# FLAMBES: Evolving Fast Performance Models

Adam Crume†, Carlos Maltzahn†, Jason Cope‡, Sam Lang‡, Rob Ross‡, Phil Carns‡, Chris Carothers*, Ning Liu*, Curtis Janssen◇, John Bent✠, Stephan Eidenbenz✠, Meghan (Wingate) McClelland✠

†University of California, Santa Cruz, ‡Argonne National Laboratory, *Rensselaer Polytechnic Institute, ◇Sandia National Laboratories, ✠Los Alamos National Laboratory

Baskin Engineering UC SANTA CRUZ

## Problem

- Need to simulate large clusters and supercomputers
- Simulating hard drives and other devices is too slow

## Goal

- Faster simulation methods that trade a little accuracy for a large gain in performance

## Our solution

- FLAMBES: Fitting scaLable Analytic Models Before Executing Simulation
- Focus on aggregate accuracy rather than request-level accuracy
- Use genetic programming based on analytic models

## Aggregate versus individual accuracy

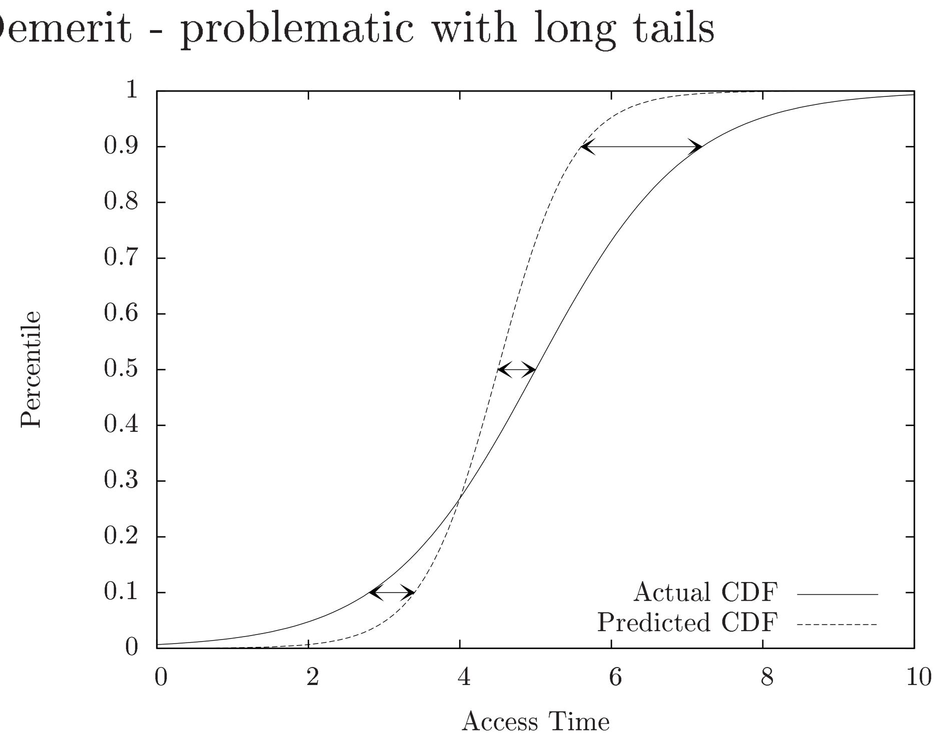
- Individual request times may be irrelevant
- Individual accuracy requires knowledge of future requests
- Ideally, always have aggregate accuracy, and have individual accuracy as much as possible

## Why genetic programming?

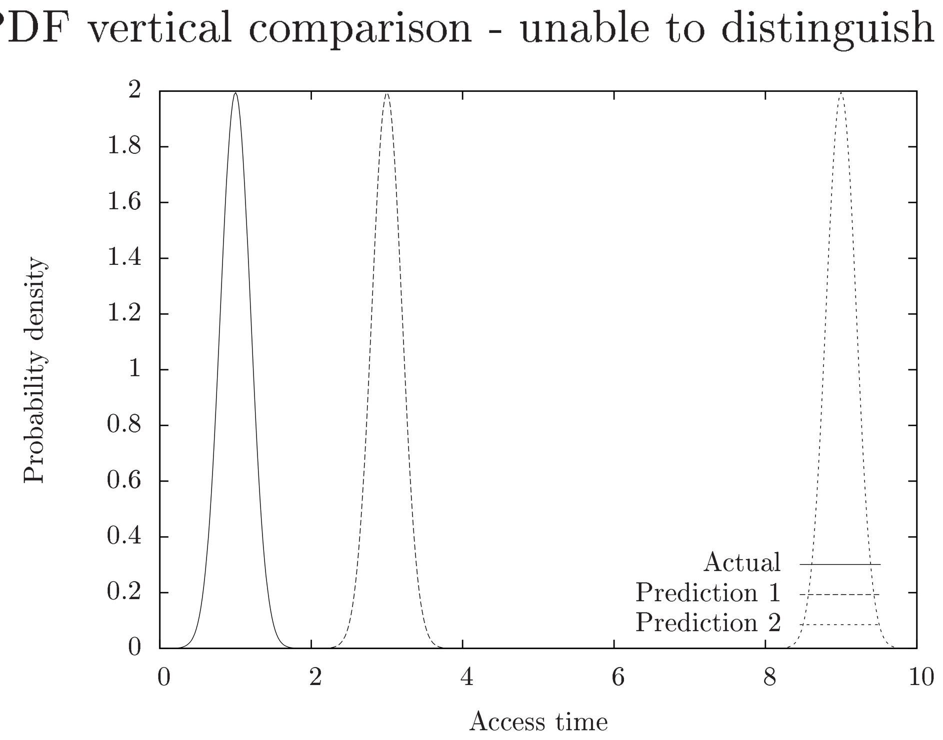
- Designing a model requires detailed expert knowledge
- Regression difficult due to high feature count
  - Large history; state depends on many previous inputs
  - Behavior is complex; many higher-order terms would be necessary
- Neural nets poorly suited for stateful problems

