Modern processors have sophisticated hardware performance monitoring units (PMUs). These PMUs are capable of measuring and monitoring a wealth of different metrics and events during an application’s execution at very little cost. Starting with the Nehalem micro-architecture (Core i3/i5/i7), the PMUs of Intel processors also include a Last Branch Record (LBR) for each hardware thread. The LBR is a circular buffer with sixteen entries, which, when enabled, always contains the recent branching history of the thread.

Software can enable and disable the LBR as well as configure the kinds of branches that it records. When recording calls and returns, the LBR contents can be used to reconstruct part of the call stack. Frequently sampling the LBR contents allows building a call tree and identifying frequently called methods, which has been part of our recent research.

The task for this thesis is to look at “regular” branches used for control flow constructs such as if-then-else, switch-case, or loops. Recording such branches with the LBR allows identifying partial paths taken within a method. Sampling a significant amount of these partial paths should reveal particularly “hot” as well as potentially unused code blocks in the methods of an application.

The described method is to be implemented for the Oracle Hotspot Java virtual machine, complete with a simple, but helpful representation of the gained information. As a simplification, the implementation may be limited to code generated by Hotspot’s less complex C1 JIT compiler (“client VM”).

Dipl.-Ing. Peter Hofer