Prospector: Synthesizing Efficient Accelerators with Statistical Learning

Atefeh Mehrabi, Aninda Manocha, Benjamin C. Lee, Daniel J. Sorin
atefeh.mehrabi@duke.edu, amanocha@princeton.edu
benjamin.c.lee@duke.edu, sorin@ee.duke.edu
Electrical and Computer Engineering Department, Duke University

Abstract

Accelerator design is expensive due to the effort required to understand an algorithm and optimize the design. Architects have embraced two technologies to reduce costs. High-level synthesis automatically generates hardware from code. Reconfigurable fabrics instantiate accelerators while avoiding fabrication costs for custom circuits. We further reduce design effort with statistical learning. We build an automated framework , called Prospector, that uses Bayesian techniques to optimize synthesis directives, reducing execution latency and resource usage in field-programmable gate arrays. Prospector discovers Pareto-efficient designs more quickly than prior approaches and permits new studies for heterogeneous accelerators.

1 The Prospector Framework

Prospector identifies synthesis parameters that best optimize an accelerator design. We leverage Bayesian optimization, which builds a probabilistic model that approximates an unknown function and serves as its surrogate [4]. Bayesian techniques are particularly effective when the function is a black box and evaluating the function is costly. The technique iteratively samples parameter values, evaluates the function with those values, and updates the probabilistic model. Over multiple iterations, the optimization approaches optimal parameter values $\hat{x} = *arg\,max_{x \in X} f(x)$ for function f.

We consider the HLS toolflow (*i.e.*, simulate, synthesize, place-and-route) an unknown function to be modeled and optimized. The function's inputs specify the placement and configuration of synthesis directives [2, 3]. The function's outputs quantify design quality, which include execution time and multiple measures of FPGA resource utilization. Although we can invoke the HLS toolflow to evaluate the function for any set of inputs, evaluations are prohibitively expensive when exploring large, complex design spaces.

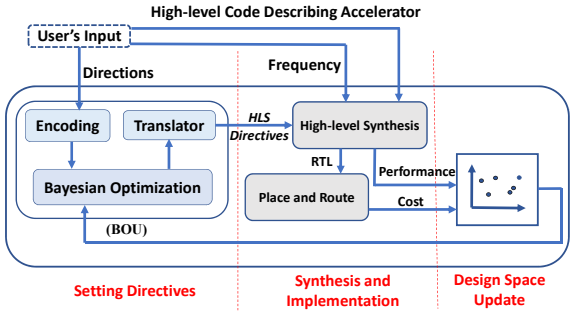

Figure 1: Prospector Framework

Figure 1 summarizes the Prospector framework. Its input is high-level source code that describes accelerator functionality. Its outputs are RTL implementations that reflect varied performance and cost trade-offs. Within Prospector, the Bayesian optimization unit (BOU) explores the design space iteratively. In each iteration, the BOU controls the choice of synthesis directives and the HLS toolflow converts source code to RTL. After multiple iterations, Prospector identifies synthesis directives and accelerator designs that balance execution time and FPGA resource utilization. Prospector supports multi-dimensional optimization. Users can select multiple figures of merit as objectives.

Prospector explores design spaces with Bayesian optimization. We devise a novel encoding for synthesis parameters, enabling Gaussian processes that model the joint effects from the placement and configuration of optimization directives. Moreover, we acquire training data judiciously, exploiting novel Bayesian methods

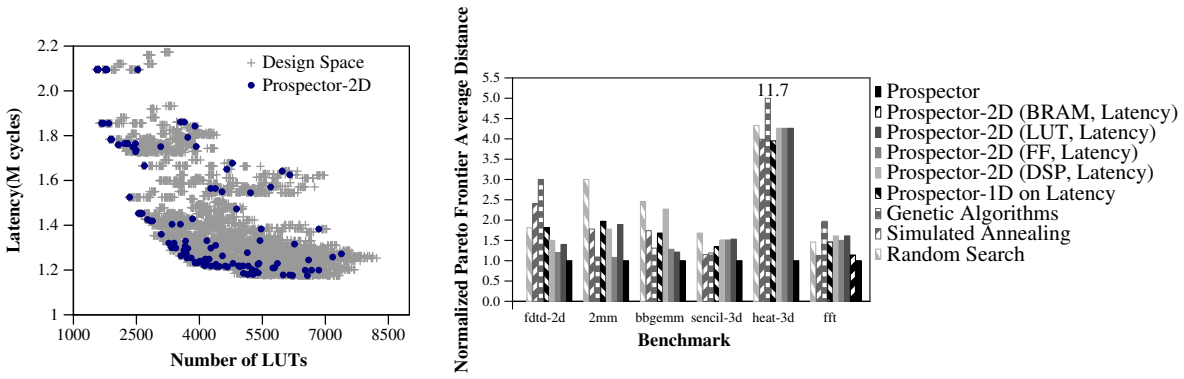

Figure 2: Exploring design space with Prospector (a) Prospector-2D on latency and LUTs (fdtd-2d) (b) Distances between golden and estimated Pareto frontiers, normalized to that from Prospector.

that identify the regions within a multi-dimensional design space that would benefit most from additional measurements. These strategies and methods synthesize accelerators that use FPGA resources more efficiently and support reconfigurability in dynamic systems.

2 Evaluation

We evaluate Prospector's ability to find optimal design points and reveal the Pareto frontier. We first perform an exhaustive characterization of the design space, which runs every design point through HLS and place-and-route, measures execution time and FPGA utilization, and identifies the "golden" Pareto optima. We then determine how closely Prospector and alternative heuristics track these golden optima. Each iteration's time is dominated by synthesis and place and route, which is common across all other approaches. Thus, we run all approaches for equal number of iterations (i.e. 100) to be fair.

Figure 2(a) visualizes success of Prospector in optimizing latency and LUT usage simultaneously for fdtd-2d benchmark. Figure 2(a) shows how Prospector reveals the broad latency-LUT Pareto frontier by trying only 100 out of 6561 samples.

Prospector is capable to optimize multiple objectives. We use Prospector to optimize all cost measures and latency of accelerators across different workloads. We assess goodness by measuring the distance between two Pareto frontiers. We calculate the Euclidean distance from every design on the source frontier to the closest point on the destination frontier, producing a number of distances equal to the number of design points in the source. Note that each design point is represented by a five-dimensional vector that quantifies latency and usage for FFs, LUTs, BRAMs and DSPs. Our measures build on related work that explored several indicators to evaluate multi-dimensional Pareto frontiers [1].

Figure 2(b) reports average distances, normalized to the distance between golden and Prospector frontiers. Prospector most accurately reveals the Pareto frontier and reports the shortest distances to the golden frontier. Optimization in fewer dimensions is less accurate and reports greater distances to the golden frontier.

References

[1] P. Bosman and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2003.

[2] J. Cong, Y. Fan, G. Han, W. Jiang, and Z. Zhang. Platform-based behavior-level and system-level synthesis. In *Proc. International SOC Conference (SOCC)*, 2006.

[3] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang. High-level synthesis for fpgas: From prototyping to deployment. *IEEE Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 2011.

[4] B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 2016.