

Performance Insights on Blue Gene/Q with HPCToolkit

John Mellor-Crummey
Department of Computer Science
Rice University
johnmc@rice.edu



<http://hpctoolkit.org>



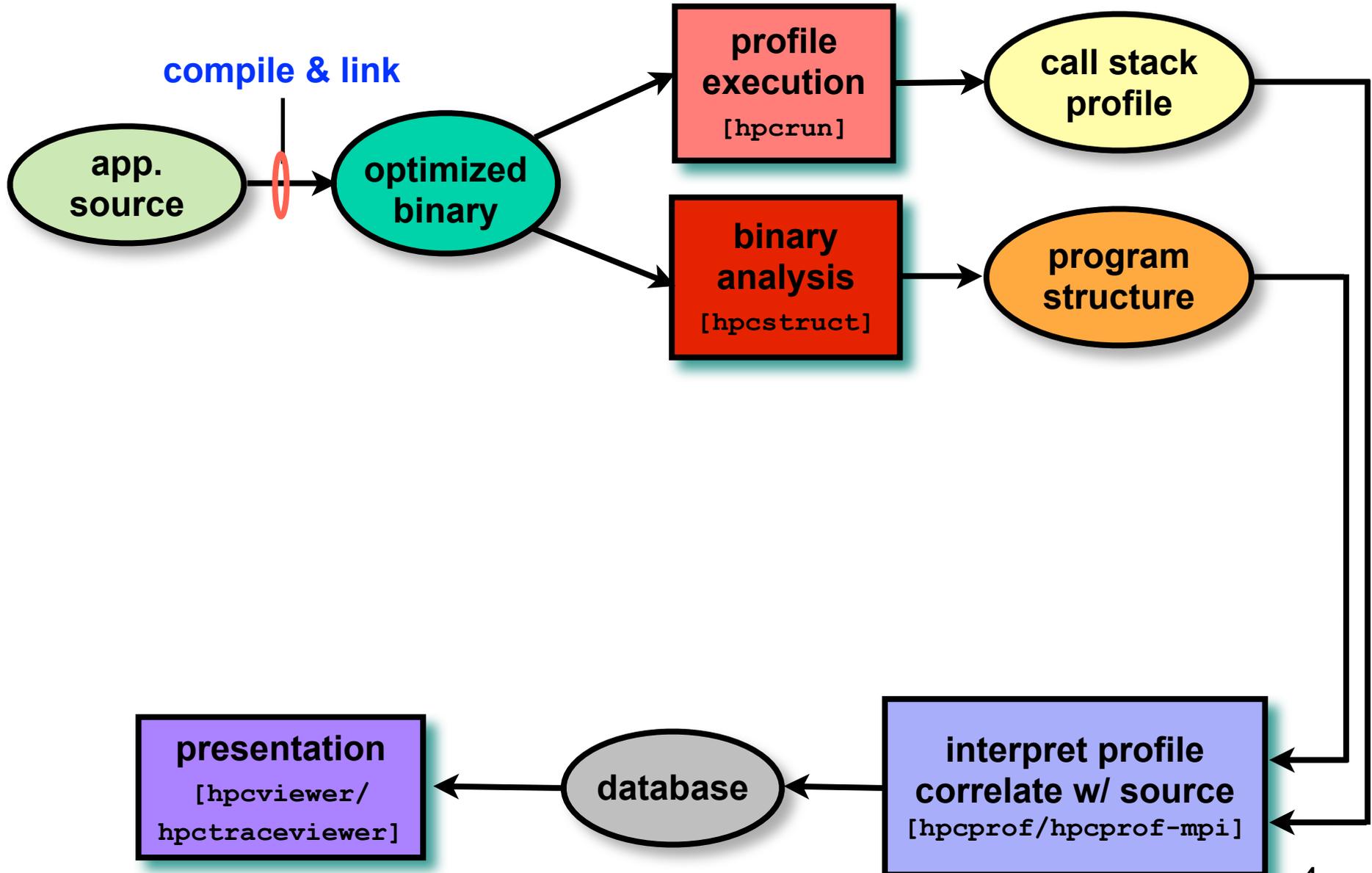
Acknowledgments

- **Funding sources**
 - **DOE Office of Science SciDAC-2 (no-cost extension for 2012)**
 - **Center for Scalable Application Development Software**
Cooperative agreement number DE-FC02-07ER25800
 - **Performance Engineering Research Institute**
Cooperative agreement number DE-FC02-06ER25762
 - **Corporate: AMD, Western Geco**
- **Project team**
 - **Research Staff**
 - **Laksono Adhianto, Mike Fagan, Mark Krentel**
 - **Students**
 - **Xu Liu, Milind Chabbi**
 - **Collaborator**
 - **Nathan Tallent (PNNL)**
 - **Alumni**
 - **Gabriel Marin (ORNL), Robert Fowler (RENCI), Nathan Froyd (Mozilla)**
 - **Summer Interns**
 - **Michael Franco (Rice), Reed Landrum (Stanford), Sinchan Banerjee (MIT)**

