Retargeting TAU for HPC++

Kurt Windisch, Al Malony, Sameer Shende,
Kathleen Lindlan, Aida Couchane

Prepared for the 1997 HPC++ Meeting

Boulder, Colorado

May 9 - 10, 1997

Retargeting TAU

- Challenges faced retargeting for HPC++:
  - robust support for C++ (e.g. templates)
  - support for large, complex projects
    - multiple files
    - multiple libraries
  - tools well-matched to the analysis problems in HPC++
  - HPC++ is an evolving system

Retargeting TAU

- Our approach:
  - **project manager**: provides coordination of project-level information to all tools
  - **browser database**: provides high-level information about functions/classes and resolves cross-file references
  - **build system**: retargetable and configurable
  - **language support components**: extensions and increased modularity

Project Manager

- Motivation:
  - Under **pC++**, projects consisted of only one file.
  - Under **HPC++**, projects consist of multiple files
  - ➠ Tools must operate on much more complex applications.

- The **Project Manager** serves the tools by managing project-level information:
  - records the set of application files
  - records the language of each file
  - facilitates addition/removal of files
  - records project-wide and file-specific compilation information
  - coordinates project updates among all tools
  - coordinates global selections among participating tools
  - records host, architecture, and filesystem information
  - serves this all above information to tools as requested

- The project manager resides in the Tau daemon

Browser Database

- Motivation - large projects:
  - Tools require quick access to high-level program information w/o overhead of the CGM
  - functions and class identifies are unique only at the file-level
  - ➠ fast resolution of symbols among multiple program files is needed for browsing

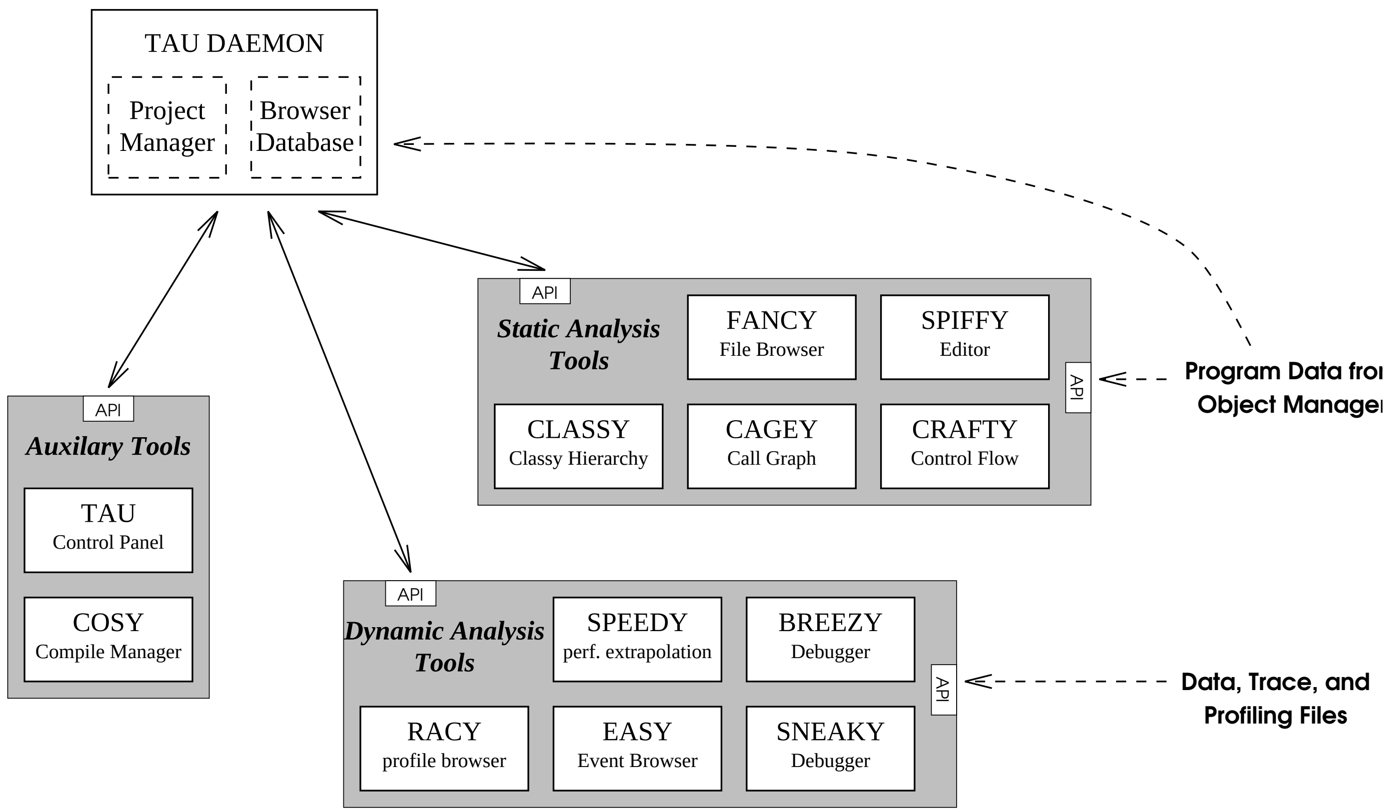
- The **Browser Database** manages function and class symbols throughout the project:
  - extracts minimal program symbol information when notified of file updates (by project manager)
  - resolve global symbol references (normally resolved at link-time).
  - tools can query for information about functions or classes (e.g., the definition location of a function or class)

