Fast Cost Efficient Designs by building upon the Plackett and Burman Method

Manish Arora
UC San Diego

Feng Wang
Qualcomm Inc.

Bob Rychlik
Qualcomm Inc.

Dean Tullsen
UC San Diego

ABSTRACT

CPU processor design involves a large set of increasingly complex design decisions, and simulating all possible designs is typically not feasible. Sensitivity analysis, a commonly used technique, can be dependent on the starting point of the design and does not necessarily account for the cost of each parameter. This work proposes a method to simultaneously analyzes multiple parameters with a small number of experiments by leveraging the Plackett and Burman (P&B) analysis method. It builds upon the technique in two specific ways. It allows a parameter to take multiple values and replaces the unit-less impact factor with cost-proportional values.

Categories and Subject Descriptors: C.4 [**Performance of Systems**][Design Studies]
General Terms: Design, Experimentation, Performance
Keywords: Plackett and Burman, Bottleneck, Cost optimized

1. INTRODUCTION

Since simulation of all design points is often not feasible; sensitivity analysis is a commonly used design exploration technique. This method varies a single parameter across its design space while keeping other parameters fixed. However, the reported importance of parameters is dependent on the choice of fixed values. The Plackett and Burman(P&B) method [2] has been proposed to address the shortcomings of sensitivity analysis [3]. P&B takes in a low value and high value for each parameter and runs a defined set of experiments. Endpoints for all parameters are varied simultaneously and an impact factor representng the percentage contribution to performance is calculated. The calculation needs O(N) experiments vs exponential brute force experiments. The method is not iterative, hence the total simulation time is bound by the longest experiment. However, existing proposals have key shortcomings. (1) There is little direction in choosing endpoint values and there is no way to get impact values for intermediate points. (2) The impact factors do not account for the resource costs of varying the parameter. We leverage P&B, but improve the technique by allowing a parameter to take multiple values and replacing the unit-less impact factor with cost-proportional values.

1.1 Plackett and Burman Method

P&B is a method for finding the dependence of some quantity to a set of independent variables, using few experiments.

SIGMETRICS'12, June 11–15, 2012, London, England, UK.
ACM 978-1-4503-1097-0/12/06.

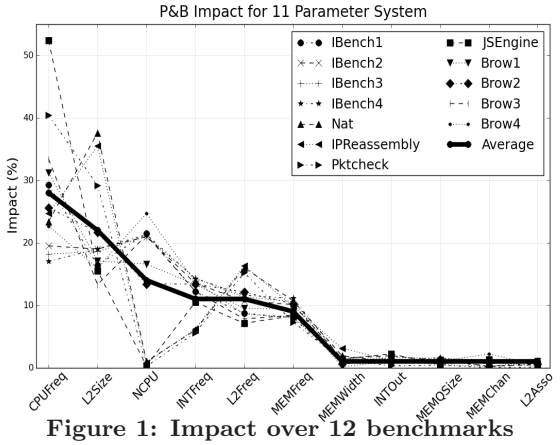

Figure 1: Impact over 12 benchmarks

Parameter	Description	Low Value (-1)	High Value (+1)	Intermediate Values
CPUFreq	CPU frequency	1GHz	2GHz	100 MHz Steps
L2Size	Shared L2 cache size	256KB	1MB	512KB
NCPU	Number of Cores	1	4	2
INTFreq	Shared interconnect frequency	200MHz	800MHz	100MHz Steps
L2Freq	Shared L2 cache frequency	1GHz	2GHz	100MHz Steps
MEMFreq	Memory system frequency	266MHz	800MHz	133MHz Steps
MEMWidth	Memory width	32 bits	64bits	-
INTOut	Interconnect queue size	4	32	-
MEMQSize	Memory controller queue size	4	16	-
MEMChan	Memory system channels	1	2	-
L2Asso	L2 cache associativity	8	32	-

Table 1: Parameter Values for our System

We use the P&B design with foldover [1], requiring 2N experiments for N parameters. For the system under test, the architect has to choose a low end value (-1) and a high end value (+1) for each parameter. Table 1 shows the values for our system. Using the chosen values, and experiments specified by the design matrix, measurements are collected. After measurements, the absolute and percentage contribution of impact of each parameter is computed as described in [3]. Figure 1 shows the impact values for our system over 12 benchmarks. CPUFreq, L2Size and NCPUs have the highest influence on performance.

2. VARYING BOUNDS

The P&B method specifies sets of experiments using combinations of +1 and -1 values but leaves the selection of the actual values up to the architect. Since typically lower bounds are more constrained, we primarily consider variations on upper bounds. Another concern with P&B is that it provides impact for the parameter over the entire range. An architect typically looks for the proverbial "knee" of the curve – if most of the marginal gain (impact) can be achieved by small increases in a parameter, we can forego the cost of fully provisioning the resource. We address this problem by setting the original upper bound to an intermediate value and rerunning the tool with a higher upper bound. This way

Description	IBench1	IBench2	IBench3	IBench4	Nat	IPReassembly	Pktcheck	JSEngine	BROW 1	BROW 2	BROW 3	BROW 4	Average
Config B vs A (% slowdown vs A)	+1.3	+5.2	+3.8	+3.4	+0.0	+0.0	+0.2	+0.6	+3.4	+0.0	+1.2	+1.3	+1.7
Config C vs A (% slowdown vs A)	+7.0	+30.8	+35.4	+30.2	+16.2	+16.3	+15.3	+16.1	+24.3	+26.5	+24.4	+41.2	+23.6
Config D vs A (% slowdown vs A)	+6.7	+19.1	+26.4	+24.9	+11.6	+11.3	+7.9	+8.8	+21.0	+18.5	+18.3	+23.8	+16.5

Table 2: Evaluating runtime performance of configurations.

