Picoso
A Parallel Interval Constraint Solver

Natalia Kalinnik\textsuperscript{1}, Tobias Schubert\textsuperscript{1}, Erika Ábrahám\textsuperscript{2}, Ralf Wimmer\textsuperscript{1} and Bernd Becker\textsuperscript{1}

\textsuperscript{1}Institute of Computer Science
Albert-Ludwigs-University of Freiburg
Germany

\textsuperscript{2}Chair of Computer Science II,
RWTH Aachen,
Germany

PDPTA 2009
1 Motivation

2 iSAT – Sequential Interval Constraint Solver

3 Picoso – Parallel Interval Constraint Solver

4 Experimental Results on Picoso
Motivation
Motivation

- Embedded systems are widely used (automotive controllers, air traffic management, medical devices, robots . . .)
- Complexity and safety-critical nature pose a major challenge for verification approaches
- Hybrid systems: discrete + continuous behavior
- Dynamics of hybrid systems may be non-linear (like $x = \sin(y)$)
Satisfiability Modulo Theories (SMT)

- The behavior of a hybrid system can be encoded as a formula
- Boolean combinations of non-linear arithmetic constraints

\[(x = \cos(y) \lor y = e^x) \land (x^2 = 2 \cdot y \lor x \geq 0)\]

SMT Formula

- Is the formula satisfiable, i.e. is there an assignment of variables such that the formula evaluates to true?
  
  Problem is undecidable in general

- Most of the SMT approaches support only decidable theories
- First parallel SMT solver
iSAT – Sequential Interval Constraint Solver
Merges Davis-Putnam-Logemann-Loveland (DPLL) and Interval Constraint Propagation

Manipulation of interval bounds:
\[ x \in [3, 7], \quad y \in [-2, 25] \]

Alternating Deductions and Decisions

**Deductions:** prune off definite non-solutions

- Unit propagation:
  \[ \cdots \land (x > 8 \lor y = x^2) \land \cdots \]
- Interval Constraint Propagation
Interval Constraint Propagation (ICP)

\[ x \in [3, 7] \land y \in [-2, 25] \land y = x^2 \Rightarrow x \in [3, 5] \land y \in [9, 25] \]
Decisions:
- Split interval in two halves
- Decide in which half to search first
- Propagate resulting information

Learning:
- Deduction can lead to an empty interval $\Rightarrow$ Conflict
- Learn reasons of a conflict in form of conflict clause

Termination:
Stop search when
- All branches lead to conflicts, formula is unsat
- Reasonably small conflict-free box is found

*Incomplete due to the interval splitting and Interval Constraint Propagation, i.e. iSAT can terminate with approximate solution*
Picoso – Parallel Interval Constraint Solver
Parallel SMT-solver for boolean combinations of linear and non-linear arithmetic constraints
Based on master/client model
Communication is carried out using Message Passing Interface (MPI)
Algorithmic core of the clients is formed by iSAT
Communication

- All communication takes place between master and the clients
All communication takes place between master and the clients
All communication takes place between master and the clients
Communication

- All communication takes place between master and the clients
All communication takes place between master and the clients.

To reduce the communication overhead, only conflict clauses of a bounded length ($\leq 6$) are distributed.
Parallelization Idea

- Divide the search space on demand into disjoint parts and solve the individual subproblems in parallel
- Use decision variables for search space partition
- To speed up the search, information in form of conflict clauses is exchanged

Initial Intervals

\[ x \in [0, 6] \quad y \in [2, 6] \quad z \in [-1, 4] \]

Implication

Decision

\[ x \in (3, 6) \]

\[ x \in [0, 3] \]

Client 1
Divide the search space on demand into disjoint parts and solve the individual subproblems in parallel

Use decision variables for search space partition

To speed up the search, information in form of conflict clauses is exchanged
Dynamic work stealing is based on the extended concept of *guiding paths*

A *guiding path* is a sequence of bounds:

- consisting of all decisions and implications and
- a flag, indicating which subproblem still needs to be solved

Each bound with a flag “B” is a candidate for a search space division
Generation of New Subproblems

Initial Intervals

\[ x \in [0, 6] \quad y \in [2, 6] \quad z \in [-1, 4] \]

Implication

Decision

\[ x \in (3, 6) \quad x \in [0, 3] \]

