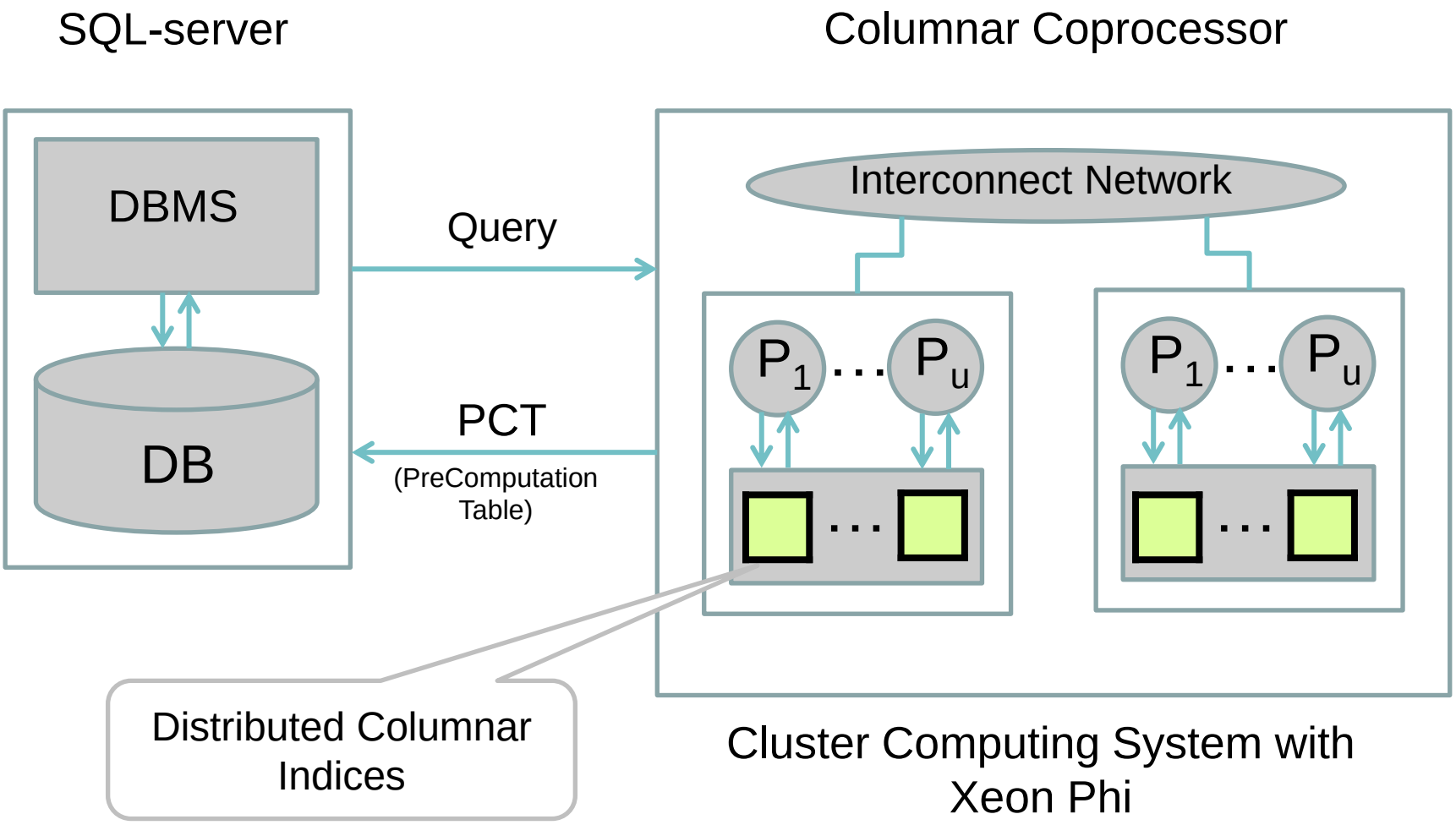
Using Intel Xeon Phi for natural join execution on in-memory compressed data