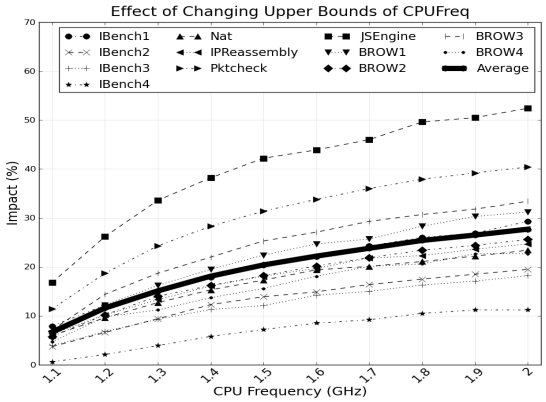

Figure 2: Impact with changing +1 values.

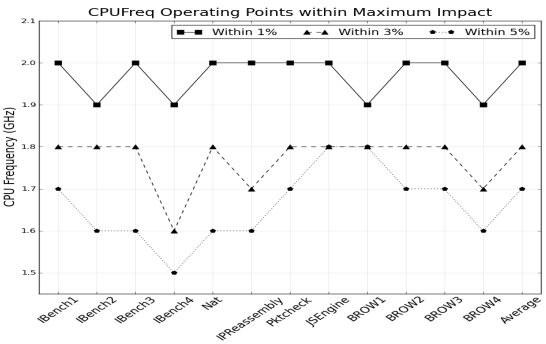

Figure 3: Operating points close to Max Impact.

we can identify the incremental impact of a parameter over various ranges and be able to identify knees. The number of extra P&B experiments for a single new max value of one parameter is N experiments. To add K design points for N parameters we require 2N + KNxN or $O(KN^2)$ experiments (far less than $O(K^N)$ exhaustive search experiments).

We add intermediate steps and analyze the impact values. Figure 2 shows results for the CPUFreq parameter. With this modification, we can determine not only what design parameters have high impact, but also what value or size of that parameter is required to get arbitrarily close to the full impact. Thus, we should be able to identify a resource-efficient design by establishing a performance threshold (e.g., within 1%, 3%, or 5% of the maximum impact), and setting each of the high-impact parameters at the lowest parameter that meets that threshold. Figure 3 shows the resulting design points varying by benchmark and within the 1/3/5% thresholds.

Based on this analysis, we can define sets of "interesting" design points. The configurations are (A) all parameters at the highest values, (B) six high impact factors set to their highest value, low impact factors set to min value (this is the design that would come out of the P&B analysis alone), (C) the six high impact factors set by the 3% threshold, (D) similar to C, but NCPU set to its highest value. Table 2 shows simulation results for these configurations relative to the execution time of the highest end system. Config B suffers only a 1.7% runtime increase versus config A. This system gives good performance, but does not allow us to compromise on resources for the high-end parameters. Cutting resources more aggressively is possible with our methodology. Config C targets the 3% operating points for the high-impact parameters. It is still within 24% of the best design, despite 5 resources being at their minimum value, and 5 of the other 6 being below their max value. Config D maximizes NCPU and goes within 17% of peak performance but has a higher energy cost despite the increased performance. Our technique allows us to navigate a very complex and dense design space with few experiments and arrive at a set of designs that provide high resource efficiency. Those designs sacrifice little in performance, yet provide significant area and energy advantages over the maximal system.

3. INCORPORATING COST MODELS

Previously we described how we can arrive at designs that use processor resources efficiently. However, we have not yet empowered the architect to make the best performance/cost decisions, since the impact factors are essentially unitless. It is difficult to compare between parameters because impact in the current form is not associated with cost required to achieve that impact. To evaluate parameters with respect to resource usage we define *cost normalized marginal impact*. Cost normalized marginal impact is defined as the P&B impact gained by moving from one design choice to another, divided by the increment of resources used at the new design. We specifically evaluate power normalized marginal impact values. Next, we formalize algorithms that utilize the cost normalized marginal impact values to derive designs that meet fixed power budgets. The algorithms work by choosing parameters with the highest cost normalized marginal impact at each step. Using our algorithm, we derive designs that consume 70% of the power but performs within 5.6% of the full design, or consume 50% of the power but perform within 19% of peak performance.

4. CONCLUSION

This work proposes a method to simultaneously analyze multiple parameters with a small number of experiments by leveraging the P&B analysis method. We extract the impact for multiple intermediate points. There is an increase in the number of experiments, but still few enough to be highly practical. With these intermediate points, we can choose incremental points in the design space that get close to full performance with a significant decrease in resources. This paper also incorporates a cost model with the P&B method. We produce results that can lead the architect directly to an architecture that sets the intermediate value of each parameter so as to maximize performance for a given cost constraint.

5. REFERENCES

[1] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2006.

[2] R. L. Plackett and J. P. Burman. The design of optimum multifactorial experiments. *Biometrika*, 1946.

[3] J. Yi, D. Lilja, and D. Hawkins. A statistically rigorous approach for improving simulation methodology. In *HPCA: High Performance Computer Architecture.*, pages 281 – 291, Feb. 2003.