Statistically Rigorous Regression Modeling for the Microprocessor Design Space

Benjamin C. Lee[1,2], David M. Brooks[1]

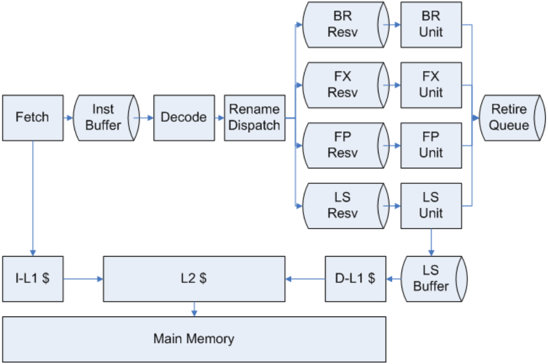
bclee@deas.harvard.edu
[1]Division of Engineering and Applied Sciences
Harvard University

[2]Center for Applied Scientific Computing Research
Lawrence Livermore National Laboratory

Outline

Outline

Microarchitectural Design Space



- Trend toward chip multiprocessors (CMP's) with varying core designs
- Power 4, Pentium 4, UltraSPARC T1
- Tractably quantify trade-offs between core complexity, count

Design Space Exploration

- **Limitations of Existing Simulation Methodology**
  - Trace sampling, compression reduce per simulation costs
  - Existing techniques do not reduce number of simulations
  - Space size increases exponentially with parameter count
  - Multi-threaded, multi-core simulations further constrained
- **Prior Design Space Analyses**
  - Consider m^p design points
  - Vary one or two parameters at fine granularity
  - Vary multiple parameters at coarse granularity
  - Hold majority of parameters at constant values

Simulation Paradigms

- **Objectives**
  - Comprehensively understand microprocessor design space
  - Selectively perform a modest number of simulations
  - Efficiently leverage simulation data
- **Random Configuration Sampling**
  - Sample points UAR from design space for simulation
  - Controls exponential increase in design count
- **Statistical Inference**
  - Reveals trends, trade-offs from sparse sampling
  - Enables prediction for metrics of interest

Statistical Inference

- **Approach**
  - Models approximate solutions to intractable problems
  - Requires initial data to train, formulate model
  - Leverages correlations from initial data for prediction
- **Regression Modeling**
  - Efficient formulation :: sample 1K of $\approx$1B, least squares
  - Accurate inference :: $4 - 7\%$ median error
  - Static accuracy :: no predictive training

Model Formulation

- **Notation**
  - n observations
  - Response :: $y = y_1, \ldots, y_n$
  - Predictor :: $x_i = x_{i,1}, \ldots, x_{i,p}$
  - Regression Coefficients :: $\beta = \beta_0, \ldots, \beta_p$
  - Random Error :: $e = e_1, \ldots, e_n$ where $e_i \sim N(0, \sigma^2)$
  - Transformations :: $f, g = g_1, \ldots, g_p$
- **Model**

$$\begin{aligned} f(y_i) &= \beta g(x_i) + e_i \\ &= \beta_0 + \sum_{j=1}^{p} \beta_j g_j(x_{ij}) + e_i \end{aligned}$$

Predictor Interaction

- **Modeling Interaction**
  - Suppose effects of predictors x_1, x_2 cannot be separated
  - Construct predictor $x_3 = x_1 x_2$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + e_i$$

- **Example**
  - Let x_1 be pipeline depth, x_2 be L2 cache size
  - Performance impact of pipelining affected by cache size

$$Speedup = \frac{Depth}{1 + Stalls/Inst}$$

Predictor Non-Linearity

- **Restricted Cubic Splines**
  - Divide predictor domain into intervals separated by knots
  - Piecewise cubic polynomials joined at knots [1]
  - Higher order polynomials provide better fits
- **Location of Knots**
  - Location of knots less important than number of knots
  - Place knots at fixed predictor quantiles
- **Number of Knots**
  - Flexibility, risk of over-fitting increases with knot count
  - 5 knots or fewer are often sufficient
  - 4 knots balances flexibility, over-fitting

[1] Stone [SS'86]

Prediction

- **Expected Response**
  - Suppose coefficients β, predictors' $x_{i,1}, \ldots, x_{i,p}$ are known
  - Expected response is weighted sum of predictor values

$$\begin{aligned} E\left[y_i\right] &= E\left[\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}\right] + E\left[e_i\right] \\ &= \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \end{aligned}$$