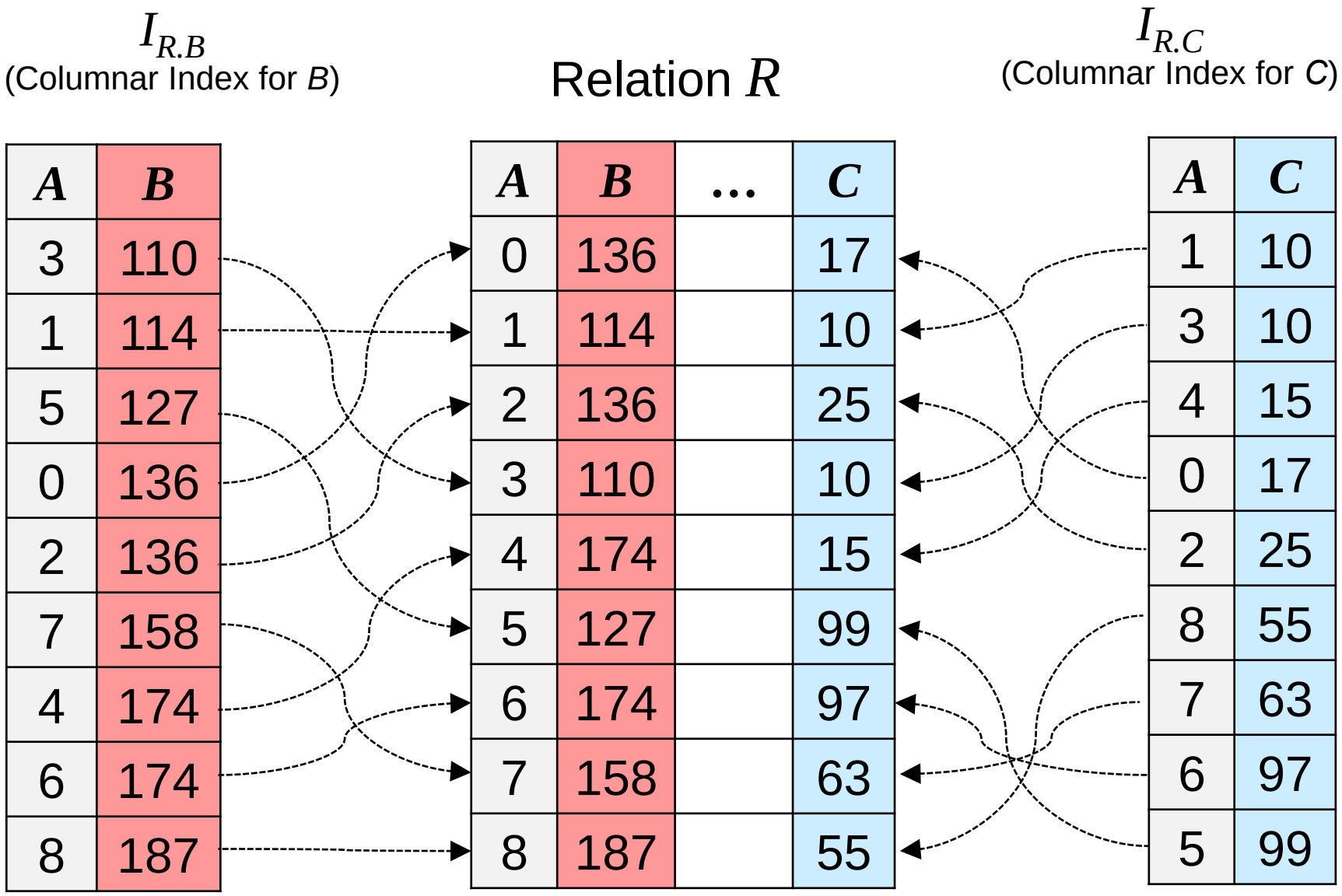
Leonid Sokolinsky, Elena Ivanova

South Ural State University

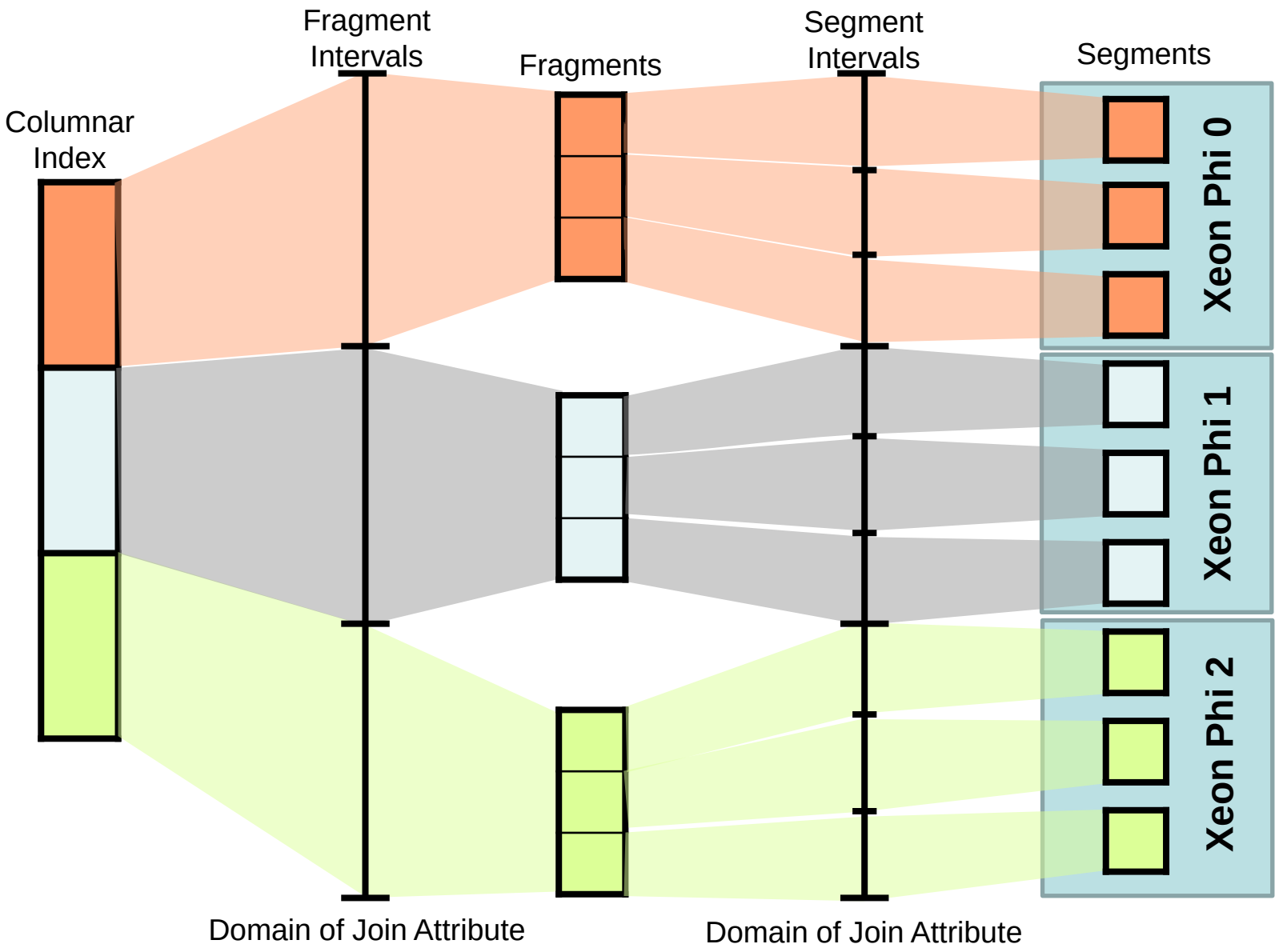
The Idea


SQL-server
Columnar Coprocessor
DBMS
DB
Query
PCT
(PreComputation Table)
Interconnect Network
P1 ... Pu
P1 ... Pu
Distributed Columnar Indices
Cluster Computing System with Xeon Phi

