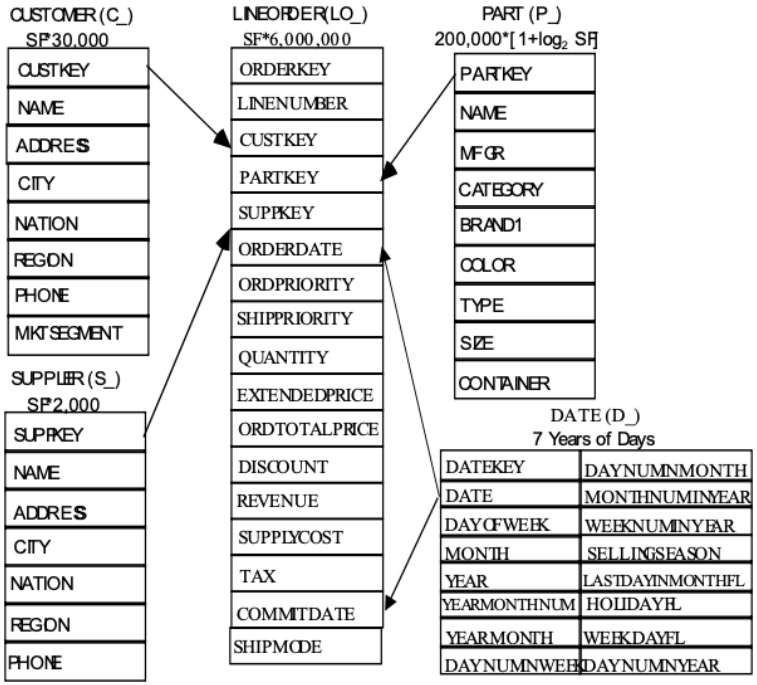
Accelerating Joins with Filters: Keeping a Limited Memory is Robust

Nicholas Corrado Xiating Ouyang

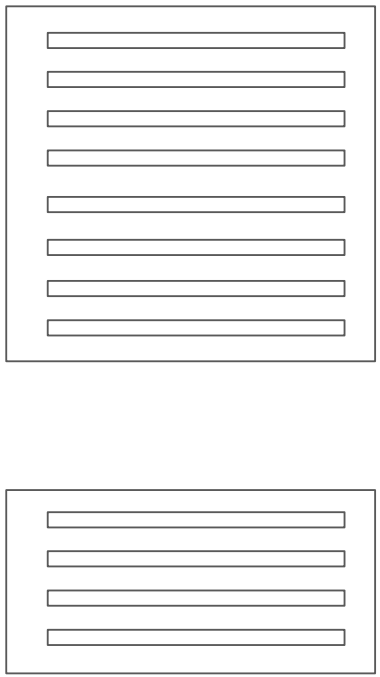
University of Wisconsin-Madison

Star Schema

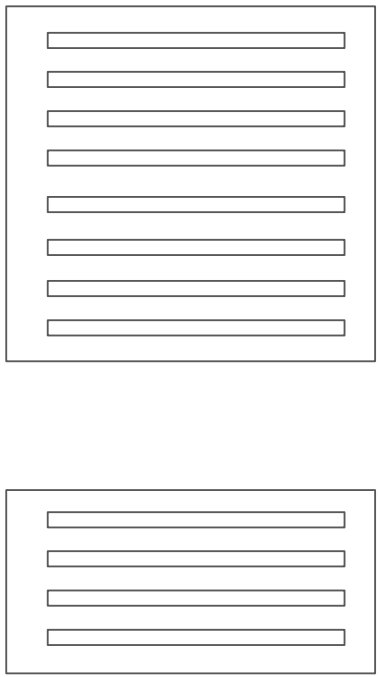


- If the query optimizer chooses a poor join order, intermediate join results may be unnecessarily large.
- Solution: try to filter out extraneous tuples before performing joins

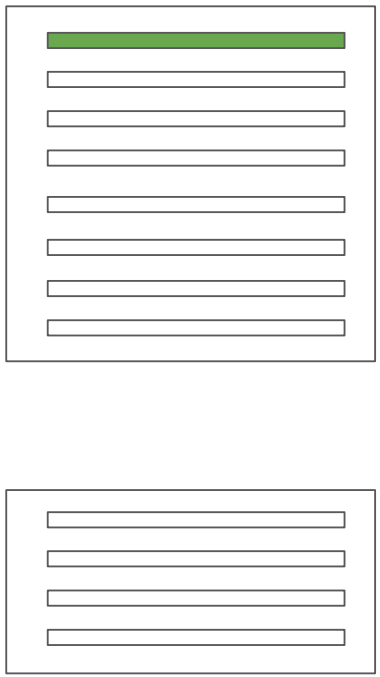
Lookahead Information Passing (LIP)



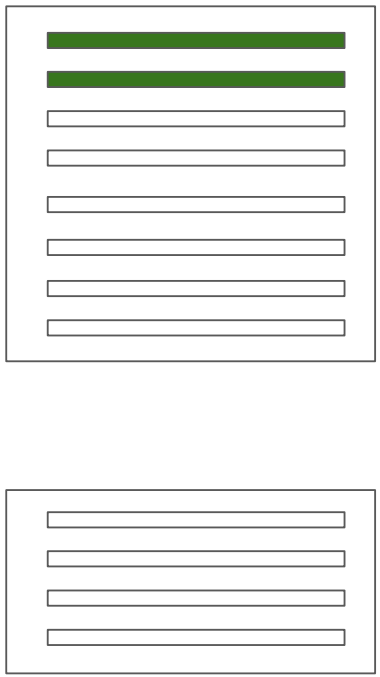
Lookahead Information Passing (LIP)



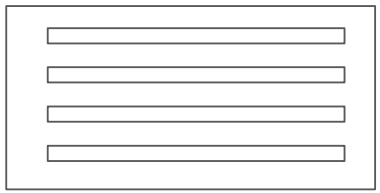
Lookahead Information Passing (LIP)