Outline

Tools and Benchmarks

- **Simulation Framework**
  - Turandot :: a cycle-accurate trace driven simulator
  - PowerTimer :: power models derived from circuit analyses
  - Baseline simulator models POWER4/POWER5 architecture
- **Benchmarks**
  - SPEC2kCPU :: compute-intensive benchmarks
  - SPECjbb :: Java server benchmark
- **Statistical Framework**
  - R :: software environment for statistical computing
  - Hmisc and Design packages[2]

[2] Harrell [Springer,'01]

Configuration Sampling

- **Design Space Size**
  - For $i \in [1, p]$, S_i defines possible values for parameter x_i
  - $S = \prod_{i=1}^{p} S_i$ defines design space
  - $|S| = \prod_{i=1}^{p} |S_i|$ defines space size
  - B defines set of benchmarks, $|B| \times |S|$ potential simulations
  - $|S| \approx 10^9$ and $|B| = 22$
- **Sampling Uniformly at Random (UAR)**
  - Sample $n = 4,000$ design points and benchmarks
  - Unbiased observations from full range of parameter values
  - Trends, trade-offs between parameters at fine granularity

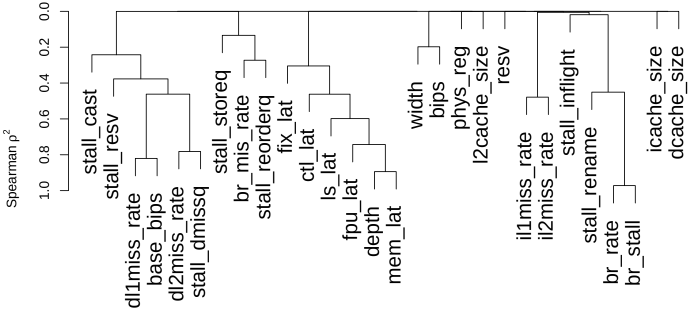
Predictors :: Microarchitecture

	Set	Parameters	Measure	Range	$\|S_i\|$
S_1	Depth	depth	FO4	9::3::36	10
S_2	Width	width L/S reorder queue store queue functional units	insn b/w entries entries count	4,8,16 15::15::45 14::14::42 1,2,4	3
S_3	Physical Registers	general purpose (GP) floating-point (FP) special purpose (SP)	count count count	40::10::130 40::8::112 42::6::96	10
S_4	Reservation Stations	branch fixed-point/memory floating-point	entries entries entries	6::1::15 10::2::28 5::1::14	10
S_5	I-L1 Cache	i-L1 cache size	log_2(entries)	7::1::11	5
S_6	D-L1 Cache	d-L1 sache size	log_2(entries)	6::1::10	5
S_7	L2 Cache	L2 cache size L2 cache latency	log_2(entries) cycles	11::1::15 6::2::14	5
S_8	Control Latency	branch latency	cycles	1,2	2
S_9	FX Latency	ALU latency FX-multiply latency FX-divide latency	cycles cycles cycles	1::1::5 4::1::8 35::5::55	5
S_{10}	FP Latency	FPU latency FP-divide latency	cycles cycles	5::1::9 25::5::45	5
S_{11}	L/S Latency	Load/Store latency	cycles	3::1::7	5
S_{12}	Memory Latency	Main memory latency	cycles	70::5::115	10

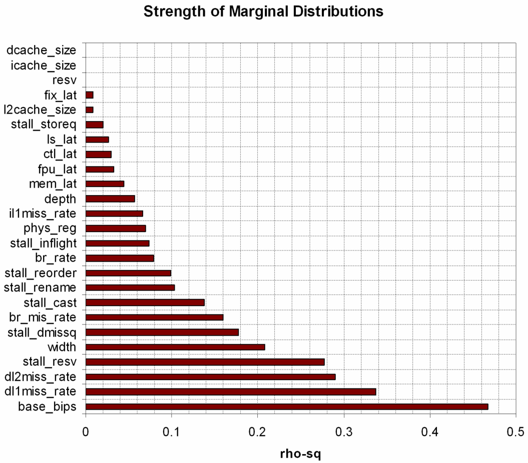
Predictors :: Application-Specific

- **Application Characteristics**
  - Collect program characteristics on baseline architecture
  - Baseline instruction throughput (BIPS)
  - Cache access patterns (i-L1, d-L1, L2 miss rates)
  - Branch patterns (branch frequency, mispredict rate)
  - Sources of pipeline stalls (per queue stall histograms)
- **Application Effects**
  - Characteristics are significant predictors when interacting with microarchitectural predictors
  - Example :: Impact of d-L1 cache affected by access rates

