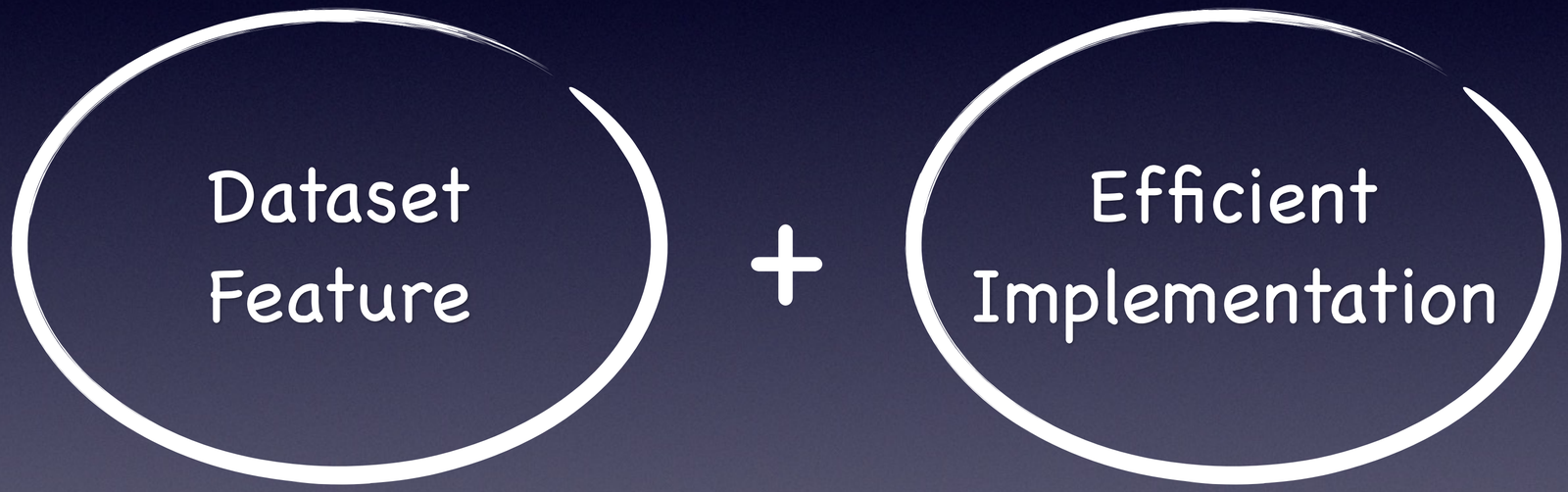
SIGMOD 2015 Programming Contest

Team CUT_HERE_____
KAIST and Seoul National University

