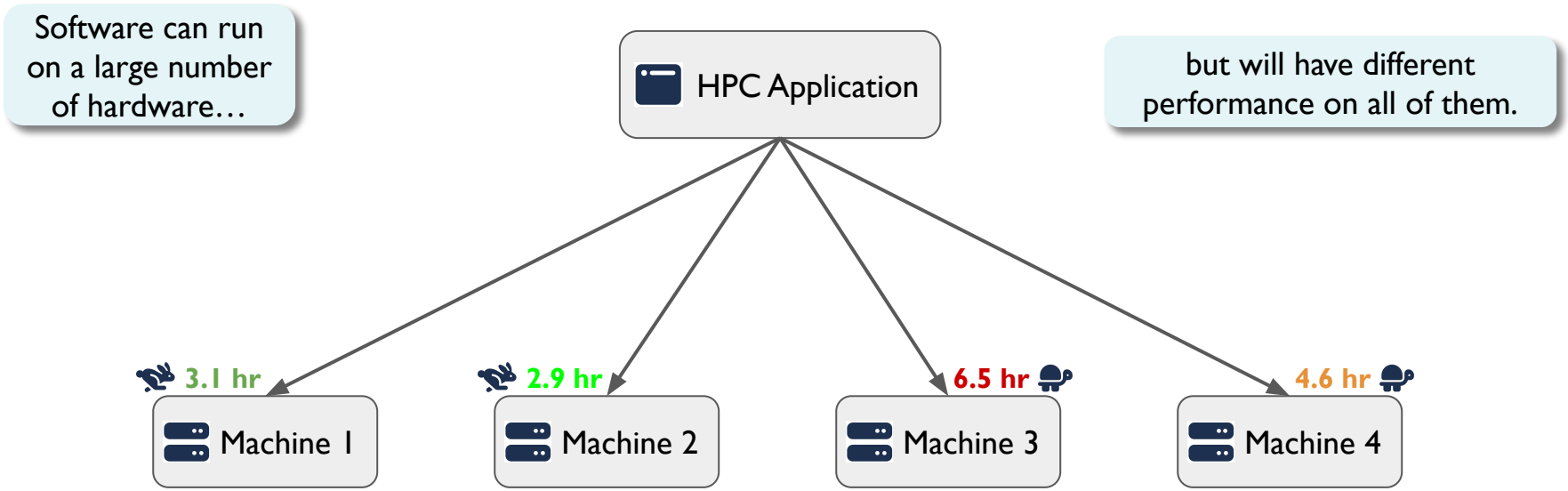
# Predicting Cross-Architecture Performance of Parallel Programs

Daniel Nichols*, Alexander Movsesyan*, Jae-seung Yeom†, Abhik Sarkar†, Daniel Milroy†, Tapasya Patki†, Abhinav Bhatele*

* University of Maryland

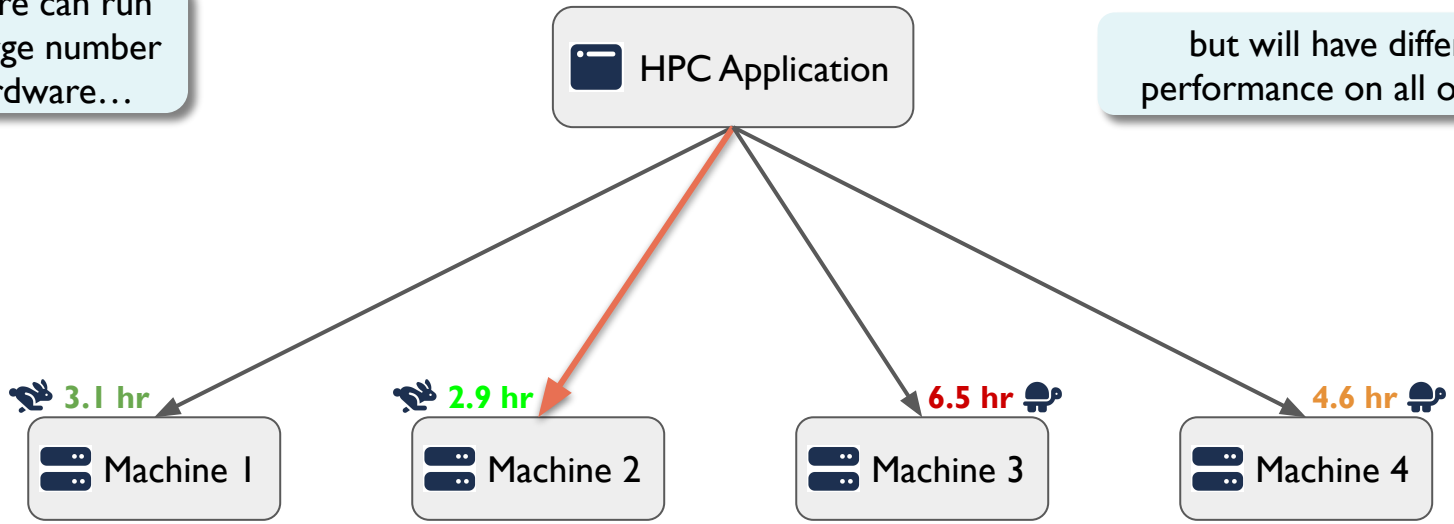
† Lawrence Livermore National Laboratory


# Motivation: Running Applications on the Fastest Hardware



# Motivation: Running Applications on the Fastest Hardware

Software can run on a large number of hardware…




but will have different performance on all of them.

