Fast Distributed Property Checking

P. Nalla, P. Peranandam, J. Ruf, S. Lämmermann, J. Behrend, R. Weiss, T. Kropf, W. Rosenstiel
Computer Engineering Department
University of Tübingen, Germany

Verification Methodology

- bounded property checking
- universal and existential reasoning about finite traces with AR automata
- keep only current state set, no fix-point
- standard kernel uses BDD-based image computation
- symbolic traversal of system model

Optimizations

- Standard Optimizations
  - cone of influence reduction
  - partitioned transition relation
  - early quantification

- Early result detection
- Reduce memory consumption

Motivation

- Symbolic Verification Methodology
  - k
  - t1
  - Problems:
  - late nodes remove states already visited at step
  - overlap between state subsets

Parallelization

- Basic Approach
  - traverse initial state set until threshold size reached
  - MPI for communication, state sets dumped to network drive

- Advantages
  - computation on a smaller subset requires less memory
  - not sensitive to the partition scheduling
  - enables verification of larger models

State set distribution

- Characteristics of Parallelization
  - two stages (sequential and parallel)
  - asynchronous communication
  - dynamic load balance between computation each node
  - distributed termination detection

Splitting algorithm

- Minimal overlap
  - aims at minimizing the state overlap

- Algorithm – step by step
  - statically analyze the partitioned transition relations
  - make a dependency table
  - present state variable with the influenced next state variables
  - variables with the influence factor are selected for splitting

Approach using dynamic overlap reduction

- Overlap Reduction
  - uses $n = 1$ number of nodes, where $n = \text{pow}(2,k)$ and $k \in \{1, 2, 3, \ldots\}$
  - state overlap main scalability problem
  - nodes dump their current state set every $n$ steps
  - later nodes remove states already visited at step by other nodes
  - overlap reduction and dynamic load balancing
  - Minimal overlap algorithm is used for splitting

Windowning approach

- Goal
  - reduce the effort spent on network nodes by avoiding the redundant computations
  - balanced partitions

- Algorithm – step by step
  - Kepler Cluster Tübingen
  - 98 Dual Intel Pentium 3 @ 650 MHz, 1 GB main memory per node
  - 32 Dual AMD Opteron @ 1.67 GHz, 2 GB main memory per node

Computation environment

- Kepler Cluster Tübingen
- 98 Dual Intel Pentium 3 @ 650 MHz, 1 GB main memory per node
- 32 Dual AMD Opteron @ 1.67 GHz, 2 GB main memory per node

Results

- Performance graphs
- checked circuits:
  - ISCAS2009 benchmarks
  - a model of a robotic production line
  - IBM benchmarks