Client 1

\[ GP_1 = [ (x \geq 0, N), (x \leq 6, N), (y \geq 2, N), (y \leq 6, N), (z \geq -1, N), (z \leq 4, N), (x \leq 3, B) ] \]
Generation of New Subproblems

\[
GP_1 = [ (x \geq 0, N), (x \leq 6, N), \\
(y \geq 2, N), (y \leq 6, N), \\
(z \geq -1, N), (z \leq 4, N), \\
(x \leq 3, N) ]
\]
Generation of New Subproblems

Initial Intervals

\[ x \in [0, 6] \]  \[ y \in [2, 6] \]  \[ z \in [-1, 4] \]

Implication

\[ x \in (3, 6] \]  \[ x \in [0, 3] \]

Decision

\[ x \in (3, 6] \]

\[ x \in [0, 3] \]

\[ GP_1 = \left[ (x \geq 0, N), (x \leq 6, N), (y \geq 2, N), (y \leq 6, N), (z \geq -1, N), (z \leq 4, N), (x \leq 3, N) \right] \]

\[ GP_2 = \left[ (x \geq 0, N), (x \leq 6, N), (y \geq 2, N), (y \leq 6, N), (z \geq -1, N), (z \leq 4, N), (x > 3, N) \right] \]
Generation of New Subproblems

\(x \in [0, 6]\)
\(y \in [2, 6]\)
\(z \in [-1, 4]\)

Initial Intervals

\(x \in (3, 6]\)

Decision
\(x \in [0, 3]\)

\(GP_1 = \left[ (x \geq 0, N), (x \leq 6, N),
(y \geq 2, N), (y \leq 6, N),
(z \geq -1, N), (z \leq 4, N),
(x \leq 3, N) \right] \)

\(GP_2 = \left[ (x \geq 0, N), (x \leq 6, N),
(y \geq 2, N), (y \leq 6, N),
(z \geq -1, N), (z \leq 4, N),
(x > 3, N) \right] \)

\(GP_2' = \left[ (x \leq 6, N),
(y \geq 2, N), (y \leq 6, N),
(z \geq -1, N), (z \leq 4, N),
(x > 3, N) \right] \)
Guiding Paths with a Volume-Based Heuristic

Initial: $x \in [0, 6], y \in [2, 8], z \in [-1, 4]$

Which client ($C_1$ or $C_3$) should be selected to split-off a sub-problem?
Which client \((C_1 \text{ or } C_3)\) should be selected to split-off a sub-problem?
Guiding Paths with a Volume-Based Heuristic

Initial: \( x \in [0, 6], y \in [2, 8], z \in [-1, 4] \)

\[
\begin{align*}
V(\text{GP}(C_1)) &= 2 \cdot 1 \cdot 3 = 6 \\
V(\text{GP}(C_3)) &= 3 \cdot 6 \cdot 5 = 90
\end{align*}
\]

We choose the one whose guiding path has maximal volume.
Picoso terminates with

- UNSAT, if all clients are idle, or with
- SAT, if a client was able to find a model, or with
- APPROXIMATE SOLUTION, if one of the clients has found an approximate solution
Experimental Results on Picoso
Experimental Results

- Workstation with eight 2.3 GHz AMD processor cores
- One master process and 2, 4, and 8 clients
- Clients are allowed to share conflict $\leq 6$
- Computation times for Picoso are the arithmetical mean over 3 runs
- Time-out limit: 30000 s
iSAT and Picoso with 2, 4, and 8 Clients

<table>
<thead>
<tr>
<th>benchmark</th>
<th>iSAT</th>
<th>Picoso_2</th>
<th>Picoso_4</th>
<th>Picoso_8</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed-up</td>
<td>3.16</td>
<td>3.67</td>
<td></td>
<td>6.21</td>
</tr>
<tr>
<td>#Timeouts</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Future Work

- Heuristics for conflict clause sharing
- Algorithm Portfolio Design
- Combination of different strategies (Search Space Splitting and Algorithm Portfolio)
- Integration of shared memory concepts
Parallel SMT-solver for the first-order theory of the reals extended with transcendental functions

The algorithmic core of the clients is formed by iSAT

Employs dynamic work stealing based on guiding paths and volume heuristics

Exchanges information in form of conflict clauses

Yields a (nearly) linear speedup for both linear and non-linear benchmarks
Thank you!