Exploiting 3D Memory for Accelerated In-Network Processing of Hash Joins in Distributed Databases

Johannes Wirth[1], Jaco A. Hofmann[1], Lasse Thostrup[2], Andreas Koch[1], Carsten Binnig[2]

1) Embedded Systems and Applications Group, TU Darmstadt

2) Data Management Lab, TU Darmstadt

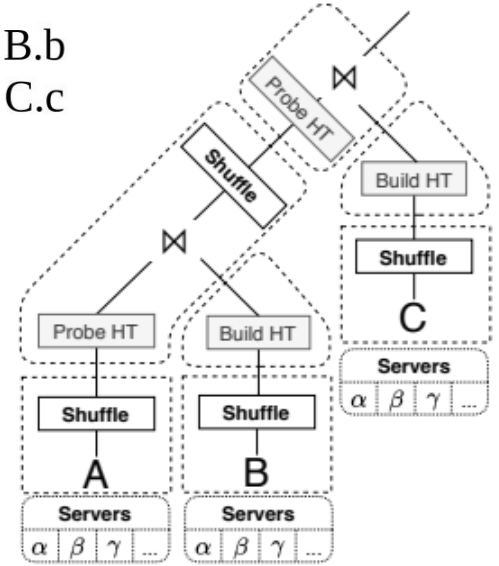
BACKGROUND

SQL Join Operation

- Common database operation
- Merges two tables/relations
  - Combine entries with matching values
- Implementation: hash join algorithm
- Problematic in distributed databases

  → Shuffling

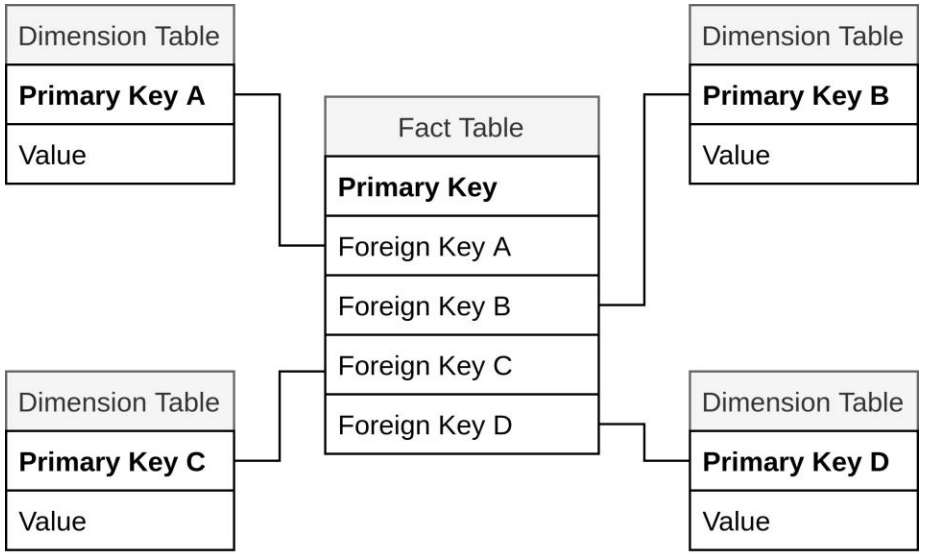
```
SELECT A.a,B.d,C.e
FROM A,B,C
WHERE A.b = B.b
    AND A.c = C.c
```



Star Schema

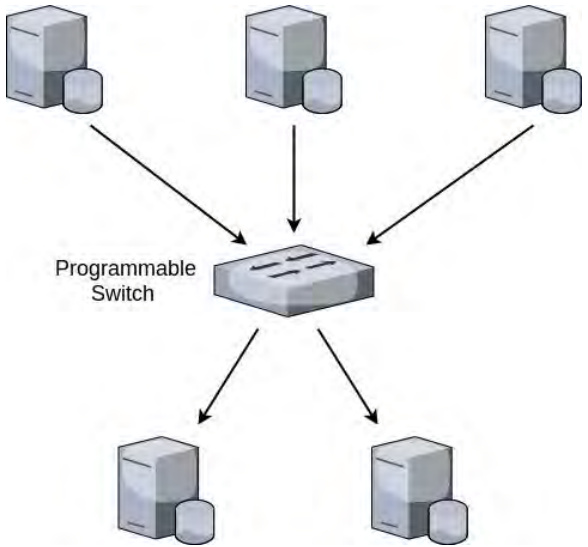
- Large fact table
- Smaller dimension tables

- Foreign key in fact table points to entry in corrresponding dimension table
- SQL Join to combine them



