Why MPI Makes You *Scream!* And How Can We Simplify Parallel Debugging?

Jayant DeSouza, Intel Corporation
Jeff Squyres, Indiana University
Goals of This BOF

- List what *we* think are the problems
  - And some possible solutions
- Hear what *you* think are the problems
  - Why are they problems for you?
  - How do you solve them now?
  - ...?
- Next steps
Jayant DeSouza

- Senior Software Engineer, Intel Corporation
  - Advanced Computing Center, Tools for petaflop architectures
- MPI tool implementer
  - Intel Message Checker
Classification of Errors in MPI

And yet, everyone uses MPI.
User Survey
State of the Tools Address

☐ Compile time lint tool for MPI?
   ■ MPI-Check?

☐ printf/write is a difficult debugging model
   ■ Requires many iterations to narrow down the error
   ■ But:
     ○ available on every system
     ○ real easy to "install", "learn", and get started

☐ Debuggers
   ■ Commercial ones may cost a lot (home equity loan)
   ■ It's hard to scale debugging and debuggers
   ■ Requires user to do the heavy lifting
State of the Tools Address

- Automated tools can help some
  - Umpire, Marmot, MPI-Check, Intel Message Checker, NEC Collectives, MPICH2 collectives
  - Still in infancy, but I believe it’s the way to go

- A combination of tools would be best

- Why do users resist tools?
MPI Implementations

- No general test suite to validate/evaluate MPI implementations
  - Is ping-pong all that matters?
- Why won't users share their bad code?
  - Hmmm, I wonder
- Should the standard be improved?
Summary

- Productivity is important
  - Programming models and tools matter
- Is there a need for more than printf?
- What are the next steps?

Professor, I left the printf in there because it fixed the bug.
Jeff Squyres

☐ Research associate, Indiana University
☐ MPI user (years ago)
☐ MPI implementer
  ■ LAM/MPI
  ■ Open MPI
Jeff’s View: MPI Is Great / Horrible

☐ MPI does some things really well
   ■ “6 function MPI” (2% of MPI!)
   ■ Simple user models, simple MPI

☐ MPI does some things really poorly
   ■ Doing complex things can be hard
   ■ Datatypes can be great, but complex to setup
   ■ Some of MPI-2 is… er… complex
   ■ Performance portability can be… a challenge

☐ MPI implementations are not created equal
Jeff’s View: User Problems

- Startup / compile problems
  - “Dot” file issues / authentication
  - Mixing compiler suites
  - Mixing MPI implementations

- Run-time problems
  - Simple message passing issues
  - Assuming MPI implementation behavior
  - Memory problems (buffer overflow, etc.)
  - Heisenbugs

- Law of Least Astonishment
Jeff’s View: User Solutions

Three kinds of users:
- I’ll do it myself (printf debugging)
- I can figure out the code (debuggers)
- I can refactor the algorithm (tracing/perf. tools)

The parallel learning curve can be steep
- Many expect it to be identical to serial
- Not enough people use tools

Not all tools are free
- …but is there something better?
Community’s View

- What about MPI makes you scream?
- How can we simplify parallel debugging?
Conclusions

☐ We believe (but are biased):
  ■ Use the tools!

☐ Users need to tell us what you want
  ■ We want to hear the whacky ideas
  ■ Sign up on the sheet to continue this discussion in e-mail
Resources (Google for These)

☐ Correctness tools
  ■ Umpire, Marmot, MPI-Check, Intel Message Checker, NEC Collectives, MPICH2 collectives

☐ Tracing / performance tools
  ■ Vampir, Intel Trace Analyzer, TAU, MPE/Jumpshot, XMPI

☐ Debuggers
  ■ FX2, Totalview, DDT, PGDBG
  ■ Gdb, Valgrind, … other serial debuggers

☐ …and probably others!
Horror Stories

- What horror stories do you have?
  - What took forever to track down?
  - How could MPI or a tool helped?
Scalability

- How many people run with:
  - 4, 8, 16, 32, 64, 256, 512, …more processes

- What problems do you run into with scalability?
  - How can MPI or a tool help?
Multiple MPI Implementations

- How many people use the same application with different MPI implementations?
  - Do you have specific code paths for specific implementations? Why?
  - Is performance *always* the most important thing?
  - What other problems have you run into?
How do You Debug?

- How do you debug your parallel applications?
  - printf / trial and error
  - Performance / correctness / tracing tools
  - Serial debuggers
  - Parallel debuggers
  - Memory-checking debuggers
  - …something else?
Do You Use MPI-2?

- What parts?
  - Dynamic processes
  - One-sided communication
  - MPI_THREAD_MULTIPLE
  - Extended collective operations
  - External interfaces
  - Parallel I/O
  - C++ / F90 bindings

- How well supported are these features?
- What is missing from MPI?
Do You Want / Need Heterogeneous?

- Architecture
  - Data size
  - Data layout (e.g., endian)
  - Processor type / speed
  - Multi-process or multi-thread?

- Multiple networks
  - Non-uniform networks