

# What Test Should I Run?

## Improving Concurrency Tests with State Space Estimation

Ben Blum, Jiri Simsa, Garth Gibson (CMU)

### BACKGROUND

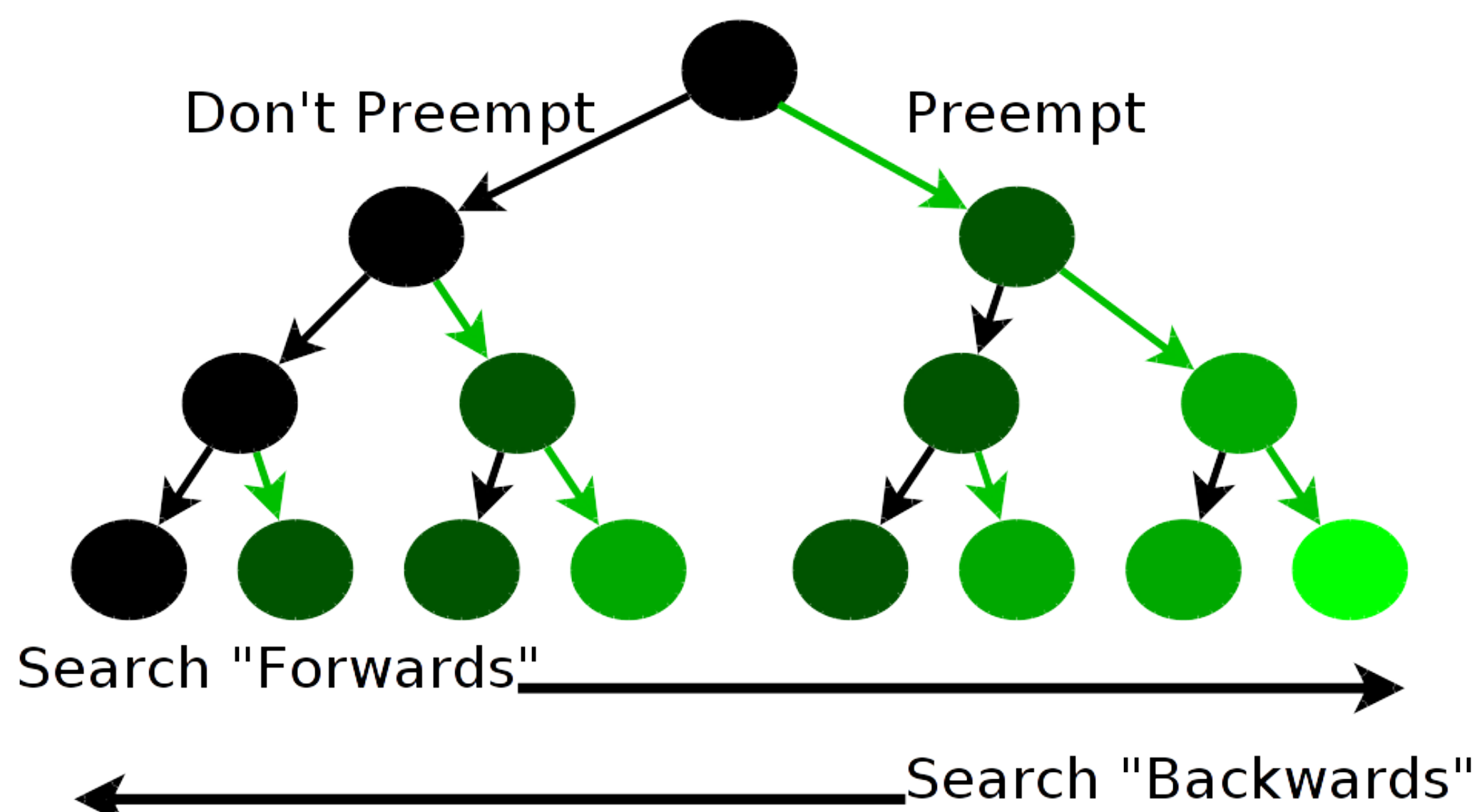
#### Systematic (exploratory) testing

- 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”.