In-Network Processing

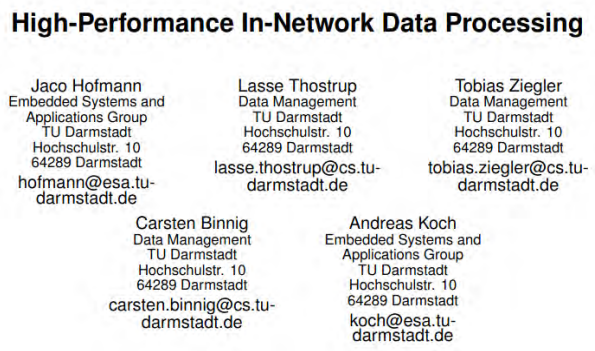
- Process in-flight data
  → Reduce network traffic
- Programmable switches (P4, Tofino, …)
  - Limited flexibility
  - Only small memory
  - Mostly stateless operations



OUR CONTRIBUTION

NetJoin-HBM

- Network accelerator for SQL Join operation
- Multiple joins at once
- Tailored to Star Schema



High-Performance In-Network Data Processing

Jaco Hofmann
Embedded Systems and Applications Group
TU Darmstadt
Hochschulstr. 10
64289 Darmstadt
hofmann@esa.tu-darmstadt.de

Lasse Thostrup
Data Management
TU Darmstadt
Hochschulstr. 10
64289 Darmstadt
lasse.thostrup@cs.tu-darmstadt.de

Tobias Ziegler
Data Management
TU Darmstadt
Hochschulstr. 10
64289 Darmstadt
tobias.ziegler@cs.tu-darmstadt.de

Carsten Binnig
Data Management
TU Darmstadt
Hochschulstr. 10
64289 Darmstadt
carsten.binnig@cs.tu-darmstadt.de

Andreas Koch
Embedded Systems and Applications Group
TU Darmstadt
Hochschulstr. 10
64289 Darmstadt
koch@esa.tu-darmstadt.de

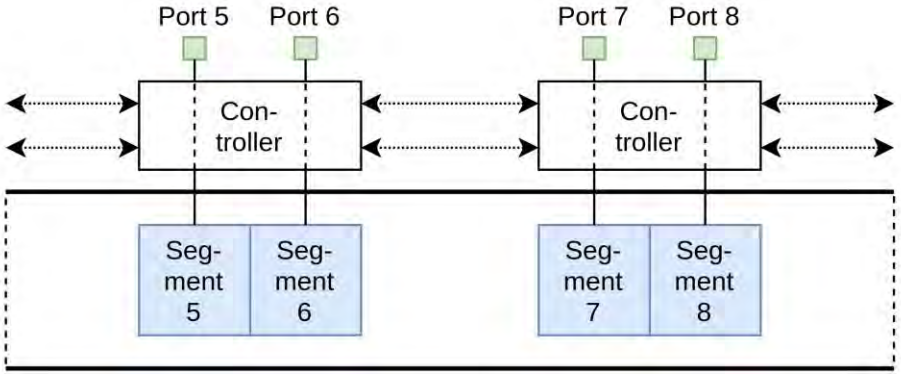
- Builds on prior work [9]

  → Make use of HBM to improve performance

High-Bandwidth Memory

TECHNISCHE UNIVERSITÄT DARMSTADT

- Fast on-chip memory
- Xilinx FPGAs:
  - 4 – 16 GB
  - 460 GB/s
- Highly parallel (32 channels)
- Optional switch: not used here



Operation

TECHNISCHE UNIVERSITÄT DARMSTADT

Operation

```
SELECT A.a,B.e,C.f,D.g
FROM A,B,C,D
WHERE A.b = B.b
    AND A.c = C.c
    AND A.d = D.d
```

Table A

a	b	c	d	*Server*
a1	1	1	1	α
a2	2	1	3	β
a3	3	2	2	γ
a4	1	2	1	α
a5	2	3	3	β
a6	3	3	2	γ

Table B

b	e	*Server*
1	e1	α
2	e2	β
3	e3	γ

Table C

c	f	*Server*
1	f1	α
2	f2	β
3	f3	γ

Table D

d	g	*Server*
1	g1	α
2	g2	β
3	g3	γ

Hash Table: 2 Buckets, 2 Slots per Bucket
Collision Handling: Use next free Slot in Bucket
Hashing: bucket(x) = x % 2

