High Performance Computing Systems (CMSC714)

Lecture 24: Machine Learning for HPC

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

Summary of last lecture

- Discrete-event simulations (DES)
- Parallel DES: conservative vs. optimistic
- Simulation of epidemic diffusion: agent-based, time-stepped modeling
- Trace-driven network simulations: model event sequences

Why machine learning?

- Proliferation of performance data
  - On-node hardware counters
  - Switch/network port counters
  - Power measurements
  - Traces and profiles
- Supercomputing facilites' data
  - Job queue logs, performance
  - Sensors: temperature, humidity, power

Types of ML-related tasks in HPC

- Auto-tuning: parameter search
  - Find a well performing configuration
- Predictive models: time, energy, …
  - Predict system state in the future
  - Time-series analysis
- Identifying root causes/factors

Understanding network congestion

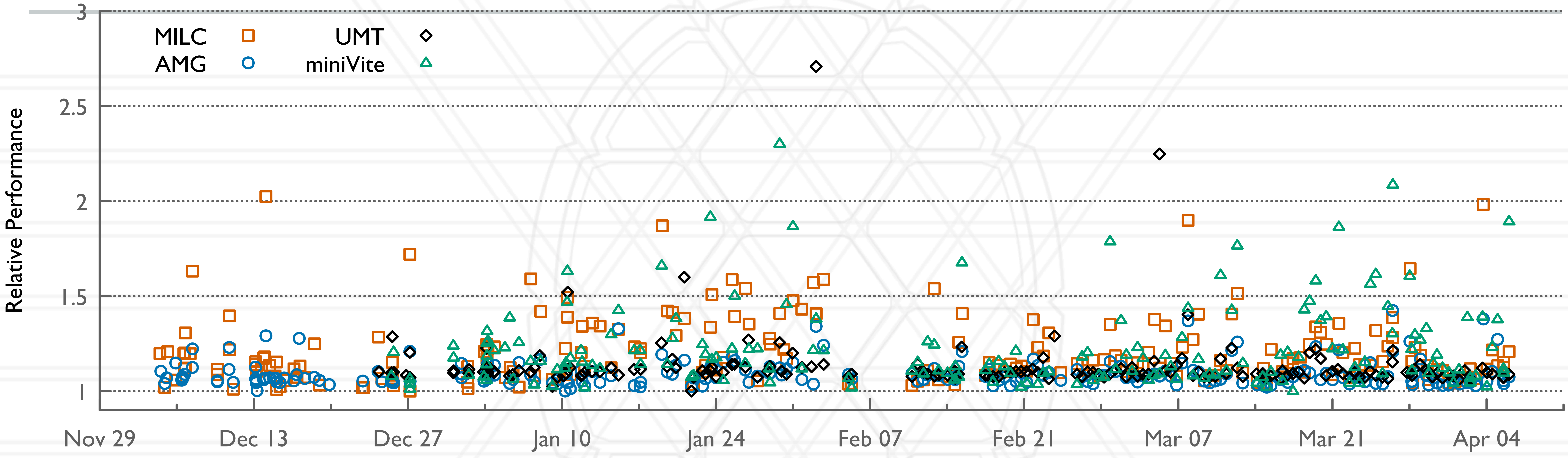
- Congestion and its root causes not well understood
- Study network hardware performance counters and their correlation with execution time
- Use supervised learning to identify hardware components that lead to congestion and performance degradation

Understanding network congestion

- Congestion and its root causes not well understood
- Study network hardware performance counters and their correlation with execution time
- Use supervised learning to identify hardware components that lead to congestion and performance degradation

Hardware resource	Contention indicator
Source node	Injection FIFO length
Network link	Number of sent packets
Intermediate router	Receive buffer length
All	Number of hops (dilation)

