In-depth Performance Visualization of Hybrid MPI/Cell Applications

Holger Brunst, Daniel Hackenberg

Center for Information Services and HPC

Technische Universität Dresden

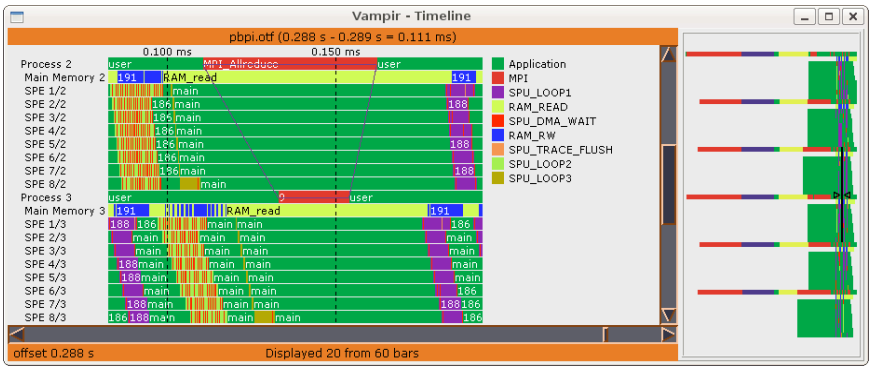
01062 Dresden, Germany

E-mail: {holger.brunst, daniel.hackenberg}@tu-dresden.de

Event-based software tracing and visualization is a widely accepted technique for developing and optimizing parallel high performance applications. Its high level of detail provides valuable and often indispensable information to application developers. This presentation is about software tracing on the Cell Broadband Engine, which introduces new opportunities but also challenges for application developers. The presented work supports developers with a hybrid program tracing and visualization methodology, which includes a rich set of performance relevant events from the Cell's SIMD-cores.

We identified the following Cell specific event classes to be very important for optimization purposes and included them in our monitor infrastructure: enter and exit of SPE user functions, start and completion (logical) of DMA transfers, and send and receive of mailbox messages. Instrumented SPE programs generate performance events at runtime. Transfers of the SPE event data to off-chip memory are hidden to avoid unnecessary program perturbation. A final processing of the data and the trace file creation is done after the program has finished. The overhead for SPE program tracing is small and usually stays below 3%.

An example of our proposal for Cell trace visualization is the timeline chart depicted to the right. The PPE processes and their SPE threads are individually assigned to a horizontal bar, similar to the processes of MPI parallel programs. An additional horizontal bar is introduced that represents the main memory. Cell communication is depicted with connecting lines between the respective bars. Likewise, in-depth explanations of all supported cell performance events and their corresponding timeline and profiling charts will be provided.



Hybrid Cell/MPI application (2 of 6 MPI tasks with 8 SPE threads each)

In a nutshell, our talk presents a general concept for software tracing on the hybrid Cell architecture. The implemented performance monitor collects the following performance relevant data from the Cell SIMD cores: thread generation, DMA operations, mailbox communication and the invocation of SPE user functions. Additionally, it covers standard performance metrics like MPI communication, hardware counters, or resource tracking on the front-end core (PPE), which is the key to hybrid Cell/MPI support on large heterogeneous systems. Our concepts were implemented and verified in the scope of the robust Open Source Vampir-Trace performance monitor available at: http://www.tu-dresden.de/zih/cell/trace.