Streaming HyperCube: A Massively Parallel Stream Join Algorithm

Yuan Qiu, Serafeim Papadias, Ke Yi

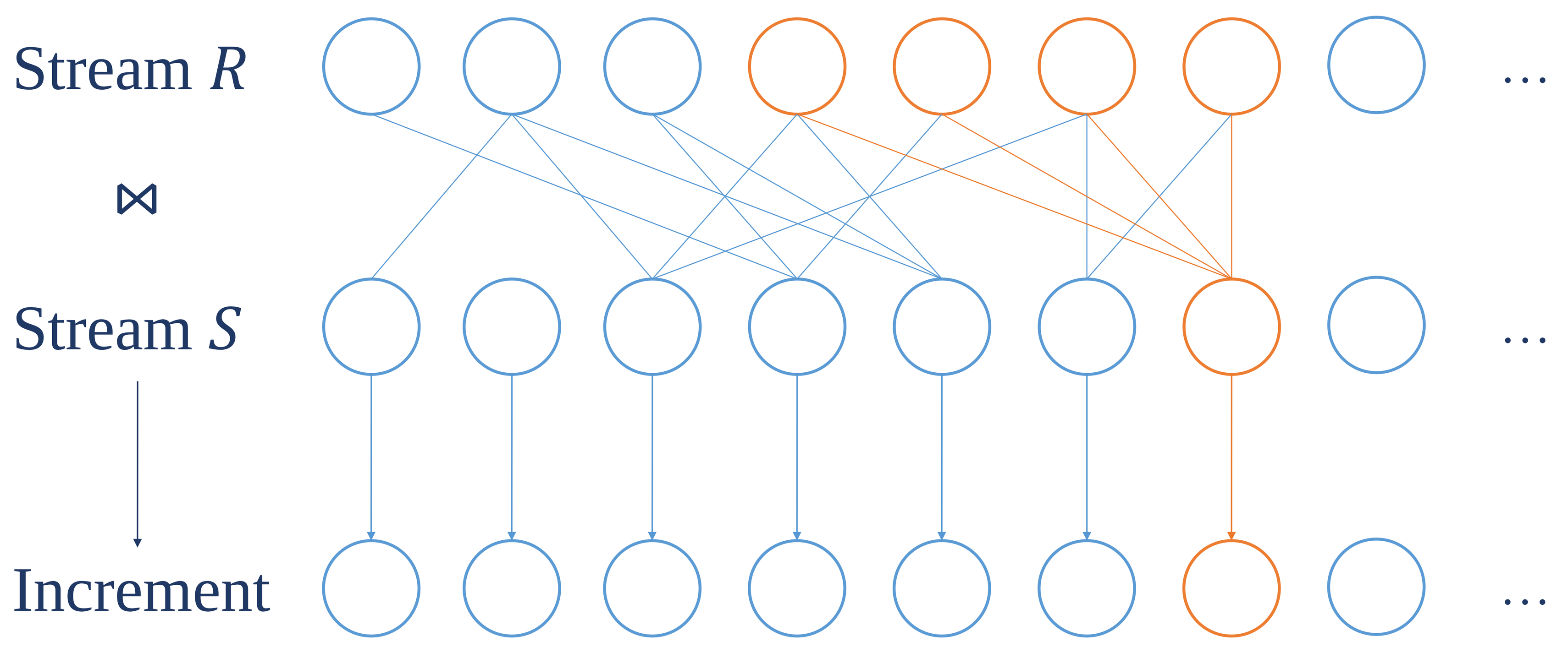
HKUST

{yqiuac, spapadias, yike}@cse.ust.hk

Abstract

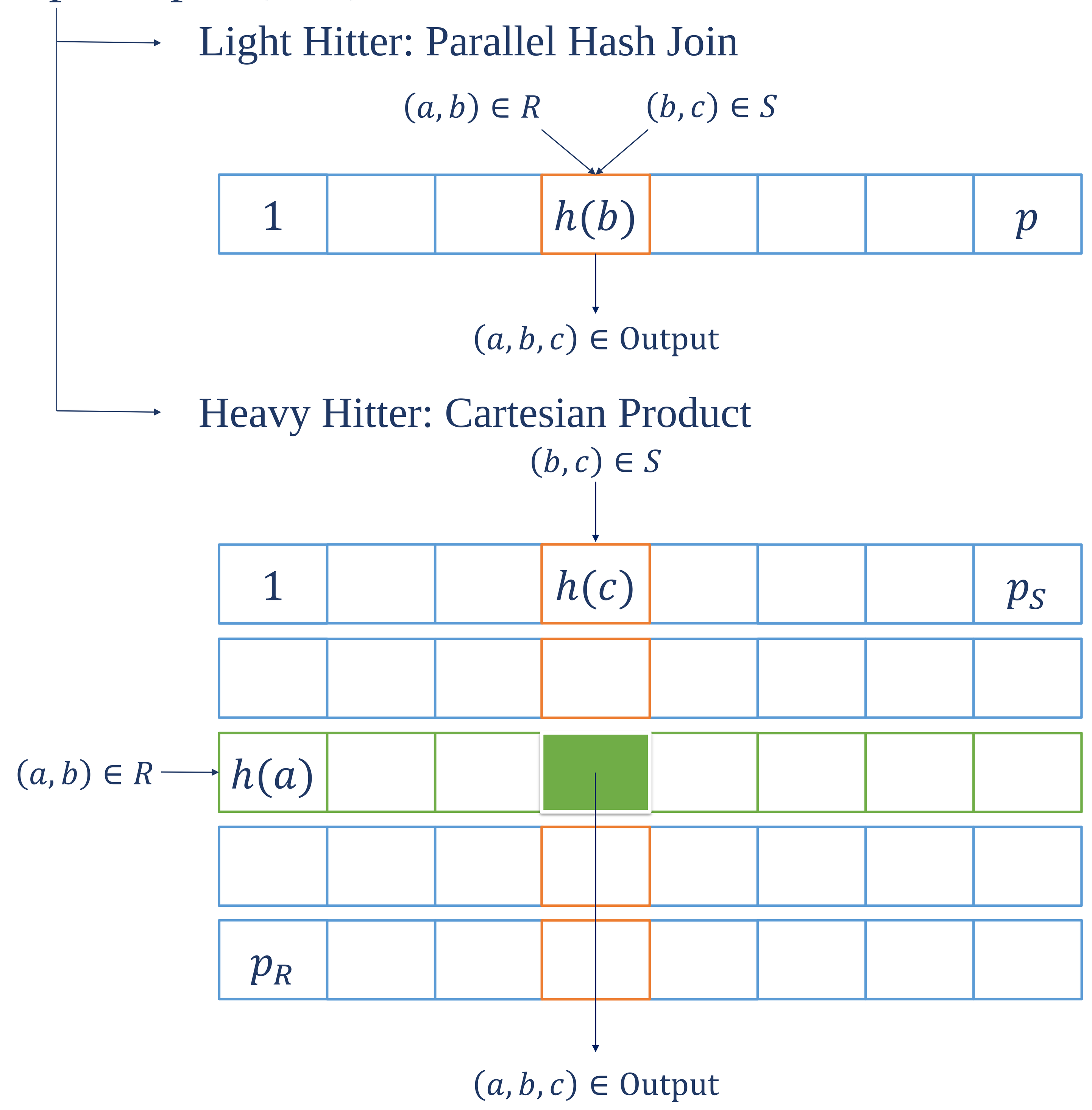
Stream join is an essential operation in many real-time applications. On static data, the HyperCube algorithm ensures a balanced load across all processors in an optimal way. We extend this algorithm to the streaming setting, which can adapt the HyperCube configuration depending on the current data distribution.

