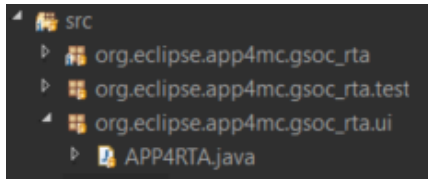


APP4RTA

For Analyzing Response Time & End-to-End Chain Latency

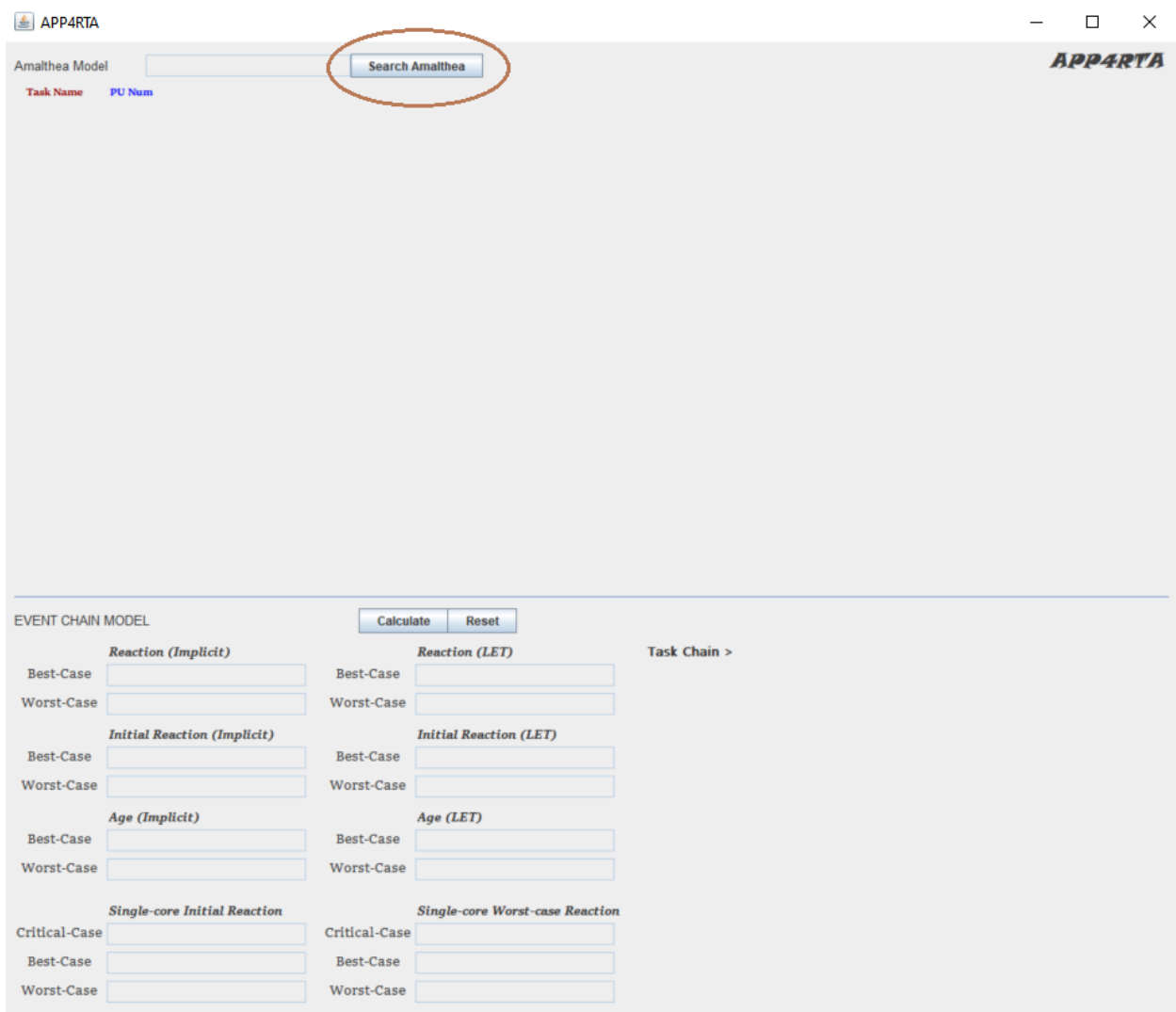
Source: <https://gsoc-doc.readthedocs.io/en/latest/contents/ui.html>

1. APP4RTA Location



Run 'APP4RTA.java' in 'org.eclipse.app4mc.gsoc_rta.ui' package.