- The browser database resides in the Tau daemon

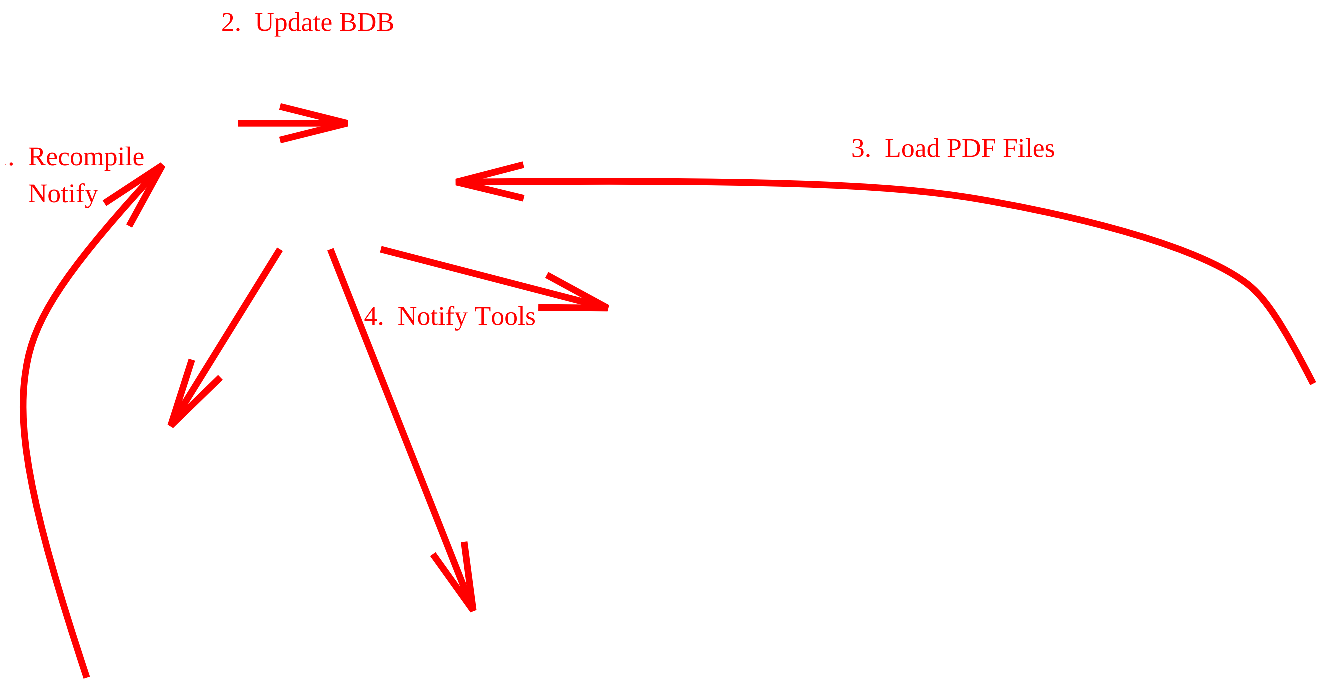
Build System

- Makefile-based
- Extensible to new
  - target architectures
  - target compilation modes
  - languages