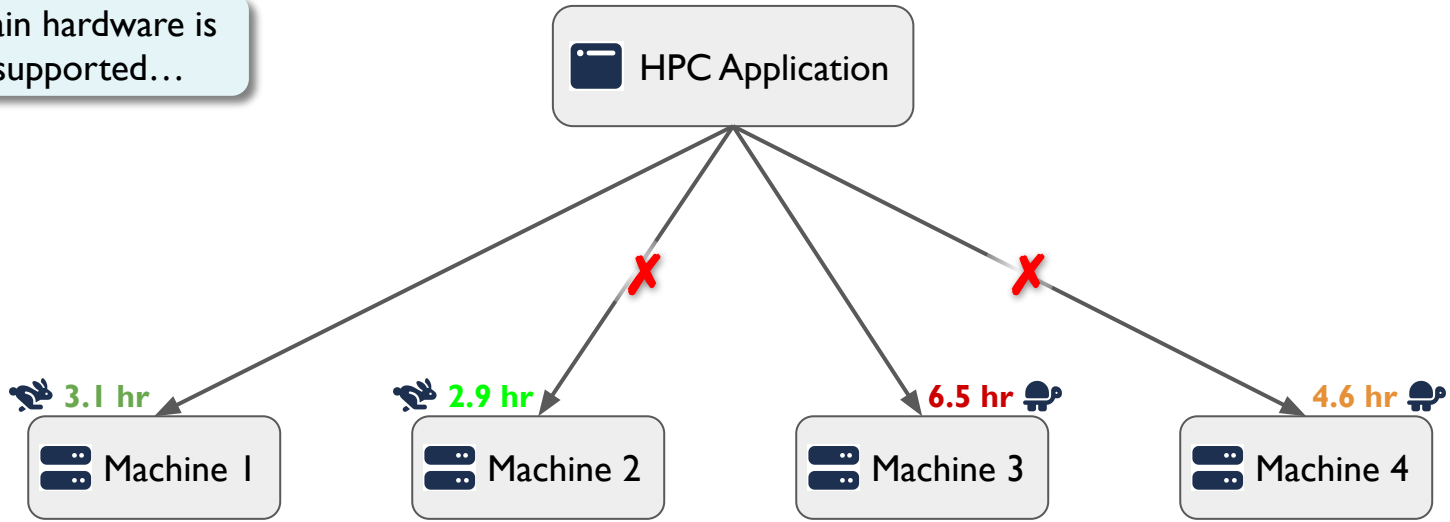
Users or a scheduler can pick the best system if they know what hardware will be fastest.

PSSG

UNIVERSITY OF MARYLAND

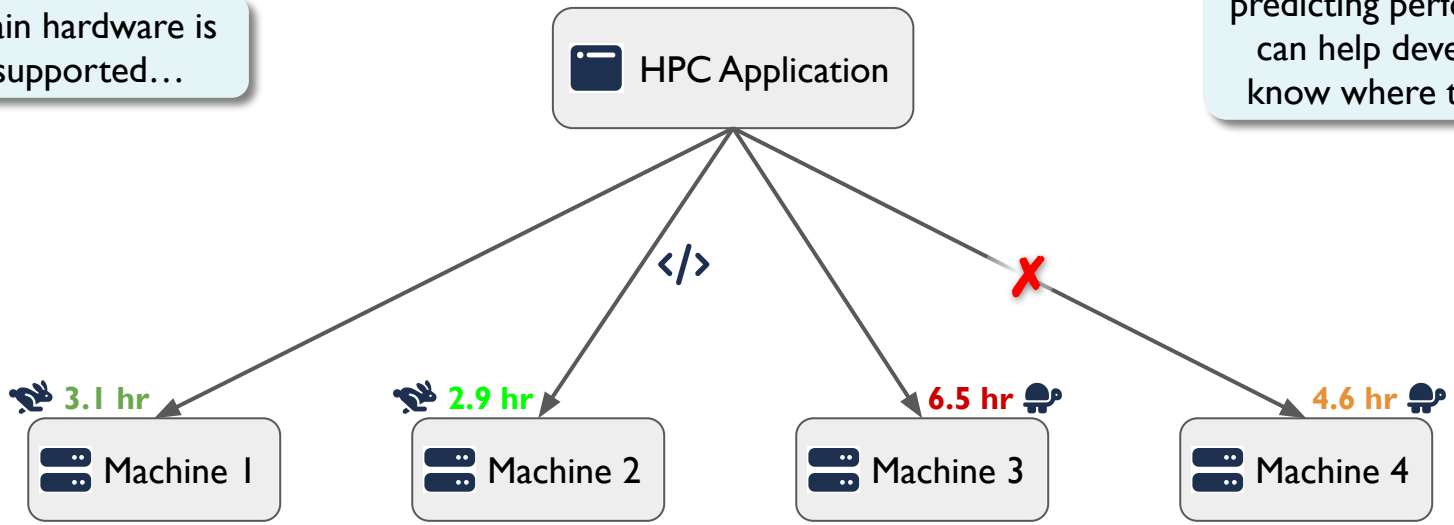
# Motivation: Running Applications on the Fastest Hardware

If certain hardware is not supported…



# Motivation: Running Applications on the Fastest Hardware

If certain hardware is not supported…




predicting performance can help developers know where to port.

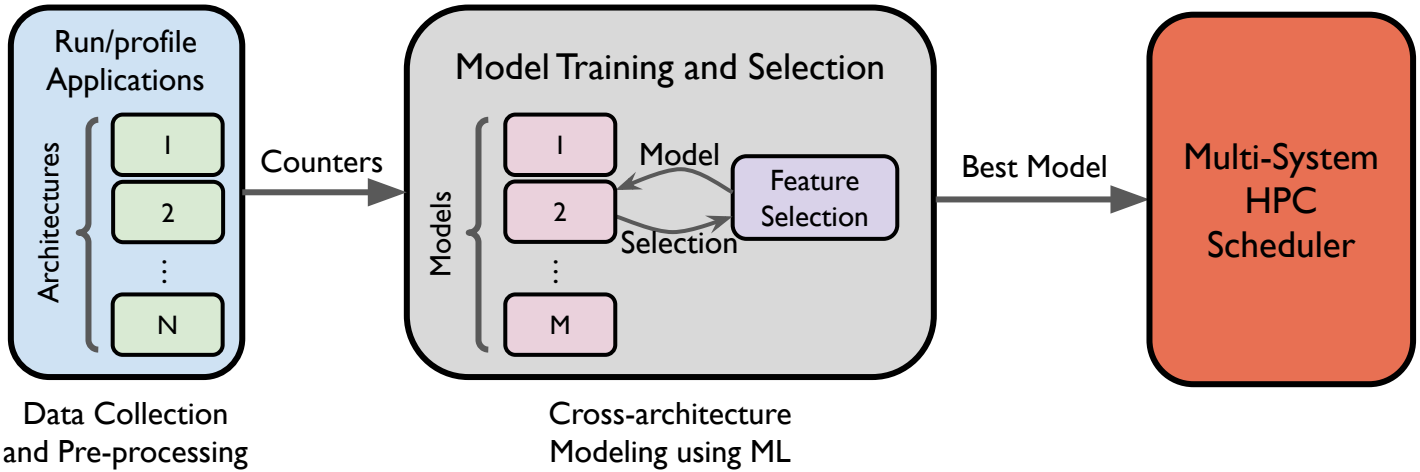
# Research Questions

**RQ1** – *How can we accurately predict cross architecture performance of HPC applications for multiple architectures at once?*