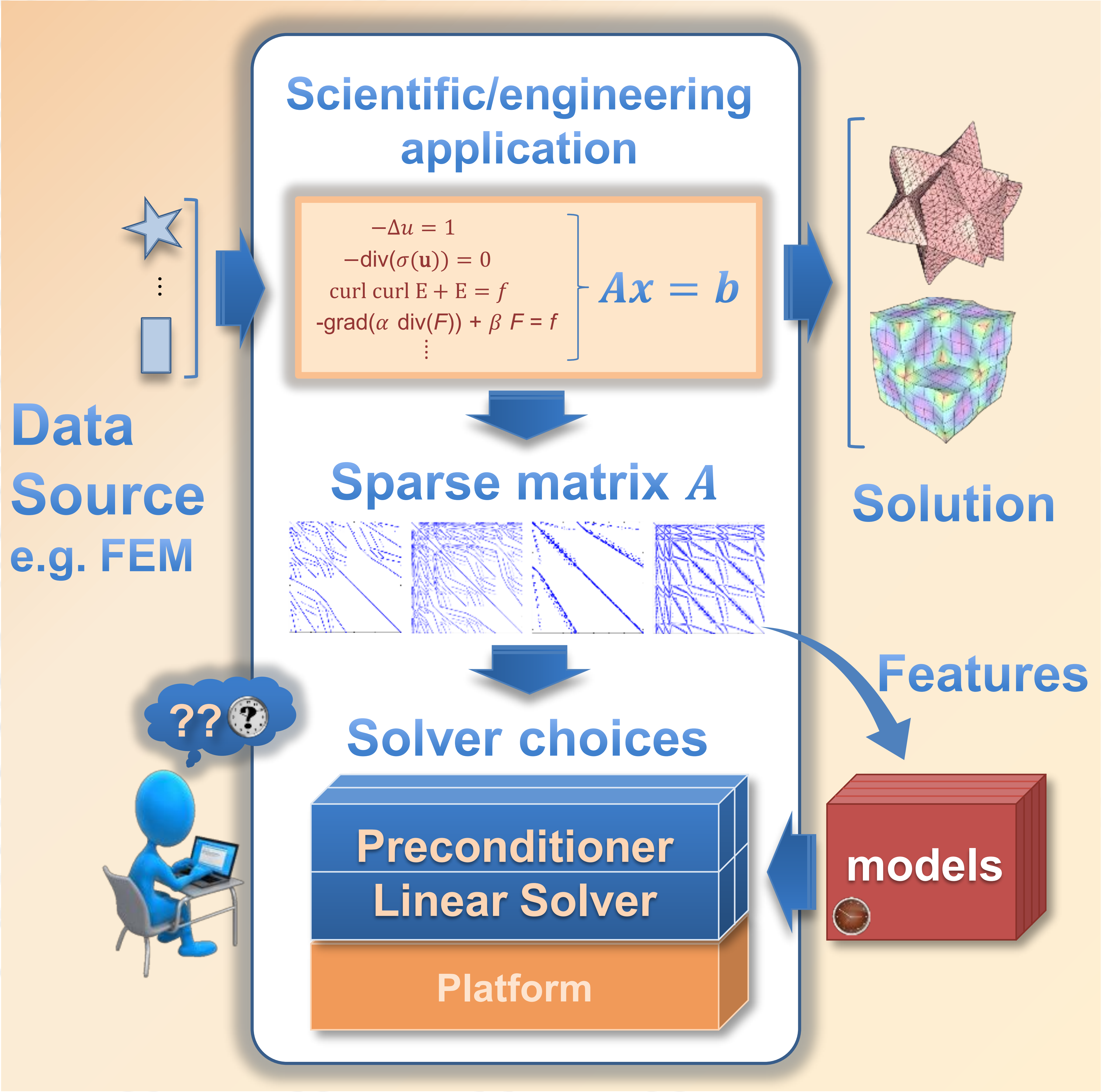
Investigating performance variability



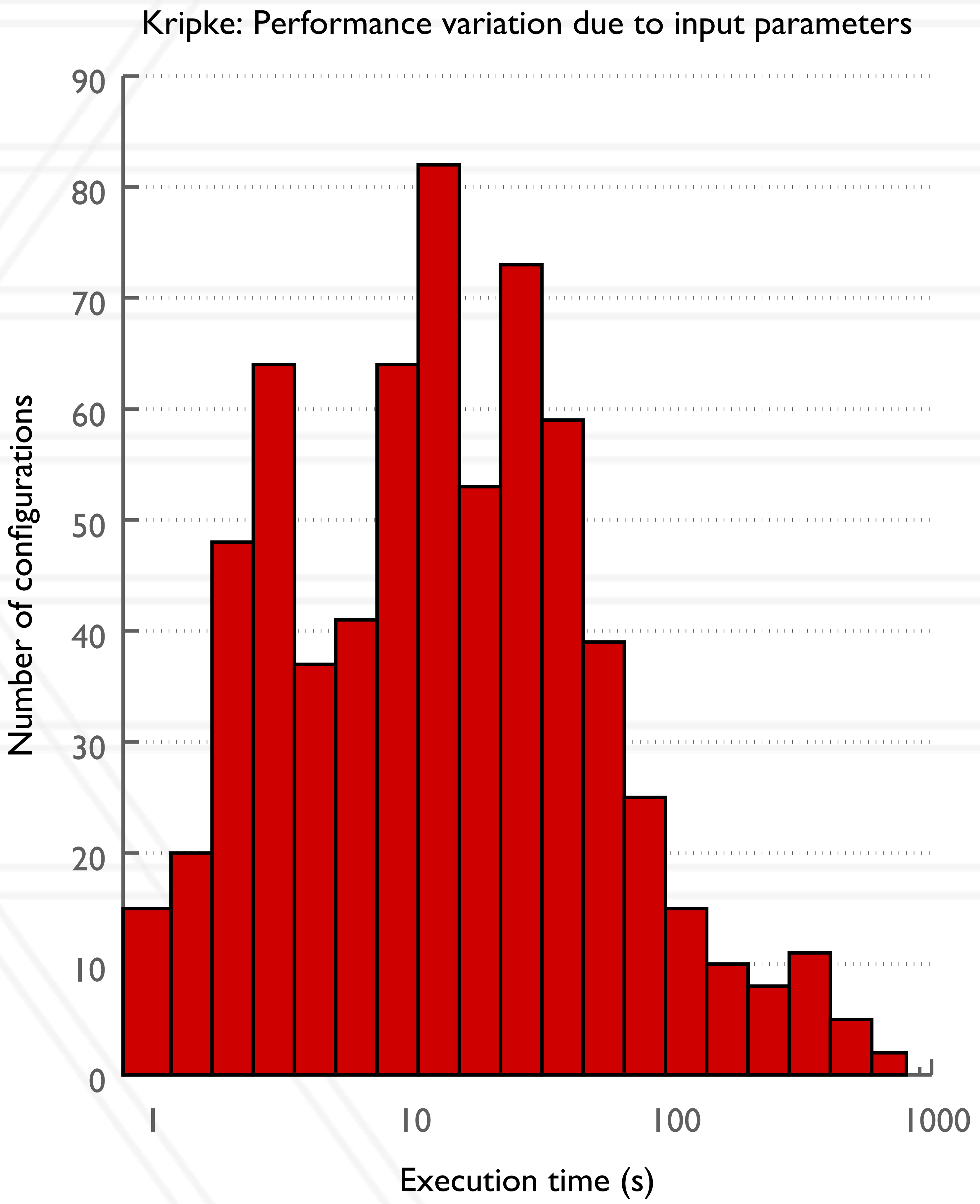
- Identify users to blame, important network counters
- Predict future performance based on historical time-series data

Identifying best performing code variants

- Many computational science and engineering (CSE) codes rely on solving sparse linear systems
- Many choices of numerical methods
- Optimal choice w.r.t. performance depends on several things:
  - Input data and its representation, algorithm and its implementation, hardware architecture

