Design Space Modeling and Simulation for Physically Constrained 3D CPUs

Caleb Serafy
Oracle Corporation
Santa Clara, CA, USA
caleb.serafy@oracle.com

Zhiyuan Yang and Ankur Srivastava
University of Maryland
College Park, MD, USA
{zyyang, ankurs}@umd.edu

ABSTRACT

Design space exploration (DSE) is becoming increasingly complex and 3D integration compounds the problem by imposing a more complex design space. Moreover, 3D design is a new frontier of CPU design, expected to rely more heavily on statistical modeling than designer intuition. Past work has used regression modeling with random sampling of the design space. In this paper we propose a directed simulation technique where intermediate predictions more efficiently direct simulation resources. Our results show over 98% model accuracy while simulating less than 5% of the design space and reducing simulation time more than 4.5x compared to random sampling.

1. INTRODUCTION

Design space exploration (DSE) involves the evaluation of a multitude of design choices prior to detailed implementation. Such a technique is necessary to identify regions of interest in the design space and perform educated trade-off analysis of conflicting objectives. In its simplest form, DSE can be performed by exhaustively simulating the entire design space. However as CPU designs become ever more complex in the pursuit of Moore's law performance scaling, the DSE problem has become increasingly intractable as the design space grows combinatorially in the number of design parameters. Exhaustive simulation across such large design spaces is inefficient and potentially infeasible or unaffordable in terms of run-time.

Past work has attempted to overcome the computational infeasibility of exhaustive simulation in two ways. One is to reduce simulation time by orders of magnitude using techniques such as host-compiled simulation [2] or statistical simulation [3]. Although these approaches can make exhaustive simulation possible, the accuracy of such fast simulation techniques is reduced, and the applicability of the techniques is limited in scope. Another approach to the DSE problem is to simulate only a small subset of the full design space and use modeling techniques to predict the properties of un-simulated designs. Modeling approaches [8, 7, 11, 9] have shown promising results on large architectural design spaces.

Vertical integration of circuits (3D ICs) is an up-and-coming technology that shows great potential towards improving circuit power and performance as well as facilitating new CPU architecture paradigms such as stacked memory and highly connected on-chip networks [18]. However 3D ICs also bring new challenges, chief among them thermal management [15, 19], and moves the architectural design problem into uncharted territory where traditional domain knowledge and designer intuition may no longer apply. Moreover, past work [18, 19] has shown that significant portions of the 3D CPU design space can be infeasible due to physical con-

straint violations. 3D integration significantly complicates the DSE problem as follows:

- 3D integration brings many new architectural opportunities that compound the intractability of exhaustive simulation.
- The effects of these new architectures on the design trade-off space are currently not well understood.
- 3D ICs are more thermally sensitive to architectural changes than equivalent 2D chips due to their physical structure.
- 3D ICs can eliminate communication bottlenecks that are inherent in 2D ICs, making performance and power more sensitive to architectural changes.
- *Ad hoc* fixes late in the design cycle due to poor architectural design choices can be more costly in 3D ICs because of resource conflicts between transistors and TSVs.

Physically aware DSE is becoming more important, especially in the context of 3D ICs. Past work [13, 19] has examined the effect of physical constraints on a CPU design space, but has only done so with exhaustive simulation over a small design space. On the other hand, the literature on design space modeling [8, 7, 11, 9] has only attempted to model optimization variables such as performance or energy efficiency with no consideration of physical constraints.

In this paper we introduce a modeling and simulation technique for 3D CPUs. The proposed technique models physical properties (*e.g.*, power, area and temperature) and traditional optimization metrics (*e.g.*, instructions per second or energy-delay-product) and uses these models to direct simulation effort towards user-defined regions of interest in the design space for the purpose of identifying interesting trends such as the Pareto optimal trade-off curve. Our models accurately predict the performance and temperature of a diverse 3D CPU design space and identify the optimal feasible design point (Pareto optimal design set) with 100% (98%) accuracy while simulating less than 2% (5%) of the design space.