**RQ2** – *What features and data sources contribute most to predicting cross architecture performance?*

**RQ3** – *How can we use cross architecture performance predictions to schedule jobs across HPC systems more effectively?*

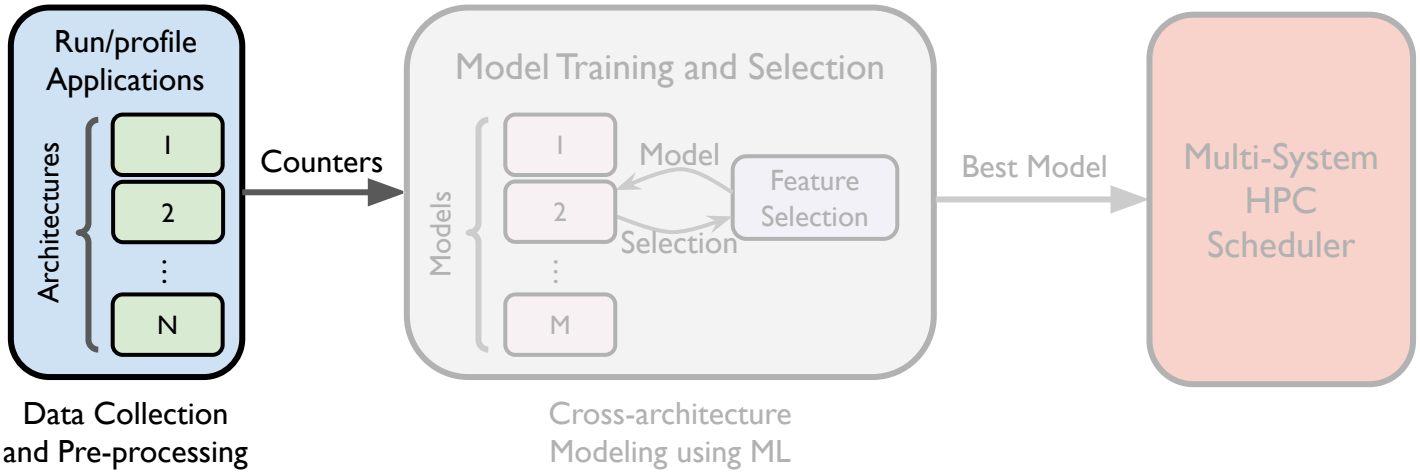
# Overview of our Approach




PSSG

UNIVERSITY OF MARYLAND

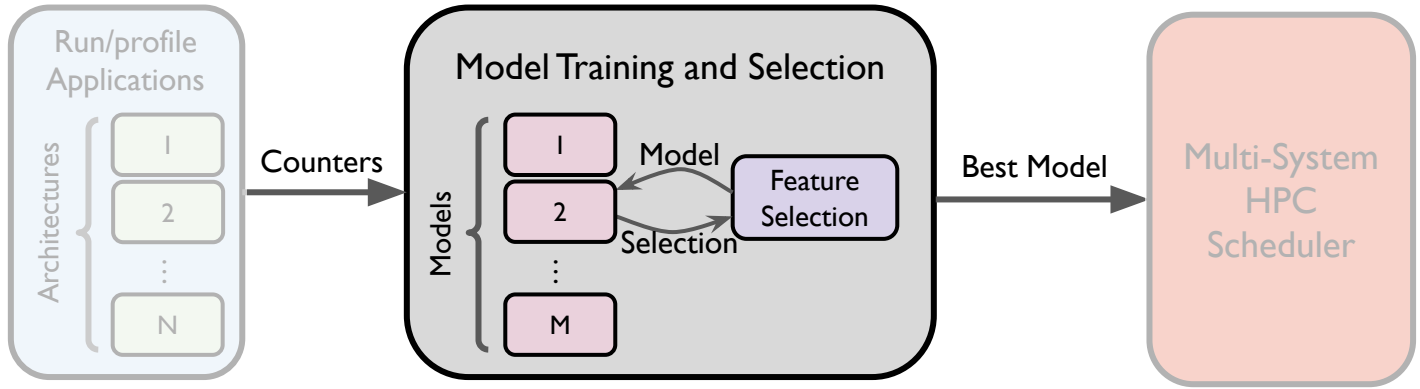
# Overview of our Approach




PSSG

UNIVERSITY OF MARYLAND

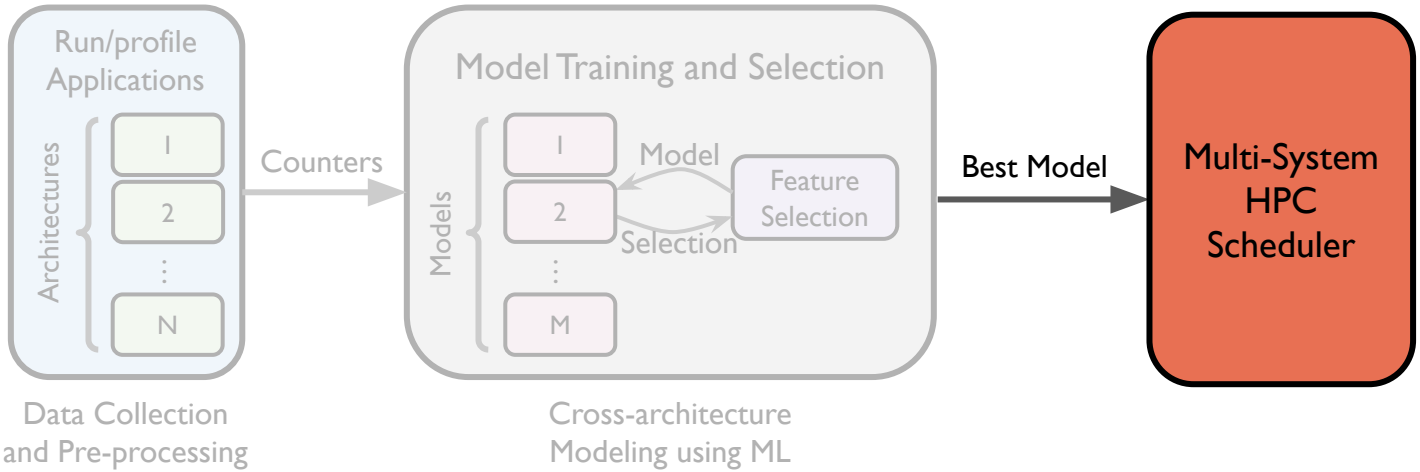
# Overview of our Approach




PSSG

UNIVERSITY OF MARYLAND

# Overview of our Approach




PSSG

UNIVERSITY OF MARYLAND

# Data Collection: Overview

- Ran 20 scientific applications on 4 systems at LLNL
  - Applications from ECP proxy app suite and E4S
  - Systems (2 CPU; 2 GPU)
    - Quartz – Intel Xeon E5-2695
    - Ruby – Intel Xeon CLX-8276
    - Lassen – Power9 CPUs and V100 GPUs
    - Corona – AMD Rome CPUs and MI50 GPUs
  - Ran on many inputs to gather more data points
- Collected counter data from each run
- Final dataset has ~11k samples

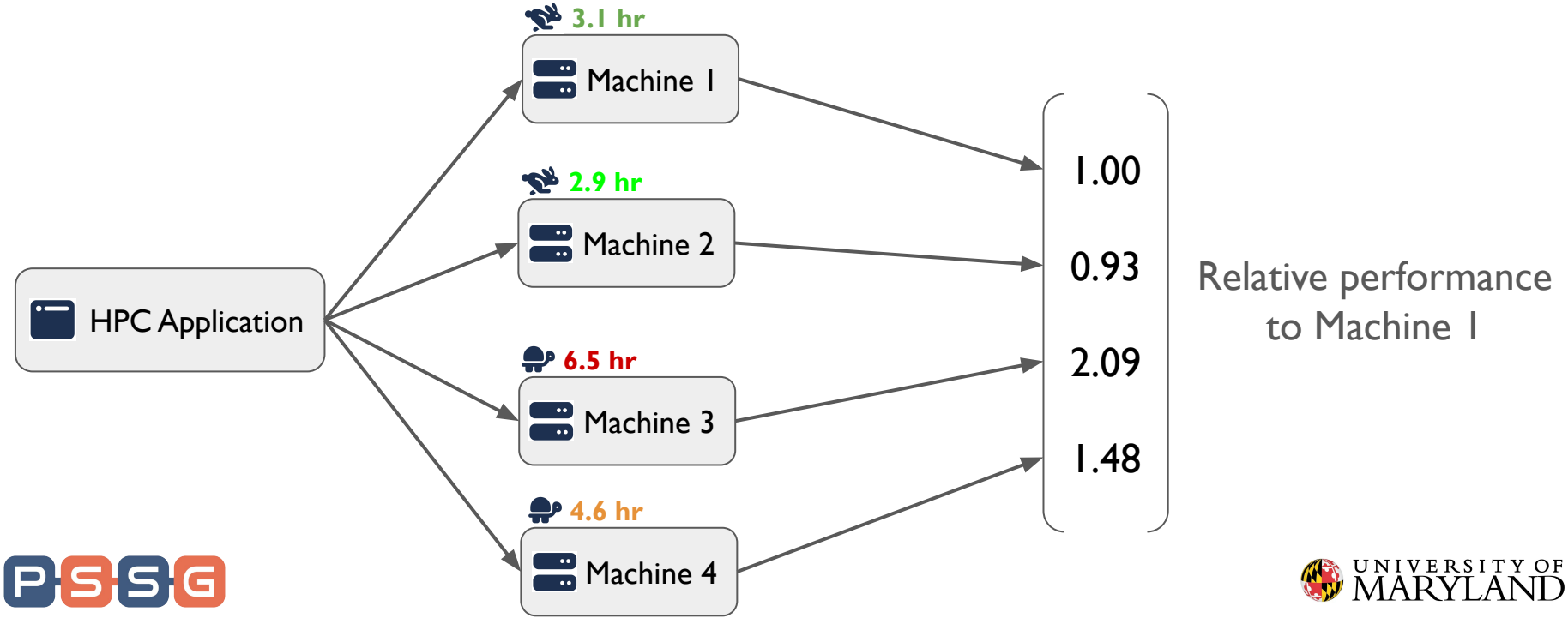
# Data Collection: Counters

- Counters chosen to cover 4 areas
  - Data locality and memory
  - Control flow & parallelism
  - IO
  - Run configuration
- Best counters for each feature on each architecture were selected

| Feature | Description |
|---|---|
| Branch Intensity | Ratio of branch instructions to total instructions |
| Store Intensity | Ratio of store instructions to total instructions |
| Load Intensity | Ratio of load instructions to total instructions |
| Single FP Intensity | Ratio of single precision FP instructions to total instructions |
| Double FP Intensity | Ratio of double precision FP instructions to total instructions |
| Arithmetic Intensity | Ratio of integer arithmetic instructions to total instructions |
| L1 Load Misses | L1 cache load misses |
| L1 Store Misses | L1 cache store misses |
| L2 Load Misses | L2 cache load misses |
| L2 Store Misses | L2 cache store misses |
| IO Bytes Written | Bytes written to IO |
| IO Bytes Read | Bytes read from IO |
| Extended Page Table | Extended page table size |
| Memory Stalls | Memory stalls |
| Nodes | Number of nodes |
| Cores | Number of cores |
| Uses GPU | 1 if counters from GPU; 0 otherwise |
| Architecture | one-hot-encoded vector for what architecture these counters were recorded on |

# Learning Objective: What do we want to predict?

- Direct runtime prediction is difficult.