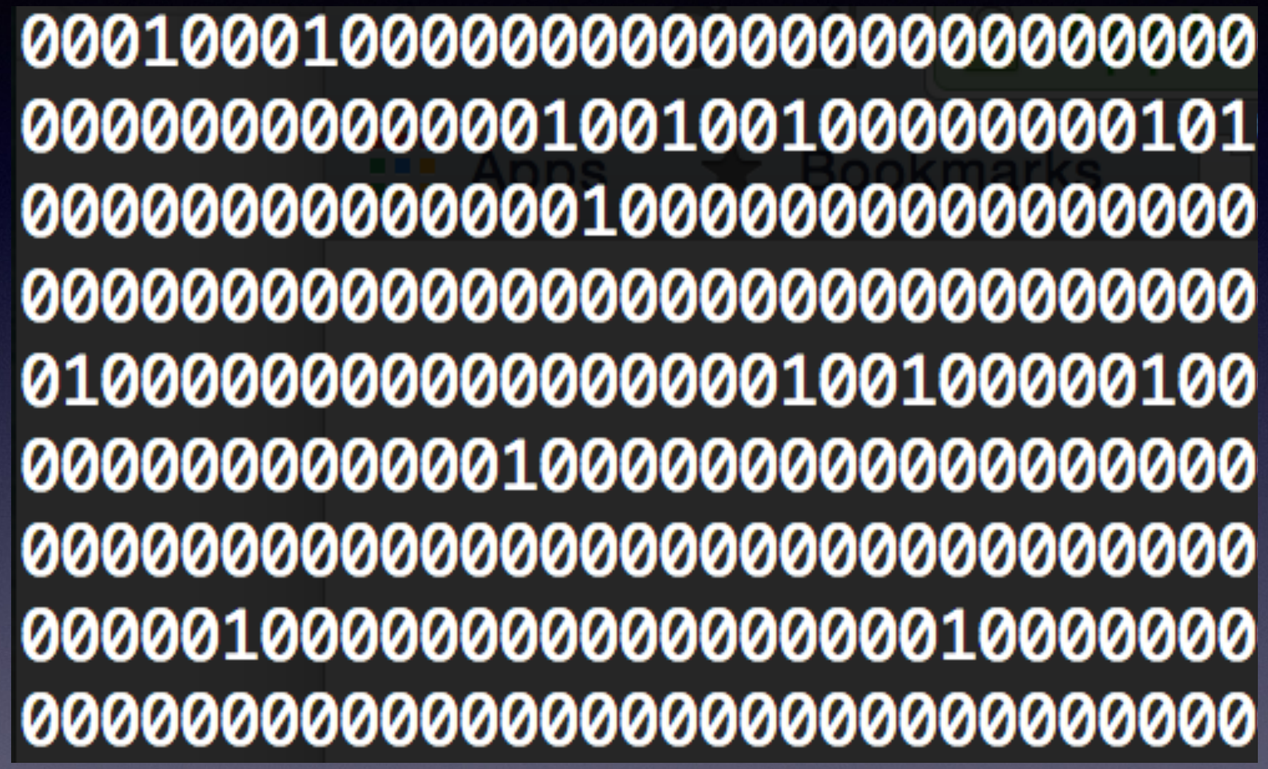
Team Members

- Wonha Ryu: BS student at KAIST
- Seoung Gook Sohn: PhD student at SNU
- Eunjin Song: MS student at SNU

