

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

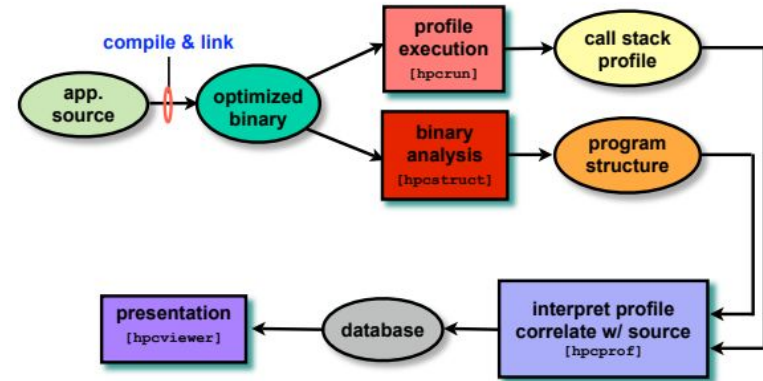
- HPC systems and applications are very complex.
- Achieving top performance is important.
- Problems with tools at that time:
 - Relying on instrumentation and compromise measurement accuracy.
 - High overhead.
 - Not fully capable of correlating measurements with the source code.
 - Using call graph 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

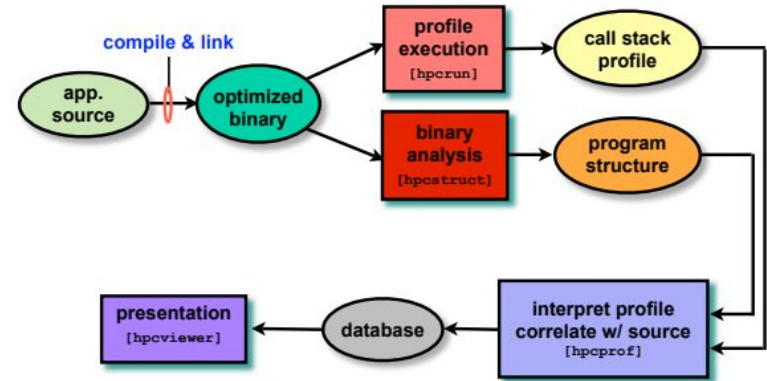


HPCToolkit workflow

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

Analysis - hpcstruct & hpcprof

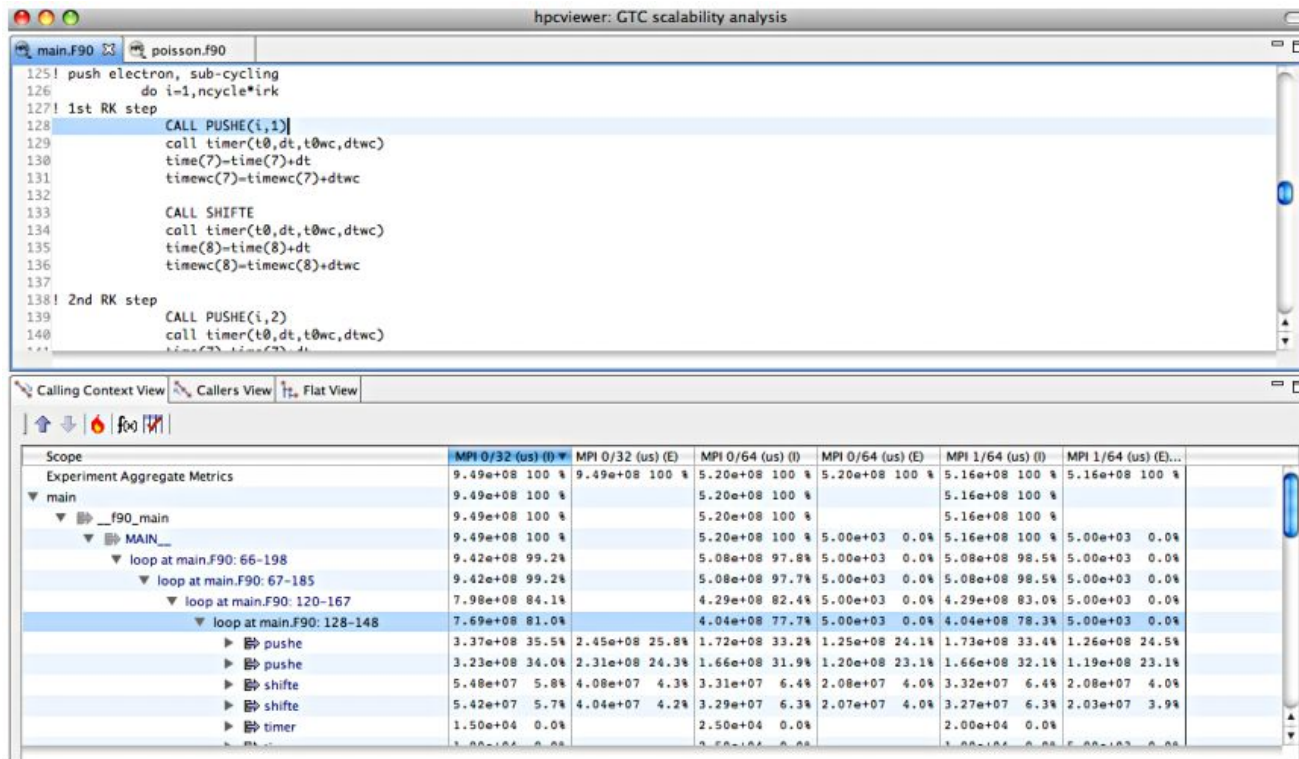
- hpcstruct
 - Recover the program structure using binary analysis
 - Mapping between object code and its associated source code structure
 - Generate a *.hpcstruct* file
- hpcprof
 - Attribute measurements to the application's source code using the program structure file.
 - Generate a performance database directory



