HPCTOOLKIT: tools for performance analysis of optimized parallel programs

Onur Cankur

HPCTOOLKIT: tools for performance analysis of optimized parallel programs

- Laksono Adhianto, Sinchan Banerjee, Mike Fagan, Mark Krentel, Gabriel Marin, John Mellor-Crummey, and Nathan R. Tallent
- Department of Computer Science, Rice University
- Oak Ridge National Laboratory
- Concurrency and Computation: Practice and Experience
- 2010

- Developed for their own use at Rice University.
- More papers on HPCToolkit: http://hpctoolkit.org/publications.html

Introduction

- HPCToolkit
  - Collecting performance measurements of fully optimized executables.
  - Analyzing application binaries to understand the structure of optimized code.
  - Correlating measurements with program structure.
  - Presenting the resulting performance data.

- Pinpoint performance and scalability bottlenecks in complex applications.

Motivation

- HPC systems and applications are very complex.
- Achieving top performance is important.

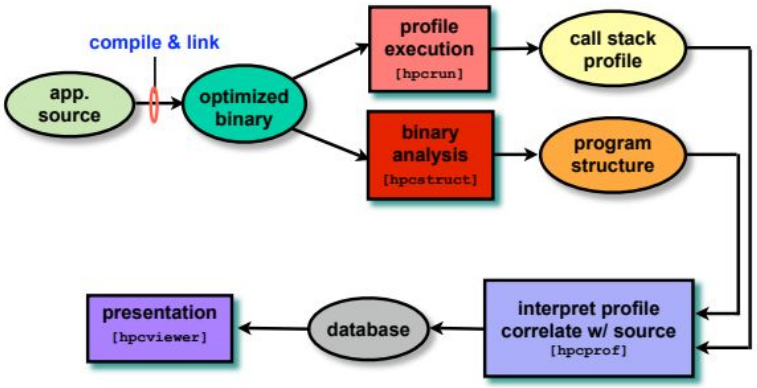
- Problems with tools at that time:
    - Relying on instrumentation and compromise measurement accuracy.
    - High overhead.
    - Not fully capable of correlating measurements with the source code.
    - Using call graph structure or not fully capable of understanding full calling context of optimized code.
    - Problem focused analysis

Measurement Methodology

- Scalable measurement and analysis
- Supports C, C++, and Fortran
  - Directly works with application binaries
- Avoid code instrumentation
  - Uses statistical sampling
- Avoid blind spots
  - Source code might not be available (e.g. math and communication libraries)
  - Performs binary analysis
- Calling context tree
- Multiple metrics
- Present in a hierarchical fashion

Performance Measurement - hpcrun

- Call path profiling and tracing
- Statistical sampling
- Coping with fully optimized binaries
  - Unwind the call stack at any point in a program’s execution
- Event triggers
  - Measure different aspects of the program performance.
    - Cache misses, I/O, memory allocations, etc.
- Control over parallel applications
  - Intercepts certain process control routines using library preloading.
- Handling dynamic loading
- Generate a measurement directory

