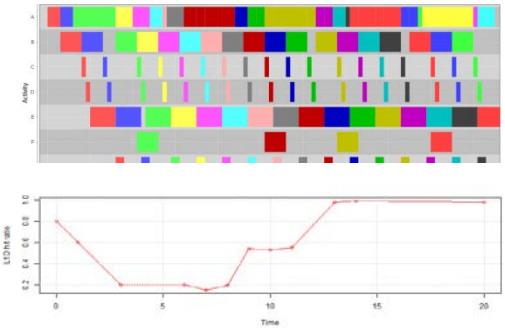
Visualization and analysis of data caching

Description

Many embedded systems are data intensive. Examples include the video processing system in an interventional X-ray machine, and the processing of image data in the datapath of high-end printers. Often, these systems are implemented, at least partially, in software which runs on multi-core embedded platforms. Furthermore, there are strict performance requirements, e.g., on the throughput of the system. Clearly, the software implementation has significant impact on the system performance. One key aspect in these data-intensive systems is how the software implementation uses the hardware data caches. An actual assessment whether cache usage can be optimized and what its system-level performance benefits are, however, is difficult because it requires the user to gather detailed information from several sources through specialized tools (e.g., Intel VTune), and to manually combine information for the final interpretation.

The assignment



The goal of this assignment is to provide an integrated approach to the above-sketched problem based on two types of execution traces of the system: (1) a memory reference trace, and (2) an activity trace. Memory reference traces contain very detailed, low-level information on data access. Activity traces contain system-level activities which can be visualized with the TRACE tool (http://trace.esi.nl/). The following questions are addressed in this assignment:

- What is relevant information in the memory reference trace, and how does it depend on the hardware architecture (e.g., memory hierarchy)? Models of the memory hierarchy may play a crucial role in answering this question.
- How can we extract this relevant information, relate it to the activity trace, and how can we visualize it in a meaningful way? For instance, the figure above sketches an approach in which some state variable (e.g, L1 data cache hit ratio) evolves with system activities (shown as a Gantt chart).
- Can we develop automated analysis techniques based on the extended information? For instance, can we link activities on the critical path (existing analysis technique) to L1 data cache misses?

You can start after the summer vacation (i.e., after August 29). Depending on your interest, the assignment can have the form of an internship at Océ, TU/e or ESI. Since a significant part of the assignment involves extension of the TRACE tool, we expect a solid knowledge of the Java programming language.

More information

- Prof. dr. ir. Twan Basten (A.A.Basten@tue.nl)
- Dr. Martijn Hendriks (martijn.hendriks@tno.nl)