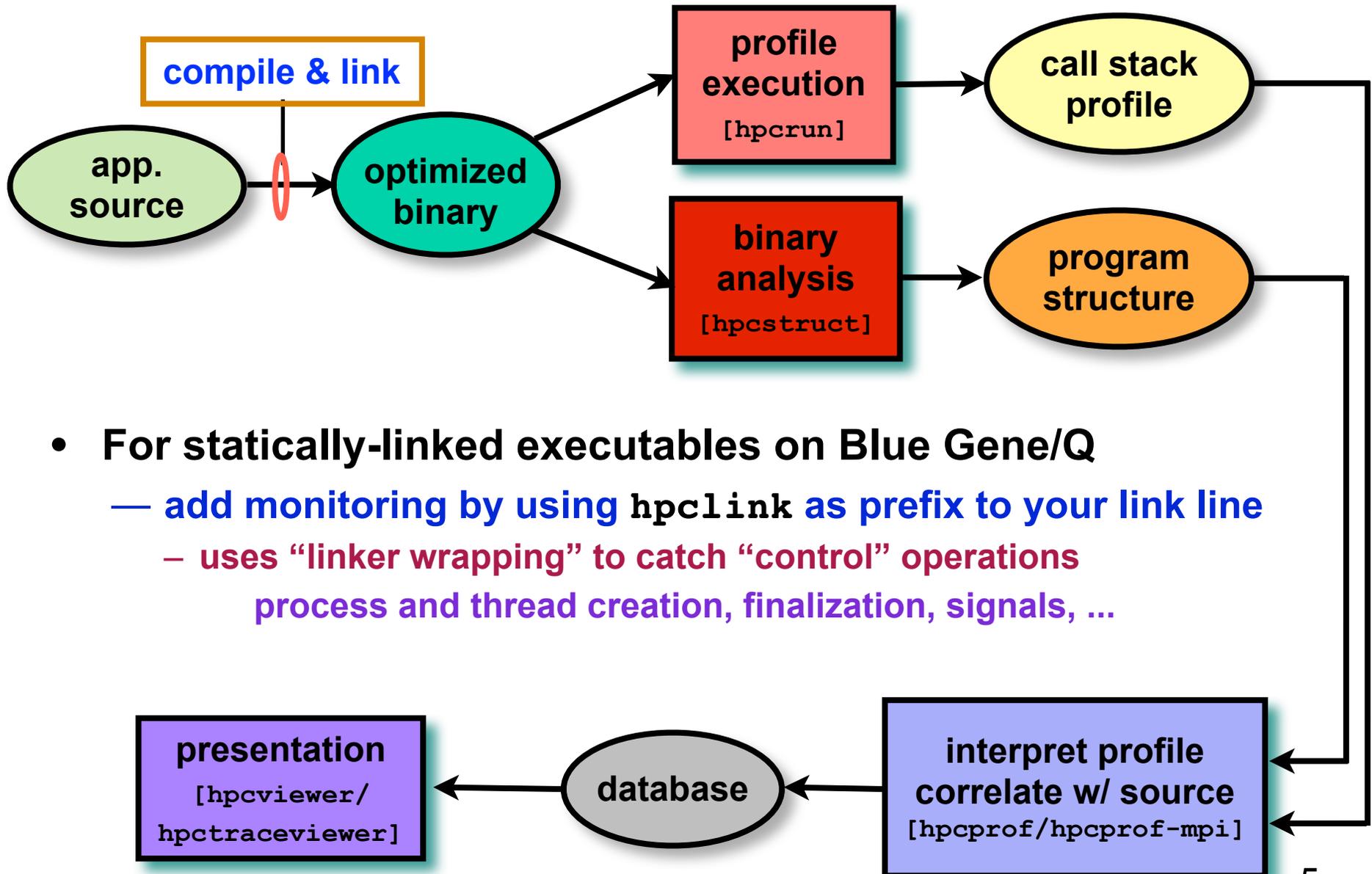
Rice University's HPCToolkit

- **Employs binary-level measurement and analysis**
 - observe **fully optimized**, **dynamically linked executions**
 - support **multi-lingual codes** with external binary-only libraries
- **Uses sampling-based measurement (avoid instrumentation)**
 - **controllable overhead**
 - **minimize** systematic error and avoid blind spots
 - enable data collection for **large-scale parallelism**
- **Collects and correlates multiple derived performance metrics**
 - diagnosis typically requires more than one species of metric
- **Associates metrics with both static and dynamic context**
 - **loop nests**, **procedures**, **inlined code**, **calling context**
- **Supports top-down performance analysis**
 - identify costs of interest and drill down to causes
 - up and down call chains
 - over time

