Semi-Automatic Models of Communication Volume and Frequency for SPMD Applications

Gabriel Marin
Oak Ridge National Laboratory

CScADS Workshop 2009
Why Performance Modeling

• Understand application behavior on current systems
• Understand how applications will perform at different scales or on future systems
• Gain insight into performance bottlenecks
• Identify barriers to scalability
Performance Modeling Challenges

• Performance depends on:
  — architecture specific factors
  — application characteristics
    – memory access patterns
    – instruction mix and schedule dependencies
    – communication frequency and bandwidth
  — input data parameters

• Analyzing performance at scale is expensive
Approach

Separate contribution of application characteristics

• Measure the application-specific factors
  — static analysis
  — dynamic analysis

• Construct scalable models

• Explore interactions with hardware
Single Node Performance Modeling

**Object Code**
- Binary Analyzer
  - Control flow graph
  - Loop nesting
  - Instruction dependences
  - BB instruction mix

**Static Analysis**
- IR code
- Modulo Scheduler
- Performance Prediction for Target Architecture

**Dynamic Analysis**
- Instrumented Code
  - BB & Edge Counts
  - Memory Reuse Distance

**Scalable Models**
- Modeling Program
  - Architecture neutral model
  - Evaluate

**Cross Architecture Models**
- Architecture Description
- Scalable Models
How to Extend to Parallel Programs?

• Performance scales with
  —input size
  —processor count

• MPI traces not suited for scalable modeling
  —number and type of MPI events in the trace vary with input size and processor count

• Prior work looked at
  —identifying patterns in traces
  —apply regression on the time spent in communication and computation
A Statistical Approach

• Think of program execution as a series of computation intervals

• Computation intervals bounded by two consecutive communication events

• Collect and aggregate data at interval level

• Model the frequency and cost of intervals as a function of
  — input size
  — processor count
An Early Preliminary Prototype

• Implemented on top of mpiP
• Modified mpiP to collect data at interval level
  — intervals uniquely defined by the stack unwinds of the two delimiting MPI primitives
• For each interval collect
  — information about computation cost
  — message size and communication cost for the MPI primitive closing the interval
• Aggregate information into histograms
  — histograms provide more insight than any single value statistic (e.g. median, mean+stdev)
Preliminary Results

- Collected data for Sweep3D on a Cray XT4 machine

- Solves a 3D cartesian geometry neutron transport problem

- IQ loop
  - MPI communication
  - Node computation
  - MPI communication
Flow Chart of Computation Intervals

• Nodes correspond to distinct MPI calls
• Edges represent different computation intervals
  – labels correspond to execution frequency
Data Collection

• For each interval collect
  — distribution of message sizes
  — distribution of communication times
  — distribution of computation times
  — several other scalar values

• Collect data for multiple input sizes and multiple processor counts

• Goal: model the structure and scaling of data histograms as a function of problem size and processor count
3D Histogram Representation

Normalized frequency

Message size

50%  30%  20%

2  13  40

Normalized frequency

Problem size

Message size
Distribution of Message Sizes

• Interval Recv_0x418c9a - Send_0x418c35
Distribution of Communication Times

- Interval Recv_0x418c9a - Send_0x418c35
Results for SMG2000

• Parallel semicoarsening multigrid solver
• Modified solver to execute a fixed number of iterations
• Collected data at interval level for different grid sizes and different processor counts
Distribution of Message Sizes

- As a function of grid size
Distribution of Communication Times

- As a function of grid size
Distribution of Message Sizes

- As a function of processor count
Distribution of Communication Times

- As a function of processor count
Summary

• This is a work in progress
  — no end-to-end predictions
  — preliminary results do not contradict the approach
  — Sweep3D results show that understanding topology is important

• Not a replacement for tracing and network simulators

• Wants
  — StackWalkerAPI and SymtabAPI