2. Search Amalthea



APP4RTA

Amalthea Model

Task Name PU Num

EVENT CHAIN MODEL

Reaction (Implicit)		Reaction (LET)		Task Chain >
Best-Case	<input type="text"/>	Best-Case	<input type="text"/>	
Worst-Case	<input type="text"/>	Worst-Case	<input type="text"/>	
Initial Reaction (Implicit)		Initial Reaction (LET)		
Best-Case	<input type="text"/>	Best-Case	<input type="text"/>	
Worst-Case	<input type="text"/>	Worst-Case	<input type="text"/>	
Age (Implicit)		Age (LET)		
Best-Case	<input type="text"/>	Best-Case	<input type="text"/>	
Worst-Case	<input type="text"/>	Worst-Case	<input type="text"/>	
Single-core Initial Reaction		Single-core Worst-case Reaction		
Critical-Case	<input type="text"/>	Critical-Case	<input type="text"/>	
Best-Case	<input type="text"/>	Best-Case	<input type="text"/>	
Worst-Case	<input type="text"/>	Worst-Case	<input type="text"/>	

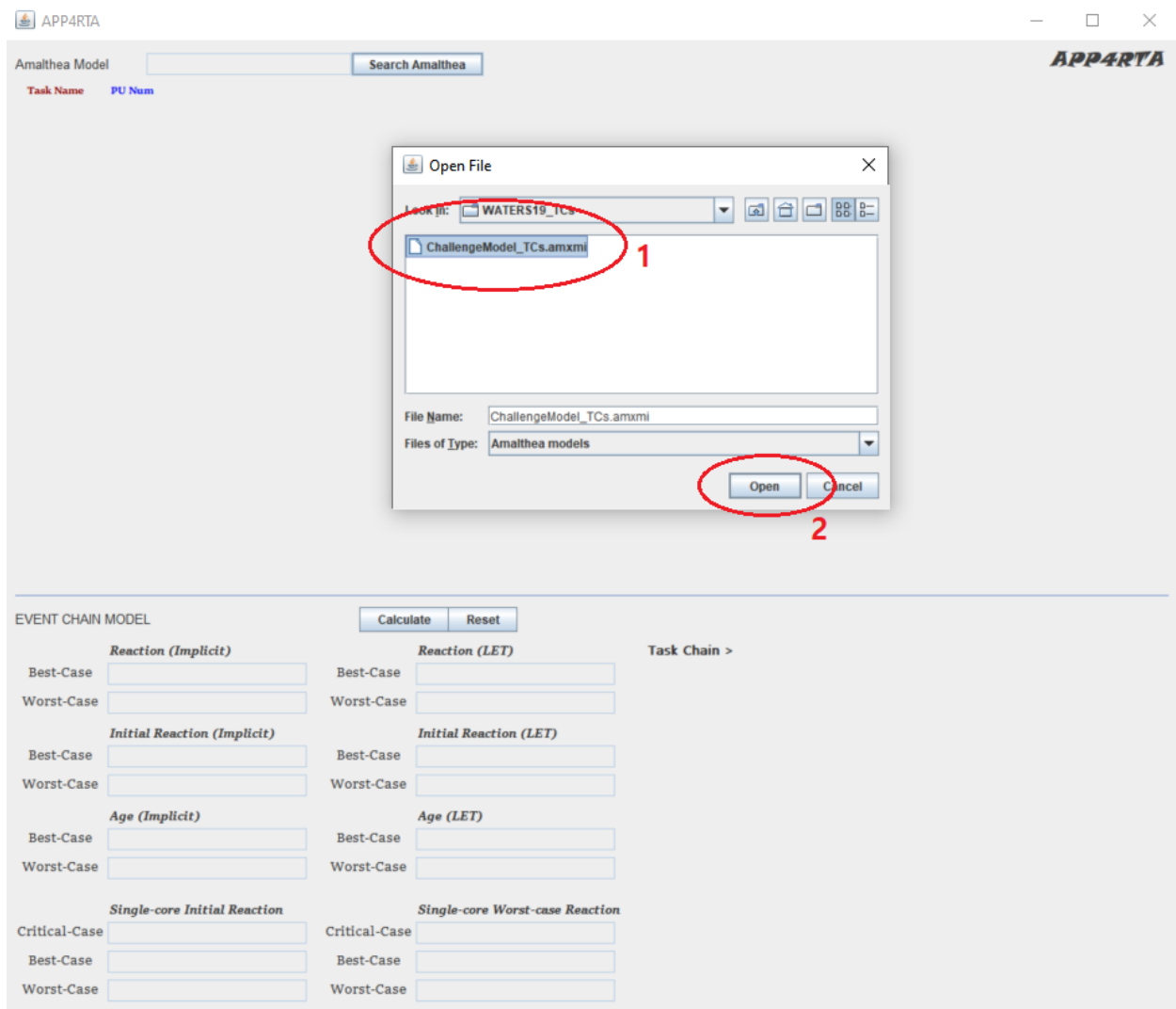
Based on the horizontal line on the middle, the upper part is for response time and mapping analysis and the lower part is for end-to-end event chain latency analysis. The first thing to do is deciding a target Amalthea model. Click the 'Search Amalthea' button.