Stream Join



HyperCube

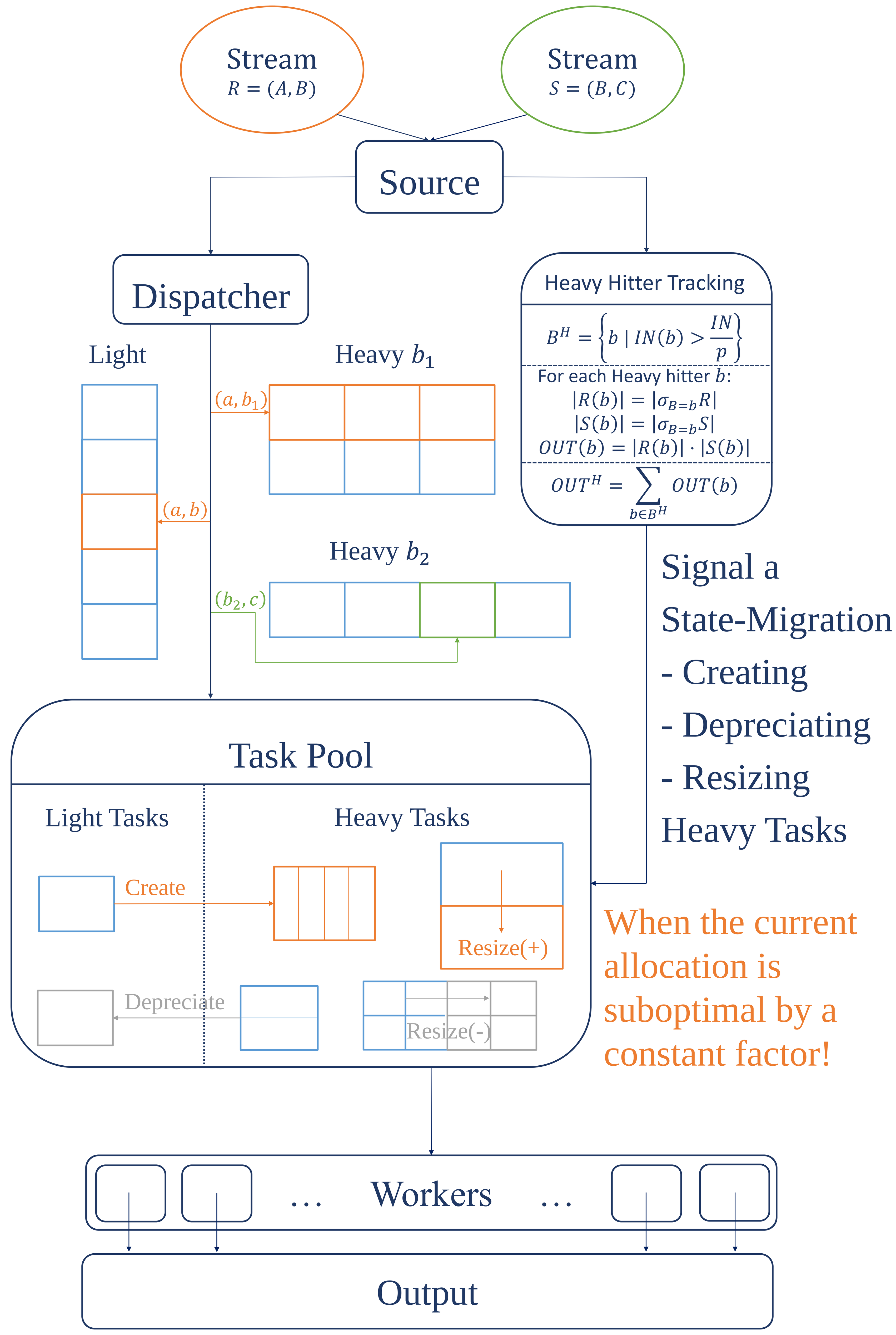
Input tuple $(a, b) \in R$:



Challenge

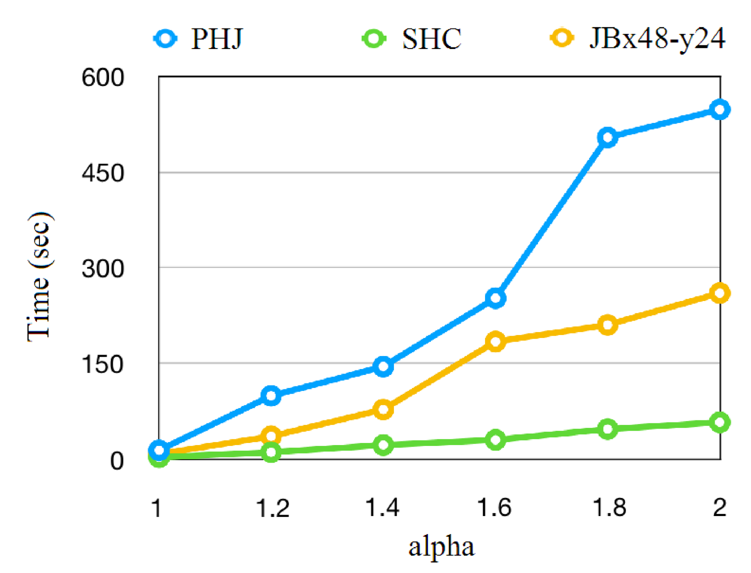
- All heavy hitter information is needed to decide the configuration of each cube.
- The heavy hitter set may change throughout the stream processing.

