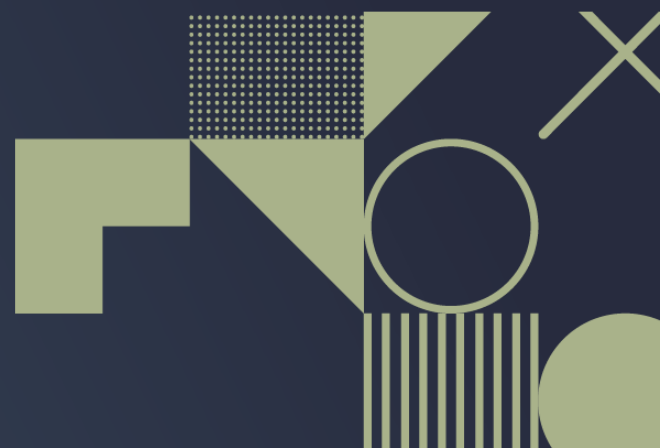# Introduction to HPCToolkit
Measurement and Analysis of Unmodified, Optimized Applications

John Mellor-Crummey

Department of Computer Science
Rice University

15 November 2021

Rice University's HPCToolkit Performance Tools

Measure and analyze performance of CPU and GPU-accelerated applications

- **Easy: profile unmodified application binaries**
- **Fast: low-overhead measurement**
- **Informative: understand where an application spends its time and why**
  - call path profiles associate metrics with application source code contexts
  - optional traces to understand execution dynamics
- **Broad audience**
  - application developers
  - framework developers
  - runtime and tool developers

