Performance Modeling and Scalability Analysis of Stream Computing in ESSPER FPGA Clusters

Ryota Miyagi*, Ryota Yasudo**, Kentaro Sano***, Hideki Takase*
*The University of Tokyo, **Kyoto University, ***RIKEN

Objective and Motivation

Current Work: Accelerating structure learning of Bayesian networks
Achievement: Single FPGA-based accelerator [1]
Next step: Scaling up the approach using Multiple-FPGA
😍 Cascaded Streaming PEs enhance performance per memory access
😍 Direct inter-FPGA streaming preserves scalability
😳 Stalls may occur due to limited inter-FPGA bandwidth
😳 Extended pipeline overhead and communication latency

🤔 **Concern**: Does it really scale?

Contribution

- Development a performance model for stream computing in 1D-ring cascaded FPGAs based on [2]
- Demonstration of model accuracy and applicability through practical application on ESSPER FPGA Cluster

😍 The performance model provides performance estimation and optimization guidance and facilitates other stream computing.
😍 The performance model confirms the scalability of our Multi-FPGA accelerator for Bayesian network structure learning in ESSPER.

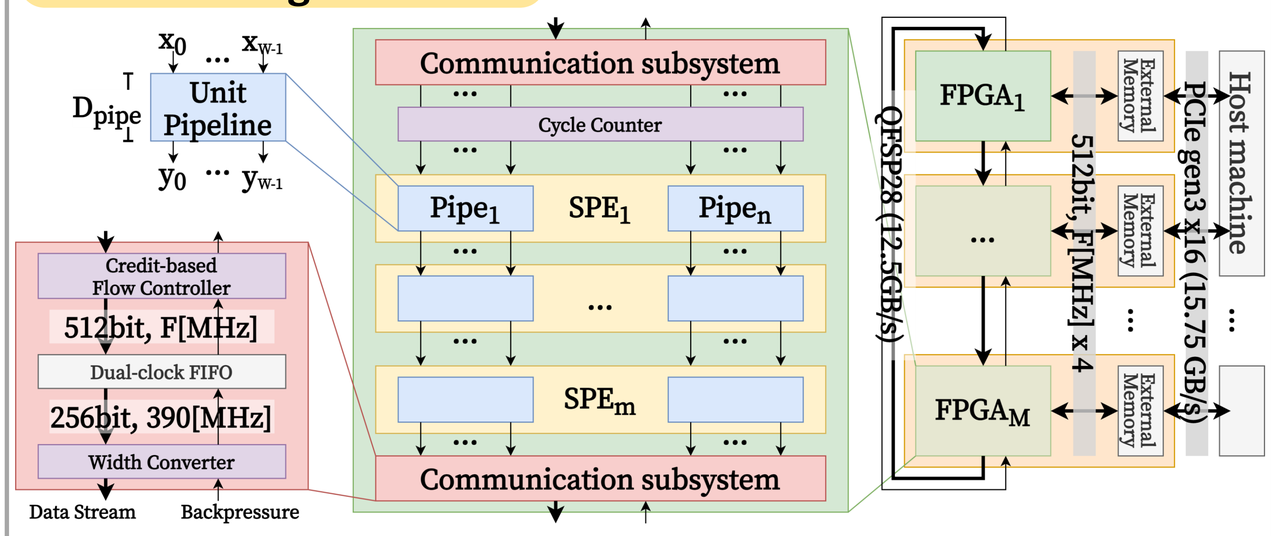
ESSPER: FPGA Cluster [3]

Extension for the Fugaku supercomputer, a superior alternative for highly scalable computing
(e.g., numerical simulations, machine learning)

ESSPER's Specification:

- **Servers**: 8 x86 FPGA servers, each with 2 Intel PAC D5005 (16 total).
- **Intel PAC D5005**: Stratix 10 GX FPGA, 32GB DDR4 (4x8GB), PCIe Gen3 x16, dual 100Gbps QSFP28 ports
- **Stratix 10 GX FPGA**: 2753K LEs, 229 Mb BRAMs, 5760 FP DSPs
- **Dedicated inter-FPGA network** with QSFP28 for arbitrary topology
- **AFUShell SoC**: In-house system stack that provides core functions including DMAs, crossbar, and inter-FPGA network interface with credit-based flow control

