Nonparametric Distribution Models for Predicting and Managing Computational Performance Variability

Thomas C. H. Lux, Layne T. Watson, Tyler H. Chang, Jon Bernard, Bo Li, Xiaodong Yu, Li Xu, Godmar Back, Ali R. Butt, Kirk W. Cameron, Yili Hong, Danfeng Yao

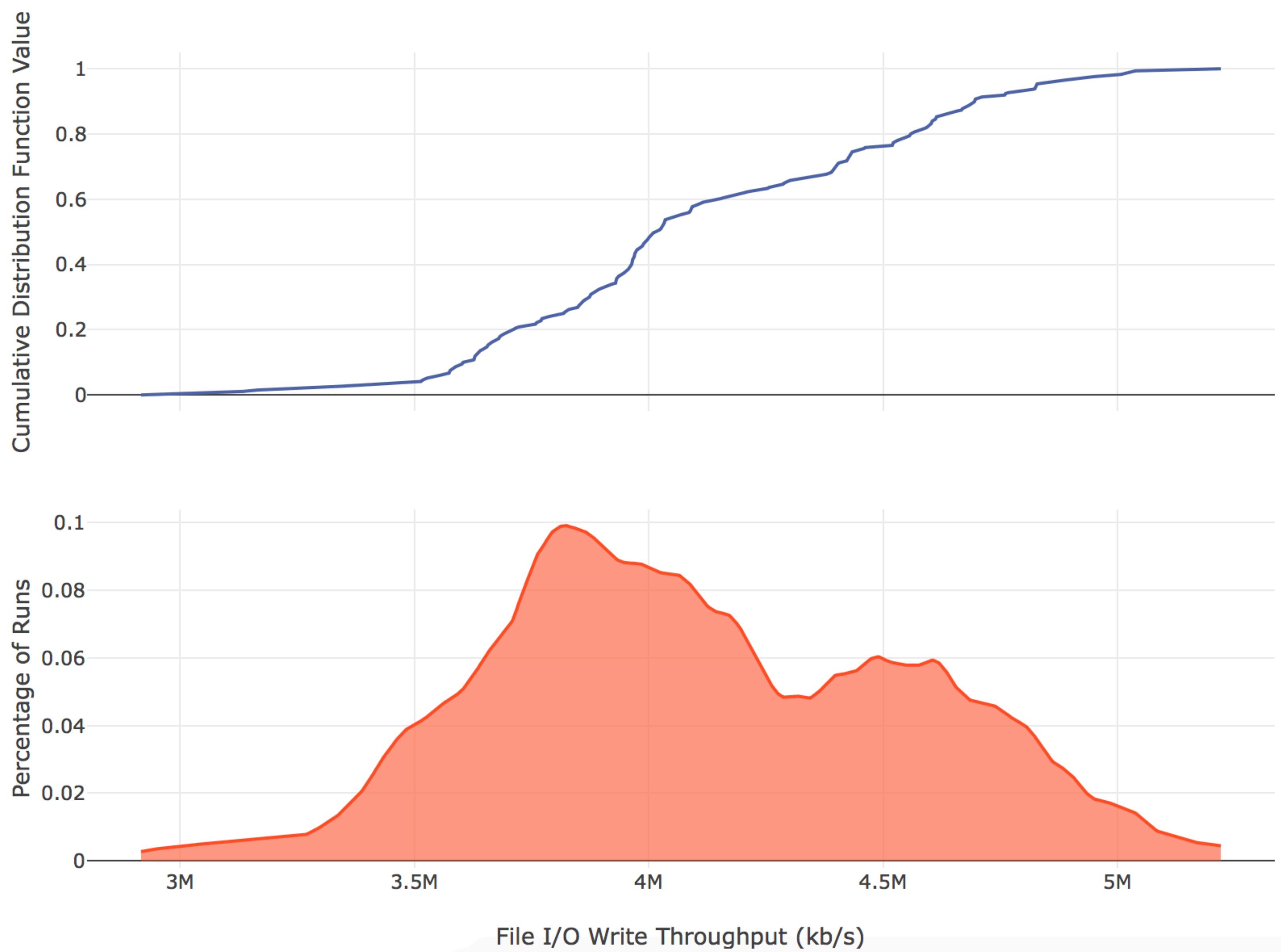
Departments of Computer Science, Mathematics,
Aerospace & Ocean Engineering, and Statistics