In the rest of this paper we discuss the related work in the literature, describe our power-performance-temperature simulation methodology and design space modeling technique, and demonstrate its effectiveness and accuracy using two case studies.

1.1 Previous Work

Methodologies to facilitate large scale DSE have taken two orthogonal approaches: drastically reduce simulation time or predict the properties of un-simulated design points through design space modeling. Towards the former, techniques such as statistical simulation [3] and host-compiled simulation [2] have been pursued. Both techniques massively reduce simulation time, but at the cost of reduced accuracy and limited applicability.

Design space modeling likewise trades off accuracy for decreased simulation time by predicting rather than actually simulating significant portions of the design space. Historically, design space modeling techniques [8, 7, 11, 9] have used uniform random sampling to build models of the entire design space. Early work by Joseph *et al.* [9] used linear regression to model instructions per cycle (IPC) across a 23-variable CPU design space. However only two factors of each variable were considered which is not sufficient to evaluate the accuracy of the technique. Two similar works were published the same year by Lee and Brooks [11] and İpek *et al.* [7]. These applied spline regression and artificial neural network

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GLSVLSI '17, May 10-12, 2017, Banff, AB, Canada

© 2017a ACM. ISBN 978-1-4503-4972-7/17/05...\$15.00

DOI: <http://dx.doi.org/10.1145/3060403.3060499>

Figure 1: 3D CPU simulation flow

models to accommodate multi-factor non-linear design space properties. They yielded average errors less than 10% and maximum error around 50%. More recent work by Jia *et al.* [8] applied spline regression to GPUs. This technique reduced maximum error to around 15% and had average error in the single-digit range. However, a limitation of the current state of the literature is a lack of focus on application of the proposed modeling techniques to efficiently solve DSE problems.

Moreover, in the aforementioned prior art there is a missed opportunity. A significant potential advantage of modeling approaches is the ability to control the accuracy of the model in different regions of the design space, which we refer to as directed simulation. This is important because it is often the case that accuracy of the simulations is only important in a small subset of the design space, such as the Pareto front for the design objectives at hand, or the region of physically feasible design points. Directed simulation can improve the efficiency of design space modeling by optimizing the amount of useful information in each simulation.

1.2 Contributions

This paper makes the following contributions:

- We propose a directed simulation technique that builds regression models to identify the region of the design space that is of interest to the designer and predict optimization metrics and physical properties within that region while only simulating a small subset of the design space.
- To the best of our knowledge our work is the first to apply design space modeling techniques to 3D CPUs. 3D CPU design is expected to rely more on design space modeling than traditional CPU architectures due to a lack of designer experience and intuition regarding this emerging technology and architectural paradigm.
- To the best of our knowledge our work is the first to apply design space modeling to physical properties such as temperature to predict the feasibility region of a design space. This is extremely important for designing 3D CPUs which are known to be heavily thermally constrained.

2. 3D CPU SIMULATION FLOW

The simulation flow used to evaluate power, performance, area and temperature of a specific 3D CPU architectural design point is shown in Figure 1. The details of each component of the simulation flow are explained in the following subsections.

2.1 Performance Simulation

Performance simulation is performed by Multi2Sim (M2S) [22], a cycle accurate CPU simulator. Architectural definitions are passed to the simulator that describe parameters such as: number of cores, pipeline width, buffer/queue/register size, cache size/associativity/latency, network-on-chip (NOC) topology/latency *etc.* Cache and register latencies are determined using CACTI [21] to provide realistic architectural setups to the simulator. DRAM latency is calculated using the model proposed in [18] and NOC topology/latency is calculated as explained in Section 2.4. M2S simulates the execution of an x86 binary file on the described CPU. The simulator outputs a list of performance statistics such as IPC, memory reads,

writes, hits and misses, branch prediction rate *etc.* The specific architectural configurations and software workloads simulated for this study are discussed in Section 4.

2.2 Power and Area Estimation

Power and area estimation is performed by McPAT [12], a multi-core power, area and timing estimation tool. Based on the provided architectural parameters, the area and energy per access and leakage power density of each CPU component is estimated for a selected technology node. Using the performance statistics generated by M2S, the dynamic power dissipation of each CPU component can be estimated. The leakage power density is estimated for a nominal temperature, but is thermally scaled during thermal simulation (Section 2.5).

