COMPARISON OF MODELS OF I/O CHARACTERISTICS IN HIGH PERFORMANCE COMPUTING SYSTEMS

Thomas C. H. Lux

Dept. of Computer Science
Virginia Polytechnic Institute
& State University
Blacksburg, VA 24061
tchlux@vt.edu

Layne T. Watson

Dept. of Computer Science
Dept. of Mathematics
Dept. of Aerospace & Ocean Eng.
Virginia Polytechnic Institute
& State University

Tyler H. Chang
Jon Bernard
Bo Li

Dept. of Computer Science
Virginia Polytechnic Institute
& State University

Li Xu

Dept. of Statistics
Virginia Polytechnic Institute
& State University

Godmar Back
Ali R. Butt
Kirk W. Cameron

Dept. of Computer Science
Virginia Polytechnic Institute
& State University

Yili Hong

Dept. of Statistics
Virginia Polytechnic Institute
& State University

ABSTRACT

Each of high performance computing, cloud computing, and computer security have their own interests in modeling and predicting the performance of computers with respect to how they are configured. An effective model might infer internal mechanics, minimize power consumption, or maximize computational throughput of a given system. This paper analyzes a seven-dimensional dataset measuring the input/output (I/O) characteristics of a cluster of identical computers using the benchmark IOzone. The I/O performance characteristics are modelled with respect to system configuration using multivariate interpolation and approximation techniques. The analysis reveals that accurate models of I/O characteristics for a computer system may be created from a small fraction of possible configurations, and that some modeling techniques will continue to perform well as the number of system parameters being modeled increases. These results have strong implications for future predictive analyses based on more comprehensive sets of system parameters.

SpringSim-HPC, 2018 April 15-18, Baltimore, MD, USA; ©2018 Society for Modeling & Simulation International (SCS)

Keywords: Regression, approximation, other stuff

1 INTRODUCTION

Performance tuning is often an experimentally complex and time-intense chore necessary for configuring HPC systems. The procedures for this tuning vary largely from system to system and are often subjectively guided by the system engineer(s). Once a desired level of performance is achieved, HPC systems often remain in that base configuration with all subsequent modifications being incremental. The changes made are often associated with updates and job-specific customizations. In the case that a system has changing workloads or non-stationary performance objectives that range from maximizing computational throughput to minimizing power consumption and system variability, it becomes obvious that a more effective and automated tool is needed for configuring systems. This scenario presents a challenging and important application of multivariate approximation and interpolation techniques.

1. The value of multivariate modlling
2. The data context
3. The proposed method for using multivariate models
4. The impact of effective models

1.1 Approximation

This paper compares five multivariate approximation techniques that operate on inputs in $\mathbb{R}^d$ (vectors of d real numbers) and produce predicted responses in $\mathbb{R}^1$. Three of the techniques are regression based and produce knowingly-inaccurate, but hopefully more generalizable models. The ramaining two techniques are interpolation techniques that reproduce input data exactly at the expense of higher model complexity. The sections below outline the mathematical formulations of each technique and provide computational complexity bounds.

1.1.1 Multivariate Adaptive Regression Splines

1.1.2 Multi-Layer Perceptron Regressor

1.1.3 Support Vector Regressor

1.2 Interpolation

1.2.1 Linear Shepard

1.2.2 Delaunay

2 RELATED WORK

1. Not sure how much to include here? Shooting for thoroughness or simply necessary coverage?

3 METHODOLOGY

3.1 Data

3.2 Dimensional Analysis

1. Cycling the categorical settings
2. Selecting subsets of 1,2,3 up to 4 dimensions
3. Cycling different training : testing ratios (5:95 → 95:5)
4. Generating 200 random training : testing splits that ensure the testing points are not outside the convex hull of the training.
5. Selecting training points to be well-spaced using QNSTOPP algorithm.

3.3 Prediction

1. For each file generated from the dimensional analysis, train on the training data, evaluate at the testing data points

4 RESULTS

4.1 I/O Throughput Mean

4.2 I/O Throughput Variance

4.3 Increasing Dimension

5 DISCUSSION

5.1 Modelling the System

5.2 Quantifying Variability

5.3 Extending the Analysis

6 FUTURE WORK

REFERENCES