3. Navigate to The Amalthea Folder

The screenshot shows the APP4RTA application window. At the top, there is a search bar labeled 'Search Amalthea' and the APP4RTA logo. Below this, there is a table with columns 'Task Name' and 'PU Num'. An 'Open File' dialog box is open, showing the path 'app4mc.example.tool.java'. The dialog box lists several folders: '.settings', 'bin', 'META-INF', 'model-input', 'src', and 'xtend-gen'. The 'model-input' folder is circled in red. Below the dialog box, there is a section titled 'EVENT CHAIN MODEL' with 'Calculate' and 'Reset' buttons. This section contains several input fields for different reaction types: 'Reaction (Implicit)', 'Reaction (LET)', 'Initial Reaction (Implicit)', 'Initial Reaction (LET)', 'Age (Implicit)', 'Age (LET)', 'Single-core Initial Reaction', and 'Single-core Worst-case Reaction'. Each of these sections has 'Best-Case' and 'Worst-Case' input fields. A 'Task Chain >' link is also visible.

Navigate to the folder where the target Amalthea model file is located.

4. Select & Open Amalthea



Select and open an Amalthea file. In this example, a multi-core Amalthea model is chosen.

5. Amalthea Model Loaded

The screenshot displays the APPARTA software interface. The top window title is 'APP4RTA'. The main area is titled 'Amalthea Model' and shows a loaded model 'ChallengeModel_TCs.amxml'. A search bar is present. The left sidebar lists tasks with checkboxes and a 'Task Name' column, circled in red and labeled '1'. The central area shows a grid of task processing units, each with a task name and 'Response Time' label, circled in red and labeled '2'. Below the grid, there are various configuration options like 'Schedulability', 'Cumulated Memory-Access Cost', etc. At the bottom, an 'EVENT CHAIN MODEL' section contains several input fields for different reaction and age metrics, with 'Calculate' and 'Reset' buttons.

After a model is loaded, it shows all the tasks (1) and processing units (2) that the selected model has.

6. Integer Mapping

The screenshot displays the APP4RTA software interface. At the top, the window title is 'APP4RTA'. Below it, the 'Amalthea Model' section shows the file 'ChallengeModel_TCs.amxmi' and a 'Search Amalthea' button. The main interface is divided into two columns. The left column lists tasks with their priority values (P) and a column for integer mapping (IA). The right column shows response time boxes for various tasks, labeled with their priority and the unit they are mapped to (e.g., '0: Denver', '1: Denver', '2: A57', '3: A57', '4: A57', '5: A57', '6: GPU_def').