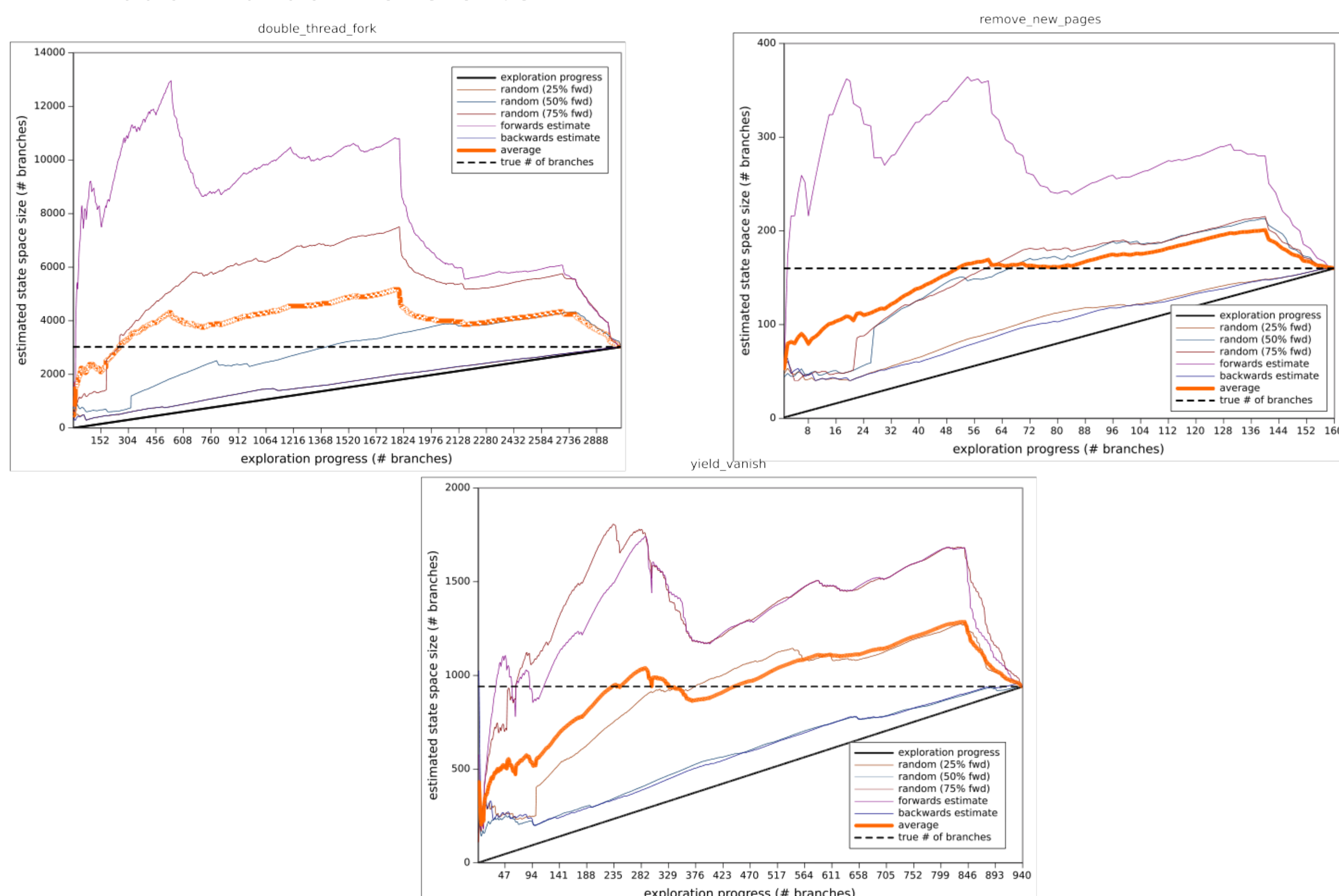
- Resource allocation for scheduling many tests [4]
- Dynamic reduction interferes with estimation

### IMPROVING ESTIMATION QUALITY



**Insight:** Sampling estimates from many parts of the tree at once improves estimation quality.

- Exploration ordering: “forwards” vs “backwards”
- Random explorations can be weighted to prefer forwards or backwards more often.