HPCToolkit Workflow

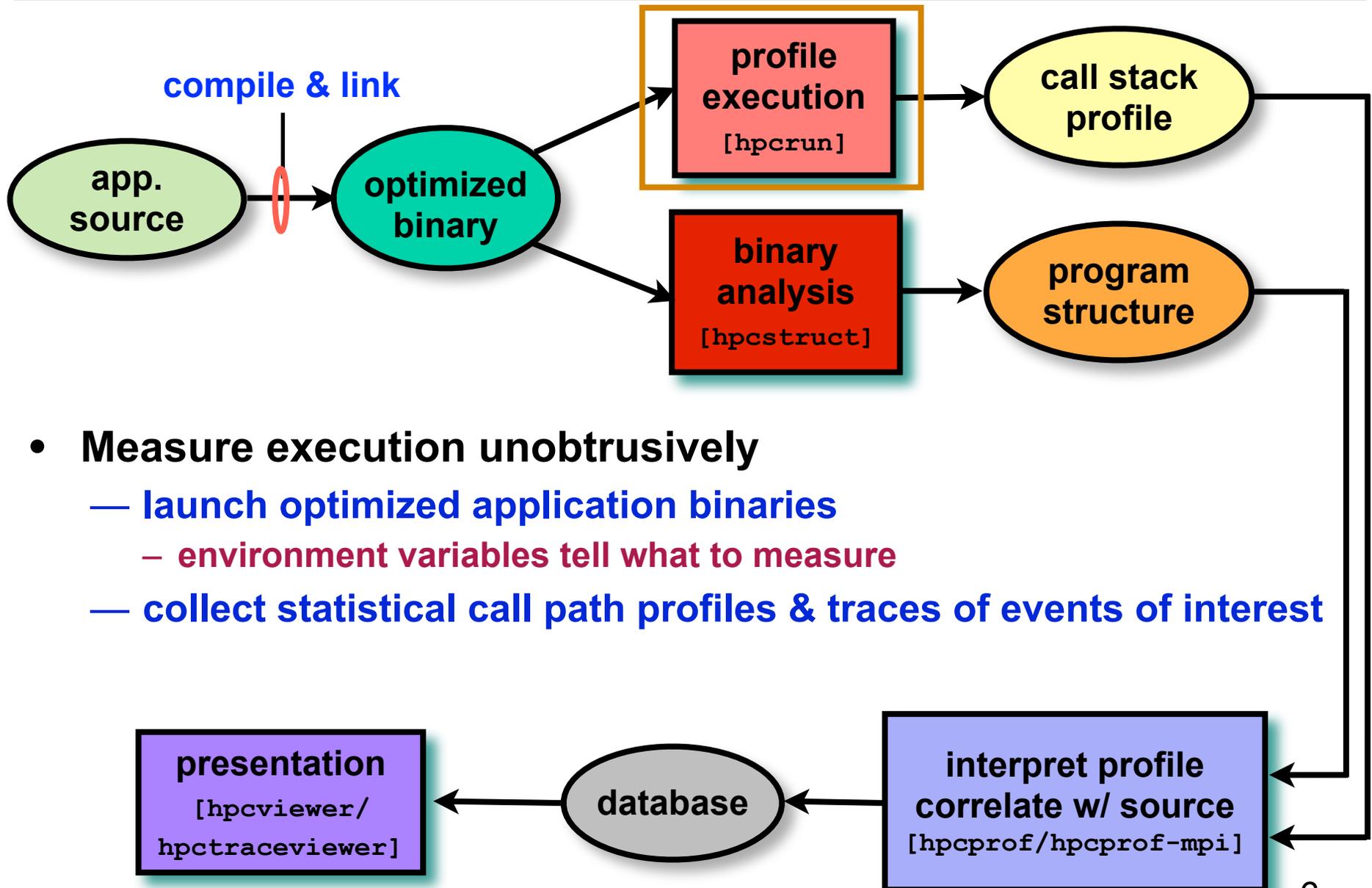


HPCToolkit Workflow



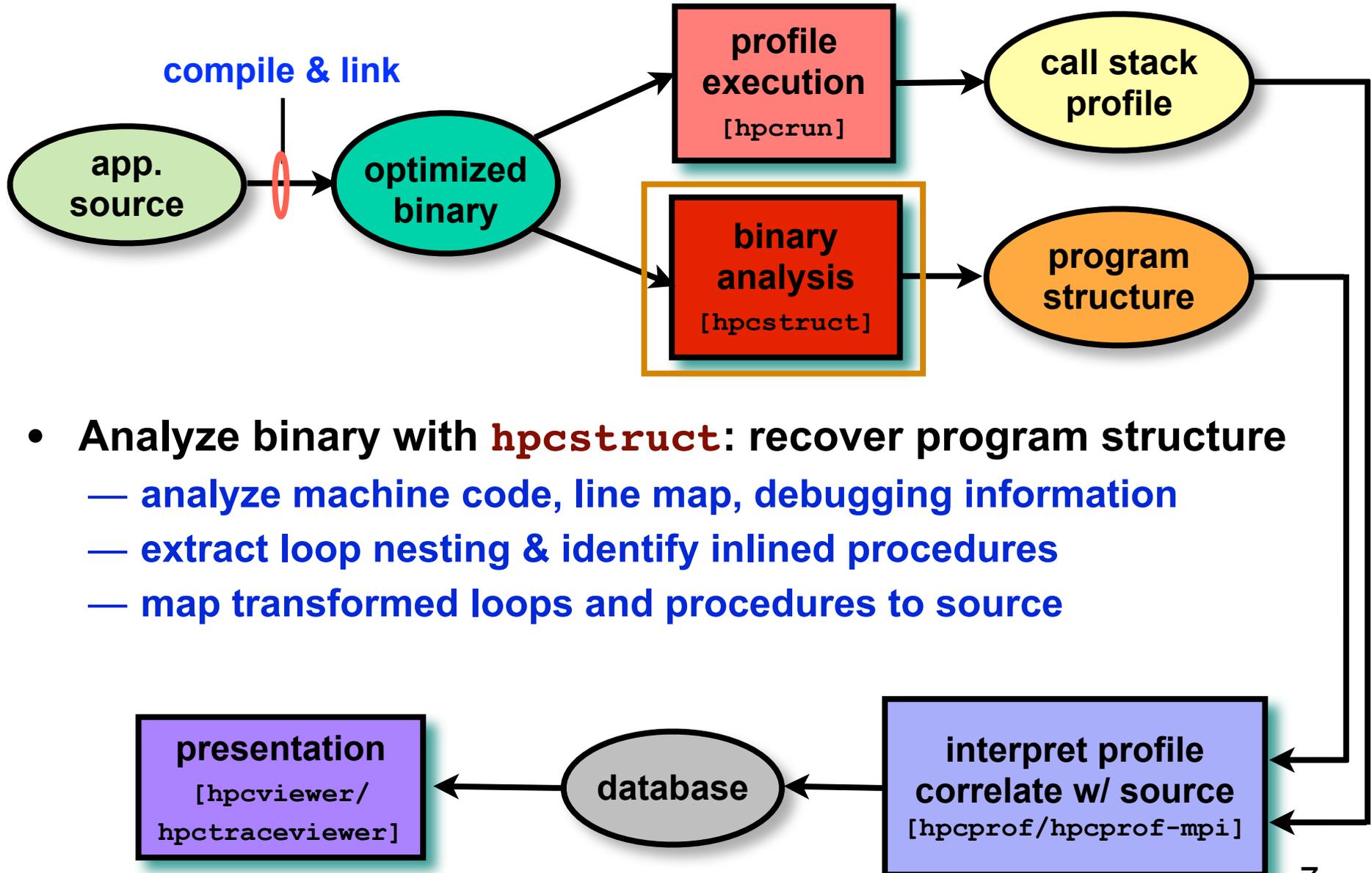
- For statically-linked executables on Blue Gene/Q
 - add monitoring by using `hpcLink` as prefix to your link line
 - uses “linker wrapping” to catch “control” operations
process and thread creation, finalization, signals, ...

