Worst-case Stall Analysis for Multicore Architectures with Two Memory Controllers

Muhammad Ali Awan[1] (muaan@isep.ipp.pt)
Pedro F. Souto[2] (pfs@fe.up.pt)
Konstantinos Bletsas[1] (ksbs@isep.ipp.pt)
Benny Akesson[3] (benny.akesson@tno.nl)
Eduardo Tovar[1] (emt@isep.ipp.pt)

1. CISTER Research Centre and ISEP, Porto, Portugal
2. University of Porto, Faculty of Engineering and CISTER Research Centre, Porto, Portugal
3. Embedded Systems Innovation, Eindhoven, the Netherlands

1 Summary to run experiments

i. Download the code from the following link.
https://drive.google.com/file/d/1y_mu1OrRjHwSOJStRoVzEx7NiSky5m3F/view?usp=sharing
ii. Install Java (JDK) using apt-get command and Matlab
iii. Give access to folder "ShellScriptsAndJavaTool" with command
"chmod –R a+rwx ShellScriptsAndJavaTool"
iv. Run "super_Shell_Script.sh" shell script available in ShellScriptsAndJavaTool folder with command
"./super_Shell_Script.sh &"
v. Wait for roughly 2.5-3 hours to finish the other scripts invoked by super_Shell_Script.sh
vi. Run the matlab script "WeightedSchedulability.m" available in a "MatlabScript" folder
vii. Find the results in PDF format in "Results" folder

2 Source code:

Our software can be downloaded from the following link. It is zipped file which contains all the required documents for the artifact evaluation. Please, unzip it after downloading.

https://drive.google.com/file/d/1y_mu1OrRjHwSOJStRoVzEx7NiSky5m3F/view?usp=sharing

3 Required Software:

Our experimental setup requires two software: a) Java (JDK) and c) Matlab. It can be configured both on Linux or Windows operating system, however, this file only provides a setup for Linux operating system. Experiments presented in our ECRTS paper were run on standard Ubuntu operating system.

a) JAVA (JDK) can be installed with "apt-get …" command on Ubuntu operating system.
b) A recent version of Matlab such as 8.3.0.532 (R2014a) is used in our experiments, though any other version of Matlab should suffice.