Our Approach

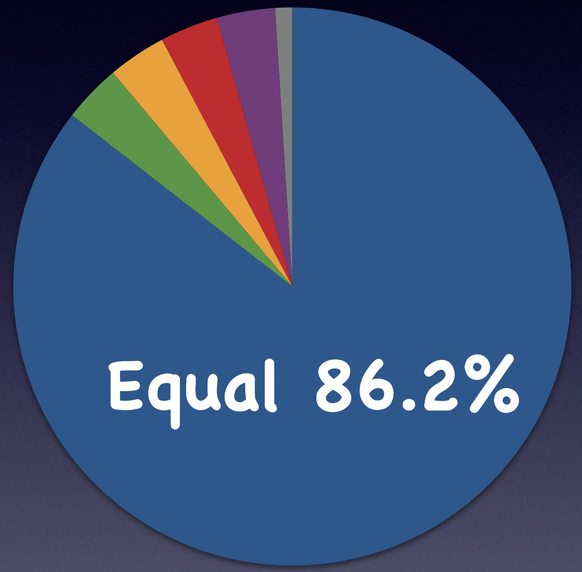


Feature 1: Validation Result



The majority of validation results are zero

Feature 2: Operator Type



Equal operators dominate the types of operators

Feature 3: Column Range

C_1	C_2
3074	8896128387
3077	6811549715
3075	7782881939
3077	5819204839

Query: $C_1 \overset{?}{==} 3076$ $C_2 \overset{?}{==} 5739273029$

Chance of conflict: High Low

Each column has a fixed-range of values

Feature 4: Transaction Size

…

T_n Insert rows: 4
Delete rows: 4

T_{n+1} Insert rows: 1
Delete rows: 2

T_{n+2} Insert rows: 1
Delete rows: 1

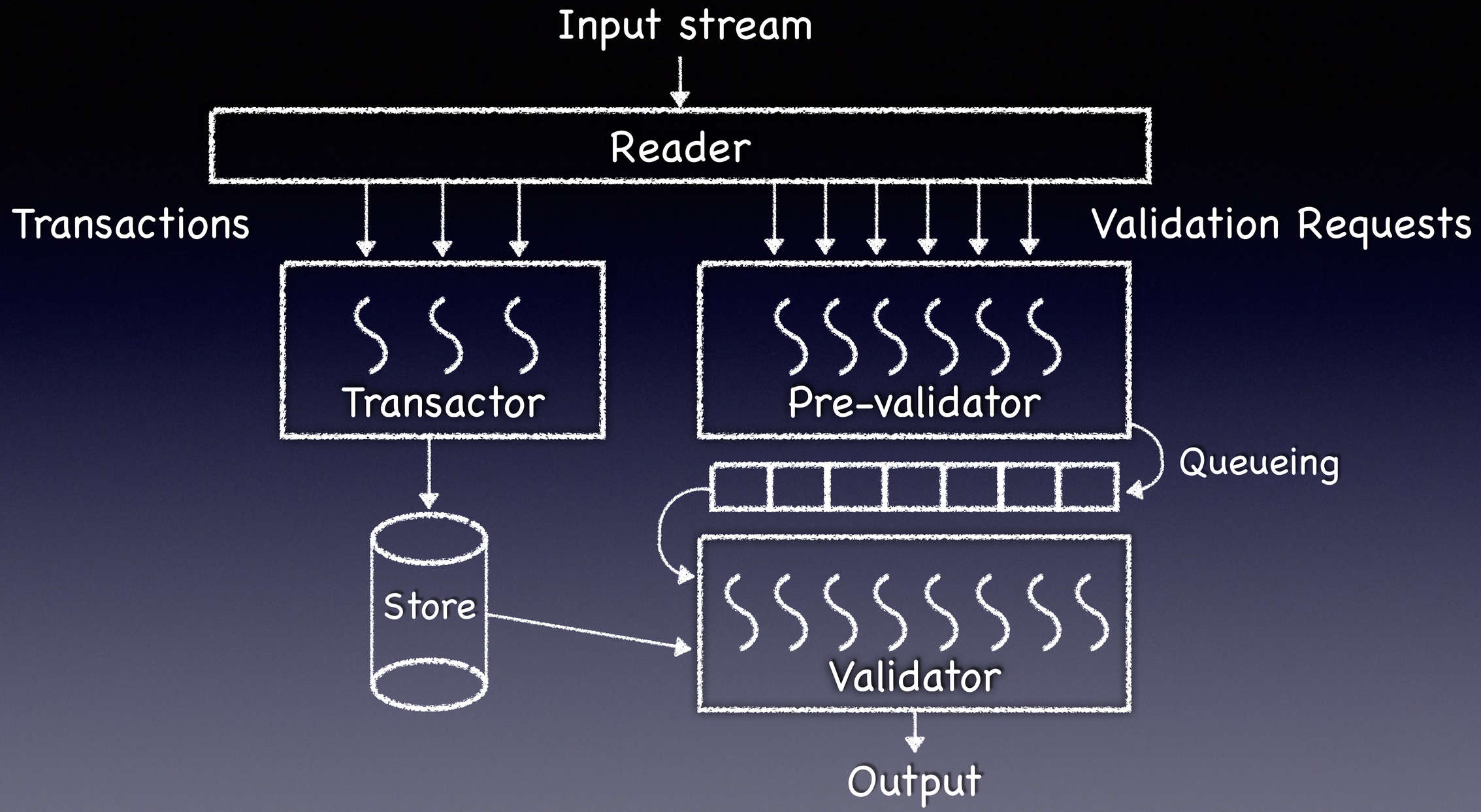
T_{n+3} Insert rows: 2380
Delete rows: 1643

