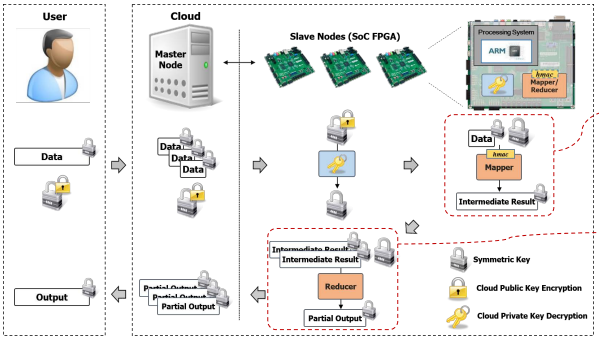
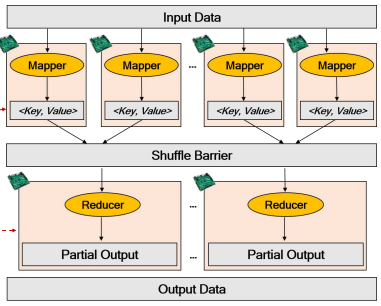
Optimizing Hardware-Based Privacy-Preserving MapReduce

Han-Yee Kim, Rohyoung Myung, Sangwoo Park, Sukyong Choi, Heonchang Yu, Taeweon Suh

Korea University

Introduction





Previous System

Map-Reduce on Cloud computing is good choice for big data processing. However, **data privacy becomes a big concern** in this situation.

In our previous work, we proposed a **hardware-based security concerned MapReduce framework**.
In the framework, for the security, all **data should be accommodated on memory as encrypted format**.
Data decryption, calculation with the plain-text, and output data encryption are conducted on FPGA.

In this paper, we polished and enhance the performance of our previous work by **optimizing the utilization of FPGA**.
For maximizing the utilization, we use two schemes for target applications: **Separate barrier scheme**, **Combining scheme**

Previous Work: Privacy preserving large scale dna read-mapping in mapreduce framework using fpgas, *24th FPL*

Proposed Schemes & its Experimental Results on Target Applications

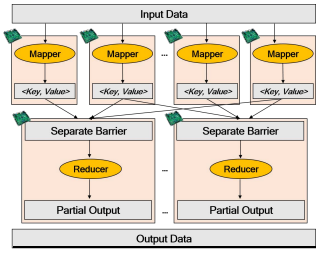
Target Applications & its Parameters

Applications	Parameters	
K-means	# Centroid (16, 32, 48)	# Dimension (4, 6, 8)
DNA sequencing	Block size (16, 32, 48)	String size (6, 8 , 10)

ALGORITHM 1. Integrated MapReduce Module Configuration Algorithm for Separate Barrier

D = # bits of AES decrypt module
E = # bits of AES encrypt module
i = 1;
j = 1;
$d_i = \lfloor D/(128*i) \rfloor$;
$e_j = \lfloor E/(128*j) \rfloor$;

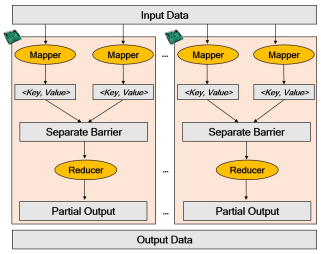
repeat
if (d_i*cost(AES decrypt) > e_j*cost(AES encrypt))
 i = i+1;
else
 j = j+1;
until (d_i*cost(AES decrypt) + cost(mapper or reducer) + e_j*cost(AES encrypt) < capacity)
configure SoC;



ALGORITHM 2. Integrated Combining Module Configuration Algorithm

M = # of mappers to one reducer
D = # bits of AES decrypt module
E = # bits of AES encrypt module
T_m = # of cycle for mapper
T_d = # of cycle of AES decrypt module
T_e = # of cycle of AES encrypt module
i = 1;
j = 1;
$d_i = \lfloor D/(128*i) \rfloor$;
$e_j = \lfloor E/(128*j) \rfloor$;
$m_i = \lfloor M/i \rfloor$;

repeat
if (d_i*cost(AES decrypt) + m_i*cost(mapper) + cost(reducer) + e_{j+1}*cost(AES encrypt) < capacity)
 $j = j+1$;
else if (d_{i+1}*cost(AES decrypt) + m_{i+1}*cost(mapper) + cost(reducer) + e_j*cost(AES encrypt) < capacity)
 $i = i+1$;
else
if $(\{(d_i - d_{i+1}) * cost(\text{AES decrypt}) + (m_i - m_{i+1}) * \text{cost(mapper)}\} / (T_m + T_d)$
 $< (e_j - e_{j+1}) * cost(\text{AES encrypt}) / T_e)$
 $j = j+1$;
else
$i = i+1$;
until (d_i*cost(AES decrypt) + m_i*cost(mapper) + cost(reducer) + e_j*cost(AES encrypt) < capacity)
configure SoC;



*Combining scheme is valid when there is no complicated shuffling between mappers and reducers so that they can be serialized by certain input data chunk

Applications	Modules	Parameters	FPGA Configuration
K-means	Mapper	4 Dimension	Fully Configured
		6 Dimension	Fully Configured
		8 Dimension	Fully Configured
	Reducer	16 Centroid	Fully Configured
		32 Centroid	AES Decrypt 1/2
		48 Centroid	AES Decrypt 1/2
DNA sequencing	Mapper	16 Block, 6 String	Fully Configured
		16 Block, 8 String	Fully Configured
		16 Block, 10 String	Fully Configured
		32 Block, 6 String	AES Encrypt 1/2
		32 Block, 8 String	AES Encrypt 1/2
		32 Block, 10 String	AES Encrypt 1/2
		48 Block, 6 String	AES Encrypt 1/2
		48 Block, 8 String	AES Encrypt 1/2
		48 Block, 10 String	AES Encrypt 1/2
	Reducer	16 Block, 6 String	AES Decrypt 1/2
		16 Block, 8 String	Fully Configured
		16 Block, 10 String	Fully Configured
		32 Block, 6 String	AES Decrypt 1/3
		32 Block, 8 String	AES Decrypt 1/3
		32 Block, 10 String	AES Decrypt 1/3
		48 Block, 6 String	AES Decrypt 1/4
		48 Block, 8 String	AES Decrypt 1/5
		48 Block, 10 String	AES Decrypt 1/5

Applications	Parameters	FPGA Configuration
K-means	4 Dimension, 16 Centroid	AES Decrypt, Map 1/3
	4 Dimension, 32 Centroid	AES Decrypt, Map 1/5
	4 Dimension, 48 Centroid	AES Decrypt, Map 1/7
	6 Dimension, 16 Centroid	AES Decrypt, Map 1/3
	6 Dimension, 32 Centroid	AES Decrypt, Map 1/6
	6 Dimension, 48 Centroid	AES Decrypt, Map 1/9
	8 Dimension, 16 Centroid	AES Decrypt, Map 1/4
	8 Dimension, 32 Centroid	AES Decrypt, Map 1/8
	8 Dimension, 48 Centroid	AES Decrypt, Map 1/12
DNA sequencing	16 Block size, 6 Sub	Fully Configured
	16 Block size, 8 Sub	Fully Configured
	16 Block size, 10 Sub	Fully Configured
	32 Block size, 6 Sub	Fully Configured
	32 Block size, 8 Sub	Fully Configured
	32 Block size, 10 Sub	Fully Configured
	48 Block size, 6 Sub	Fully Configured
	48 Block size, 8 Sub	Fully Configured
	48 Block size, 10 Sub	Fully Configured