Construction and Use of Linear Regression Models for Processor Performance Analysis

P. J.Joseph Kapil Vaswani Matthew J. Thazhuthaveetil

Department of Computer Science & Automation
Indian Institute of Science, Bangalore, India.
E-mail: {peejay,kapil,mjt}@csa.iisc.ernet.in

Abstract

Processor architects have a challenging task of evaluating a large design space consisting of several interacting parameters and optimizations. In order to assist architects in making crucial design decisions, we build linear regression models that relate processor performance to micro-architectural parameters, using simulation based experiments. We obtain good approximate models using an iterative process in which Akaike's information criteria is used to extract a good linear model from a small set of simulations, and limited further simulation is guided by the model using D-optimal experimental designs. The iterative process is repeated until desired error bounds are achieved. We used this procedure to establish the relationship of the CPI performance response to 26 key micro-architectural parameters using a detailed cycle-by-cycle superscalar processor simulator. The resulting models provide a significance ordering on all micro-architectural parameters and their interactions, and explain the performance variations of micro-architectural techniques.

1. Introduction

Modern processors are evolving at a rapid pace and in the quest for better performance architects introduce sophisticated micro-architectural enhancements in each new generation of processors. However, the increasing complexity of modern architectures has direct consequence on the process of designing processors. To arrive at optimal design points, architects are expected to evaluate a large design space consisting of several micro-architectural parameters such as cache sizes and associativities, queue sizes, branch predictor configuration, pipeline depth etc., each with a wide range of potential settings. Complex interactions between these parameters make it hard to gain an intuitive understanding of their impact on performance.

Designers in many disciplines of science and engineering deal with design complexity by building abstract models of the system that relate input parameters to the response. The models help designers gain a better understanding of the system and answer several key questions; for instance, which input parameters have the largest impact on response? How does a particular parameter interact with the others? What is the expected benefit of an enhancement? Although there have been several attempts to build models for processor performance over the past years [6, 9], they rely on prior knowledge about the significant parameters, fail to model the design space in sufficient detail, and their validity across a larger design space is unknown.

The aim of our research is to develop empirical models for processors that characterize the relationship between processor response and micro-architectural parameters. As a first step in building such models we quantify the *significance* of micro-architectural parameters and their interactions. Quantifying the interactions between micro-architectural parameters is important, and is best illustrated through a simple experiment. We measured the improvement in average instructions issued per cycle (IPC) due to out-of-order issue over in-order issue for different L1 data cache configurations. Improvements in IPC for the SPEC *twolf* benchmark are plotted in Figure 1. The impact of out-of-order issue varies with data cache size, and the variation depends on cache latency. Such significant interactions need to be included in the model. In this paper, we show that precise estimates of the significance of all parameters and interactions can be obtained by building *linear regression models* using simulation-based experiments. We show how the parameters of the regression model, which reflect the significance of the corresponding terms, can be empirically computed without any prior knowledge or understanding of processor dynamics. Since these significant factors have a large impact on performance and are usually small in number, they are ideal candidates for further analysis.

In this paper, we draw from past research in the field of design of experiments and linear model construction and propose an iterative process for constructing accurate regression models of processor performance consisting of all significant main effects and interaction terms using a rea-

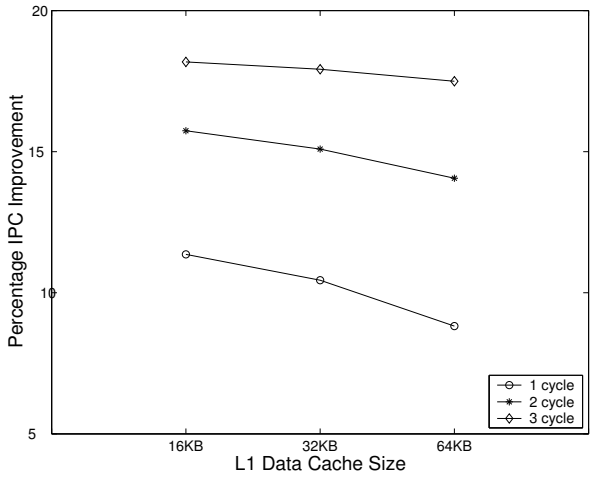


Figure 1. IPC performance improvement of out-of-order issue over in-order issue.

sonable number of simulations. We use this procedure to build a linear model relating superscalar processor performance to 26 key micro-architectural parameters. We also show how parameters that were originally not a part of the experiment can be added to the linear model using few additional simulations.

The rest of the paper is organized as follows. In Section 2, we discuss the basic concepts of linear regression models. We describe our iterative procedure for building the linear model in Section 3, and the experimental framework in Section 4. Section 5 presents the results of model construction. We present an overview of existing processor modeling techniques in Section 6 and conclude in Section 7 with a discussion of future work.

