2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines

A Scalable FPGA-based Accelerator for High-Throughput MCMC Algorithms

Morteza Hosseini, Rashidul Islam, Amey Kulkarni and Tinoosh Mohsenin
Department of Computer Science & Electrical Engineering
University of Maryland, Baltimore County

I. Introduction

Markov Chain Monte Carlo (MCMC) algorithms are used to obtain samples from any target probability distribution, and are widely employed in stochastic processing techniques. In a conventional MCMC sampler, the algorithm initiates with a random sample, $x^{(1)}$, and the next samples are then generated based on a transition probability that can be described by a 1^{st} order Markov chain as shown in equation 1:

$$p(x^{(i)}|x^{(i-1)}, x^{(i-2)}, ..., x^{(1)}) = p(x^{(i)}|x^{(i-1)}) \quad (1)$$

One of the challenges of MCMC acceleration in hardware is the sequential nature of Markov chain where the generation of every sample can be achieved only if the previous sample is available.

II. Proposed Technique

Parallel Tempering (PT) algorithm is one of the most powerful MCMC samplers that has proven better mixing and convergence for high-dimensional and multi-modal distributions compared to other popular MCMC algorithms. In this work, we impose a new parameter, D, to the transition probability of the PT algorithm, that gets it to generate the next sample based on the previous D^{th} sample, rather the one sample before. In this case the new algorithm should be initialized to D random samples. The probability transition of the new algorithm, named Multiple Parallel Tempering (MPT), can be described by:

$$p(x^{(i)}|x^{(i-1)}, x^{(i-2)}, ..., x^{(i-D)}, ..., x^{(1)}) = p(x^{(i)}|x^{(i-D)}) \quad (2)$$

Equation 2 is representable by D independent partitioned subsets of the set of variable x. Each subset is inherently a 1^{st} order Markov chain. The bottleneck of the sequential behavior when implementing an MCMC algorithm on hardware, can be resolved by choosing an appropriate D, which can be architecture and application specific.

III. Hardware Implementation and Results

In this work, the probability distribution from which we intend to sample, is a 4-dimensional i.i.d GMM and can have up to 16 modes. In order to design an MPT sampler for the given $p(x)$, the architecture is initially designed from a PT point of view, where it has two streams of data that can be assumed as a clockwise circular flow: forward and backward dataflows. The forward dataflow, whereby every next sample is chosen between either a previous sample or a proposed sample, is designed as a purely combinational logic. The proposed sample is made by summing the previous sample and a Gaussian random number. The designed PT architecture can then be converted to an MPT by applying D pipeline stages to the forward dataflow. We explore to find the smallest D that results in the maximum system frequency and a low hardware overhead. By doing so, the throughput of both the system and each independent PT kernel inside the MPT reaches to their maximum amount. With a trial and error method and placement of pipeline stages at different locations of the forward dataflow of the PT architecture, we determine the near-optimal values of D for each MPT configuration.

TABLE I: Post Place and Route Implementation results on Artix-7 for PT 1-8 and MPT 1-8 architectures in terms of resource utilization and throughput. Each implementation is performed with 18-bit width.

Architecture-Chain	PT-1	**MPT-1**	PT-2	**MPT-2**	PT-8	**MPT-8**
Slice Count	2,757	**3,083**	6,579	**6,948**	23,241	**25,755**
Register Count	66	**14,726**	133	**29,851**	969	**119,439**
Memory (BRAM)	1	**1**	2	**2**	8	**8**
No. of Pipelines "D"	1	**45**	1	**48**	1	**54**
Throughput (Msps)	5.04	**156.49**	4.70	**147.4**	4.09	**114.60**
Thr. Improvement	Base	**31.05**	Base	**31.36**	Base	**28.02**

The proposed reconfigurable PT and MPT architectures with 1, 2, and 8 number of chains are implemented using Verilog HDL, and synthesized and placed & routed on tiny and low power Artix-7 FPGA using Xilinx ISE tools. Table I shows the D value, device utilization, and throughput for each configuration. At the expense of respectively 6%, 11%, and 44% extra register utilization of the FPGA registers, for MPT 1-8 configurations, an average speedup of 30× in throughput is achieved as compared to the analogous PT 1-8 configurations.

IV. Conclusion

Inspired by a special case of D^{th} order Markov chains, we added a new parameter, D, to the PT algorithm, and proposed MPT algorithm that manages to increase the sampling throughput over a given $p(x)$, to near the maximum frequency achievable by the targeted FPGA. Compared to our PT sampler, the MPT sampler yields 31×, 31×, and 28× speedup in throughput for chain number 1, 2, and 8 configurations, respectively.

References

[1] A. Kulkarni, C. Shea, H. Homayoun, and T. Mohsenin, "Less: Big data sketching and encryption on low power platform," in *2017 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2017.

[2] A. Page, A. Jafari, C. Shea, and T. Mohsenin, "Sparcnet: A hardware accelerator for efficient deployment of sparse convolutional networks," *ACM Journal on Emerging Technologies in Computing (JETC),2017.*

978-1-5386-4037-1/17 $31.00 © 2017 IEEE
DOI 10.1109/FCCM.2017.56