Flow Join

Adaptive Skew Handling for Distributed Joins

UTN - CloudDB Project SS24
Irene Santana Martin, Tim Leonard, Luca Heller

Agenda

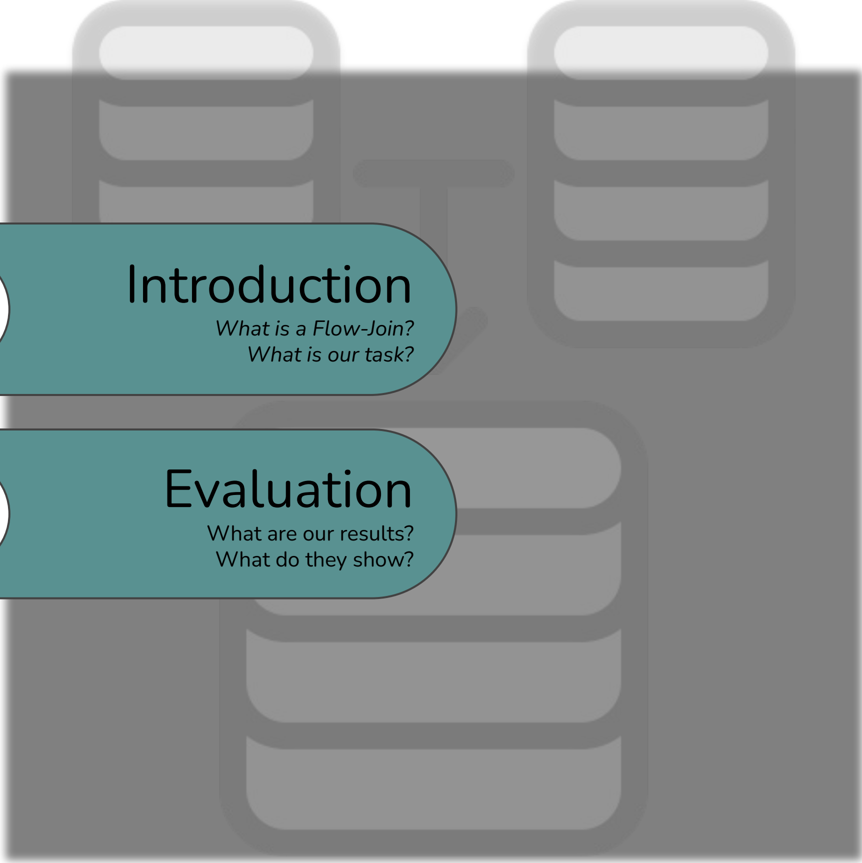
1 **Problematic**
What problem are we trying to solve?

2 **Introduction**
What is a Flow-Join?
What is our task?

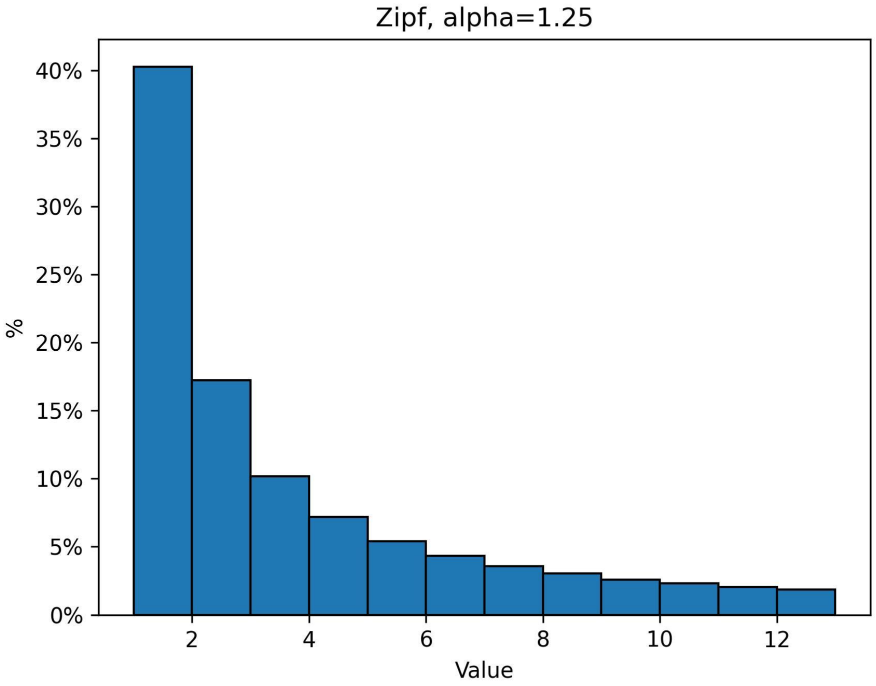
3 **Implementation**
How does our solution look like?