2.3 Floorplan

Each core is partitioned into a set of components (*e.g.*, data cache, register alias table (RAT), branch predictor, instruction queue and execution unit) which are interconnected by an abstract netlist. Detailed descriptions of each component can be found in [12]. A general core floorplan (FP) topology was generated offline which minimizes the wirelength between connected components.

2.4 Core Tiling and NOC Design

The core floorplan is replicated on an $i \times j \times k$ grid such that $i \cdot j \cdot k = n$ where n is the total number of cores. The values i , j and k are chosen such that:

- Total footprint area $i \cdot \text{width}_{\text{core}} \cdot j \cdot \text{height}_{\text{core}} < A_{\text{max}}$.
- Total number of layers is minimized.
- Layer aspect ratio $(i \cdot \text{width}_{\text{core}} / j \cdot \text{height}_{\text{core}})$ is relatively square.

NOC topology is defined as an $i \times j \times k$ 3D mesh [18] and NOC latency is defined as the wire delay of length $\max(\text{width}_{\text{core}}, \text{height}_{\text{core}})$, calculated using the wire delay model from [16] with technology parameters extracted from McPAT source code. NOC topology and latency are fed back into the performance simulator to get accurate inter-core communication simulations.

2.5 Thermal Model

Once the chip floorplan has been constructed and component power estimation is complete, we have a power density map for each tier of the 3D stack. Power density maps are converted into thermal maps using our compact thermal model [20]. A 3D grid is constructed representing the physical structure of the 3D IC. Likewise the power map is discretized into a 3D grid and the total power of each power grid is assigned to the respective physical grid.

2.6 Leakage Model

McPAT reports a base leakage value for each CPU component which is estimated at a fixed temperature T_0 . To obtain more accurate leakage power estimates we iteratively solve our thermal model and then scale leakage estimates at each grid based on the estimated temperature of that grid after the previous iteration.

3. DSE MODELING TECHNIQUE

In this section we introduce our modeling and simulation technique for 3D CPU DSE subject to physical constraints. We use the

Figure 2: Modeling and simulation technique

smoothing spline analysis of variance (SS-ANOVA) [5] modeling technique to build models for each design parameters of interest (*e.g.*, performance, temperature and power). Models are composed from cubic spline functions evaluated on combinations of design variables (*i.e.* model terms). First we give some background on SS-ANOVA modeling and then describe our proposed modeling and directed simulation technique. Figure 2 illustrates the overall flow of our modeling and simulation technique, and details are given in the subsections below. The basic flow is an iterative back-and-forth between model building and choosing new simulation points based off the intermediate model predictions.

3.1 SS-ANOVA Modeling

A spline is a piecewise polynomial function [5]. Splines are both differentiable and continuous at the piecewise boundaries [5]. The smoothing spline is a regression technique to smooth noisy data by fitting a spline function to the data. Analysis of variance (ANOVA) is a statistical technique for analyzing the underlying source of variations in a population [5]. Multi-factor ANOVA can be used to generate models of a data set as a function of descriptive properties of each observation. An observation f can be modeled as a function of the variables $\mathbf{x} = x_1, x_2, \dots, x_n$ as shown in Equation (1) [5]. SS-ANOVA limits the set of functions $\{f_1, \dots, f_{1,2,\dots,n}\}$ to be spline functions which operate on some subset of the variables in $\mathbf{x}$. Each unique subset of input variables is called a term, and the order of a term is the size of the subset.

$$f(\mathbf{x}) = c + \sum_{j=1}^n f_j(x_j) + \sum_{j=1}^n \sum_{k=j+1}^n f_{j,k}(x_j, x_k) + \dots + f_{1,2,\dots,n}(x_1, x_2, \dots, x_n) \quad (1)$$

In this work we use the `gss` [4] package for the statistical computing environment R [17] to generate a unique smoothing spline model for each design property of interest. To generate each model, `gss` requires a set of simulation data and a set of model terms. However, choosing the appropriate simulation points and model terms are nontrivial problems. The choice of model terms and simulations points strongly affects the quality of the model and suboptimal choices have a high cost in terms of total simulation time and model complexity. Our iterative technique for model term and simulation point selection and is explained in detail in the following subsections.

