# 02: jvisualvm to detect memory leak – a quick tutorial style Java demo

Posted on March 16, 2016 by Arulkumaran Kumaraswamipillai



This is a companion post to 8 Java Memory Management Interview Q&A demonstrating a memory leak scenario step by step with jvisualvm, which is a free profiling tool that gets shipped with JDK.

## Step 1: Code that causes memory leak

Here is a sample code that causes memory leak. It is shown with a never ending while loop for the demo purpose, but in a real production application this could be a logic within a method that gets accessed very frequently.

```
1
2  import java.util.HashMap;
3  import java.util.Map;
4  import java.util.concurrent.TimeUnit;
5
6  public class MemoryLeakDemo {
7      public static void main(String[] args) throw
8          Map<Key, String> map = new HashMap<Key,
9          int counter = 0;
10         while (true) {
11             // creates duplicate objects due to
12             map.put(new Key("dummyKey"), "value"
13             counter++;
14             if (counter % 1000 == 0) {
15                 System.out.println("map size: " +
16                 TimeUnit.SECONDS.sleep(2);
17             }
18         }
19     }
20
21     // inner class key without hashcode() or equ
22     static class Key {
23         private String key;
24
25         public Key(String key) {
26             this.key = key;
27         }
28     }
29 }
30
31
```

In the above code the “Key” class that is used for storing values into a map is not properly implemented by overriding equals() & hashCode(), hence it will be using Object class’s implementation which uses the memory location of each new object created.

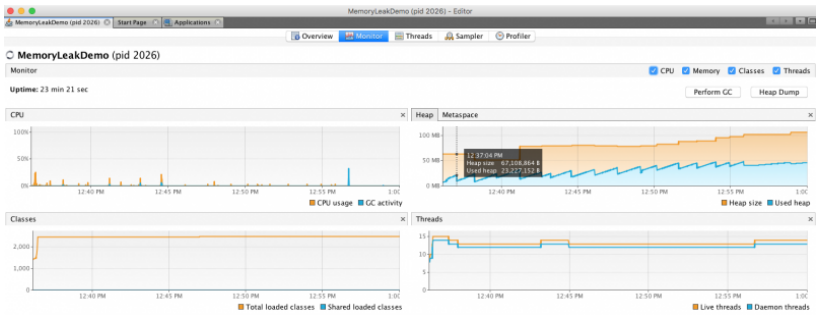
# Step 2: Start jvisualvm

```
1
2  $ jvisualvm
3
```

# Step 3: Run the code & monitor jvisualvm

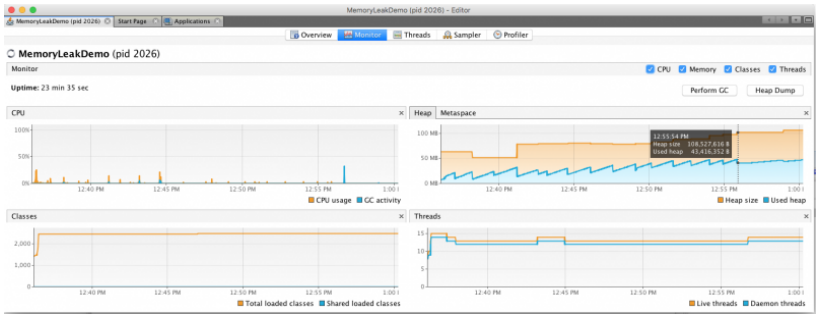


## 16 Technical Key Areas



Monitoring the heap – when started running the application

After 35 minutes:

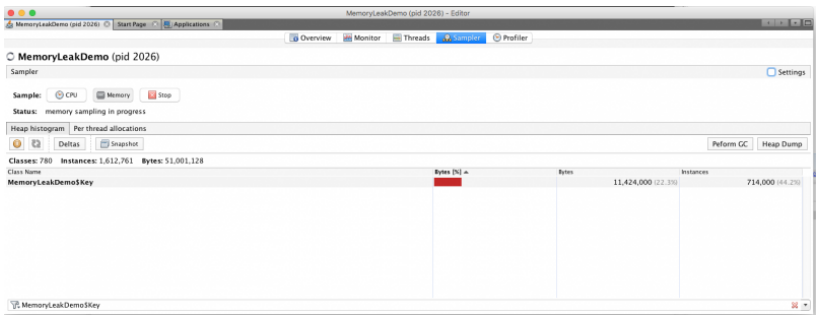


Monitoring the JVM heap after 35 minutes

As, you can see the JVM heap memory usage is keep going up, The “saw tooth” like diagram shown above indicates memory leak. The memory used has gone up from 23MB to 43MB within 35 minutes.

# Step 4: Uncontrolled creation of the instances of Key class is the culprit

As you can see 714K instances created at this point.




