Input-Output Performance Evaluation:
Self-Scaling Benchmarks, Predicted Performance

by

Peter Ming-Chien Chen

B.S. (Pennsylvania State University) 1987
M.S. (University of California at Berkeley) 1989

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY of CALIFORNIA at BERKELEY

Committee in Charge:

Professor David A. Patterson, Chair
Professor Randy H. Katz
Professor Ronald Wolff

The dissertation of Peter Ming-Chien Chen is approved:

Chair Date

Date

Date

University of California at Berkeley

1992

Input-Output Performance Evaluation:

Self-Scaling Benchmarks, Predicted Performance

Copyright © 1992

by

Peter Ming-Chien Chen

Abstract

Input-Output Performance Evaluation: Self-Scaling Benchmarks, Predicted Performance

by

Peter Ming-Chien Chen

Doctor of Philosophy in Computer Science

University of California at Berkeley

Professor David A. Patterson, Chair

Over the past 20 years, processor performance has been growing much faster than input/output (I/O) performance. As this occurs, overall system speed becomes more and more limited by the speed of I/O systems and hence I/O systems are evolving to keep up with processor performance. This evolution renders current I/O performance evaluation techniques obsolete or irrelevant, despite their increasing importance. This dissertation investigates two new ideas in I/O evaluation, self-scaling benchmarks and predicted performance.

This dissertation's self-scaling benchmark seeks to measure and report relevant workloads for a wide range of input/output systems. To do so, it scales aspects of its workload to account for the differences in I/O systems. For example, it dynamically discovers the size of the system's file cache and reports how performance varies both in and out of the file cache. The

general approach taken is to scale based on the range of workloads the system performs well.

The self-scaling benchmark helps the evaluator gain insight into the system's performance by displaying how performance varies against each of five workload parameters: amount of file space, request size, fraction of reads, fraction of sequential accesses, and number of simultaneous accesses. The utility of the benchmark is demonstrated by running it on a wide variety of I/O systems, ranging from a single disk, low-end workstation to a mini-supercomputer with an array of four disks. On each system, the benchmark helps provide performance insights, such as the size of the file cache, the performance increases due to larger requests, the file cache's write policy, and the benefits of higher workload concurrency.

Predicted performance restores the ability to compare two machines on the same workload, which was lost in the self-scaling benchmark. Further, it extends this ability to workloads that have not been measured by estimating performance based on the graphs from the self-scaling benchmark. Prediction is accurate to within 10-15% over a wide range of I/O workloads and systems. This high level of accuracy demonstrates how a large workload space can be described using a few tens of points and a simple product form performance equation.

Professor David A. Patterson, Chair

Dedicated to my wife, Janet

for your help in persevering,

your reminders for me to keep an eternal perspective,

and your unconditional love and commitment.