3.2 Choosing Model Terms

The maximum number of terms associated with n variables is 2^n . However as a rule of thumb a model is unreliable when the number of terms is greater than $s/20$ [6] where s is the number of simulated points. If too many model terms are used, the model can suffer from over-fitting, making it very accurate with respect to the observed data, but a poor predictor of the un-simulated data we wish to predict. Thus the number of model terms must be

kept relatively small in order to maintain model accuracy when the number of simulations is small. The intended goal of the modeling and simulation approach is to build accurate models while requiring only a small number of simulations, so avoidance of the over-fitting problem is of critical importance.

The coefficient of determination (R^2) is a commonly used metric to evaluate model quality. However, by construction R^2 monotonically increases as new terms are added to a model [8]. Thus optimization of R^2 itself would inevitably lead to inclusion of all model terms, unnecessarily complicating the model and potentially causing over-fitting. Adjusted R^2 ($\bar{R}^2$) (Equation (2)) scales R^2 relative to the number of model terms, m , and the number of data points, s , allowing the significant terms to be identified.

$$\bar{R}^2 = 1 - (1 - R^2) \cdot \frac{s-1}{s-m-1} \quad (2)$$

We use a forward selection $\bar{R}^2$ based technique to select the terms in the model. The model building technique is shown in the bottom half of Figure 2. Starting with an empty model we consider each model consisting of one first order term. We evaluate the $\bar{R}^2$ metric for each model and accept the one with the largest value. We then consider adding each remaining first order term (in decreasing order of improvement) and accept the terms that increase the quality of the model by at least θ .

Every time a new first order term is added to the model, we consider all second order interaction terms created by combining the new term with any other terms already in the model. Amongst all new second order terms generated this way we add (in decreasing order of improvement) any that cause the model quality to improve by at least θ . The model is complete once all first order terms have been added to the model, or when adding any new first order terms causes model quality to improve less than θ .

3.3 Adding Simulation Points

The designer defines a discovery metric, which defines the point(s) in the design space they are interested in identifying. Some examples of potential discovery metrics are the optimal design point subject to a set of constraints (*e.g.*, design space optimization), or the set of Pareto optimal designs (*e.g.*, trade-off analysis). The optimality metric (*e.g.*, performance or energy efficiency), constraints (*e.g.*, temperature, power, area or timing) and Pareto metrics are defined by the designer. The goal of our proposed modeling and simulation technique is to identify these points by iteratively predicting them and concentrating simulator effort around the predicted point(s) to improve the accuracy of the prediction.

Initial models are built using a random sampling of η simulation points. Using model predictions, the predicted design point(s) of interest are identified. Due to model error, the identified point(s) are not necessarily the true points of interest. However, the true points

Table 1: Architectural design space (baseline values shown in bold).

Variable	Value(s)
Technology Node	32 nm
Number of cores (<i>core</i>)	{ 8 , 16, 32}
Memory controllers	<i>core</i> · {1/2, 1/4, 1/8}
Clock frequency	{ 2.4 , 3.0} GHz
NOC width	128 bits
L2 cache size (per core)	{ 256 , 512, 1024} kB
L2 cache associativity	{4, 8, 16}
L1 cache size (per core)	{ 16 , 32, 64} kB
L1 cache associativity	1
Pipeline width	{2, 4, 6}
Branch predictor	Tournament
Local history table	1024 8-bit entries
Global predictor	4096 2-bit entries
BTB size	32 kB
BTB associativity	1
Reorder buffer length (<i>rob</i>)	{ 96 , 128, 160}
Issue queue length	0.4 · <i>rob</i>
Load-store queue length	0.5 · <i>rob</i>
Fetch queue length	64
Int architectural registers	0.67 · <i>rob</i>
FP architectural registers	0.33 · <i>rob</i>
RAT size	<i>rob</i> 8-bit entries
DRAM size	4 GB
Cache line size	64 B
DRAM bus width	64 B

of interest are likely to be close to the predicted points of interest. Thus a region of interest (ROI) is defined which contains the design points which are close to the predicted point(s) of interest, and additional simulation effort is concentrated towards this ROI to improve model fidelity in that region.