2. Linear Regression Models

A regression model is a compact mathematical representation of the relationship between the response variable and the input parameters in a given design space [8]. Linear regression models are widely used to obtain estimates of parameter significance as well as predictions of the response variable at arbitrary points in the design space. One of the simpler forms of such models is

$$y = \beta_0 + \sum_{i=1}^{m} \beta_i x_i + \epsilon \qquad (1)$$

where y is the dependent or response variable, $\{x_i | 1 \leq i \leq m\}$ are the independent or *regressor* variables and ϵ is the *residual* - the error due to lack of fit. β_0 is interpreted as the intercept of the response surface with the y-axis and $\{\beta_i | 1 \leq i \leq m\}$ are known as the *partial regression coefficients*. The co-efficient values represent the expected change in the response y per unit change in x_i and indicate the relative significance of the corresponding terms.

It is often the case that the regressor variables interact i.e. the effect of a change in x_i on y depends on the value of x_j. In such cases, the simple model in Eq. 1 is not sufficient. It is necessary to introduce terms that explicitly model *two-factor interactions* as shown below.

$$y = \beta_0 + \sum_{i=1}^{m} \beta_i x_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \beta_{i,j} x_i x_j + \epsilon \qquad (2)$$

Eq. 3 represents a complete model that include *three-factor*, *four-factor* and all higher order interactions. There are 2^m terms in this model and an equal number of unknown regression co-efficients.

$$\begin{aligned} y = \beta_0 + \sum_{i=1}^{m} \beta_i x_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \beta_{i,j} x_i x_j + \\ \sum_{i=1}^{m} \sum_{j=i+1}^{m} \sum_{k=j+1}^{m} \beta_{i,j,k} x_i x_j x_k + \dots \\ \cdots + \beta_{1,2,\dots m} x_1 x_2 \dots x_m \end{aligned} \qquad (3)$$

The linear regression models we develop in this paper can be represented as a sum of k terms from this complete linear model, expressed in a generic form as

$$y = \beta_0 + \beta_1 x_{i^1} + \beta_2 x_{i^2} \cdots + \beta_{k-1} x_{i^{k-1}} + \epsilon \qquad (4)$$

where each x_{ij} is a distinct term from the complete model, and can be single factor, two factor, three factor or of any higher order. The collection of terms chosen for a given linear model will be referred to as the *model terms*.

In matrix terms, Eq. 4 can be written as

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \qquad (5)$$

where β is the vector of regression coefficients and X is the *model matrix*. The model matrix has columns corresponding to the regressor variables $x_1, x_2, \dots, x_m$, columns for interaction terms of any order, and a column of ones defining the intercept.

Our goal in this paper is to accurately estimate all significant micro-architectural parameters and interactions affecting processor performance. We achieve this by performing simulation-based experiments in which the regressor variables are set to different values and the resulting performance metric is fitted as per Eq. 4, minimizing the number of terms k and the residual error ϵ simultaneously. As a by-product, we obtain precise estimates of the partial regression co-efficients and hence estimates and an ordering of the significant factors and interactions affecting processor performance.

There are a few guidelines that an architect must keep in mind while planning and designing the simulation experiments. The guidelines help in the selection of the response

variable, the set of factors to be included in the experiment, their ranges and the levels at which each factor is varied during the experiments.

Factor selection: A designer must initially identify a set of factors that can potentially influence the system response. For our experiments, we assume no prior knowledge of factor effects and select factors conservatively i.e. factors are included in the experiments even though their impact on performance may not be significant.

Factor ranges and levels: The choice of factor ranges and levels is primarily governed by technological constraints and the objective of the experiment. If the primary objective is to determine significant coefficients in a linear model, as is the case with our study, wide factor ranges are preferred. We choose a range consisting of values just lower and higher than potential factor settings under current technology. During our experiments, each factor is varied at two levels (encoded as -1 and 1) corresponding to the low and high value of its range.

Response variable and output transformations: Our simulator reports simulated processor performance measured in IPC. However, to build a linear model, it is often beneficial to use a transformation of the response variable instead of the response variable itself [8]. Such transformations might result in response surfaces that are more linear and easier to fit. We evaluate a family of such transformations in the context of processor modeling in the following section.

2.1. A Case Study on Building a Linear Model

We illustrate our approach by building a linear regression model that relates the IPC of SPEC CPU2000 integer benchmarks to six micro-architectural parameters – namely the pipeline depth, reorder buffer size, issue queue size, L2 cache size, out-of-order capability, and the memory configuration. We limit the number of factors to six so that all regression co-efficients in the complete linear model can be exactly determined using a reasonable number of simulations. The factor ranges were chosen based on experimental design guidelines. During the experiment, simulations were conducted for the 2^6 possible combinations of factor levels and the regression co-efficients were obtained by solving Eq. 3 for the measured IPC. The experiment led to the following observations.

Output transformations: We measured the effect of output transformations on the accuracy of linear models. We considered a family of transformations $y^t, t = -2, -1, -0.5, 0.5, 1, 2$ and $log(y)$ as potential candidates. For each transformation, we computed the regression coefficients and sorted them in decreasing order of magnitude. We then built linear models incorporating the first k terms, for different values of k and used the models to predict the IPC of 64 processor configurations mentioned above. We then computed the residuals by subtracting the actual IPC from the predicted IPC and used the maximum residual value as an indicator of the accuracy of the model.