T_{n+4} Insert rows: 1
Delete rows: 3

…

Small-sized transactions are substantial

Large-sized transactions are exceptional



Design Overview

Transaction Format

Delete **Insert**

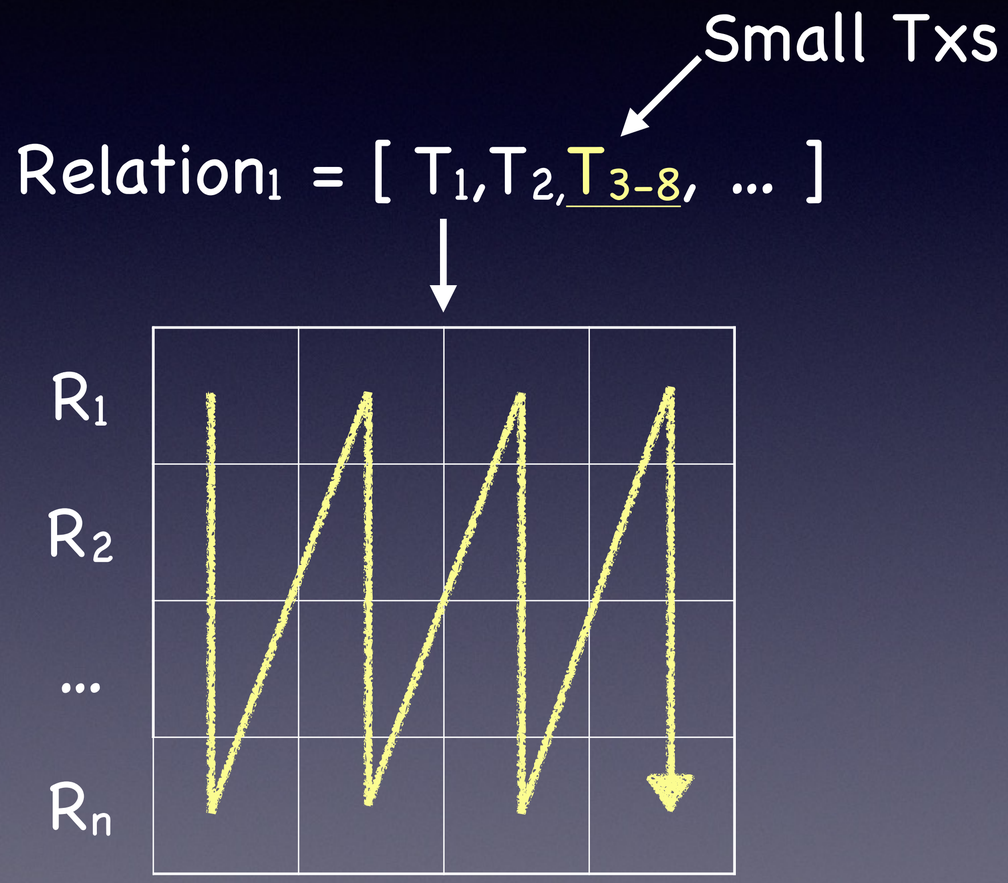
$Transaction_1$ = [A [2] , A [8], A [2, 3, 4], B [4, 5, 6, 7]]

- Delete
  - Rows with PK 2 and 8 in relation A
- Insert
  - Tuple (2, 3, 4) into relation A
  - Tuple (4, 5, 6, 7) into relation B

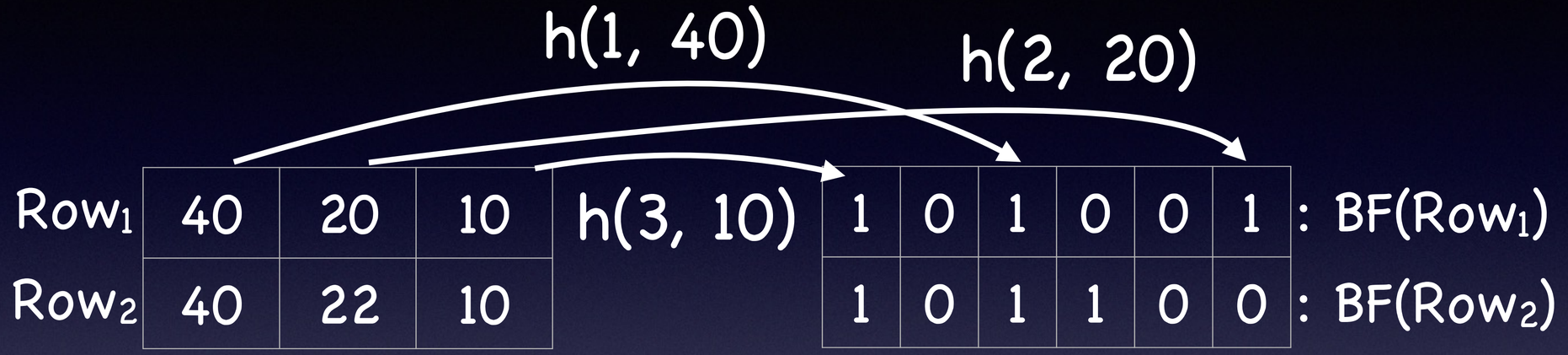
Relation A =
[[2, 8, (2, 3, 4)], T_2, T_3, …]

