Anomaly Detector program manual

by Martin Olsson
Oliver Nilsson

Table of contents

1.0 Anomaly Detector

1.1 Starting the program

1.1.1 linux command line execution

Go to the directory where you have placed the JVMAnomalyDetector.jar and perform the following command line:

java -jar JVMAnomalyDetector.jar localhost:3502 1, localhost:3503 1, localhost:3504 1

Source code compilation execution solution:

The following assume that you are in the main directory of the project (where Main.java is located for example)

Perform this command in order to compile all the required files for preparation of program execution:

javac GUI/*.java Logs/*.java Listeners/*.java AnomalyDetector/*.java Main.java

Program execution command line, together with two monitored processes with each of a value for specified interval in minutes.

java -cp sqlite-jdbc-3.7.2.jar:. AnomalyDetector.AnomalyDetector localhost:3501 1, localhost:3502 1

1.1.2 windows command line execution

Go to the directory where you have placed the JVMAnomalyDetector.jar and perform the following command line:

java -jar JVMAnomalyDetector.jar localhost:3502 1, localhost:3503 1, localhost:3504 1

Source code compilation execution solution:

You may need to set path for java first, for example:
C:\mywork> set path=%path%;C:\Program Files\Java\jdk1.5.0_09\bin

Then you enter the main directory where the program files are located and perform the following command.
Program execution command line, together with three monitored processes with each of a value for specified interval in minutes.
java -cp sqlite-jdbc-3.7.2.jar;. AnomalyDetector/AnomalyDetector localhost:3500 25, localhost:3501, localhost:3502

1.2 program functionality

1.2.1 User command input

In order to get the available commands, you can then write “help”. In order to quit the program, you write “shutdown”.

The following is presented if you write help:

Examples use:
COMMAND or COMMAND -PARAMETER (Some commands require a parameter others do not have any parameters)

clear (Clears database of all log entries (EXAMPLE: clear -all))
Paramers:
-all
-HOST:PORT
-HOST:PORT, ..., HOST:PORT

browse (Opens LogBrowser (EXAMPLE: browse))

connect (Connects to a JVM process (EXAMPLE: connect -localhost:1111, localhost:1212, localhost:1313))
Parameters:

-HOST:PORT
-HOST:PORT, ..., HOST:PORT

disconnect (Disconnects from a monitored process (EXAMPLE: disconnect -localhost:1111, localhost:1212, localhost:1313))
Parameters:
-HOST:PORT
-HOST:PORT, ..., HOST:PORT

settings (Displays settings (EXAMPLE: settings))

excessivegc (Sets Excessive GC Time Warning in milliseconds (EXAMPLE: excessivegc -1000)
Parameters:
-MILLISECONDS

memthreshold (Sets memory increase warning threshold in percent (EXAMPLE: threshold -10))
Parameters:
-DOUBLE

connections (Displays all connections and their status (EXAMPLE: connections))

setinterval (Set analysis interval in minutes for spec. process (EXAMPLE: setinterval -localhost:3500:5))
Parameters:
-HOST:PORT:INTEGER

anomaly (Get all anomalies for one process (EXAMPLE: anomaly -localhost:3500))
Parameters:
-HOST:PORT

shutdown (Shuts down program (EXAMPLE: shutdown))

1.2.2 error reporting

Error reporting is done in the creation of an **AnomalyReport** which is sent as a notification to the program and if the user is connecting to the **Anomaly Detector** via **Anomaly Detector Remote Connector**, then he/she receives it there as well.

1.2.3 Process Report

For each monitored process, a **ProcessReport** will be created.

Available info for each of these reports are the following:

First GC - Memory usage after the first executed Garbage Collection
Last GC - Memory usage after the last executed Garbage Collection
Current Status - Can be either LIKELY_MEMORY_LEAK / SUSPECTED_MEMORY_LEAK / POSSIBLE_MEMORY_LEAK / EXCESSIVE_GC_SCAN / OK
Consecutive increase - amount of times the memory has increased in a row.
Daily stats Avg Dif - Daily average difference in memory stats.
Daily stats Min Dif - Daily average minimum difference in memory stats.
Daily stats Increase Count - Daily stats increase count in memory usage.
Daily stats Descending Count - Daily stats for number of times the memory usage has gone down.
Daily stats Report count - Number of daily reports created for this process.