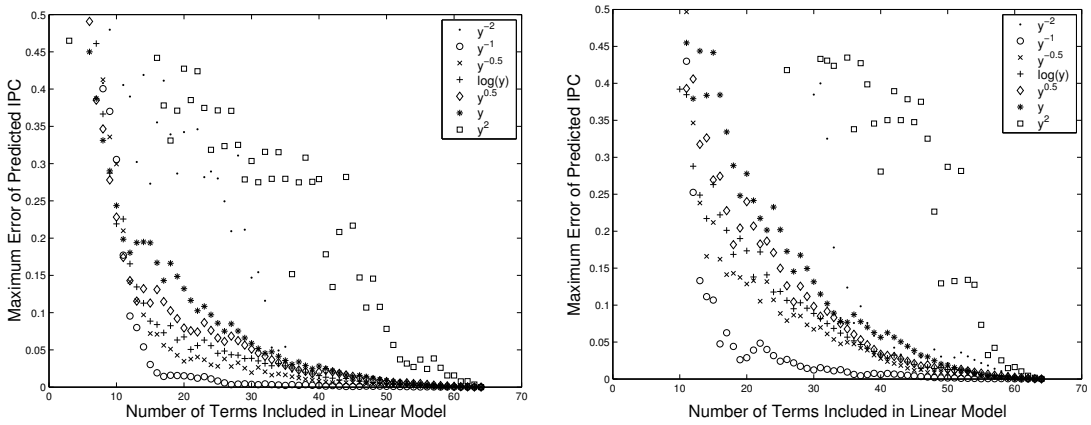
Terms	
intercept	1.230
pipe_depth	-0.566
ROB_size	-0.480
pipe_depth*ROB_size	0.378
IQ_size	-0.347
ROB_size*IQ_size	0.289
pipe_depth*IQ_size	0.274
pipe_depth*ROB_size*IQ_size	-0.219
mem_config	-0.037
L2_size	-0.033
issue_order	-0.026
L2_size*mem_config	0.023
ROB_size*issue_order	-0.015
pipe_depth*issue_order	-0.009
IQ_size*issue_order	-0.007
pipe_depth*ROB_size*IQ_size*L2_size	0.006

Table 1. The most significant terms from the 6-factor model for the weighted mean CPI of benchmarks. * denotes interaction.

Figure 2 plots the maximum residual value against the number of terms incorporated in the model. We observe that for all output transformations, the maximum residual reduces to acceptable levels even when many terms have been excluded from the models. Further, the inverse transformation (y^{-1}) results in the lowest maximum residuals. We observe similar behavior for all benchmarks. Hence, using the inverse transformation allows the construction of accurate linear models with the least number of terms. Note that the inverse transformation of IPC results in the CPI metric. CPI is an intuitive metric expressible as a dot product of event frequencies and event penalties within a processor [2] and it is most amenable for linear model construction. We use CPI as the performance metric in the rest of this paper.

Sparsity of effects: Our experiment reveals that processors exhibit the principle of *sparsity of effects* — system response is largely governed by a few main factors and low order interactions and the influence of higher order interactions on response is marginal. As a consequence, terms corresponding to higher order interactions can be excluded from the model. Table 1 lists the 16 terms, out of the 64 terms in our experiment, that we identified for inclusion in a simplified model of the processor's CPI performance. This observed sparsity of significant effects has a large bearing on the feasibility of our simulation-based model building process. We discuss this further in section 2.2.

Significance of factors: Table 1 shows the most significant factors and interactions in our regression model. Some of the two-factor interaction terms are also critical, as is the three-factor interaction between pipeline depth, ROB size and issue queue size. These observations highlight the im-



(a) gcc (b) vortex

Figure 2. Number of terms required in linear model under different data transformations.

portance of incorporating second and third order interactions in models.

2.2. Linear Models Incorporating All Factors

Our aim is to build a linear model involving all processor parameters. For this purpose it is infeasible to experiment with all combinations of factor levels as was done in the previous section. However, sparsity of effects makes it possible to have accurate and compact linear models of the type specified in Eq. 4 incorporating only significant effects. Coefficients of the k most significant parameters can usually be estimated from processor's response at n different factor level combinations using least square fitting, where $n > k$ and $n << 2^m$. The least square estimates of the k-dimensional coefficient vector $\beta = (\beta_0, \beta_1.., \beta_{k-1})$ in Eq. 5 is

$$\hat{\beta} = (\mathbf{X'X})^{-\mathbf{1}}\mathbf{X'y} \tag{6}$$

where X is the model matrix for the experimented factor settings. The variance of the error term in Eq. 5 can be estimated as

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-k} \tag{7}$$

where $\hat{y} = X\hat{\beta}$ is the fitted response. This least square fitting method provides useful approximations with simulations reduced close to the number of significant effects.

The coefficients computed as above are only estimates of the actual linear model coefficients. However, we can compute a bound on this error. Under the simplifying assumption that errors due to non-incorporated terms are normally distributed, a $100(1-\alpha)$ percent confidence error bound for β_j is

$$|\beta_j - \hat{\beta}_j| \leqslant t_{\alpha/2,n-k}\sqrt{C_{jj}\hat{\sigma}^2} \tag{8}$$

where $t_{\alpha/2,n-k}$ is the upper $\alpha/2$ percentage point of the t distribution with $n-k$ degrees of freedom, and C_{jj} is the $(j,j)^{th}$ entry of $(X'X)^{-1}$ matrix [8]. It follows that the accuracy of our estimates depend on:

- *Error Degree of Freedom:.*

  Increasing $n-k$, the degree of freedom for the error term, reduces $t_{\alpha/2,n-k}$ in Eq. 8. However, this reduction decreases as $n-k$ is increased beyond a point.