Each iteration of the flow identifies χ new design points from the predicted ROI and queues them for simulation. Once the simulations are performed, the model is rebuilt and the process repeats. If the initial model mispredicts the ROI, additional simulation effort in the mispredicted region will reduce model residuals in that region and improve ROI prediction on the next iteration.

3.4 Stopping Criteria

Stopping criteria could involve reaching a maximum number of simulations, or a sustained convergence in predictions of ROI and/or point(s) of interest across multiple iterations. In order to evaluate the behavior of our algorithm we simply set the stopping criteria to terminate when the total number of simulations reaches ζ . We investigate the trade-off between number of simulations and optimality of our selected design space in Section 5, and the point at which prediction convergence is achieved is observed *post hoc*.

4. EXPERIMENTAL SETUP

In this section we describe the experimental setup to evaluate the proposed modeling and simulation technique (Section 3). In the following subsections we introduce the 3D CPU design space, the discovery metrics and associated ROI definitions considered in our case studies and the metrics we use to measure the effectiveness of our approach. Results are presented and discussed in Section 5.

4.1 Architectural Design Space

Our study searches the architectural design space given in Table 1 which contains 4374 unique design points. We consider 3D CPUs with stacked DRAM which is considered to be one of the primary advantages of 3D CPUs [10, 14]. By integrating the DRAM on chip using through silicon vias (TSVs), the core-memory bandwidth can be increased drastically due to increased memory bus width, more memory controllers and faster bus speed [15, 18].

Each architectural design point is evaluated using a set of software workloads from the SPLASH-2 [23] and PARSEC [1] benchmark suites. The performance of each workload is normalized to the baseline architecture (Table 1) to allow unbiased averaging across workloads. Maximum temperature for a design point is the maximum across all workloads.

4.2 Discovery Metrics

The goal of our DSE study is to identify the design point(s) of interest as defined by the discovery metric chosen by the designer. Two discovery metrics are considered as case studies in this paper, but our proposed methodology is applicable to any arbitrary discovery metric. In this study the modeled design parameters are performance and temperature and the discovery metrics are:

- **“Optimal”**: design point with highest normalized performance subject to thermal constraint $temp_p < T_{violation}$.
- **“Pareto”**: Pareto optimal set of design points in thermal-performance space.

Each discovery metric defines an accompanying ROI of radius $\phi = (\phi_{perf}, \phi_{temp})$. The ROI for the “Optimal” and “Pareto” discovery metrics are given in Equations (3) and (4)* respectively, where $perf_i$ and $temp_i$ are the performance and temperature of design point i and Ω is the design space. Design point p is the predicted optimal feasible point for the discovery metric “Optimal”. The defined ROI is the set of points within distance ϕ of the identified point(s) of interest.

$$ROI_{Optimal} = \{i \in \Omega \mid \left| \frac{perf_i - perf_p}{perf_p} \right| \leq \phi_{perf} \wedge |temp_i - temp_p| \leq \phi_{temp}\} \quad (3)$$

$$ROI_{Pareto} = \{i \in \Omega \mid \forall_{(j \neq i) \in \Omega} perf_j \cdot (1 - \phi_{perf}) \leq perf_i \vee (temp_j + \phi_{temp}) \geq temp_i\} \quad (4)$$

4.3 Modeling and Simulation Parameters

The modeling and simulation technique introduced in Section 3 can be parametrized to make trade-offs between simulation time and optimality of the selected design point. In this study we use the following parameters which were found experimentally to offer favorable trade-offs:

- We sample $\eta = 40$ simulation points at random from the design space to build the initial model. This parameter should be large enough to generate an initial model with reasonable accuracy but small enough to avoid degeneration towards random sampling.
- The threshold for accepting new model terms is $\bar{R}_{new}^2 - \bar{R}_{current}^2 > \theta = 0$. By increasing θ , the model complexity could be reduced at the expense of quality.
- We use ROI radius of $\phi = (8\%, 4^\circ\text{C})$ for discovery metric “Optimal” and $\phi = (5\%, 3^\circ\text{C})$ for discovery metric “Pareto”. Larger values of ϕ reduce the probability of convergence to local minima, but generally increase simulation time.
- We iteratively simulate chosen design points in increments of $\chi = 5$. Smaller values of χ inefficiently increase the number of times model building is performed. However excessively large values of χ over-invest simulation resources into a potentially inaccurate intermediate estimation of ROI.
- We use a nominal stopping criteria of $\zeta = 200$ simulations. The trade-off of optimality vs. number of simulations is investigated in Section 5.

4.4 Evaluation Metrics

The goal of the experiment is to identify the ROI defined by the discovery metric, while minimizing the total number of simulations performed. Thus the primary metrics used to evaluate the quality of our technique will be accuracy of identification, number of simulations and run-time overhead of the model building.

When the discovery metric is “Optimal”, the distance between the identified point and the true solution is quantified as optimality,

*Equation (4) presents a ϕ -relaxed definition of Pareto optimality that includes all points such that no other point is better by a degree of ϕ in all metrics of interest.

Figure 3: Thermally feasible performance distribution under different thermal constraints

which is the ratio per_{fp}/per_{fo} where p is the predicted optimal feasible point and o is the true optimal feasible point (determined by exhaustive simulation for evaluation).

When the discovery metric is “Pareto”, the distance between the identified points and the true Pareto set is quantified as accuracy, which is the average Pareto optimality of the predicted Pareto set. The Pareto optimality of design point k is determined by finding the smallest value of ϕ such that k is included in the ROI. Specifically, the Pareto optimality of k is α_k and the smallest value of ϕ that includes k in the ROI is $\phi = (1 - \alpha_k)(100\%, 60^\circ\text{C}^\dagger)$.

In general the optimality/accuracy of the predicted point(s) will increase as more simulations are performed, eventually degenerating into the exhaustive simulation. The net speedup of our technique consists of the reduction in total number of simulations minus the run-time overhead of building the models.

4.5 Comparison to Other Techniques

The rudimentary technique to which our technique is compared is a random sampling methodology where some portion of the solution space is sampled at random and the best design amongst the sampled designs is selected[‡]. Additionally we could consider modeling-only version of our proposed technique that uses SS-ANOVA model building to predict the design point(s) of interest, but simply uses random sampling to provide data to the model builder. The modeling-only approach is representative of design space modeling techniques proposed in past work [8, 7, 11, 9]. In Section 5 we compare the trade off curves of simulation count vs. quality for the three aforementioned techniques:

- **Proposed:** modeling and directed simulation
- **Modeling-Only:** modeling and random simulation[§]
- **Random Sampling:** random simulation (no modeling)

5. RESULTS

In this section we first provide some characterization of the design space explored in our study, and then compare the quality of the investigated methodologies (Section 4.5) for the “Optimal” and “Pareto” discovery metrics.

5.1 Design Space Characterization

We begin by examining the properties of the design space. Exhaustive simulation was performed for the purpose of evaluation. Exhaustive simulation took weeks to perform using university servers with over 100 cores, further motivating the strong need for techniques such as the one proposed in this paper in order to reduce simulation time. We provide some statistics of the design space properties in order to give context for the results of this study.

Figure 3 shows the distribution of normalized performance across all architectural design points. We can see that the design space is biased heavily towards the low-performance region. Furthermore, thermal feasibility constraints bias the design space even further as

[†]60 °C was roughly the thermal range of the design space considered in this work as shown in Figure 4.

[‡]Exhaustive simulation is a degenerative case of random sampling.

[§]Representative of past work [8, 7, 11, 9].

Figure 4: Temperature vs. performance of entire design space

the constraints tighten (*i.e.* $T_{violation}$ is reduced). This implies that random sampling is not a very good technique for discovering the “Optimal” design point since the probability of randomly sampling a high-performance thermally-feasible design point is low.

