An FPGA Accelerator of Bayesian Network Structure Learning Using Parallel Calculation of Local Scores

Ryota Miyagi* Hideki Takase**

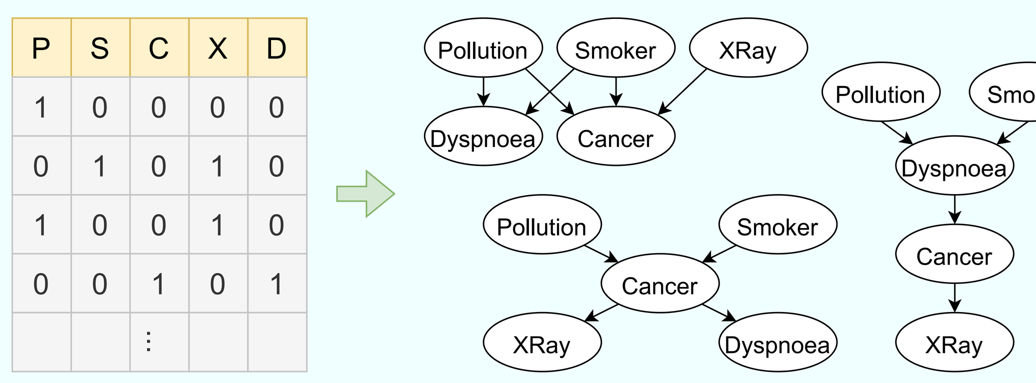
*Kyoto University **The University of Tokyo

Bayesian Networks

- Probabilistic graphical model that encodes conditional independence relations among random variables using DAG.
- Application in various fields (e.g. medical diagnosis, financial analysis, genetic phylogenetic analysis, gene sequence analysis, etc.)

Structure Learning

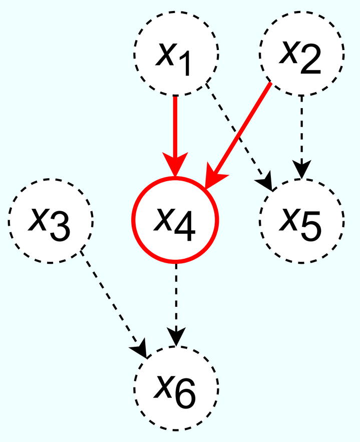
- Learning DAG structure of BN from data
- However, optimal structure learning is NP-hard and time-consuming.



Local Scores

- Structure learning is reduced to a combinatorial optimization that maximizes the log marginal likelihood score of the entire graph.
- The entire graph score is decomposed into local scores.

$$
\begin{aligned}
s(\mathcal{D},\mathcal{G}) = & \; LocalScore(x_1, \emptyset, \mathcal{D}) \\
& + LocalScore(x_2, \emptyset, \mathcal{D}) \\
& + LocalScore(x_3, \emptyset, \mathcal{D}) \\
& + LocalScore(x_4, \{x_1, x_2\}, \mathcal{D}) \\
& + LocalScore(x_5, \{x_1, x_2\}, \mathcal{D}) \\
& + LocalScore(x_6, \{x_3, x_4\}, \mathcal{D})
\end{aligned}
$$



- Calculating local scores in advance simplifies the evaluation of the entire graph.
- However, The time to calculate a huge number of local scores is critical.

Research Objective and Our Approach

Research Objective

Accelerate local score calculation for large BNs structure learning with

- Domain-specific dataflow architecture using FPGAs
- Parallelization by utilizing FPGA resources
- Scalable implementation for FPGA clusters

Our Approach

Each local score calculation depends on the entire dataset.
It is impossible to store a vast dataset for each local score calculation module.
However, storing it in one place will cause memory contention.
→ Dataflow architecture with FPGA