- *Experimental Design:*

  The specific set of factor level combinations used for the experiment determine the C_{jj} term in Eq. 8. Hence appropriate experimental designs are critical to achieving good accuracy. We chose D-optimal experimental designs[8], which maximize det(X'X) for a set of model terms, for our model construction process[5].

- *Error Variance:*

  Minimization of the error variance, $\hat{\sigma}^2$, in Eq. 8 requires all significant terms to be in the model. However, since the significant terms are not known *a priori* they need to be identified from experimental data. We next describe the procedure we use to identify and include all significant effects.

3. Procedure for Model Construction

Our aim in model construction is to obtain accurate estimates of all significant coefficients with minimum number of simulations. Since the significant effects are not known *a priori*, this information has to be extracted from experimental data. However, designing the best experimental strategy requires knowledge of significant terms; D-optimal designs are optimized for an identified set of model terms. Hence,

in order to obtain the best experimental designs having minimal simulations we use an iterative procedure where initial small D-optimal designs are used to learn significant effects, and this information is used to guide further simulation using augmented D-optimal designs, until an adequate model is achieved. Each of the required steps and the complete iterative procedure are described in the following subsections.

3.1. Obtaining the Best Model

Given an experimental design and simulation results we have to determine the best model that fits the data well. We use Akaike's Information Criteria (AIC) [11] to select a model that fits well and has a minimum number of parameters, resulting in the most significant effects being included in the model, and thus reducing the model over-fitting problem. In our procedure we use the corrected version of AIC (AIC_c) developed for small experimental samples [4] since it allows us to keep the simulation count low. AIC_c for linear regression models can be written as

$$AIC_c = nlog(\hat{\sigma}^2)+2k+\frac{2(k+1)(k+2)}{(n-k)}+constant \quad (9)$$

where n is the number of simulations, k is the number of terms in the linear model, and $\hat{\sigma}^2$ is the error variance. This measure accounts for model accuracy using error variance, and model simplicity using the count of model terms. Amongst several models possible for fixed experimental data, the best model has the lowest AIC_c [11].

Arriving at the best model involves searching for the set of model terms producing the lowest AIC_c. Since an exhaustive search of all model combinations is computationally expensive we used a procedure which stepwise refines an initial model. This procedure is based on our observation that the more significant higher order terms are typically composed of the significant lower order terms and is detailed in [5].

3.2. Determining Model Adequacy

We use the maximum error of estimated coefficients as the main measure of model adequacy, since our aim is to get accurate estimates of all significant effects. This error bound can be obtained from Eq. 8 as $t_{\alpha/2,n-k}\sqrt{\hat{\sigma}^2 maxC_{jj}}$, where $maxC_{jj}$ is the maximum C_{jj} value of the model. We follow the iterative model construction scheme until a prescribed maximum bound on error is achieved. We also check for violations of the basic assumptions in estimating regression coefficients and their bounds by examining the *residuals*.

3.3. The Iterative Procedure

Our model construction procedure takes as inputs m (the number of experimented factors), θ (the prescribed maximum error), and α (the confidence level required on the error), and outputs Error Bounded Linear Models (EBLMs). We describe it below.

1. Design an initial D-optimal experimental design S with $2m$ runs for an initial linear model having all main effects. Obtain the best linear model L.
2. Measure the error variance $\hat{\sigma}^2$ of model L, and the maximum error in estimated coefficients. Stop and finalize the model if the error is less than θ, and residual plots are free from gross deviations.
3. Augment the experimental design S with additional D-optimal experiments ensuring the following: (i) the new design is D-optimal for a model containing union of terms in L and the main effect terms, (ii) there are sufficient number of additional experiments such that the maximum C_{jj} value is reduced by half.
4. Obtain the best linear model L for the new augmented design S. Go to step 2.

The above procedure can be used to obtain linear models at any specified level of accuracy. We implemented this procedure as a MATLAB script which takes the required inputs and completed simulation experimental data, and provides either the accurate coefficient estimates or a prescription for further experimentation.

4. Experimental Framework

4.1. Processor Simulator and Benchmarks

We developed and validated a detailed superscalar processor simulator for use in our experimentation. Our simulation framework - FAFSIM - models pipelined, multiple-issue, dynamically scheduled, speculative execution processors. It models all the performance critical micro-architectural events and structures in superscalar processors and is detailed in [5]. The pipeline, caches, branch direction and target predictors, various micro-architectural queues, functional units, DRAM device timing, queuing at the memory controller, and contention for the memory bus are all modeled. We verified the functionality of each component, and in addition validated the summary statistics against another similarly configured verified simulator, *alphasim* [1]. This validation was done for several design points to verify the simulator's accuracy across the design space.

