2nd Workshop on Energy Aware High Performance Computing
(ISC High Performance 2017 Workshop)

Combining Global Regression and Local Approximation in Server Power modeling

Xiaoming Du & Cong Li

Intel Corporation

Outline

- Server Power Modeling Using High-Level Resource Utilization Input
- Related Work
- Our Approach: Combining Global Regression & Local Approximation
- Experimental Results
- Conclusion & Future Work

Server Power Modeling

To Evaluate Energy Use in Green Clusters

- Important to understand server power consumption

One Category of Approach: Full-System Simulation

- Analytical models tying to low level architectural events
- Drawbacks: simulation speed & portability

Alternative

- Modeling power based on high level resource utilization input
- On-the-fly full-system power characterization in a non-intrusive manner

Server Power Modeling (cont.)

Server Power Model with High-Level Resource Utilization Input

- $P = f(\mathbf{u})$
- Trained/calibrated with $\{(\mathbf{u}_1, P_1), \ldots, (\mathbf{u}_n, P_n)\}$

Usages

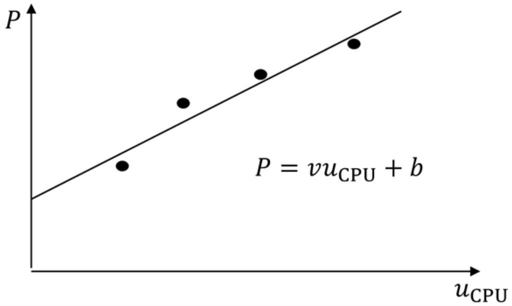
- Monitor power for servers without power sensor instrumentation
- What-if power analysis given hypothesized resource utilization
- Power aware scheduling

Server power models are useful in energy efficiency evaluation

Related Work: Linear Power Model

Linear Regression

- $P = \mathbf{v} \cdot \mathbf{u} + b$: linear change of power w.r.t. utilization
  - One-dimensional example



Characteristics

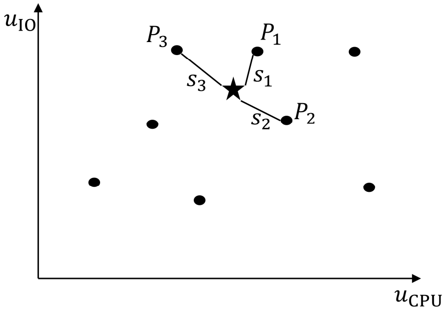
- Coarse-grained, trained with minimum data
- Not comprehensive enough in capturing subtle nonlinearity

Linear models are not good enough in a complicated context

Related Work: k-Nearest Neighbor Regression

k-NN Regression

- $P = \frac{\sum_{i=1}^{k}(P_{N_i}/\|\mathbf{u}_{N_i} - \mathbf{u}\|)}{\sum_{i=1}^{k}(1/\|\mathbf{u}_{N_i} - \mathbf{u}\|)}$: approximated with a set of neighbors $\{\mathbf{u}_{N_1}, \dots, \mathbf{u}_{N_k}\}$
  - Two-dimensional 3-NN example



$$P = \frac{\frac{P_1}{s_1} + \frac{P_2}{s_2} + \frac{P_3}{s_3}}{\frac{1}{s_1} + \frac{1}{s_2} + \frac{1}{s_3}}$$

Characteristics

- Superior in approximating local nonlinearity with near neighbors
- Not capable to generalize well in lack of near neighbors

kNN models are not good enough in sparse & unbalanced datasets

New Model

Combining Global Regression & Local Approximation

$$P = \mathbf{v} \cdot \mathbf{u} + b + \sum_{i=1}^{n} w_i \|\mathbf{u}_i - \mathbf{u}\|$$

Sub-model of global regression: retain robustness with the coarse-grained generalization capability

New Model

Combining Global Regression & Local Approximation

$$P = \mathbf{v} \cdot \mathbf{u} + b + \sum_{i=1}^{n} w_i \|\mathbf{u}_i - \mathbf{u}\|$$

Sub-model of spatial interpolation for local approximation: compensate global regression by capturing subtle nonlinearity

New Model

Combining Global Regression & Local Approximation

$$P = \mathbf{v} \cdot \mathbf{u} + b + \sum_{i=1}^{n} w_i \|\mathbf{u}_i - \mathbf{u}\|$$

The new model is expected to enjoy the both advantages

Model Training

Optimization of Joint Objective of Model Errors & Model Complexity

$$\min_{\mathbf{v},b,\mathbf{w}} \sum_{i=1}^{n} \left(\mathbf{v}\cdot\mathbf{u}_i + b + \sum_{j=1}^{n} w_j \left\|\mathbf{u}_j - \mathbf{u}_i\right\|\right)^2 + \beta\left\|(\mathbf{v}, b)\right\|^2 + \gamma\left\|\mathbf{w}\right\|^2$$