Key elements in the interface include:

- Task List (Left):** Tasks include OS_Overhead (P: 4), Lidar_Grabber (P: 1), DASM (P: 1), CANbus_polling (P: 3), EKF (P: 4), Planner (P: 0), PRE_SFM_gpu (P: 3), PRE_Localizati (P: 3), PRE_Lane_det (P: 3), PRE_Detection (P: 0), SFM (P: 6), Localization (P: 2), Lane_detection (P: 5), and Detection (P: 6).
- Configuration (Left):** Includes buttons for 'Default IA' (circled in red and labeled '1'), 'Enter IA', 'Calculate', and 'Reset'. There are also radio buttons for 'Synchronous' and 'Asynchronous', and options for 'Worst-Case', 'Average-Case', and 'Best-Case'. Below these are input fields for 'Schedulability', 'Cumulated Memory-Access Cost', 'Cumulated Contention', 'Computation', and 'Response Time Sum'.
- Response Time Boxes (Right):** A grid of boxes for tasks 0 through 6, each with a 'Response Time' label. The boxes are currently empty.
- EVENT CHAIN MODEL (Bottom):** Includes 'Calculate' and 'Reset' buttons, and a 'Task Chain >' link. Below are input fields for 'Reaction (Implicit)', 'Reaction (LET)', 'Initial Reaction (Implicit)', 'Initial Reaction (LET)', 'Age (Implicit)', 'Age (LET)', 'Single-core Initial Reaction', and 'Single-core Worst-case Reaction', each with 'Best-Case' and 'Worst-Case' sub-sections.

When the 'Default IA' (1) button is clicked, each task's box (2) is automatically filled with an integer number. This indicates that a task is about to be mapped to the corresponding identity number of processing unit. One can also write an integer number in each box manually. The 'Default IA' means an integer array to map all the tasks to processing units and that is specifically designed to make the 'ChallengeModel_TCs.amxmi' model schedulable. Therefore it is always possible that it does not serve for other multi-core models. However, the 'Default IA' would only contain numbers of 0 when a single-core model is loaded.

7. Assign Tasks to Processing Units

The screenshot displays the APP4RTA software interface. On the left, a task list is shown with columns for 'Task Name' and 'PU Num'. The 'Enter IA' button is circled in red and labeled with a red '1'. Below the task list are buttons for 'Default IA', 'Calculate', and 'Reset'. The main area shows seven processing units (0: Denver, 1: Denver, 2: A57, 3: A57, 4: A57, 5: A57, 6: GPU_def) arranged in two rows. Each unit has a 'Response Time' column and a list of tasks mapped to it. A large red oval labeled with a red '2' encompasses the task lists and response time columns for all units. At the bottom, the 'EVENT CHAIN MODEL' section contains various reaction and age parameters with input fields for Best-Case, Worst-Case, and Critical-Case scenarios.

When the 'Enter IA' (1) button is clicked, each task is mapped to the corresponding processing unit (2). Since there are 7 processing units in the 'ChallengeModel_TC.amxmi' model, it shows 7 pairs of lists. The list on the left side of each pair is for listing names of the tasks that are mapped to the corresponding processing unit while one on the right side is for listing response times of the corresponding tasks.

8. Measure Response Time

The screenshot displays the APP4RTA software interface. On the left, a task list includes OS_Overhead, Lidar_Grabber, DASM, CANbus_polling, EKF, Planner, PRE_SFM_gpu..., PRE_Localizati..., PRE_Lane_det..., PRE_Detection..., SFM, Localization, Lane_detection, and Detection. The 'Synchronous' mode is selected (1), and the 'Worst-Case' execution case is chosen (2). The 'Calculate' button is highlighted (3). The main area shows a grid of task response times for different processors (Denver, A57, GPU_def). A large red oval (4) encompasses the response time data. Below this, a summary section (5) provides overall metrics: Schedulability (Scheduleable!), Cumulated Memory-Access Cost (5361668000 ps), Cumulated Contention (24795710000 ps), Computation (635075050500 ps), and Response Time Sum (665232428500 ps). The bottom section, 'EVENT CHAIN MODEL', contains various reaction and age parameters for different cases.

