0; i < largeNumber; i++) {
            two.add(i+7000);
        }

        long start = System.currentTimeMillis();
        ListUtils.subtract(one, two); // problem manifests here
        long stop = System.currentTimeMillis();
        System.out.println("Time is " + (stop - start));
    }
}
//***PATCH***
Index: src/main/java/org/apache/commons/collections/ListUtils.java
===================================================================
--- src/main/java/org/apache/commons/collections/ListUtils.java (revision 1342815)
+++ src/main/java/org/apache/commons/collections/ListUtils.java (working copy)
@@ -23,6 +23,7 @@
 import java.util.Iterator;
 import java.util.List;

+import org.apache.commons.collections.bag.HashBag;
 import org.apache.commons.collections.list.FixedSizeList;
 import org.apache.commons.collections.list.LazyList;
 import org.apache.commons.collections.list.PredicatedList;
@@ -106,9 +107,12 @@
      * @throws NullPointerException if either list is null
      */
     public static <E> List<E> subtract(final List<E> list1, final List<? extends E> list2) {
-        final ArrayList<E> result = new ArrayList<E>(list1);
-        for (E e : list2) {
-            result.remove(e);
+        final ArrayList<E> result = new ArrayList<E>();
+        HashBag<E> bag = new HashBag<E>(list2);
+        for (E e : list1) {
+            if (!bag.remove(e, 1)) {
+                result.add(e);
+            }
         }
         return result;
     }
```

3. Once an adequate number of bugs have been found and classified, *select a representative sample* to replicate in order to gain a better understanding of how and why performance bugs occur. Preferably, instructions are written with the exact steps needed to replicate each bug to provide for easy verification by other developers.

Below is an example set of instructions for reproducing a bug in Glassfish using the Linux terminal.

```
Prerequisites-
   Maven 3.0.3 and above, JDK 1.7.0_09 and above

Open terminal, checkout glassfish revision 61421
      svn checkout https://svn.java.net/svn/glassfish~svn/trunk/main -r61421

Set proper environment variables
      export MAVEN_OPTS="-Xmx1024M -XX:MaxPermSize=512m"

      export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64

if you wish, test they are set properly with echo
      echo $MAVEN_OPTS

navigate to glassfish install_dir/main and install (takes ~20 minutes)
      mvn install

create a server domain
      cd <install_dir_main>/appserver/distributions/glassfish/target/stage/
                                                          glassfish4/glassfish/bin
      ./asadmin create-domain --adminport 4848 domain2

Make the following changes to DomainXmlPersistence class
      cd <install_dir_main>/nucleus/core/kernel/src/main/java/com/sun/
                                                          enterprise/v3/server
      nano DomainXmlPersistence.java

in the imports add "import java.lang.*;"

scroll down and after these two lines-

"protected void saved(File destination) {
 logger.fine("Configuration saved at " + destination);

insert the following-

      try{
      long timeNow = System.currentTimeMillis();
      String data = Long.toString(timeNow) + "\n";
      File file =new File("DemonstratingRedundantDomainSaving.txt");
```

```
      https://java.net/jira/browse/GLASSFISH-18746
short: org.apache.felix.bundlerepository.jar activated at startup even though it isn't a default
      bundle, increases footprint 10-12 MB and total deployment benchmark regresses 5-6%
test: yes, run the developer scenario benchmark with and without the
      org.apache.felix.bundlerepository.jar file in the modules directory. The provided test
      results show footprint reduced from 410900 KB to 400708 KB, a ~10 MB difference.
      *Note*: this test is NOT a fix, only demonstrates impact of activating the module.
reproduce: yes
interesting: The bug has an easy/less preferrable fix (change hk2/osgi-adapter to add obr
      bundle when the flag is present) and an intrusive fix involving the implementation of
      conditional provisioning, ultimately the unintrusive fix is taken.
fix: unintrusive, low risk
pin point: jar is found at- glassfish4/glassfish/modules/org.apache.felix.bundlerepository.jar
      fix is within /main/nucleus/osgi-platforms/felix/src/main/resources/config/osgi.properties
      (see url)
revision: 60460
full story: upon GF startup, the status of org.apache.felix.bundlerepository bundle is Active
      even when in GF's osgi.properties, glassfish.osgi.ondemand=false. The issue is this code
      within  osgi.properties:

      obr.bundles=$
      {com.sun.aas.installRootURI}

      modules/org.apache.felix.bundlerepository.jar and obr.bundles is contained in $
      {core.bundles}

      which makes felix auto-start the org.apache.felix.bundlerepository bundle. to fix this,
      osgi.properties is configured so that the bundle is not activated.

url: https://java.net/projects/glassfish/sources/svn/revision/60460
related to: https://java.net/jira/browse/GLASSFISH-18693
duplicate bug report: https://java.net/jira/browse/GLASSFISH-19168
```

4. Although this phase is not yet reached, the goal of all of this research is to be able to write tool support for developers that can help predict, locate, and fix performance bugs which otherwise may never have been noticed. This has the capability to save countless man-hours by circumventing traditional bug-finding techniques. Additionally, this would lead to better program performance for the end-user and less energy waste. A win for everyone involved!

References

Nistor, A., Jiang, T., Tan, L.. Discovering, Reporting, and Fixing Performance Bugs | Nistor, A., Chang, P., Radoi, C., Lu, S.. Caramel: Detecting and Fixing Performance Problems That Have Non-Intrusive Fixes.. | Jin, G., Song, L., Shi, X., Scherpelz, J., Lu, S.. Understanding and Detecting Real-World Performance Bugs..