Method Execution Report

Date: 13-04-2017
Type: FN / NF
System: eudve

Session overall:

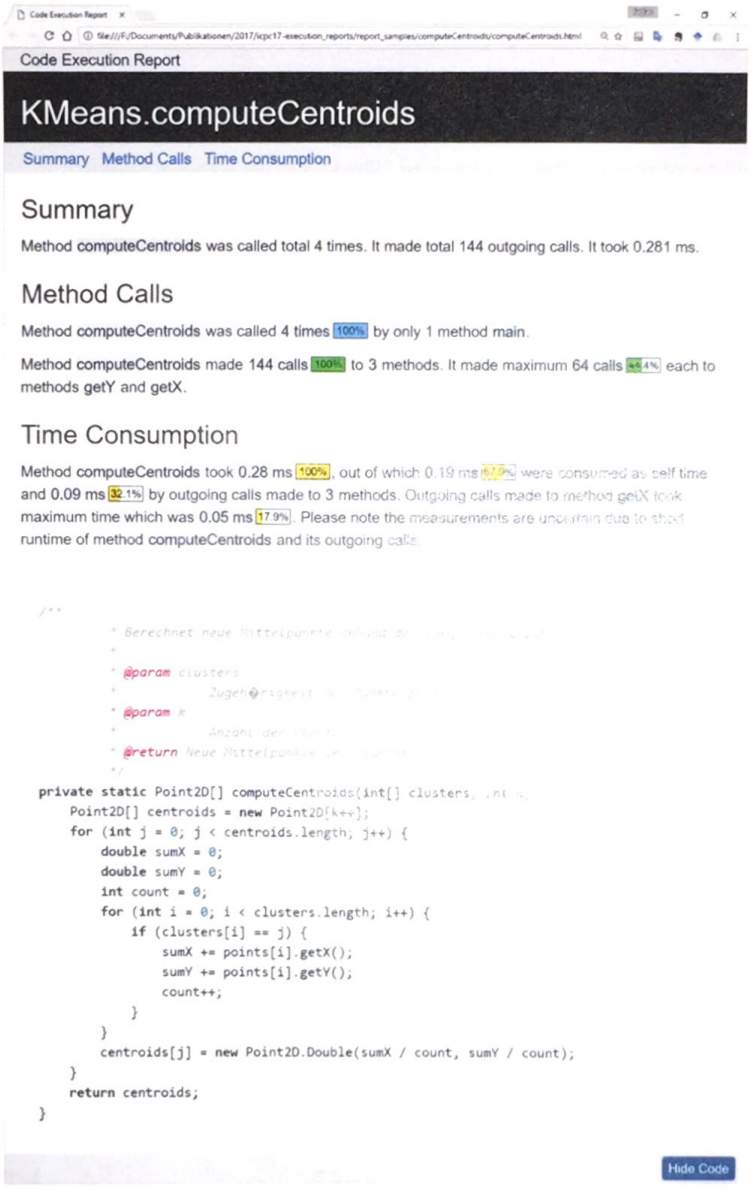
This session aims at gathering impression of the *Method Execution Report* from practitioners
The session is relatively short, only a few minutes long. The session is divided into three phases. The first phase is about yourself. The second phase present Method Execution Report, and the third phase is about the evaluation of the report.
Note that all your answers are treated anonymously.

Phase 1: About yourself

- Are you Female or Male ? Male
- How many years of experience do you have in programming? 14
- How long have you been programming in Java for? For 8 years
- Which Java programming environments (IDE) are you familiar with? Eclipse IDE
- Which other programming languages and programming environments do you use?
  Mainly Python, Scilab and C. I only use Vim as IDE.
- While programming, if your application does not behave as you expect, what do you usually do? How do you usually debug an application?
  I usually inspect the code and add some print functions to get variable values and useful info. If not enough, I tend to use GDB for debugging
- How do you usually do to improve the performance of a particular method?
  I run profiling tools to obtain bottlenecks. Then, I search on the web for more efficient ways to achieve what I'm trying to do.