Columnar Index



Data Distribution



Natural Join $R \underset{R.B=S.B}{!} S$

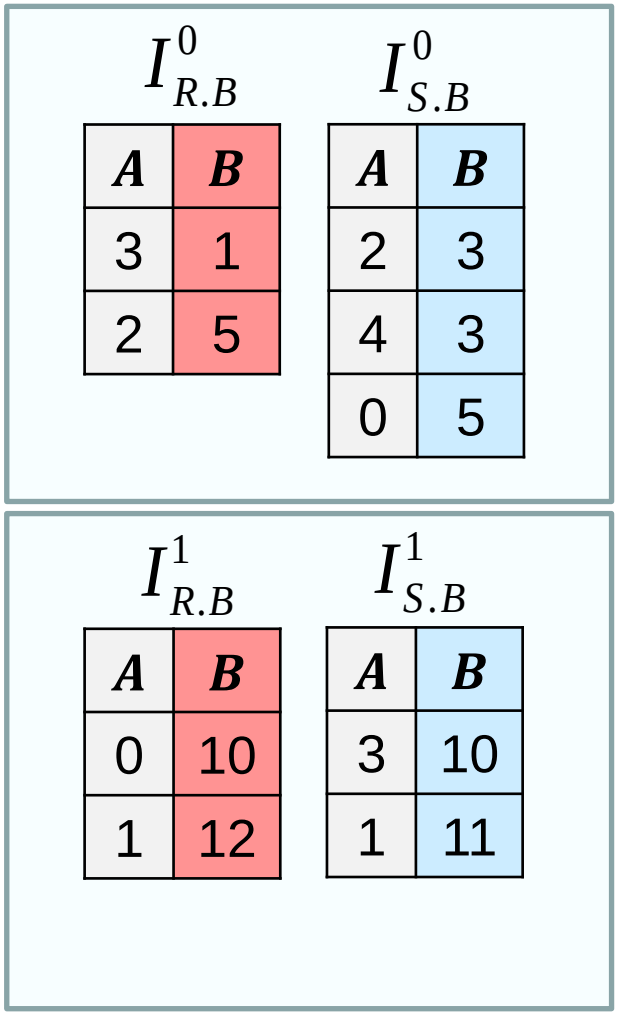
R

A	B	C
0	10	Ni
1	12	Au
2	5	Pb
3	1	Ag

S

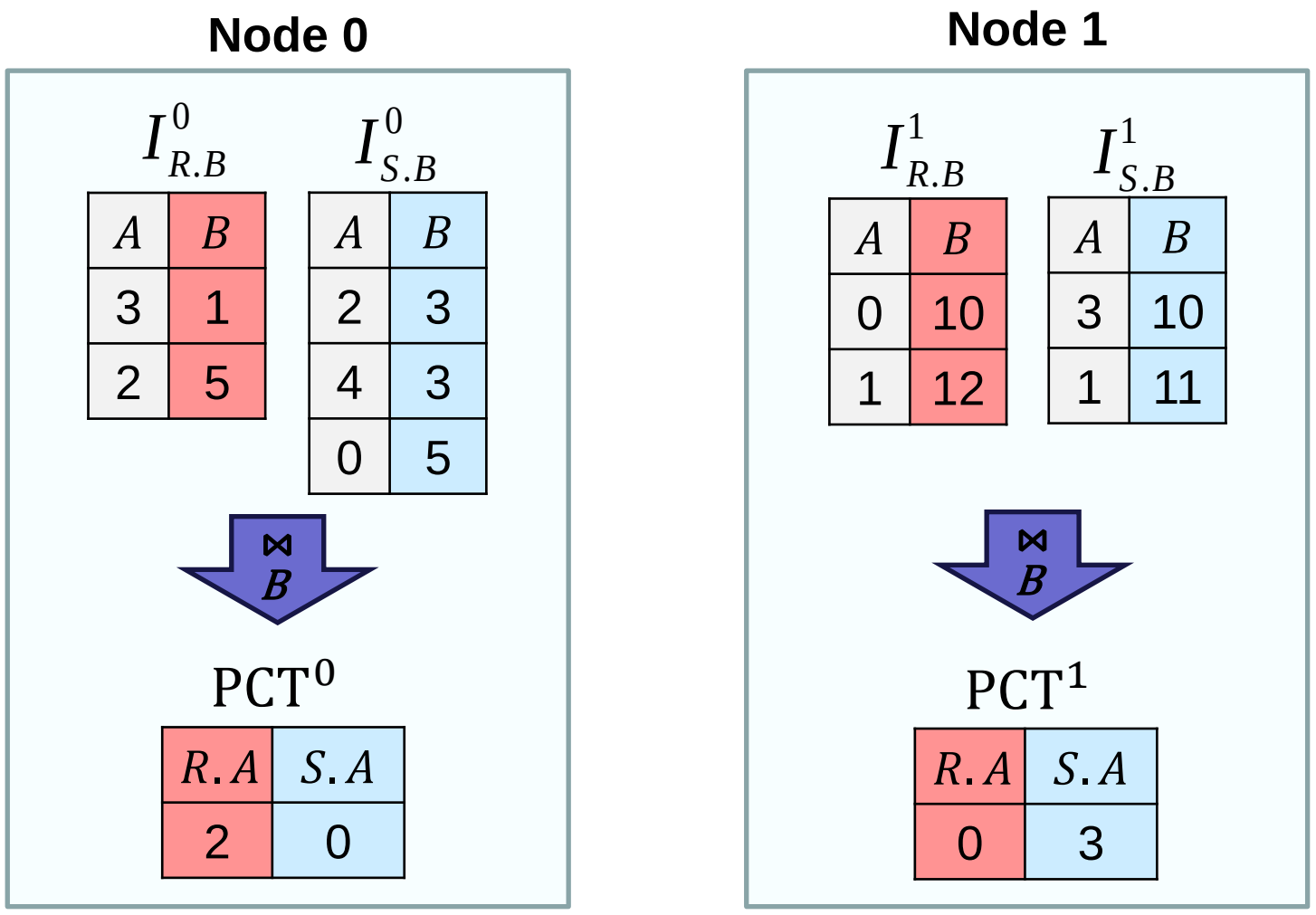
A	B	D
0	5	Pb
1	11	Pb
2	3	Ni
3	10	Fr
4	3	Ag