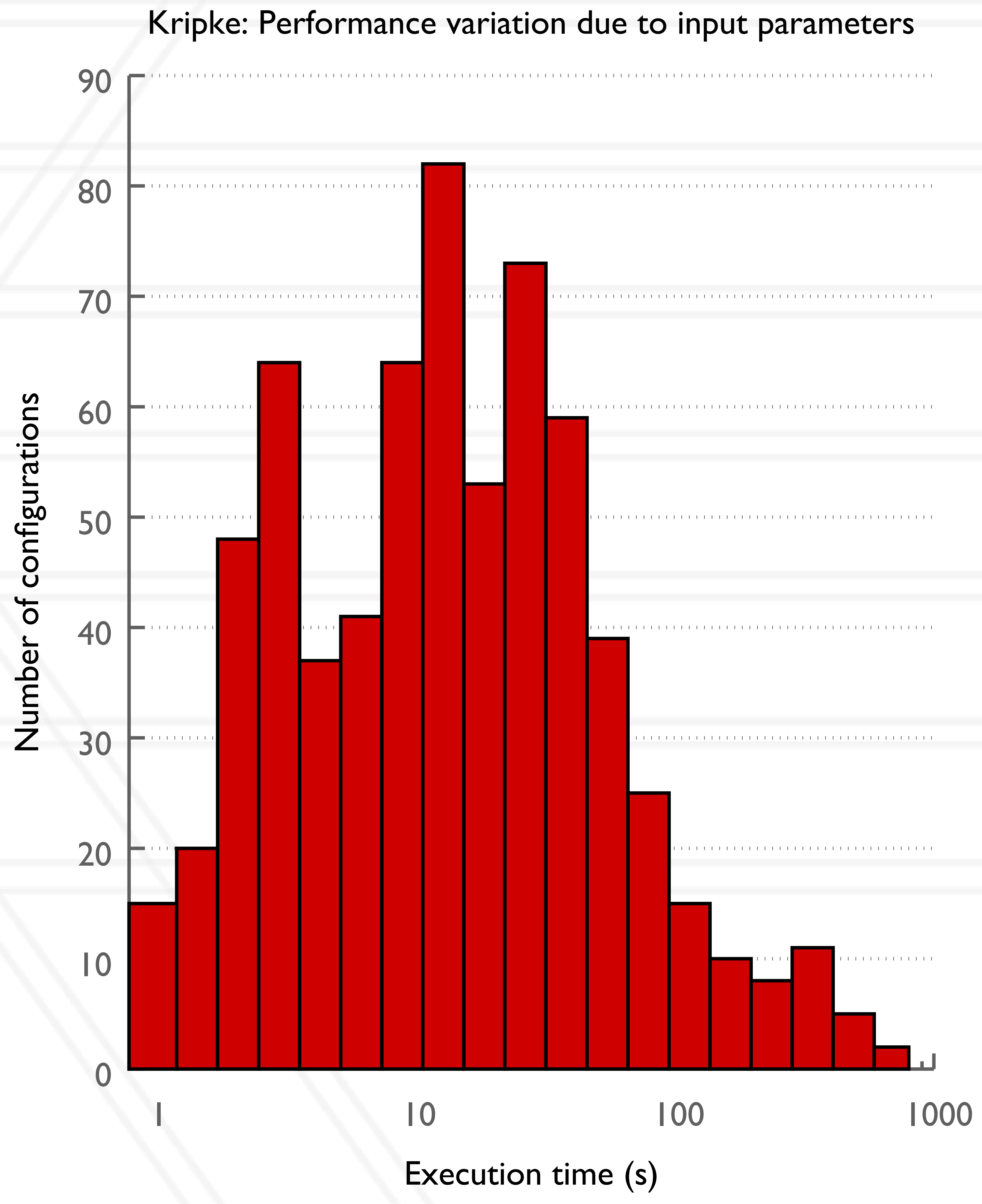


Auto-tuning with limited training data



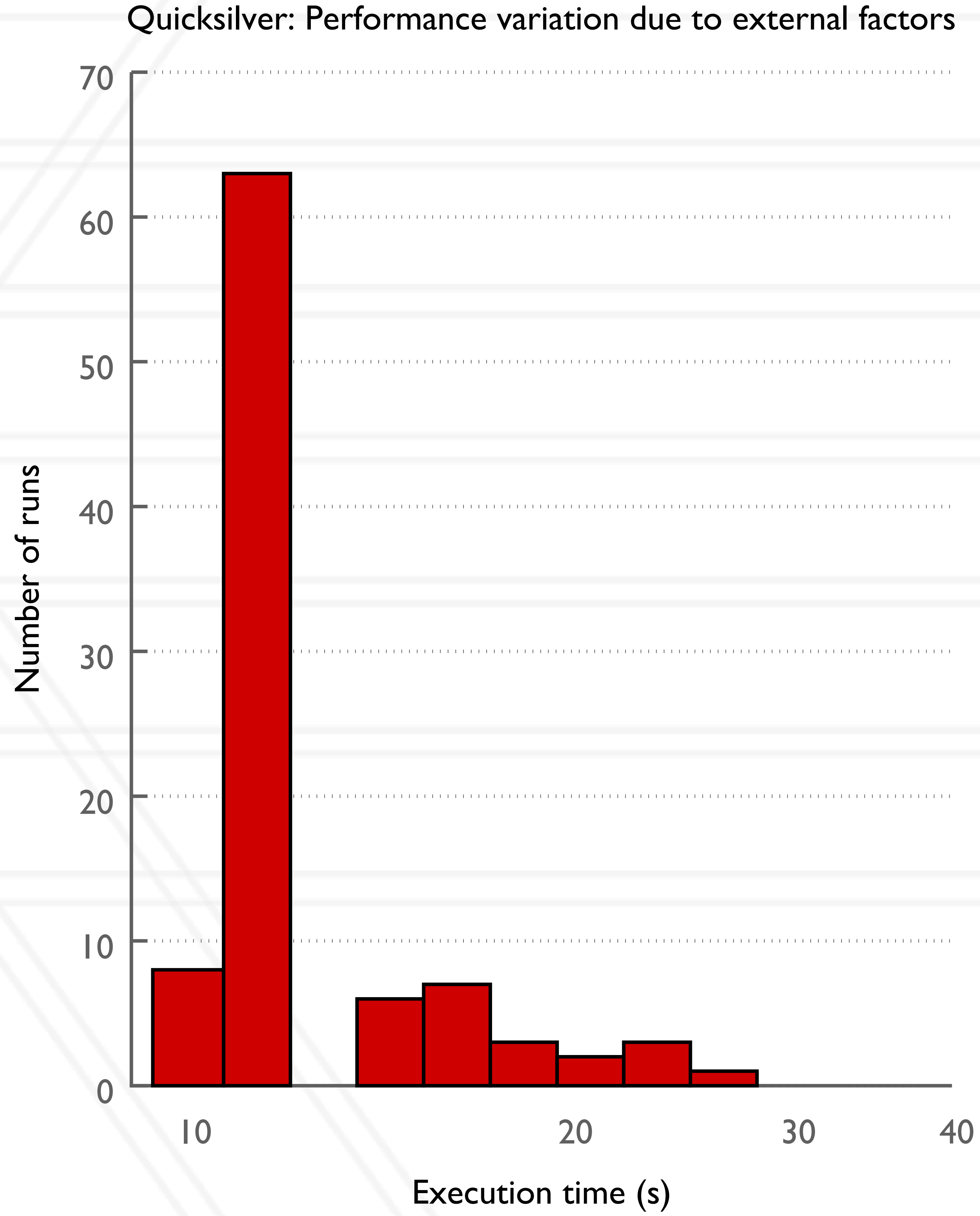
Auto-tuning with limited training data

- Application performance depends on many factors:
  - Input parameters, algorithmic choices, runtime parameters