- Java Debugging (21)
- Judging Experience I
- Low Latency (7)
- Memory Managemen
- Performance (13)
- QoS (8)
- Scalability (4)
- SDLC (6)
- Security (13)
- Transaction Managen


## 80+ step by step Java Tutorials


open all | close all

- Setting up Tutorial (6)
- Tutorial - Diagnosis (2
- Akka Tutorial (9)
- Core Java Tutorials (2
- Hadoop & Spark Tuto
- JEE Tutorials (19)
- Scala Tutorials (1)
- Spring & HIbernate Tu
- Tools Tutorials (19)
- Other Tutorials (45)


## 100+ Java pre-interview coding tests


open all | close all

- Can you write code? (
- ♦ Complete the given
- Converting from A to I
- Designing your classe
- Java Data Structures
- Passing the unit tests
- What is wrong with th

Sampling the JVM heap

# Step 5: How to fix the code?

Implement the hashCode() & equals() method to the “Key” class and run the code and profile with jvisualvm.

```
1
2  import java.util.HashMap;
3  import java.util.Map;
4  import java.util.Objects;
5  import java.util.concurrent.TimeUnit;
6
7  public class MemoryLeakDemo {
8      public static void main(String[] args) throw
9          Map<Key, String> map = new HashMap<Key,
10         int counter = 0;
11         while (true) {
12             // creates duplicate objects due to
13             map.put(new Key("dummyKey"), "value"
14             counter++;
15             if (counter % 1000 == 0) {
16                 System.out.println("map size: " +
17                 TimeUnit.SECONDS.sleep(2);
18             }
19         }
20     }
21
22     // inner class key without hashcode() or equ
23     static class Key {
24         private String key;
25
26         public Key(String key) {
27             this.key = key;
28         }
29
30         @Override
31         public int hashCode() {
32             return Objects.hash(key); // Java 8
33         }
34
35         @Override
36         public boolean equals(Object obj) {
37             if (obj == null) {
38                 return false;
39             }
40             if (getClass() != obj.getClass()) {
41                 return false;
42             }
43
44             Key other = (Key) obj;
45
46             return Objects.equals(this.key, othe
47         }
48     }
49 }
50
51
```

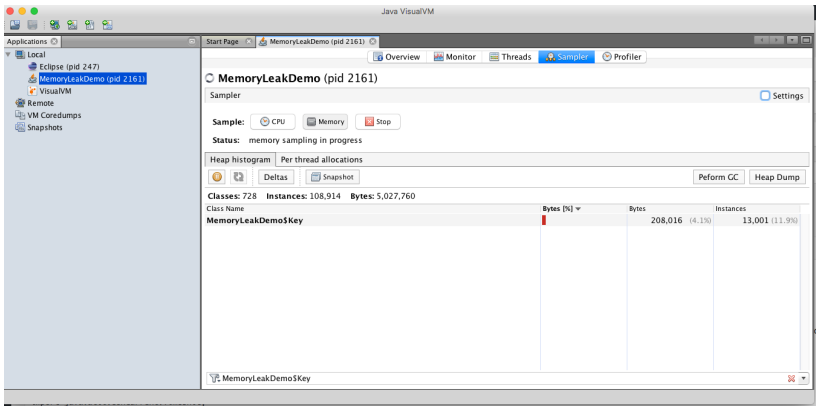
Writing Code Home A
Written Test Core Jav
Written Test JEE (1)

## How good are your .....?

open all | close all
Career Making Know-
Job Hunting & Resum

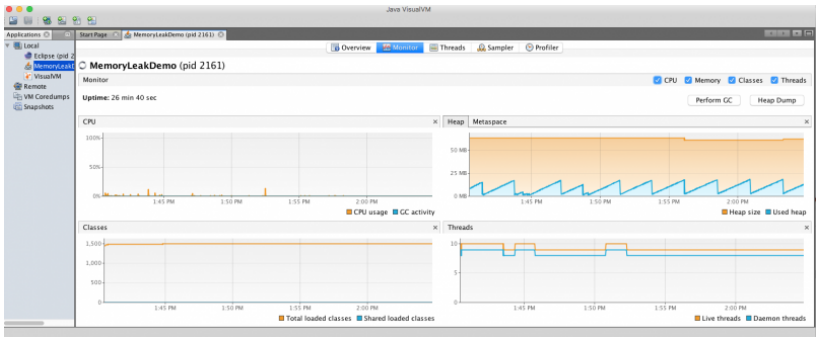
# Step 6: jvisualvm sampling after fixing the code

Even though the instances count shows 13,000 instances, it was because the GC has not been kicked in yet. Click on the "Perform GC" button a few times and you will see the count go down to 1.




JVM heap sampling

# Step 7: jvisualvm heap memory monitoring after fixing the code




Heap Memory Monitoring

As you can see, the memory usage is fully under control without any leaks. The key objects created in the while loop periodically gets garbage collected.

# Popular Posts

♦ 11 Spring boot interview questions & answers

**825 views**

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

**766 views**

18 Java scenarios based interview Questions and Answers

**400 views**

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

**388 views**

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

**295 views**

♦ 7 Java debugging interview questions & answers

**293 views**

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

**285 views**

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

**279 views**

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

**239 views**

001B: ♦ Java architecture & design concepts interview questions & answers

**201 views**

Bio | Latest Posts

## Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription

based site.**945+** paid members. join my LinkedIn Group. **Reviews**

**About** Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ 01: jvisualvm to sample Java heap memory

11: Threads performing tasks by talking to each other ›

**Posted in** jvisualvm profiling, JVM, Memory Management

# Empowers you to open more doors, and fast-track

### Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

### Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

# Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category

# © Disclaimer



 ↑ 