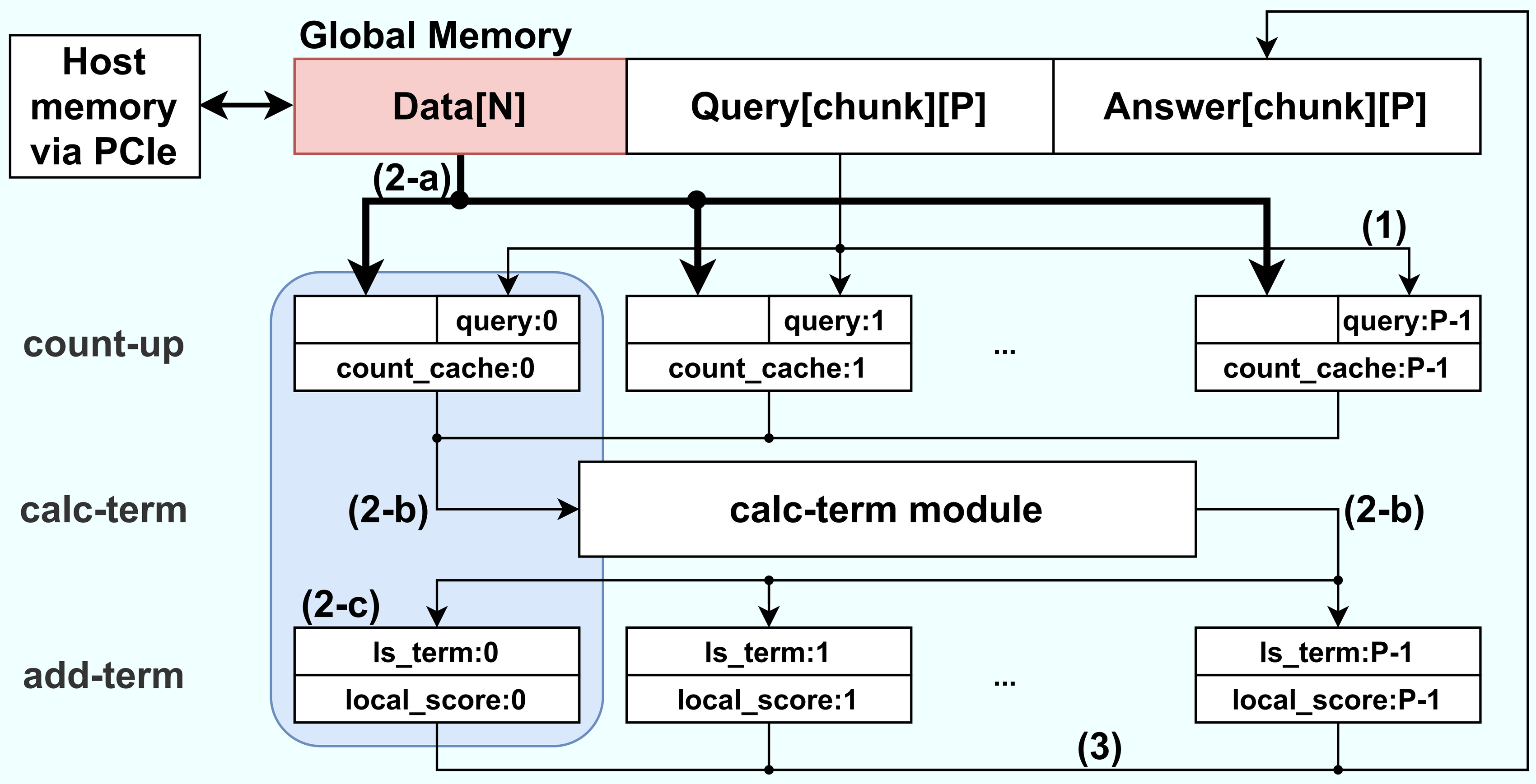
Architecture

We place parallel calculation modules according to FPGA resources.
The dataset is stored in one place and streamed to each module.
Each module counts data supporting each target substructure concurrently.

- High degree of data and pipeline parallelism with few memory resources
- Scalability : performance improves as FPGA resources increase

Calculation Flow

(1) Each count-up module identifies the target local score as a query.
(2) Iterate the following three steps for all combinations of parent variable values.
  (2-a) Counts the data supporting each target substructure from the streaming data concurrently.
  (2-b) Calculate the term for each local score in the calc-term pipeline based on counted numbers.
  (2-c) Add each term calculated in the calc-term pipeline to each partial sum.
(3) Return the calculated local scores as an answer.



Log-Gamma Function Calculation

Each local score calculation requires log-gamma function of given value.

Lanczos Approximation

- Calculate the log-gamma function numerically with High accuracy
- Consist of only the constants and elementary functions

→ Ideal for FPGAs to calculate the log-gamma function

Pipeline

- Despite the Lanczos approximation, the floating-point log-gamma function calculation module still consumes many DSP resources in FPGAs.
- Therefore, each parallel calculation module shares the pipelined log-gamma function calculation module to save DSP resources.

→ The upper limit of parallelism breaks free from DSP resource constraints.

Evaluation

Environment

- Intel Xeon W-2265 / 64GB / Ubuntu 18.04 / Xilinx Alveo U50
- BN with 30 binary random variables
- Accelerator is designed in C/C++ using Vitis 2020.2
  - SW: single-core software execution
  - HW(P = 1024): using accelerator with parallelism 1024
  - HW(P = 2048): using accelerator with parallelism 2048

Synthesis Results

- Few BRAMs and DSPs, but many LUTs

Resource		LUT	LUTRAM	FF	BRAM	DSP
Available		870016	402016	1740032	1344	5940
Usage	P=1024	324129	16861	441186	213	490
	P=2048	476544	12417	656236	200	490
Utilization(%)	P=1024	37.26	4.19	25.36	15.85	8.25
	P=2048	54.77	3.02	37.71	14.88	8.25

Performance Evaluation of the Accelerator

- HW accelerates the calculations, thus enabling too time-consuming calculations for software (N/A: terminated after 18,000 s).
- Comparison of HWs proves that the performance improves as the FPGA resources increase.

N	m	SW	HW(P=1024)	HW(P=2048)
1000	5	130.059	1.824	1.667
	6	949.879	14.270	12.987
	7	5930.838	93.125	84.612
	8	N/A	510.549	463.556
	9	N/A	2378.568	2158.771
10000	5	1268.540	7.552	5.010
	6	9175.792	60.080	39.712
	7	N/A	394.153	260.212
	8	N/A	2166.152	1429.310
	9	N/A	10104.604	6665.651

N: data size, m: number of parent variables.

Conclusion and Future Work

Conclusion

- Calculate local scores in parallel using dataflow architecture with FPGA.
- Extract high parallelism with few memory resources by streaming the dataset.
- Scalability : performance improves as FPGA resources increase

Future Work

- Practical evaluation on FPGA cluster, such as ESSPER by RIKEN