HPCToolkit workflow

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

Presentation - hpcviewer



Presentation - hpcviewer

hpcviewer: GTC scalability analysis

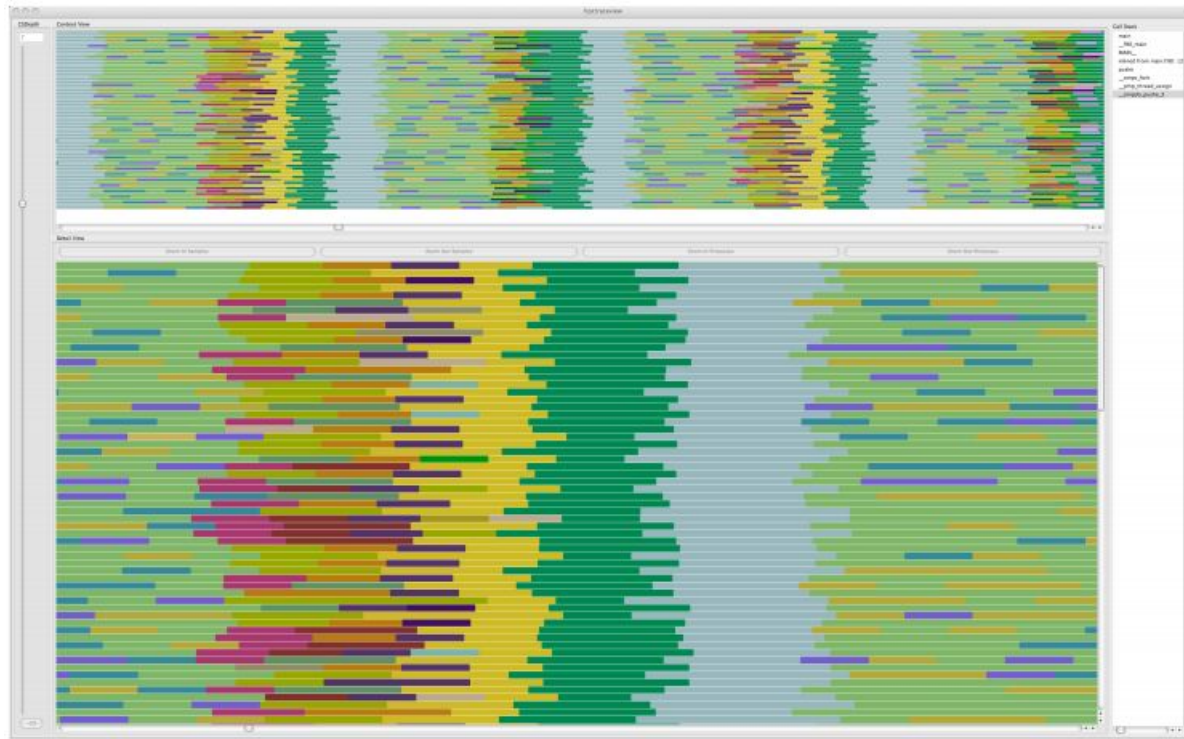
main.f90 poisson.f90

```
1 subroutine poisson(iflag)
2   use global_parameters
3   use field_array
4   use particle_decomp
5   implicit none
6
7   integer iflag,i,it,ij,j,k,n,iteration,mring,mindex,mtest,ierr
8   integer,dimension(:,,:),allocatable :: nindex
9   integer,dimension(:,,:),allocatable :: indexp
10  real(wp),dimension(:,,:),allocatable :: ring
11  real(wp) gamma,tmp,prms,perr(mgrid)
12  real(wp) ptilde(mgrid),phitmp(mgrid),dentmp(mgrid)
13
14  integer :: ipartd,nzeta,izeta1,izeta2
15  real(wp),dimension(:,,:),allocatable :: sendbuf,recvbuf
16
17  save nindex,indexp,nzeta,izeta1,izeta2
```