Storing Transactions

- Represent a relation as a list of transactions
- Save each transaction in column-store
- Group small-sized transactions into one



Simple Bloom Filter



- Construct bloom filters for each row
  - [40, 20, 10] ⇒ {(1, 40), (2, 20), (3, 10)}
- Only a single hash function is used for performance issues

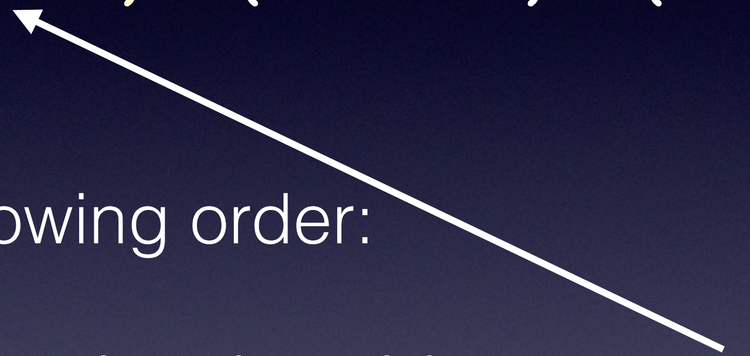
Validation Requests

- Validation requests are given in DNF
  - e.g. (Relation 1: $Q_1 \wedge Q_2 \wedge Q_3$) $\vee$ (Relation 2: $Q_4 \wedge Q_5$)
- Independently process each conjunction
- Optimizations:
  - Check if queries are within [min, max] range
  - Reorder queries
  - Utilize bloom filters for equal queries

Reordering Queries

Original: $(c1 < v1) \wedge (c2 == v2) \wedge (c3 == v3) \wedge (c4 >= v4)$

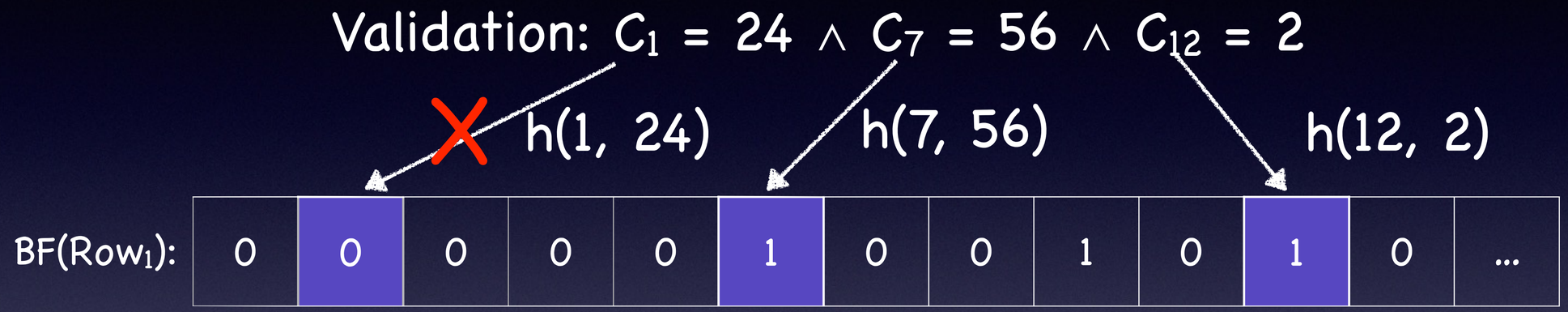
Reordered: $(c2 == v2) \wedge (c3 == v3) \wedge (c1 < v1) \wedge (c4 >= v4)$