Data Base

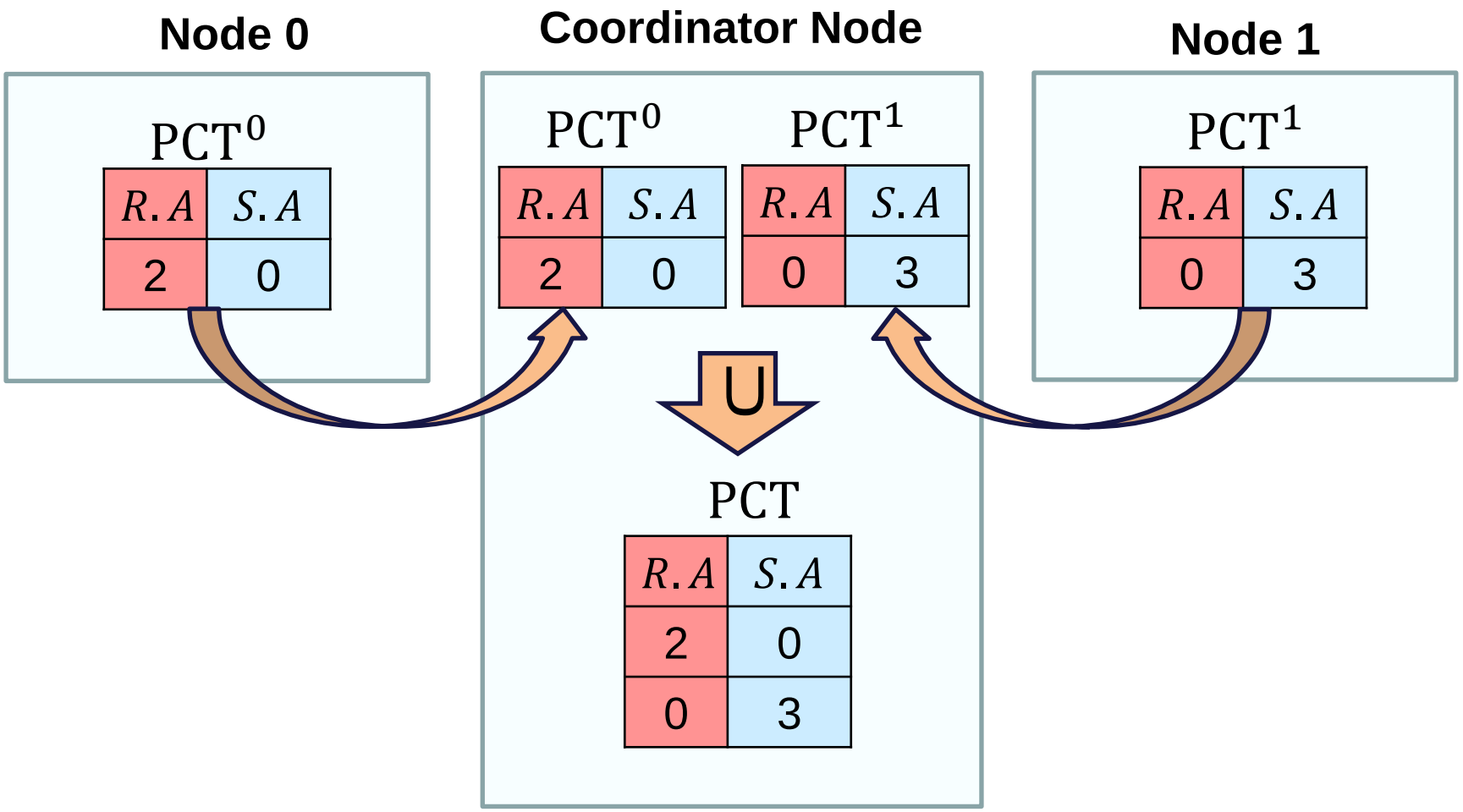


Distributed Columnar Indices

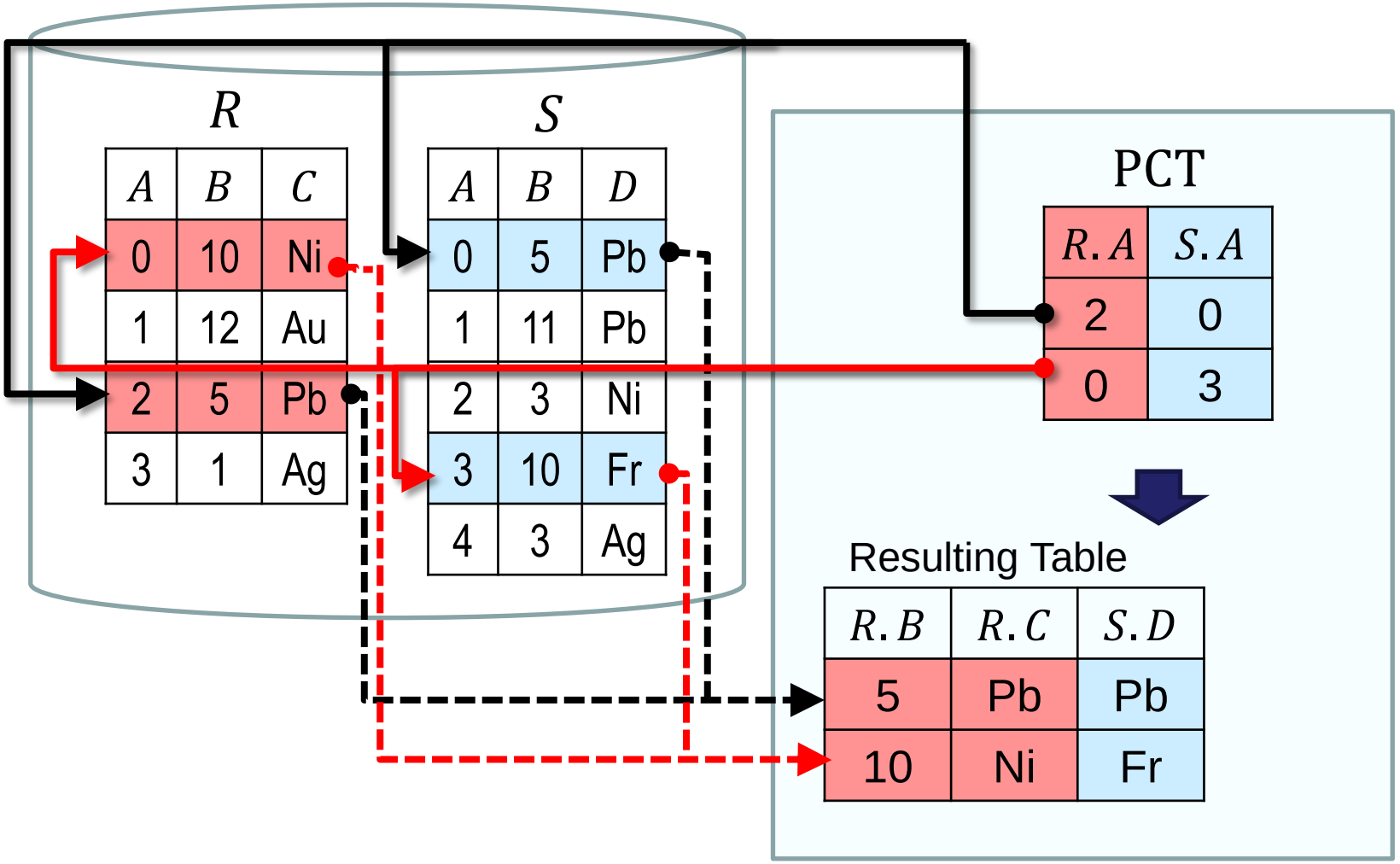
PCT calculation without data exchange



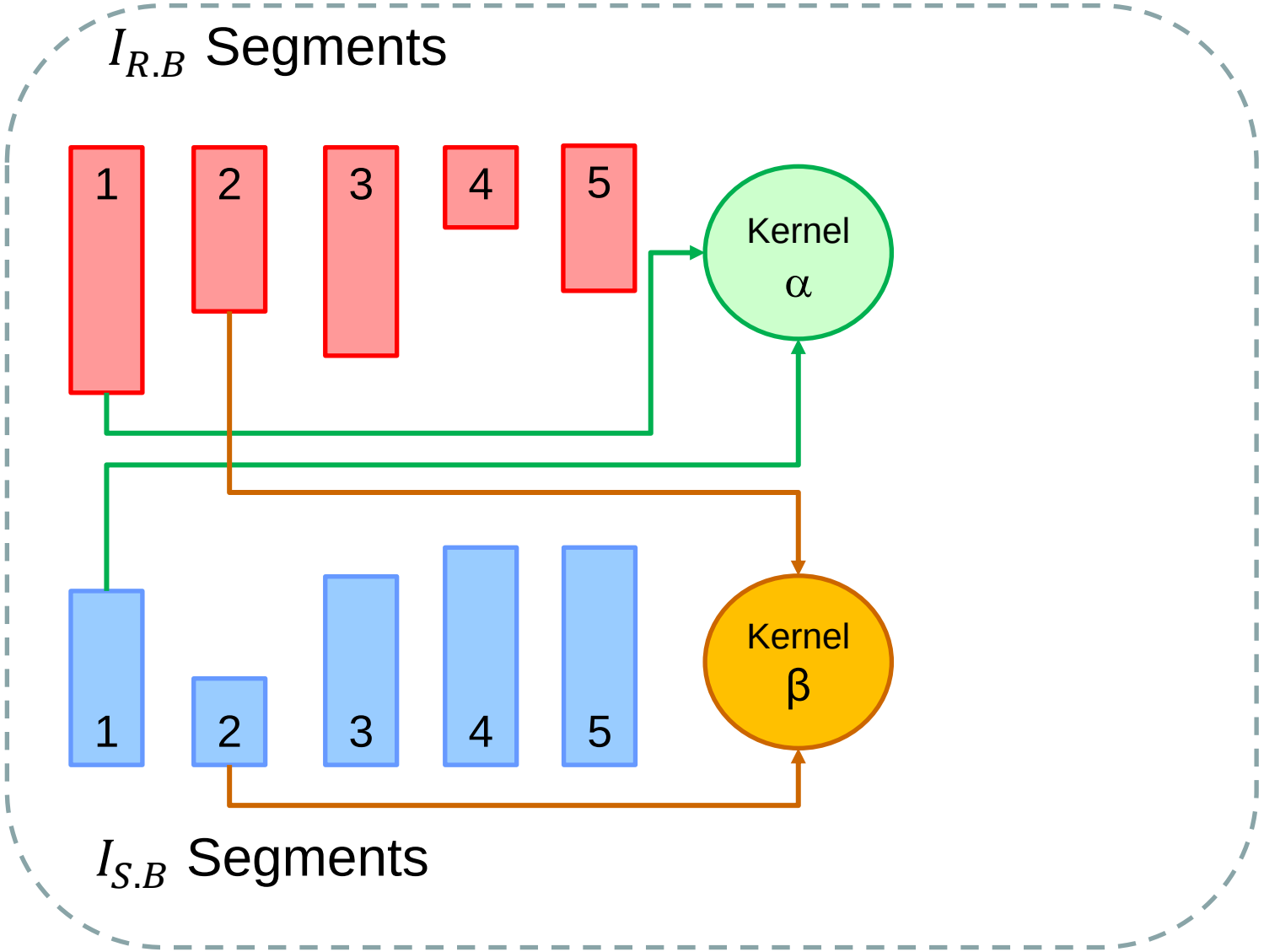
Sending the Fragments of PCT to Coordinator Node



