Find the Number of Islands
Term Project Presentation

Berker Acır
berker.acir@metu.edu.tr

Wireless Systems, Networks and Cybersecurity Laboratory
Department of Computer Engineering
Middle East Technical University
Ankara Turkey

June 18, 2022

Outline of the Presentation

Agenda

Find the Number of Islands

Variation of Counting the Number of Connected Components in an Undirected Graph

The problem is to find the number of islands in the given 2D boolean matrix. A cell in the matrix can be connected to 8 neighboring cells and a group of connected 1s forms an **island**.

1	0	1	1	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	1	1	0	1

(a) 1 island

1	1	0	1	0
0	1	0	0	1
1	0	0	1	1
0	0	0	0	0
1	0	1	0	1

(b) 5 islands

1	0	1	0	1
0	0	0	0	0
1	0	1	0	1
0	0	0	0	0
1	0	1	0	1

(c) 9 islands

Figure: 2D boolean input matrices and their island counts

Find the Number of Islands

Algorithm

The problem can be solved by applying **Depth First Search (DFS)** on each component. In each DFS call, a connected component is explored. Then, DFS is called on the next unexplored cell until there are none left. The number of DFS calls gives the number of islands.

- **Breadth First Search (BFS)** can be also used instead of DFS.
- Time complexity is $O(\mathbf{ROWxCOL})$.

Agenda

FHE Friendly Approach

Limitations of Homomorphic Encryption

Branching on ciphertext is impossible with Fully Homomorphic Encryption even though subset of the arithmetic operations (such as multiplication, addition, negetion, shift etc.) on ciphertext are allowed.

1. Calculate ***reduced matrix*** from input matrix (*Optional*)
2. Obtain ***adjacency matrix*** from input (or reduced) matrix
3. Calculate ***distances matrix*** from adjacency matrix
4. Count the number of islands from input (or reduced) matrix and adjacency matrix

Reduced Matrix (*Optional*)

This optional step is to **reduce the number of 1s in the input matrix** so that the client can count the islands with fewer computations.

Elements of reduced matrix R is created with the following computation from input matrix I:

$$R[i,j] = I[i,j].\left(1 + I[i-1,j-1] + I[i-1,j] + I[i-1,j+1] + I[i,j-1]\right)$$

1	1	0	1	0
0	1	0	0	1
1	0	0	1	1
0	0	0	0	0
1	0	1	0	1

→

1	2	0	1	0
0	3	0	0	2
2	0	0	2	3
0	0	0	0	0
1	0	1	0	1

(a) Input matrix with 11 ones and 5 islands is converted into reduced matrix with 5 ones.

1	0	1	1	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	1	1	0	1

→

1	0	1	2	2
2	0	3	0	3
2	0	2	0	2
2	0	2	0	2
2	4	3	0	2

(b) Input matrix with 17 ones and 1 island is converted into reduced matrix with 2 ones.

Figure: Input matrices and their conversion into reduced matrices

Adjacency Matrix

Input matrix needs to be converted into adjacency matrix in order to compute the distances between each cells. Each cell in input matrix is treated as a node so that adjacency matrix with size n^2xn^2 is created from input matrix with size nxn. Cell (i, j) of the input matrix has the index value of $i.n + j$ in rows and columns of the adjacency matrix.

A	B
C	D

→

	A	B	C	D
A	0	b	c	d
B	a	0	c	d
C	a	b	0	d
D	a	b	c	0

(a) 2x2 matrix and its 4x4 adj. matrix

A	B	C
D	E	F
G	H	I

→

	A	B	C	D	E	F	G	H	I
A	0	b	0	d	e	0	0	0	0
B	a	0	c	d	e	f	0	0	0
C	0	b	0	0	e	f	0	0	0
D	a	b	0	0	e	0	g	h	0
E	a	b	c	d	0	f	g	h	i
F	0	b	c	0	e	0	0	h	i
G	0	0	0	d	e	0	0	h	0
H	0	0	0	d	e	f	g	0	i
I	0	0	0	0	e	f	0	h	0

(b) 3x3 matrix and its 9x9 adj. matrix

Figure: Input matrices and their conversions into adjacency matrices

Distances Matrix

Since rows of the adjacency matrix gives 1-hop distances information, it is possible to calculate distances from each node to every other reachable nodes. After distances are obtained, distances matrix is used to check whether there is a path from a node to other nodes.