HPCToolkit Workflow



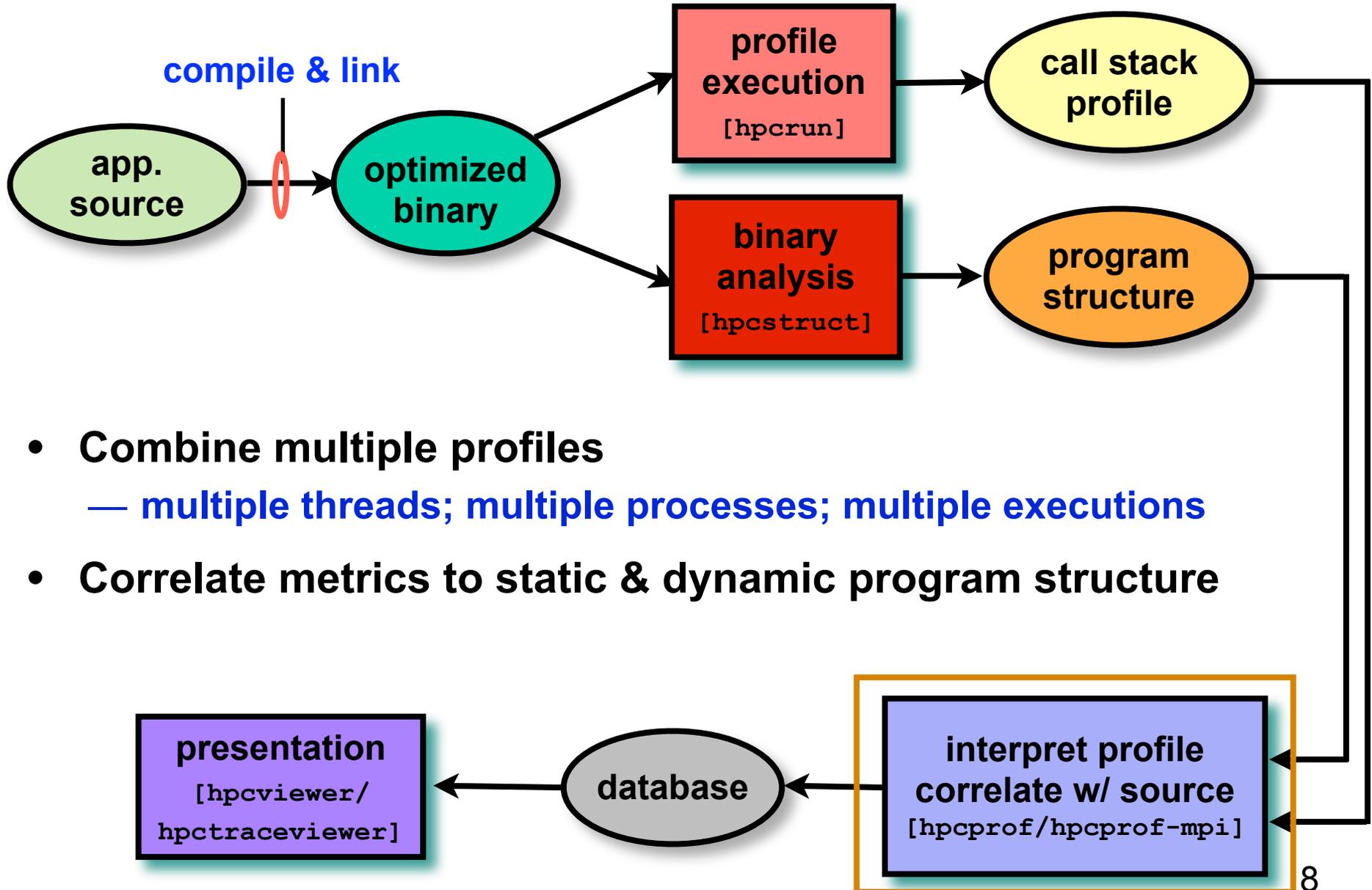
- **Measure execution unobtrusively**
 - launch optimized application binaries
 - environment variables tell what to measure
 - collect statistical call path profiles & traces of events of interest