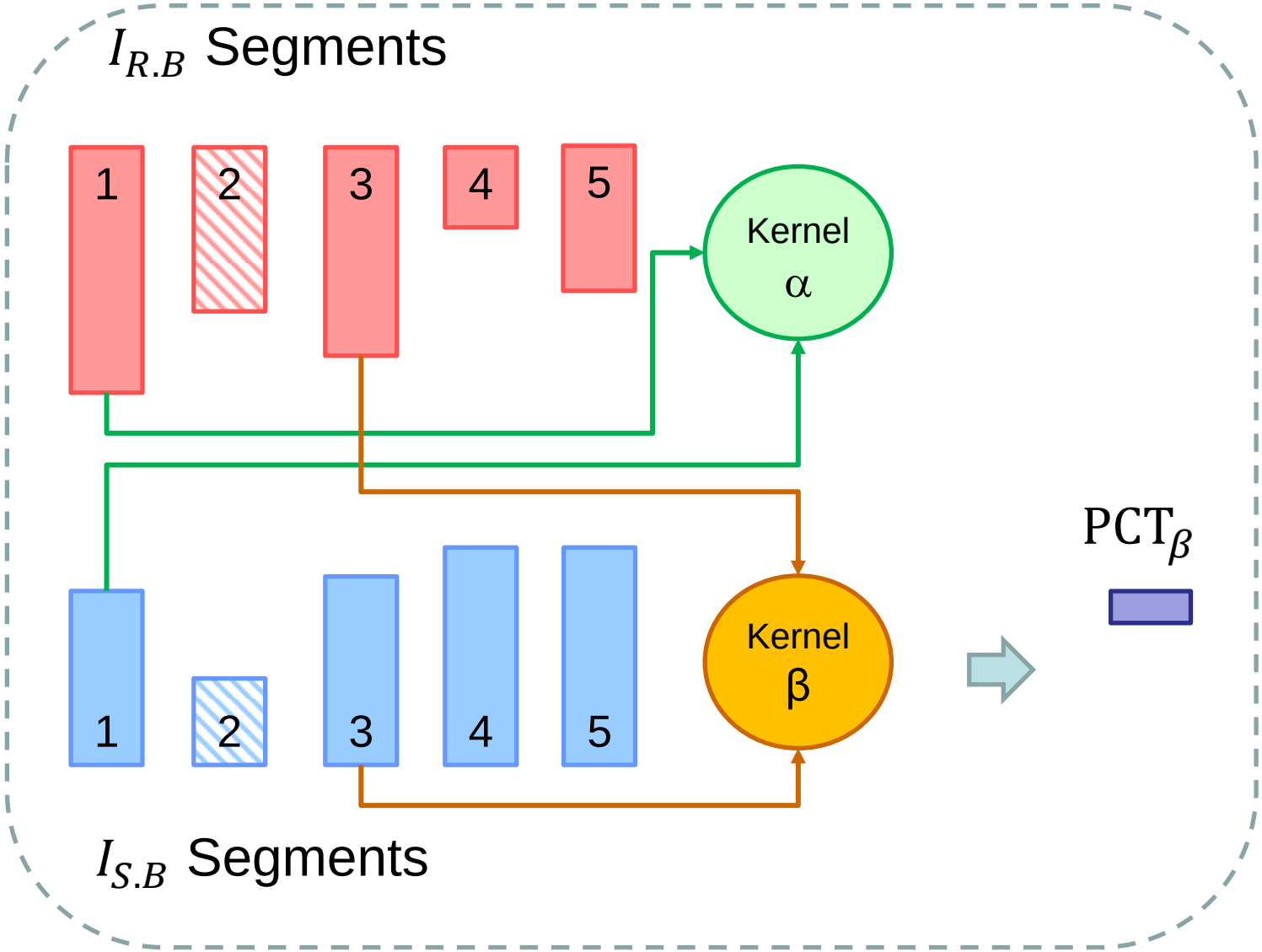
Reconstruction of the Natural Join Resulting Table using PCT