Phase 2: Description of Method Execution Report

Method Execution Report is a textual and interactive report that summarizes the execution of a particular method for a given software execution. The report provides an overview of the dynamic calls and time consumption.



Code Execution Report

Code Execution Report

KMeans.computeCentroids

Summary Method Calls Time Consumption

Summary

Method computeCentroids was called total 4 times. It made total 144 outgoing calls. It took 0.281 ms.

Method Calls

Method computeCentroids was called 4 times 100% by only 1 method main.

Method computeCentroids made 144 calls 100% to 3 methods. It made maximum 64 calls 44.4% each to methods getY and getX.

Time Consumption

Method computeCentroids took 0.28 ms 100%, out of which 0.19 ms 67.9% were consumed as self time and 0.09 ms 32.1% by outgoing calls made to 3 methods. Outgoing calls made to method getX took maximum time which was 0.05 ms 17.9%. Please note the measurements are uncertain due to short runtime of method computeCentroids and its outgoing calls.

```
/**
         * Berechnet neue Mittelpunkte (Methode des ...
         *
         * @param clusters
         *            Zugeh...rigkeit der Punkte zu ...
         * @param k
         *            Anzahl der Cluster
         * @return Neue Mittelpunkte der Cluster
         */
private static Point2D[] computeCentroids(int[] clusters, int k,
    Point2D[] centroids = new Point2D[k++];
    for (int j = 0; j < centroids.length; j++) {
        double sumX = 0;
        double sumY = 0;
        int count = 0;
        for (int i = 0; i < clusters.length; i++) {
            if (clusters[i] == j) {
                sumX += points[i].getX();
                sumY += points[i].getY();
                count++;
            }
        }
        centroids[j] = new Point2D.Double(sumX / count, sumY / count);
    }
    return centroids;
}
```

Hide Code

Figure on the left gives an example of the report. The report is structured into three sections, and lists the code of the method.

The summary gives first essential data, including the number of incoming and outgoing calls, and how long the summed executions of the method took.

The second section details the description of the method calls, including the most important caller and callees.

The third part provides further information about the timing.

Phase 3:

TIME BEGIN: 17:54:45

QUESTION 1:

What do you think about the content of the textual description?

- I find it easy to understand? *(strongly agree, agree, neutral, disagree, strongly disagree)*

Please, justify

Neutral. Indeed there is useful information, but is not easy to follow because lacks visual helpers.

- I find it useful? *(strongly agree, agree, neutral, disagree, strongly disagree)*

Please, justify

strongly agree. This kind of information can improve a lot the performance of a snippet of code.

QUESTION 2:

What do you think about the interaction and the visual elements offered by the report?

- I find them easy to understand? *(strongly agree, agree, neutral, disagree, strongly disagree)*

Please, justify

Disagree. There is data, but is not easy to follow because there aren't visual elements for understanding the flow (e.g graphs, trees, etc).

- I find them useful? *(strongly agree, agree, neutral, disagree, strongly disagree)*

Please, justify

Disagree. Same as above

QUESTION 3:

Overall, do you feel that such a report is useful?

(strongly agree, agree, neutral, disagree, strongly disagree)

Please, justify

strongly agree. As said, this kind of reports can help a lot to improve the code, detect bottlenecks, etc.

QUESTION 4:
In what scenarios and for solving which maintenance tasks would developers use Method Execution Reports?
Please, justify

It would be useful when there is an iterative process that takes some time to run (e.g simulations). Detecting bottlenecks can improve the simulation time.

QUESTION 5:
What tools would you use instead of Method Execution Reports to retrieve the same information?
Please, justify

I would use profilers, depending on the language and platform. Nevertheless, they usually don't show some useful information like the code itself.

QUESTION 6:
Do you have any suggestion on how to improve the report? Any critic?
Please, justify

I think it should be very important to have more visual elements to help the user understanding the code flow.
I think a graph or tree is enough

TIME END: 18:05:56