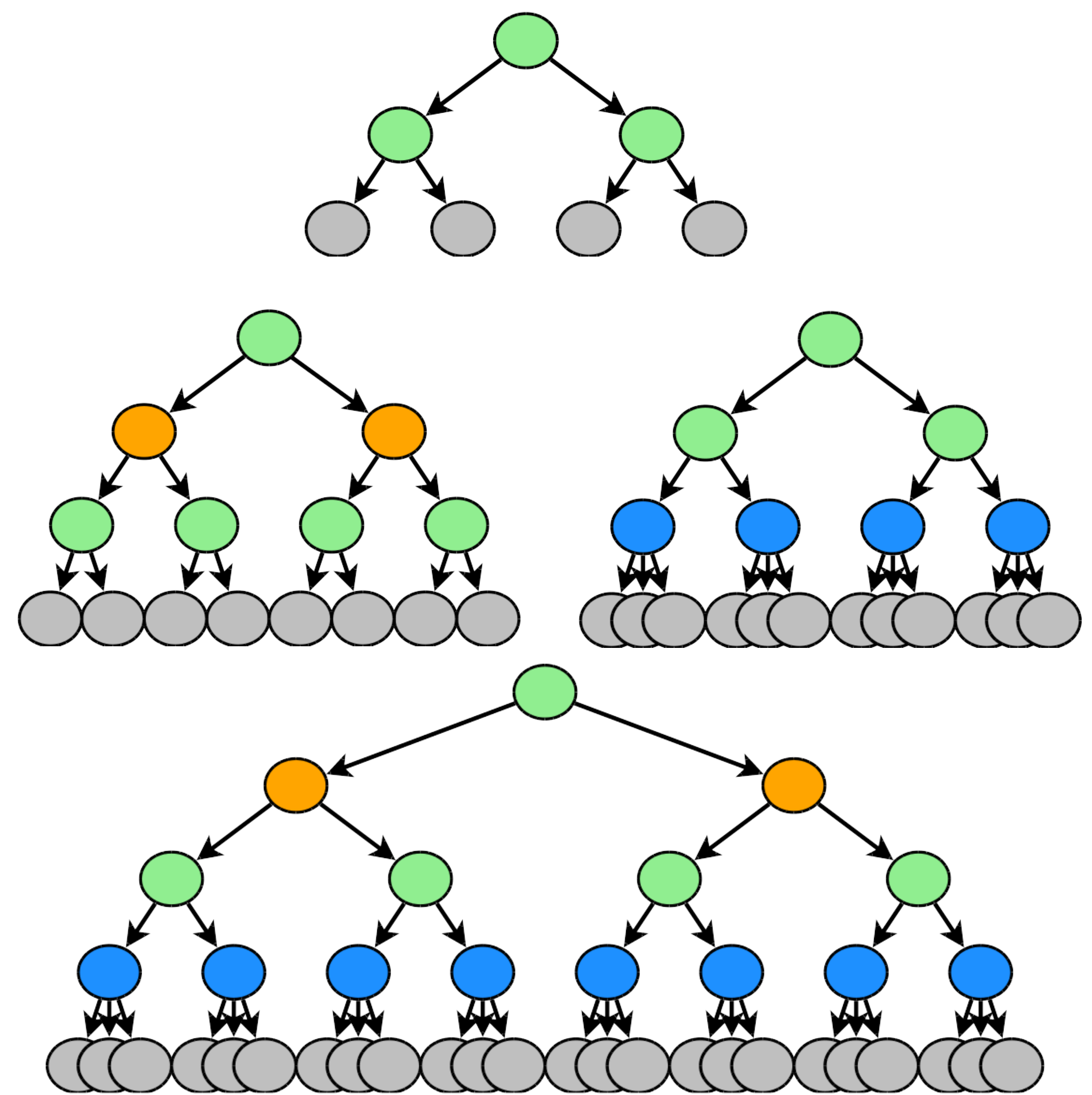
**Problem:** Taking more samples is more expensive.

- Need to make a decision after X CPU-seconds.
- Should we..
  - Spend more time on one part of the tree?
  - Spend more time collecting many samples?
- What is the right balance? (How do we measure?)

### 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

- [1] Patrice Godefroid. VeriSoft: A Tool for the Automatic Analysis of Concurrent Reactive Software. CAV 1997.
- [2] Cormac Flanagan and Patrice Godefroid. Dynamic Partial Order Reduction for Model Checking Software. POPL 2005.
- [3] Huayang Guo, Ming Wu, Lidong Zhou, Gang Hu, Junfeng Yang, and Lintao Zhang. Practical software model checking via dynamic interface reduction. SOSP '11.
- [4] Jiri Simsa. Runtime Estimation and Resource Allocation for Concurrency Testing. CMU-PDL-12-113. December 2012.
- [5] Ben Blum. Landslide: Systematic Dynamic Race Detection in Kernel Space. CMU-PDL-12-118. May 2012.