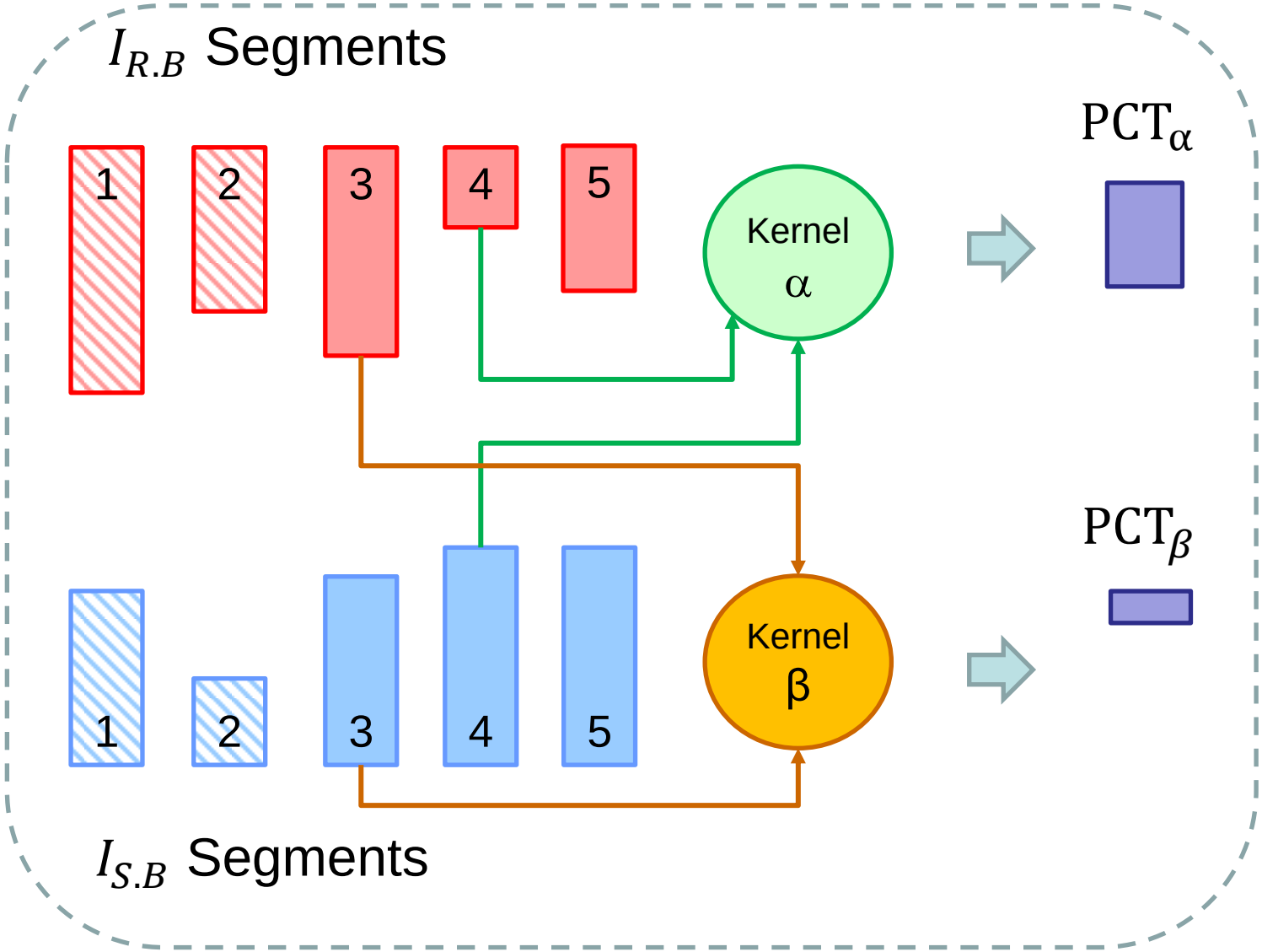
Load Balancing



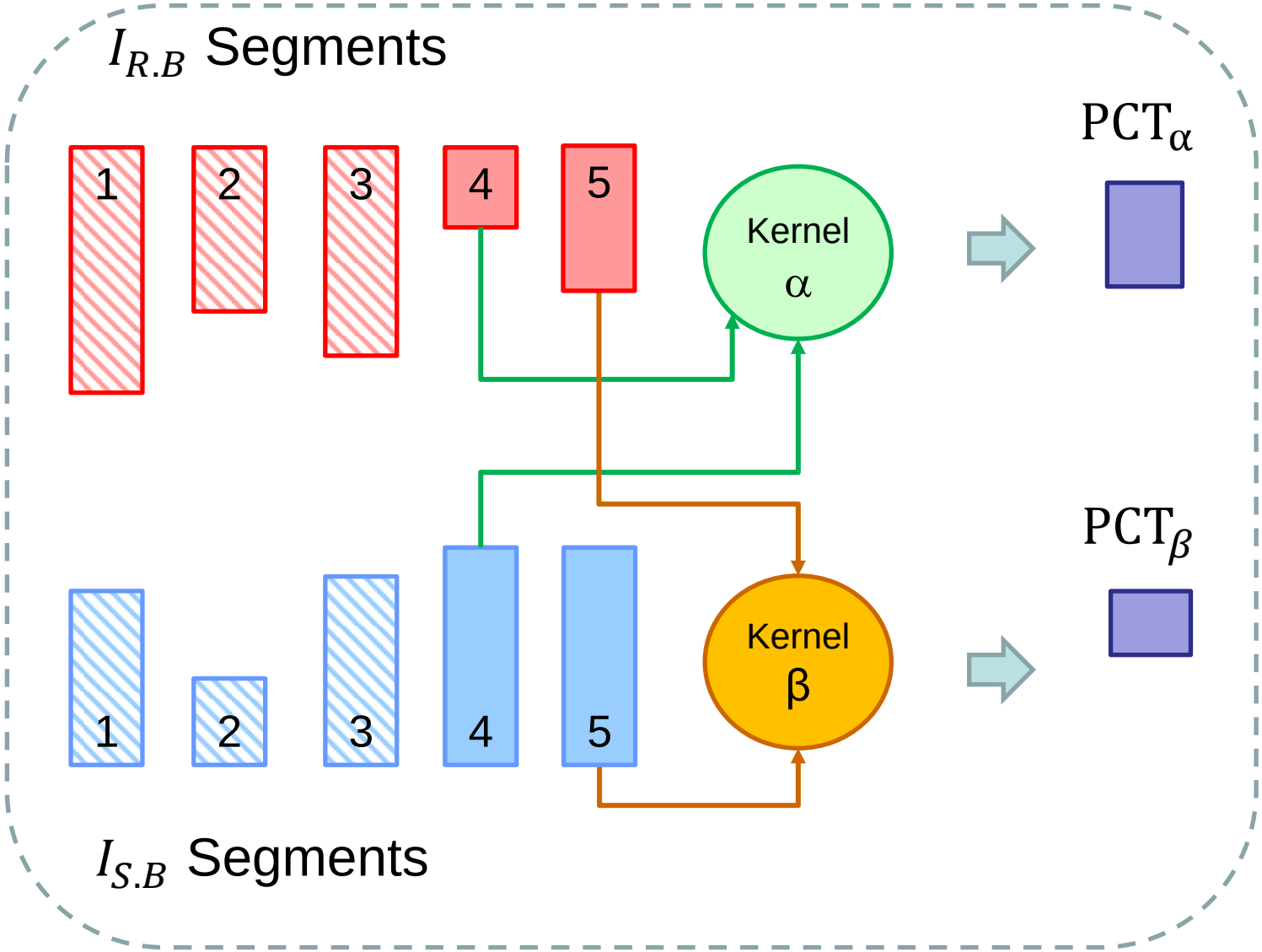
Load Balancing



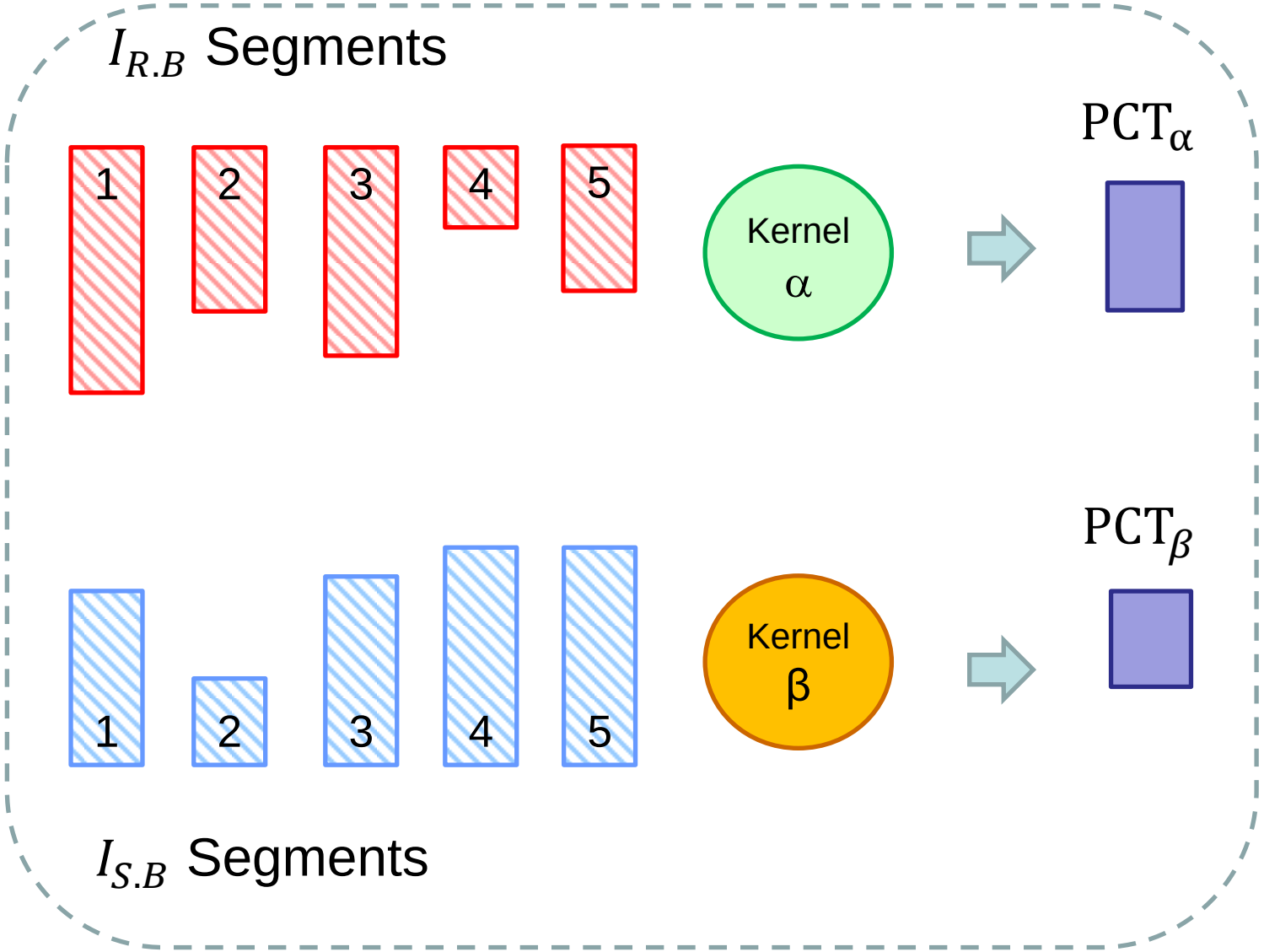
Load Balancing



Load Balancing