Variable Clustering



Strength of Marginal Relationships



Regression Model Specification

- **Interactions**
  - Pipeline width/depth interact with
    - instruction bandwidth structures (queues, register file)
    - cache hierarchy
  - Cache hierarchy sizes interact with
    - adjacent levels in hierarchy
    - application-specific access rates
  - Baseline performance interacts with resource sizings
- **Restricted Cubic Splines**
  - Weaker relationships (latencies, caches, queues) :: 3 knots
  - Stronger relationships (depth, registers) :: 4 knots
  - Baseline application performance :: 5 knots

Outline

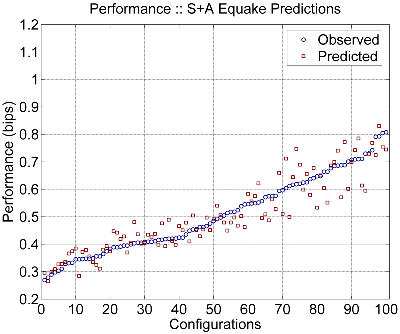
Validation Approach

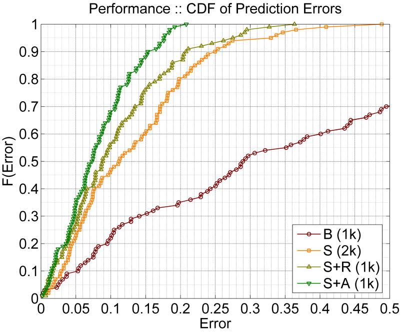
- **Framework**
  - Formulate models with $n_* < n = 4,000$ samples
  - Obtain 100 additional random samples for validation
  - Quantify percentage error, $100 * |\hat{y}_i - y_i|/y_i$
- **Model Variants**
  - **Baseline (B):** Model non-transformed response
  - **Variance Stabilized (S):** Model square-root of response
  - **Regional (S+R):** For each query, reformulate model with samples most similarly configured to query

$$d = \left[\sum_{i=1}^{p} \left(\frac{a_i - b_i}{a_i} \right)^2 \right]^{1/2}$$

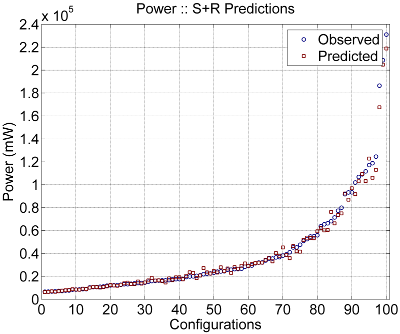
  - **Application-Specific (S+A):** Fix sample benchmarks

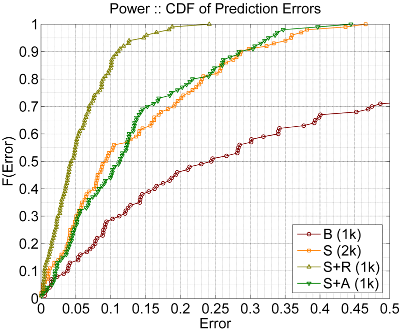
Performance Prediction





Power Prediction





Performance-Power Comparison

- **Performance Accuracy**
  - 7.4% median error for S+A model
  - S+A reduces performance variance across applications
  - S+R ineffective since application is primary determinant of performance
- **Power Accuracy**
  - 4.3% median error for S+R model
  - S+R reduces power variance across configurations
  - S+A ineffective since resource sizings are primary determinants of power

Outline

Summary

- **Simulation Challenges**
  - Limited design space studies due to simulation costs
  - Existing frameworks reduce per simulation costs only
- **Regression Models**
  - Sampling :: 1K of $\approx$1B configurations UAR
  - Specification :: correlation analyses
  - Refinement :: stabilizing transformations
- **Model Evaluation**
  - 7.4%, 4.3% median errors for performance, power
  - S+A, S+R more effective for performance, power

Future Directions

- **Model Applications**
  - Demonstrate applicability to prior studies
  - Models enable more aggressive studies
  - Construct a CMP simulation framework
- **Model Improvements**
  - Techniques, transformations to further reduce error, bias
- **Survey Approaches in Statistical Inference**
  - Compare regression modeling with machine learning

Publications

www.deas.harvard.edu/∼bclee

- B.C. Lee and D.M. Brooks.
  Statistically rigorous regression modeling for the microprocessor design space.
  ISCA-33: Workshop on Modeling, Benchmarking, and Simulation, June 2006.
- B.C. Lee and D.M. Brooks.
  Accurate, efficient regression modeling for microarchitectural performance, power prediction.
  ASPLOS-XII: International Conference on Architectural Support for Programming Languages and Operating Systems, Oct 2006. (To Appear)

References I

Y. Li, B.C. Lee, D. Brooks, Z. Hu, K. Skadron.
CMP design space exploration subject to physical constraints.
HPCA-12: International Symposium on High-Performance Computer Architecture, Feb 2006.

