Harnessing the power of GPUs for model checking

TU/e Technische Universiteit Eindhoven University of Technology

Anton Wijs
Eindhoven University of Technology

joint work with

Dragan Bošnački
Eindhoven University of Technology

Stefan Edelkamp & Damian Sulewski
University of Bremen

Joost-Pieter Katoen
RWTH Aachen University

On-The-Fly State Space Exploration

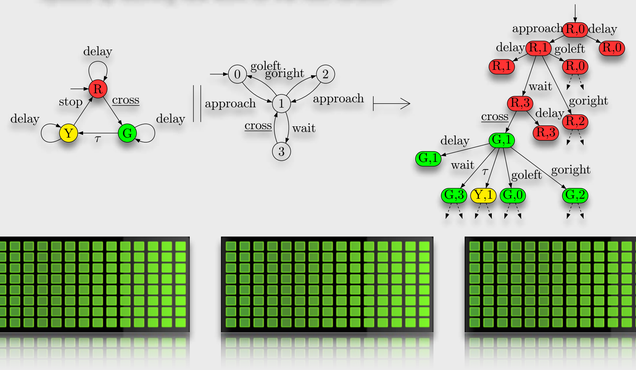
Construct a state space, given a model of a concurrent system [3]
Model = set of interacting finite-state Labelled Transition Systems

New hash-table design for GPUs, with fine-grained parallelism
Elements are placed in buckets using *warp-the-line* technique

Threads work in groups to generate state successors
Parallelism at state-level

Block-local shared memory used for state caches
Local duplicate detection reduces global hash table access

Work forwarding per block from one search iteration to the next
Speeds up fetching new work for the next iteration



10-100x speedup

State Space Decomposition

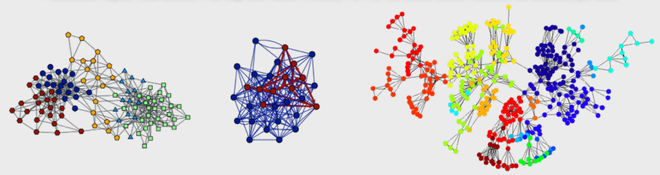
Decompose explicit graph into Strongly Connected Components
&
Decompose graph of Markov Decision Process into Maximal End Components [2]

Decompositioning based on Forward/Backward Breadth-First Search
Uses *trimming* to remove trivial components in each iteration

In each iteration, many BFSs can be performed in the search regions in parallel
Achieves massive parallelism

Novel combined forward/backward thread kernel
Combines both procedures in one state scan

New simplified pivot selection for each search region at the start of an iteration
Reuses input transition array as hash table for enforced data races to select pivots



15-79x speedup

Probability Computations

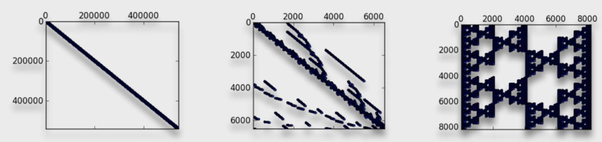
Perform numerical computations for probabilistic model checking [1, 4]
Needed to check if a probabilistic property holds in a discrete or continuous time Markov Chain

Solving systems of linear equations and performing matrix-vector multiplication
Parallel matrix-vector multiplication used in Jacobi method for solving equation systems

Parallel termination checking achieves significant speedup
Fast checking if next iteration is needed

Novel restructuring of input ensures coalesced memory access by threads
Faster reading of input reduces multiplication run time up to four times

States / transitions are grouped in segments of 16 and 32 states
Coincides with a half and a full *warp* of threads



20-35x speedup

References

[1] ***Parallel Probabilistic Model Checking on General Purpose Graphics Processors***
D. Bošnački, S. Edelkamp, D. Sulewski, and A.J. Wijs
International Journal on Software Tools for Technology Transfer 13(1) 21-35 (2011)

[2] ***GPU-Based Graph Decomposition into Strongly Connected and Maximal End Components***
D. Bošnački, J.-P. Katoen, and A.J. Wijs
in *Proceedings of the 26th International Conference on Computer Aided Verification (CAV'14)*, volume 8559 of LNCS, pp. 309-325 (2014)

[3] ***GPUexplore: Many-Core On-The-Fly State Space Exploration Using GPUs***
A.J. Wijs and D. Bošnački
in *Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'14)*, volume 8413 of LNCS, pp. 233-247 (2014)

[4] ***Improving GPU Sparse Matrix-Vector Multiplication for Probabilistic Model Checking***
A.J. Wijs and D. Bošnački
in *Proceedings of the 19th International SPIN Workshop on Model Checking of Software (SPIN'12)*, volume 7385 of LNCS, pp. 98-116 (2012)

Tools available at http://www.win.tue.nl/~awijs