4 **Evaluation**
What are our results?
What do they show?

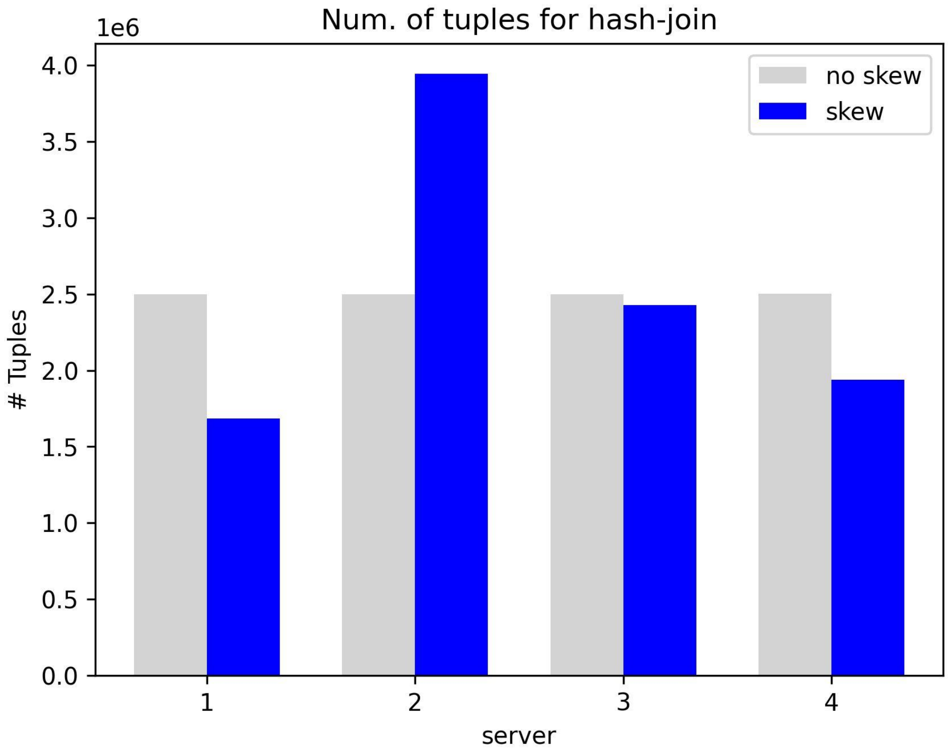
5 **Conclusion**



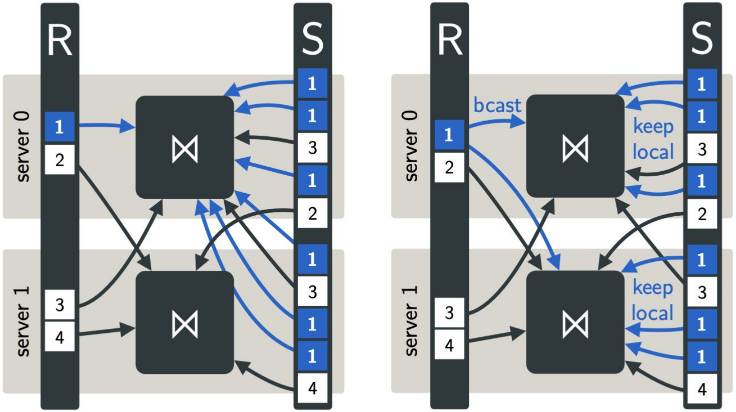
The Problem of Skewed Data



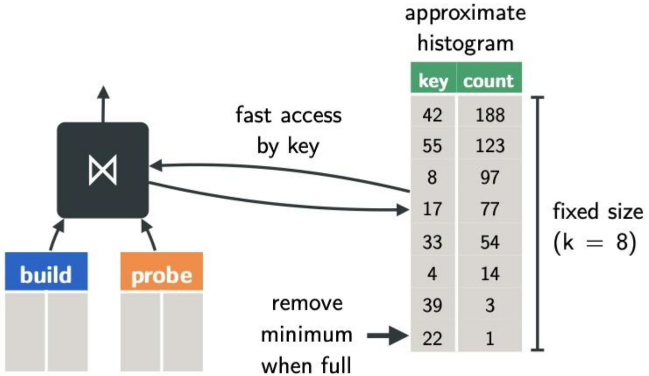
The Problem of Skewed Data



Flow Join (Rödiger W. et al.)