Virginia Polytechnic Institute and State University

VIRGINIA TECH™

Problem Outline – Variability

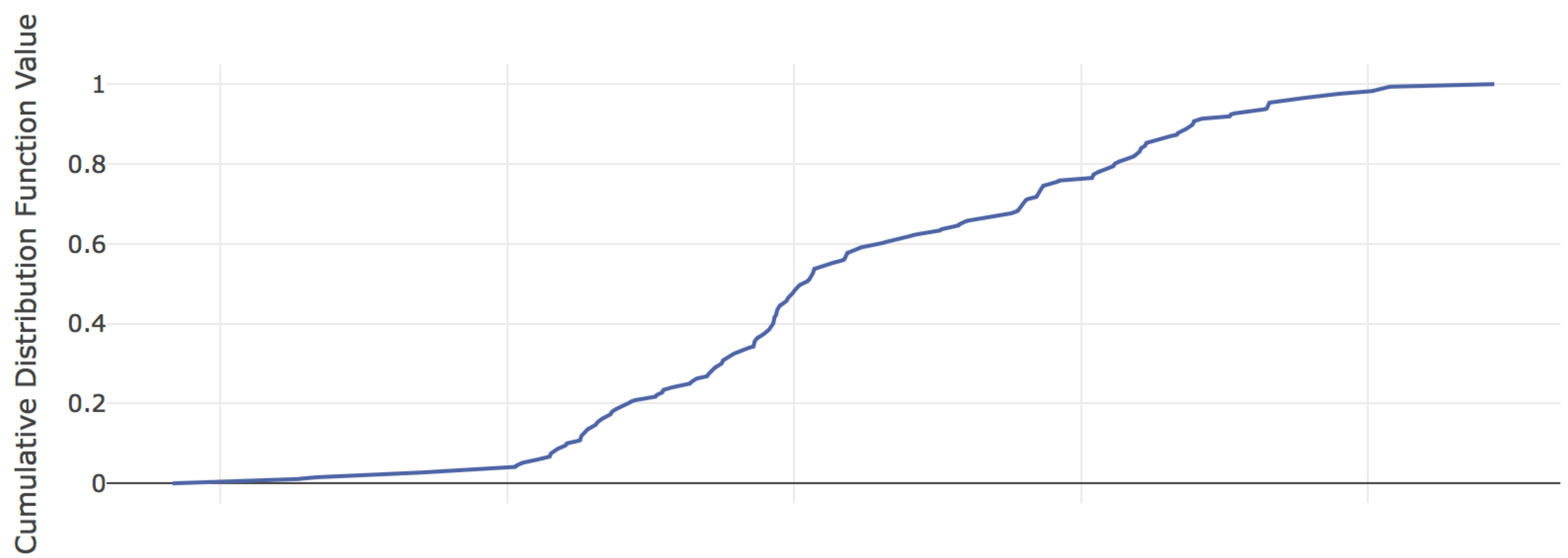
When the same computer is used to execute the same program repeatedly, most measurable performance characteristics will vary.



COMPUTER SCIENCE
VIRGINIA TECH.

Problem Outline – Variability

When the same computer is used to execute the same program repeatedly, most measurable performance characteristics will vary.



Managing this performance variability can be very important in applications where we want to meet some set requirements.

Data Summary

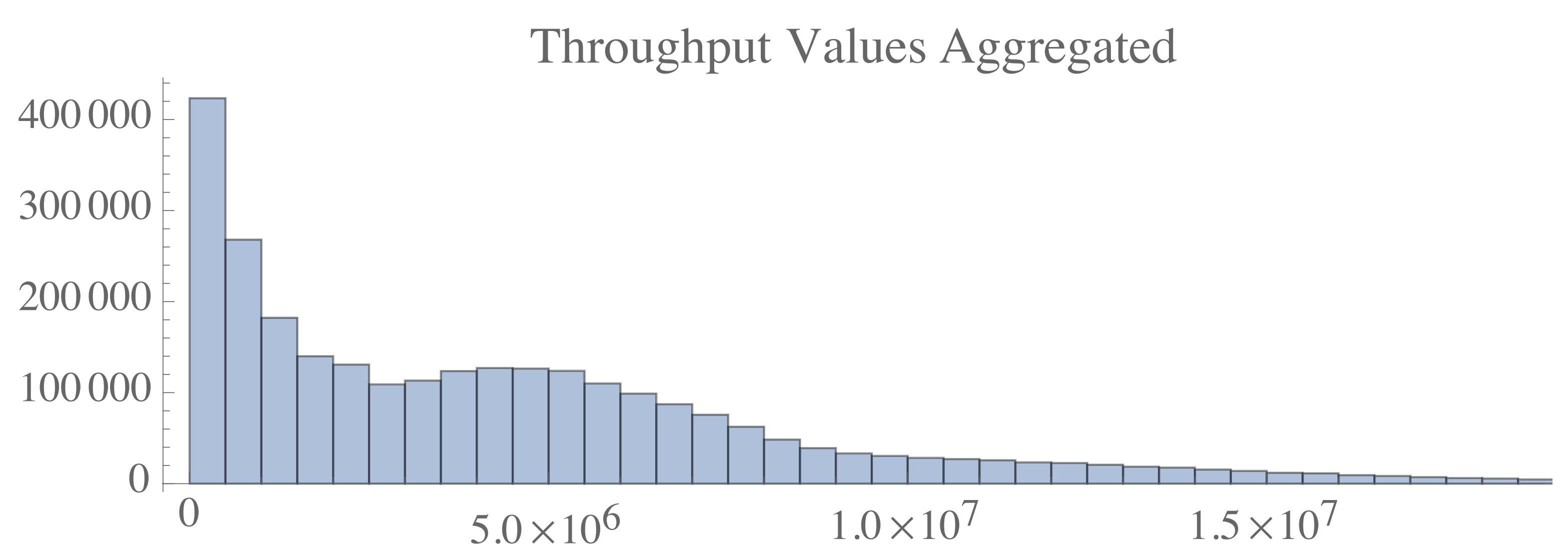
This paper will consider the execution of IOzone, a file I/O benchmark for HPC systems.

Two Intel Xeon E5-2637 CPUs with total 16 CPU cores and 16GB DRAM per node, at 12 nodes.

Ext4 filesystem above an Intel SSDSC2BA20 SSD drive.