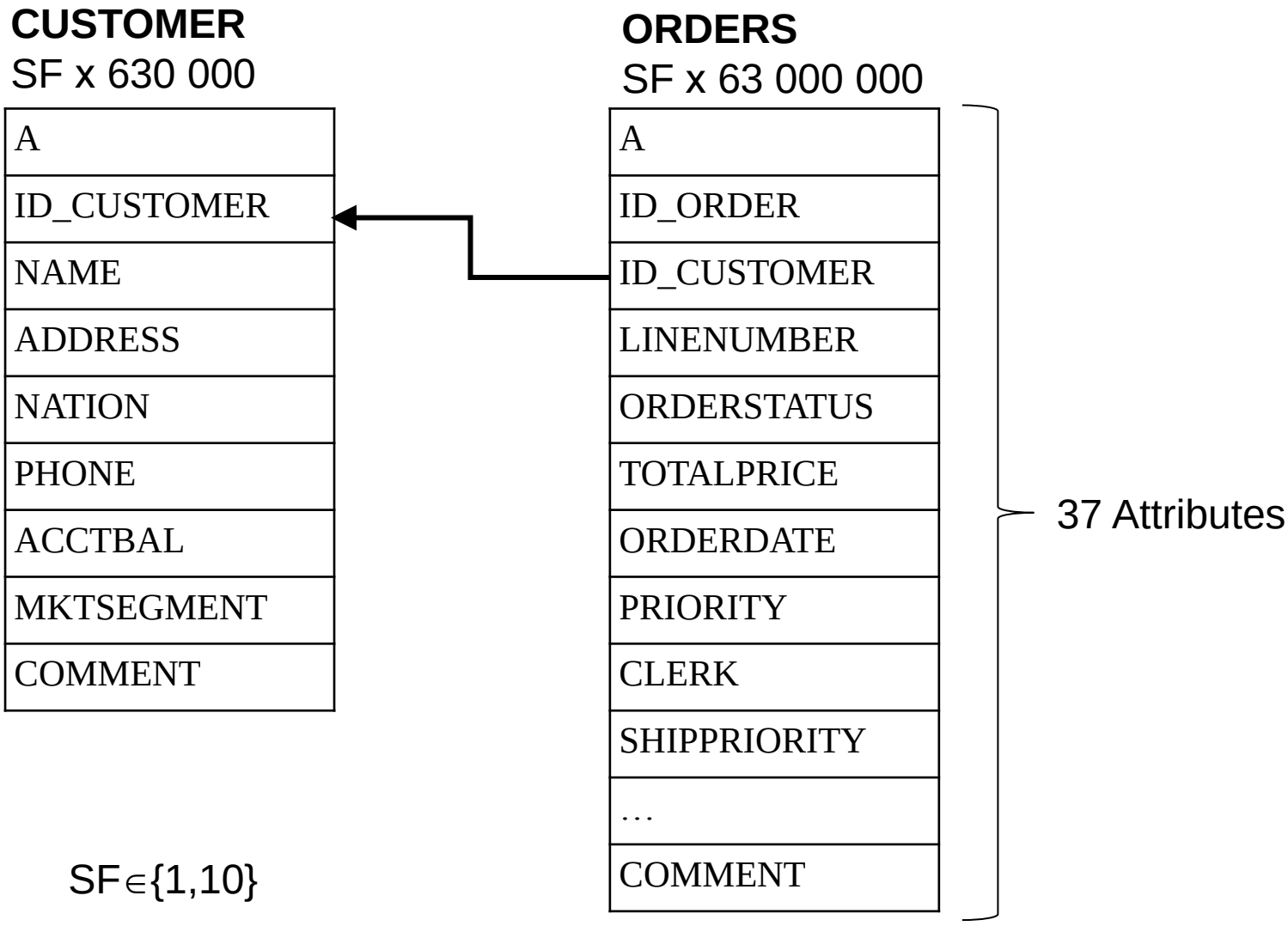
Load Balancing



RSC PetaStream

Nodes:	64
Processors:	Intel Xeon Phi 7120D (61 Kernel по 1.24 ГГц)
RAM per Node:	16 Гб
Interconnect:	InfiniBand FDR + Gigabit Ethernet
OS:	Linux CentOS 7.0