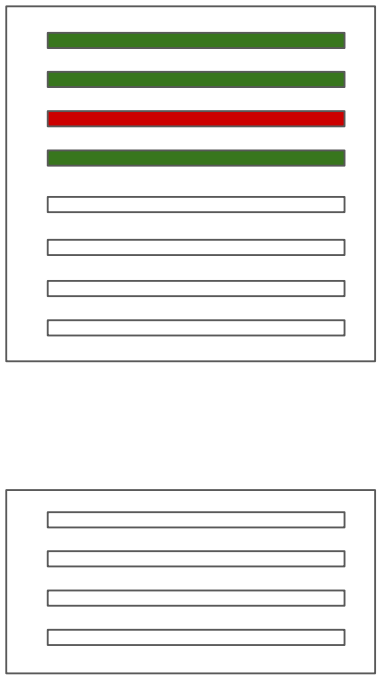
Lookahead Information Passing (LIP)



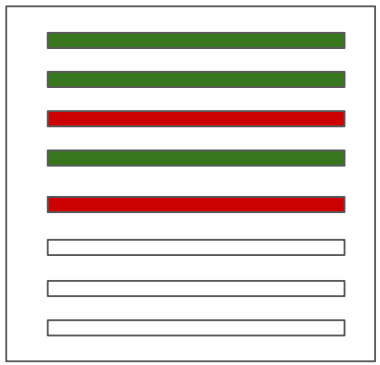
Lookahead Information Passing (LIP)

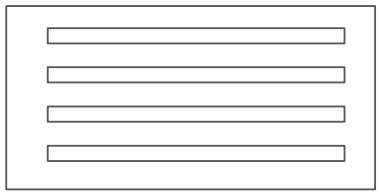


Lookahead Information Passing (LIP)

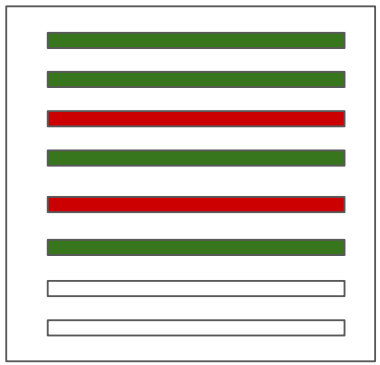


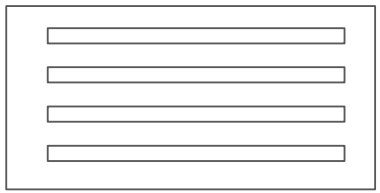
Lookahead Information Passing (LIP)



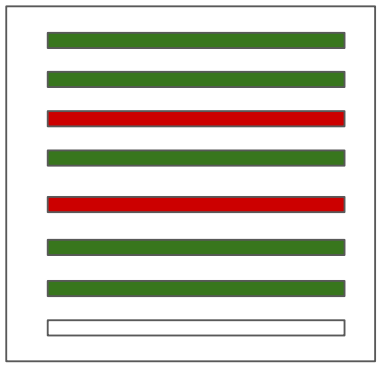


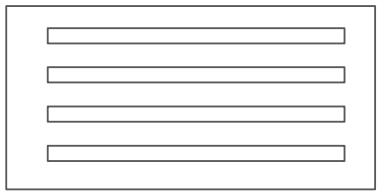
Lookahead Information Passing (LIP)



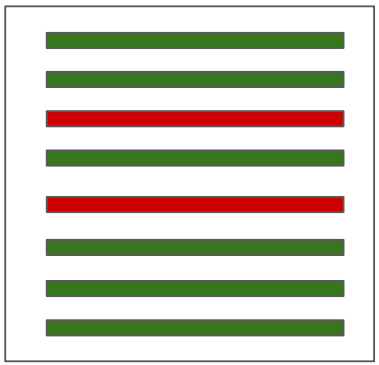


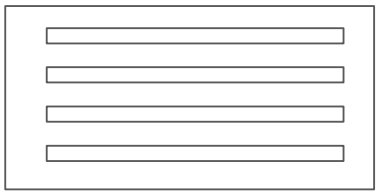
Lookahead Information Passing (LIP)



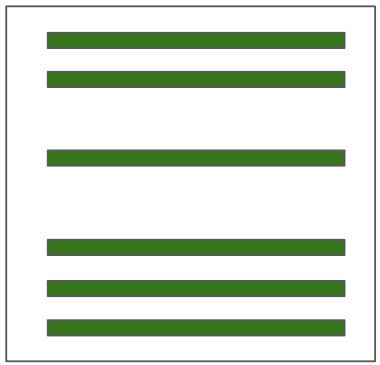


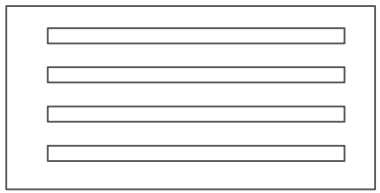
Lookahead Information Passing (LIP)



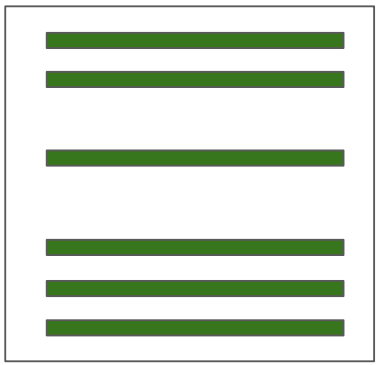


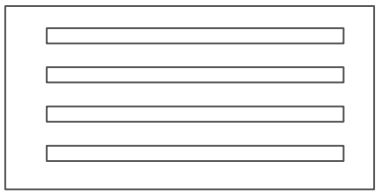
Lookahead Information Passing (LIP)