Distances matrix is initialized with adjacency matrix, then the rows of distances matrix are updated as the follows:

$$\text{DIST}[r] = \sum_{\text{R}=1}^{\lceil N/2 \rceil} \sum_{\text{c}=1}^{N} \text{DIST}[r, c].\text{ADJ}[c]$$

Partial Calculation of Distances Matrix

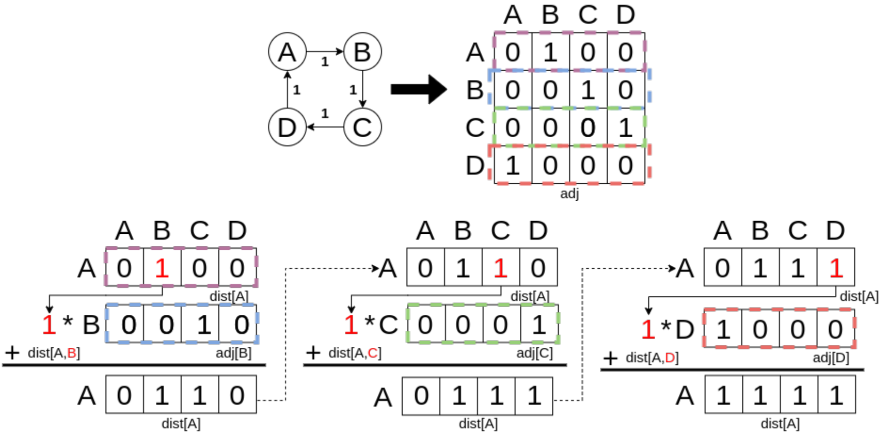


Figure: Directed graph with 4 nodes together with its adjacency matrix and partial calculation of its distances matrix for the first row (cell *A*).

Islands

The client has input matrix, or reduced matrix (optional) and distances matrix. Client can find the number of islands from these matrices as distances matrix has the connectivity information of every nodes.

1	0	1
0	0	1
1	1	0

input

1	0	1
0	0	2
1	3	0

reduced

	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
(0,0)	0	0	0	0	0	0	0	0	0
(0,1)	1	0	1	0	0	1	1	1	0
(0,2)	0	0	1	0	0	1	1	1	0
(1,0)	1	0	1	0	0	1	1	1	0
(1,1)	1	0	1	0	0	1	1	1	0
(1,2)	0	0	1	0	0	1	1	1	0
(2,0)	0	0	1	0	0	1	1	1	0
(2,1)	0	0	1	0	0	1	1	1	0
(2,2)	0	0	1	0	0	1	1	1	0

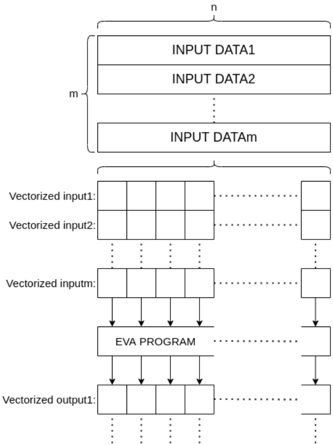
distances

Figure: Input matrix and its reduced version with distances matrix.

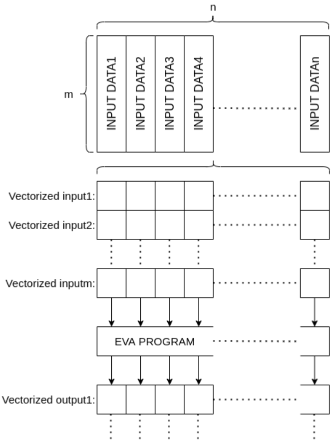
Note that 1 values in distances matrix means that there is a path/paths from the cell in the row to the cell in the column.

Agenda

Vectorized vs Parallelized Input



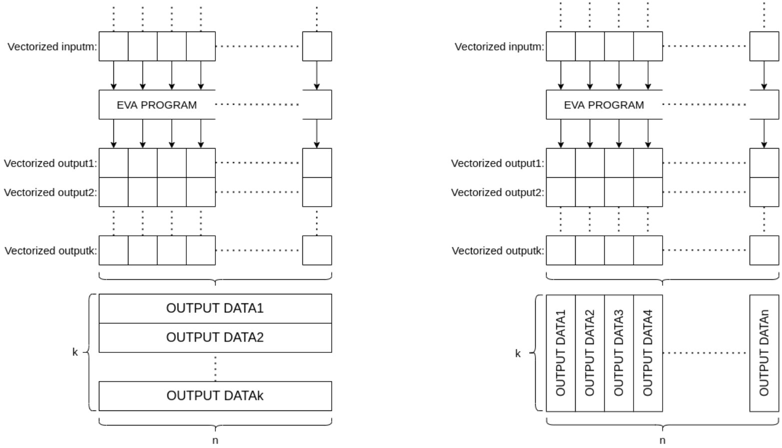
(a) Vectorized



(b) Parallelized

Figure: Vectorized vs Parallelized Inputs

Vectorized vs Parallelized Output



(a) Vectorized (b) Parallelized

Figure: Vectorized vs Parallelized Outputs

EVA Programs and Implementations

EVA Programs:

- **Reduce Ones:** Converts input matrix into reduced matrix
- **Compute Distances:** Computes distances matrix

Implementations:

- **Vectorized:** Vectorized inputs and outputs
- **Parallelized:** Parallelized inputs and outputs

Implementation Comparisons

Step\Implementation	**Vectorized**	**Parallelized**	**P. w/out Loops**
Reduce Ones	Server	Server	Server
Adjacency Matrix	Client	Server	Client
Distances Matrix	Server	Server	Server
Islands	Client	Client	Client

Table: Solution steps and where they are executed in the implementations

EVA Program\Impl.	**Vectorized**	**Parallelized**	**P. w/out Loops**
Reduce Ones	1	1	1
Compute Distances	$\lceil n^2/2 \rceil .n^2$	1	$\lceil n^2/2 \rceil .n^2$

Table: The number of times that EVA Programs are called in the implementations where n^2 is the size of input matrix/matrices

Implementation Comparisons

Imp.	Vectorized		Parallelized		P. w/o. Loops	
EVA Prog.	**Red.**	**Comp.**	**Red.**	**Comp.**	**Red.**	**Comp.**
Input S.	1	3	n^2	n^2	n^2	$2.n^4$
Vector S.	$2^{\lceil 2\log(n+2)\rceil}$	$2^{\lceil 4\log(n)\rceil}$	2^m	2^m	2^m	2^m
# of Inputs	1		2^m		2^m	

Table: Input sizes, vector sizes and number of inputs that are used in EVA Programs of the implementations where n^2 is the size of input matrix/matrices and 2^m is the number of different input matrices in the Parallelized Implementations

Agenda

Total Running Times

For only one input

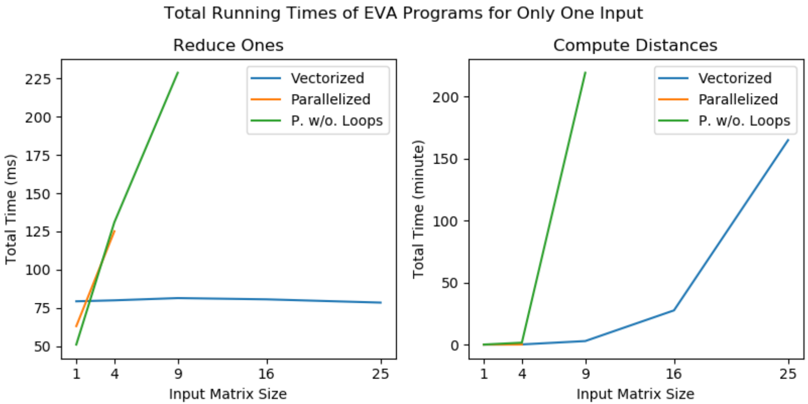


Figure: Total runtimes of EVA Programs for only one input

Total Running Times

Amortized analysis

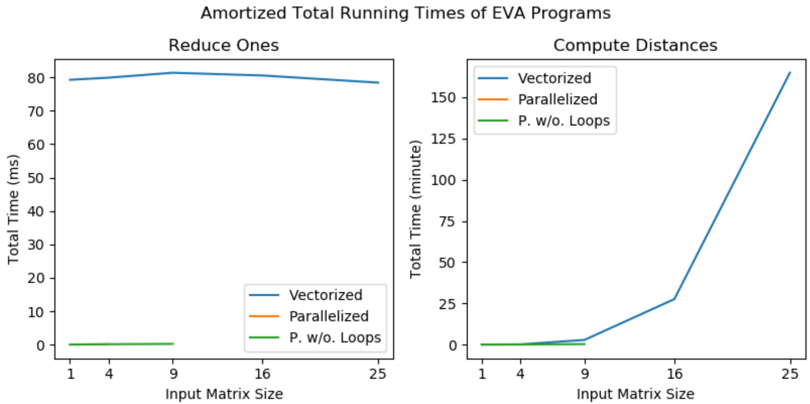


Figure: Amortized total runtimes of EVA Programs by implementations

Agenda

- Finding number of islands with FHE
- Optimizations
- Improvements

Questions

THANK YOU

Find the Number of Islands
Term Project Presentation

presented by Berker Acır
berker.acir@metu.edu.tr

June 18, 2022