Weekly stats Avg Dif - Weekly average statistics difference
Weekly stats Min Dif - Weekly minimum statistics difference
Weekly stats Increase Count - Weekly statistics increase count
Weekly stats Descending Count - Weekly descending count for statistics
Weekly stats Report count - Number of weekly reports created for this process.
Monthly

Monthly stats Avg Dif - Monthly average statistics difference
Monthly stats Min Dif - Monthly minimum statistics difference
Monthly stats Increase Count - Monthly statistics increase count
Monthly stats Descending Count - Monthly statistics descending count
Monthly stats Report count - Number of monthly reports created for this process.

1.2.4 GcReport

GcReports contain combined information from several garbage collections. For example it could contain information about all garbage collections for one day. Statistics for garbage collections can either be added to a report simply by adding a GcStats-object and then recalculating the report's values or by adding another GcReport and recalculating the report's values based on the added GcReport.
By keeping track of summed values and the amount of garbage collections this report can provide an overview of several garbage collections with average/min/max values.
Available info for each of these reports are the following:

sumCollectionTime - The sum of all garbage collections added to this report.
minCollectionTime - The minimum time it took for a Garbage Collection to be executed.
maxCollectionTime - The maximum time it took for a Garbage Collection to be executed.
sumTimeBetweenGc - The summed time between garbage collections for all garbage collections added to this report.
minTimeBetweenGc - The minimum time between executed Garbage Collections.

maxTimeBetweenGc - The maximum time between executed Garbage Collections.
sumCollected - The sum of all garbage collection by all the garbage collections added to this report.
minCollected - The minimum amount of executed Garbage Collections.
maxCollected - The maximum amount of executed Garbage Collections.
sumMemoryUsage - The summed memory usage after a garbage collection for all garbage collections added to this report.
minMemoryUsage - The minimum memory usage
maxMemoryUsage - The maximum memory usage
startMemoryUsage - Memory usage at the start.
endMemoryUsage - Memory usage at the end.
startTime - The start time of the GCReport expressed in UNIX TIME.
endTime - The end time of the GCReport expressed in UNIX TIME.
gcCount - The amount of Garbage Collections added to this report.
reportCount - The amount of GcReports added to this report.
sumMinMemoryUsage - The average minimum memory usage
status - Could be of the following status types: LIKELY_MEMORY_LEAK, SUSPECTED_MEMORY_LEAK, POSSIBLE_MEMORY_LEAK, EXCESSIVE_GC_SCAN, OK, UNKNOWN
consec_mem_inc_count - How many times the memory has increased in a row.
period - Could be of the following status types: MIXED, DAILY, WEEKLY, MONTHLY

1.2.5 gcStats

After a garbage collection has been performed, data is then saved as a gcStats log for this event. Available info for each of these reports are the following:

Memory Used After - Memory used after a Garbage Collection expressed in kilo bytes.
Memory Used Before - Memory used before a Garbage Collection expressed in kilo bytes.
Timestamp - Timestamp in UNIX TIME for creation of the current GCStats log.
CollectionTime - Timestamp in UNIX TIME it took to execute the Garbage Collection.

1.2.6 AnomalyReport

When a memory leak or excessive GC scan has been detected, an AnomalyReport is then created and notified to the AD program and the occasional connected users of the AnomalyDetector Remote Connector.

Available info for each of these reports are the following:

Anomaly Status - could be of the following status: EXCESSIVE_GC_SCAN / LIKELY_MEMORY_LEAK / SUSPECTED_MEMORY_LEAK / POSSIBLE_MEMORY_LEAK / UNKNOWN
Timestamp - The exact time and date for the created AnomalyReport
Start time increase - From the time where the memory started increasing leading up to the AnomalyReport creation.