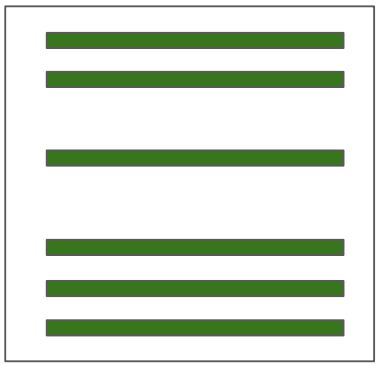


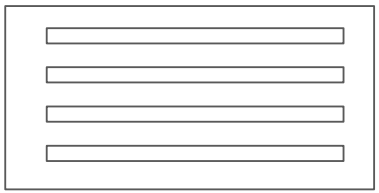
Lookahead Information Passing (LIP)





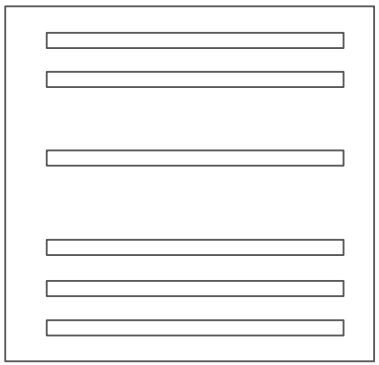
Lookahead Information Passing (LIP)

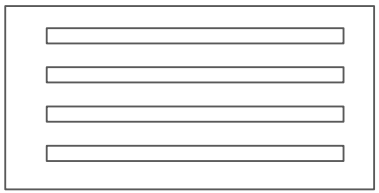




6/8

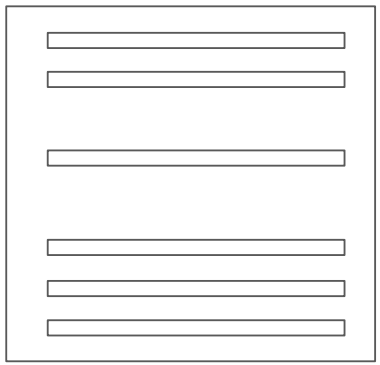
Lookahead Information Passing (LIP)

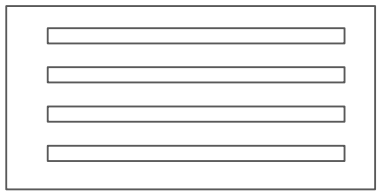




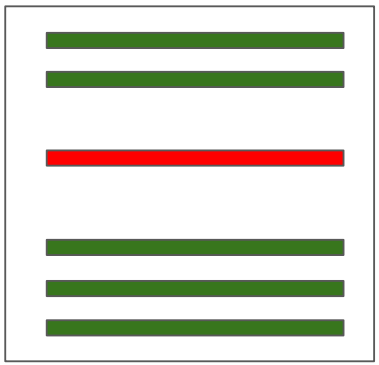
6/8

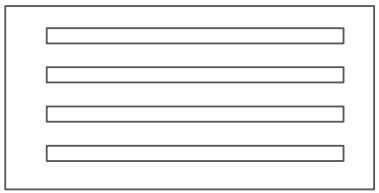
Lookahead Information Passing (LIP)

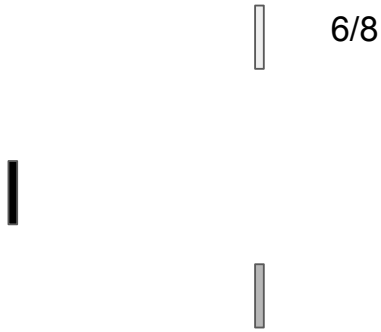




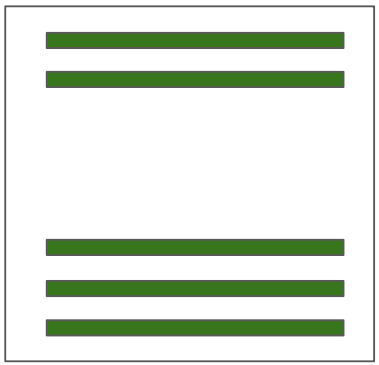
Lookahead Information Passing (LIP)

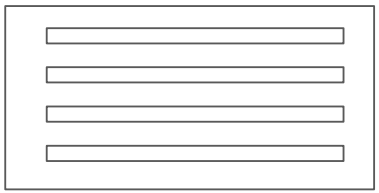


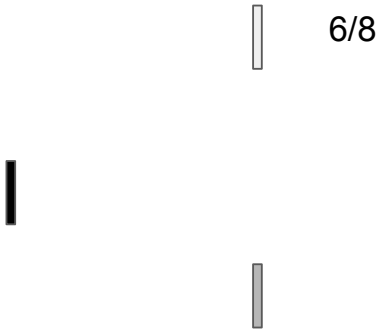




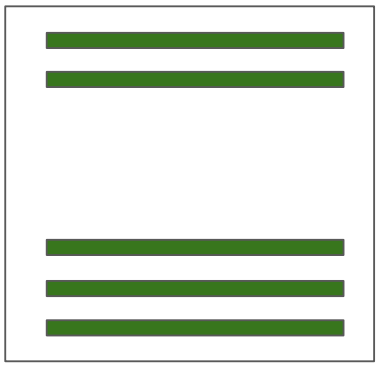
Lookahead Information Passing (LIP)

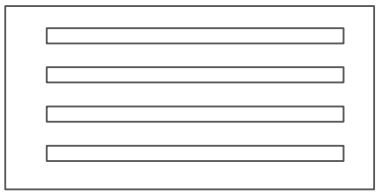






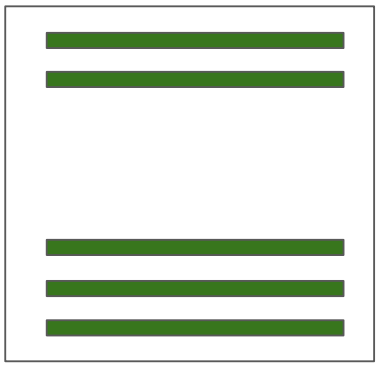
Lookahead Information Passing (LIP)