## Aggregate error metric

- Demerit - problematic with long tails




- PDF vertical comparison - unable to distinguish predictions 1 and 2 below

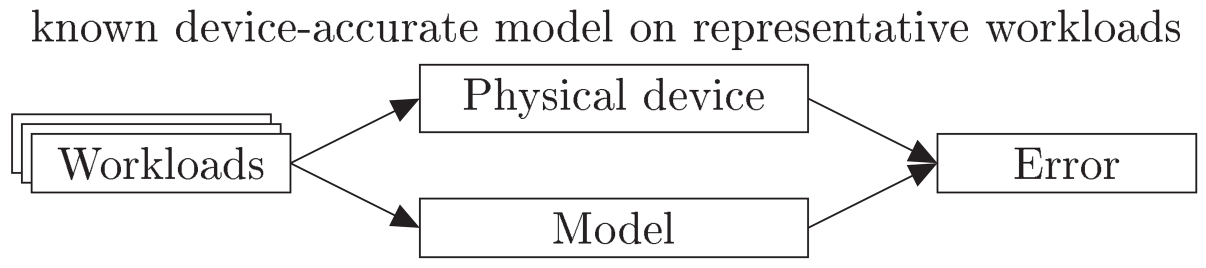



## Individual error metric

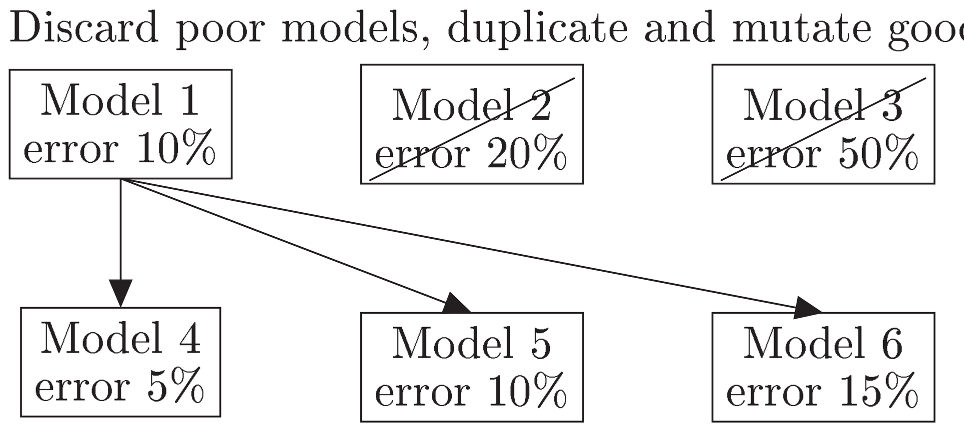
- $\frac{1}{n}\sum\frac{|actual_i - predicted_i|}{actual_i}$
  - Penalizes overestimation more than underestimation
- $\frac{1}{n}\sum\left|\log\frac{predicted_i}{actual_i}\right| = \frac{1}{n}\sum\log\frac{\max(actual_i, predicted_i)}{\min(actual_i, predicted_i)}$
  - Symmetric
  - Allows very large ratios
- $\frac{1}{n}\sum\exp\left(\left|\log\frac{predicted_i}{actual_i}\right|\right) = \frac{1}{n}\sum\frac{\max(actual_i, predicted_i)}{\min(actual_i, predicted_i)}$
  - Best so far

## Step 1: Fit the model

- Offline calculation done once per device template
- Independent of simulation size
- Fitting is fast: typically a few minutes
- Final model is accurate for workloads represented by the training set
- Algorithm:
  1. Initialize population by generating several models with random parameters
  2. Evaluate each model by comparing its predictions with device performance or a known device-accurate model on representative workloads




  3. Discard poor models, duplicate and mutate good models




  4. Repeat until error is low

## Step 2: Use the model

- Very fast calculation (not event-driven)
- Very low state: a few floating point numbers

## Result

- Fast, accurate simulation of large distributed systems

## Related work

- DiskSim[1] is request-level accurate, but slow
- Sharkawi in [2] used GAs to match applications to similar benchmarks. FLAMBES uses genetic programming to directly predict performance.

## References

[1] John S. Bucy, Jiri Schindler, Steven W. Schlosser, Gregory R. Ganger, and Contributors. *The DiskSim Simulation Environment Version 4.0 Reference Manual*. Carnegie Mellon University, Pittsburgh, PA, May 2008.

[2] Sameh Sharkawi, Don DeSota, Raj Panda, Rajeev Indukuru, Stephen Stevens, Valerie Taylor, and Xingfu Wu. Performance projection of hpc applications using spec cfp2006 benchmarks. In *Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*, pages 1–12, Washington, DC, USA, 2009. IEEE Computer Society.

## Funding

This work was supported by Department of Energy grant DE-SC0005428 and the Los Alamos National Laboratory/University of California, Santa Cruz Institute for Scalable Scientific Data Management (ISSDM).