

# Method Execution Report

Date: 13-04-2017

Type: FN / NF

System: draw

## 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?

10

• How long have you been programming in Java for?

0

• Which Java programming environments (IDE) are you familiar with?

Eclipse

• Which other programming languages and programming environments do you use?

C/C++, Python, C#, PHP, Visual Studio

• While programming, if your application does not behave as you expect, what do you usually do?  
How do you usually debug an application?

Use the debugger. Printf, trace the application, Unit tests

• How do you usually do to improve the performance of a particular method?

Profile it to identify the method. Use an efficient algorithm and/or data structure. Micro optimizations in Assembly with vectorial instructions when needed

## 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.

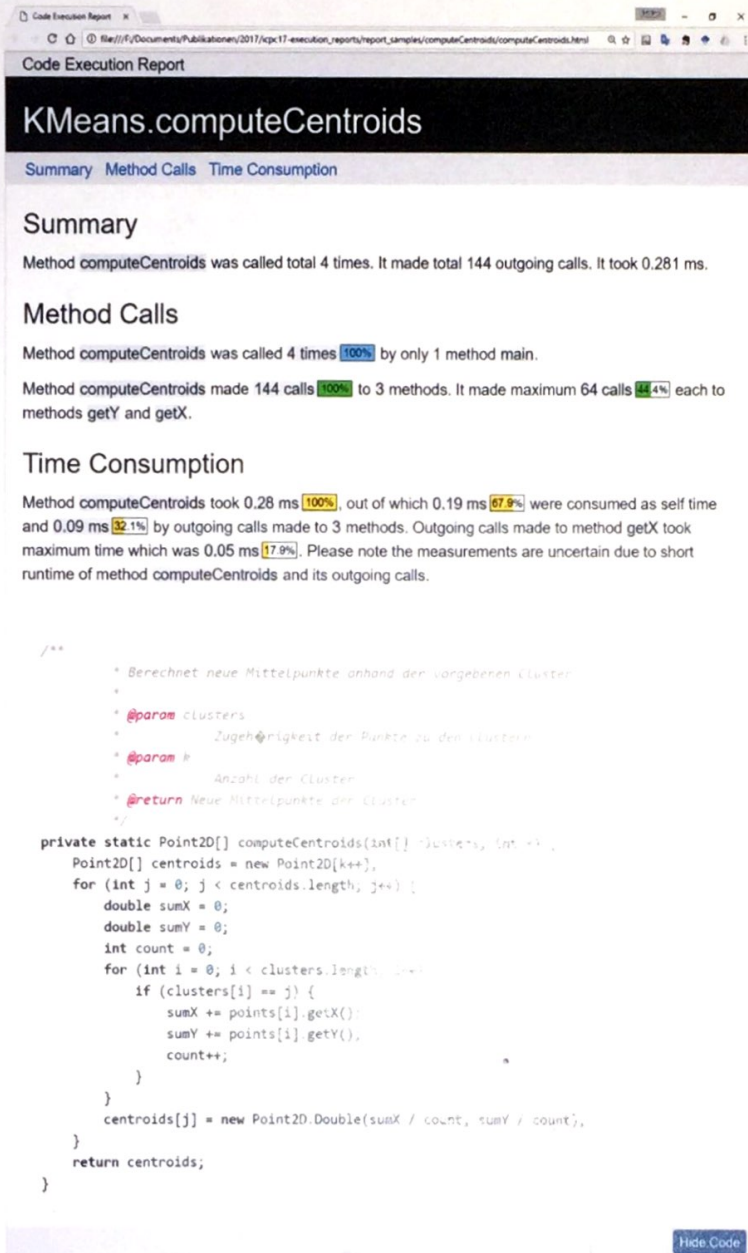


Figure on the left gives an example of the report. The report is structured into three sections, and lists the code of the method.

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: 11:21

#### 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

Strongly agree. It is a clear and concise description about the calls

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

Please, justify

Neutral. In this case, due to the names of the method and the lack of self time, it is clear that the performance problems are in another method

#### 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

agree. We are used to the hyperlinks in the web.

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

Please, justify

Strongly agree. They allow to see the details, such as the time that is spent in an outgoing method

#### QUESTION 3:

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

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

Please, justify

Agree. The only missing element is being able to go deeper in the call tree and see a similar report for the outgoing method



**QUESTION 4:**

In what scenarios and for solving which maintenance tasks would developers use Method Execution Reports?

*Please, justify*

It can be used for finding <sup>some</sup> performance problems and finding where they are located. It is more useful for finding ~~where~~ whether a loop is expensive by itself or because of the methods that it calls.

**QUESTION 5:**

What tools would you use instead of Method Execution Reports to retrieve the same information?

*Please, justify*

In C/C++ I would use gprof. In Java I would use Message Fully. These are tools that are available in other languages, and they provide a global view of the whole call tree. However, the reports ~~are~~ generated by these other tools are not as nice to read as this one.

**QUESTION 6:**

Do you have any suggestion on how to improve the report? Any critic?

*Please, justify*

Add the ability for going deeper into the call tree.

TIME END:

11:41