Each of ~20K unique system configurations were run 150 times.

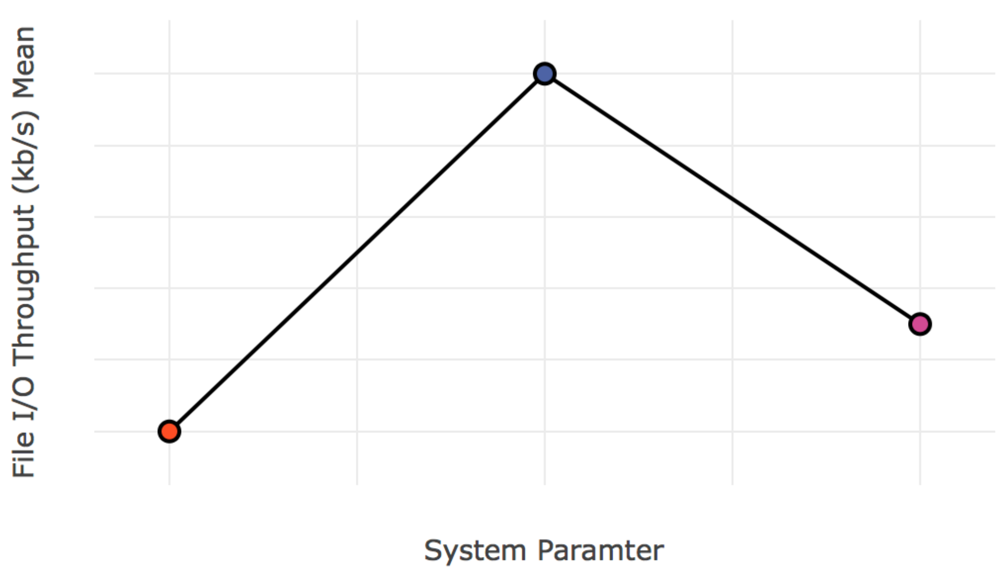
System Parameter	**Values**
File Size (KB)	4, 16, 64, 256, 1024, 4096, 8192, 16384
Record Size (KB)	4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384
Thread Count	1, 8, 16, 24, 32, 40, 48, 56, 64
Frequency (GHz)	1.2, 1.6, 2, 2.3, 2.8, 3.2, 3.5
Test Type	Readers, Rereaders, Random Readers, Initial Writers, Rewriters, Random Writers

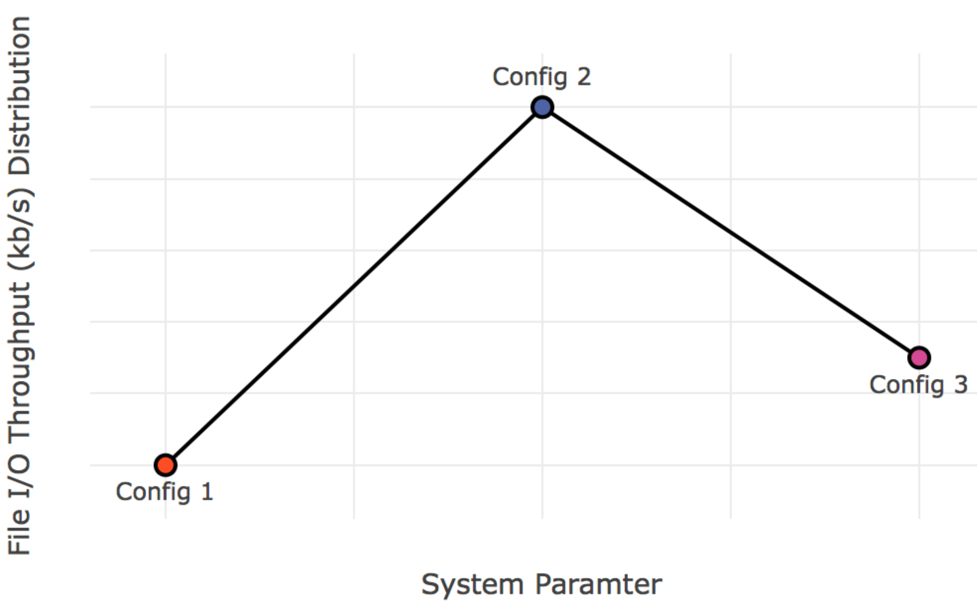


Distributions as Response Values

Existing techniques for modeling system performance have modeled real-valued summary statistics, because it's generally more practical to model functions $f : \mathbb{R}^d \to \mathbb{R}$.

In this work, we consider modeling the system in the form $f : \mathbb{R}^d \to \{g \mid g : \mathbb{R} \to \mathbb{R}\}$, that is predicting the CDF at any system configuration.





COMPUTER SCIENCE
VIRGINIA TECH.

