Methods of Inference and Learning for Performance Modeling of Parallel Applications

Benjamin C. Lee
David M. Brooks

Engineering & Applied Sciences
Harvard University

Bronis R. de Supinski
Martin Schulz

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

Karan Singh
Sally A. McKee

Computer Systems Laboratory
Cornell University

Symposium on Principles and Practice of Parallel Programming
16 March 2007

[0] UCRL-PRES-228986

Outline

Outline

- Motivation & Background
  - Modeling Challenges
  - Empirical Modeling Paradigm
  - Applications
- Parameter Characterization
  - Measurement Sampling
  - Hierarchical Clustering
  - Association & Correlation
- Performance Modeling
  - Piecewise Polynomial Regression
  - Artificial Neural Networks
  - Model Evaluation
- Conclusion

Modeling Challenges

- **Increasing Complexity**
  - System :: memory architectures, interconnect topologies
  - Algorithm :: data decomposition, communication costs
  - Expensive combinatorial optimization
- **Limits of Analytical Modeling**
  - Simplifying assumptions mitigate complexity
  - Performance bounds, scalability trends

Empirical Modeling Paradigm

- **Comprehensively understand parameter space**
  - Specify large, high-resolution parameter space
  - Consider all parameters simultaneously
- **Selectively measure configurations**
  - Sample randomly from space for measurement
  - Decouple resolution of space and measurement
- **Efficiently leverage measured data with inference**
  - Reveal trends, trade-offs from sparse sampling
  - Enable predictions for metrics of interest

Semicoarsening Multigrid (SMG2000)

- Solve discretized 3-D differential equations
- Coarsens only z-dimension, relaxes xy planes

	Set	Parameters	Measure	Range	$\lvert S_i \rvert$
S_1	N_x	x-dim working set	grid points	10 - 510	501
S_2	N_y	y-dim		10 - 510	501
S_3	N_z	z-dim		10 - 510	501
S_4	P_x	x-dim processors	log_2(procs)	0 - 9	10
S_5	P_y	y-dim		0 - 9	10
S_6	P_z	z-dim		0 - 9	10

High-Performance Linpack (HPL)

- Solve dense linear system
- LU decomposition, backward substitution

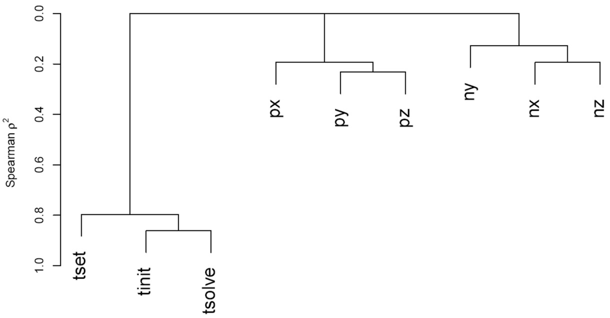
	Set	Parameters	Measure	Range	$\lvert S_i \rvert$
S_1	Matrix Size	N	sq. matrix dim	1000	1
S_2	Block Size	NB	sq. block dim	10 - 80	8
S_3	Processor Distribution	rows (P) columns (Q)	log_2(procs) log_2(procs)	0 - 9 9-P	10
S_4	Panel Factor	$PFACT$	algorithm	L,R,C	3
S_5	Recursive Factor	$RFACT$	algorithm	L,R,C	3
S_6	Recursive Base	$NBMIN$	block size	1 - 8	8
S_7	Recursive Sub-Panels	$NDIV$	sub-panels	2 - 4	3
S_8	Broadcast	$BCAST$	algorithm	1rg, 1rM, 2rg, 2rM, Lng, LnM	6

Outline

Measurement Sampling

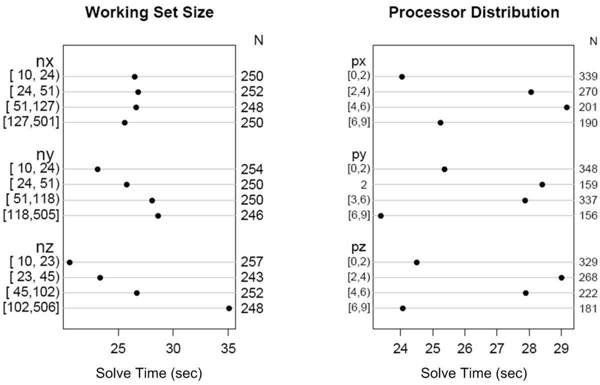
- **Uniform at Random**
  - Unbiased samples from full range of parameter values
- **Stratification**
  - Additional weight to samples with small measured values
  - Mitigates least squares bias toward large measured values
- **Regional Sampling**
  - Per-query models using samples most similar to query
  - Similarity quantified by Euclidean distances

Hierarchical Clustering

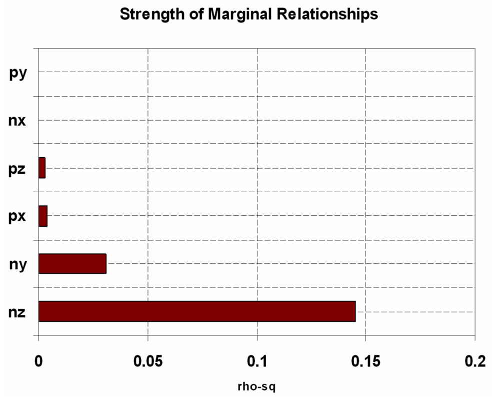


- Create a cluster for each parameter
- Merge most similar pair of clusters
- Repeat merge until single cluster obtained

