A Platform-Independent Tool for Modeling Parallel Programs

ACM Southeast 2011

Kennesaw State University

March 2011

Ferosh Jacob, Jeff Gray, Yu Sun, Purushotham Bangalore

Department of Computer Science
University of Alabama
fjacob@crimson.ua.edu, gray@cs.ua.edu

THE UNIVERSITY OF ALABAMA COMPUTER SCIENCE

Department of Computer and Information Sciences
University of Alabama at Birmingham
{yusun, puri}@cis.uab.edu

Overview

Problem formulation

- Parallel programming models
- Current Challenges
- Possible solution approach

PPmodel

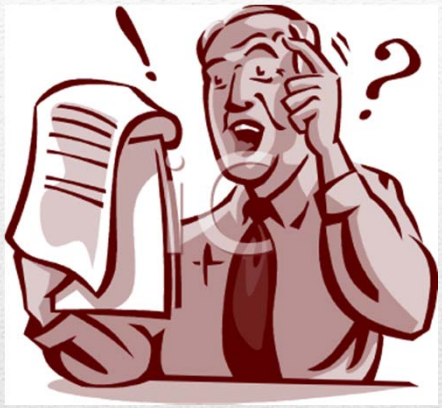
- From Developers viewpoint
- Implementation details
- Related work

Conclusion

Main contributions

CUDA

p-threads



OpenMPI

OpenCL

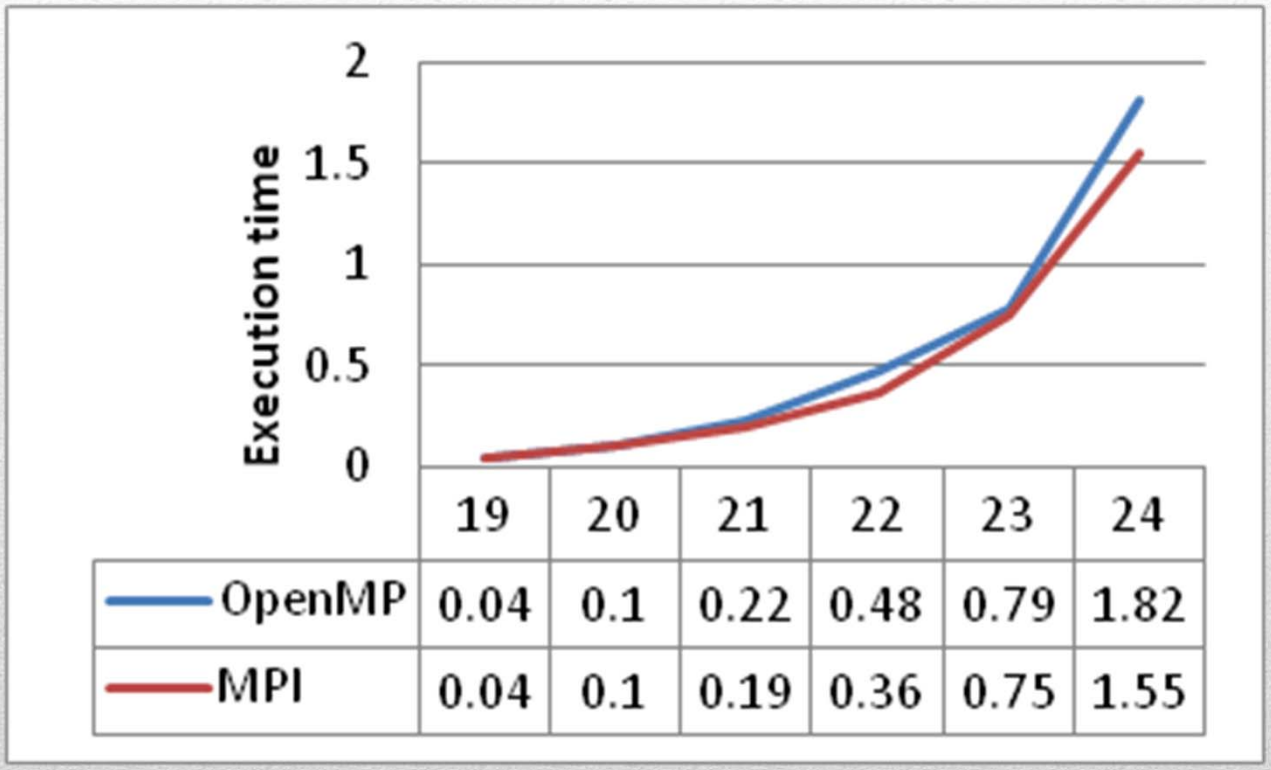
OpenMP

Cg

Programming Models

Parallel Programming Challenges

- **Which programming model to use?**



	19	20	21	22	23	24
OpenMP	0.04	0.1	0.22	0.48	0.79	1.82
MPI	0.04	0.1	0.19	0.36	0.75	1.55