Table of Contents

Chapter 1: Introduction 1

1.1 Why I/O Performance? 1

1.2 Overview of the Dissertation 3

1.3 Metrics 4

1.4 Trends in I/O Systems 7

1.5 References 12

Chapter 2: Previous Work 17

2.1 I/O Benchmarks 17

2.2 The Ideal I/O Benchmark 18

2.3 System Platforms 20

2.4 Overview of Current I/O Benchmarks 21

2.4.1 Application Benchmarks 22

2.4.1.1 Andrew 22

2.4.1.2 TPC-B 23

2.4.1.3 Sdet 25

2.4.2 Synthetic Benchmarks 26

2.4.2.1 Bonnie 26

2.4.2.2 IOStone 27

2.4.2.3 Sample Scientific Workload 28

2.4.2.4 LADDIS 28

2.5 Critique of Current Benchmarks 30

2.6 References 32

Chapter 3: Workload Model 36

3.1 Overview 36

3.2 I/O Workloads Used in Research 37

3.3 I/O Tracing Studies 38

3.4 Willy 40

3.4.1 Workload Parameters 40

3.4.1.1 Locality 40

3.4.1.2 Request Characteristics 41

3.4.1.3 Load 42

3.4.2 Other Workload Issues 42

3.5 Modeling Real Applications 43

3.6 Representativeness of Willy 44

3.7 References 46

Chapter 4: A Self-Scaling Benchmark 49

4.1 Overview 49

4.2 Single Parameter Graphs 50

4.3 The Knee Point 53

4.4 A Global Knee Self-Scaling Benchmark 58

4.4.1 Example Run of Global-Knee, Self-Scaling Benchmark 59

4.4.2 Problems 60

4.5 A Better Self-Scaling Benchmark 62

4.5.1 Examples 64

4.5.1.1 SPARCstation 1+ 64

4.5.1.2 DECstation 5000/200 66

4.5.1.3 HP 730 69

4.5.1.4 Convex 70

4.5.1.5 Solbourne 70

4.5.1.6 Raw Disk Interface 74

4.5.1.7 Client-Server 74

4.5.1.8 Unannounced Workstation 75

4.5.2 Running Time 77

4.6 Conclusions 78

4.7 References 79

Chapter 5: Predicted Performance 81

5.1 Introduction 81

5.2 Prediction Models 82

5.3 Verification of Prediction Model 84

5.3.1 SPARCstation 1+ 86

5.3.2 DECstation 5000/200, HP 730, Convex C240 88

5.4 Comparison Against Orthogonal Sampling 98

5.5 Application of Predicted Performance—Performance Ratios 100

5.6 Conclusions 100

5.7 References 103

Chapter 6: Conclusions and Future Work 104

6.1 Conclusions 104

6.2 Future Work 106

Appendix A: Data Used in Prediction 109

Appendix B: Prediction Algorithm 120

Appendix C: Proof of Performance Equation 123

List of Figures

Figure 1.1 Contrasting trends of CPU and disk performance improvements 2

Figure 1.2 Example response time versus throughput graph 5

Figure 1.3 Combining multiple smaller disks to improve performance 9

Figure 2.1 TPC-B results 23

Figure 2.2 Sdet results 25

Figure 2.3 Current state of I/O benchmarks 30

Figure 4.1 Results from a self-scaling benchmark that scales all parameters 52

Figure 4.2 Workloads reported by a set of single parameter graphs 53

Figure 4.3 Difficulty in defining the knee 55

Figure 4.4 Global knee point with two parameters 57

Figure 4.5 Instability in graphs on the border of performance regions 61

Figure 4.6 Slope of uniqueBytes curve for SPARCstation 1+ 63

Figure 4.7 Results from a better self-scaling benchmark for a SPARCstation 1+ 65

Figure 4.8 Self-scaling benchmark for DECstation 5000/200—Part 1 67

Figure 4.9 Self-scaling benchmark for DECstation 5000/200—Part 2 68

Figure 4.10 Self-scaling benchmark for HP 730 69

Figure 4.11 Self-scaling benchmark for Convex C240 71

Figure 4.12 Self-scaling benchmark for Solbourne 72

Figure 4.13 Self-Scaling Results using the Raw Disk Interface 73

Figure 4.14 Self-Scaling Results for Two Client-Server Configurations 75

Figure 4.15 Self-scaling benchmark for unannounced workstation (beta release) 76

Figure 5.1 Predicting performance of unmeasured workloads 83

Figure 5.2 Predicting performance of four traditional benchmarks 85

Figure 5.3 Evaluation of prediction accuracy for SPARCstation 1+ with one disk 87

