BACKGROUND

Systematic (exploratory) testing
• Exhaustive state space search [1]
• Defined by “decision points”, exponential in size
• Reduction techniques make completion feasible
  • Dynamic Partial Order Reduction [2]
  • Dynamic Interface Reduction [3]

On-line estimation provides a “progress bar”.
• Resource allocation for scheduling many tests [4]
• Dynamic reduction interferes with estimation

TEST CASE REFINEMENT

Goal: Given a test and a CPU time budget, explore as many “meaningful” interleavings as possible.
• User studies in 15-410 suggest an “iterative deepening” approach [5].

Challenge: Automatically-inserted decision points should be useful, not random.
• Lock, semaphore, other scheduling operations
• Use dynamic data race detection to add decision points on suspicious memory accesses.

REFERENCES