HPCToolkit Workflow



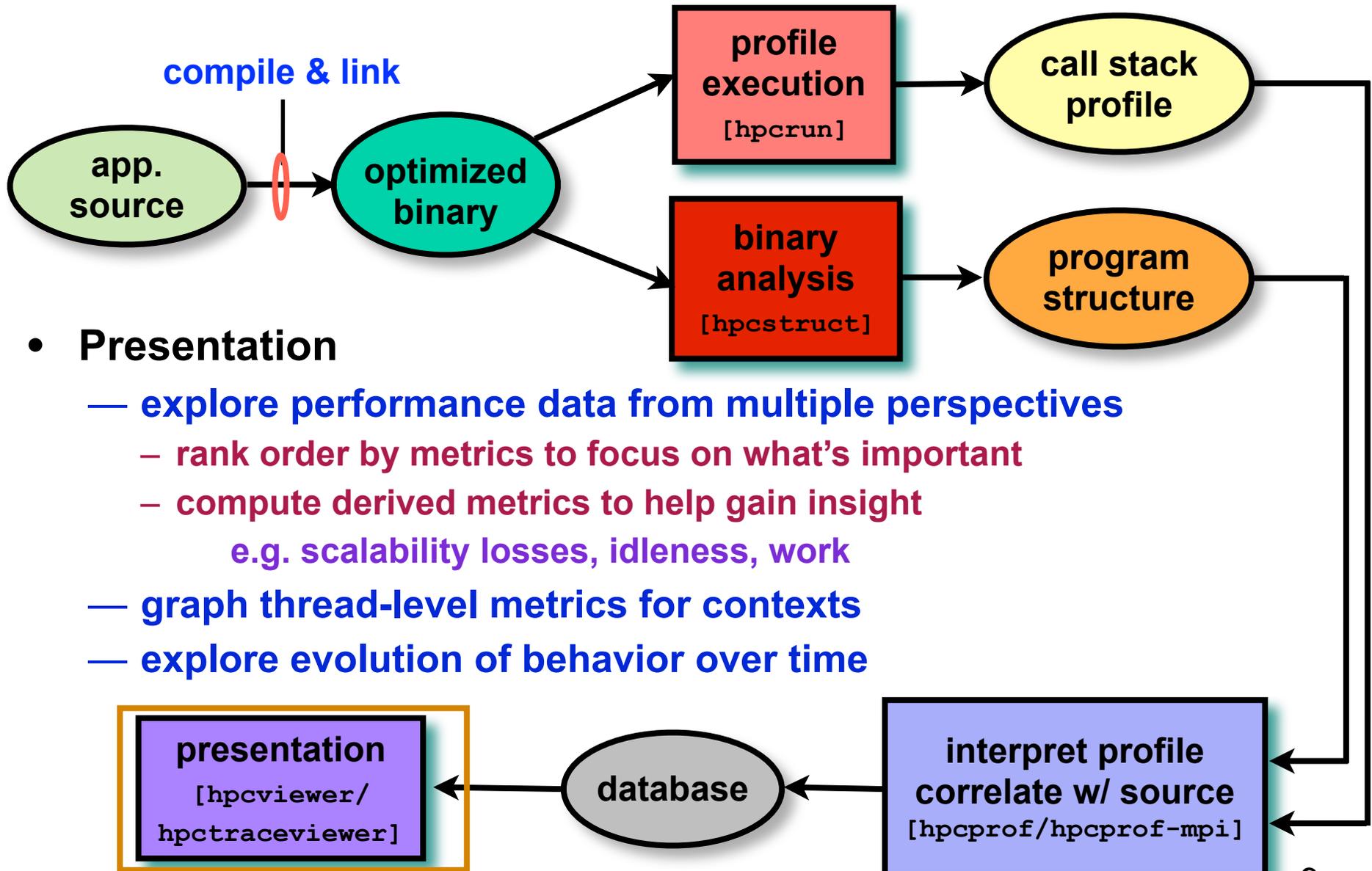
- Analyze binary with **hpcstruct**: recover program structure
 - analyze machine code, line map, debugging information
 - extract loop nesting & identify inlined procedures
 - map transformed loops and procedures to source