- **Why are parallel programs long?**

No	Program Name	Total LOC	Parallel LOC	No. of blocks
1	2D Integral with Quadrature rule	601	11 (2%)	1
2	Linear algebra routine	557	28 (5%)	4
3	Random number generator	80	9 (11%)	1
4	Logical circuit satisfiability	157	37 (18%)	1
5	Dijkstra's shortest path	201	37 (18%)	1
6	Fast Fourier Transform	278	51 (18%)	3
7	Integral with Quadrature rule	41	8 (19%)	1
8	Molecular dynamics	215	48 (22%)	4
9	Prime numbers	65	17 (26%)	1
10	Steady state heat equation	98	56 (57%)	3

Parallel Programming Challenges

Modeling Parallel Programs

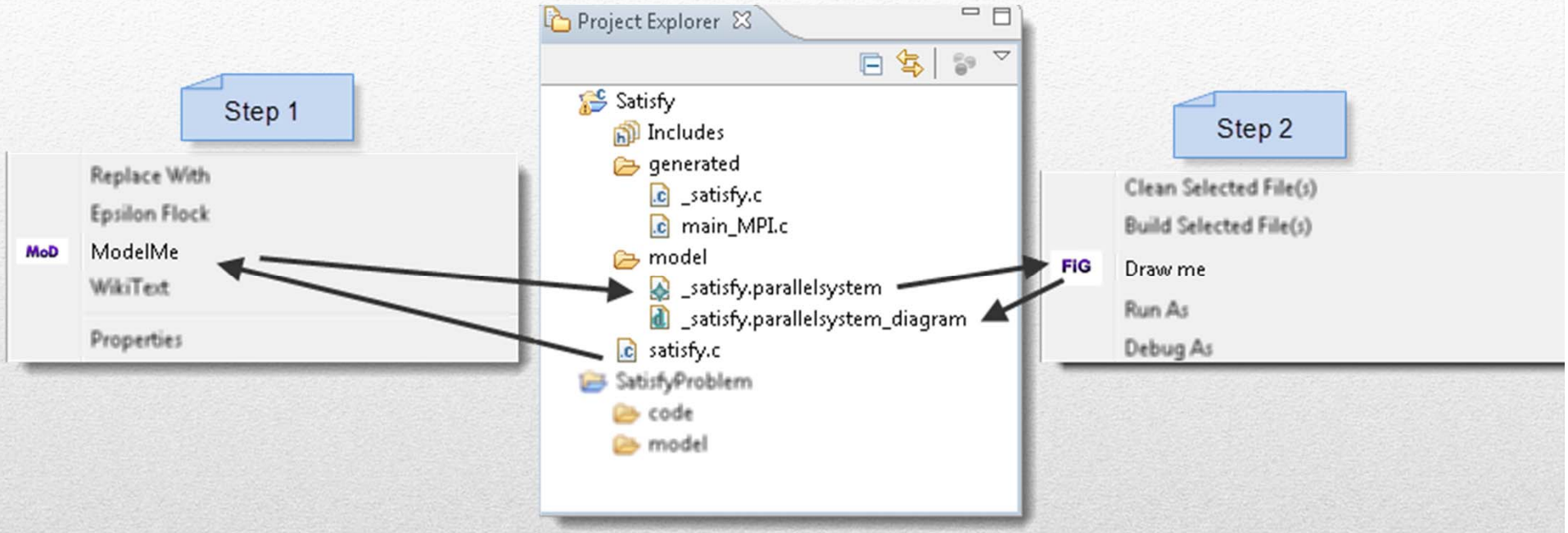
1. **Goals**
   - Separate the parallel sections from the sequential parts of a program, which allows a programmer to focus more on the parallelism.
   - Define a new execution strategy for the computation intensive part of the program without changing the flow of the program.
2. **PPmodel**
   - Using PPmodel, the parallel part of the program can be separated from the sequential part of the program, re-designed, and then regenerated.
   - Programmers can switch between technical solution spaces (e.g., MPI , OpenMP, CUDA and OpenCL) without actually changing the program.