Calling Context View Callers View Flat View

Scope	MPI 0/32 (us) (I)	MPI 0/32 (us) (E)	MPI 0/64 (us) (I)	MPI 0/64 (us) (E)	MPI 1/64 (us) (I)	MPI 1/64 (us) (E)	Percent Excess Work
Experiment Aggregate Metrics	9.49e+08 100 %	9.49e+08 100 %	5.20e+08 100 %	5.20e+08 100 %	5.16e+08 100 %	5.16e+08 100 %	9.13e+00
viutil_spinandwaitcq	3.24e+07 3.4%	2.89e+07 3.0%	2.87e+07 5.5%	2.65e+07 5.1%	2.95e+07 5.7%	2.73e+07 5.3%	2.62e+00
MPID_DeviceCheck	3.24e+07 3.4%	2.89e+07 3.0%	2.87e+07 5.5%	2.65e+07 5.1%	2.95e+07 5.7%	2.73e+07 5.3%	2.62e+00
MPID_SendComplete	2.40e+07 2.5%	2.14e+07 2.3%	1.91e+07 3.7%	1.76e+07 3.4%	2.00e+07 3.9%	1.85e+07 3.6%	1.56e+00
PMPI_Waitall	2.40e+07 2.5%	2.14e+07 2.3%	1.91e+07 3.7%	1.76e+07 3.4%	2.00e+07 3.9%	1.85e+07 3.6%	1.56e+00
PMPI_Sendrecv	2.40e+07 2.5%	2.14e+07 2.3%	1.91e+07 3.7%	1.76e+07 3.4%	2.00e+07 3.9%	1.85e+07 3.6%	1.56e+00
pmpl_sendrecv	2.40e+07 2.5%	2.14e+07 2.3%	1.91e+07 3.7%	1.76e+07 3.4%	2.00e+07 3.9%	1.85e+07 3.6%	1.56e+00
MPID_RecvComplete	8.36e+06 0.9%	7.50e+06 0.8%	9.65e+06 1.9%	8.82e+06 1.7%	9.48e+06 1.8%	8.80e+06 1.7%	1.07e+00
poisson	1.66e+07 1.7%	1.64e+07 1.7%	1.70e+07 3.3%	1.68e+07 3.2%	1.71e+07 3.3%	1.69e+07 3.3%	1.83e+00
pushe	6.60e+08 69.5%	4.76e+08 50.1%	3.38e+08 65.0%	2.45e+08 47.2%	3.38e+08 65.5%	2.46e+08 47.6%	1.58e+00
_nanosleep_nocancel	3.00e+04 0.0%	3.00e+04 0.0%	4.78e+06 0.9%	4.78e+06 0.9%	1.47e+06 0.3%	1.47e+06 0.3%	6.56e-01
smooth	7.20e+06 0.8%	5.44e+06 0.6%	7.96e+06 1.5%	5.42e+06 1.0%	7.94e+06 1.5%	5.10e+06 1.0%	5.36e-01
intra_RDMA_barrier	5.06e+06 0.5%	4.28e+06 0.5%	5.15e+06 1.0%	3.82e+06 0.7%	5.20e+06 1.0%	3.69e+06 0.7%	3.39e-01

Presentation - hpctraceview



Thank you for listening.

Q&A