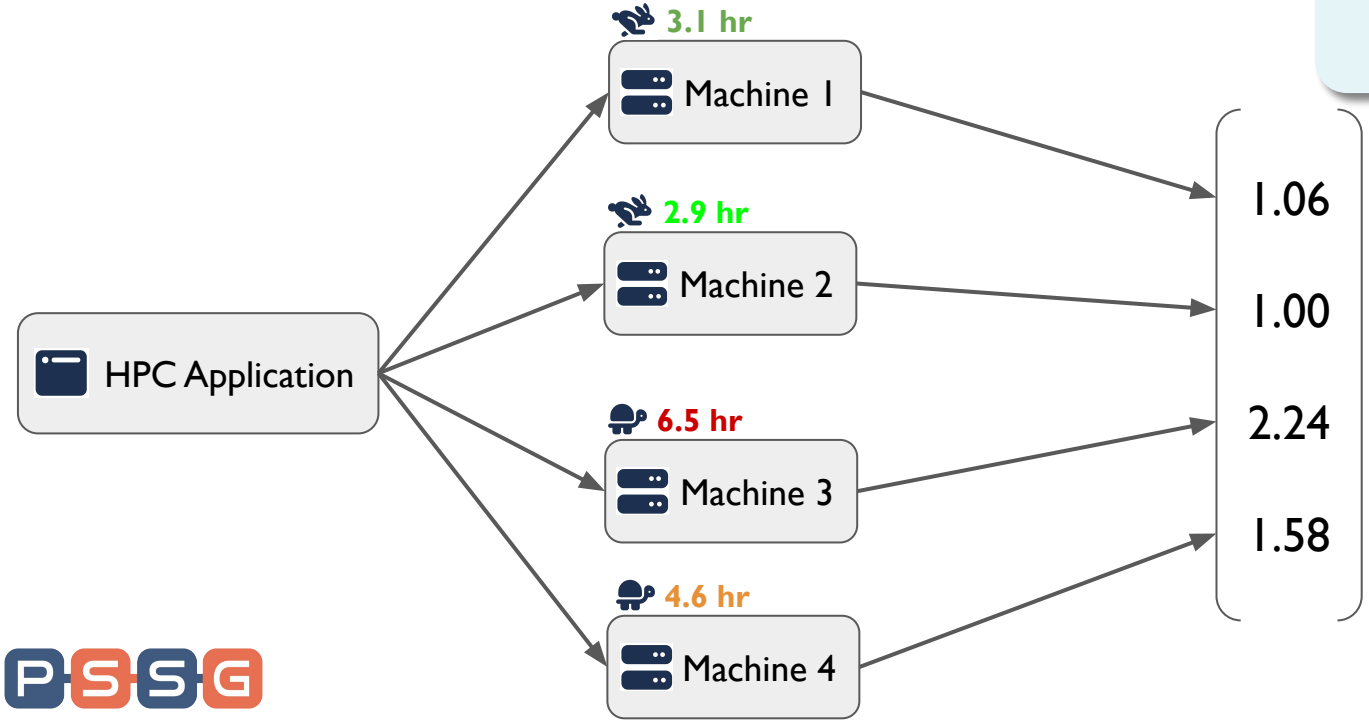

PSSG

UNIVERSITY OF MARYLAND

# Learning Objective: What do we want to predict?

- Direct runtime prediction is difficult.

*Relative performance vectors* provide a good target metric for predicting performance across architectures.




PSSG

UNIVERSITY OF MARYLAND

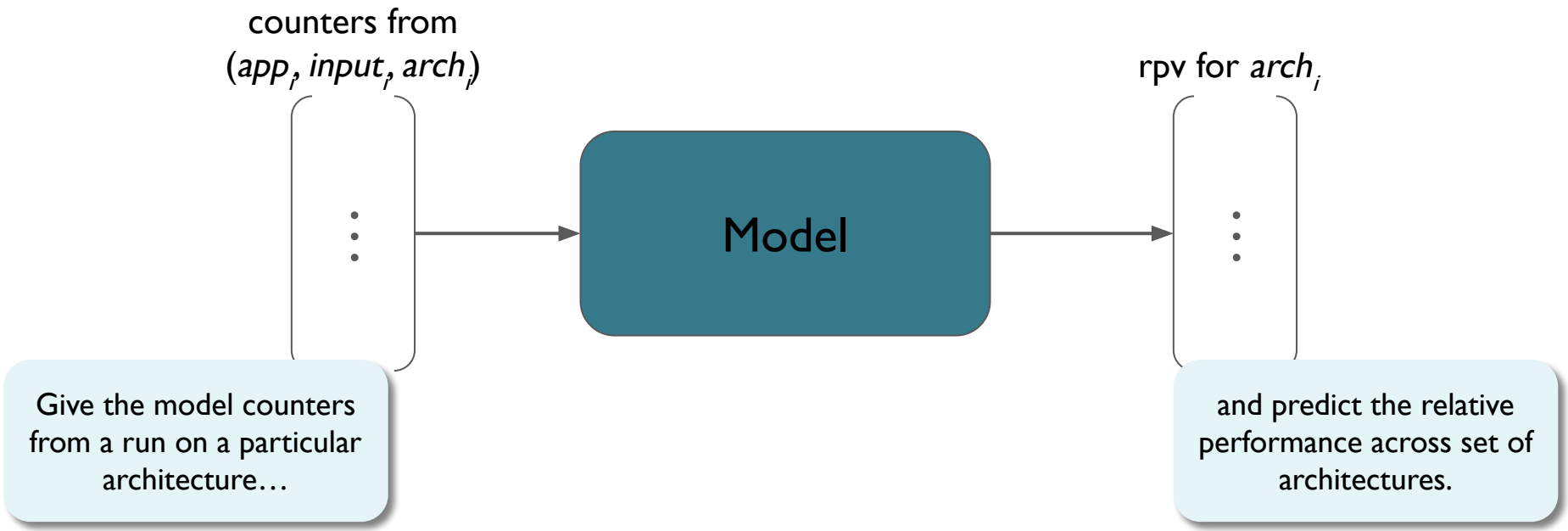
# Models & Training

- Train four models on dataset
  - Mean predictor – baseline
  - Linear regression
  - Decision forest
  - XGBoost
- 90-10 train-test split with 5-fold cross validation
- Predict *relative performance vector* for 4 machines given counters from 1
- Record two metrics
  - Mean absolute error (MAE)
  - Same order score (SOS)

