

DynInst/DPCL User Experiences

Ted Hoover

Team Lead - Application Development Tools

SP Development - IBM Poughkeepsie

hoov@us.ibm.com

User Experiences

- DynInst
 - ▶ Library User
 - ▶ Library Developer
- DPCL

DynInst User Experiences

- Licensed DynInst to Use as Base for DPCL
- Ported to IBM xLC Compiler
- Integrated into DPCL Daemon
- Initial Experiences
 - ▶ Signal Interference
 - ▶ Error Handling
 - ▶ "Not Yet Implemented"

DynInst Users Experiences (cont.)

- Needed More Function to Support DPCL and End User Tools
- Needed Product Level Robustness
- Needed More than We Could Reasonably Ask For...

Solution

- Initiated Design to Expand the DynInst Interface to Maintain Portability
- Initiated IBM DynInst Development Effort

DynInst Developer Experiences

- Added Additional Classes/Member Functions
- Switched to STL Library
- Diagnostic Logging
- Multithreaded
- Status
 - ▶ Open Issue with Error Codes
 - ▶ Better Definition of Semantics
 - ▶ Need Forum to Propose Changes

DPCL User Experiences

- Changes Were Made Based on User Feedback
 - ▶ Moved Away From Probe Language to C for ProbeModules
 - ▶ Phases Improved to Allow Instrumentation at Application Exit
 - ▶ Minor Member Function Name Changes
 - ▶ Type Checking Too Strong