Title	Optimized-FDanQ: Implementation of Hybrid Neural Network "DanQ" on Cloud Multi-FPGA and its Optimization under Given Costs
Author(s)	稲葉，貴大
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17082
Rights	
Description	Supervisor:井口　寧，先端科学技術研究科，修士（情報科学）

Abstract

In this information society, the amount of data is rapidly increasing, especially in the field of Astronomy, Twitter, Youtube, and Genomics. In these four fields, storage for genomics is increasing the most, and the way to process it fast has been one of the big tasks for a long time. There are so many DNA sequences that people have to work on. One of the genome analysis that people have to face is to find the function of DNA from a DNA sequence. Recently, Machine Learning has been used to find the function of DNA from a DNA sequence. However, training a machine learning model for DNA sequences takes much time due to the size of the dataset. Besides, since DNA sequences are represented by four types of the base, which are Adenine, Guanine, Thymine, and Cytosine, it can be represented by the bit-width of two. FPGA has a substantial advantage of processing these kinds of string operations because FPGA can construct a dedicated state machine. Also, FPGA can be a useful resource for processing fast by pipelining.

In addition, more and more companies are using cloud services such as AWS for their acceleration. Since cloud users always have to consider the trade-off between execution time and cloud instance usage fee, it is necessary to optimize these two things depending on each cloud user.

In this paper, we propose the following two ideas.

- Mutli-FPGA Implementation
- Cloud Optimization under Give Costs

We tried to accelerate a deep learning model called DanQ using FPGAs. It is said that FPGA is sufficient for data such as genomics data because DNA sequence can be represented by 1 bit and does not require a large bit-width for processing. We mainly focused on a BiLSTM layer, which is the most time-consuming part of the DanQ model. We quantized the parameters of the BiLSTM layer to the bit width of 16 in order to implement on FPGA without losing the training accuracy. We also implemented the BiLSTM layer to multiple FPGAs to obtain a better execution time. As a result, we could accelerate the DanQ model by using a single FPGA by 1.05x compared to our CPU implementation. Besides, our implementation on 8 FPGAs gets 2.87x faster than the dual FPGA implementation and 6.00x faster than the CPU implementation.

Also, our implementations can change the resource size during the execution to optimize the execution time or cloud instance usage fee depending on the users' needs. Comparing a case of using 8 FPGAs for all time and a case in which we optimized the number of FPGAs during the training with our model, we obtained the result that we can save the cloud usage fee for 56.28% by only taking 16.00% extra time.