Interpolating Distributions

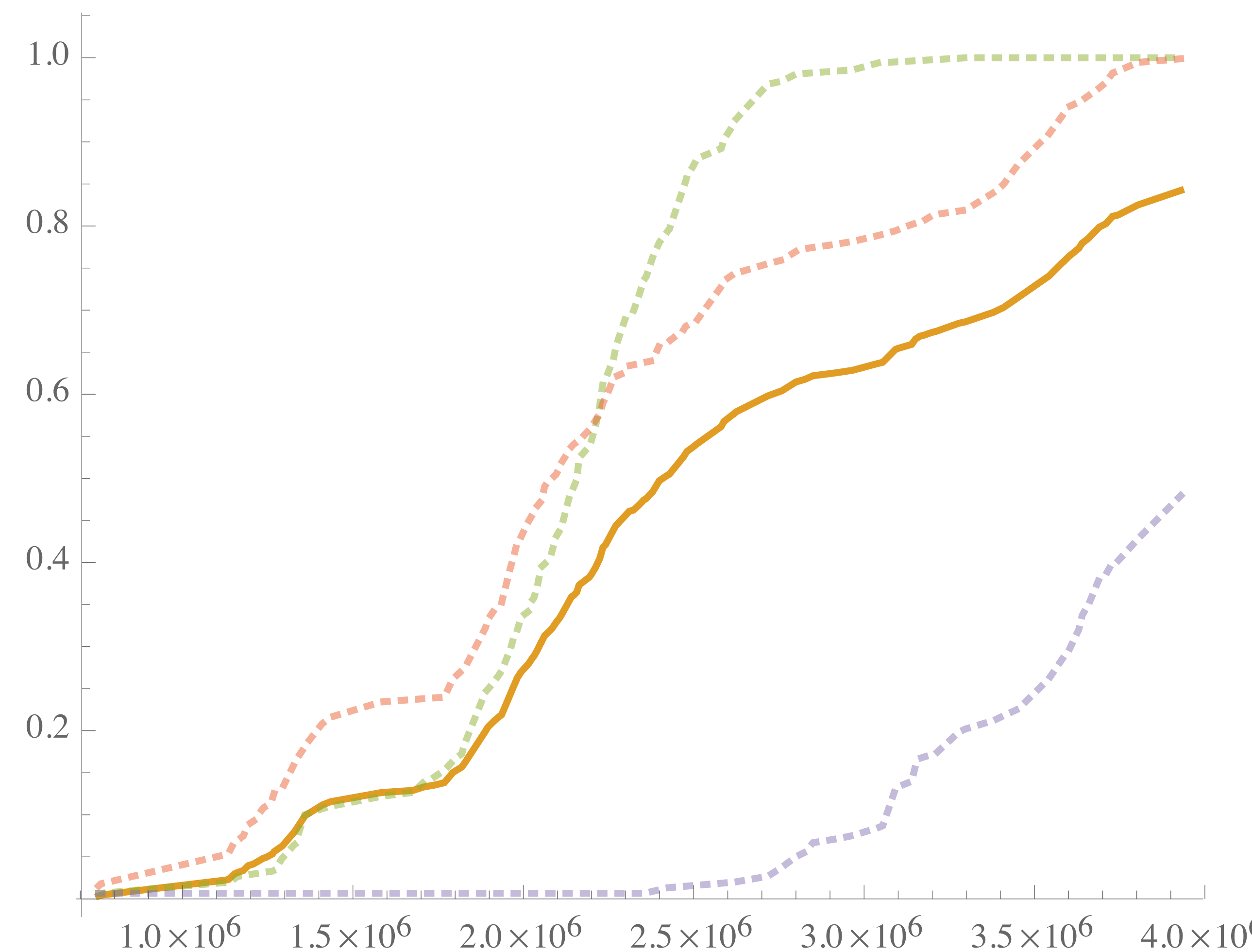
A Cumulative Distribution Function (CDF) $F : \mathbb{R} \to \mathbb{R}$ must maintain the properties:

$$F(x) \in [0, 1]$$

$F(x)$ is right continuous and nondecreasing.

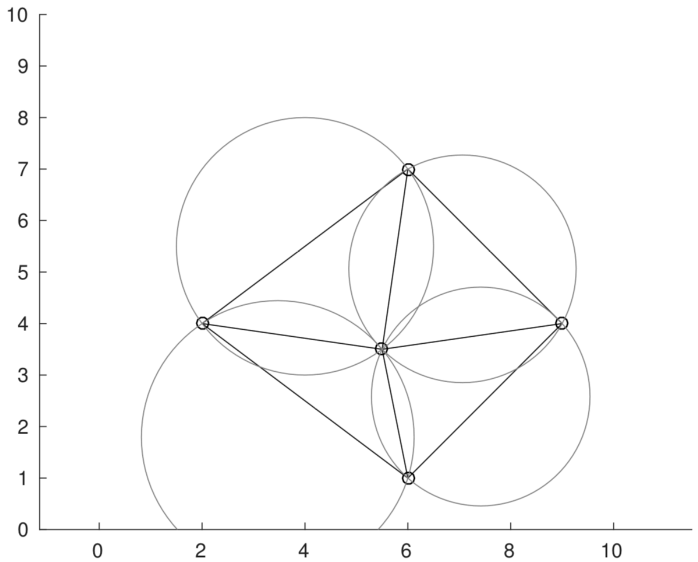
A convex combination of CDFs results in a valid CDF. Consider this example, solid line is the weighted sum:

{.3 Red, .4 Green, .3 Blue}



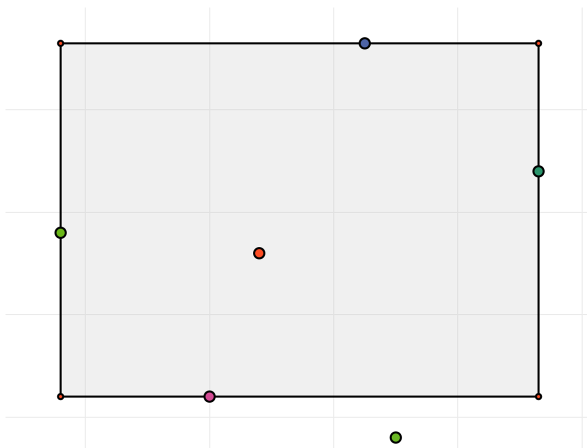
COMPUTER SCIENCE
VIRGINIA TECH.

