Scalable Full Hardware Logic Architecture for Gradient Boosted Tree Training

Tamon Sadasue
RICOH Company
Kanagawa, Japan
tamon.sadasue@jp.ricoh.com

Tsuyoshi Isshiki
Tokyo Institute of Technology
Meguro, Tokyo
isshiki@ict.e.titech.ac.jp

***Abstract*—Gradient Boosted Tree is most effective and standard machine learning algorithm in many fields especially with various type of tabular dataset. Besides, recent industry field and robotics field require high-speed, power efficient and real-time training with enormous data. FPGA is effective device which enable custom domain specific approach to give acceleration as well as power efficiency. We introduce a scalable full hardware implementation of Gradient Boosted Tree training with high performance and flexibility of hyper parameterization. Experimental work shows that our hardware implementation achieved 11-33 times faster than state-of-art GPU acceleration even with small gates and low power FPGA device.**

I. Hardware Architecture

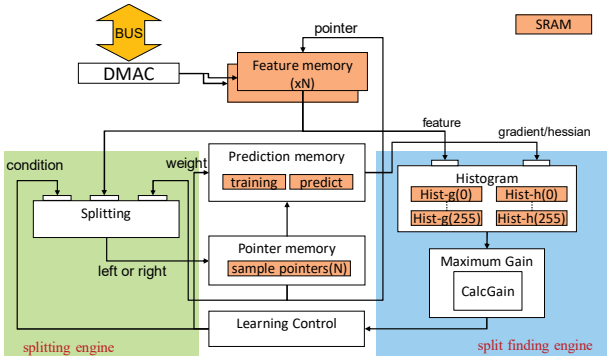

Fig. 1. Overview of System Pipeline Architecture

We divide training process to three stages: split finding stage for maximum gain search, split stage for tree making, and prediction update stage. Split finding engine, splitting engine and state memory module composed of Prediction and Pointer memory correspond to them respectively while each engine is fully pipelined architecture represented in Fig 1. We use histogram based approach (feature data binning) which can reduce the computing cost for sorting drastically as reported[1][2][3], and this approach suits well to FPGA. Besides, we employed two types of parallelism, feature level and data level while tree level parallelism is not applicable because a new tree depends on previous trees. About design methodology, we extended C based RTL design framework named C2RTL reported[4] for capability of not only module level but also system level hardware description with C/C++ codes. This enable highly generic description with variation of hyper parameters such as bit-width of variables, number of bins, equation for prediction value, and loss functions..

II. Results and Conclusion

We implemented generated RTL of typical binary classification GBT training case with logistic regression loss function to the target of Xilinx Kintex-7 UltraScale+ KCU116 evaluation board at 250MHz clock frequency. We compared training accuracy and performance on FPGA with software implementation of both CPU only and GPU accelerated version of 'xgboost 0.90' executed on Intel corei7-8700 and NVIDIA GeForce-RTX2060. Experimental result (TableI, TableII) is 11-33 times speed-up to the GPU implementation.

TABLE I. Training time for 10000 samples

Dataset	NumBins	CPU	CPU+GPU	FPGA	Speed-up	
Higgs	256	8.67	3.85	0.212	40.9	18.2
	64	4.41	3.66	0.0726	38.3	31.8
Airplane	256	3.67	3.63	0.134	17.3	17.1
	64	2.95	3.85	0.0726	25.7	33.5

TABLE II. Training time for 100000 samples

Dataset	NumBins	CPU	CPU+GPU	FPGA	Speed-up	
Higgs	256	24.6	9.12	0.825	29.8	11.1
	64	20.24	9.16	0.727	27.8	12.6
Airplane	256	15.54	8.8	0.825	18.8	10.7
	64	14.59	8.21	0.727	20.1	11.3

References

[1] T. Chen, C. Guestrin, "Xgboost: A scalable tree boosting system", CoRR, vol. abs/1603.02754, 2016, [online] Available: http://arxiv.org/abs/1603.02754.

[2] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree", Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS), pp. 3146-3154, 2017.

[3] R. Mitchell, A. Adinets, T. Rao, E. Frank, "Xgboost: Scalable GPU accelerated learning", CoRR, vol. abs/1806.11248, 2018, [online] Available: http://arxiv.org/abs/1806.11248.

[4] T. Isshiki, K. Date, D. Kugimiya, D. Li, H. Kunieda, "C-Based RTL Design Framework for Processor and Hardware-Il' Synthesis", Proc. of SASIMI 2015, pp. 40-45, 2015.

2576-2621/20/$31.00 ©2020 IEEE
DOI 10.1109/FCCM48280.2020.00067