



parallel tools platform

parallel debugging

challenges & opportunities

Oak Ridge National Laboratory

23-24 May 2007

## State of the Art

- Traditional debugging methodology
  - Follows execution flow
  - Breakpoint, inspect, step, repeat (B.I.S.R)
- May not work with advanced languages
- Will not work with petascale architectures
  - With  $10^6$  threads of execution?
- Assumption is that gdb and TotalView will solve the problem
- When all else fails: printf

# Debugging Challenges

- Identifying error occurrence
  - Predicates/assertions
  - Message/data patterns
- Locating error data
  - Visualize global program state
  - Comparison/search across application
- Rewinding/replaying
  - Holy grail?

## Debugging Challenges (cont...)

- Smarter debugger
  - Can the debugger off-load some of the burden from the developer?
- Leverage other tools to assist with debugging
  - e.g. Performance tools to identify hot spots/regions of interest
- Timing issues?
- ... and still be as easy (or easier) than printf

# Heterogeneous Architectures

- What about heterogeneity issues?
- Needs to support different processor architectures
- Multiple levels of hierarchy
- Different programming models at each level

## How to get out of the rut?

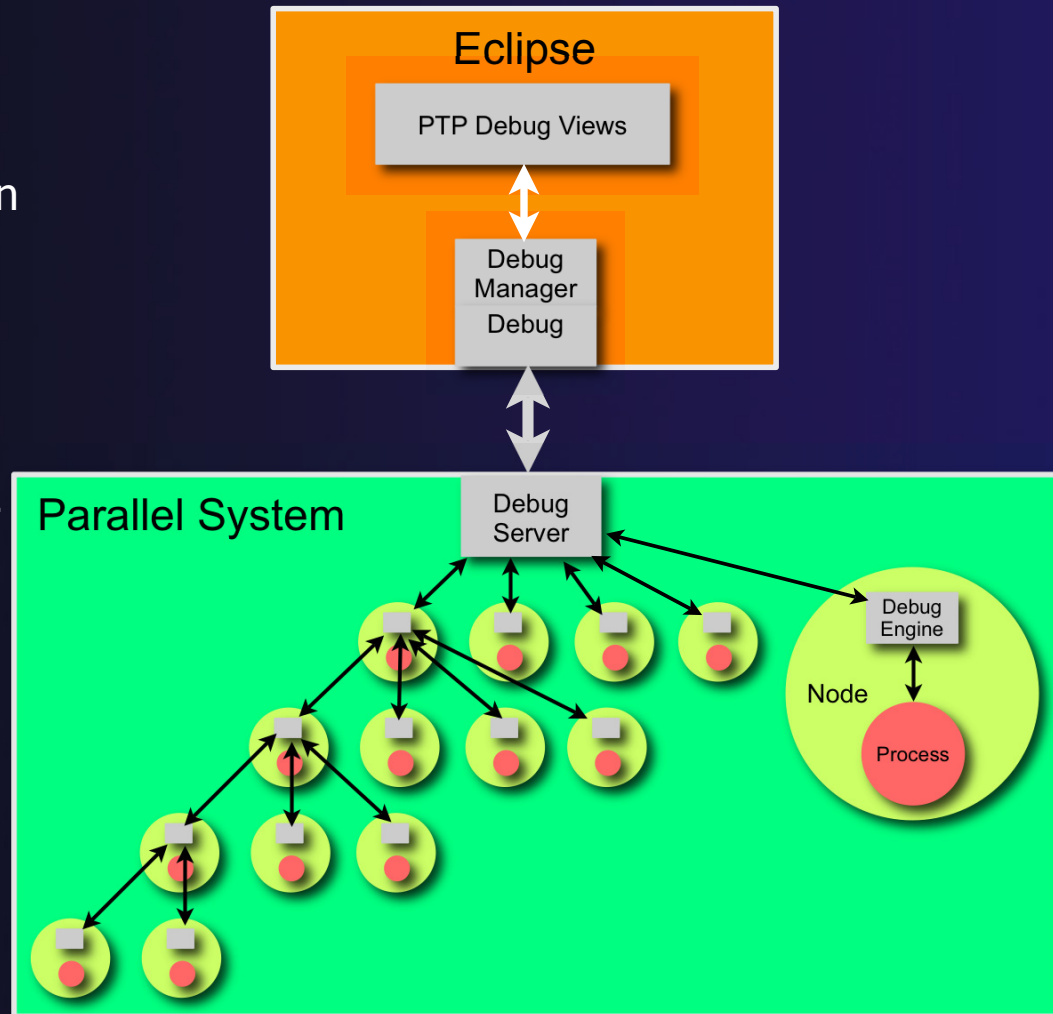
- Universal parallel debugging platform for research development
- Rich user interface
- Community willing to undertake research/development
  - Must be open source
  - Too hard to develop from scratch
- Some level of confidence
  - Technology can be advanced
  - Will aid developers, not hinder
  - Longevity of platform, solution

## PTP Debug Platform: Now

- Traditional debug views
- Groups of processes
  - debug commands (e.g. step, resume, etc.)
  - breakpoints
  - execution location
- Client/server debug architecture
  - Debug application launch
  - Command broadcast
  - Event Aggregation

# PTP Debug Architecture

- Debug server is an MPI program
- Debug engines are started on each node, one per process
- Debug engines act as message forwarders/aggregators
- High level debug API allows replacement of debug server
- GDB currently used for low-level debug operations





## PTP Debug Platform: Future

- **Advanced debug views**
  - Visualize data from multiple processes
  - Support for multiprocess/multithreaded applications
  - Better scalability in views
- **Generic server architecture**
  - Launch other tools, not just debuggers
  - Generalized command broadcast/event aggregation framework
  - Arbitrary message format for tool communication
- **Support for proprietary backend debug engines**