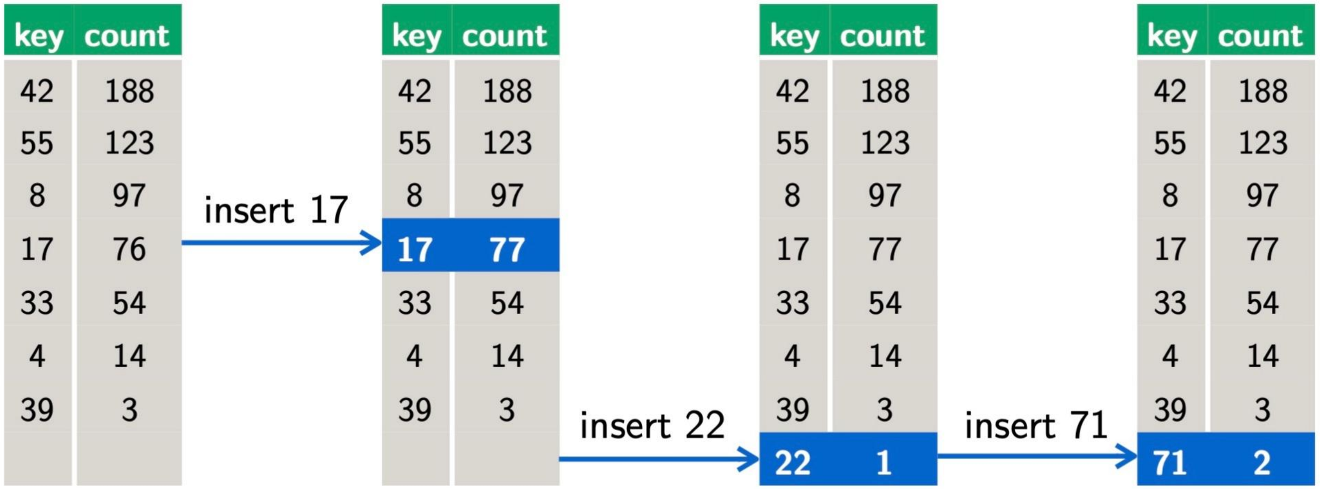
Source [1], Fig. 4



key	count
42	188
55	123
8	97
17	77
33	54
4	14
39	3
22	1

Source [1], Fig. 5

SpaceSaving Algorithm (Metwally A. et al.)