Not affiliated with:
　　IBM HPC Toolkit
　　Intel oneAPI HPC Toolkit

HPCToolkit Capabilities

- **Employs binary-level measurement and analysis**
  - Observes executions of fully optimized, statically-linked or dynamically-linked applications
  - Supports multi-lingual codes with external binary-only libraries
- **Measures CPU activity using sampling**
  - Controllable overhead
  - Minimizes systematic error and avoid blind spots
  - Enables data collection for large-scale parallelism
- **Measures GPU performance**
  - Function wrapping or callbacks monitor launch of GPU operations
  - Monitoring thread records information about asynchronous operations on GPU devices
- **Associates metrics with both static and dynamic context**
  - Loop nests, procedures, inlined code, calling context on both CPU and GPU
- **Computes derived CPU and GPU performance metrics of your choosing**
  - Diagnosis may require more than one species of metric
- **Supports top-down performance analysis**
  - Identify costs of interest and drill down to causes: up and down call chains, over time
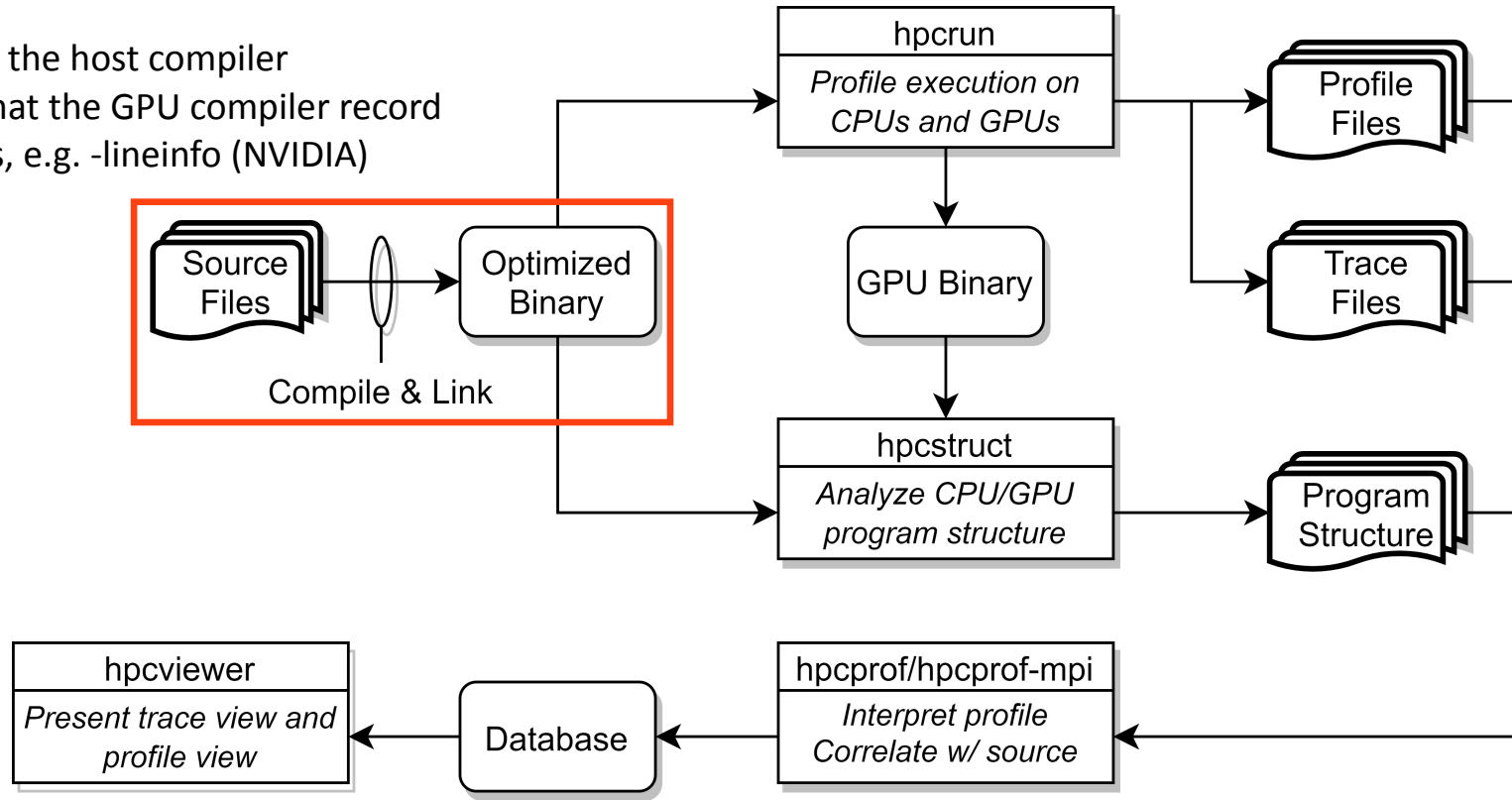
HPCToolkit's Workflow for CPU Applications
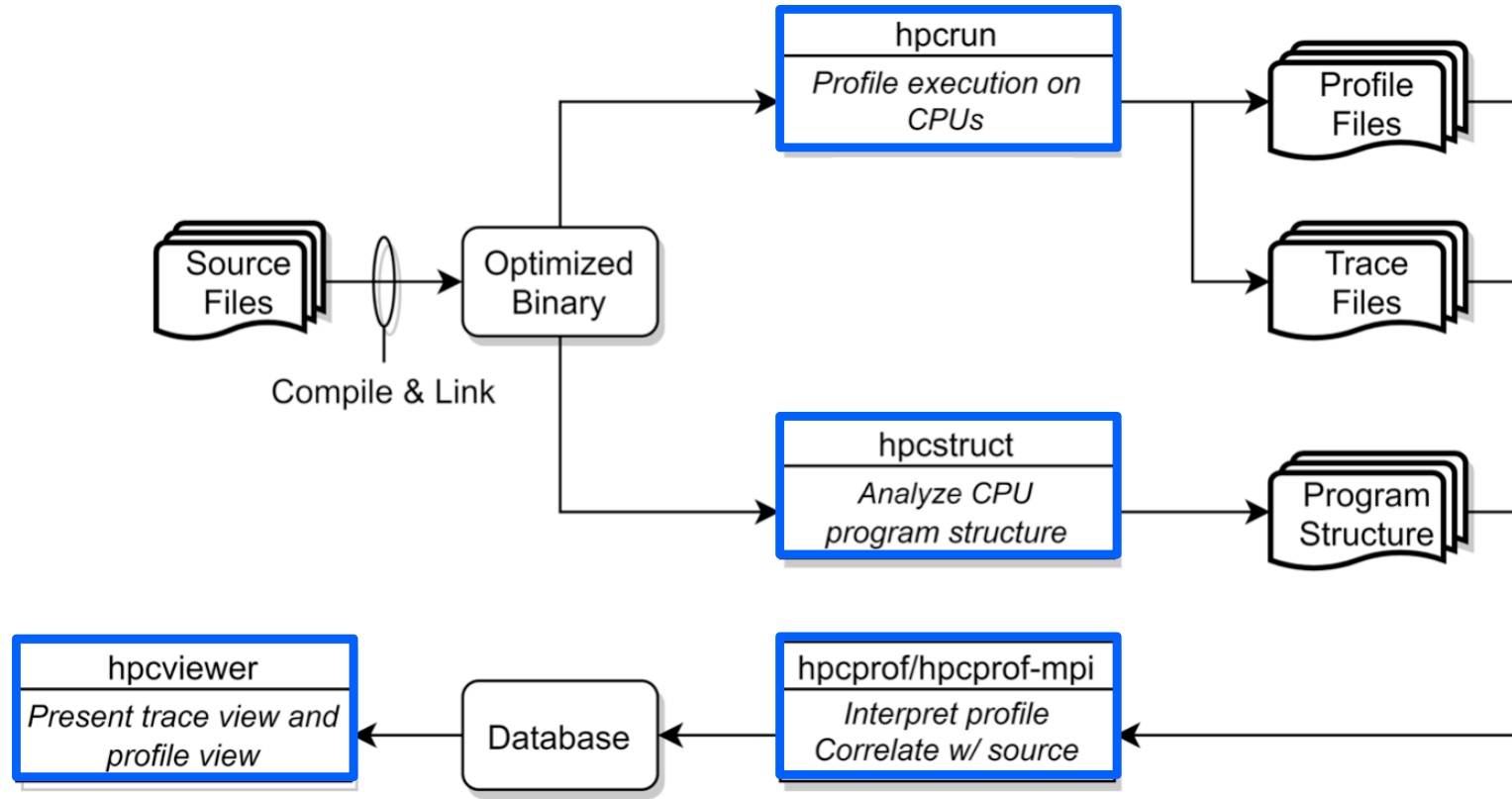
HPCToolkit's Workflow for GPU-accelerated Applications
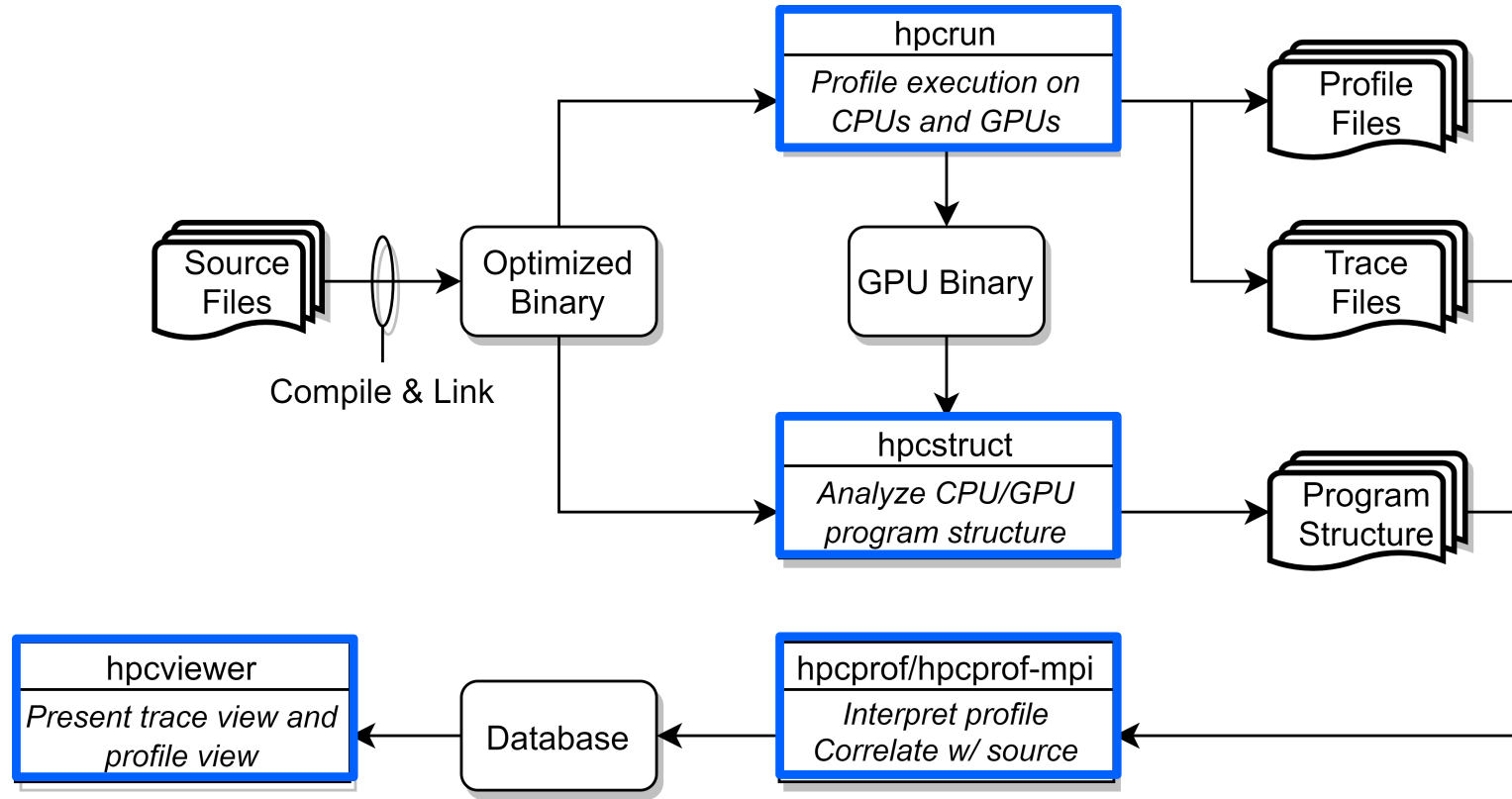
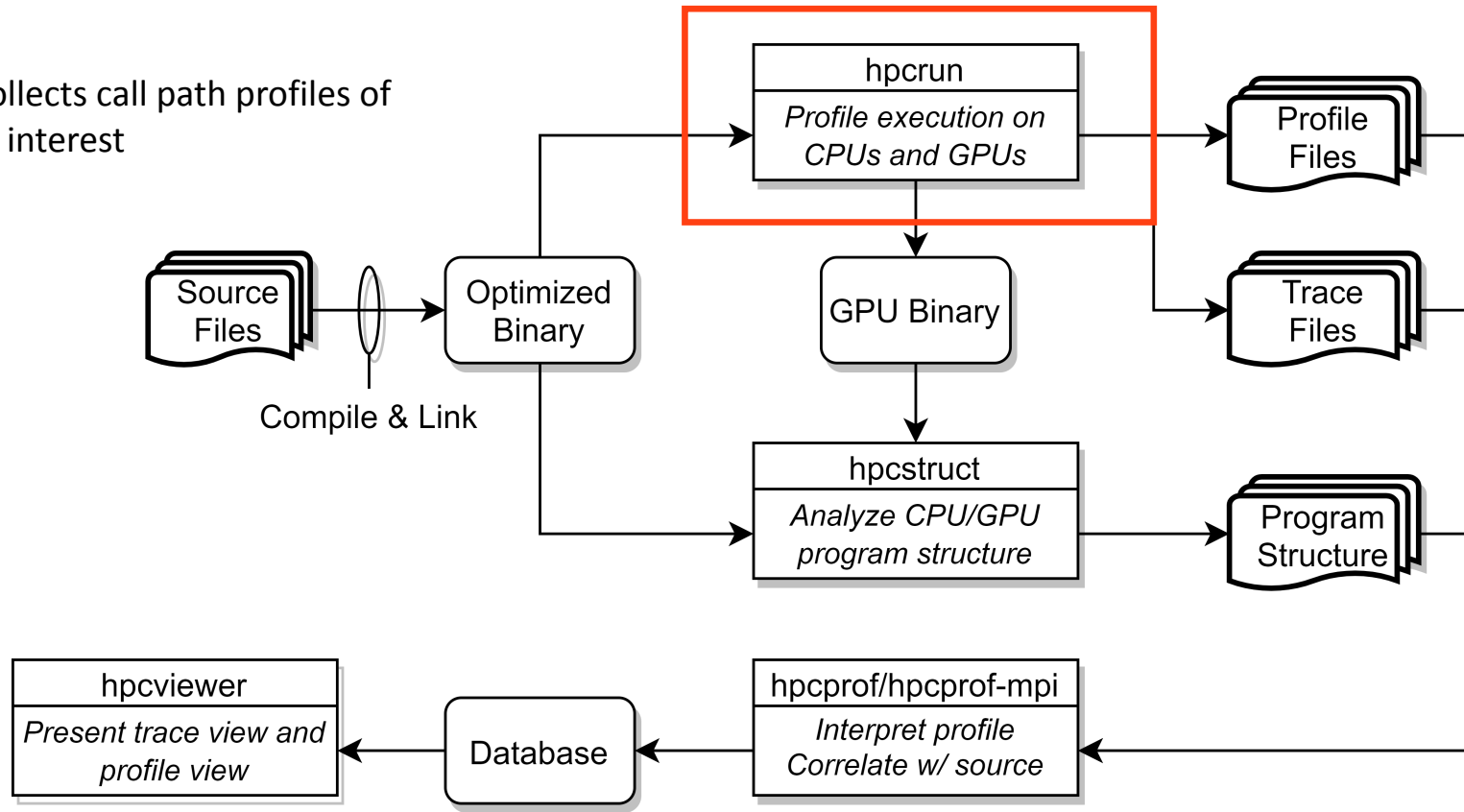# HPCToolkit's Workflow for GPU-accelerated Applications

