http://www.sc.informatik.tu-darmstadt.de/edu/abschlussarbeiten/abschlussarbeiten.en.jsp

Thesis-topic (BSc/MSc):
Hybrid Performance Measurement Techniques

Hinweis: Dieses Thema kann sowohl in englischer als auch in deutscher Sprache bearbeitet werden.

Background: All performance tools for HPC systems require sampling or instrumentation to gather data. In sampling approaches, usually the measurement tool stops the target application in fixed intervals or at specific events and samples the execution pointer along with the applications' stack-frames. At this point, additional information about hardware and program state can also be recorded. By unwinding the stack-frames, the tool is then able to roughly pinpoint the corresponding source position from the debug information.
In the instrumentation approach, the target program is augmented with measurement points, called probes, to gather the necessary information. These probes are typically implemented as calls to a tool specific measurement library, called hooks. The most widely used form of instrumentation installs probes at all entries and exits of a given function.
Both sampling and instrumentation are, in principle, capable of gathering the same information, though they differ in their ability to reliably gather all events of interest. As sampling is performed outside of the scope of the observed application, important events occurring in between sampling points may potentially be missed. In contrast, an event, once instrumented, will always be observed though automatic placement of event-probes difficult to control and has tendencies to fail for current C++ codes due to immense overhead.

Sampling, e.g. the event based controlled interruption of the execution in order to acquire the current state of the application, is easy to control though expensive at runtime and provides only statistical information. Instrumentation, the direct modification of the target application with measurement codes, provides deterministic and reproducible results, though is difficult to control and has tendencies to fail for current C++ codes in terms of overhead.

A hybrid approach, here the combination of instrumentation and sampling, promises to provide the best of both worlds.
Your goal for this thesis-topic is the creation and development of a hybrid instrumentation and sampling approach, capable of controlling the sampling rate from within the application by instrumentation, but also to support the sampling by providing query function for the sampling mechanism to exploit. Your thesis should also include a comparative study of your developed approach.

Several tools exist and can be used for this thesis:

- The source-to-source compiler framework ROSE
- The Pin - Dynamic Binary Instrumentation Tool
- Paradyn/Dyninst

Requirements (or the disposition to acquire the necessary knowledge):

- Good command of C/C++ and Assembly,
- Basic understanding of Unix system programming
- Basic understanding of the C/C++ runtime

Contact:

Christian.Iwainsky@sc.tu-darmstadt.de
06151 16-75573