Hashing 1

Hashing: bucket(x) = x % 2
Collision Handling: Use next free Slot in Bucket

Table B

b	e	*Server*
1	e1	α
2	e2	β
3	e3	γ

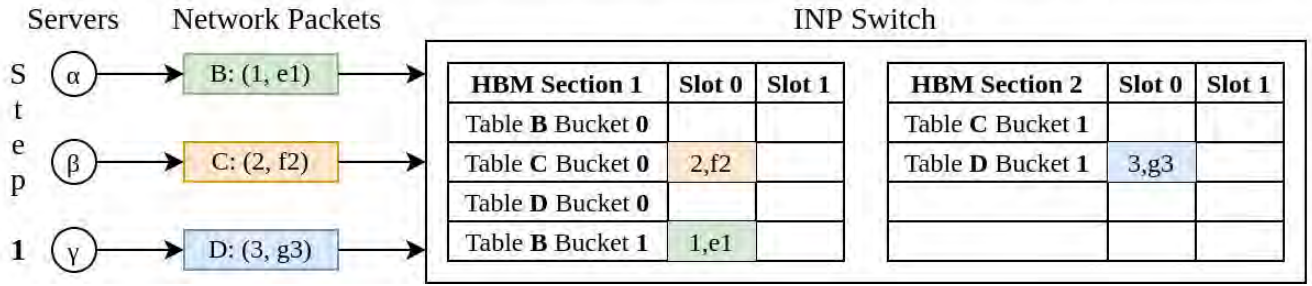
Table C

c	f	*Server*
1	f1	α
2	f2	β
3	f3	γ

Table D

d	g	*Server*
1	g1	α
2	g2	β
3	g3	γ

Hashing Phase: Transfer dimension tables B,C,D to INP switch and store in Hash Tables



Hashing 2

Hashing: bucket(x) = x % 2
Collision Handling: Use next free Slot in Bucket

Table B

b	e	*Server*
1	e1	α
2	e2	β
3	e3	γ

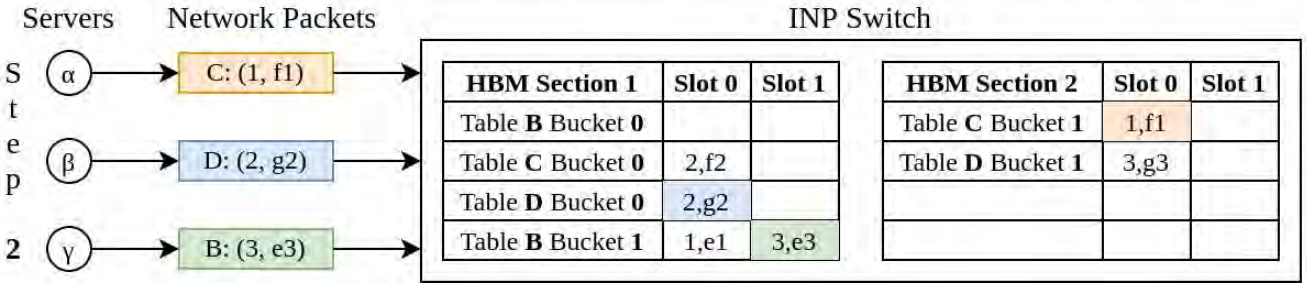
Table C

c	f	*Server*
1	f1	α
2	f2	β
3	f3	γ

Table D

d	g	*Server*
1	g1	α
2	g2	β
3	g3	γ

Hashing Phase: Transfer dimension tables B,C,D to INP switch and store in Hash Tables