We used our simulator to run all SPEC CPU2000 integer benchmarks using the *lgred* data set in MinneSPEC [7] reduced data sets. This was done using traces generated with

Pipeline Depth	pipe_depth
Reorder Buffer Size	ROB_size
Issue Queue Size	IQ_size
L2 Cache Size	L2_size
L2 Cache Associativity	L2_assoc
L2 Cache Block Size	L2_bsize
L2 Cache Latency	L2_lat
Instruction Cache Size	il1_size
Instruction Cache Associativity	il1_assoc
Instruction Cache Block Size	il1_bsize
Instruction Cache Latency	il1_lat
Data Cache Size	dl1_size
Data Cache Associativity	dl1_assoc
Data Cache Block Size	dl1_bsize
Data Cache Latency	dl1_lat
FTB entries	ftb_ent
FTB associativity	ftb_assoc
Operation Latency	op_lat
Issue Order	issue_order
Load-Store Queue Size	LSQ_size
Number of Functional Units	num_units
DRAM Memory Configuration	mem_config
Predictor Type	pred_type
Predictor Size	pred_size
Processor Width	width
Return Address Stack Size	RAS_size

Table 2. Micro-architectural parameters.

Parameter	Low Value	High Value
pipe_depth	24	7
ROB_size	24	128
IQ_size	(1/4)* x_2	x_2
L2_size	256KB	8MB
L2_assoc	1	8
L2_bsize	64	256
L2_lat	20	5
il1_size	8KB	128KB
il1_assoc	1	8
il1_bsize	16	64
il1_lat	2	1
dl1_size	8KB	128KB
dl1_assoc	1	8
dl1_bsize	16	64
dl1_lat	4	1
ftb_ent	128	8192
ftb_assoc	1	8
issue_order	In-order	Out-of-order
LSQ_size	(1/4)*x_2	x_2
pred_type	*gshare*	*perceptron*
pred_size	2KB	16KB
width	4	8
RAS_size	4	64

Table 3. Parameter ranges.

IBM PowerPC executables, compiled with xlc compiler applying the -O3 option.

4.2. Micro-architectural Parameters and Ranges

We experimented with the 26 key micro-architectural parameters listed in Table 2. The range of parameters are chosen to include the complete range of settings possible under current technology, and is listed in Table 3. Some parameters like issue order have only two potential settings and the low and high values correspond to these. Some of the parameters - operation latency, number of functional units, and the DRAM configuration - combine several parameters for ease of experimentation and their settings are reported in Tables 4, 5 and 6.

5. Results

5.1. Results of Model Construction

Table 7 presents the most significant terms in the constructed error bounded linear models. The significance ordering in the table is based on an EBLM of the weighted mean CPI response computed from the EBLMs constructed for all benchmarks. Apart from providing an ordering of the performance significance of effects, the models also provide

Functional unit	**Settings**			
	Low Width		**High Width**	
	Low	**High**	**Low**	**High**
Integer ALU	1	4	2	8
Integer mult/div	1	2	1	4
Float	1	4	2	8
Float mult/div	1	2	1	4
Branch	1	2	1	4
Load/store	1	2	1	4

Table 4. Number of functional units *(num_units)*.

Operation	**Latency**		**Func. Unit**
	Low	**High**	
Int. arith./logic	2	1	Integer ALU
Int. mult.	15	2	Integer mult/div
Simple float	5	1	Float
Float. mult.	5	2	Float mult/div
Float. div.	35	10	Float mult/div
Branch	3	1	Branch
Load	Cache latency		Load/store
Store	NIL		Load/store

Table 5. Operation latencies *(op_lat)*.