key	count
42	188
55	123
8	97
17	76
33	54
4	14
39	3

insert 17

key	count
42	188
55	123
8	97
17	**77**
33	54
4	14
39	3

insert 22

key	count
42	188
55	123
8	97
17	77
33	54
4	14
39	3
22	**1**

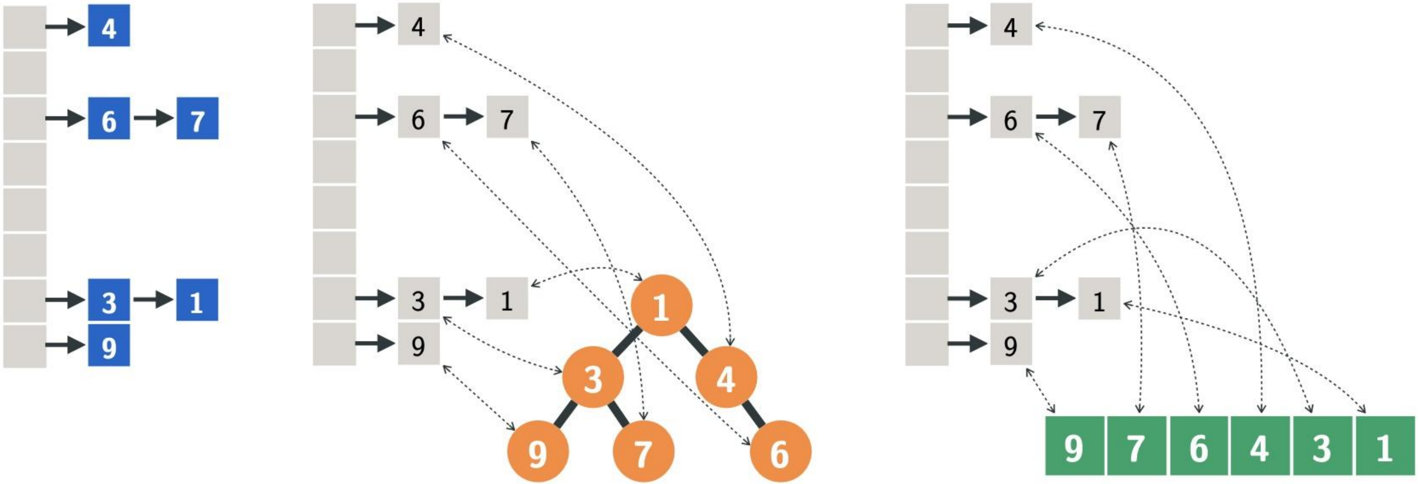
insert 71

key	count
42	188
55	123
8	97
17	77
33	54
4	14
39	3
71	**2**

Source [1], Fig. 7

Implementation

Comparison Data Structures

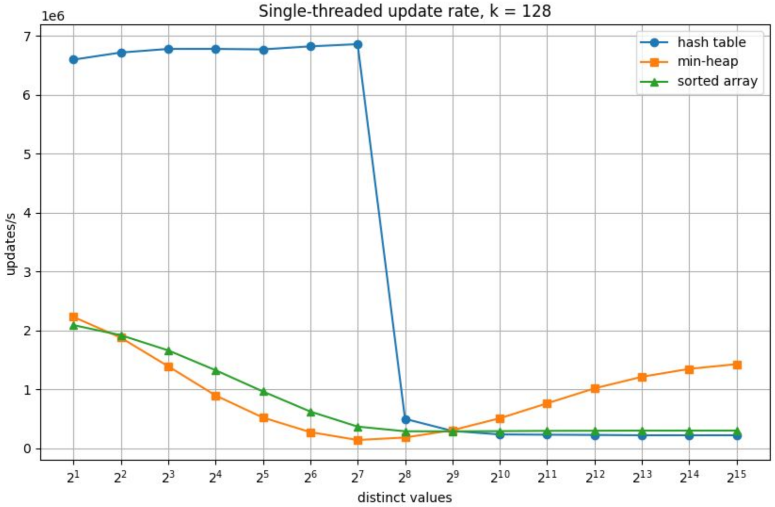


(a) Hash table (b) Hash table + heap (c) Hash table + sorted array

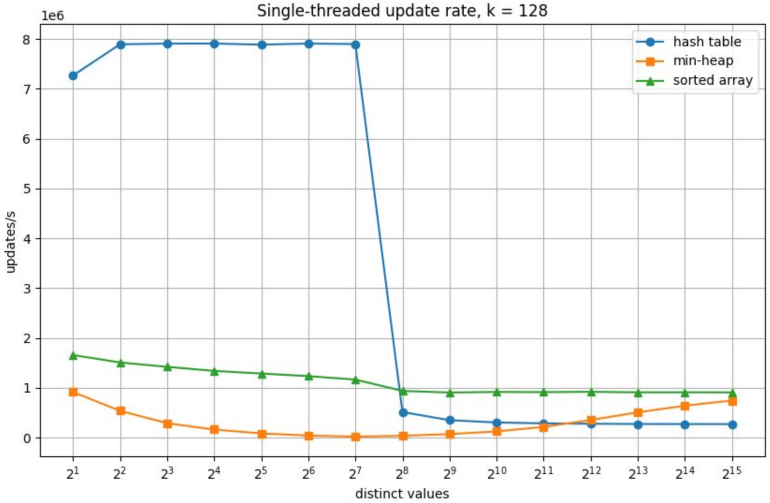
Source [1], Fig. 6

Comparison Data Structures