Hashing 3

Hashing: bucket(x) = x % 2
Collision Handling: Use next free Slot in Bucket

Table B

b	e	*Server*
1	e1	α
2	e2	β
3	e3	γ

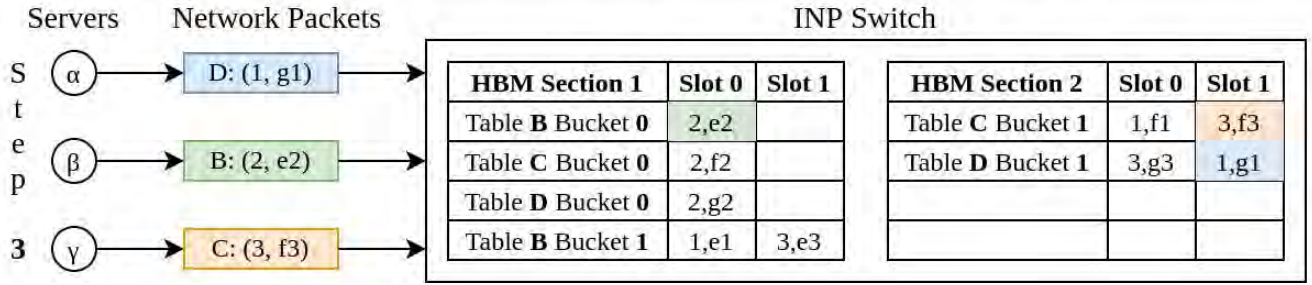
Table C

c	f	*Server*
1	f1	α
2	f2	β
3	f3	γ

Table D

d	g	*Server*
1	g1	α
2	g2	β
3	g3	γ

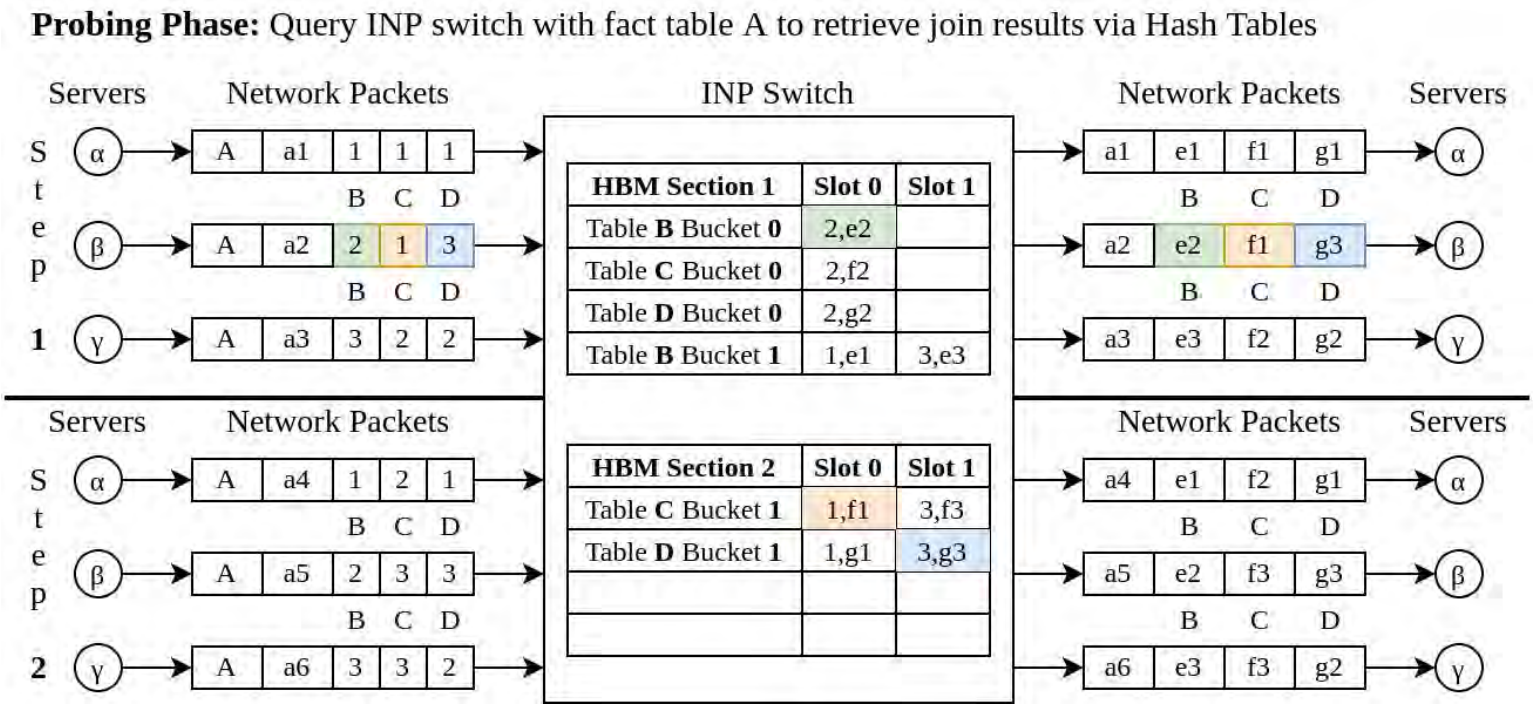
Hashing Phase: Transfer dimension tables B,C,D to INP switch and store in Hash Tables



HBM Section 1	Slot 0	Slot 1
Table **B** Bucket **0**	2,e2	
Table **C** Bucket **0**	2,f2	
Table **D** Bucket **0**	2,g2	
Table **B** Bucket **1**	1,e1	3,e3

HBM Section 2	Slot 0	Slot 1
Table **C** Bucket **1**	1,f1	3,f3
Table **D** Bucket **1**	3,g3	1,g1