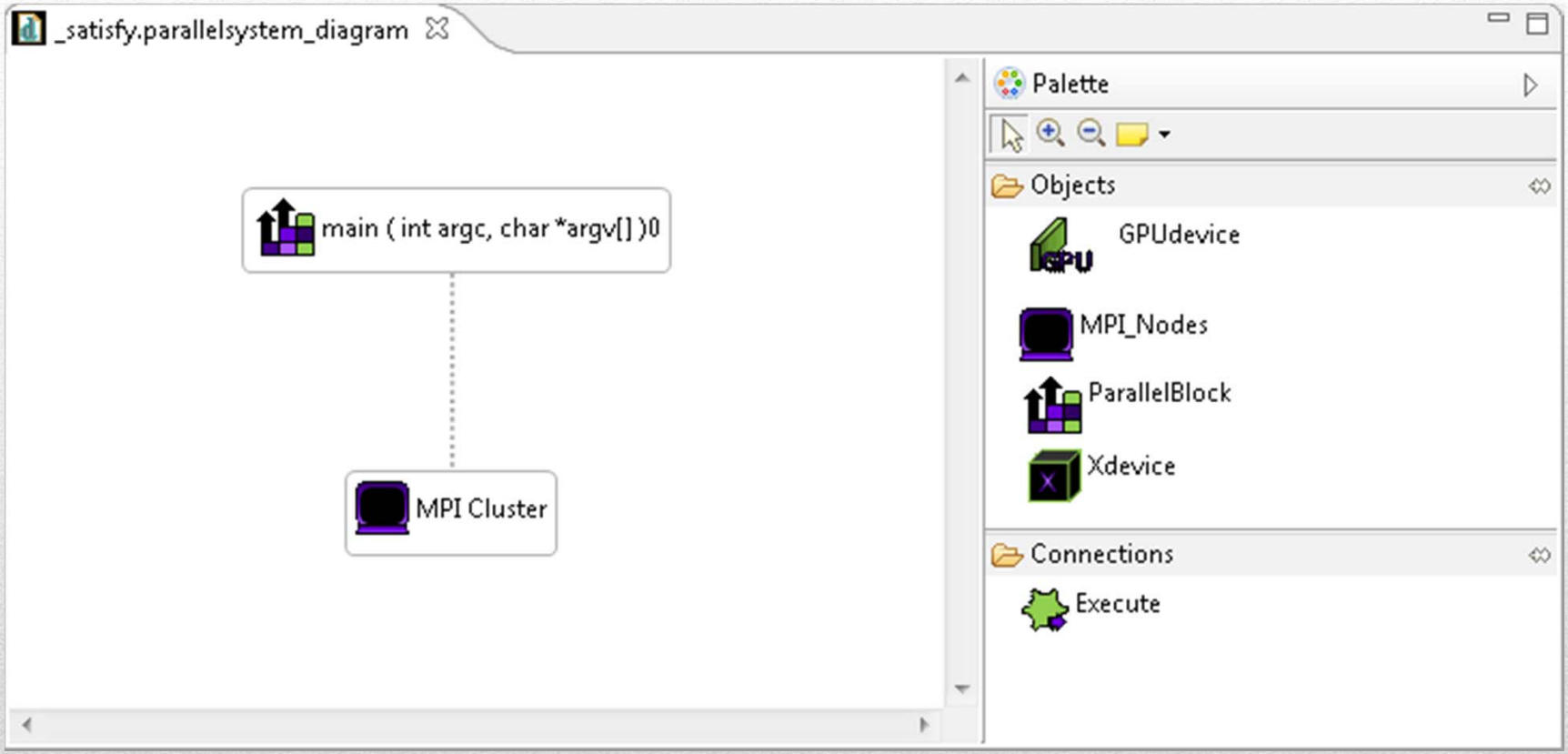
Working with PPmodel

PPmodel is explained from a user's perspective with the Circuit Satisfiability problem example.

1. **Model creation for Circuit Satisfiability problem**
   - Create a model from an existing program: *The parallel part of the program is extracted from the sequential part and stored separately.*
   - Create a visual representation of the model: *The model representation of the cluster is generated.*
2. **Modeling the Circuit Satisfiability problem:** *Modeling helps the programmer specify the automatically detected blocks to execute in a different platform.*
3. **Code generation :** *Code integration involves replacing the OpenMP code with the newly added MPI code, adding libraries to execute MPI code, and some code to initialize MPI-specific variables.*