Terms	**Weighted Mean**	**crafty**	**eon**	**gap**	**gcc**	**gzip**	**vortex**
intercept	1.974	2.279	2.085	2.325	2.304	1.664	1.885
pipe_depth	-0.517	-0.563	-0.501	-0.542	-0.574	-0.494	-0.511
ROB_size	-0.444	-0.399	-0.477	-0.408	-0.385	-0.456	-0.458
pipe_depth*ROB_size	0.346	0.317	0.351	0.317	0.315	0.367	0.317
IQ_size	-0.280	-0.291	-0.247	-0.283	-0.270	-0.307	-0.302
ROB_size*IQ_size	0.264	0.268	0.233	0.274	0.270	0.283	0.260
L2_lat	-0.233	-0.409	-0.312	-0.195	-0.314	-0.113	-0.249
pipe_depth*IQ_size	0.232	0.227	0.208	0.277	0.221	0.251	0.236
pipe_depth*ROB_size*IQ_size	-0.209	-0.216	-0.191	-0.156	-0.209	-0.227	-0.202
il1_size	-0.177	-0.466	-0.287	-0.164	-0.294	0.014	-0.319
dl1_lat	-0.153	-0.081	-0.150	-0.201	-0.170	-0.152	-0.126
il1_size*L2_lat	0.120	0.293	0.210	0.107	0.168	–	0.175
dl1_size	-0.100	-0.102	-0.093	-0.039	-0.124	-0.102	-0.060
op_lat	-0.092	-0.095	-0.130	-0.113	-0.077	-0.072	-0.047
issue_order	-0.082	-0.060	-0.066	-0.111	-0.056	-0.113	-0.042
il1_bsize	-0.079	-0.222	-0.170	-0.007	-0.099	0.011	-0.079
ftb_ent	-0.075	-0.239	-0.082	-0.068	-0.160	–	-0.151
il1_size*il1_bsize	0.074	0.206	0.162	–	0.072	–	0.071
L2_size	-0.071	-0.070	-0.013	-0.155	-0.134	-0.076	-0.078
dl1_size*L2_lat	0.064	0.049	0.066	0.017	0.057	0.077	0.028
L2_size*mem_config	0.059	0.050	–	0.201	0.099	0.069	0.092
dl1_assoc	-0.059	-0.041	-0.065	-0.158	-0.035	-0.059	-0.062
il1_bsize*L2_lat	0.059	0.156	0.137	–	0.053	–	0.031
mem_config	-0.058	-0.053	–	-0.260	-0.100	-0.063	-0.069
il1_size*il1_bsize*L2_lat	-0.053	-0.140	-0.109	–	-0.063	–	-0.040
ROB_size*LSQ_size	0.052	0.034	0.082	–	0.038	0.042	0.061
IQ_size*LSQ_size	-0.051	-0.034	-0.085	–	-0.021	-0.046	-0.051
LSQ_size	-0.049	-0.038	-0.070	-0.060	-0.023	-0.050	-0.023
pipe_depth*LSQ_size	0.045	0.033	0.076	–	0.017	0.041	0.044
L2_assoc	-0.044	-0.030	-0.019	-0.068	-0.053	-0.054	-0.085
ROB_size*IQ_size*LSQ_size	0.043	0.030	0.055	–	0.029	0.046	0.049
num_units	-0.042	-0.047	-0.061	-0.014	-0.034	-0.036	-0.016
ROB_size*issue_order	-0.038	-0.043	-0.036	-0.056	-0.022	-0.049	-0.022
pipe_depth*ROB_size*LSQ_size	-0.037	-0.024	-0.060	–	-0.030	-0.027	-0.039
L2_size*L2_assoc	0.036	0.035	–	0.046	0.058	0.049	0.058
L2_assoc*mem_config	0.034	0.024	–	0.060	0.052	0.047	0.057
pipe_depth*IQ_size*LSQ_size	0.034	–	0.071	–	0.002	0.029	0.027
pipe_depth*ROB_size*IQ_size*LSQ_size	-0.031	–	-0.044	–	-0.017	-0.034	-0.033
icache_assoc	-0.028	-0.074	-0.025	-0.088	-0.052	–	-0.089
L2_size*L2_assoc*mem_config	-0.028	-0.017	–	–	-0.046	-0.031	-0.048
issue_order*dl1_lat	0.027	0.013	–	–	0.050	0.038	0.033
issue_order*dl1_size	0.024	0.012	0.048	–	0.002	0.022	–

Table 7. The most significant terms and their co-efficients in the EBLMs of the CPI performance response of superscalar processors.

Diagnostics	**crafty**	**eon**	**gap**	**gcc**	**gzip**	**vortex**
Number of simulations	364	217	158	287	222	222
Number of terms	193	108	93	139	94	119
Error variance	0.001	0.002	0.001	0.002	0.001	0.002
Max. C_{jj}	0.007	0.011	0.019	0.008	0.009	0.012
Error bound (95% confidence)	0.006	0.009	0.008	0.008	0.006	0.008

Table 8. Diagnostics of the EBLMs with error bound of 0.01 at 95% confidence level.

DRAM parameter	Low	High
CPU clock:DRAM clock ratio	24	6
Module read latency	5	3
Module burst length	4	8
Module page size	512 bytes	
Bus width in bytes	8	16
CPU clock:Bus clock ratio	12	3
No. of DRAM channels	1	2
No. of banks	1	8

Table 6. mem_config parameter settings.

information on the interdependence of parameter settings on processor response. For example, the terms involving issue order provide the complete set of parameters which determine the performance variation of out-of-order issue. From the wealth of information provided by these EBLMs we summarize a few important observations below:

- Pipeline depth, reorder buffer size, and issue queue size are the three most important parameters influencing CPI performance of superscalar processors. The two-factor interactions, and the three-factor interaction involving these three parameters are highly significant. The load store queue size and its interactions with these three parameters are also significant, though at a lower level. Hence, a processor architect should simultaneously tune these parameters for optimum system performance.

- L2 cache size and latency have high impact on performance. L2 latency interacts with instruction cache size and block size, and with data cache size and associativity. L2 cache size interacts with instruction cache size and DRAM configuration. Since for a given implementation technology, L2 latency is primarily determined by L2 cache size, the processor architect should choose an optimal L2 cache size in conjunction with these interacting parameters.

- The performance with out-of-order issue is largely dependent on reorder buffer size, data cache size, and data cache latency.

- Operation latency has high significance, but it has no significant interactions. Hence, there is performance to be gained by reducing operation latency irrespective of the other processor settings.

- Amongst the predictor related parameters, the number of fetch target buffer (FTB) entries has the highest significance and the branch predictor size and type are less significant.

- The processor width has negligible impact on performance. However, the functional unit settings chosen for a given width is important. Hence, providing an adequate number of functional units is likely to be more beneficial than increasing the issue width beyond 4 instructions.

