Scalable and Portable Architecture for Probability Density Function Estimation on FPGAs

Karthik Nagarajan, Brian Holland, Clint Slatton, Alan D. George
NSF Center for High-Performance Reconfigurable Computing (CHREC)
ECE Department, University of Florida
E-mail: {nagarajan,holland,slatton,george}@chrec.org

1. Introduction

Significant success has been achieved in FPGA application research by selecting, developing, and testing diverse applications from various fields. The Parzen window-based, non-parametric estimation algorithm for Probability Density Function (PDF) is known to suffer from super-linear increases in computational time with increasing data size and number of signals being processed. It forms an integral part of numerous machine-learning applications and hence fast and efficient execution of the algorithm is highly desirable. Different FPGA architectures (i.e. diverse platforms) offer varying degrees of performance for different applications. Thus, it is best to determine at a preliminary stage of system design the maximum performances attainable in migrating a specific algorithm to a variety of FPGA platforms. Along this line, this paper describes the design, development, and analysis of a scalable and portable architecture for multi-dimensional, non-parametric PDF estimation using Gaussian kernels on FPGAs.

2. Algorithm Overview

The computational complexity of the Parzen-window algorithm [1] is of order $O(Nn^d)$, where N is the total number of data points, n is the number of discrete points at which the PDF along a dimension are estimated (i.e. bins), and d is the number of dimensions. The data flow in the algorithm follows an exhaustive data permutation pattern wherein all of the input data affects the value of every output data point. Mathematically, the probability that point i falls in a d-dimensional space is given by

$$p_i = \frac{\sqrt{2\pi}\,h^{-d}}{n_1..n_d}\sum_{j_1=1}^{n_1}..\sum_{j_2=1}^{n_d} e^{\frac{-(x_i-x^{j_1})^2}{2h^2}}\;..e^{\frac{-(y_i-y^{j_2})^2}{2h^2}} \qquad (1)$$

where h is a tuning parameter of the algorithm, $(x^j,\ldots y^j)$ denotes the bin at which the PDF is estimated, and $(x_i,\ldots,y_i)$ denotes the input in a d-dimensional space. To make the algorithm more suitable to hardware, a second-order Taylor series expansion is applied to the exponential function. Specifically, the expression for the 1-D PDF is given in Eq. 2.

$$p_i = \frac{1}{\sqrt{2\pi}\,hn}\sum_{j=1}^{n}\left(1-\frac{(x_i-x^j)^2}{2h^2}\right) \qquad (2)$$

3. FPGA Core Design

After understanding the PDF estimation algorithm, an FPGA core design to efficiently perform the algorithm can be crafted. The general architecture of the 1-D PDF estimation algorithm is highlighted in Fig. 1. A kernel computes the PDF value for a particular data sample x_i at a particular point x^j (i.e. at a particular bin). Parallel pipelines for k such kernels are created. Each kernel is *seeded* differently and performs the same set of operations on every input data sample independently. Load balancing and reducing communication and synchronization requirements are needed to ensure that hardware outperforms its sequential software counterpart. The Parzen window technique is an embarrassingly parallel algorithm that has no communication between kernels and requires minimal synchronization.

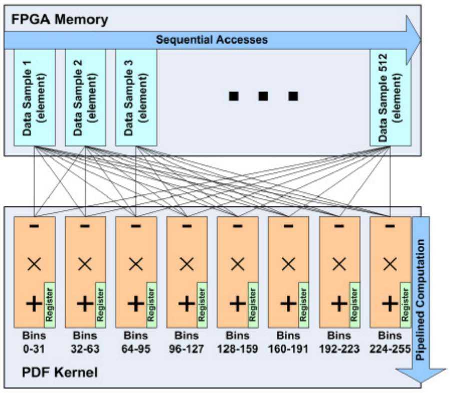

Figure 1. Architecture of 1-D PDF algorithm

This work was supported in part by the I/UCRC Program of the National Science Foundation under Grant No. EEC-0642422. The authors also gratefully acknowledge vendor equipment and/or tools provided by Xilinx, Cray, and Nallatech that helped make this work possible.

This trait allows the possibility to distribute data samples across kernels that can process data in a parallel fashion (see Fig. 1). In estimating multi-dimensional PDFs, operations in each dimension are independent of the others and hence performed in a parallel fashion within each pipeline. Internal registering for each bin keeps a running total of the impact of all processed data. These cumulative totals comprise the final estimation of the PDF function.