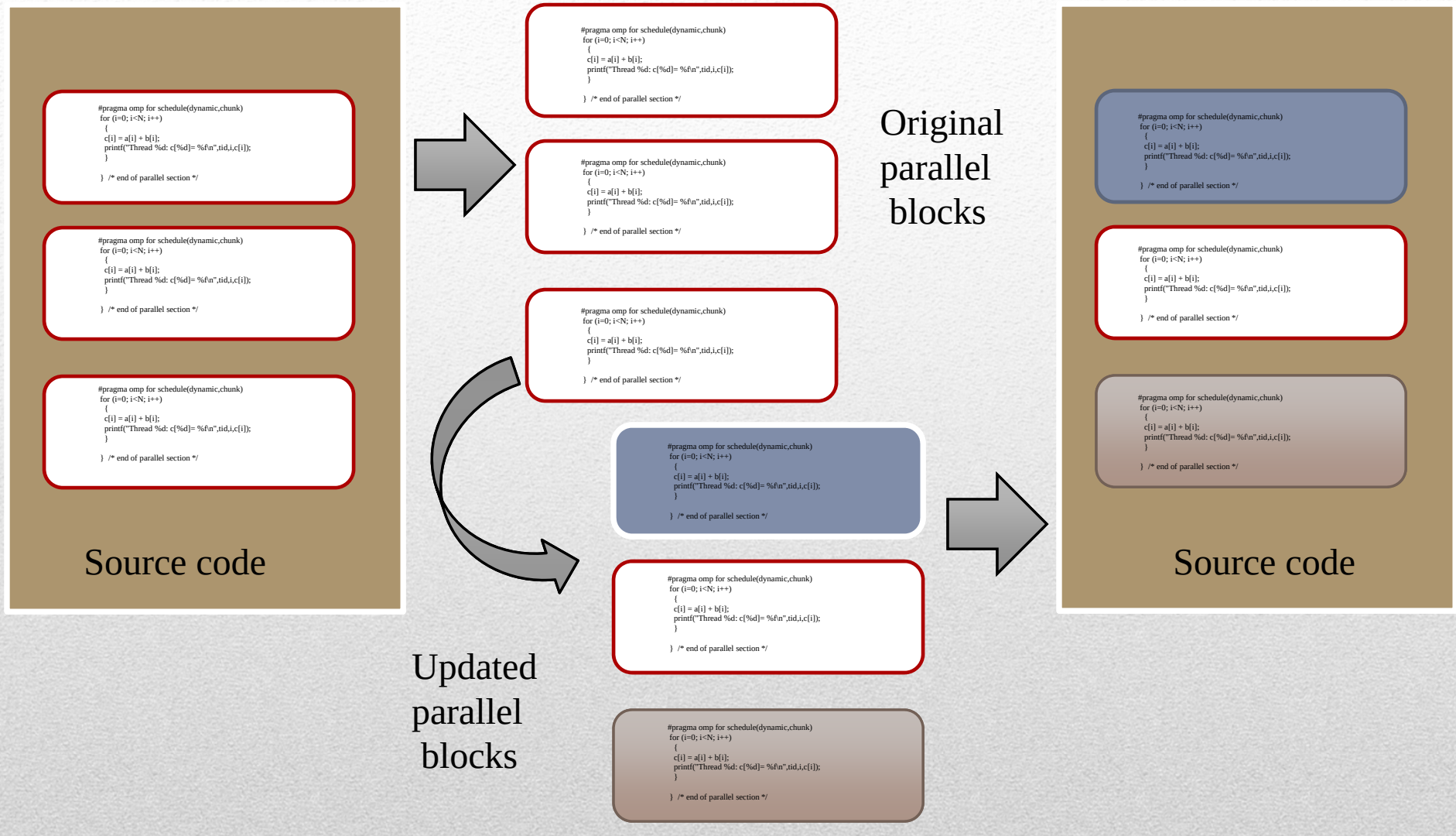
Working with PPmodel





Working with PPmodel

Implementation Details of PPmodel



Related Works

There have been a few modeling efforts in the parallel programming domain.

- **Graphical programming languages:** CODE programming language is based on a generalized dependency graph to express the computation in a unified parallel computation model without any implementation details.

  CODE is a graphical programming environment, but PPmodel is a complete modeling tool to create parallel programs from sequential or parallel programs written for another target platform. GASPARD is another visual parallel programming environment supporting task and data parallelism.

Related Works

There have been a few modeling efforts in the parallel programming domain.

- **OpenMP to GPGPU:** converts OpenMP programs to CUDA code

  The goal of PPmodel is to express the parallel part of a program in a way separated from the sequential part so as to allow the programmers to focus more on the parallel problem than the program as a whole.

- **Program transformation tools:** *Other related works include program transformations from sequential to parallel and abstractions in parallel programs.*

- The tool currently can model only C OpenMP programs and generate target code for the MPI library.
- PPmodel can be extended to support a GPU programming language like CUDA.
- Similar implementations can be created for other programming languages and platforms.
- The programming language determines the refactoring framework to use and the platform decides the code to be inserted or re-factored

Limitations and Future work

Conclusion

- PPmodel is a tool that can be used to separate the parallel part from the sequential part of a program.
- Using the modeling PPmodel framework, programmers can execute the parallel blocks in a different platform without actually rewriting the program.
- The approach is independent of any platform or language and can be extended to any language.
- An OpenMP program written to solve the Circuit Satisfiability problem was redesigned to execute in multiple nodes using MPI.

Questions

Thanks