PhD Qualifier Exam for Simone Atzeni
Examiners: Ganesh, Zvonimir, Ryan, Hari

Start : Jan 18, 5pm
End : Jan 25, 5pm

Answer submission: Via a TinyURL that we can read

Ganesh

This paper by Raychev describes parallelizing user-defined aggregations using Symbolic Execution

http://research.microsoft.com/pubs/256579/143-raychev.pdf

Describe the ideas in this paper and see if these ideas could be used for OpenMP race detection (inability to parallelize = race). If not, why not, or in which cases? Could this help your research in any way?

Zvonimir

Thoroughly study the following paper by Koushik Sen on using dynamic testing to establish whether an alarm is a real data race or not:
K. Sen, "Race Directed Random Testing of Concurrent Programs," in Proc. ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'08), 2008, pp. 11-21.
http://srl.cs.berkeley.edu/~ksen/papers/raced.pdf

Describe in detail the technique introduced in this paper. Make sure to critically assess its pros and cons. In addition, connect this work to your own research. Could you envision it being used in your own work on OpenMP data race detection? If so, then describe how. If not, then clarify why not. Your response should be roughly 3-4 pages long. Good luck.

Ryan

Read and consider:

Read-Log-Update A Lightweight Synchronization Mechanism for Concurrent Programming
Matveev, et al. SOSP 2015
http://sigops.org/sosp/sosp15/current/2015-Monterey/printable/077-matveev.pdf

In 3 to 4 pages:
1. Briefly outline how RLU works.
2. Explain the problem that RLU Deferring (Section 3.7) solves and give a pathological example scenario where RCU's "synchronize" operation would have high overheads but RLU Deferral would not.
3. Explain: can RLU Deferral result in linearizability violations?
4. Explain: does code using RLU Deferral in 3.7 meet the definition of being "lock-free".

Hari

Read the following paper

Distributed wait state tracking for runtime MPI deadlock detection, Hilbrich et al., SC 13
http://dl.acm.org/citation.cfm?doid=2503210.2503237

1. Summarize the paper.
2. What are the main challenges for distributed deadlock detection?
3. Can you suggest an alternative distributed algorithm for deadlock detection or and improvement over the proposed algorithm?
4. What is the overhead of the proposed method? Will this vary depending on the kind of application? Which applications are affected most?