Our error bounding procedure makes it feasible to produce a complete list of such effects to any desired level of significance.

5.2. Diagnostics of Model Construction

Our model construction process constructs the EBLMs with number of simulations close to the minimum required. Table 8 reports the number of simulations and other diagnostics of the construction process of the EBLMs reported in the previous section. Note that the simulation count is approximately twice the number of extracted significant terms for each model. The table also reports the maximum C_{jj} values for constructed experimental designs, the error variance of the EBLMs, and the actual error bounds achieved by the models.

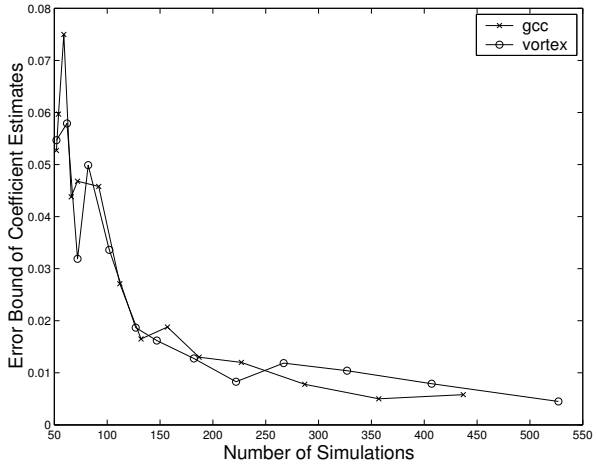


Figure 3. Tightening of error bounds.

Achieving better error bounds is possible at the cost of increased simulation. Figure 3 plots simulation counts against achieved error bounds. The error bounds globally decrease with increased simulation, though there are some local increases in the estimated error bounds. While we have used an error bound of 0.01 to obtain our EBLMs, better bounds are necessary to accurately estimate effects with values near and less than 0.01. In order to understand actual CPI variations caused by effects at this range, we studied them with all other parameters kept at suitably chosen center points. We observe that they contribute less than 2% of CPI variation. Hence, error bounds smaller than 0.01 are necessary only for studying performance variations at a finer level. The error bounding procedure can be used to construct models at any level of accuracy.

Benchmark	gcc			vortex		
Model type	$EBLM$	PB	PB/AIC_c	$EBLM$	PB	PB/AIC_c
Number of simulations	287	56	56	222	56	56
Number of model terms	139	26	22	119	26	24
Error bound (95% confidence)	0.008	0.231	0.072	0.008	0.238	0.078

Table 9. Chief characteristics of the linear models constructed using different procedures.

	eon	gzip	vortex
Additional simulations	70	40	64
Error Bound	0.010	0.006	0.009

Table 10. The number of additional simulations to include a new factor in model.

Terms	eon	gzip	vortex
il1_bsize*across_taken	-0.023	–	-0.039
L2_assoc*across_taken	-0.017	–	–
ROB_size*across_taken	–	–	0.025
ftb_ent*across_taken	–	–	-0.021
ftb_assoc*across_taken	–	–	0.017
across_taken	0.010	–	0.005

Table 11. The additional terms in new model involving fetch across taken branch.

In other work [12, 13], foldover Plackett-Burman experimental designs have been used to obtain a significance ordering of micro-architectural parameters. Hence, we next compared the simulation costs and estimation error of using such designs against that of our approach. We ran simulations for the foldover Plackett-Burman designs and computed the coefficients in a main effects model as done in [12] (PB model), and also extracted the best model from the results using the AIC_c based model extraction procedure (PB/AIC_c model). Table 9 presents the simulation count, number of model terms, and error bounds for PB,PB/AIC_c, and $EBLM$ for *gcc* and *vortex*. Though the required simulation count is approximately four times lower for PB and PB/AIC_c, the achieved error bounds are much higher. PB/AIC_c achieves lower error bounds with a small number of simulations. However, these bounds still preclude the correct estimation of many significant effects. More simulations are clearly required to extract all the significant effects.

5.3. Evaluating Enhancements

We used our error bounding procedure to rebuild the model after incorporating a micro-architectural enhancement. An additional parameter was introduced into the model, with the absence of the enhancement as its low value and its presence as the high value. We conduct augmented experiments using our error bounding procedure, after ensuring that the base model without the parameter is *stable* - i.e. increased simulations do not increase the error bound. The augmented experiments typically have the enhancement, and our procedure uses the result data from the additional simulations to extract the significant parameter estimates under the enhancement. It produces a new set of significant terms and estimates, and typically some terms with the new factor.

The enhancement we examined is the capability to fetch beyond taken branches up to the processor width, which was the motivation behind the design of trace caches [10]. Table 10 gives the number of additional simulations that were required to produce the 27 factor model with coefficient error bounds less than 0.01. The number of experiments is always less than thrice the total number of experimented factors. Table 11 gives the terms involving the new factor which account for the performance variations of the enhancement. The instruction cache block size has the highest variational effect for both *eon* and *vortex*. The main effect term has low significance, showing that the benefit of this optimization is heavily dependent on other parameter settings. *gzip*'s performance is almost unaffected by the enhancement.