Model error on training data

Model Training

Optimization of Joint Objective of Model Errors & Model Complexity

$$\min_{\mathbf{v},b,\mathbf{w}} \sum_{i=1}^{n} (\mathbf{v}\cdot\mathbf{u}_i + b + \sum_{j=1}^{n} w_j \|\mathbf{u}_j - \mathbf{u}_i\|)^2 + \beta\|(\mathbf{v},b)\|^2 + \gamma\|\mathbf{w}\|^2$$

Model complexity measured in L_2 regularization

Model Training

Optimization of Joint Objective of Model Errors & Model Complexity

$$\min_{\mathbf{v},b,\mathbf{w}} \sum_{i=1}^{n} (\mathbf{v} \cdot \mathbf{u}_i + b + \sum_{j=1}^{n} w_j \|\mathbf{u}_j - \mathbf{u}_i\|)^2 + \beta \|(\mathbf{v}, b)\|^2 + \gamma \|\mathbf{w}\|^2$$

Optimization problem solved with quadratic programming

Model Characteristics

Impact of Close Neighbors

- $\mathbf{u}$ close to $\mathbf{u}_i \rightarrow f(\mathbf{u})$ close to P_i
  - Similar to k-nearest neighbor regression

Unbalanced Training Data

- Artificial unbalanced training set duplicating $\mathbf{u}_n$ for m times
- Model trained in the original optimization problem → close-to-optimal solution of new optimization problem

The new model possesses desired characteristics

Experiments – Configuration

Server

- Intel Romley generation server: 2 x 8 cores x 2 threads, 32GB RAM, 500GB hard drive
- Windows Server 2008 R2 & Ubuntu Server 14

Data Collection for Every 10s

- Utilization data: CPU utilization, last level cache misses, disk I/O transfers
- Power data: input power to PSU aggregated by Node Manager

Experiments – Benchmarks

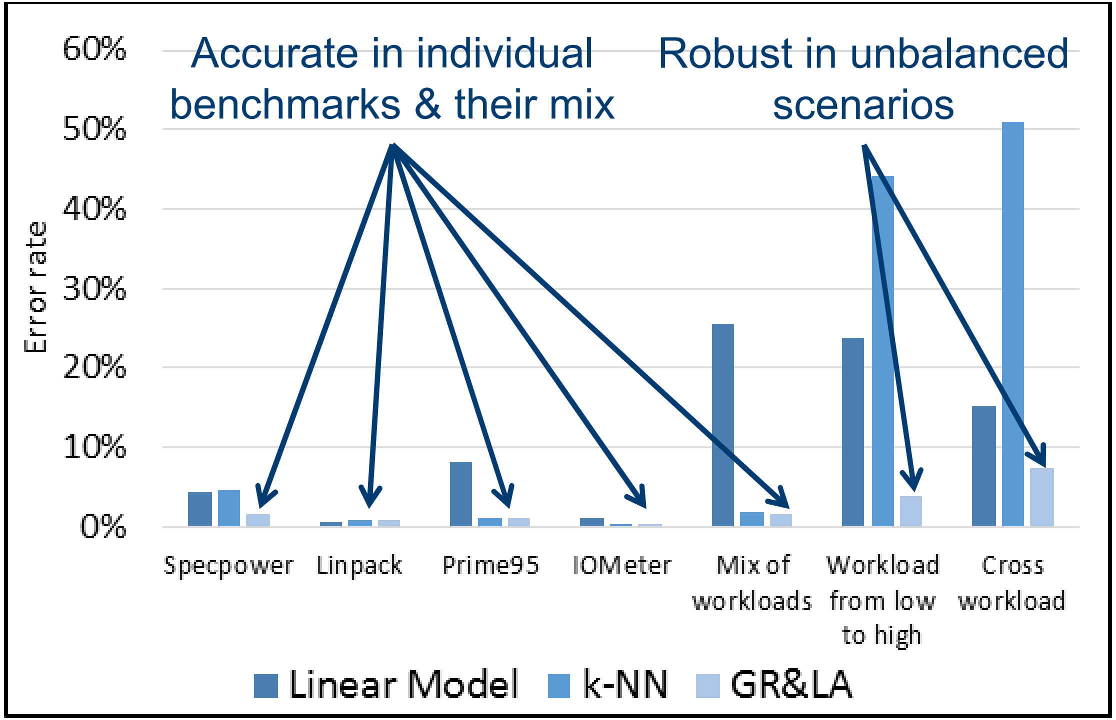
Individual Benchmarks – Random Split

- Prime95 (intensity varied)
- Linpack
- SPECPower
- IOMeter

Derived Datasets

- Mix of workloads (4 benchmarks + idle + FIRESTARTER) – random split
- SPECPower – chronological split
- SPECPower for training & Linpack for testing

