

APP4RTA

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

The screenshot displays the APP4RTA software interface, which is used for analyzing response time and end-to-end event-chain latency. The interface is divided into several sections:

- Task Configuration (Left Panel):** Lists tasks and their PU numbers. Tasks include OS_Overhead (4), Lidar_Grabber (1), DASM (1), CANbus_polling (3), 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), and Detection (6). Buttons for "Default IA", "Enter IA", "Calculate", and "Reset" are present.
- Analysis Results (Right Panel):** Displays response times for various tasks across different processors (0: Denver, 1: Denver, 2: A57, 3: A57, 4: A57, 5: A57, 6: GPU_def). For example, the "Planner" task on a Denver processor has a response time of 13358534500 ps, while the "SFM Detection" task on a GPU_def processor has a response time of 20000000000 ps.
- Event Chain Model (Bottom Panel):** Shows the "CA-EK-P-DA" model. It includes sections for "Direct & Implicit Communication Paradigm", "LET Communication Paradigm", "Data Age", "Task Chain Age (Direct & Implicit)", "Early Reaction (Direct & Implicit)", and "Task Chain". The "Task Chain" section lists tasks for different cores: 1: Core3 (A57) - CANbus_polling, 2: Core4 (A57) - EKF, 3: Core0 (Denver) - Planner, and 4: Core1 (Denver) - DASM.

Before executing the code, please install the Java GUI software.

➤ To install Java GUI softwares:

1. Eclipse > `Help`
2. `Install New Software` > Work with: Eclipse Repository
(<http://download.eclipse.org/releases/oxygen>)
3. `General Purpose Tools` > all click from `Swing Designer` to `WindowBuilder XML Core` (requires Eclipse WTP/WST)
4. `Next` > `Next` > `accept` > `Finish`

APP4RTA

AMALTHEA MODEL

Search Amalthea

APP4RTA

Task Name PU Num

1. Select an Amalthea Model

Response Time & Mapping Analysis Part

EVENT CHAIN MODEL

Calculate Reset

Direct Implicit

Direct & Implicit Communication Paradigm

WC Reaction

BC Reaction

Task Chain Age (Direct & Implicit)

WC Age

BC Age

Task Chain >

LET Communication Paradigm

WC Reaction

BC Reaction

Early Reaction (Direct & Implicit)

WC E-Rct

BC E-Rct

Data Age

Contained Labels

Worst-case Age

Best-case Age

End-to-End Event-Chain Latency Part

Run `APP4RTA.java` in `org.eclipse.app4mc.gsoc_rta.ui` package, then this window will show up.

Based on the horizontal line on the middle, the upper part is for response Time & 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.

1. The window browser for searching Amalthea models shows up when the `Search Amalthea` button clicked.

The list on the left side of each pair is for listing names of the tasks which are mapped to the corresponding processing unit while one on the right side is for listing response times of the corresponding tasks.

Basically, we can map the tasks with these boxes by entering the number of each processing unit which is stated on the top of the lists on the left-side.

4. The user can either manually type numbers for every box or simply click the 'Default IA' button which would automatically fill up every box with the pre-defined integer array values.

5. Once every 'PU Num' box is filled, click 'Enter IA' button to assign tasks to processing units according to each integer value. Once this is done, the mapped tasks would appear on the left-side lists.

6. Choose the offloading mode between 'Synchronous' case and 'Asynchronous' case.

7. Choose the execution case between 'Worst' case and 'Average' case and 'Best' case.

8. By clicking the 'Calculate' button, all calculation results will be printed out on the text-fields ('Schedulability', 'Cumulated Memory-Access Cost', 'Cumulated Contention', 'Computation').

The screenshot displays the APP4RTA software interface, specifically the AMALTHEA MODEL configuration window. The window is titled 'APP4RTA' and 'AMALTHEA MODEL ChallengeModel_TCs.amxml'. It features a search bar and a 'Search Amalthea' button.

Task Name / PU Num: A list of tasks is shown on the left, each with a corresponding 'PU Num' box. The tasks are: OS_Overhead (4), Lidar_Grabber (1), DASM (1), CANbus_polling (3), 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), and Detection (6). Buttons for 'Default IA', 'Enter IA', 'Synchronous', 'Asynchronous', 'Worst-Case', 'Average-Case', and 'Best-Case' are present.

Calculation Results: The results are displayed in a grid format. The first row shows results for 0: Denver, 1: Denver, 2: A57, and 3: A57. The second row shows results for 4: A57, 5: A57, and 6: GPU_def. The results include 'Response Time' and 'Cumulated Memory-Access Cost'.

9. Choose an Event-Chain: A red circle highlights the 'EVENT CHAIN MODEL' section. It includes a dropdown menu with options: LD-P-DA, SF-P-DA, CA-P-DA, CA-EK-P-DA, CA-Lo-EK-P-DA, LI-Lo-EK-P-DA, LI-P-DA, and D-P-DA. Below the dropdown are buttons for 'Calculate' and 'Reset', and radio buttons for 'Direct' and 'Implicit'.

Task Chain Age (Direct & Implicit): This section includes input fields for 'WC Age', 'BC Age', and 'E-Rct'.

Data Age: This section includes input fields for 'Worst-case Age' and 'Best-case Age'.

The event-chain combo-box becomes visible once the user clicks `Enter IA` to assign tasks to processing units according to each integer value in the boxes.

9. To analyze end-to-end event-chain latency, an event-chain in the combo-box should be selected first.

The screenshot shows the CA-EK-P-DA software interface. At the top, the 'EVENT CHAIN MODEL' is set to 'CA-EK-P-DA'. There are 'Calculate' and 'Reset' buttons. Below these, there are radio buttons for 'Direct' and 'Implicit' communication paradigms, with 'Implicit' selected. A red annotation '10. Select a Communication Paradigm' points to the 'Implicit' button. Below the radio buttons, there are several text fields for reaction and age calculations, grouped under 'Direct & Implicit Communication Paradigm', 'Task Chain Age (Direct & Implicit)', 'LET Communication Paradigm', and 'Early Reaction (Direct & Implicit)'. A red annotation '11. Click the `Calculate` Button.' points to the 'Calculate' button. On the right side, there is a 'Task Chain >' section with a list of tasks: '1: Core3 (A57)', 'CANbus_polling', '2: Core4 (A57)', 'EKF', '3: Core0 (Denver)', 'Planner', '4: Core1 (Denver)', and 'DASM'. At the bottom, there is a 'Data Age' section with three columns: 'Contained Labels', 'Worst-case Age', and 'Best-case Age'. The 'Contained Labels' column lists various data labels like 'Occupancy_grid_host', 'Vehicle_status_host', 'x_car_host', 'y_car_host', 'yaw_car_host', 'vel_car', 'yaw_rate', 'steer_objective', and 'steer_objective'. The 'Worst-case Age' and 'Best-case Age' columns show corresponding numerical values in picoseconds (ps) and nanoseconds (ns).

10. Select the communication paradigm between direct Communication and implicit communication.

11. Finally, click the `Calculate` button.

Then all calculation results regarding reaction, age of data, task-chain in the worst and best cases will be printed out to the corresponding text fields or lists.