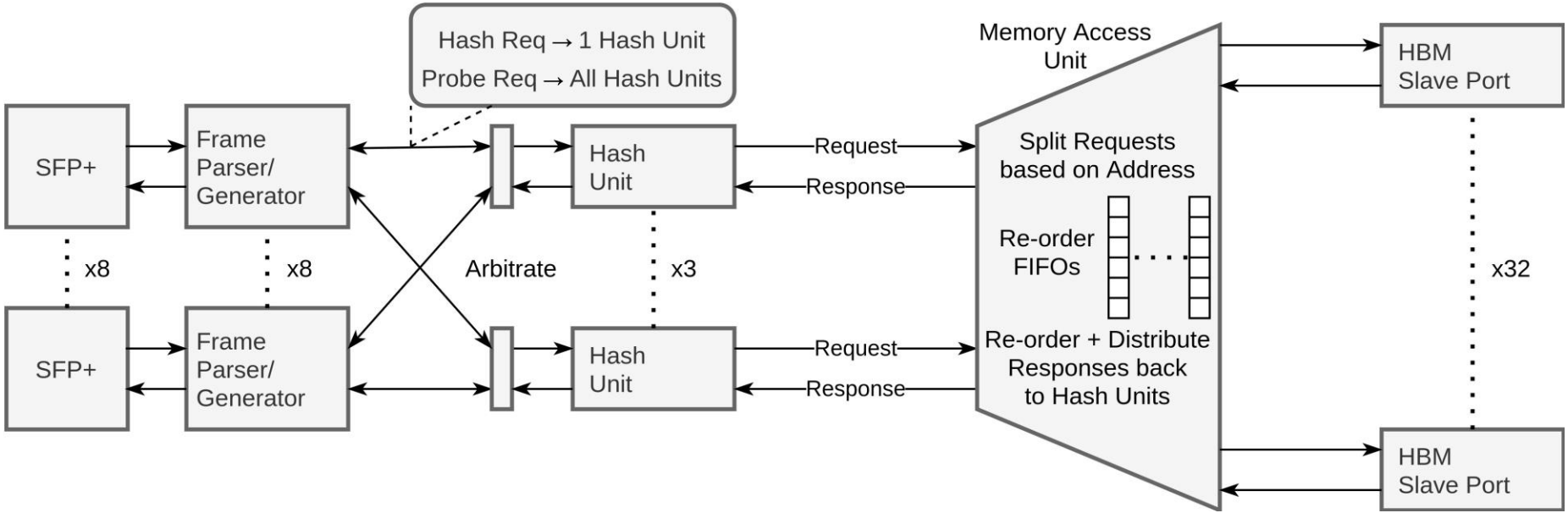
Probing


Probing Phase: Query INP switch with fact table A to retrieve join results via Hash Tables
Servers
Network Packets
INP Switch
Network Packets
Servers
Step 1
α
β
γ
A a1 1 1 1
B C D
A a2 2 1 3
B C D
A a3 3 2 2
HBM Section 1 | Slot 0 | Slot 1
Table B Bucket 0 | 2,e2 |
Table C Bucket 0 | 2,f2 |
Table D Bucket 0 | 2,g2 |
Table B Bucket 1 | 1,e1 | 3,e3
a1 e1 f1 g1
B C D
a2 e2 f1 g3
B C D
a3 e3 f2 g2
Servers
Network Packets
Network Packets
Servers
Step 2
α
β
γ
A a4 1 2 1
B C D
A a5 2 3 3
B C D
A a6 3 3 2
HBM Section 2 | Slot 0 | Slot 1
Table C Bucket 1 | 1,f1 | 3,f3
Table D Bucket 1 | 1,g1 | 3,g3
a4 e1 f2 g1
B C D
a5 e2 f3 g3
B C D
a6 e3 f3 g2

TECHNISCHE UNIVERSITÄT DARMSTADT

ARCHITECTURE

Architecture Overview

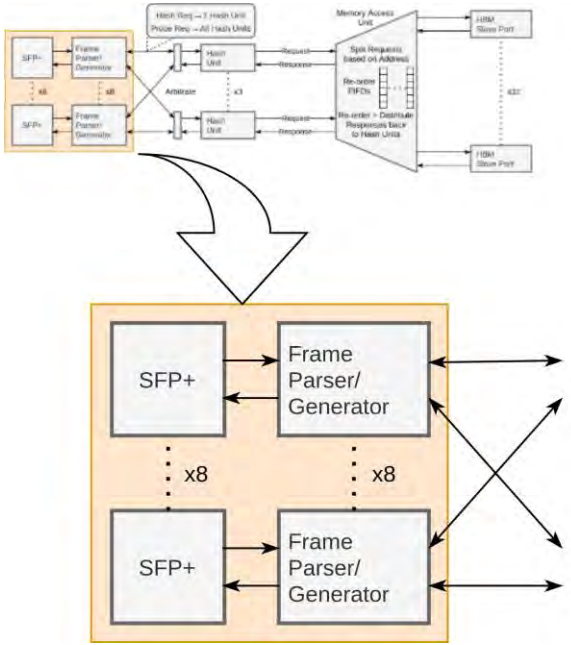


Networking

- Parse/create ethernet frames
- 10G Ethernet
- Separate clock domain (156 MHz)
- Must not block