Figure 5.4 What parameters cause errors for SPARCstation 1+ 89

Figure 5.5 Evaluation of prediction accuracy for Sprite DECstation 5000/200 90

Figure 5.6 What parameters cause errors for DECstation 5000/200 91

Figure 5.7 Evaluation of prediction accuracy for HP 730 92

Figure 5.8 What parameters cause errors for HP 730 93

Figure 5.9 Evaluation of prediction accuracy for Convex C240 94

Figure 5.10 What parameters cause errors for Convex C240 95

Figure 5.11 Enhanced prediction accuracy 97

Figure 5.12 Prediction accuracy using interpolation on an orthogonal sample 99

Figure 5.13 Measured versus predicted ratio 102

List of Tables

Table 1.1	Metrics for two I/O systems	7
Table 1.2	Magnetic disk performance improvement over the past 20 years	8
Table 1.3	1992 storage capacity	10
Table 2.1	System platforms	20
Table 2.2	List of contacts for various benchmarks	21
Table 2.3	Results from the Andrew benchmark	22
Table 2.4	Results from Bonnie	26
Table 2.5	Results from IOStone	27
Table 2.6	Results from two sample scientific workloads	28
Table 3.1	Workload characterization of benchmarks/applications	44
Table 3.2	Representativeness of Willy	45
Table 4.1	Description of other machines used in self-scaling benchmark examples	64
Table 5.1	Summary of median prediction errors	96
Table 5.2	Workloads to be run on all systems	101
Table A.1	Raw data for SPARCstation 1+ scatter plot	112
Table A.2	Raw data for DECstation 5000/200 scatter plot	114
Table A.3	Raw data for HP 730 scatter plot	116
Table A.4	Raw data for Convex C240 scatter plot	119
Table A.5	Raw data for ratio prediction	119

Acknowledgements

I am grateful to many people for helping me finish this thesis. My advisor, David Patterson, has been the best advisor I could imagine. Since the time he welcomed me on board the RAID project, he has been providing interesting research ideas, encouragement to persevere through graduate school, wise advice through the ‘‘fallacies and pitfalls’’ of an academic career, enthusiasm to learn and teach, and fatherly friendship. He has been a great role model for me as I begin my vocation as a university professor, and I hope to be, like him, a teacher and researcher who genuinely cares for my students.

Randy Katz has been a very supportive secondary advisor. As Principal Investigator of the Berkeley RAID project, he gave me the opportunity to be involved in an interesting research project, RAID-II. Through this project, I have gained valuable, hands-on implementation experience. I am thankful to Randy and my third reader, Ronald Wolff, for giving me insightful, timely feedback and suggestions for this thesis.

Most Ph.D. students have only one advisor; I was lucky enough to have not only Dave Patterson and Randy Katz but also another mentor, fellow graduate student (now professor) Garth Gibson. He, with Dave and Randy, guided me through graduate school and shaped the way I approach research problems and write papers.

I have enjoyed working with the RAID and Sprite project members. Students from both projects have been good sounding boards for this research. In particular, Garth Gibson and Ed Lee have helped shape much of my research, from discussions on striping units to I/O workloads. Mendel Rosenblum and John Hartman helped me understand file systems and file caches. Ken Shirriff helped me develop an informal proof about the global knee of Chapter 4. Ken Lutz taught me countless tips on how to debug real hardware in the context of RAID-II.

My parents have consistently supported me through life, teaching me the value of education and helping me form a sensible priority system. Without my parents, I would not have striven for an academic position and may not have persevered through graduate school. The friends I met through Campus Crusade for Christ have been my family in California and have given zest to my life here.

One of those friends in Campus Crusade for Christ eventually has become my closest family. My wife, Janet, has been a wonderful partner in life while I've written this dissertation. Our first one and a half years of marriage have been the best period of my life, both personally and academically.

Finally, I would like to explain, in story form, a truth I have understood while in graduate school. Let us say you are a judge and I am a drunk driver. Not only are you a judge, you are also my father. While driving through town, I crash through a very expensive store window. The penalty for this crime is either paying for the store window (worth, say, $100,000) or working without pay for the store until the debt is paid off. As my father, you know I am broke and can not pay the $100,000. The dilemma is this: naturally, you love me and do not want me to work the rest of my life paying back the store. But, being a just judge, you can not arbitrarily overlook my crime.

Here is a possible solution to this quandary: you could offer to pay the penalty for me, thus accepting the cost of my error on yourself. I need only accept your offer.

This hypothetical situation describes fairly closely our situation before God. All of us (including myself) are naturally selfish and have done wrong things, whether of the more heinous variety, such as murder, or the more everyday variety, such as hatred. The penalty for these wrongdoings is spiritual death, which is eternal separation from God. God, as a Father who loves us, does not wish for us to pay this penalty, but, nonetheless, the penalty must be paid for God to be just. He solves this dilemma by penalizing Himself in the form of Jesus.

Jesus's death was God's payment for our mistakes. God offers to apply this payment for each person, but not everyone has accepted His offer, either due to misunderstanding or due to a desire to trust solely in one's own ability to pay. If you are willing to accept God's payment for your mistakes, simply tell God of your acceptance and begin to yield to Him as your new master.

For the wages of sin is death, but the free gift of God is eternal life in Christ Jesus our Lord. (Romans 6:23)