Task Name	PU Num	0: Denver	1: Denver	2: A57	3: A57
Planner	4	13358534500	1302430000 p	392590097500	602880000 ps
PRE_Detection	1	73565439500	18265272000		2071995000
Lidar_Grabber	1				0 ps (GPU Ta)
DASM	3				0 ps (GPU Ta)
CANbus_polling	3				0 ps (GPU Ta)
EKF	4				
Planner	0				
PRE_SFM_gpu...	3				
PRE_Localizati...	3				
PRE_Lane_det...	3				
PRE_Detection...	0				
SFM	6				
Localization	2				
Lane_detection	5				
Detection	6				

Task Name	PU Num	4: A57	5: A57	6: GPU_def
EKF	3	4788430000 p	56045200000	2000000000 p
OS_Overhead	0	73942150000		2000000000 p

Metric	Value
Schedulability	Scheduleable! :)
Cumulated Memory-Access Cost	5361668000 ps
Cumulated Contention	24795710000 ps
Computation	635075050500 ps
Response Time Sum	665232428500 ps

(1) Choose the offloading mode between 'Synchronous' case and 'Asynchronous' case. (2) Choose the execution case between 'Worst-', 'Average-', and 'Best-Case'. (3) By clicking the 'Calculate' button, each task's response time is calculated and printed on the right list of each list pair (4). All analysis results appear in (5) which include: 'Schedulability', 'Cumulated Memory-Access Cost', 'Cumulated Contention', 'Computation', and 'Response Time Sum'.

9. Task Chain Analysis

The screenshot displays the APPARTA software interface for task chain analysis. The top section shows a list of tasks and their assigned processing units (PUs). The central area contains a grid of task response times for various cores (0: Denver, 1: Denver, 2: A57, 3: A57, 4: A57, 5: A57, 6: GPU_def). The bottom section, titled 'EVENT CHAIN MODEL', shows a selected task chain 'LI-Lo-EK-P-DA' and a 'Calculate' button. Below this, the results of the analysis are displayed, including 'Reaction (Implicit)', 'Reaction (LET)', 'Initial Reaction (Implicit)', 'Initial Reaction (LET)', 'Age (Implicit)', 'Age (LET)', and 'Single-core Initial Reaction' and 'Single-core Worst-case Reaction'.

Key elements highlighted in the image:

- 1**: Selection of the task chain 'LI-Lo-EK-P-DA' in the 'EVENT CHAIN MODEL' dropdown.
- 2**: Clicking the 'Calculate' button to initiate the analysis.
- 3**: The resulting task chain visualization on the right, showing the sequence of tasks: Lidar_Grabber, Localization, EKF, Planner, and DASM.
- 4**: The 'Reaction (Implicit)' and 'Reaction (LET)' results, showing best-case and worst-case values in picoseconds (ps).
- 5**: The 'Single-core Initial Reaction' and 'Single-core Worst-case Reaction' results, which are marked as 'Not Single-Core'.

Now that every task's response time is measured, it is possible to measure end-to-end task chain latency with the derived task response times. (1) To analyze end-to-end task chain latency, a task chain in the combo-box should be selected first. (2) Click the 'Calculate' button, then the selected task chain would be illustrated (3) and all measurement results would also be printed out (4)(5). Since the observed Amalthea model is a multi-core model here, the single-core analysis results are not available (5).

10. Change The Model

The screenshot displays the APPARTA software interface. On the left, a task list includes OS_Overhead, Lidar_Grabber, DASM, CANbus_polling, EKF, Planner, PRE_SFM_gpu..., PRE_Localizati..., PRE_Lane_det..., PRE_Detection..., SFM, Localization, Lane_detection, and Detection. The main area shows a table of tasks with columns for core (0: Denver, 1: Denver, 2: A57, 3: A57), task name, and response time. An 'Open File' dialog box is overlaid, showing the file 'ChallengeModel_SingleTCs.amxmi' selected. The 'Open' button in the dialog is circled in red. The background shows various performance metrics such as 'Cumulated Memory-Access Cost' and 'Cumulated Contention'.

It is possible to change the observed model without clicking the 'Reset' buttons. Apply the same process but this time with the 'ChallengeModel_SingleTCs.amxmi' file that is a single-core Amalthea model (1) (2) (3).