Three Viable Interpolants



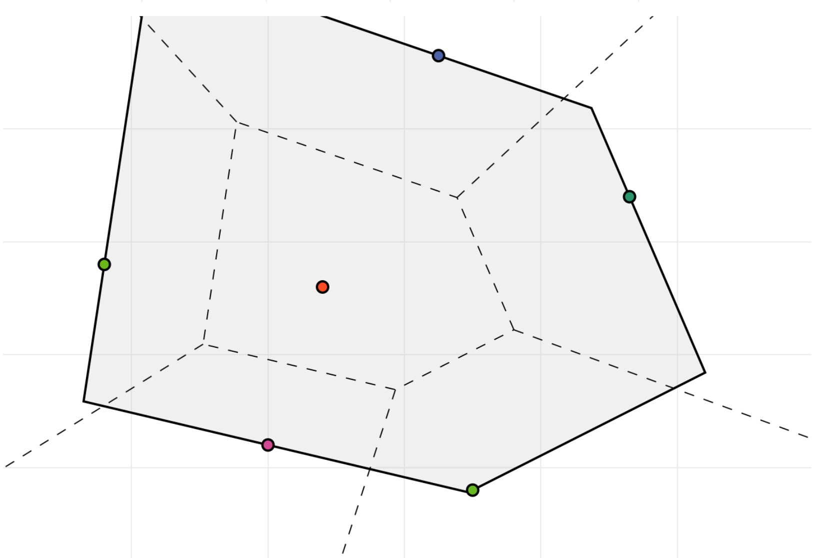
Delaunay Triangulation

$$F_y = \sum_{i=0}^{d} w_i F_{x^{(i)}}$$



Max Box Mesh

$$F_y = \frac{\sum_i b^{x^{(i)}}(y) F_{x^{(i)}}}{\sum_i b^{x^{(i)}}(y)}$$



Voronoi Mesh

$$F_y = \frac{\sum_i v^{x^{(i)}}(y) F_{x^{(i)}}}{\sum_i v^{x^{(i)}}(y)}$$

Tuning the Interpolant

Initially, all system parameters are normalized to be in the unit hypercube. All three interpolants use distance to determine weights in predictions. It may be possible to differentially weight system parameters and improve prediction performance.

Given an error measure, we can use a gradient free (zero order) minimization technique to search for an optimal parameter weighting, one that minimizes apparent error in predictions.

We use 300 iterations of simulated annealing, where the provided "training" data that would usually construct an unweighted model is split up into two sets. 80% is used to calculate a fit and 20% is used to calculate the estimated error for any given weighting.

COMPUTER SCIENCE
VIRGINIA TECH.

Measuring Error in a Prediction

Kolmogorov Smirnov (KS) statistic, max-norm difference.

Null hypothesis (of distributions being same) is rejected at confidence level p according to