Overview of Tool Architecture



- Tool Architecture

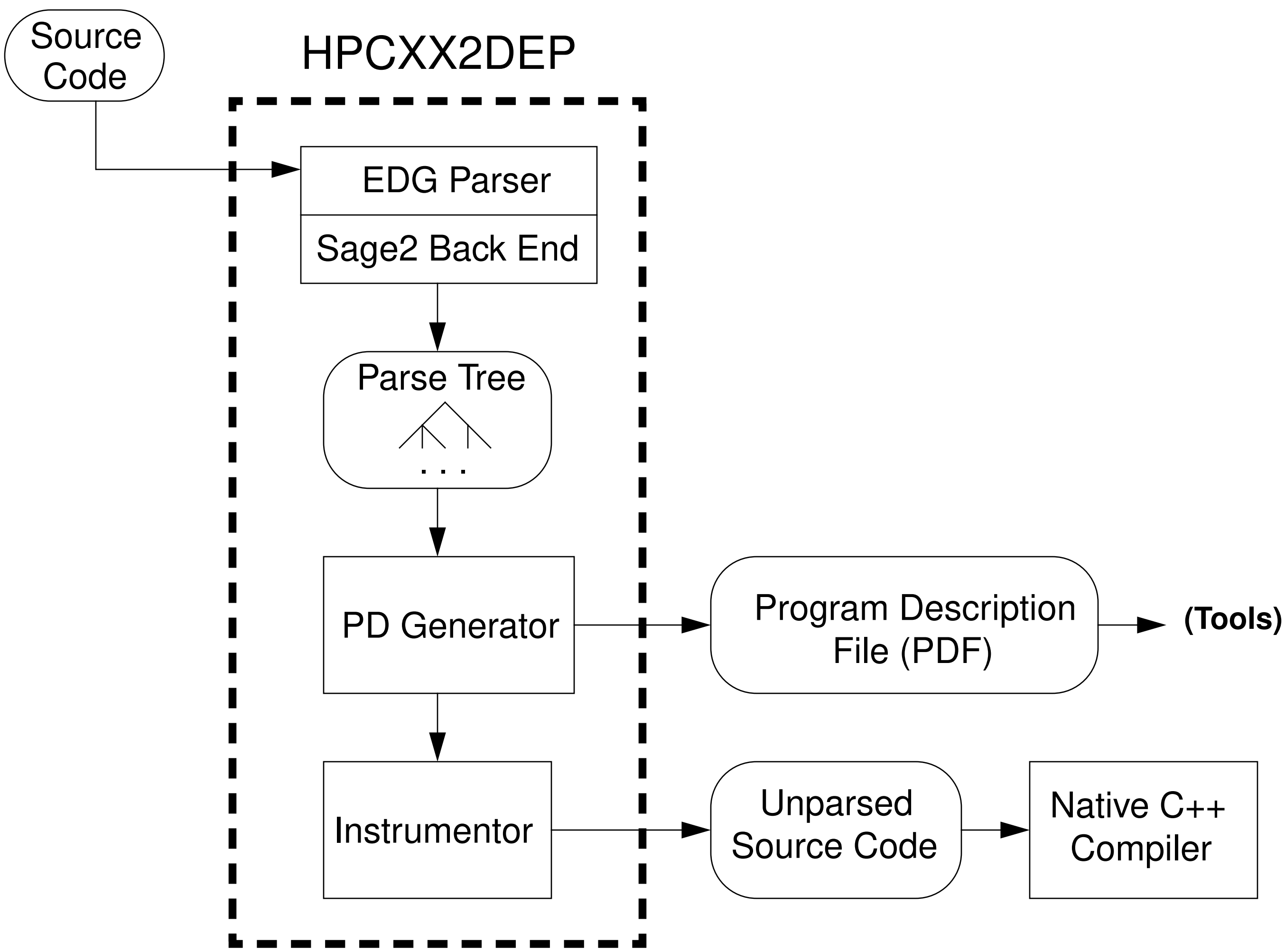


- Recompile Steps
  1) COSY notifies the Project Manager
  2) Project Manager notifies Browser Database
  3) Browser Database updates from new program description files
  4) Project Manager notifies all tools

HPC++ Language Support

- ❑ Sage2-based CGM extracts static program information
- ❑ Sage2-based instrumentor adds profiling objects to the application before compilation.
- ❑ What is Sage2?
  - ❍ Sage2 is C++ compiler toolkit
  - ❍ developed by Carl Kesselman (USC-ISI) and Dennis Gannon (University of Indiana)
  - ❍ successor of Sage1.x
  - ❍ built above the Edison Design Group C++ Front End, a robust, commercial C++ front end parser
  - ❍ Sage2 provides a back-end that builds powerful, object-oriented abstract syntax trees
  - ❍ Sage2 facilitates investigation and transformation of the program's structure.

Sage2 in TAU



Source Code
HPCXX2DEP
EDG Parser
Sage2 Back End
Parse Tree
. . .
PD Generator
Program Description File (PDF)
(Tools)
Instrumentor
Unparsed Source Code
Native C++ Compiler

Sage2 Issues

- Sage2 is still under development

- Currently, Sage2 lacks features critical to full support of HPC++:
  - template support is not robust, undeveloped, or buggy. Issues:
    - instantiation with locally scoped types,
    - virtual template functions, and
    - sage node types for template functions/classes
  - file location information is not detailed:
  - Sage2 will not parse or unparse full HPC++ programs

- Robust template support is critical to STL/PSTL support in TAU

- Workarounds in TAU
  - “sanitized” libraries for parsing
  - implementation of some needed functionality
  - hand instrumentation

Outstanding Problems

- Sage2
- Adapting to evolutionary changes in HPC++

Future Development

- CGM extensions
    - templates
    - control flow
- Optimizations
- Interoperability - new communication options
- HPC++-specific tools
    - static
    - dynamic
- integration with HPC++ compilers
- extension to other languages
- ???

Toolkit Status

	pC++	pgHPF	Fortran 77 Fortran – M	ANSI C C++	HPC++
cosy	X	C	C	X	X
mity	X	X	X	X	X
fancy	X	X	X	X	X
cagey	X	X	X	X	X
classy	X			X	X
racy	X	X		X	X
speedy	X				
crafty	P	P	P	P	P
spiffy	X	X	X	X	X
sneezy	X			X	X
sneaky	X			?	?
easy	P				
breezy dandy DAQV	X P P	 X X			 ? ?

- HPC++Lib: All HPC++ static tools and RACY profiler
- PSTL: Only RACY profiler

  ...However, all HPC++ static tools will work when Sage2 problems are solved