Prophesy: A Web-based Performance Analysis and Modeling System for Parallel and Grid Applications

Valerie Taylor
Texas A&M University

Xingfu Wu, Joseph Paris
Northwestern University

Rick Stevens, Ivan Judson, Mark Hereld
Argonne National Laboratory
Outline

- Prophesy System
- Prophesy Database
- Data Collection: PAIDE System
- Data Analysis: Model Builder
- Summary
Prophesy System

- Data Collection
  - PAIDE system

- Prophesy Database
  - Systems Database
  - Performance Database
  - Template Database

- Data Analysis
  - Model Builder
  - Symbolic Predictor
Prophesy Database

- Hierarchical organization
- Organized into 4 areas:
  - Application
  - Executable
  - Run
  - Performance Statistics
Applications

Application
(Molecular Dynamics)

Module (md.c)
Function (check_radius)

Module (partition.c)
Function (comp_inter)

Basic Unit (For(…))

Basic Unit
Basic Unit
Basic Unit
User Input

- User should register an account and an application online first.
- It requires information about the user and application such as user name, password, email, application name, and version, etc.
- Done once for all executables of the application the user owns.
Data Collection: PAIDE System

- Automated Source-code instrumentation at the multiple levels via PAIDE
- Support for C, Fortran77 and 90 programs
- Minimal instrumentation overhead and code
- Performance Data entered into the database automatically via PAIDE or manually via web site
PAIDE System

Source code

Parser

Instrumented source code

Compiler

Instrumented executable

Performance Data

Performance relations

Call Graph

SOAP Server with Perl Script

Prophesy Database
- **Options:**

  - **ALL:** Instrument all procedures and outer loops
  - **PROC:** Instrument all procedures
  - **LOOP:** Instrument all loops
  - **NOP:** Instrument all procedures not nested in loops
  - **FTP:** Use Perl SOAP scripts to automatically transfer performance data to the Prophesy database

**Default:** Instrument procedures and outer loops
Performance Data Files

For user and application:
- User name
- Password
- Email
- Application Name
- Application Version
For each executable:

✓ Executable Name
✓ Problem Size
✓ Total Number of Processors
✓ Total Execution Time
✓ Processor Number
✓ System Name
✓ Run Date and Time
For each event (procedure or loop):

- Event ID
- Start Line Number
- End Line Number
- Event type (Procedure or Loop)
- Procedure Name (if event type is Procedure)
- Caller Name
- Module Name
- Runtime
- Square of runtimes
Data Entry

✓ Use Perl SOAP scripts to automatically process the performance data files at the end of program execution, and put the data into the Prophesy database.

✓ Use web form interfaces to manually put the data into the Prophesy database.

✓ Use Perl SOAP script to automatically process performance data files generated by SvPablo, and put them into the database.
Data Analysis: Model Builder

- Develop performance models
  - Make runtime predictions
  - Identify best implementation
  - Identify performance trends and performance bottlenecks
Develop Performance Models

- Utilize information in the Prophesy databases
  - Performance database
  - Template database
  - System database
- Three techniques
  - Curve Fitting
  - Parameterization
  - Coupling
Curve Fitting Method

- Uses least squares
- Uses database information
  - Executable information
    - Runtime
    - Inputs (problem size)
    - Number of Processors
    - User selected model order
- Does not expose system parameters
Curve Fitting: Usage

Analytical Equation
(Octave: LSF)

Matrix-matrix multiply:
LSF : 3

Performance Data

Model Template

Application Performance
Function Performance
Basic Unit Performance
Data Structure Performance
Parameterization Method

- Requires manual analysis of the kernel or function
  - Hand count operations
  - Expose system parameters
  - Only needs to be done once per kernel
- Uses database information
  - System database
  - Model template database
**Parameterization: Usage**

**Analytical Equation**
*(Octave: Parameterization)*

**Matrix-matrix multiply:**
*Parameterization:*

Parameter(P, SGI Origin2000, N, ADDM, MPISR, MPIBC)

**System Data:**
MPISR, MPIBC, ADDM

---

**Model Template**

**Resource**

**Connection**

**Systems**
Coupling Method

- Represents an application in terms of its kernels or components
- Does not require manual analysis
- Uses database information
  - Coupling values
  - Performance data
Coupling Method: Usage

**Adjacent Kernels**

- Functions
- Control Flow

**Data and System Info**

- Run
- Inputs
- Systems

**Coupling Values and Performance data**

- Function Performance
- Coupling
Summary

- Instrument at the level of basic blocks and/or procedures automatically via PAIDE.
- Enter data into the database automatically via PAIDE or manually via web site.
- Present the automated modeling component of Prophesy with three techniques:
  - Curve Fitting
  - Parameterization
  - Coupling