

Methodology and Toolset for Timing Analysis

Luis Miguel Pinho & Tiago Carvalho

CERCIRAS Training School 2023: Advanced Topics in Resource-Aware
Computing

Goals

- Learn mechanisms for time analysis of parallel applications
- Analyze parallel applications with the provided tools
- Understand the benefits of time analysis for parallel applications

Pre-requirements

- Docker Engine: [Install Docker Engine](#)
- Docker image provided in the course
 - Build + run scripts also provided
- Suggested tools in host machine:
 - Visual Studio Code with suggested C/C++ plugins: [Visual Studio Code - Mac, Linux, Windows](#)
 - Docker plugin to easily work inside the docker container
 - [Docker extension](#)
 - Graphviz (dot) plugin for VS Code for task-dependency graph visualization
 - [Graphviz \(dot\) language support](#)
 - Chrome web browser to visualize results

Building the Docker Image

1. Open a terminal (either in windows or ubuntu)
2. Enter the resources folder containing the docker image

```
tdrc@timing_analysis_tools$ dir
Dockerfile  build  build.bat  course  run  run.bat
```

3. Build the docker image by running the “build” script (build.bat for windows)
 - The compilation will take some time (depending on the platform, ~30 min)
 - Let’s skip to the next slides while it is building...

Tools used for Profiling and Timing Analysis

- **PAPI**: API for runtime performance counter extraction (Extrae dependency)
- **Extrae**: monitor and generate traces of parallel applications
- **BSC's LLVM**: adapted version of LLVM able to generate task-dependency graphs (TDGs) and specify static task-to-thread mapping
- **tdg-instrumentation-script**: converts extrae's output to json format
- **Time-predictability**: framework for the analysis of TDGs and calculate timing-related metrics

Time Analysis Flow

Essentially divided in three phases:

1. Compilation

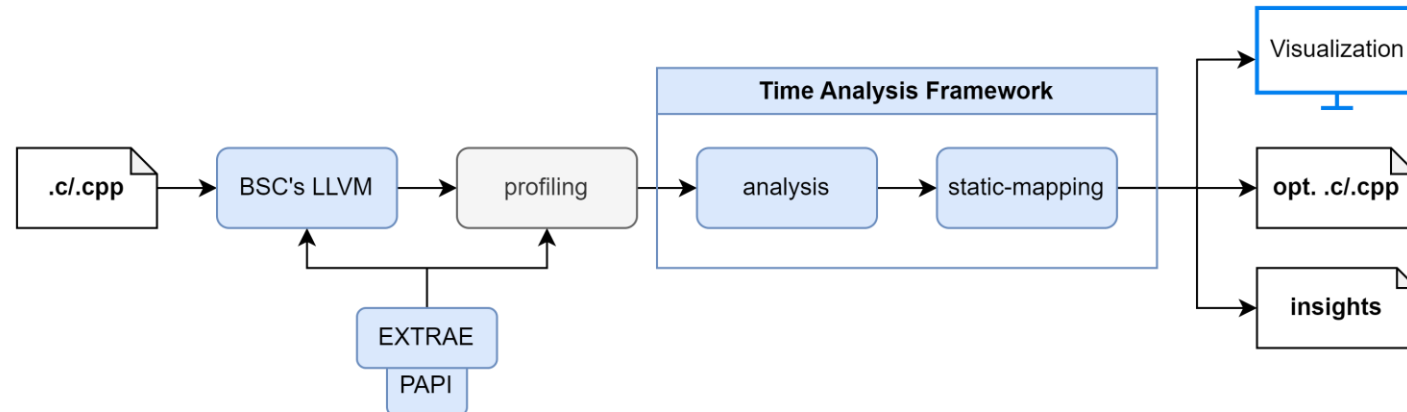
- Compile instrumented code ready to be profiled

2. Profiling

- Execute application several times to extract measurements

3. Analysis

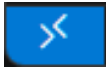
- Extract timing metrics from the measurements, analyse results, “optimize”



Running the Docker container

1. Open a terminal (either in windows or ubuntu)
2. Execute the “run” script (run.bat for windows)

```
tdrc@timing_analysis_tools$ ./run  
cts@92108d85862a:~/course$
```

- This will create a docker container, and keep an open terminal to use
 - For the rest of the course, we suggest the use of VSCode:
 - assuming the installation of the docker plugin
1. Click on the bottom-left button with the icon 
 2. On the top-middle region that appears, select “Attach to running container...”
 3. Select the newly created container
 4. Vscode will open a new window that is working inside the container
 - We suggest the use of “Open Folder” and select the folder “/home/cts/course/”
 - This will open that folder in the explorer view