Max Wang

June 6th, 2023

Week 1

Memory Alert Systems

# Summary

Many memory analyzer tools exist online, each with certain capabilities ranging from simple alerts to full debugging output. Eclipse has a built-in analyzer tool for Java programs that utilize heaps. Its capabilities range from listing the heap usage of all objects in a class, as well as reports on leak suspects and the most expensive objects. It has a stand-alone version and an Eclipse-incorporated version. However, there is no simple way to send alerts, beyond copying the report manually and sending an email with a Python script. I looked at other tools with similar capabilities, and here are some observations.

# PVS Studio-Java - BlameNotifier

[PVS Studio-Java](https://pvs-studio.com/en/) is a static code analysis tool designed to detect and prevent bugs, vulnerabilities, and code quality issues in Java applications. It performs a thorough analysis of the Java source code without executing the program, allowing it to identify potential problems early in the development process. One particular feature PVS Studio-Java offers is the feature [BlameNotifier](https://pvs-studio.com/en/docs/manual/0038/), which is meant “for automating the process of notifying developers who have committed the code in the repository for which the PVS-Studio analyzer has issued warnings. The analyzer report is passed to the blame-notifier with specification of additional parameters; the utility finds files that triggered warnings and generates an HTML-report for each "guilty" developer. It is also possible to send a full report: it will contain all warnings related to each "guilty" developer.”

# Recipients of the full report

username\_1 \*email\_1

...

username\_N \*email\_N

# Recipients of individually assigned warnings

username\_1 email\_1

...

username\_N email\_N

This program seems to address the goals; however, this feature is only possible given an enterprise license from PVS-Studio. While the personal and educational license are free, this feature is not included in those licenses, thus I cannot test whether it works the way it is described in the API.

# Eclipse Memory Analyzer Tool (MAT)

[Eclipse Memory Analyzer Tool (MAT)](https://www.eclipse.org/mat/) is the premier tool for memory management in Java. It is fast and intuitive, generating detailed reports of heap distribution across Objects and leak suspects. It can batch export HPROF files, which is a tool for heap dumps and CPU profiling shipped with every JDK release, which can be sent as an email attachment and opened with VisualVM or Eclipse.

The streamlined process required is shown below. Each step will be explained in further detail.

## Automatic HPROF

The Java program must be running for this process to work, since the simplest way to get a heapdump is using [*jmap*](https://stackoverflow.com/questions/407612/how-to-get-a-thread-and-heap-dump-of-a-java-process-on-windows-thats-not-runnin), which lists the process running as a prerequisite. We can use jmap to get a dump of any process running, assuming the *pid* of the process is known.

jmap -dump:format=b,file=heap.hprof <pid>

A heapdump can be generated this way, and placed in a folder where the python script can access it.

A picture containing text, electronics, screenshot, software

Description automatically generated

I plan to use a batch process script to automate this process and execute the python script once the HPROF heapdump is generated. Running these lines in cmd prompt allows the user to specify the java process they want to run, the place they installed their MAT, and the directory where they store the HPROF files and the programs.

cd <MemoryLeakAlertSystem.bat directory address>

MemoryLeakAlertSystem.bat <pid> <ParseHeapDump.bat location> <project folder>

<recipient email address>

A picture containing text, screenshot, software, multimedia software

Description automatically generated

Utilizing this batch process allows the program to be run on a schedule using Windows scheduler, and further improvements and enhancements can be added to allow more flexibility with parameters.

## HPROF Processing

After the HPROF heapdump is created, the python script can be executed. After locating the address of both the HPROF file, the output folder, and the MAT executable, it runs a MAT report. I used the Leak Suspect Report, but any report can be run by changing the arguments.

def run\_mat\_leak\_suspect\_report(mat\_path, hprof\_file):

command = [

        mat\_path,

        hprof\_file,

        "-output=.html",

        "-redact=BASIC",

        "org.eclipse.mat.api:suspects"

    ]

try:

# Execute the command

process = subprocess.Popen(command, stdout=subprocess.PIPE, stderr=subprocess.PIPE)

    print(command)

output, error = process.communicate()

    if process.returncode == 0:

       print("Leak suspect report generated successfully.")

       return True

By specifying the last argument in the command, I can pick the report that needs to be run, and it would output the results to a zipped folder, which is then extracted by the PY script. From the extracted directory, the python script pulls certain HTML pages and runs text analysis on them, looking for key words such as “Problem Suspect” to scrape from the page and process by the script.

## Email Alert

If no problem suspect is found, the program refrains from sending an email. Through the smtplib library, automatic emails can be sent. If there is a potential memory leak, then the program sends an email, along with a summary of the issue and the variables and objects that are causing the issue. Later improvements to the content can be made for the content of the email to include attachments of histograms and further detail.

## Overall process – Final points

Running the batch process in its entirety through an admin CMD prompt gives an output that looks something like this.

A picture containing text, screenshot, software, multimedia software

Description automatically generated

The email message that is received can be enhanced with further detail in the future, perhaps with attachments as well. I created two Gmail accounts for testing purposes, but this program can send to @COMPANY.com mail accounts as well.

A screenshot of a computer

Description automatically generated