The RC Amenability Test (RAT) [2] is a methodology developed for predicting the performance (*speedup*) of a specific application design on a specific platform and features two important steps. The first step deals with estimating the communication burden (t_{comm}) involved in transferring data in and out of the FPGA. The second step involves the estimation of time spent in performing computation (t_{comp}) over the data transferred under an assumed FPGA clock frequency. The software baseline for computing speedup values was measured from an optimized C program executed on a 3.2 GHz Xeon processor using single-precision floating point. The predicted speedups for the 1-D and 2-D PDF algorithms were 10.5 and 6.8, respectively.

4. Results

The experimental platform primarily used in these experiments was a Nallatech H101 board with a Xilinx Virtex-4 LX100 FPGA. The board communicates with the host processor over a PCI-X interconnect. A 32-bit middleware communication channel is used of which 9 bits were allocated for the fixed-point fractional segment. The single-core 1-D and 2-D PDF designs operated at FPGA clock speeds of 150 MHz and 100 MHz, respectively. The number of kernels, *k*, in the core and the number of data samples per transfer were set to 8 and 512, respectively. It can be seen from the comparisons made in Table 1A that the actual speedups obtained are reasonably close to the predicted speedups. The deviation between the values is primarily due to the inaccurate estimate of t_{comm} values (variation in the interconnect channel efficiency for small data transfers). The t_{comp} predictions are relatively closer to the experimental values owing to the deterministic nature of the algorithm.

Dual-core architectures were developed and evaluated (see Table 1B) to study inherent characteristics for scalability on single- or multiple-device systems by replicating the single-core designs sharing the same interconnect. In addition to having scalability impacts, application characteristics also contribute greatly to platform selection and performance. The *utilization factors* in RAT indicate the nature of a particular design in terms of whether it is communication-bound or computation-bound and aid in the selection of an ideal platform. While platforms with FPGAs having more multipliers (e.g., the Cray XD1 with Virtex-2 Pro FPGAs) and faster interconnects (e.g., RapidArray in XD1) might intuitively suggest improvements in speedup by performing more computations in the PDF algorithm in parallel, these could be negated by poor read and write efficiencies during data communication. This case was particularly true with the Cray XD1 system (see Table 1C). Although it housed a theoretically faster interconnect, significantly low read speeds (~4MB/s for small data transfers) led to a small speedup in the 2-D PDF design.

Table 1. Speedup values for A) single-core designs, B) dual-core designs, and C) designs across platforms

A	Predicted		Experimental	
	1-D	2-D	1-D	2-D
t_{comm}(sec)	6.0E-6	1.6E-3	2.5E-5	1.1E-2
t_{comp}(sec)	1.3E-4	5.6E-2	1.4E-4	6.5E-2
Speedup	**10.5**	**6.8**	**7.8**	**5.2**

B	Single-core		Dual-core	
	1-D	2-D	1-D	2-D
Speedup	**7.8**	**5.2**	**13.4**	**7.2**

C	Nallatech		Cray XD1	
	1-D	2-D	1-D	2-D
Speedup	**7.8**	**5.2**	**20.6**	**4.0**

5. Conclusions

Significant performance improvements in terms of speedup were obtained from our design for the Parzen window-based PDF estimation algorithm on FPGAs. Dual-core architectures were developed on a single-device system by exploiting scalability in the algorithm. Key design parameters were identified for tuning the architecture to suit different platforms. We believe that with the successful design of a scalable and portable architecture for the PDF algorithm, rapid development of hardware designs for similarly patterned algorithms (e.g. k-means clustering, correlation functions, and information theoretic measures) is possible by investigating and generalizing design patterns.

6. References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, "*Pattern Classification*," 2nd edition, John Wiley & Sons, Inc., 2001.

[2] Brian Holland, Karthik Nagarajan, Chris Conger, Adam Jacobs, and Alan D. George, "RAT: A Methodology for Predicting Performance in Application Design Migration to FPGAs," *Proc. of High-Performance Reconfigurable Computing Technologies & Applications Workshop (HPRCTA 2007)*, SC'07, Reno, NV, Nov. 11, 2007.