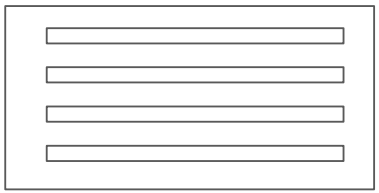




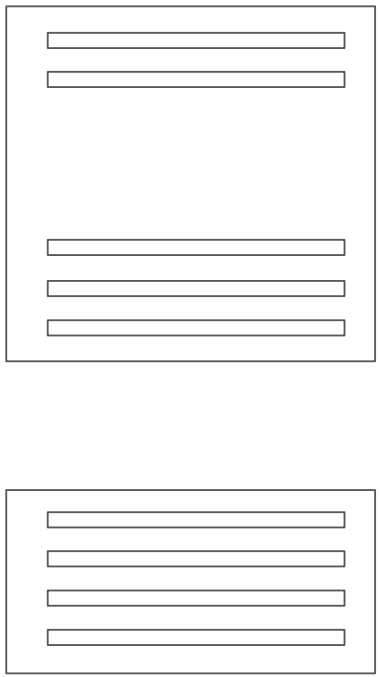
6/8

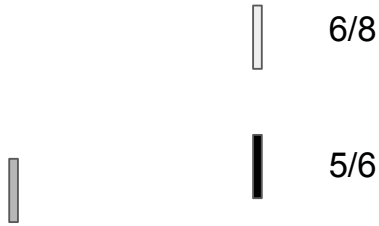
Lookahead Information Passing (LIP)



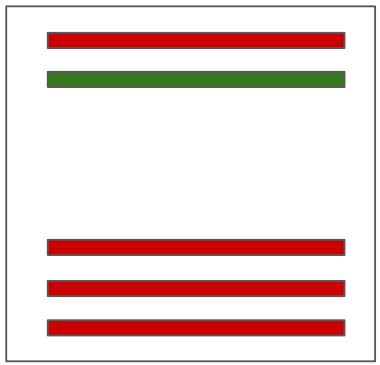


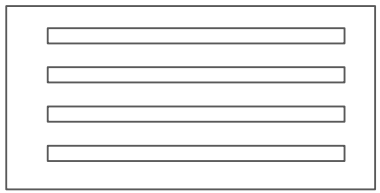
Lookahead Information Passing (LIP)

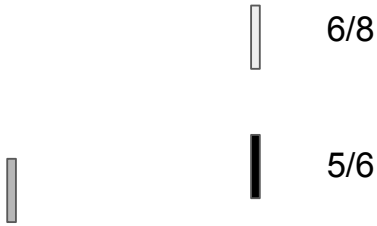




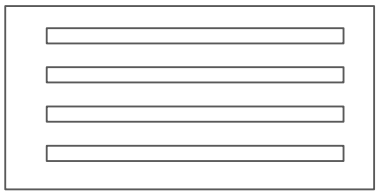
Lookahead Information Passing (LIP)

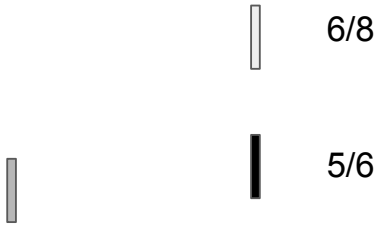




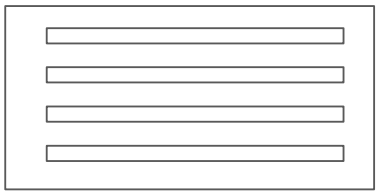


Lookahead Information Passing (LIP)

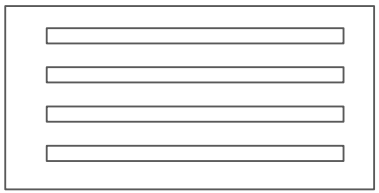




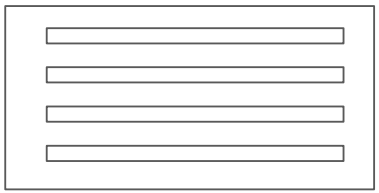
Lookahead Information Passing (LIP)



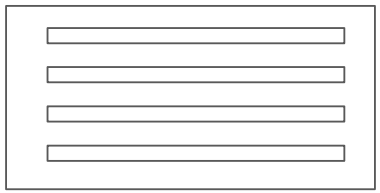
Lookahead Information Passing (LIP)

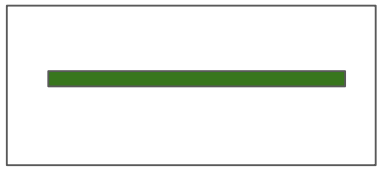


Lookahead Information Passing (LIP)

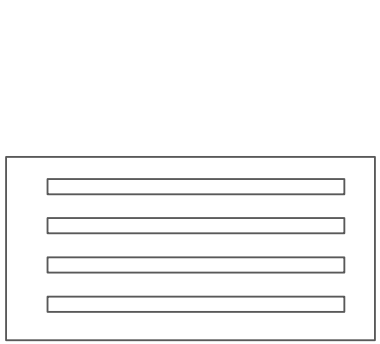


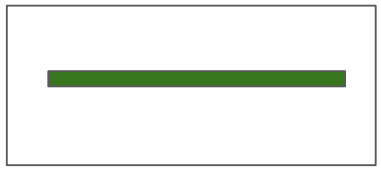
Lookahead Information Passing (LIP)