  → Drop frame if queue full

  → Re-request data from sender

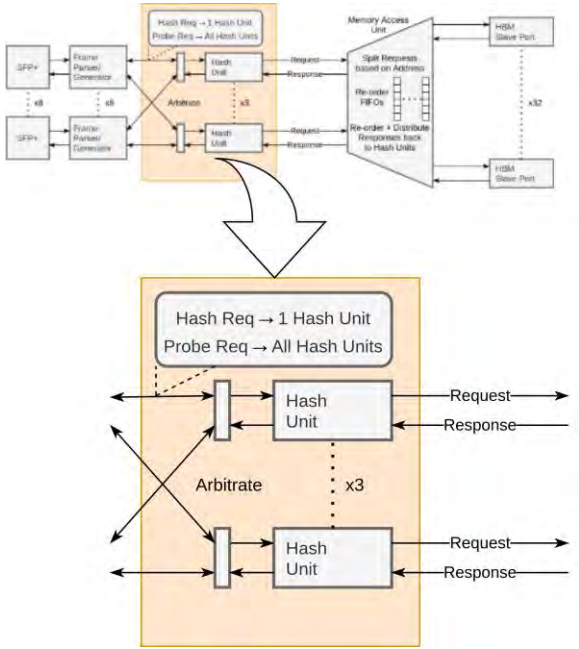


Hash Unit

- One hash unit per join
- Throughput optimized
- Four slots per bucket

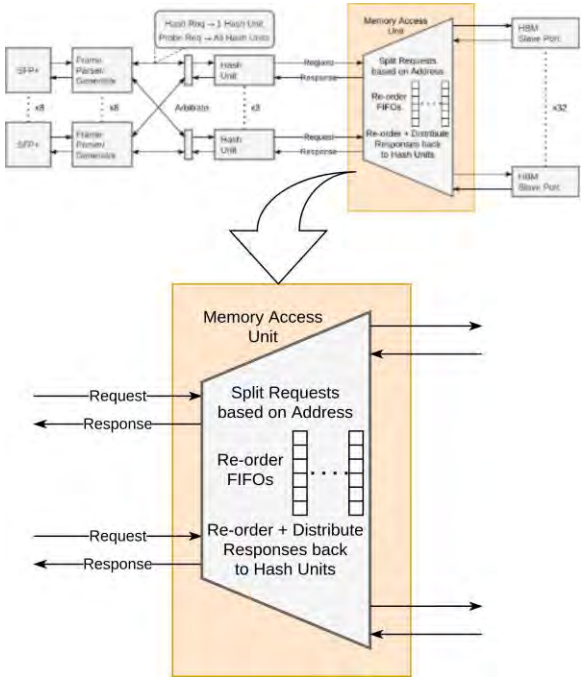
  → Bucket width matches HBM width
- Hashing: apply integer hash function to key

  → Bucket = hash % #buckets



Memory Access

- Connect hash units to all HBM ports
  → Challenge: Routability
  → Standard interconnect not feasible
- Intermediate stage (4 HBMs per group)
- Re-order responses to hash units
  → Keep track of original request order