Figure 4 shows a scatter plot of the performance and temperature of each design point in the design space. We can see that identification of both the optimal feasible design point and the Pareto optimal design set without exhaustive simulation is non-trivial. The vast majority of design points in the design space are far from the point(s) of interest using either discovery metric. Moreover the correlation between performance and temperature is weak, motivating the need for independent models of each design property.

5.2 “Optimal” Discovery

There exists a fundamental trade-off between the number of simulations and the quality of the identified solution. We compare the random sampling and modeling-only technique to our proposed modeling and simulation technique and show that our technique significantly reduces the number of simulations required to discover the region of interest in the design space. First we evaluate the techniques using the “Optimal” discovery metric.

Figure 5 plots the number of simulations required to identify a design point of interest with a specified optimality. We observe that modeling alone is a large contributor to the optimality of the identified point. With less than 1% of the solution space sampled (40 points), the two modeling techniques can already identify a solution within 90% of the optimal. However the true power of the proposed technique becomes clear as the optimality target increases. Optimality can be improved significantly for both modeling techniques by adding only a nominal amount of additional simulation samples. However the modeling-only technique begins to degenerate towards random sampling for optimality targets beyond 98% whereas the proposed technique shows no such degradation. By using models to direct simulation effort on each iteration towards the ROI, the technique is able to make roughly linear improvements to prediction accuracy for each additional simulation. Our proposed technique is able to identify the optimal feasible design point while simulating less than 2% (roughly 80 points) of the entire design space, saving 100s of simulation hours compared to the other two techniques.

Robustness to Constraint Tightness: The previous results were evaluated at $T_{violation} = 85^\circ\text{C}$. However as Figure 3 shows, reducing $T_{violation}$ to 65°C significantly reduces the size of the thermal feasibility region. It is expected that a smaller feasibility region will reduce the quality of the random sampling technique significantly, but it is unclear how it would affect the techniques that use model building. We exercise the robustness of the investigated techniques to constraint tightness by lower the thermal constraint. However, the resulting observations would be extendable to analogous situations such as using a lower quality heatsink or a more power-dense manufacturing technology.

Figure 6 plots the number of additional simulations required to accommodate the reduced thermal constraint. We notice that the number of additional simulations required for our proposed method is less than 30 (<1% of the entire design space) and moreover, remains roughly constant as the optimality target is tightened. On the other hand random sampling and modeling-only both require su-

Figure 5: Number of simulations vs. optimality of design[¶]

Figure 6: Additional simulations required for 65°C vs. 85°C constraint[¶]

perlinearly increasing amounts of additional simulations in order to meet optimality targets. Although modeling-only scales reasonable well for small optimality targets, the point at which the technique begins to degenerate is significantly reduced to roughly 95%. Thus we show that compared to a modeling-only technique, our directed simulation technique is significantly more robust to design space feasibility constraints.

5.3 “Pareto” Discovery

Figure 7 shows the accuracy of the considered methods when the “Pareto” discovery metric is applied. Although the general trends and relative ordering of the method results are similar to the “Optimal” case, there are some significant differences. The quality of both model-based techniques is reduced because identification of a set of Pareto points is a more challenging problem which inherently requires more simulation. However the relative improvement of our proposed technique vs. the modeling-only technique is substantially more obvious under this discovery metric. This indicates that, similar to constraint tightness, the proposed technique is significantly more robust to increased problem complexity.

Likewise, the modeling-only technique degenerates into random sampling at an even lower accuracy target than was observed in the previous studies. The conclusion here is that models built with random sampling can approximating a single design much better than the relative ordering of all design points. Directed simulation towards the ROI is of utmost importance for estimation of the Pareto design set, even for rather loose accuracy targets.

5.4 Overhead of modeling approach

There is obviously some run-time overhead associated with model-building in the investigated modeling approaches. We observed that the time consumed building models in our proposed approach was less than the time consumed to simulate a single design point ($< 0.025\%$ of the design space). Our results clearly show that such overhead is negligible compared to the savings in number of required simulations compared to random sampling.