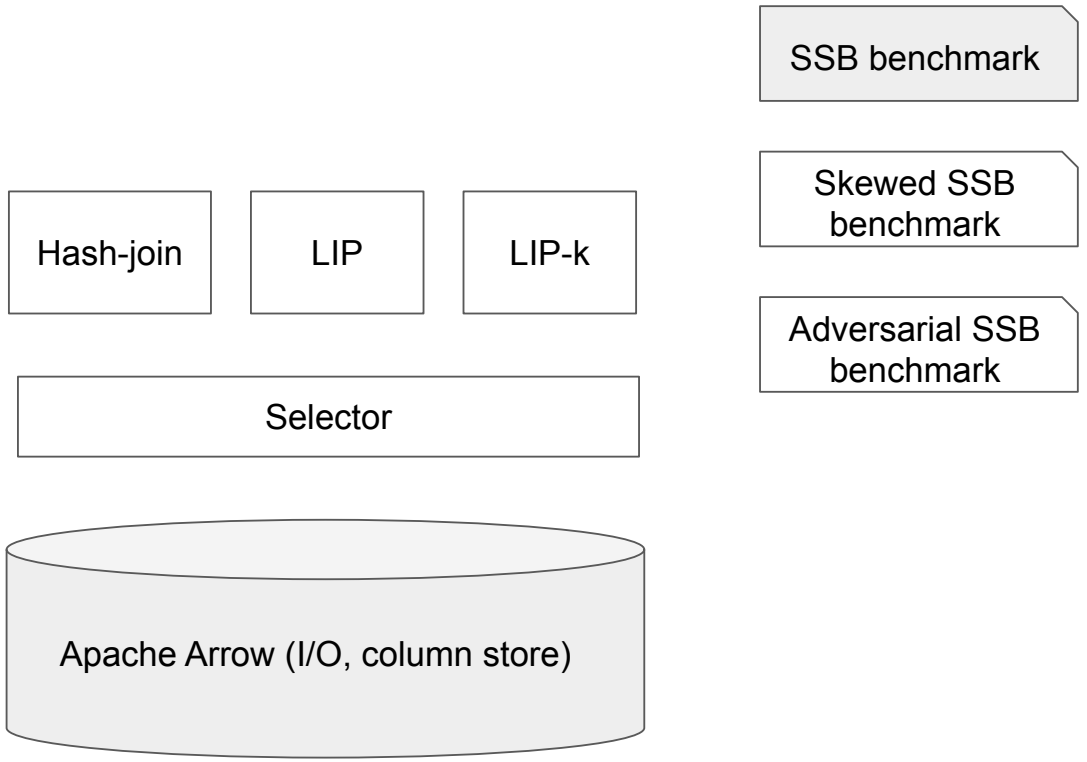
Lookahead Information Passing (LIP)



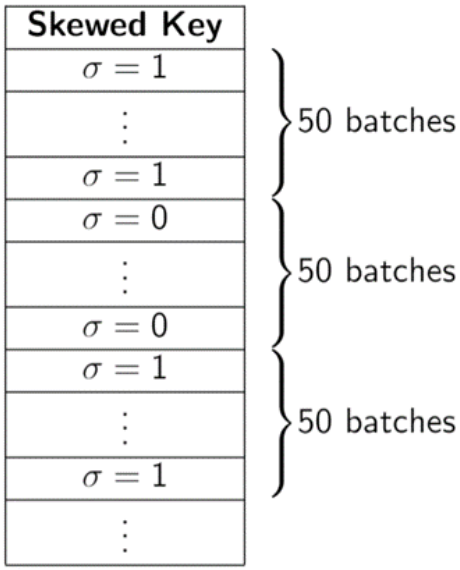


- LIP uses statistics from all previous batches to compute σ
  - Slow response to local changes in key distributions
- **LIP-k**: Only use the previous k batches to compute σ