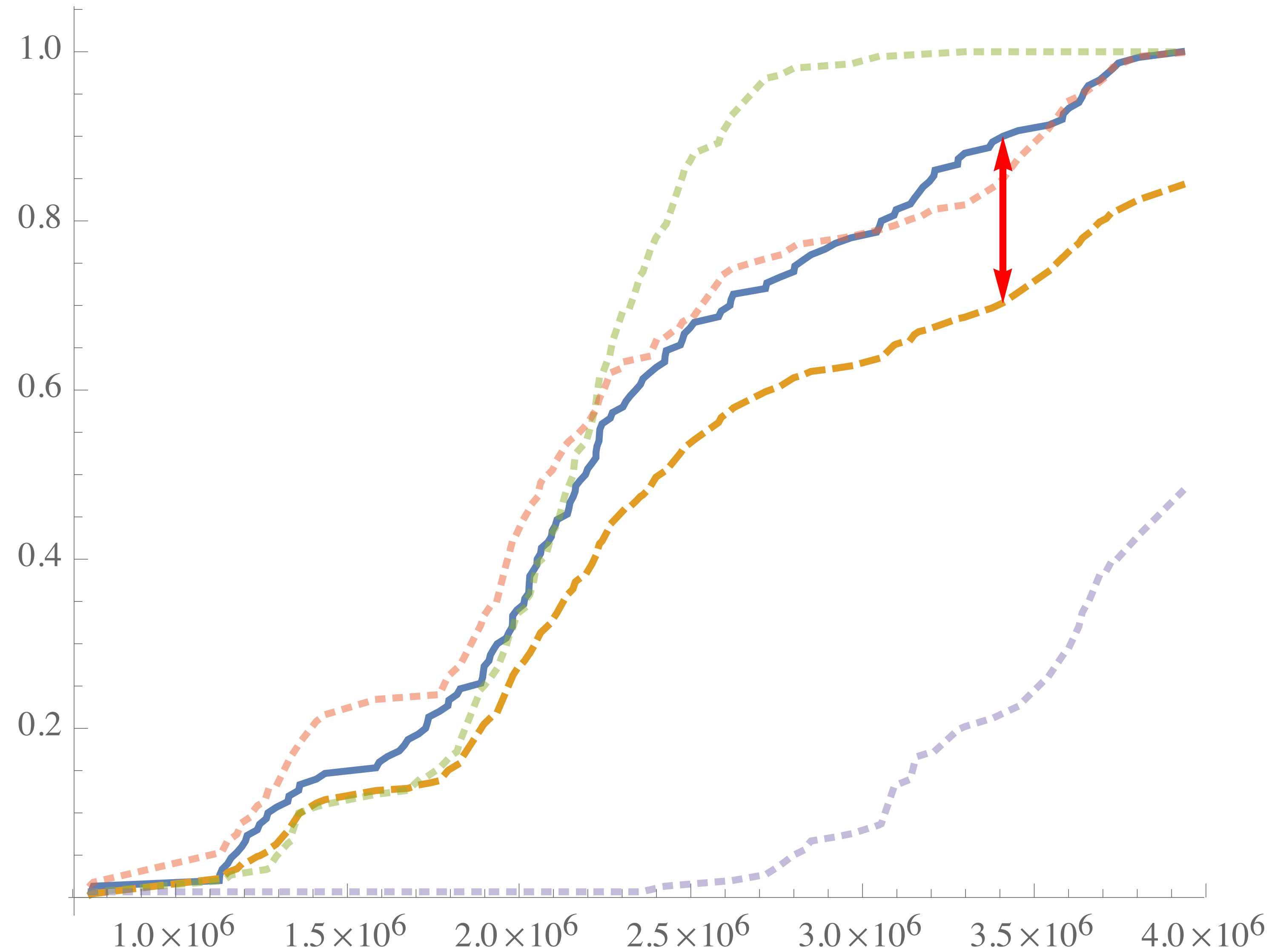
$$KS > \sqrt{-\frac{1}{2}\ln\left(\frac{p}{2}\right)}\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}$$

Dotted lines –
  source CDFs

Dashed line –
  predicted CDF (Delaunay)

Solid line –
  true CDF

Red arrow –
  KS statistic between predicted and true (.2)