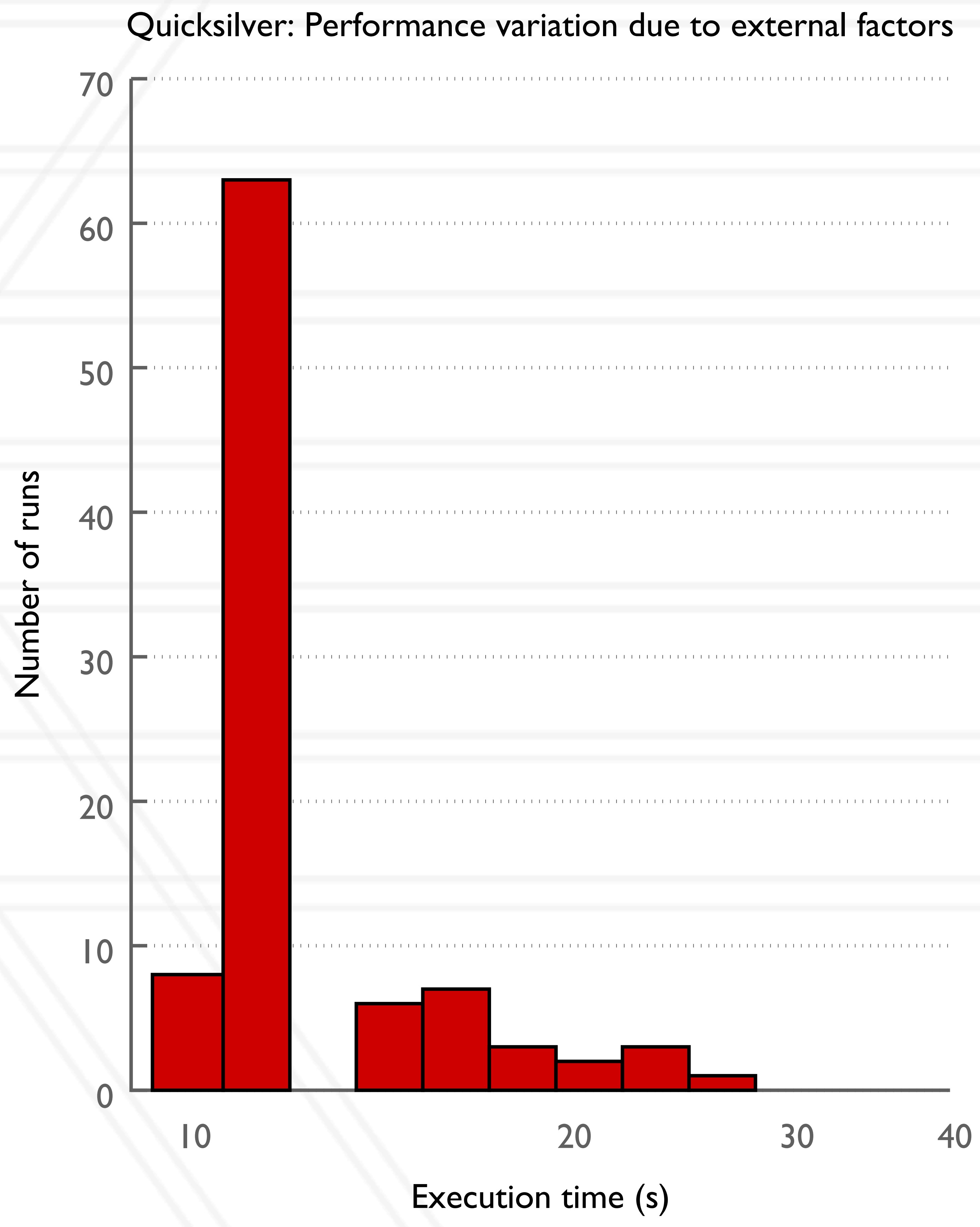
Auto-tuning with limited training data

- Application performance depends on many factors:
  - Input parameters, algorithmic choices, runtime parameters
- Performance also depends on:
  - Code changes, linked libraries
  - Compilers, architecture



Auto-tuning with limited training data

- Application performance depends on many factors:
  - Input parameters, algorithmic choices, runtime parameters
- Performance also depends on:
  - Code changes, linked libraries
  - Compilers, architecture
- Surrogate models + transfer learning



Questions

Identifying the Culprits behind Network Congestion

- Can you go over the differences between weak and strong models in ML?
- How is OS noise accounted for in these runs or is Blue Gene/Q truly noiseless?
- Why do you use R^2 and RCC?
- Was the goal of the paper to identify parameters that are important or generate accurate predictions? Seems like the parameters identified at the end were pretty reasonable to expect to be important in the first place.
- Why do the authors choose 5D torus network in particular?

Questions

Identifying the Culprits behind Network Congestion

- What about using deep learning techniques to do the nonparametric regression to predict the relation, since a lot of data samples have been collected?
- Why is the Huber loss preferred over L1 or L2 loss in this task? How is the value of delta selected?
- Why is it necessary to perform an exhaustive search over all possible feature combinations? If some features are useless, wouldn't they be automatically ignored by the machine learning models?
- Regarding the zero R^2 scores, what does it mean by an "artifact" of scaling? If we scale the features in the same way for both the training and the testing set, why is there a problem? How does standardization differ from scaling? If standardization is better, why wasn't it used in this paper?
- What are the problems or limitations of selecting features according to the rank correlation between every single feature and the execution time?

Questions

Bootstrapping Parameter Space Exploration for Fast Tuning

- The mapping of the parameter space to an undirected graph was a little confusing, could you go over it?
- How does the label propagation routine choose 'prior beliefs'? How many labelled nodes are required for the results to converge?
- If the models are pre run, how can you be sure to sample the configuration space (page 6, second column) properly? Does GEIST require an exhaustive search of the space?
- If the hyperparameters are set by the initial random sample, is it possible to start with an anomalous sample that reduces performance dramatically (especially with the 'hard optimization problems' where there a few optimal solutions)
- What dictates the choice of the number of iterations to perform in GEIST?

Questions

Bootstrapping Parameter Space Exploration for Fast Tuning

- How could we determine b_{ik}, which denotes the prior belief on associating node i with label k?
- How is the stability of this GEIST Algorithm? Since we predict the labels and do the sampling based on previous iterations, leading to propagation of error.
- Can the autotuning problem be modeled as a regression task?
- How do we represent configurations as vectors so that the set of nearest neighbors can be determined by computing L1 distances?
- Is the prior belief term b_{ik} used in the GEIST algorithm? Although GEIST does not require prior knowledge, can we further reduce the number of samples to collect by incorporating expert knowledge through this term?
- How difficult is it to find suitable hyperparameters? Can this be time-consuming, because we might have to run GEIST multiple times with different hyperparameter settings?

Questions?

UNIVERSITY OF MARYLAND

Abhinav Bhatele

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu