HPCToolkit

Exascale machines will be highly complex systems that couple multicore processors with accelerators and share a deep, heterogeneous memory hierarchy. Understanding performance bottlenecks within and across the nodes in extreme-scale computer systems is a first step toward mitigating them to improve library and application performance. The HPCToolkit project is providing a suite of software tools that developers need to measure and analyze the performance of their software as it executes on today's supercomputers and forthcoming exascale systems.

In recent years, the complexity and diversity of architectures for extreme-scale parallelism have dramatically increased. At the same time, the complexity of applications is also increasing as developers struggle to exploit billion-way parallelism, map computation onto heterogeneous computing elements, and cope with the growing complexity of memory hierarchies. While library and application developers can employ abstractions to hide some of the complexity of emerging parallel systems, performance tools must assess how software interacts with each hardware component of these systems.

The HPCToolkit project is working to develop performance measurement and analysis tools to enable application, library, runtime, and tool developers to understand where and why their software does not fully exploit hardware resources within and across nodes of current and future parallel systems. To provide a foundation for performance measurement and analysis, the project team is working with community stakeholders, including standards committees, vendors, and open-source developers, to improve hardware and software support for measurement and attribution of application performance on extreme-scale parallel systems.

The HPCToolkit team is focused on influencing the development of hardware and software interfaces for performance measurement and attribution by community stakeholders; developing new capabilities to measure, analyze, and understand the performance of software running on extreme-scale parallel systems; producing a suite of software tools that developers can use to measure and analyze the performance of parallel software as it executes; and working with developers to ensure that HPCToolkit's capabilities meet their needs. Using emerging hardware and software interfaces for monitoring code performance, the team is working to extend capabilities to measure computation, data movement, communication, and I/O as a program executes to pinpoint scalability bottlenecks, quantify resource consumption, and assess inefficiencies, enabling developers to target sections of their code for performance improvement.

Progress to date

- The team developed novel capabilities for measurement, analysis, and attribution of applications that employ GPU accelerators. Today, HPCToolkit can report performance about accelerated applications in source-code centric profiles views and time-centric visualizations of an execution's dynamics.

- To relate performance measurements of accelerated applications back to source code constructs, the team improved HPCToolkit's ability to recover control flow graphs from machine code, which enabled HPCToolkit to relate application performance to inline functions, templates, and loops in highly optimized code on both host processors and accelerators.

- The team added a new measurement substrate to HPCToolkit to measure code performance using the native Linux performance monitoring substrate known as the perf events interface. In addition to measuring application performance, Linux perf events enable HPCToolkit to measure operating system activity and thread blocking.

- The team developed support for handling programming models with short-lived dynamic threads.

PI: John Mellor-Crummey, Rice University

Collaborators: Rice University, University of Wisconsin – Madison