Memory increase (in percentage) - How much the memory has increased in percent.
Memory increase (bytes) - How much the memory has increased in bytes.
Error message - Could either state that error is due to excessive GC scan or due to a suspected memory leak.

1.2.7 Logging and SQLite info

We have utilized SQLite as the core of the logging system. It is here AnomayReports, ProcessReports, GCStats and GCReports are stored and fetched.

SQLite is as the name speaks about, a lite version of an SQL database. While bigger databases works as a separate process, SQLite is an integrated part of a program. We chose it because it felt like we could make good use of it, and it also made development time on the Logging front taking considerably less time.

1.2.8 Old data handling

We have clearing of old data functionality implemented in the log system.
GCStats - is cleared once every 2 months.
Daily GCReports - is cleared once every 2 months.
Weekly GCReports - are cleared every 4 months.
Monthly GCReports - are cleared once every 12 months.
AnomalyReports are cleared once every 6 months.

1.2.9 data log reorganizing

As time passes and the AnomalyDetector program is running, it will exhibit the following behaviour:

- A daily report is created from several GcStats-objects by saving **min/max and average values** of the statistics contained in GcStats.

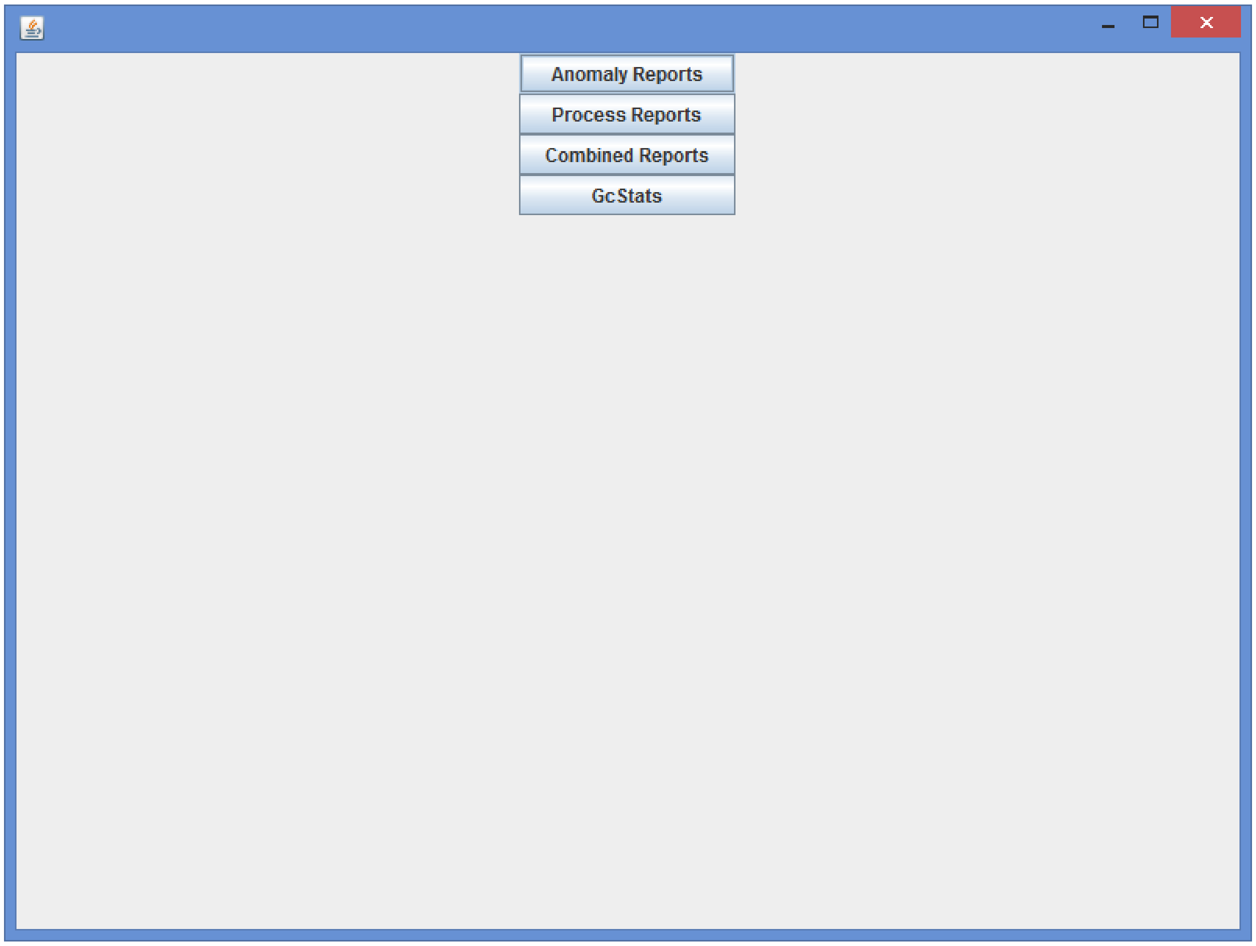
- A weekly report is like a daily report but for a weekly period. Instead of creating values from GcStats-objects it instead gets it values by **combining seven daily reports** and once again save **min/max and average values.**

