MPI-NeTSim: A network simulation module for MPI

Brad Penoff and Alan Wagner
Dept. of Computer Science
University of British Columbia
Vancouver, Canada

Michael Tüxen and Irene Rüngeler
Dept. of Electrical Engineering and Computer Science
Münster University of Applied Sciences
Steinfurt, Germany

ICPADS 2009
Shenzhen, China
Dec 8 - 11
What is MPI?

**Message Passing Interface**

- The de facto standard for writing parallel applications in High Performance Computing
- Provides application portability from one system to the next
- Used by a distributed application with multiple processes exchanging messages to collectively accomplish a goal
MPI Execution Example

- Tools show computations and communications at the MPI level
  - MPI calls shown
  - Communications = lines connecting processes

- Details underneath are abstracted
  - Network effects not shown
  - All non-communications (computation, sleep, etc.) look the same
Motivation

• Understand **performance** of MPI programs
  – Can **easily** test computational requirements of an application by varying the number of processes

  – **Difficult** to vary the network to understand the network’s exact impact on performance
Goal

To have a tool where the network can be easily varied to study its impact on MPI application performance.
What makes varying the network hard?

- More **parameters** to vary inside a network
- More **difficult to measure** network performance
- MPI applications’ **communications vary**
- Application communication **requirements less clear** than CPU and memory
  - Application bandwidth and latency needs?
  - Will there be congestion? Other network effects?
  - What network is “good enough” (the least $$$)?
Design Goals of Tool

• Use **existing** MPI applications
• **Accurate** packet-level simulation
  – Network traces
  – Complete transport and middlebox statistics
• **No extra permissions** for devices/traces/stats
  – Can run in user-space
  – No special kernels
• **Easy to vary** the network!
MPI-NeTSim

- Use **existing** MPI applications
- Accurate **packet-level** simulation
  - Network traces
  - Complete transport and middlebox statistics
- **No extra permissions** for devices/traces/stats
  - Can run in user-space
  - No special kernels
- **Easy to vary** the network!
MPI-NeTSim Components

- MPI implementation
- Network simulator
MPI implementation = MPICH2

• Modular, open source MPI middleware developed at Argonne National Lab

• Several instances of communication components
  – TCP, shared memory, Infiniband, SCTP, etc.

• Modified our SCTP component for MPI-NeTSim
Network simulator = OMNeT++

• Discrete event simulation environment
Network simulator = OMNeT++

- Discrete event simulation environment
- Extensible, modular, component-based C++ simulation library and framework
- Easy to vary the network!
- Users get access to traces and network stats
- Users must model their applications
  - Can’t run them unmodified

From omnetpp.org
MPICH2 & OMNeT++ create MPI-NeTSim

Modified MPI implementation & simulator

- MPICH2 uses simulator for communication
- OMNeT++ uses external MPI processes as input for network and transport protocol simulation
MPI-NeTSim Design

- Only MPI communication component modified
  - Uses our SCTP-like API
- Processes connect to central IPCScheduler inside of the simulator
- Internal IPCApp per MPI process model a host
- Emulated processes using simulated network
Example MPI-NeTSim Execution
Network Simulation Problem

Ideal

Slow simulator causes Process A to wait!
Solution: Slowdown Technique

Time Factor = 4 is similar to ideal

Slow simulator causes Process A to wait!
To validate the slowdown technique...
Validation – Target Network

1 Gbps
Link Delay=1ms
As time factor increases, results converge and remain the same beyond a threshold.
Needed a metric for measuring the simulator’s accuracy...
Time slowdown technique converges to consistent runtime with an acceptable accuracy.

To illustrate the intended use of the tool...
Varying the Network
1) “is” has the most collectives, uses “all-to-all”
2) “bt”, “lu”, and “mg” have same collectives (e.g., broadcast)
3) “cg” only has a barrier
4) “ep” has no collectives and few communications
Conclusions

• Introduced and validated slowdown technique, for accurate results
  – Successfully “opened up” OMNeT++ for use with external processes
  – Modified MPICH2 to connect to OMNeT++

Overall, MPI-NeTSim is a useful tool for easily varying the network to do packet-level network experiments of real MPI applications.
Future Work

• Decrease maxBehind
• Scale to larger number of processes
  – Connect via TCP to the simulator
  – Modify the model for different bottlenecks
• Dynamically adjust the slowdown factor
• Use the tool more to explore the network impact on applications
• Simulate different interconnects and transport protocols (e.g., Infiniband)
Thank you!

謝謝您

Questions? Comments?

Google “mpi sctp” to see more of our work.
Extra Slides
Slowdown Technique Illustration

Time Factor = 4
Middleware for MPI

- Glues necessary components together for parallel environment