6.895 Theory of Parallel Systems Wang Junqing (HT035516N)

Project Proposal

Content:

I suppose to make a Nondeterminator data-race detector run in parallel in my project.

The main function of Nondeterminator is to detect that whether there exists determinacy races in a parallel-multithreaded program. The current version of Nondeterminator we have [1] is a serial program based on SP-bags algorithm, which uses the fact that any Cilk program can be executed on one processor in a depth-first fashion and conform to C program when removing all the Cilk reserved words.

The SP-bags algorithm is the core of the Nondeterminator and it seems inherently serial, because it heavily relies on the serial execution order of the parallel program. Nevertheless, it may be possible to develop a parallel version of the SP-bags algorithm. That's what I plan to do in this project.

In [1], it suggests a parallel scheme in which each of several processors executing the program uses the SP-bags algorithm locally, but when a remote child procedure returns, it reconciles its shadow spaces in a manner similar to the BACKER algorithm [2], [3] for maintaining dag consistency.

But as stated in [1], such a result may be mostly of theoretical interest. However, since debugging is usually done in the development phase of a program using small data sets, and thus typically, the performance of the debugger is not a crucial concern.

Reference:

1. *Efficient Detection of Determinacy Races in Cilk Programs*, M.Feng, C.E.Leiserson.
2. *An analysis of dag-consistent distance shared-memory algorithms*, R.D.Blumofe, M. Frigo, C.F. Joerg, C.E.Leiserson, K.H.Randall.
3. *'Dag-Consistent Distributed Shared Memory,* R.D.Blumofe, M.Frigo, C.F. Joerg, C.E.Leiserson, K.H.Randall.
4. *Detecting Data Races in Cilk Programs that Use Locks*, G.I. Cheng, M.Feng, C.E.Leiserson, K.H. Randall, A.F.Stark.
5. *LCA Problem Revisited*, M.A.Bender, M.F.Colton.