L. Eeckhout, S. Nussbaum, J. Smith, and K. DeBosschere.
Statistical simulation: Adding efficiency to the computer designer's toolbox.
IEEE Micro, Sept/Oct 2003.

R. Liu and K. Asanovic.
Accelerating architectural exploration using canonical instruction segments.
In *International Symposium on Performance Analysis of Systems and Software*, Austin, Texas, March 2006.

T. Sherwood, E. Perelman, G. Hamerly, and B. Calder.
Automatically characterizing large scale program behavior.
ASPLOX-X: Architectural Support for Programming Languages and Operating Systems, October 2002.

B.C. Lee and D.M. Brooks.
Effects of pipeline complexity on SMT/CMP power-performance efficiency.
ISCA-32: Workshop on Complexity Effective Design, June 2005.

References II

C. Stone.
Comment: Generalized additive models.
Statistical Science, 1986.

F. Harrell.
Regression modeling strategies.
Springer, New York, NY, 2001.

J. Yi, D. Lilja, and D. Hawkins.
Improving computer architecture simulation methodology by adding statistical rigor.
IEEE Computer, Nov 2005.

P. Joseph, K. Vaswani, and M. J. Thazhuthaveetil.
Construction and use of linear regression models for processor performance analysis.
In *Proceedings of the 12th Symposium on High Performance Computer Architecture*, Austin, Texas, February 2006.

S. Nussbaum and J. Smith.
Modeling superscalar processors via statistical simulation.
In *PACT2001: International Conference on Parallel Architectures and Compilation Techniques*, Barcelona, Sept 2001.

Assessing Fit

- **Multiple Correlation Statistic**
  - R^2 is fraction of response variance captured by predictors
  - Large R^2 suggests better fit to observed data
  - $R^2 \to 1$ suggests over-fitting (less likely if $p < n/20$)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \frac{1}{n}\sum_{i=1}^{n} y_i)^2}$$

- **Residual Distribution Assumptions**
  - Residuals are normally distributed, $e_i \sim N(0, \sigma^2)$
  - No correlation between residuals and response, predictors
  - Validate by scatterplots and quantile-quantile plots

$$\hat{e}_i = y_i - \hat{\beta}_0 - \sum_{j=0}^{p} \hat{\beta}_j x_{ij}$$

Predictor Non-Linearity I

- **Polynomial Transformations**
  - Undesirable peaks and valleys
  - Differing trends across regions
- **Linear Splines**
  - Piecewise linear regions separated by knots
  - Inadequate for complex, highly curved relationships
- **Restricted Cubic Splines**
  - Higher order polynomials provide better fits
  - Continuous at knots
  - Linear constraint on tails

Predictor Non-Linearity II

- **Location of Knots**
  - Location of knots less important than number of knots
  - Place knots at fixed predictor quantiles
- **Number of Knots**
  - Flexibility, risk of over-fitting increases with knot count
  - 5 knots or fewer are often sufficient [3]
  - 4 knots is a good compromise between flexibility, over-fitting
  - Fewer knots required for small data sets

[3] Stone [SS'86]

Significance Testing I

- **Approach**
  - Given two nested models, hypothesis H_0 states additional predictors in larger model have no response association
  - Test H_0 with F-statistics and p-values
- **Example**
  - Predictor interaction requires comparing nested models
  - Consider a model $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$.
  - Test significance of x_1 with null hypothesis $H_0 : \beta_1 = \beta_3 = 0$

Significance Testing II

- **F-Statistic**
  - Compare two nested models using their R^2 and F-statistic
  - R^2 is fraction of response variance captured by predictors

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \frac{1}{n}\sum_{i=1}^{n} y_i)^2}$$