Model Settings and Score



- **1D-Ring Topology**: Master FPGA controls the data stream, and slave FPGAs passively process and pass data sequentially
- **Inter-FPGA Communication:** Flow controllers provide back pressure to stall the pipeline in case of insufficient bandwidth.
- **Streaming process element, SPE**: Algorithm-specific computing core, working in parallel and exploiting data-level parallelism
- **Parallelism:** M-cascaded FPGAs, each of which has m-cascaded SPEs with n-parallelized unit pipelines
- **Homogeneity**: Uniform computing capabilities across all SPEs.
- **Data Movement**: Restricted to intra-SPEs or adjacent inter-SPEs.
- **External Memory Access**: Fixed-width sequential read/write

Performance Modeling

Total number of stream cycles: Since n-parallelized pipelines process n data in a cycle, total number of stream cycles $C_{\text{stream}}(n)$ is given by

$$C_{\text{stream}}(n) = \left\lceil \frac{N}{n} \right\rceil$$

, where N is the number of input data.

Total number of operations: Since $n \cdot m \cdot M$ unit pipelines process all inputs, the total number of operations $O(n, m, M)$ is given by

$$O(n, m, M) = n \cdot (m_{\text{master}} + (M-1) \cdot m_{\text{slave}}) \cdot O_{\text{pipe}} \cdot C_{\text{stream}}$$

where O_{pipe} is the number of operations per unit pipeline

Stall and utilization rate: Stall occurs if bandwidth is insufficient.

- Let $B_{\text{mem}}/b_{\text{mem}}$ be the Available/Required memory bandwidth.
- Let $B_{\text{link}}/b_{\text{link}}$ be the Available/Required inter-FPGA link bandwidth.

In each section of the pipeline, the utilization rate $(1 - \text{r})$ is given by

$$1 - r = \min\left(\frac{B_{\text{link}}}{b_{\text{link}}}, \frac{B_{\text{mem}}}{b_{\text{mem}}}, 1\right)$$

The total utilization u is bottlenecked by the lowest one and given by

$$u = \min(1 - r_{\text{master}}, 1 - r_{\text{slave}})$$

Total number of cycles: The total propagation cycle is the sum of the number of input streams and pipeline latency and is given by:

$$C_{\text{total}} = \frac{C_{\text{stream}} + D_{\text{pipeline}}}{u}$$

$$D_{\text{pipeline}} = D_{\text{read}} + D_{\text{write}} + (m_{\text{master}} \cdot D_{\text{pipe}} + D_{\text{link}}) + (m_{\text{slave}} \cdot D_{\text{pipe}} + D_{\text{link}}) \cdot (M-1)$$

D_{pipe} and D_{link} are the delays of a unit pipeline and inter-FPGA link delay. $D_{\text{read}}/D_{\text{write}}$ are the memory read/write delay of the start/end of stream.

The total performance:

$$P(n, m, M) = \frac{(\text{Total numbetr of operations})}{(\text{Total computing time})} = P_{\text{peak}} \cdot u \cdot \frac{1}{1 + \frac{D_{\text{pipeline}}}{C_{\text{stream}}}}$$

$$P_{\text{peak}} = n \cdot (m_{\text{master}} + (M-1) \cdot \text{m}_{\text{slave}}) \cdot \text{F} \cdot O_{\text{pipe}}$$

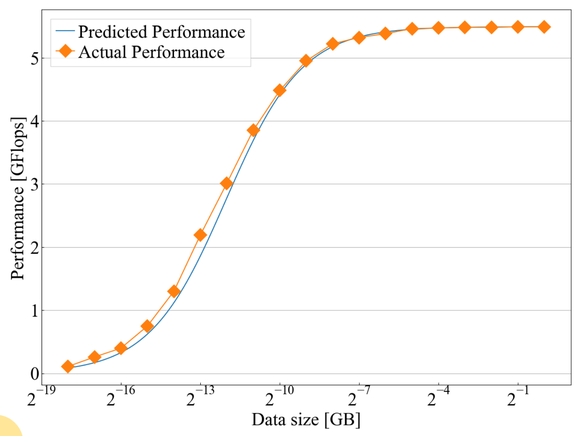
Performance factors:

- Utilization rate: $0 < u < 1$
- Pipeline overhead, reduced with larger data stream: $0 < \frac{1}{1 + \frac{D_{\text{pipeline}}}{C_{\text{stream}}}} < 1$