Step 1:
- Add -*g* to the host compiler
- Specify that the GPU compiler record line maps, e.g. -lineinfo (NVIDIA)

Source Files → Compile & Link → Optimized Binary

hpcrun — *Profile execution on CPUs and GPUs* → Profile Files

GPU Binary

Trace Files

hpcstruct — *Analyze CPU/GPU program structure* → Program Structure

hpcviewer — *Present trace view and profile view* ← Database ← hpcprof/hpcprof-mpi — *Interpret profile Correlate w/ source*

HPCToolkit's Workflow for GPU-accelerated Applications

Step 2:
- *hpcrun* collects call path profiles of events of interest

HPCToolkit's Workflow for GPU-accelerated Applications

Step 3:
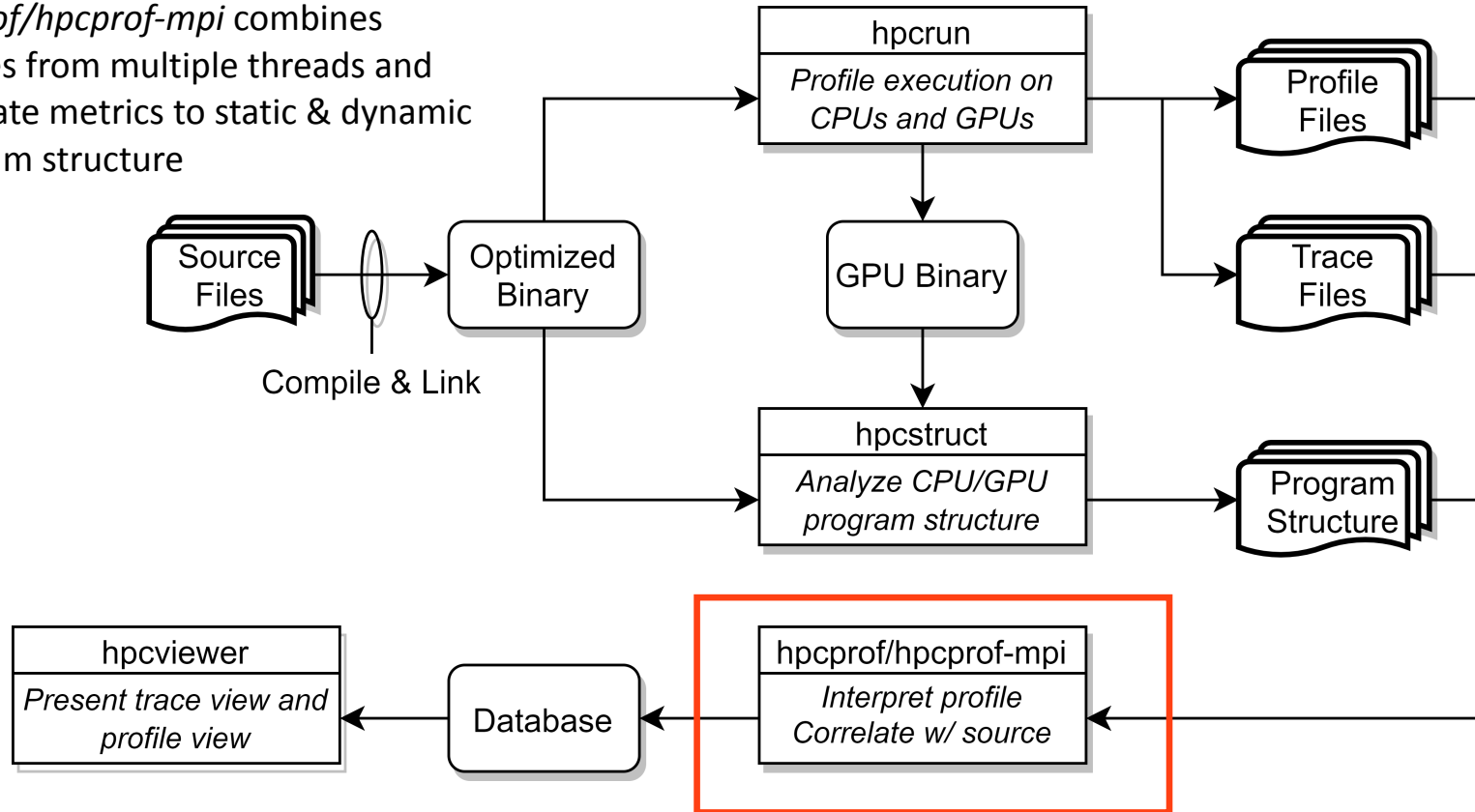- *hpcstruct* recovers program structure about lines, loops, and inlined functions
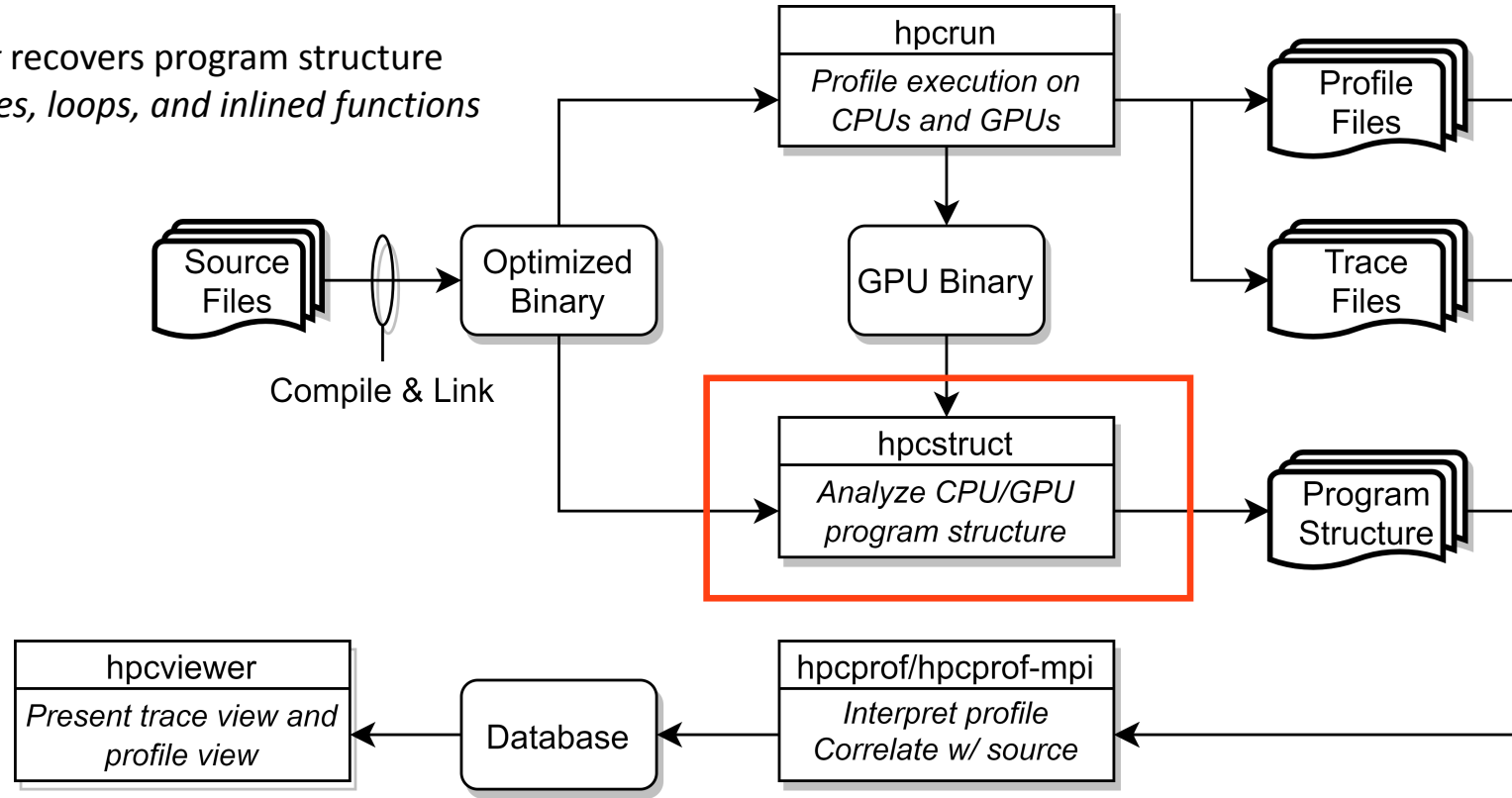
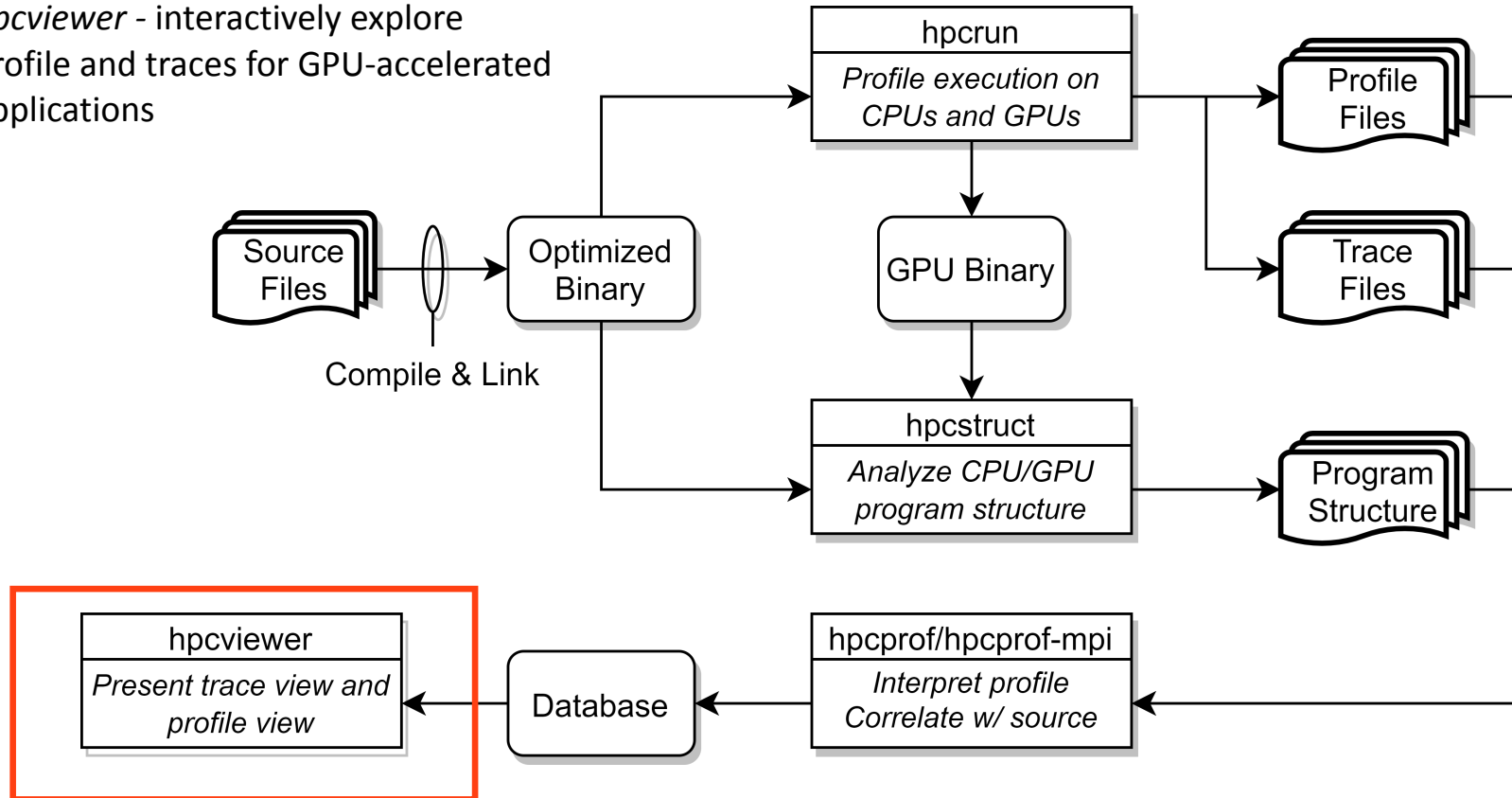# HPCToolkit's Workflow for GPU-accelerated Applications

Step 4:

- *hpcprof/hpcprof-mpi* combines profiles from multiple threads and correlate metrics to static & dynamic program structure

HPCToolkit's Workflow for GPU-accelerated Applications

Step 4:
- *hpcviewer* - interactively explore profile and traces for GPU-accelerated applications

Code-centric Analysis with hpcviewer

Time-centric Analysis with hpcviewer

Links and Contact Information

- **Project web site: http://hpctoolkit.org**
- **Documentation: http://hpctoolkit.org/documentation.html**
- **Installation: http://hpctoolkit.org/software.html**
- **Long format tutorials: http://hpctoolkit.org/training.html**

- **Questions, problems, bug reports: hpctoolkit-forum@rice.edu**