  - F-statistic of two nested models follows F distribution

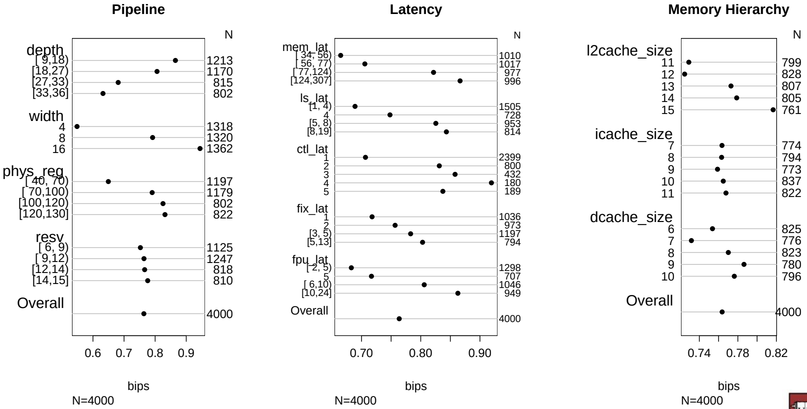
$$F_{k,n-p-1} = \frac{R^2 - R^2_*}{k} \times \frac{n-p-1}{1-R^2}$$

- **P-Values**
  - Probability F-statistic greater than or equal to observed value would occur under H_0
  - Small p-values cast doubt on H_0

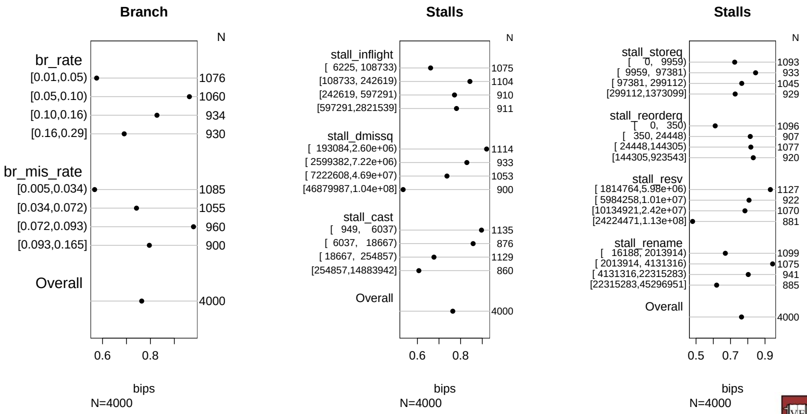
Treatment of Missing Data

- **Missing Completely at Random (MCAR)**
  - Treat unobserved design points as missing data
  - Sampling UAR ensures observations are MCAR
  - Data is missing for reasons unrelated to characteristics or responses of the configuration
- **Informative Missing**
  - Data is more likely missing if their responses are systematically higher or lower
  - "Missingness" is non-ignorable and must also be modeled
  - Sampling UAR avoids such modeling complications

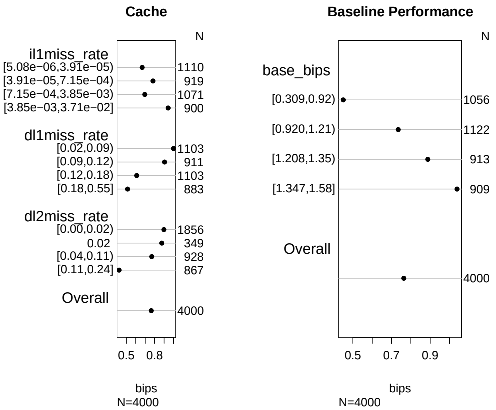
Performance Associations I



Performance Associations II



Performance Associations III



Significance Tests

- **Microarchitectural Predictors**
  - Majority of F-tests imply significance (p-values $< 2.2E - 16$)
  - Several predictors were less significant
    - Control latency (p-value $= 0.1247$)
    - Reservation station size (p-value $= 0.1239$)
    - L1 instruction cache size (p-value $= 0.02941$)
- **Application-Specific Predictors**
  - Majority of F-tests imply significance (p-values $< 2.2E - 16$)
  - Pipeline stalls classified by structure are less significant
    - Completion and reorder queue stalls (p-values > 0.4)

Related Work

- **Statistical Significance Ranking**
  - Yi :: Plackett-Burman, effect rankings
  - Joseph :: Stepwise regression, coefficient rankings
  - Bound parameter values to improve tractability
  - Require simulation for estimation
- **Synthetic Workloads**
  - Eeckhout :: Profile workloads to obtain synthetic traces
  - Nussbaum :: Superscalar and SMP simulation
  - Obtain distribution of instructions and data dependencies
  - Require simulation with smaller traces for estimation