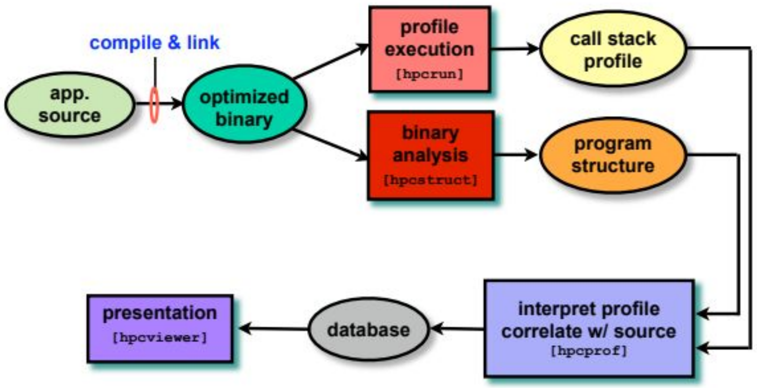


HPCToolkit workflow
http://hpctoolkit.org/pubs/cpe-2010-hpctoolkit.pdf

Analysis - hpcstruct & hpcprof

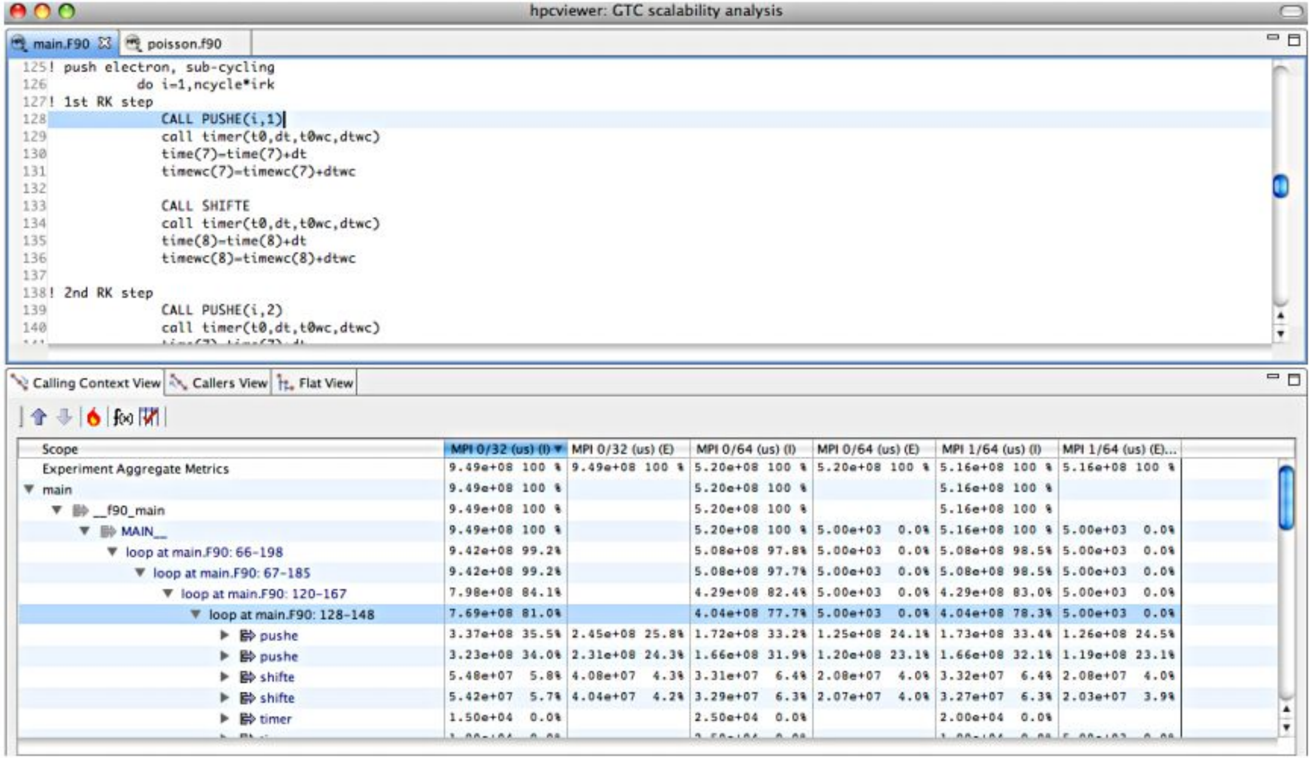
- hpcstruct
  - Recover the program structure using binary analysis
    - Mapping between object code and its associated source code structure
  - Generate a *.hpcstruct* file

- hpcprof
  - Attribute measurements to the application’s source code using the program structure file.
  - Generate a performance database directory



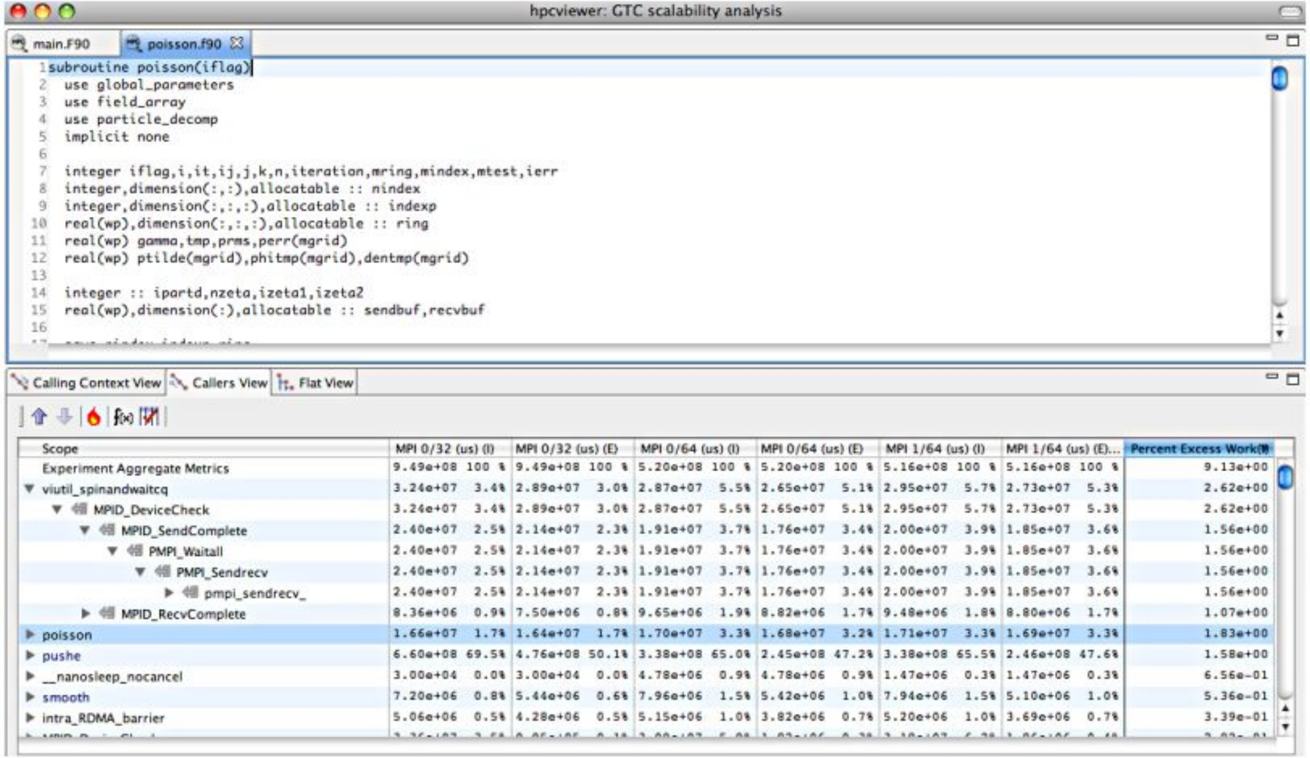
HPCToolkit workflow
http://hpctoolkit.org/pubs/cpe-2010-hpctoolkit.pdf