HPCToolkit Workflow



- **Combine multiple profiles**
 - multiple threads; multiple processes; multiple executions
- **Correlate metrics to static & dynamic program structure**

HPCToolkit Workflow



- **Presentation**

- explore performance data from multiple perspectives
 - rank order by metrics to focus on what's important
 - compute derived metrics to help gain insight
 - e.g. scalability losses, idleness, work
- graph thread-level metrics for contexts
- explore evolution of behavior over time

Rice's Work to Date on Blue Gene/Q

- Developed specification for kernel functionality needed to support sample-based profiling on BG/Q
 - became part of Mira procurement
- Worked with IBM Rochester and UTK to bring up BGPM (Blue Gene Performance Monitor) and PAPI on BG/Q
 - provided feedback on evolving design
 - designed tests, tested counter overflow support needed for sampling
 - identified race conditions associated with threading
- Ported Rice University's HPCToolkit to BG/Q
- Invented and prototyped new ideas for performance analysis of MPI+OpenMP codes
 - demo today!
- Coordinating with IBM to get them to add prototype support for our profiling strategy in their production XL OpenMP runtime
- Working with OpenMP ARB tools group to develop a standard tools interface for OpenMP



What's New for BG/Q?

- **Problem: MPI everywhere is no longer enough**
 - **must use threading to fully exploit the power of BG/Q's A2**
 - **16 compute cores; 4 hardware thread contexts per core**
- **Impact: developers need guidance while adding threading**
- **HPCToolkit uses a novel technique to provide unique insight**
 - **low-overhead measurement using sampling**
 - **within and across nodes of MPI+threads**
 - **new measurement methodology for root cause analysis of performance losses in MPI+OpenMP programs**
 - **assess idleness vs. work: for loops, procedures, call chains, program**
 - **code-centric views precisely attribute time, work, idleness**
 - **space-time diagrams illustrate performance over time**



Understanding Low Performance with Threads

- Inside OpenMP runtime systems
 - worker threads await you to
 - enter parallel regions
 - dispatch work to threads
 - SPMD work within a parallel region
 - OpenMP work sharing: parallel loops, parallel sections
 - OpenMP tasking
- Knowing that worker threads are idling or blocked is a symptom of a problem
- What about the causes? You need to know:
 - where in your code idling is a problem?
 - how badly is idling contributing to performance losses in different parts of your program?
 - where are your biggest gains to be had?
 - how much can changes improve overall program performance
 - within a program region
 - across the program as a whole

Root Cause Analysis of Losses with OpenMP

HPCToolkit's measurement approach

- **Add lightweight instrumentation of OpenMP runtime system**
 - keeps track of when a thread is working or idle
 - precisely quantifies instantaneous idleness and work
 - e.g. 6 out of 16 cores are working productively at this instant
- **Precisely attribute blame for idleness to its causes**
 - **examples of what this means**
 - when the master thread executes outside a parallel region leaving its OpenMP threads sit idle
 - associate idleness of workers with code executed by the master
 - when an OpenMP worksharing loop unevenly distributes work
 - associate idleness of workers with the code for the loop
 - **call stack unwinding associates costs with code in its full calling context**

Where to Find HPCToolkit

- **Path to the tools on VEAS**
 - </home/projects/hpc/pkg/hpctoolkit/bin>
- **User manual**
 - <http://hpctoolkit.org/manual/HPCToolkit-users-manual.pdf>
 - </home/projects/hpc/pkg/hpctoolkit/share/doc/hpctoolkit/manual/HPCToolkit-users-manual.pdf>
- **Man pages**
 - </home/projects/hpc/pkg/hpctoolkit/share/man>
- **All the above and more documentation on our website**
 - <http://hpctoolkit.org>