Validation of Performance Model

- 1D-ring of 2-cascaded FPGA
- 16 parallel 4B floating-point add
- Intel HLS Compiler21.1
- Quartus Prime Pro 21.1
- Master (266MHz) / Slave (284 MHz)

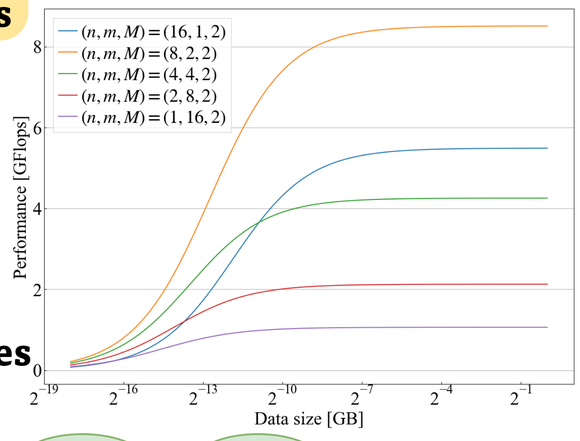
😍 **Results support the performance model's correctness!**



Virtual Performance Optimizations

- Fixed-Resource Comparison
- 2 FPGAs (i.e., $M = 2$)
- 32 SPEs in total (i.e., $n \cdot m \cdot M = 32$)
- Inter FPGA link bottlenecked: B
- Close to peak performance: O
- Memory bottlenecked: G, R, P

😍 **The performance model facilitates performance optimization**

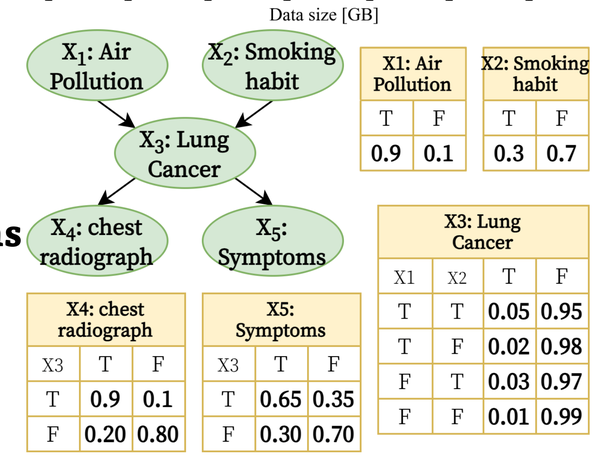


Future Work

WIP: Multi-FPGA accelerator for Bayesian network structure learning

😍 **The performance model confirms the scalability of it in ESSPER**

🚀 Accelerate the Development



X1: Air Pollution	
T	F
0.9	0.1

X2: Smoking habit	
T	F
0.3	0.7

X4: chest radiograph		
X3	T	F
T	0.9	0.1
F	0.20	0.80

X5: Symptoms		
X3	T	F
T	0.65	0.35
F	0.30	0.70

X3: Lung Cancer			
X1	X2	T	F
T	T	0.05	0.95
T	F	0.02	0.98
F	T	0.03	0.97
F	F	0.01	0.99

Acknowledgments

We sincerely thank Mr. Tomohiro Ueno and Mr. Emanuele Del Sozzo of RIKEN R-CCS for their expert guidance on FPGA clusters, which has been invaluable to our research.
A part of this work was supported by JST CREST (JPMJCR21D2) and JST SPRING (JPMJSP2108).

References

[1]R. Miyagi, R. Yasudo, K. Sano, and H. Takase, "Elastic sample filter: An fpga-based accelerator for bayesian network structure learning," FPT 2022, vol. 5, p. 310, 2022.
[2] A. Mondigo, T. Ueno, K. Sano, and H. Takizawa, "Scalability analysis of deeply pipelined tsunami simulation with multiple fpgas," IEICE TRANSACTIONS on Information and Systems, vol. 102, no. 5, pp. 1029–1036, 2019.
[3] K. Sano, A. Koshiba, T. Miyajima, and T. Ueno, "Essper: Elastic and scalable fpga-cluster system for high-performance reconfigurable computing with supercomputer fugaku," in Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, pp. 140–150, 2023.