6. Related Work

Much of the early research in the area of modeling and analysis of processors focused on deriving performance limits imposed by programming model constraints such as data and control dependencies while assuming infinite hardware resources. Subsequent modeling techniques have used these limits to estimate performance for realistic processor configurations by accounting for slowdowns due to various hardware limitations [6, 9]. For instance, Karkhanis and Smith [6] propose an analytical model in which performance is composed of two components, a constant idealistic throughput in the absence of any miss events and the loss in throughput due to branch mispredictions and cache misses. The impact of individual miss events on performance is modeled and estimates of the loss in throughput are obtained using various branch misprediction and cache miss statistics collected via trace driven simulations. In another approach, Fields et al. [3] model program execution

using a dynamic dependence graph and measure the significance of micro-architectural events and their interactions by the change in the length of the critical path through the dependence graph brought by idealizing these events. Perhaps the approach closest to our work is the experimental methodology proposed by Yi et al. [12, 13]. Here, the significance of the main micro-architectural parameters is obtained using experiments based on the Plackett-Burman design.

Our approach overcomes several drawbacks of these techniques. 1) In existing modeling schemes, the designer is assumed to have prior knowledge or must make simplifying assumptions about the relative significance of micro-architectural parameters and their interactions. For instance, experimentation based on the PB design inherently assumes that all parameter interactions are negligible, and Karkhanis and Smith [6] assume a set of significant miss events. Our procedure for building linear models does not require any prior knowledge or assumptions; all parameters and interactions are assumed to be significant until experiments prove otherwise. 2) Instead of modeling the effect of micro-architectural events, our approach directly captures the impact of individual micro-architectural parameters on performance. The model therefore enhances the designer's understanding of the influence of hardware changes on performance. 3) Adding parameters to the model is a matter of a few additional simulations.

7. Conclusion and Directions of Work

We have developed an algorithmic procedure to determine accurate estimates of all significant micro-architectural parameters and interactions using data from a reasonable number of simulations. This procedure builds linear models relating a processor's performance response to the micro-architectural parameters. Further, it allows the impact of micro-architectural enhancements to be included in the model with a small number of additional simulations. Thus, our procedure provides a cost effective way to experiment with all relevant parameters. The constructed error bounded linear models explain the variability in performance of micro-architectural techniques.

Our use of the reduced MinneSPEC data inputs, motivated by the need to reduce simulation time, does influence the estimated coefficients and especially the data memory hierarchy related coefficients [13]. We have used the reduced inputs since our primary aim was to show the efficacy of the linear model construction procedure. The same procedure can be used to build models for full or sampled simulations using realistic data inputs.

The constructed linear models can be used to predict the response at other parameter settings. We are continuing our work to use these models to predict the response at combinations of any chosen parameter levels in the experimented range.

References

[1] R. Desikan, D. Burger, and S. W. Keckler. Measuring Experimental Error in Microprocessor Simulation. In *Proceedings of 28th Annual International Symposium on Computer Architecture*, July 2001.

[2] P. G. Emma. Understanding Some Simple Processor-Performance Limits. *IBM Journal of Research and Development*, 41(3), 1997.

[3] B. A. Fields, R. Bodik, M. D. Hill, and C. J. Newburn. Using Interaction Costs for Microarchitectural Bottleneck Analysis. In *Proceedings of the 36th International Symposium on Microarchitecture*, December 2003.

[4] C. M Hurvich and C-L Tsai. Regression and Time Series Model Selection in Small Samples. *Biometrika*, 76:297–307, 1989.

[5] P. J. Joseph, K. Vaswani, and M. J. Thazhuthaveetil. Construction and Use of Linear Regression Models for Processor Performance Analysis. Technical Report IISc-CSA-TR-2005-16, Department of Computer Science & Automation, Indian Institute of Science, November 2005.

[6] T. Karkhanis and J. E. Smith. A First-order Model of Superscalar Processors. In *Proceedings of the 31st Annual International Symposium on Computer Architecture*, June 2004.

[7] A. J. KleinOsowski and D. J. Lilja. MinneSPEC: A new SPEC Benchmark Workload for Simulation-Based Computer Architecture Research. *Computer Architecture Letters*, June 2002.

[8] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley, 5th edition, 2001.

[9] D. B. Noonburg and J. P. Shen. Theoretical Modeling of Superscalar Processor Performance. In *Proceedings of the 27th International Symposium on Microarchitecture*, December 1994.

[10] E. Rotenberg, S. Bennett, and J. E. Smith. Trace Cache: A Low Latency Approach to High Bandwidth Instruction Fetching. In *Proceedings of the 29th International Symposium on Microarchitecture*, December 1996.

[11] Y. Sakamoto, M. Ishiguro, and G. Kitagawa. *Akaike Information Criterion Statistics*. Kluwer Academic Publishers, 1987.

[12] J. J. Yi, D. J. Lilja, and D. M. Hawkins. A Statistically Rigorous Approach for Improving Simulation Methodology. In *Proceedings of the 9th International Symposium on High Performance Computer Architecture*, February 2003.

[13] J. J. Yi, D. J. Lilja, R. Sendag, S. V. Kodakara, and D. M. Hawkins. Characterizing and Comparing Prevailing Simulation Techniques. In *Proceedings of the 11th International Symposium on High Performance Computer Architecture*, January 2005.