Implementation and benchmarking

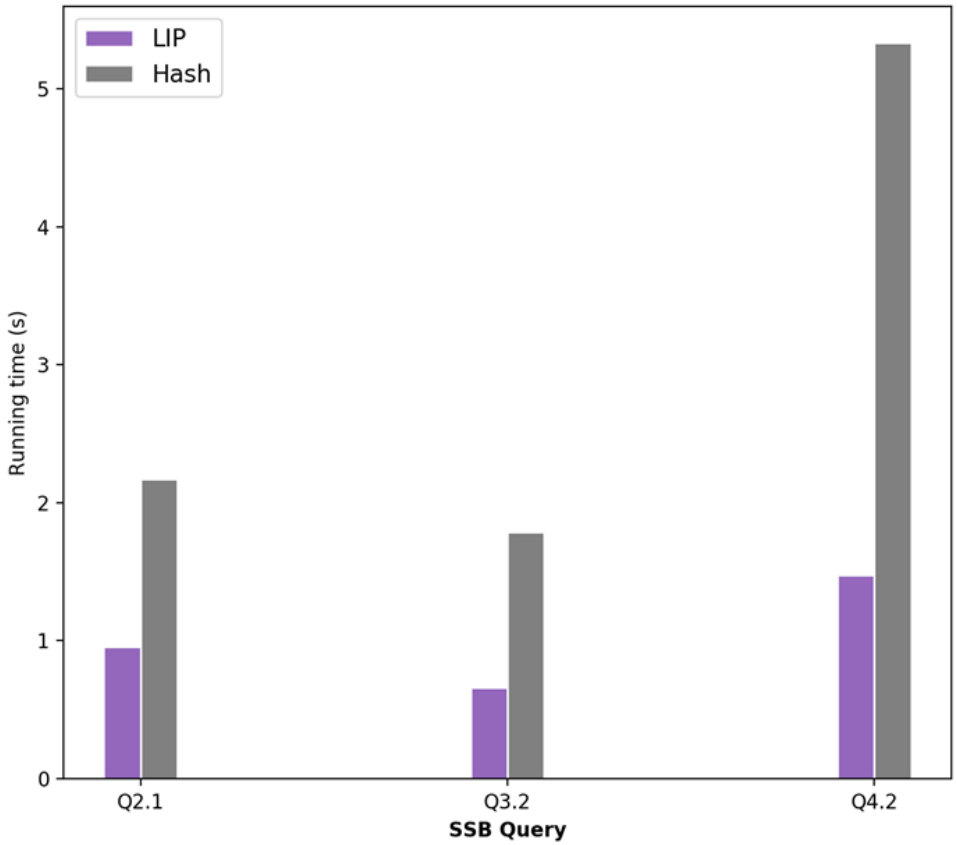


An Example Experiment



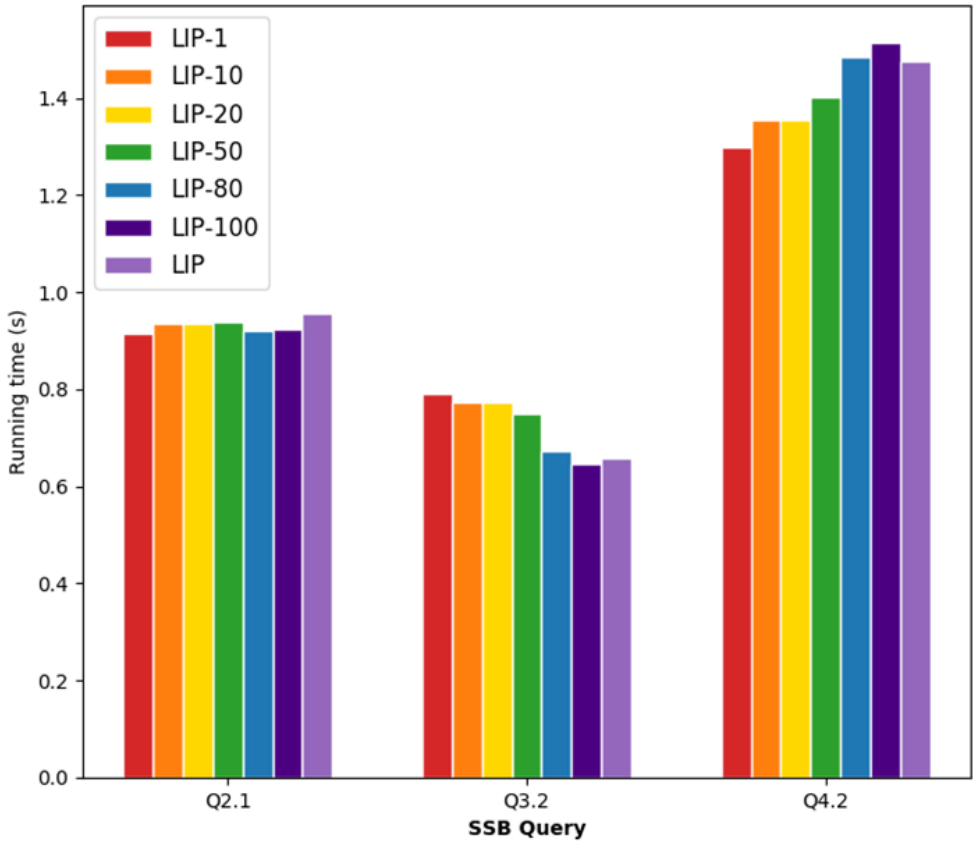
An Example Experiment

Skewed Key	
$\sigma = 1$	50 batches
⋮	
$\sigma = 1$	
$\sigma = 0$	50 batches
⋮	
$\sigma = 0$	
$\sigma = 1$	50 batches
⋮	
$\sigma = 1$	
⋮	



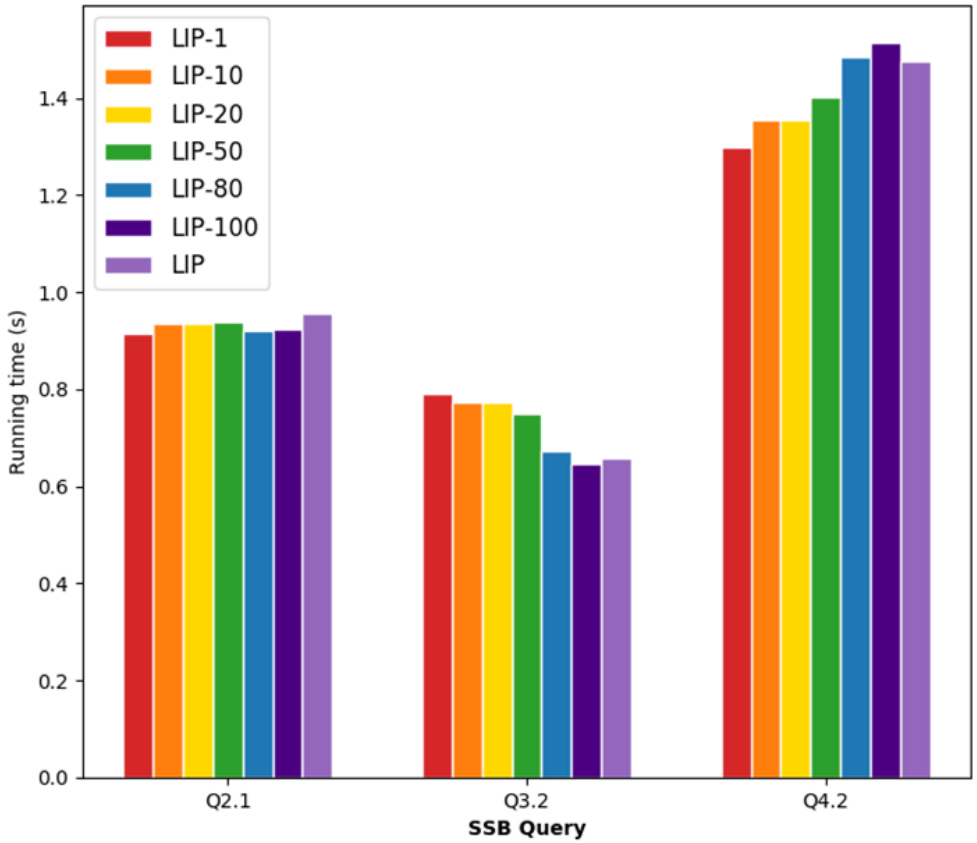
An Example Experiment

Skewed Key	
$\sigma = 1$	50 batches
⋮	
$\sigma = 1$	
$\sigma = 0$	50 batches
⋮	
$\sigma = 0$	
$\sigma = 1$	50 batches
⋮	
$\sigma = 1$	
⋮	



An Example Experiment

Skewed Key	
$\sigma = 1$	50 batches
⋮	
$\sigma = 1$	
$\sigma = 0$	50 batches
⋮	
$\sigma = 0$	
$\sigma = 1$	50 batches
⋮	
$\sigma = 1$	
⋮	



- LIP-k performs better than LIP on some queries...