Presentation - hpcviewer



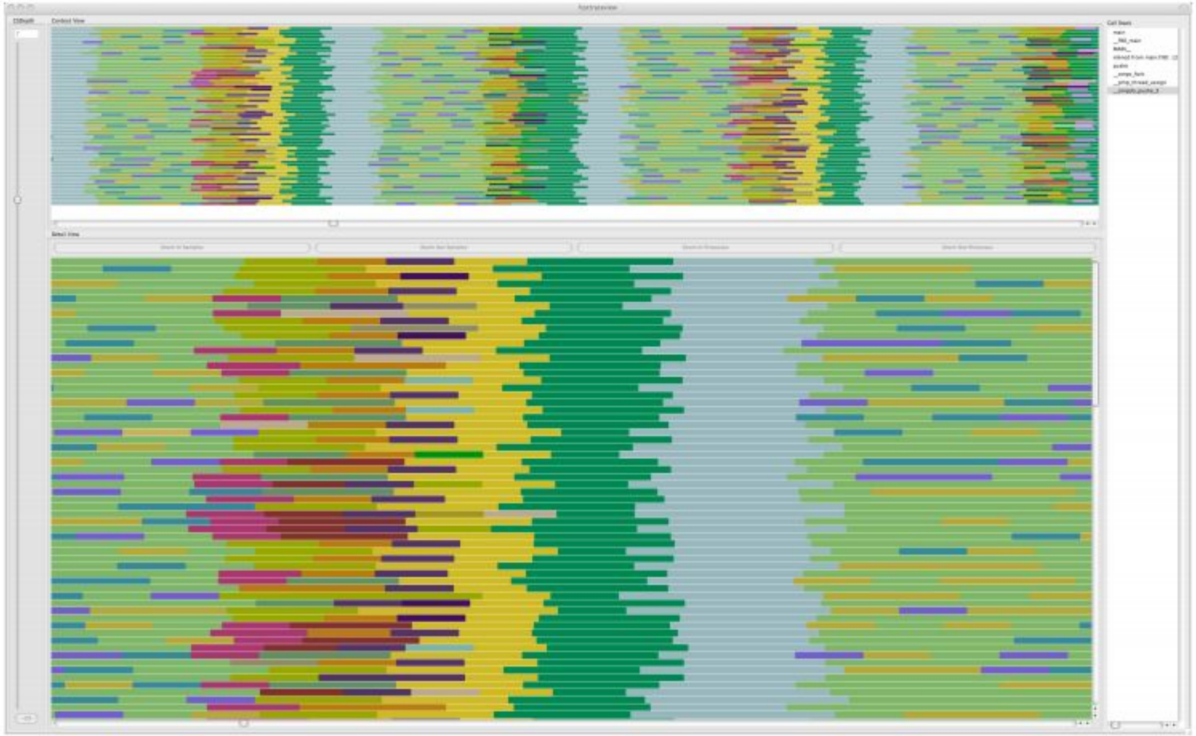
http://hpctoolkit.org/pubs/cpe-2010-hpctoolkit.pdf

Presentation - hpcviewer



http://hpctoolkit.org/pubs/cpe-2010-hpctoolkit.pdf

Presentation - hpctraceview



http://hpctoolkit.org/pubs/cpe-2010-hpctoolkit.pdf

Thank you for listening.

Q&A

User’s manual: http://hpctoolkit.org/manual/HPCToolkit-users-manual.pdf