11. Single-core RTA

The screenshot displays the APP4RTA software interface. The main window is titled 'APP4RTA' and shows the 'Amalthea Model' configuration for 'ChallengeModel_SingleTCs.amxmi'. The interface is divided into several sections:

- Task Configuration:** A table lists four tasks (Task0, Task1, Task2, Task3) with their respective PU Num (0) and Response Time (100000000000 ps, 200000000000 ps, 300000000000 ps, 900000000000 ps).
- Configuration Options:** Includes buttons for 'Default IA', 'Enter IA', and radio buttons for 'Synchronous' (selected), 'Asynchronous', 'Worst-Case', 'Average-Case', and 'Best-Case'. There are also 'Calculate' and 'Reset' buttons.
- Summary Metrics:** Shows 'Schedulability' (Scheduleable!), 'Cumulated Memory-Access Cost' (0 ps), 'Cumulated Contention' (0 ps), 'Computation' (15000000000000 ps), and 'Response Time Sum' (15000000000000 ps).
- EVENT CHAIN MODEL:** A section with 'Calculate' and 'Reset' buttons.
- Task Chain:** A list of tasks assigned to cores: 1: Core1 (Denver) - Lidar_Grabber; 2: Core2 (A57) - Localization; 3: Core4 (A57) - EKF; 4: Core0 (Denver) - Planner; 5: Core1 (Denver) - DASM.
- Performance Metrics:** A grid of input fields for 'Reaction (Implicit)', 'Reaction (LET)', 'Initial Reaction (Implicit)', 'Initial Reaction (LET)', 'Age (Implicit)', 'Age (LET)', 'Single-core Initial Reaction', and 'Single-core Worst-case Reaction'. Most fields are empty or show 'Not Single-Core'.

The 'ChallengeModel_SingleTCs.amxmi' model only has one processing unit with four tasks. As it is already mentioned, the 'Default IA' only contains numbers of 0 because a single-core model is loaded this time. The process is the same.

12. Single-core Task Chain Analysis

The screenshot displays the APP4RTA software interface for task chain analysis. The top section shows the 'Amalthea Model' configuration for 'ChallengeModel_SingleTCs.amxml'. It includes a search bar and a table of task response times for core 0 (A57):

Task Name	Response Time
Task0	100000000000 ps
Task1	200000000000 ps
Task2	300000000000 ps
Task3	900000000000 ps

Below the table, there are radio buttons for 'Synchronous' (selected), 'Asynchronous', 'Worst-Case', 'Average-Case', and 'Best-Case'. A 'Calculate' button is present. The 'Schedulability' section shows 'Scheduleable! :)'. The 'Cumulated Memory-Access Cost' is 0 ps, and 'Cumulated Contention' is 0 ps. The 'Computation' is 15000000000000 ps, and the 'Response Time Sum' is 16000000000000 ps.

The 'EVENT CHAIN MODEL' section is highlighted with a red circle. It shows the model 'EC_10-5-6-3' and a 'Task Chain' for Core0 (A57) with tasks in the following order: Task3, Task1, Task2, Task0. Below this, there are several reaction analysis sections:

Reaction Type	Best-Case	Worst-Case
Reaction (Implicit)	Not Multi-Core.	Not Multi-Core.
Reaction (LET)	Not Multi-Core.	Not Multi-Core.
Initial Reaction (Implicit)	Not Multi-Core.	Not Multi-Core.
Initial Reaction (LET)	Not Multi-Core.	Not Multi-Core.
Age (Implicit)	Not Multi-Core.	Not Multi-Core.
Age (LET)	Not Multi-Core.	Not Multi-Core.
Single-core Initial Reaction	16000000000000 ps	19000000000000 ps
Single-core Worst-case Reaction	8000000000000 ps	15000000000000 ps
	16000000000000 ps	23000000000000 ps

Now that every task's response time is measured, it is possible to measure end-to-end task chain latency with the derived task response times. The process is the same. However, a single-core model is analyzed this time. Therefore, latency results regarding single-core are only available while multi-core results are not in this case.