PSSG

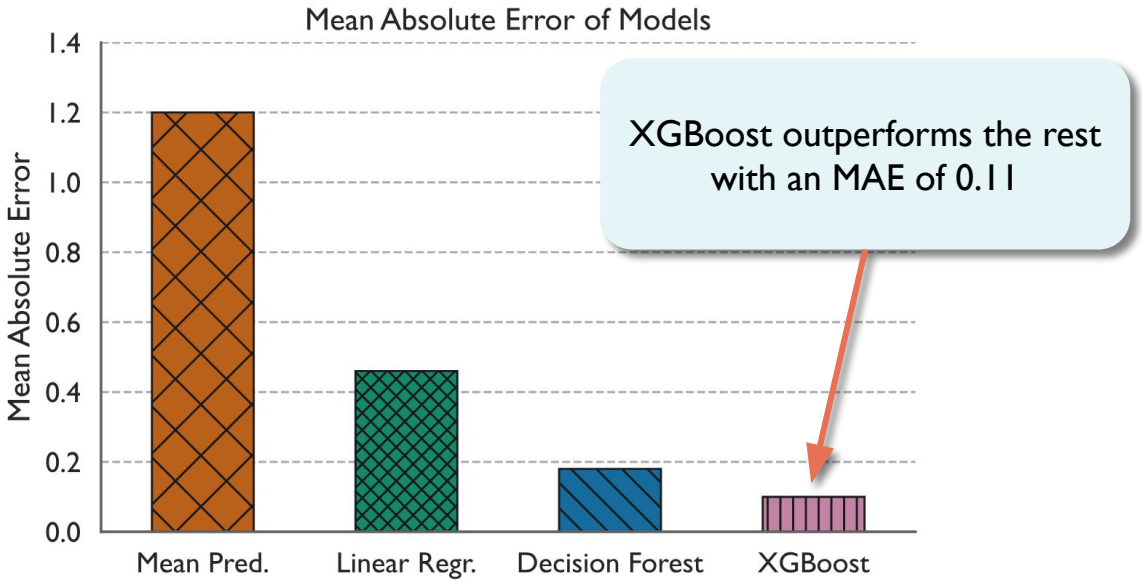
UNIVERSITY OF MARYLAND

# Model Inputs and Outputs



# Training Results

**RQ1** – *How can we accurately predict cross architecture performance of HPC applications for multiple architectures at once?*

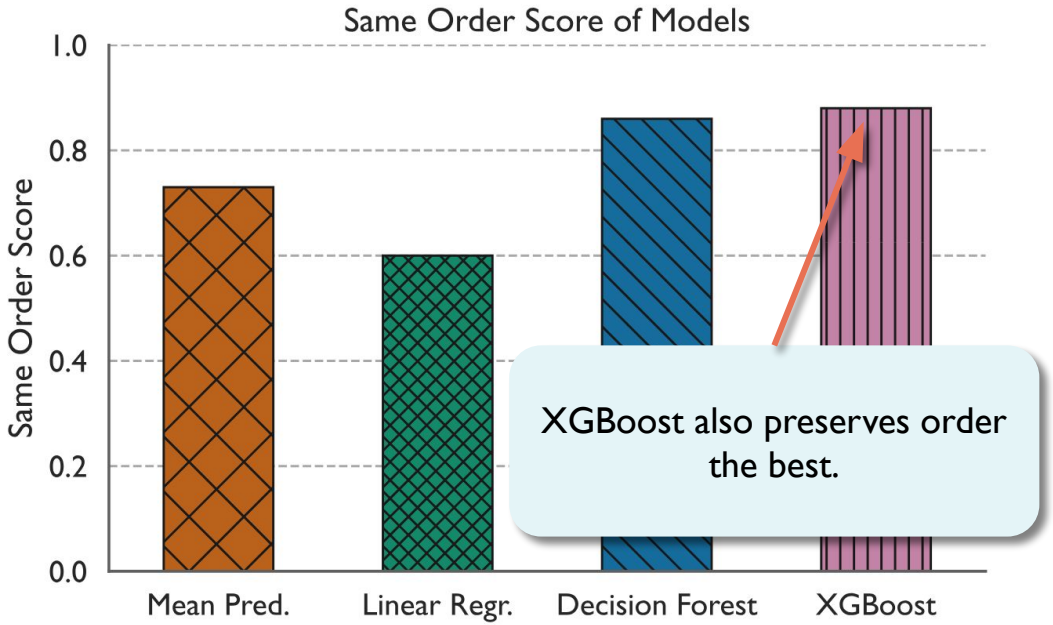



PSSG

UNIVERSITY OF MARYLAND

# Training Results

**RQ1** – *How can we accurately predict cross architecture performance of HPC applications for multiple architectures at once?*