Association Analysis



Correlation Analysis



Outline

Regression Theory

- **Statistical Inference**
  - Models approximate solutions to intractable problems
  - Requires initial data to train, formulate model
  - Leverages correlations from initial data for prediction
- **Regression Approaches**
  - Piecewise Polynomial Regression (R)
  - Artificial Neural Networks (SNNS)

Piecewise Polynomial Regression

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

$$f(y) = \beta_0 + \sum_{j=1}^{p} \beta_j g_j(x_j) + e$$

Predictor Interaction

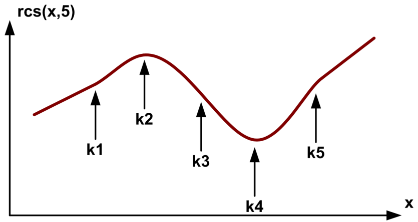
- **Modeling Interaction**
  - Suppose effects of predictors x_1, x_2 cannot be separated
  - Construct predictor $x_3 = x_1 x_2$

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

- **Example**
  - Let $x_1 = P_z$ (processor count), $x_2 = N_z$ (working set size)
  - Performance impact of working set size depends on processors in the same dimension

Predictor Non-Linearity

- **Restricted Cubic Splines**
  - Divide predictor domain into intervals separated by knots
  - Piecewise cubic polynomials joined at knots
  - Higher knot counts for more significant parameters



Construction and Prediction

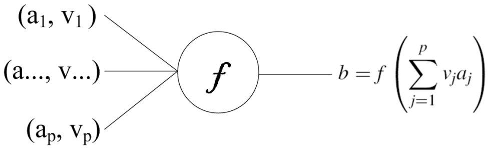
- **Least Squares**
  - Determines $\beta = \beta_0, \ldots, \beta_p$ to minimize $S(\beta)$
  - Solve system of $p+1$ partial derivatives $\delta S / \delta \beta$

$$
\begin{aligned}
S(\beta) &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\
\hat{y}_i &= \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}
\end{aligned}
$$

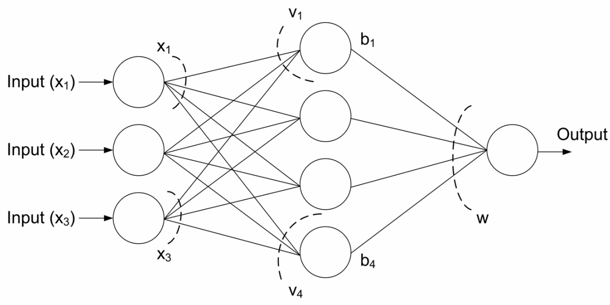
- **Prediction**
  - Weighted sum of predictors

Artificial Neural Networks

- **Notation**
  - Neuron Inputs :: $\vec{a} = a_1, \ldots, a_p$
  - Network Edges :: $\vec{v} = v_1, \ldots, v_p$
  - Neuron Output :: b
- **Neuron Model**



Network Model



$$\hat{y} = f\left(\sum_{j=1}^{H} w_j b_j\right) = f\left(\sum_{j=1}^{H} w_j\, f\left(\sum_{k=1}^{p} v_{j,k} x_k\right)\right)$$

Construction and Prediction

- **Gradient Descent**
  - Determine weights W to minimize $S(W)$
  - Iteratively update weights in direction of steepest descent

$$
\begin{aligned}
S(W) &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\
W_{(t)} &= W_{(t-1)} - \eta \, \nabla S(W_{(t-1)})
\end{aligned}
$$

- **Prediction**
  - Nested weighted sum of predictors

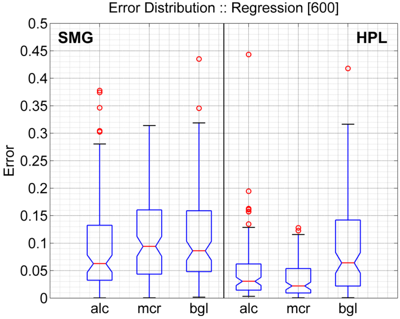
Comparison of Techniques

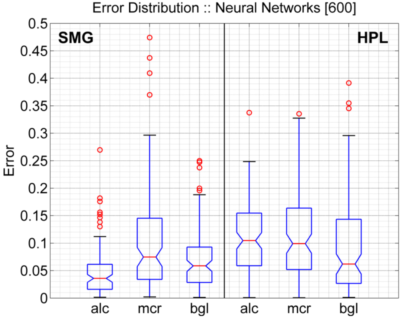
Regression	Neural Networks
Automation	
User provides model specification	Flexible model specification
Transparency	
Interactions, non-linearities exposed by specification	Network as black box, analysis through prediction
Computational Efficiency	
Linear solve $f\left(\beta_0 + \sum_{j=1}^{p} \beta_j g_j(x_j)\right)$	Gradient descent $f\left(\sum_{j=1}^{H} w_j f\left(\sum_{k=1}^{p} v_{j,k} x_k\right)\right)$

Evaluation Methodology

- **Framework**
  - Compare splines and neural networks
  - Formulate models with 600 random samples
  - Obtain 100 additional random samples for validation
- **Platforms**
  - IBM Blue Gene/L (bgl)
  - Intel Xeon (alc, mcr)
- **Applications**
  - Semicoarsening Multigrid (SMG)
  - High-Performance Linpack (HPL)

Predictive Accuracy





Outline

Conclusion

- **Empirical Modeling Paradigm**
  - Comprehensively understand parameter space
  - Selectively measure configurations
  - Efficiently leverage measured data with inference
- **Characterization and Inference**
  - Statistical significance analyses
  - Regression via splines and neural networks
  - Performance predictions with median error <10 percent
- **ISCA 2007 Tutorial**
  - Inference and Learning for Large Scale Microarchitectural Analysis