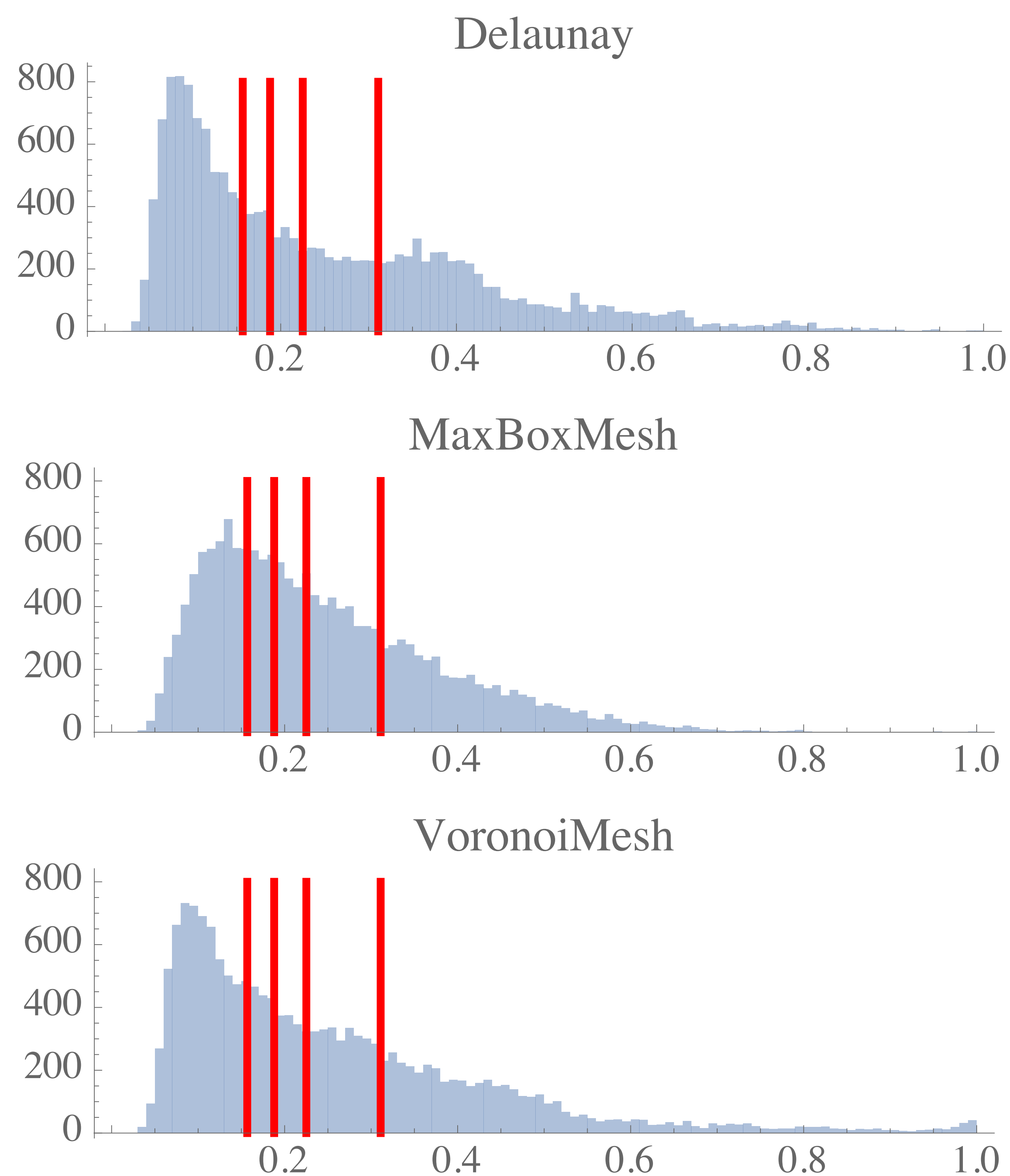


Aggregate Prediction Results

x-axis – KS Statistic
y-axis – Number of predictions

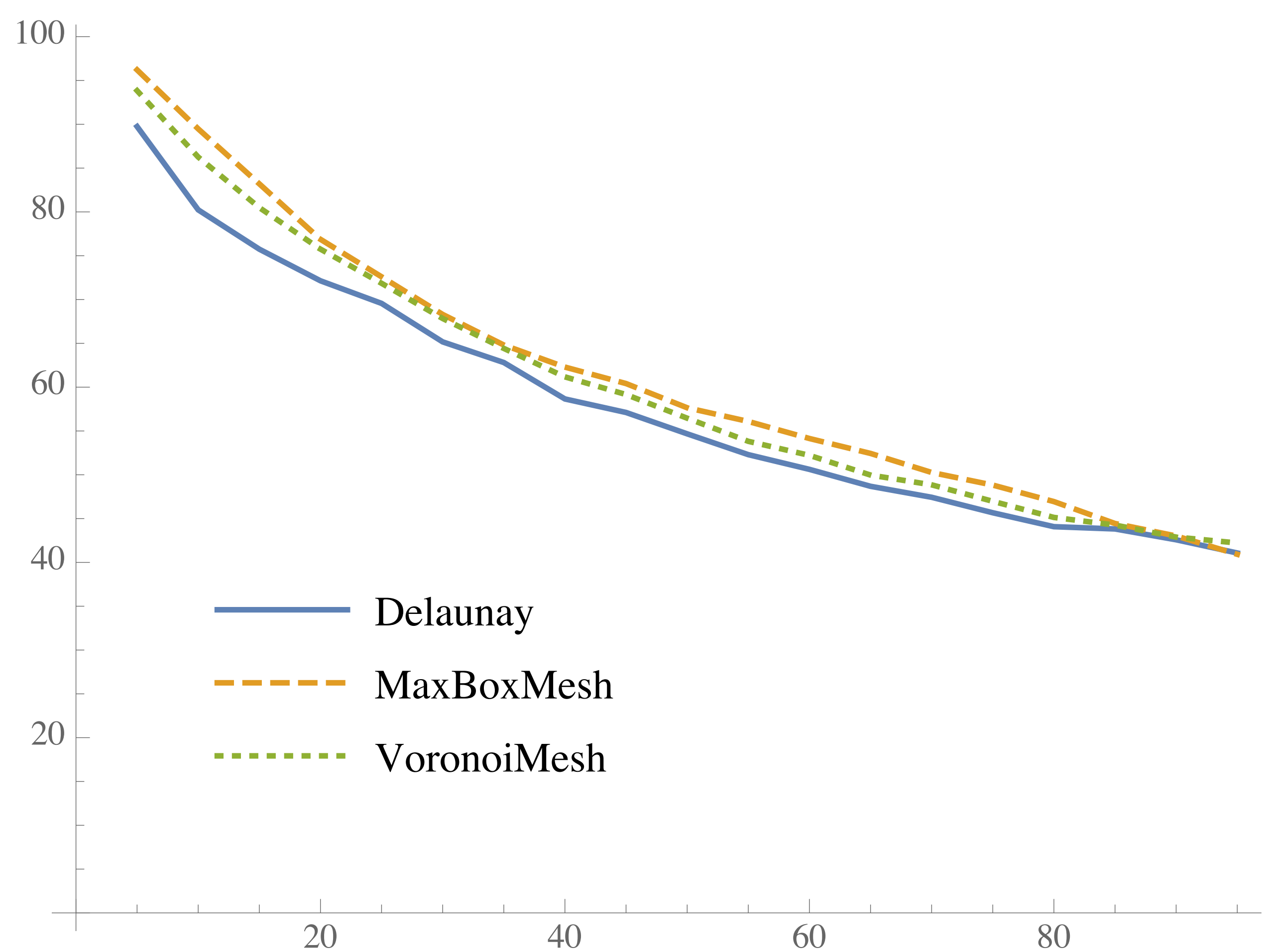
Red lines:
KS significance levels at
{.1 .05, .01, .001}