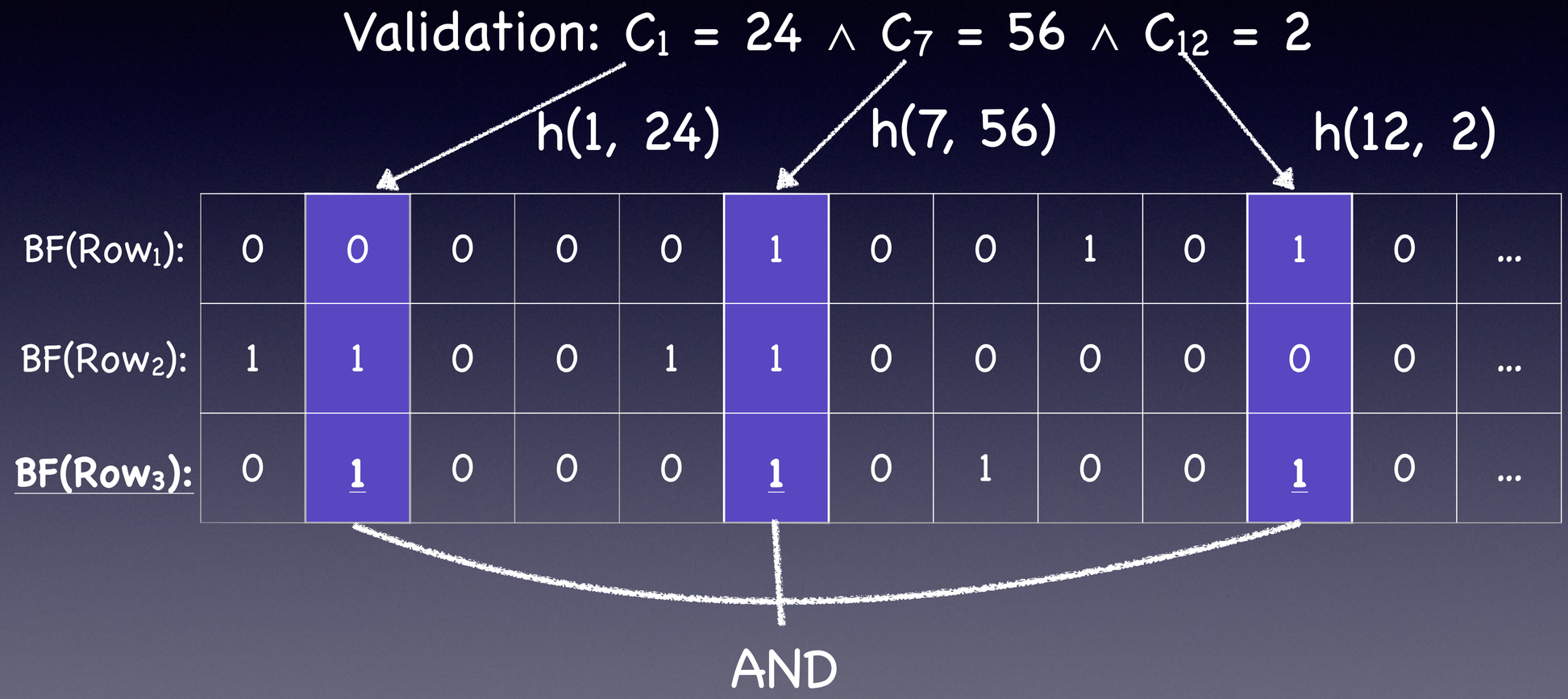
Reorder queries in the following order:

1. Query with = operator having the widest range of values
2. Queries with = operators
3. Queries with other operators

Bloom Filter

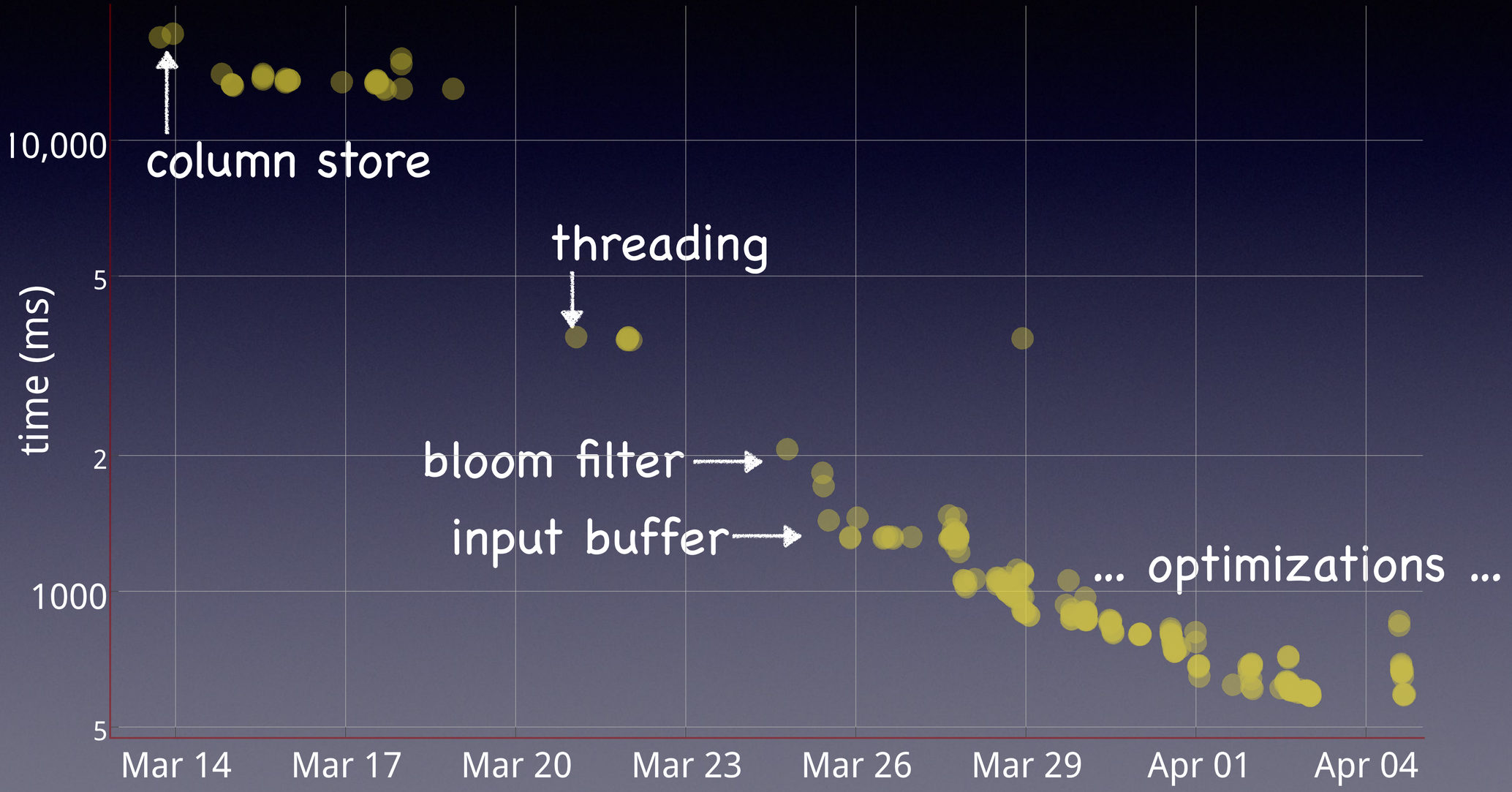


Bloom Filter



- Using AVX SIMD bitwise AND (`VANDPS`)

Test Scores


10,000
5
2
1000
5
time (ms)
column store
threading
bloom filter
input buffer
... optimizations ...
Mar 14
Mar 17
Mar 20
Mar 23
Mar 26
Mar 29
Apr 01
Apr 04

Thank you