TPC-H based Test

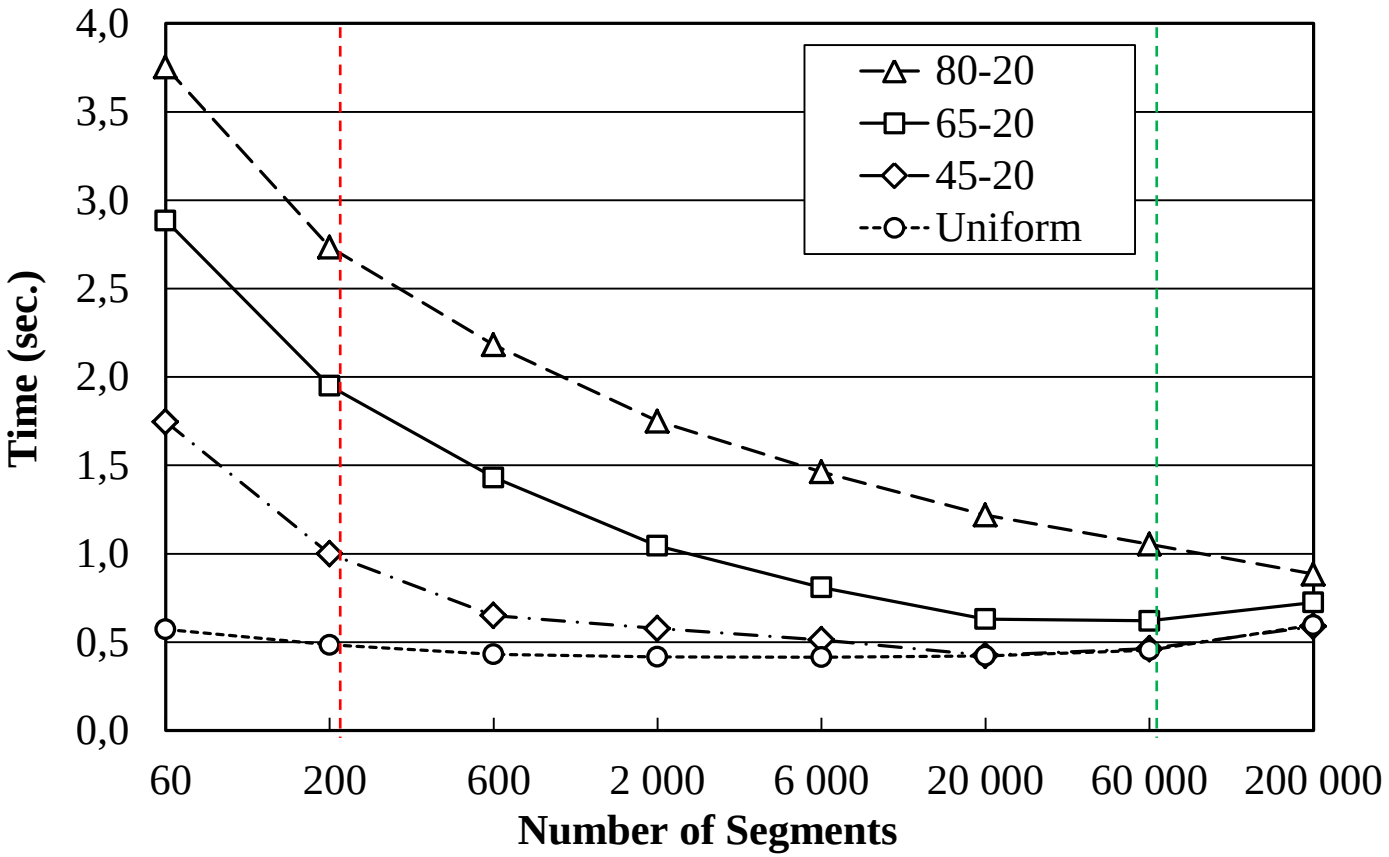


SQL-Query

IXPUG
THE INTEL® XEON PHI™ USERS GROUP

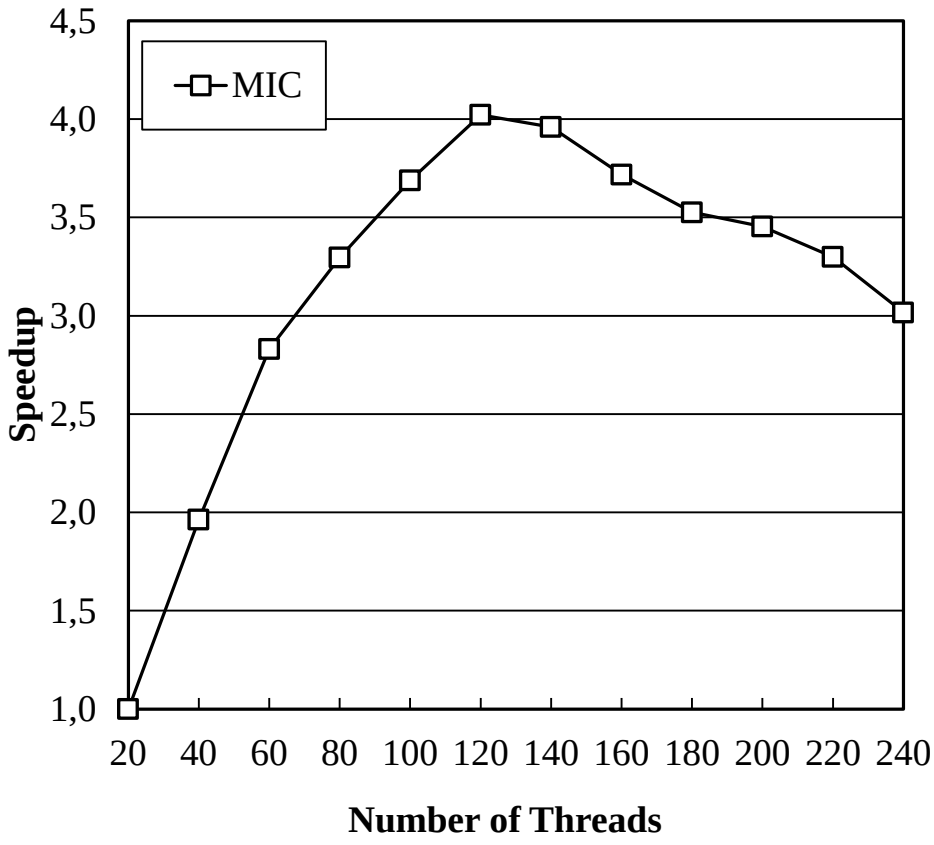
SELECT *
FROM CUSTOMER, ORDERS
WHERE (CUSTOMER.ID_CUSTOMER=ORDERS.ID_CUSTOMER)
AND (ORDERS.TOTALPRICE <= Sel*100 000).

Load Balancing on Xeon Phi



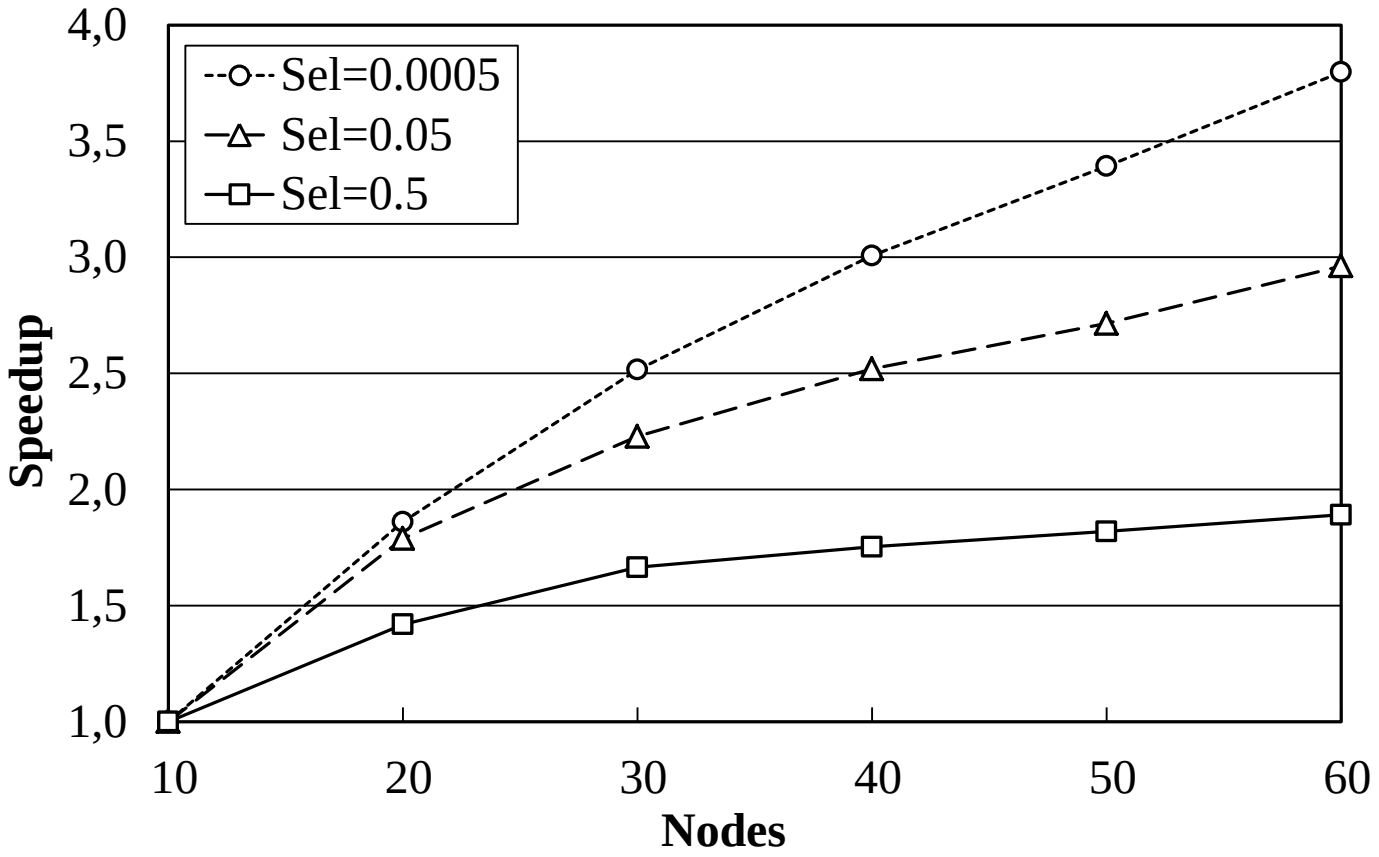
Size of Warehouse: 63 000 000

Hyper-threading



Size of Warehouse: 63 000 000

Scalability



Size of Warehouse: 63 000 000

Combining PostgreSQL and Column COProcessor (CCOP)

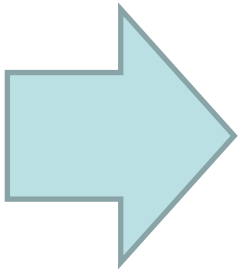
DBMS	Time (minutes)					
	Sel=0.0005		Sel=0.005		Sel=0.05	
	1-st Run	2-nd Run	1-st Run	2-nd Run	1-st Run	2-nd Run
PostgreSQL	7.3	1.21	7.6	1.29	7.6	1.57
PostgreSQL & B-Trees	2.62	2.34	2.83	2.51	2.83	2.63
PostgreSQL & CCOP	0.073	0.008	0.65	0.05	2.03	1.72
Speedup						
$\frac{t_{PostgreSQL\&B-Trees}}{t_{PostgreSQL\ \&\ CCOP}}$	36	**293**	4	50	1.4	1.53

Xeon Phi Utilization

In-memory Data

No Data Transfer between nodes

Heavy-Weight Compression



Xeon Phi Utilization 98%

Direction of Future Research

- Vectorization
- Experiments with Intel Integrated Performance Primitives from Math Kernel Library
  - Entropy-coding compression: Huffman, VLC
  - Dictionary-based compression: LZSS, LZ77
  - Burrows-Wheeler Transform (BWT)
  - MoveToFront (MTF)
  - Run-Length Encoding (RLE)
  - Generalized Interval Transformation (GIT)

Questions?