EVALUATION

Evaluation Setup

TECHNISCHE UNIVERSITÄT DARMSTADT

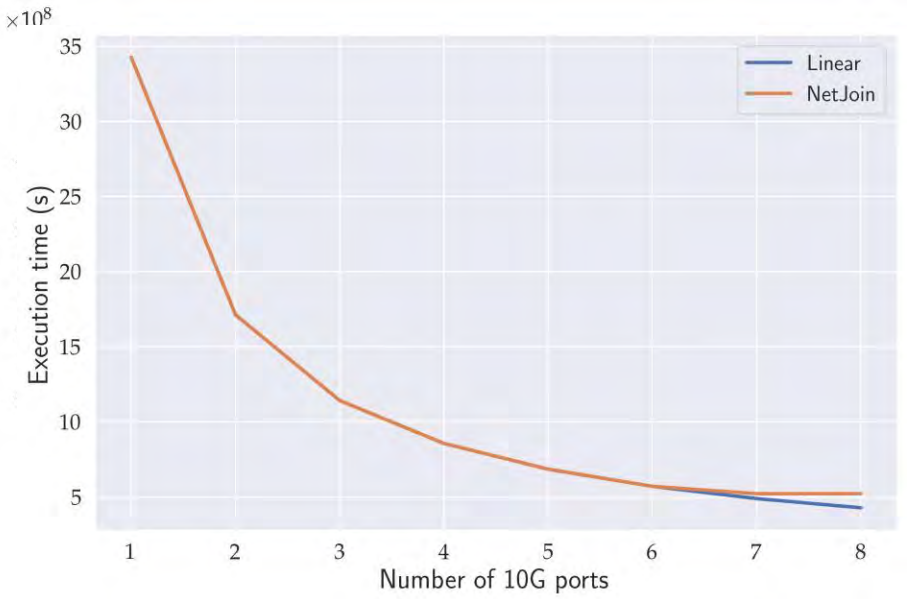
- Scenario: fact table joined with three dimension tables
- Evaluated on Bittware XUP-VVH (8GB HBM; 8 x 10G Ethernet)
- 8 database servers

- Performance scaling
- Comparison to software baseline & prior work

System Performance

- Linear scaling until 6 x 10G
- Similar for both phases
- Probing faster than hashing
- Probing caps at 250 MIOPS

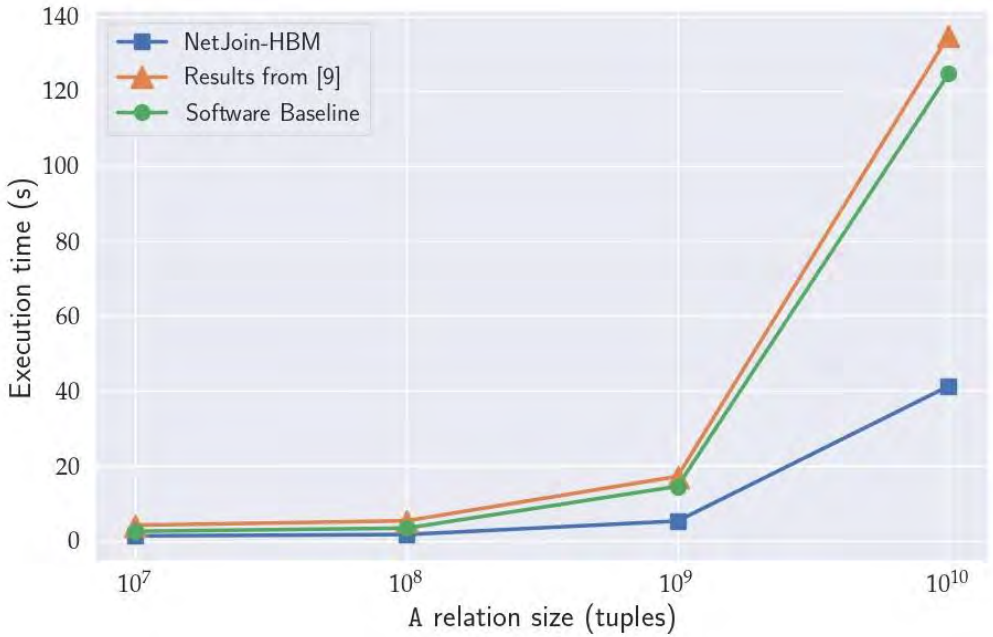
  → bottleneck hash units (250 MHz)