6. CONCLUSIONS

In this paper we propose a comprehensive technique for 3D CPU architectural design space exploration subject to physical constraints. We use smoothing spline regression modeling to efficiently direct

[¶]Data is plotted on a log-log axis, so polynomial relationships appear as a straight lines with slope proportional to degree.

Figure 7: Number of simulations vs accuracy of Pareto set[¶]

our simulation effort. We demonstrate our technique using two case studies. Our technique identifies the optimal feasible design point (Pareto optimal design set) with 100% (98%) accuracy while simulating less than 2% (5%) of the design space.

7. ACKNOWLEDGEMENTS

The authors would like to acknowledge that this work has been funded by NSF grant CCF1302375 and DARPA ICECool Project.

8. REFERENCES

- [1] C. Bienia, et al. The PARSEC Benchmark Suite: Characterization and Architectural Implications. In *PACT*, 2008. ACM.
- [2] D. Gandhi, et al. FastSpot: Host-compiled thermal estimation for early design space exploration. In *ISQED*. IEEE, 2014.
- [3] D. Genbrugge and L. Eeckhout. Chip multiprocessor design space exploration through statistical simulation. *IEEE Transactions on Computers*, 2009.
- [4] C. Gu. Smoothing Spline ANOVA Models: R Package gss.
- [5] C. Gu. *Smoothing spline ANOVA models*, volume 297. Springer Science & Business Media, 2013.
- [6] F. E. Harrell. *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*. Springer Science & Business Media, 2013.
- [7] E. İpek, et al. Efficiently Exploring Architectural Design Spaces via Predictive Modeling. In *ASPLOS*, 2006. ACM.
- [8] W. Jia, et al. Stargazer: Automated regression-based GPU design space exploration. In *ISPASS*. IEEE, 2012.
- [9] P. Joseph, et al. Construction and use of linear regression models for processor performance analysis. In *HPCA*. IEEE, 2006.
- [10] D. H. Kim, et al. 3D-MAPS: 3D Massively parallel processor with stacked memory. In *ISSCC*. IEEE, 2012.
- [11] B. C. Lee and D. M. Brooks. Accurate and efficient regression modeling for microarchitectural performance and power prediction. In *ACM SIGPLAN Notices*, volume 41. ACM, 2006.
- [12] S. Li, et al. McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *MICRO*. IEEE/ACM, 2009.
- [13] Y. Li, et al. CMP design space exploration subject to physical constraints. In *HPCA*. IEEE, 2006.
- [14] G. H. Loh. 3D-Stacked Memory Architectures for Multi-core Processors. In *ISCA*, 2008. IEEE.
- [15] J. Meng, et al. Optimizing Energy Efficiency of 3-D Multicore Systems with Stacked DRAM Under Power and Thermal Constraints. In *DAC*, 2012. ACM.
- [16] R. H. J. M. Otten and R. K. Brayton. Planning for Performance. In *DAC*, 1998. ACM.
- [17] B. D. Ripley. The R project in statistical computing. *MSOR Connections*, 2001.
- [18] C. Serafy, et al. Unlocking the True Potential of 3-D CPUs With Microfluidic Cooling. *IEEE TVLSI*, 2015.
- [19] C. Serafy, et al. Design Space Exploration of 3D CPUs and Micro-Fluidic Heatsinks with Thermo-Electrical-Physical Co-Optimization. In *InterPACK*. ASME, 2015.
- [20] B. Shi, et al. Non-uniform Micro-channel Design for Stacked 3D-ICs. In *DAC*, 2011. ACM.
- [21] P. Shivakumar and N. P. Jouppi. Cacti 3.0: An integrated cache timing, power, and area model. Technical report, Compaq Computer Corporation, 2001.
- [22] R. Ubal, et al. Multi2Sim: a simulation framework for CPU-GPU computing. In *PACT*. ACM, 2012.
- [23] S. C. Woo, et al. The SPLASH-2 Programs: Characterization and Methodological Considerations. In *ISCA*, 1995. ACM.