Performance: Python vs. C++



Python



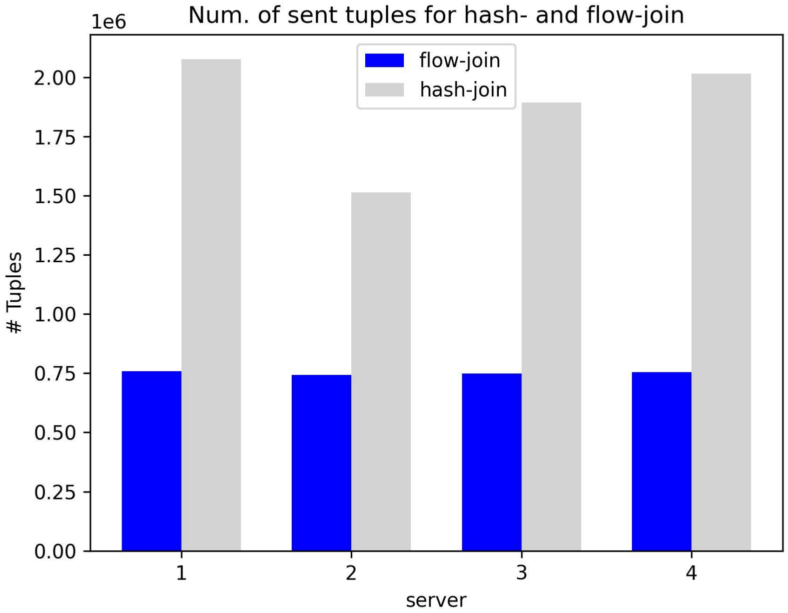
C++

Results

Results

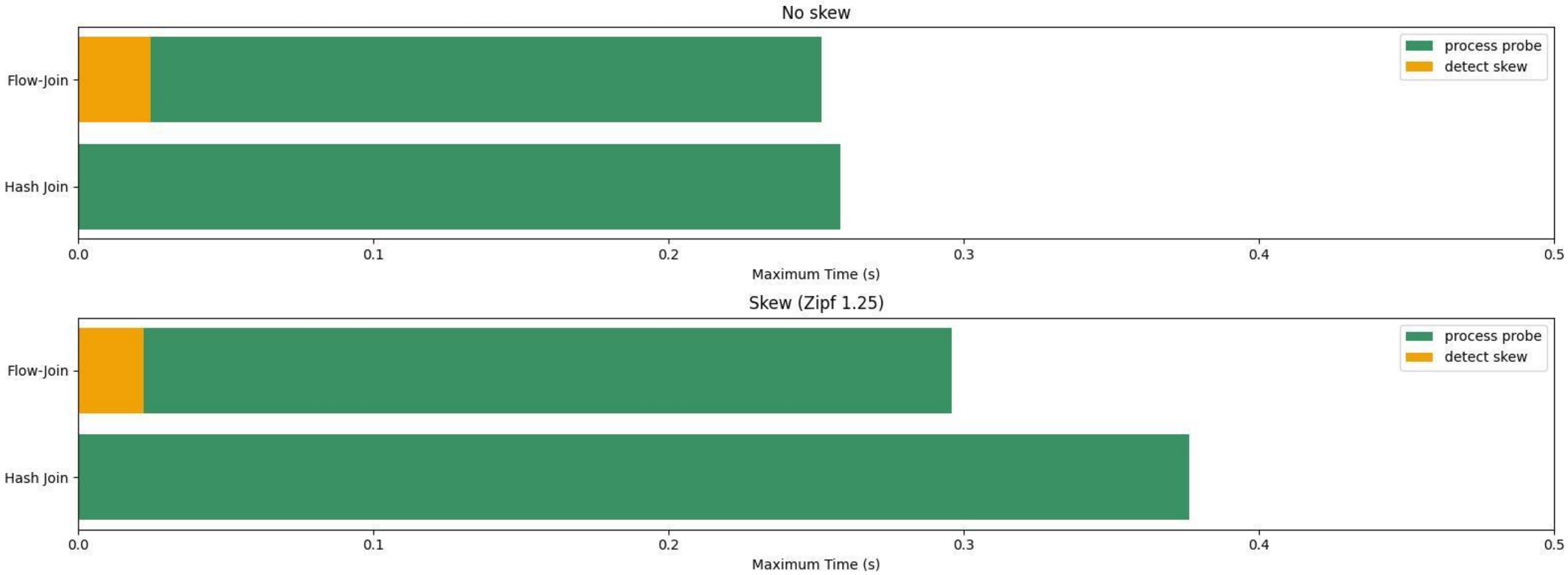
Comparison Hash Join vs Flow Join

Number of tuples sent



Results

Comparison Hash Join vs Flow Join



Conclusion

Thank you for your attention!

Q&A

Resources

[1] Roediger W. et al. *Flow-Join: Adaptive Skew Handling for Distributed Joins over High-Speed Networks*. 2016

[2] Metwally A. et al. *Efficient Computation of Frequent and Top-k Elements in Data Streams*. In *ICDT*, pages 398-412, 2005