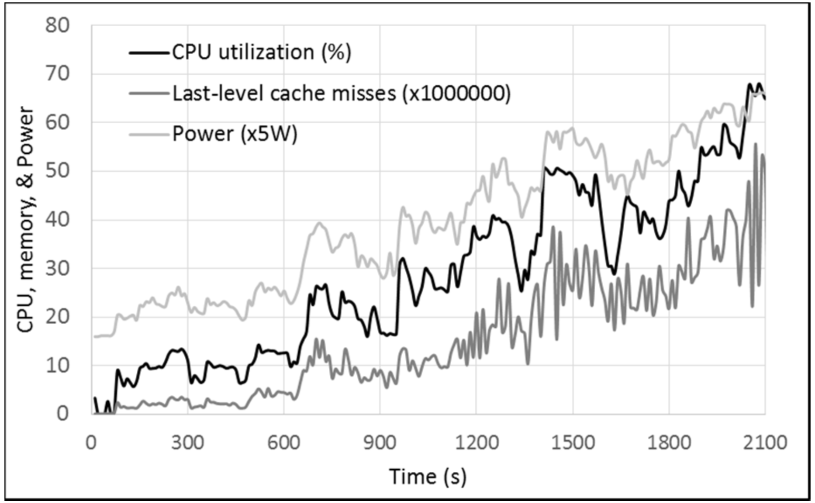
Experiments – Benchmarks (cont.)



Experiments – Real World Workloads

Mix of Real World Workloads

- Datacenter management solution running in background
- Batches of machine learning jobs
  - Java-based managed code & native code
  - Lengths varied



Training/Testing Split

- Random split
- Chronological split

Experiments – Real World Workloads (cont.)

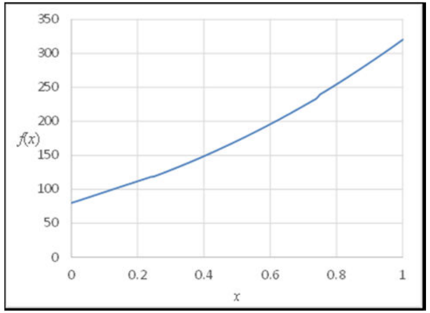
Split	Linear model	k-NN	GR&LA
Random split	5.53%	8.17%	***4.76%***
Chronological split	8.62%	46.43%	***6.55%***

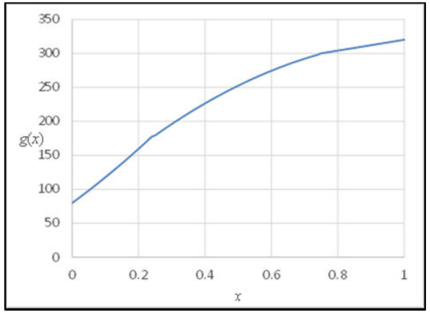
The new model works well on benchmarks & real world workloads

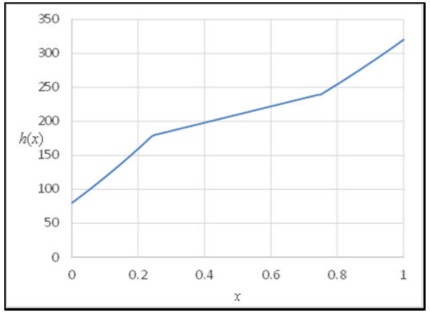
Experiments – Modeling Synthetic Functions

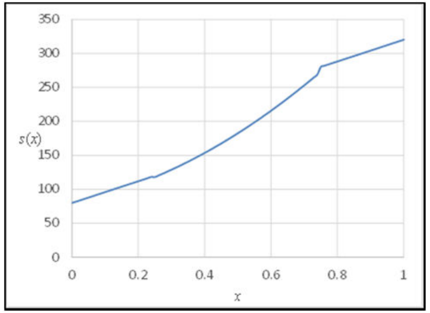
Four One-Dimensional Piecewise Functions

- Close to linear but with different nonlinearities at different regions









- Simulate power model peak at 320W & idle at 80W
- Gaussian noise in training but no noise in testing

Experiments – Modeling Synthetic Functions (cont.)

Noise: $\mathcal{N}(0, 0.05^2)$

Method	$f(x)$	$g(x)$	$h(x)$	$s(x)$
Linear model	5.12%	7.84%	6.23%	5.86%
k-NN	4.01%	4.04%	4.29%	4.03%
GR&LA	0.91%	0.76%	0.77%	0.88%

Noise: $\mathcal{N}(0, 0.1^2)$

Method	$f(x)$	$g(x)$	$h(x)$	$s(x)$
Linear model	5.35%	8.12%	6.13%	5.80%
k-NN	8.12%	8.19%	8.29%	7.69%
GR&LA	1.30%	1.57%	1.51%	1.61%

Noise: $\mathcal{N}(0, 0.2^2)$

Method	$f(x)$	$g(x)$	$h(x)$	$s(x)$
Linear model	4.99%	7.74%	6.36%	5.60%
k-NN	16.10%	15.65%	15.79%	16.29%
GR&LA	2.97%	2.79%	3.36%	2.84%

The new model is both accurate & robust against noises

Conclusion & Future Work

New Approach to Model Server Power

- Combining global regression & local approximation
  - Local approximation: capture subtle nonlinearity
  - Global regression: retain robustness

Future Work

- Fine-grained modeling with more counters from sub-systems
- Extending to non-server domains
- Incorporating in infrastructure management systems

The End

Thanks