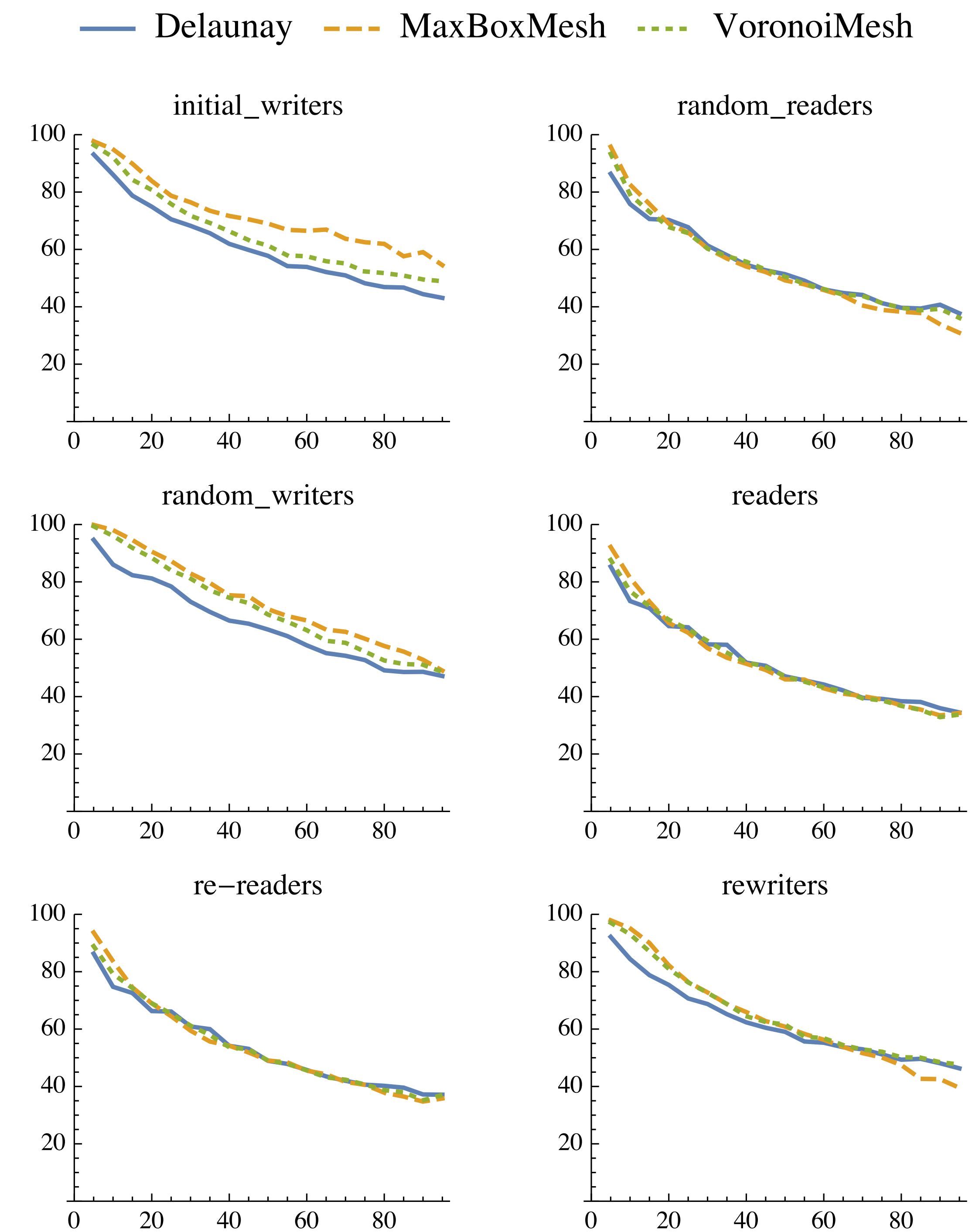
Consider all values to the right of a red line an “incorrect” prediction at that significance.



Increasing Training Data

x-axis – Percentage training data
y-axis – Percentage N.H. rejections
Below: Aggregate
Right: Breakdown by Test





COMPUTER SCIENCE
VIRGINIA TECH.

Performance Change with Tuning

Algorithm	P-Value	Unweighted % N.H. Rejection	Weighted % N.H. Rejection
Delaunay Max Box Mesh Voronoi Mesh	.05	24.9 21.3 18.7	30.2 21.2 11.3
Delaunay Max Box Mesh Voronoi Mesh	.01	21.6 16.4 14.9	27.4 16.4 7.0
Delaunay Max Box Mesh Voronoi Mesh	.001	19.7 13.1 12.3	25.4 13.1 4.6
Delaunay Max Box Mesh Voronoi Mesh	1.0e-6	17.9 11.3 8.5	23.4 11.3 2.3

Consensus optimal weighting of (.001, 2, 1.7, 1.5), for frequency, file size, record size, and number of threads. Frequency is unimportant.

COMPUTER SCIENCE
VIRGINIA TECH.

Potential Applications

Tightening of Service Level Agreements (SLAs) in Cloud Computing systems by managing scheduling under highly irregular usage patterns.

Targeted performance tuning for max throughput and minimum power consumption on HPC systems.

Defense against (and improved attack for) colocated users on systems performing side-channel attacks and targeted exploits.

Conclusions

This new modeling procedure is capable of providing new insights, extending existing analyses, and improving the management of computational performance variability.

Delaunay, Max Box Mesh, and Voronoi Mesh are each viable techniques to use for approximating system performance distributions.

A case study of I/O throughput demonstrates that parameter tuning is a viable procedure for improving performance.

The presented methodology is a notable increase in scalability and capability over existing works modeling system performance.

Acknowledgements

This work was supported by the National Science Foundation as part of the VarSys project.