Baseline Comparison

TECHNISCHE UNIVERSITÄT DARMSTADT

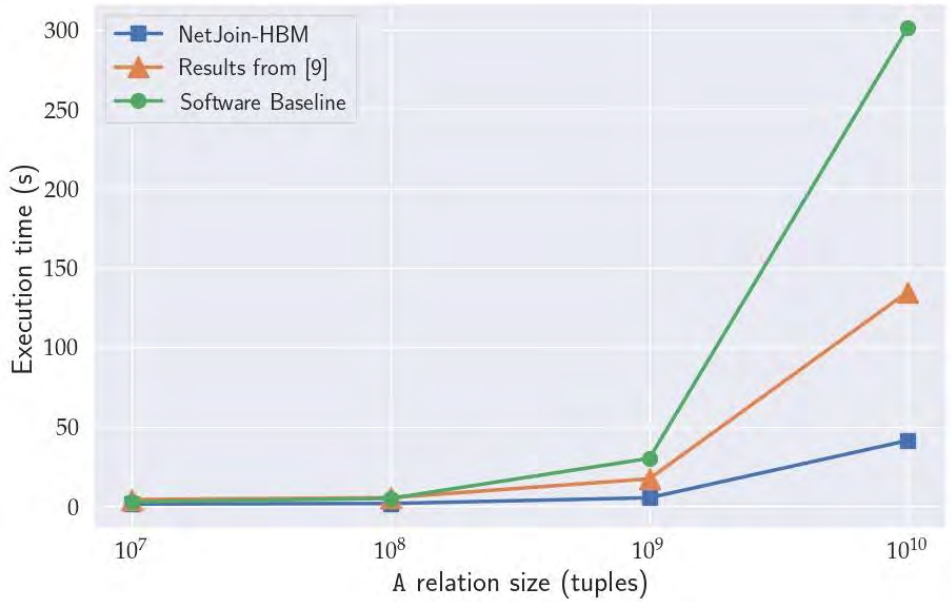
- Dimension table fixed (100 x 10^6)
- Fact table size varied
- Equally distributed
  → Optimal for baseline
- Up to three times faster



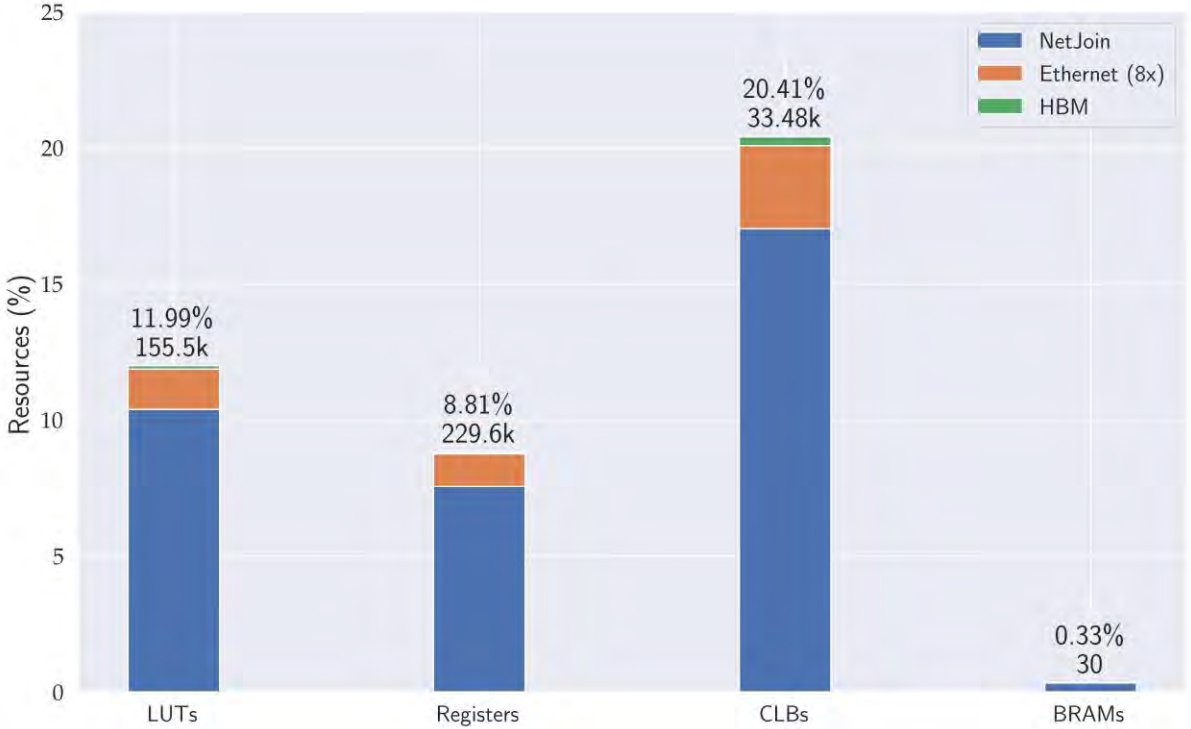
Baseline Comparison (Skewed)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Dimension table fixed (100 x 10^6)
- Fact table size varied
- Skewed distribution (34% - 2%)
- Up to 7 times faster vs baseline
- Up to 3 times faster vs prior work



Resource Usage



Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Handles at least 60 Gbit/s
- Up to 7x faster than software baseline
- Limitations
  - Limited HBM memory
  - Bottleneck: hash units
  - No FPGA-based switch

TECHNISCHE UNIVERSITÄT DARMSTADT

THANKS FOR YOUR ATTENTION!

E-Mail: wirth@esa.tu-darmstadt.de