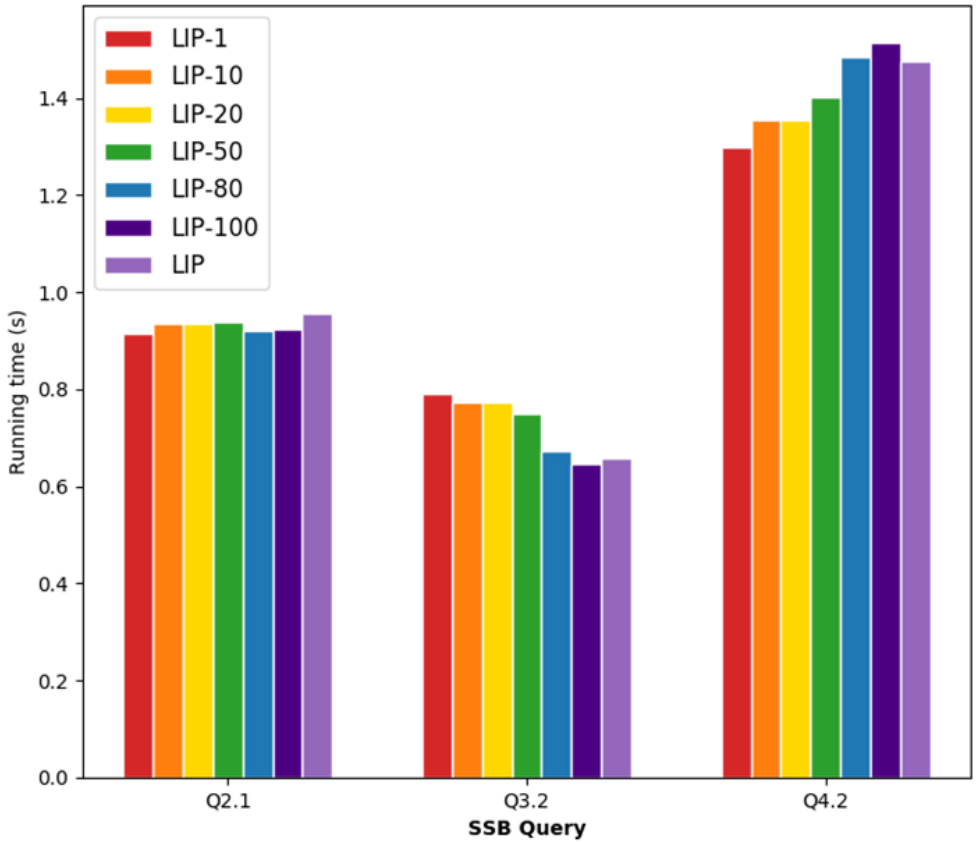
An Example Experiment

Skewed Key
$\sigma = 1$
⋮
$\sigma = 1$
$\sigma = 0$
⋮
$\sigma = 0$
$\sigma = 1$
⋮
$\sigma = 1$
⋮

50 batches

50 batches

50 batches



- LIP-k performs better than LIP on some queries...
- ...but LIP performs better on others

LIP is solving an online problem

- Tuples arriving one at a time
- Upon arrival, decide a sequence of filters
- Minimize the total probes
- Deterministic!

LIP is solving an online problem

- Tuples arriving one at a time
- Upon arrival, decide a sequence of filters
- Minimize the total probes
- Deterministic!

Theorem

There is no deterministic mechanism $\mathcal{M}$ for LIP *achieving a competitive ratio less than N, where N is the number of filters used in* LIP.

