Modeling, Analysis, and Control of Software Execution for Failure Avoidance

Hongwei Liao, Hyoun Kyu Cho, Stéphane Lafortune, and Scott Mahlke – *EECS Department, University of Michigan, Ann Arbor*
Yin Wang and Terence Kelly – *HP Labs*
Ahmed Nazeem and Spyros Reveliotis – *School of ISyE, Georgia Institute of Technology*

Graduate Symposium 2010
"Engineering for EnGREENing the World!"

Motivation

- The era of multicore architectures in computer hardware brings an unprecedented need for parallel programming. We are interested in multithreaded programs with shared data, where lock primitives are used to control access to the shared data.
- Mutual exclusion locks (mutexes) are used to protect shared data from inconsistent concurrent updates. Improper use of mutexes can lead to the "deadly embrace" problem and Circular-Mutex-Wait (CMW) deadlocks.
- In our on-going project, called Gadara, the objective is to control the execution of multithreaded programs in order to avoid deadlocks by using techniques from discrete control theory (see Fig. 1).

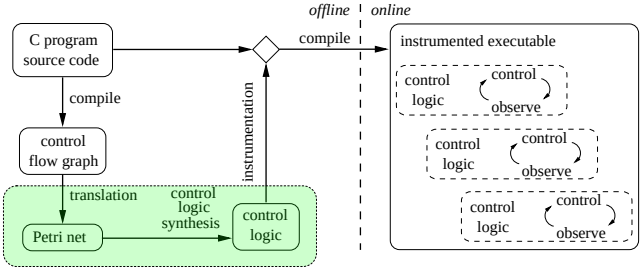

Fig. 1. The Gadara architecture

Main Contributions

- We propose a systematic program control architecture (see Fig.1) based on Discrete Control Theory, which combines the strengths of offline analysis and control synthesis with online knowledge and control to dynamically avoid CMW deadlocks in concurrent programs.
- We define a special class of Petri nets, called Gadara nets, to formally model lock allocations and releases in multithreaded programs (see Fig. 2).
- We establish necessary and sufficient conditions for liveness and reversibility of Gadara nets, which connect these *behavioral* properties of the dynamic system to a certain *structure* in the net, called siphon.
- We develop an efficient control synthesis strategy for Gadara nets, based on structural analysis. The strategy is both *correct* and *maximally permissive* with respect to the goal of liveness enforcement for Gadara nets.

Acknowledgment

- The Gadara project is supported by the National Science Foundation and by HP Labs.

Main Properties of Gadara Nets

Theorem 1:

	Program is deadlock-free	(Goal)
⇔	Gadara net is live	(Behavioral Property)
⇔	Gadara net cannot reach a problematic siphon	(Structural Property)

Iterative Control Of Gadara nets (ICOG)

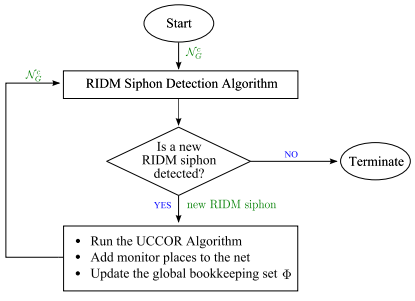

Fig. 3. Flowchart of ICOG

Properties of ICOG:

Theorem 2: ICOG is correct and maximally permissive with respect to the goal of liveness enforcement for Gadara nets.

Theorem 3: ICOG converges in a finite number of iterations.

Experiments

- Gadara has been tested on a set of real-world programs, e.g., OpenLDAP, BIND, Apache. Gadara identifies and avoids both previously known and unknown deadlocks while adding performance overheads ranging from negligible to 10%.
- In benchmark tests, Gadara successfully avoids injected deadlock faults, imposes negligible to modest performance overheads (at most 18%).

References

1. T. Kelly, Y. Wang, S. Lafortune, and S. Mahlke. Eliminating concurrency bugs with control engineering. IEEE Computer, 42(12):52–60, December 2009.
2. H. Liao, S. Lafortune, S. Reveliotis, Y. Wang, and S. Mahlke. Synthesis of maximally-permissive liveness-enforcing control policies for Gadara Petri nets. In CDC 2010.
3. Y. Wang, H. Liao, S. Reveliotis, T. Kelly, S. Mahlke, and S. Lafortune. Gadara nets: Modeling and analyzing lock allocation for deadlock avoidance in multithreaded software. In CDC 2009.
4. Y. Wang, S. Lafortune, T. Kelly, M. Kudlur, and S. Mahlke. The theory of deadlock avoidance via discrete control. In POPL 2009.

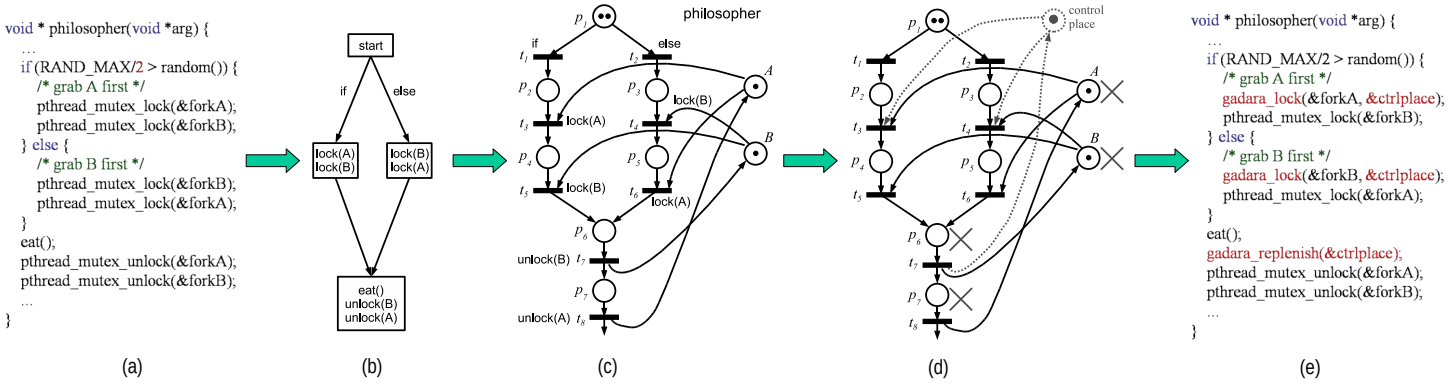

Fig. 2. Example of Gadara process