System Architecture

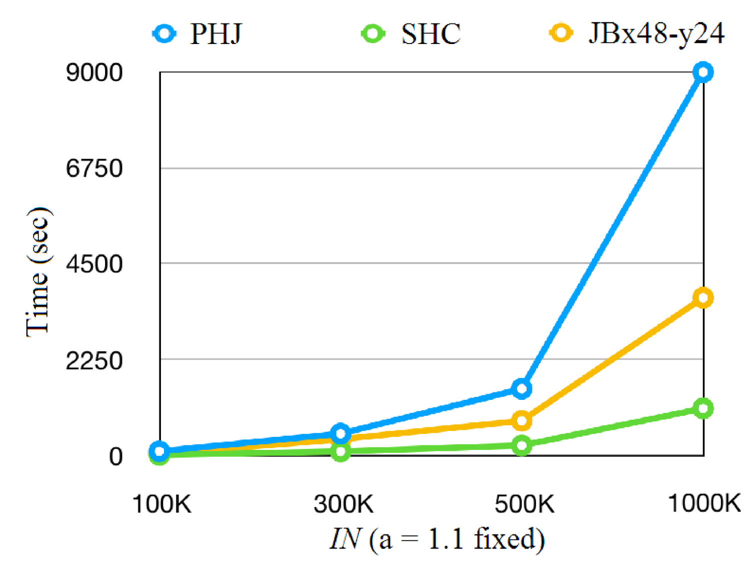


Evaluation

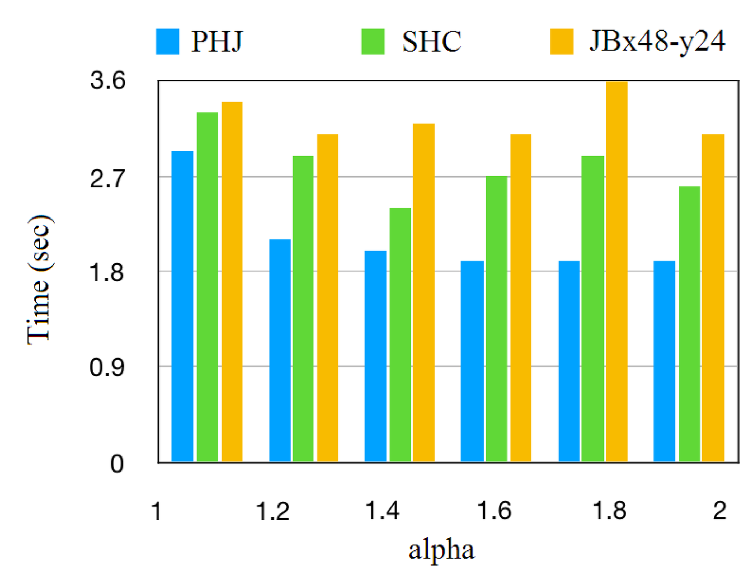
- Parallel Hash Join
- Streaming HyperCube
- Join-Biclique



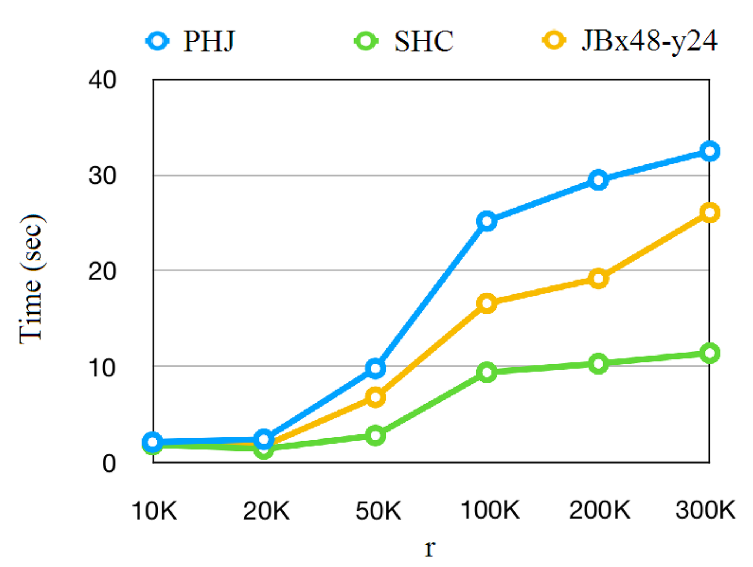
Zipf Data, Varying Skewness α



Zipf Data, Varying Input Size



TPC-H Data, Varying Skewness α



COREL Data, r defines similarity (Higher skewness with larger r)