LIP is solving an online problem

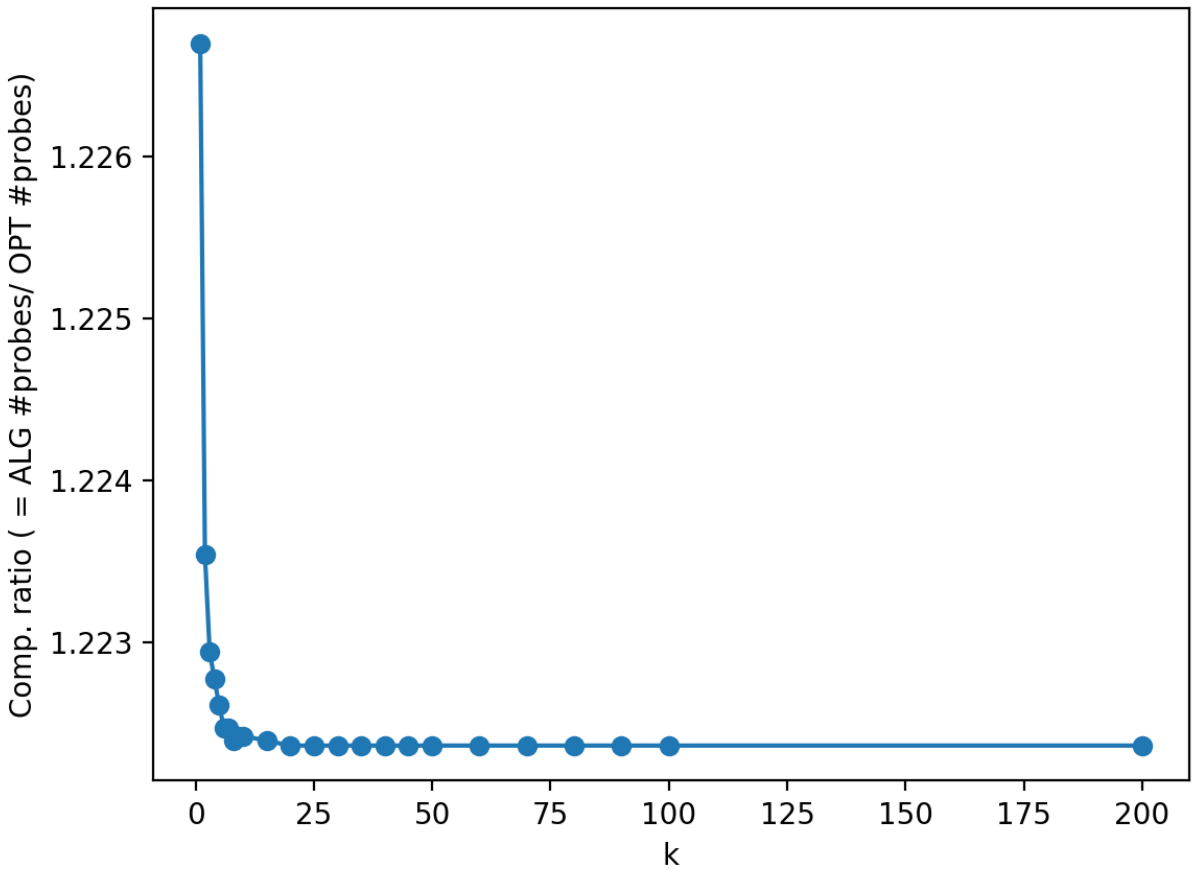
- Tuples arriving one at a time
- Upon arrival, decide a sequence of filters
- Minimize the total probes
- Deterministic!

Theorem

There is no deterministic mechanism $\mathcal{M}$ for LIP *achieving a competitive ratio less than N, where N is the number of filters used in* LIP.

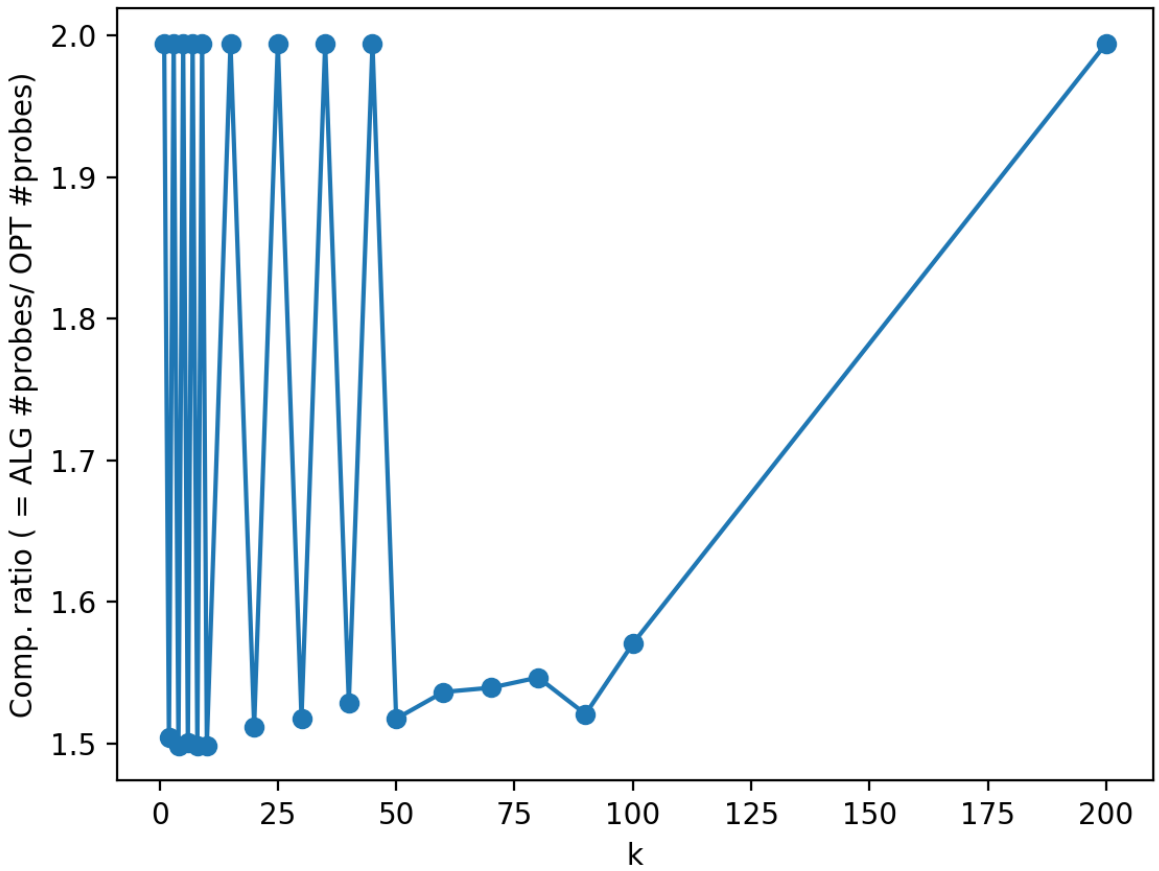
- Not observed in practice, but a theoretical lower bound
- Randomness?

Competitive Ratio vs. k on Uniform Data



Competitive Ratio vs. k on Adversarial Data

- Adversarial data set constructed such that LIP-k has worst case performance for odd k
- Run on query with $N = 2$ joins



Conclusion

- Implemented LIP and its variant LIP-k
- Relative performance of LIP and LIP-k depends on the query
- Can we use randomness to achieve a better robustness guarantee?

Thank you!

Questions?