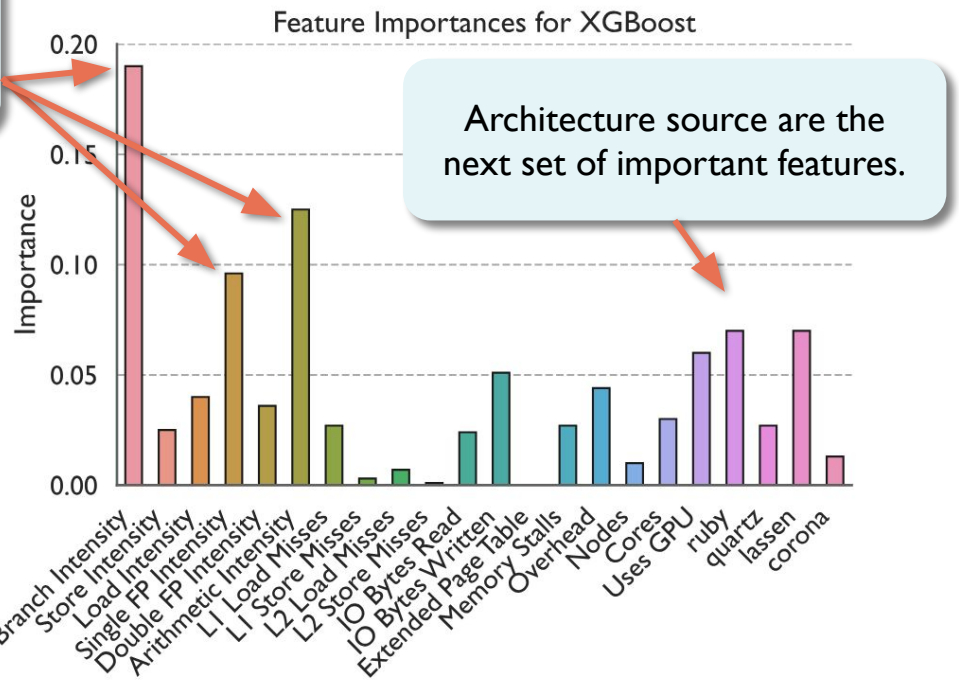

PSSG

UNIVERSITY OF MARYLAND

# Feature Importance

Branch, arithmetic, and FP intensity are most important features

**RQ2** – *What features and data sources contribute most to predicting cross architecture performance?*



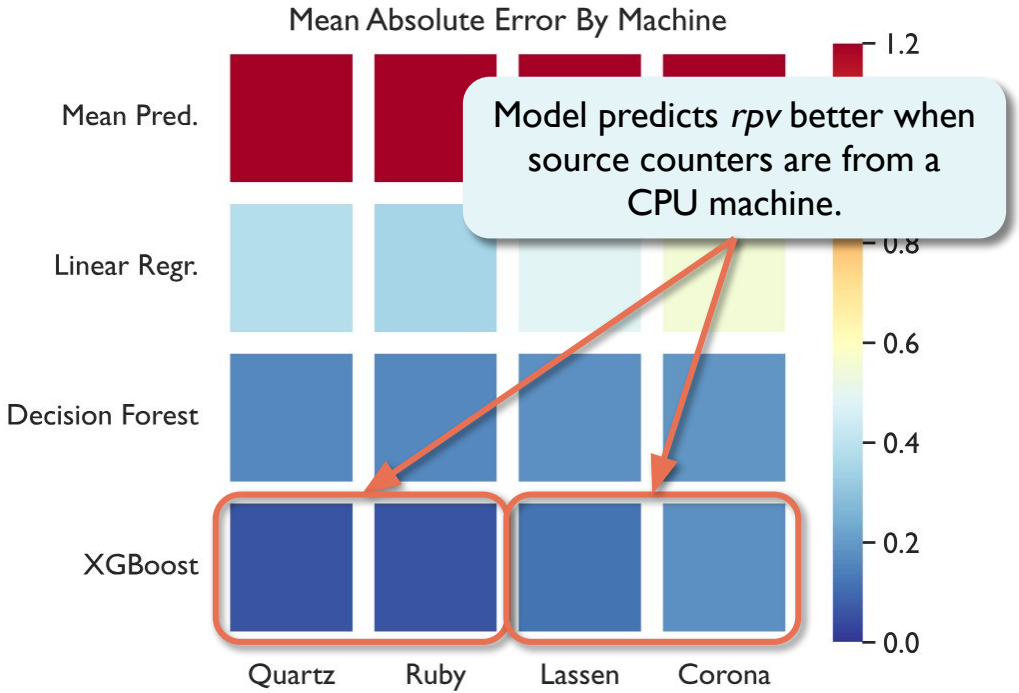

Architecture source are the next set of important features.