- A monthly report is **just like the weekly report but** instead of combining seven daily reports **it combines four weekly reports.**

All these reports will be used in the analyze-phase. There they will be compared to each other to see how a process' memory usage changes. A daily report will be compared to the previous day's report while a weekly report will be compared to the previous week's report etc.

1.2.10 Log browser

By executing the command **browse** at console in linux in a desktop environment, or in windows console, you are then able to start the log browser which provides you with the ability to browse **Anomaly Reports, Process Reports, Combined Reports and GcStats**
based on process input of ip address, port, and by choosing between **All / Daily / Weekly** or **Monthly reports.**



AnomalyReport

Available info for each of these reports are the following:

Anomaly Status - could be of the following status: EXCESSIVE_GC_SCAN / LIKELY_MEMORY_LEAK / SUSPECTED_MEMORY_LEAK / POSSIBLE_MEMORY_LEAK / UNKNOWN
Timestamp - The exact time and date for the created AnomalyReport
Start time increase - From the time where the memory started increasing leading up to the AnomalyReport creation.
Memory increase (in percentage) **-** How much the memory has increased in percent.
Memory increase (bytes) **-** How much the memory has increased in bytes.
Error message - Could either “Heap memory usage in PS OLD Gen above threshold.” or “Time between Garbage Collections has gone under <*standard Warning Time set*> milliseconds!”

ProcessReport

For each monitored process, a **ProcessReport** will be created.

Available info for each of these reports are the following:

First GC - Memory usage after the first executed Garbage Collection
Last GC - Memory usage after the last executed Garbage Collection
Current Status - Can be either LIKELY_MEMORY_LEAK /
SUSPECTED_MEMORY_LEAK / POSSIBLE_MEMORY_LEAK /
EXCESSIVE_GC_SCAN / OK
Consecutive increase - amount of times the memory has increased in a row.
Daily stats Avg Dif - Daily average difference in memory stats.
Daily stats Min Dif - Daily average minimum difference in memory stats.
Daily stats Increase Count - Daily stats increase count in memory usage.
Daily stats Descending Count - Daily stats for number of times the memory usage has gone down.
Daily stats Report count - Number of daily reports created for this process.

Weekly stats Avg Dif - Weekly average statistics difference
Weekly stats Min Dif - Weekly minimum statistics difference
Weekly stats Increase Count - Weekly statistics increase count
Weekly stats Descending Count - Weekly descending count for statistics
Weekly stats Report count - Number of weekly reports created for this process.
Monthly

Monthly stats Avg Dif - Monthly average statistics difference
Monthly stats Min Dif - Monthly minimum statistics difference
Monthly stats Increase Count - Monthly statistics increase count
Monthly stats Descending Count - Monthly statistics descending count
Monthly stats Report count - Number of monthly reports created for this process.

GCStats

Memory Used After - Memory used after a Garbage Collection expressed in kilo bytes.
Memory Used Before - Memory used before a Garbage Collection expressed in kilo bytes.
Timestamp - Timestamp in UNIX TIME for creation of the current GCStats log.
CollectionTime - Timestamp in UNIX TIME it took to execute the Garbage Collection.

Combined Reports

Avg Collection Time - The average time it took to perform a Garbage Collection
Min Collection Time - The minimum time it took to perform a Garbage Collection
Max Collection Time - The maximum time it took to perform a Garbage Collection
Avg Time between GC - The average time between performed Garbage Collections
Min Time between GC - The minimum time between performed Garbage Collections
Max Time between GC - The maximum time between Garbage Collections
Avg Collection - Average amount of Garbage Collections
Min Collection - Minimum amount of Garbage Collections
Max Collection - Maximum amount of Garbage Collections
Avg Memory use - Average Memory Usage
Min Memory use - Minimum Memory Usage
Max Memory use - Maximum Memory Usage
Start memory use - The starting memory usage for the current report.
End memory use - The ending memory usage for the current report.
Start time - The start time of the report in UNIX TIME
End time - The end time of the report in UNIX TIME
GC Count - The number of Garbage Collections performed in the timeframe of the current report.

1.3 Anomaly Detector Remote Connector

We have also developed a command line program that will remotely connect to the Anomaly Detector program which is able to send commands that can be executed within the program upon receiving, incase you want to put the Anomaly Detector program in the background and run it as a process.

1.3.1 program functionality

Connection establishment
The program will first ask you for the host / ip address of the Anomaly Detector program you may want to connect to.
After that it will ask you for the port, which has been set to standard number of “2015” in the Anomaly Detector program.

Username input
Upon connection to the program, you will be presented to enter a username of your choice. This is done in order to distinguish yourself as one among potential several other connected users to the Anomaly Detector program.

In order to get the available commands, you can then write “help”. In order to quit the program, you write “quit”, and there will be a safe disconnection from the Anomaly Detector.

The following is presented if you write help:

Examples use:
COMMAND or COMMAND -PARAMETER (Some commands require a parameter others do not have any parameters)

clear (Clears database of all log entries (EXAMPLE: clear -all))
Paramers:
-all
-HOST:PORT
-HOST:PORT, ..., HOST:PORT

Please bear in mind that the browse command doesn’t work from the Anomaly Detector Remote Connector. Only locally from the Anomaly Detector program !
browse (Opens LogBrowser (EXAMPLE: browse))

connect (Connects to a JVM process (EXAMPLE: connect -localhost:1111, localhost:1212, localhost:1313))
Parameters:
-HOST:PORT
-HOST:PORT, ..., HOST:PORT

disconnect (Disconnects from a monitored process (EXAMPLE: disconnect -localhost:1111, localhost:1212, localhost:1313))
Parameters:
-HOST:PORT
-HOST:PORT, ..., HOST:PORT

settings (Displays settings (EXAMPLE: settings))

excessivegc (Sets Excessive GC Time Warning in milliseconds (EXAMPLE: excessivegc -1000)
Parameters:
-MILLISECONDS

memthreshold (Sets memory increase warning threshold in percent (EXAMPLE: threshold -10))
Parameters:
-DOUBLE

connections (Displays all connections and their status (EXAMPLE: connections))

setinterval (Set analysis interval in minutes for spec. process (EXAMPLE: setinterval -localhost:3500:5))
Parameters:
-HOST:PORT:INTEGER

anomaly (Get all anomalies for one process (EXAMPLE: anomaly -localhost:3500))
Parameters:
-HOST:PORT

<u>NOTE: This command does not work via Anomaly Detector Remote Connector</u>
shutdown (Shuts down program (EXAMPLE: shutdown))