LLVM for Interactive Modeling and High Performance Simulation

Peng Cheng  Nathan Brewton  Dale Martin
Model Based Design

MATLAB® SIMULINK®

- Aerospace and defense
- Automotive
- Biotech and pharmaceutical
- Electronics and semiconductors
- Industrial automation and machinery
- Medical devices
- …
An Example of Interactive Modeling

[Diagram showing a DC motor with V+ and V- terminals, a resistor, inductor, and rotational electromechanical converter with labels for Motor inertia J, Friction Mr, and a graph showing speed and current over time.]
An Example of Interactive Modeling

![Diagram of a DC motor with a resistor and inductor](image)

- **V+**
- **V-**

**Graphs:**
- **Speed (rpm)**
- **Current (A)**

**Diagram Components:**
- Resistor
- Inductor
- Rotational Electromechanical Converter
- Motor inertia J
- Friction Mr
An Example of Interactive Modeling
Shared Library-Based Simulation

Model

- Multiple Components
  - Inductor
  - Component
  - Inductor DLL
  - Component DLL
  - Disk
  - Inductor C Code
  - Component C Code
  - \( u = L \frac{di}{dt} \)
  - Component Equation
  - Component
  
Model
Shared Library-Based Simulation

Model

- Multiple Components
  - Inductor
  - Component
    - Inductor DLL
    - Component DLL
      - Expensive Disk IO
      - Loss of Equation and IR Information
        - Inductor C Code
        - Component C Code
          - \( u = L \frac{di}{dt} \)
          - Component Equation
            - Locally Optimized Execution Code

- High Cost Interactive Modeling
- Disk Loss of Equation and IR Information
- Expensive Disk IO
- Interactive Modeling

- Model optimizations
  - Variable elimination
  - Index reduction
  - Constant folding
  - Function inline
Multiple Thread LLVM-Based Simulation

Model

Multiple Components

Inductor

Component

Component Equation

Model Equation

Multiple Threads

LLVM IR

Multiple Threads

LLVM IR

Optimized LLVM IR

External Functions

Machine Code

Optimized LLVM IR

External Functions

Machine Code

Model

Multiple Components

Inductor

Component

DLL

Loss of Equation and IR

DLL IOs

Inductor DLL

Low Quality Execution Code

Component DLL

High Cost Interactive Modeling
Multiple Thread LLVM-Based Simulation

**Model**
- Multiple Components
  - Inductor
    - \( u = L \frac{di}{dt} \)
  - Component
    - Component Equation

**Model Equation**
- Global Optimization in Model Equation
- LLVM IR
  - Global Optimization in LLVM IR
  - LLVM IR Functions
  - Globally Optimized Execution Code
- No Disk IO
- LLVM IR
  - Optimized LLVM IR
  - External Functions
  - Low Cost Interactive Modeling

**Model**
- Multiple Components
  - DLL Disk IOs
    - DLL
  - Inductor DLL
  - Component DLL
  - Low Quality Execution Code
  - High Cost Interactive Modeling

- Loss of Equation and IR
Results

- LLVM-based backend of new simulation engines in production code
  - Consistent floating point numerical computation
  - Support multiple threads on 64bit Linux, 32bit and 64bit Windows, 64bit Intel Mac

- Performance (win64 with LLVM 3.2 JIT)
Challenges

- Shared library support for large scale software
  - Multiple linkage of static LLVM library is painful

- JIT performance
  - CodeGenOpt: None, Less, Default, Aggressive (targeted for code quality)
  - New option targeted for fast JIT?

- Exception handling on Windows
  - Propagate through JIT code the exception thrown from the external functions

- Legacy JIT to MCJIT transition
Thank You!