PSSG

UNIVERSITY OF MARYLAND

# Ablation Study: Source Architecture

**RQ2** – *What features and data sources contribute most to predicting cross architecture performance?*



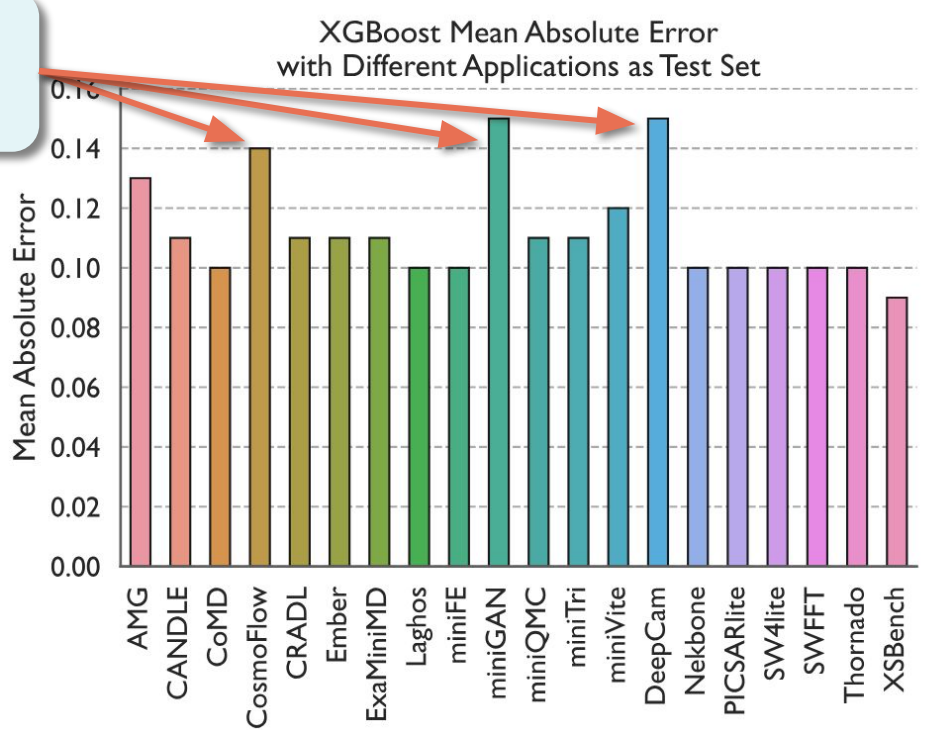

PSSG

UNIVERSITY OF MARYLAND

# Ablation Study: Source Application

Prediction is worse with Python ML applications

**RQ2** – *What features and data sources contribute most to predicting cross architecture performance?*




PSSG

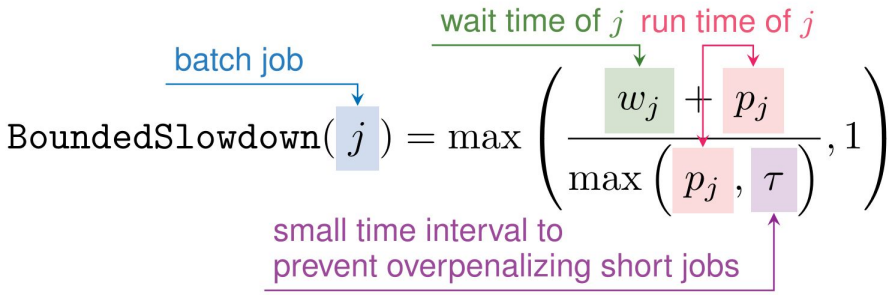
UNIVERSITY OF MARYLAND

# Incorporating the Model Into A Scheduler

- Use FCFS with EASY backfilling across four clusters
- When job is reserved use *policy* to decide the cluster
- Experiment with four policies
  - Round-Robin – rotate clusters
  - Random – randomly select cluster (uniformly)
  - User+Round-Robin – prioritize GPU clusters for GPU-enabled apps, then round-robin
  - Model-based – use model to predict *rpv*, then run on fastest cluster with available space

# Experiments and Metrics

- Create job traces from test dataset
- Simulated FCFS+EASY scheduler with each policy
- Record two metrics
  - Makespan – end-to-end time for job trace
  - Average bounded slowdown – slowdown is per-job metric to show how each job is affected by scheduling policy

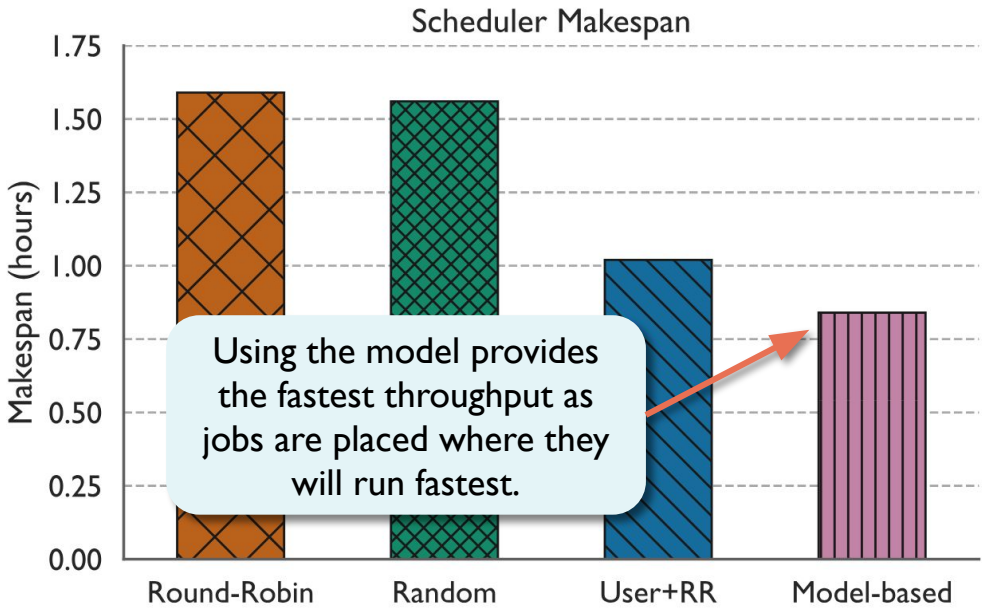



PSSG

UNIVERSITY OF MARYLAND

# Scheduling Results

**RQ3** – *How can we use cross architecture performance predictions to schedule jobs across HPC systems more effectively?*

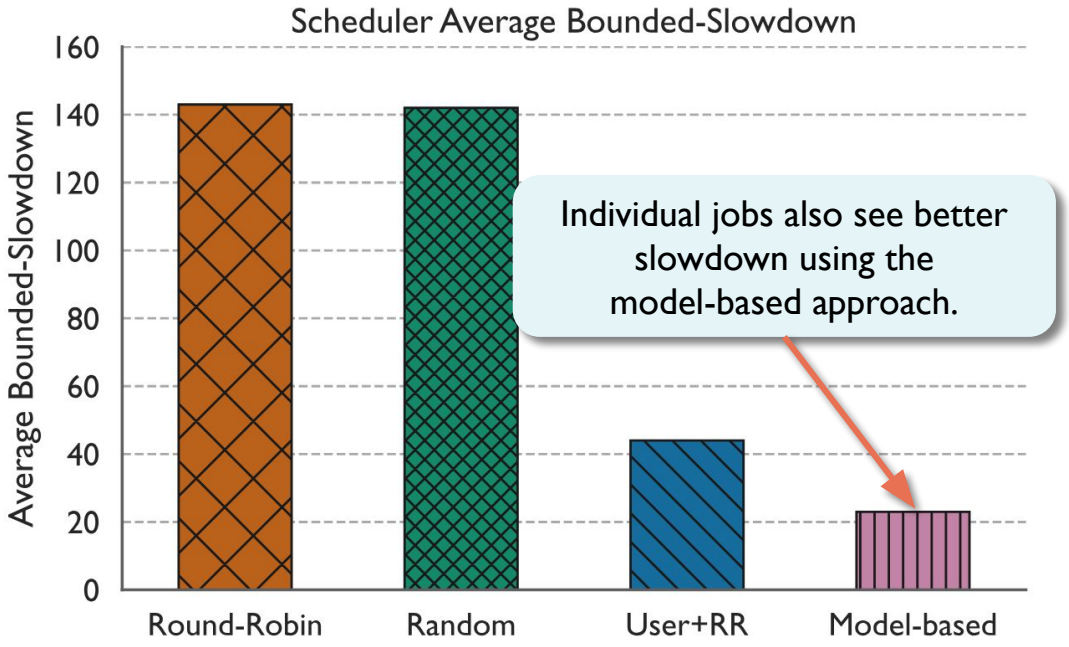



PSSG

UNIVERSITY OF MARYLAND

# Scheduling Results

**RQ3** – *How can we use cross architecture performance predictions to schedule jobs across HPC systems more effectively?*




PSSG

UNIVERSITY OF MARYLAND

# Contributions

- A model that can predict cross-architecture performance with an MAE of 0.11
- An ablation study on the features and data sources that most impact the model
- A demonstration of the potential for using cross-architecture prediction models in multi-cluster schedulers


dnicho@umd.edu