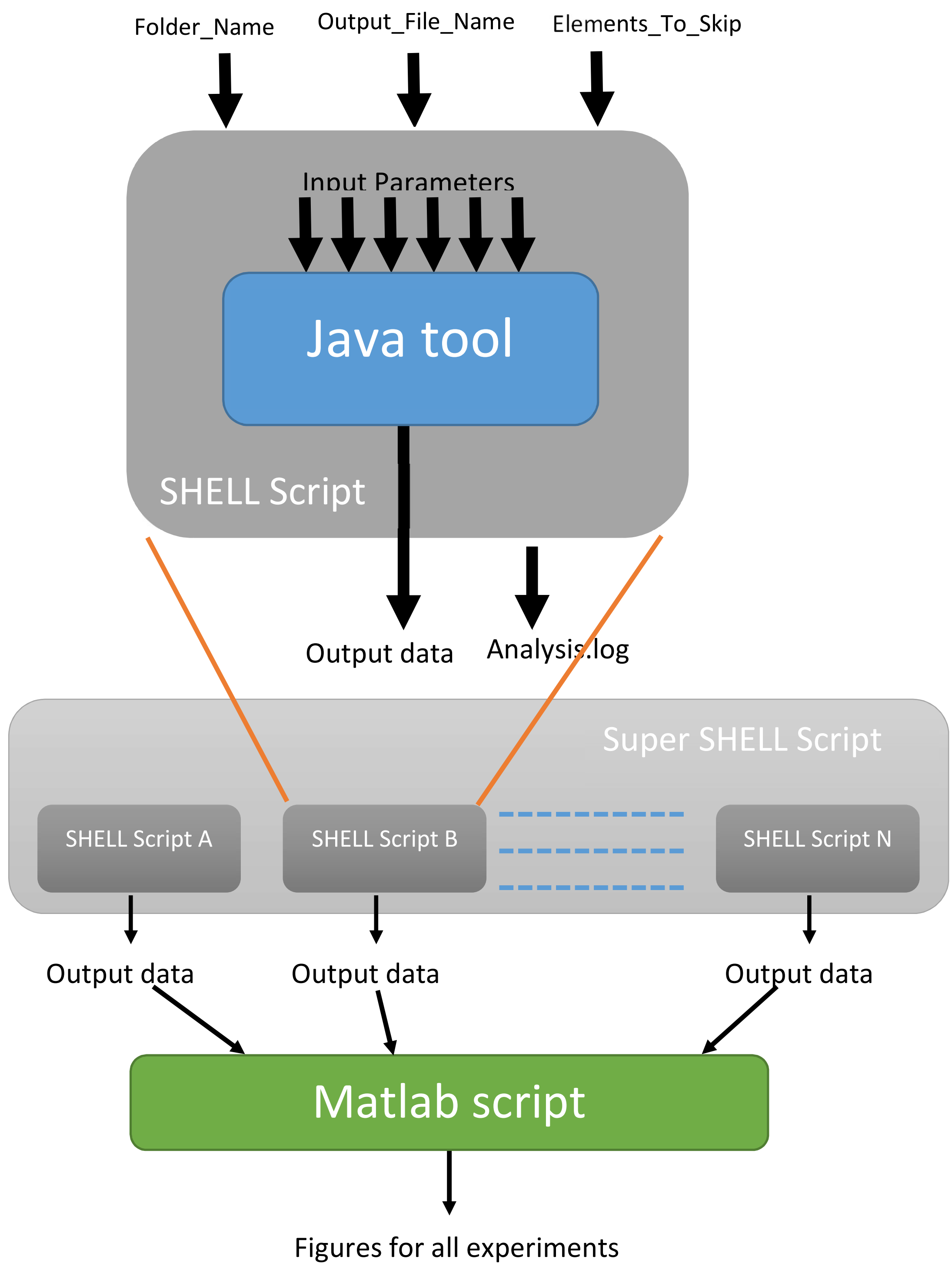

Figure 1: Overall experimental setup

4 Main components:

The experimental setup used in our ECRTS paper can be divided into four main components: super shell script, shell script wrapper, Java tool and Matlab scripts. The interaction between these main components is shown in Figure 1. We have a super shell script, that invokes several shell script wrappers, which in turn runs the Java tool and provides its required inputs. The Java tool implements our proposed algorithms (given in ECRTS paper) and generates the output data for the given set of inputs. Once, all the scripts invoked by the super shell script complete their execution and we have our output files, the Matlab script analyzes the output data to generate the resultant figures used in our ECRTS paper. The detail of each components is presented as follows.

4.1 Super shell script:

The sole purpose of this script is to just invoke the several shell script wrappers. Its name is "super_Shell_Script.sh".

4.2 Shell script wrapper:

The main purpose of the shell script wrapper is to facilitate the variation in input parameters for the Java tool. The input parameters for the Java tools are explained in ECRTS paper in Section 7.1. There is a corresponding shell script wrapper for each experiment performed in our ECRTS paper. These scripts are available in the main folder of the source code with the naming convention "script_XYZ.sh", where XYZ in this name corresponds to the type of experiment. Each shell script takes three arguments.

I) Folder_Name
II) Output_File_Name
III) Elements_To_Skip

The typical command takes the following form on the Linux shell.

./script_Cores.sh Cores cores.txt 0 &

"script_Cores.sh" shell script corresponds to the experiment analyzing the variation in number of Cores variable and hence, it feeds the Java tool with different values of this variable along with other default parameters. The first argument "Cores" is the name of the directory that a script will create to copy all the files. The second argument "Cores.txt" is the name of the output file that a script will feed as one of the inputs to Java tool to write the output data. The third argument allows to skip some of the initial input values to target some specific set of inputs. In the above mentioned example, "0" enforces to run the experiment from start to end. Figure 5 in ECRTS paper presents the results corresponding to this experiment. Table 1 presents the summary of all experiments types, their corresponding shell scripts and the figure numbers in ECRTS paper for quick reference.

No.	Experiment type	Shell script	Figure
1	Number of cores	script_Cores.sh	Figure 5
2	Memory intensity	script_Memory.sh	Figure 6
3	Period	script_Period.sh	No figure
4	Initial task sorting	script_Sorting.sh	Figure 7
5	Task-set size	script_Tasks.sh	Figure 8

Table 1: Shell scripts corresponding to different experiments.

The shell script also generates another file "analysis.log" to report any errors in the execution of Java tool, and prints the execution time of the whole experiment in the Linux terminal.

4.3 Java tool:

The java tool has two packages: a) Analysis and b) Common. The first package keeps all the analysis related files, while the second package contains the rest of the files. The Java tool executes in a sequential manner and its execution can be divided into two parts or modules. The first module (Taskset.java) generates the synthetic workload for the given set of input parameters provided through shell script. The details of the input parameters along with the properties of the task-set generator are presented in Section 7.1 of our ECRTS paper. The second module implements our proposed algorithms along with the other variants to compare against. It takes task-set and platform information as an input and performs the schedulability analysis, task-to-core allocation and memory bandwidth assignment. The java tool output all the results of the schedulability analysis into an output file whose name is provided as an input to the java tool. Please note that compilation of the Java tool is included in shell script and hence, there is no need to compile it separately.

4.4 Matlab script:

The weighted schedulability ratio graphs in our paper are generated through Matlab script. This script is configured to run in batch mode meaning it generates figures for all the experiments in one go as shown in Figure 1. Therefore, it is required to have data (output files generated by Java tool) available from all the experiments presented in our ECRTS paper. As we have different shell scripts for each experiment type (Table 1), therefore, we will have separate output file generated by Java tool for the corresponding experiment type as illustrated in Figure 1. All the paths to these files are set with reference to its current location in the provided folder. If you intend to move the location of this matlab script, you need to manually set paths to these different files. Similarly, if you want to save the results at some other location, you need to change the output path in the same way. The script is very simple and easy to understand. All the resultant figures are generated as PDF.

5 Complete step by step process to run the experiments:

a) Install Java (JDK) using apt-get command

b) Install Matlab

c) Download the code from the following link.

https://drive.google.com/file/d/1y_mu1OrRjHwSOJStRoVzEx7NiSky5m3F/view?usp=sharing

d) Give executions/all rights to a folder “ShellScriptsAndJavaTool” with the following command:
- chmod –R a+rwx ShellScriptsAndJavaTool

e) Run the super shell script (super_Shell_scrip.sh) in shell terminal as given below.
- ./super_Shell_Script.sh &

This script will invoke all the required “shell scripts wrappers” to run all the experiments for ECRTS paper and exits. Once, you have run all the shell scripts wrappers, you need to wait until these scripts complete. It usually takes around 2.5-3 hours to finish all of them on a normal pc.

f) Generate figures with matlab script with the following procedure

- Run the matlab script “WeightedSchedulability.m” available in a “MatlabScript” folder. The result will be saved in the “Results” folder. You can open matlab GUI to run this script or also run the matlab script through command line. The figures will be available in PDF format.
- **Optional:** The default values are set to the relative folder paths provided in the source code directory. You can change the Path, if needed in the script.