ARTEFACT GUIDE

This folder contains the Cheddar models files provided by the design exploration tool as trade-offs between schedulability and security.

1. Implementation

A design space exploration tool is proposed with the PAES method and Exhaustive method. Both methods require callCheddar_securityAnalysis to perform scheduling and security analysis of solutions. The Ada packages in the *framework/paes and framework/architecture_exploration_tools* folders are the source folders.
A readme file (Readme_T2P) is provided in *framework/architecture_exploration_tools* folder. It describes how to run the design exploration tool using the PAES method or the exhaustive method.
This code is part of the Cheddar source code that can be fetched at http://beru.univ-brest.fr/svn/CHEDDAR/trunk/src

2. Results of the design space exploration

With the Ada packages above, we can provide Cheddar models that correspond to trade-offs between schedulability and security for a given case study described through a cheddar model.
We performed different experiments to evaluate our tool. The results of the experiments are grouped into the folders described below.

a. Experimentation2

The case study (test_Rosace_Jpeg.xmlv3) considered in this experiment is made of a flight controller and JPEG applications. The folders Task_grain, App_grain, Mix_grain contain the provided solutions by the Paes method for each of the three mutation algorithms (task-grain, app-grain, mix-grain).

b. Experimentation3

This experiment considers different case studies made of a flight controller, multimedia based application, CFAR, and autopilot applications. This experiment is conducted for the three mutation algorithms on each case study (test_Unifast.xmlv3) generated by varying the processor utilization from 50% to 100%.
Each directory contains the provided solutions by the Paes method for each case study represented by its processor utilization.

c. Experimentation4

This experiment is conducted for the mix-grain mutation algorithm on each case study (test_Unifast.xmlv3) generated by varying the processor utilization U from 50% to 100%. It is also assumed that intra-partition communications are not vulnerable.
Each directory contains the provided solutions by the Paes method for each case study represented by its processor utilization.

d. Experimentation 5

This experiment is conducted with the mix-grain algorithm on the case studies generated in experimentation 3, with processor utilization of 60% and 90%.

The directory Experimentation_with_max_2partitions (resp. Experimentation_with_max_4partitions) contains the provided solutions by the Paes method with the maximum number of two (resp. four) partitions.

e. Experimentation6

This experiment is conducted with the three mutation algorithms on the case study generated in experimentation 3 with a processor utilization of 60%. It also assumed the overheads of intra-partition and inter-partition communications based on the execution time of the APEX calls given from the benchmark SFPBench.

The folders Task_grain, App_grain, Mix_grain contain the provided solutions by the Paes method for each of the three mutation algorithms.

f. **Experimentation7**

This experiment compares an approximate Pareto front obtained with our three PAES based approaches (mutation algorithms) to the exact Pareto front.

The directory Paes has folders Task_grain, App_grain, Mix_grain containing the provided solutions by the Paes method with each of the three mutation algorithms.

The directory Exhaustive contains the provided solutions by the exhaustive method.