Adapting to Workloads Characteristics

We make the following contributions:
1. We identify the workload characteristics which impact the applicability and performance of window aggregation techniques.
2. We contribute general stream slicing, a generally applicable and highly efficient solution for streaming window aggregation.
3. We analyze the implications of workload characteristics and show that stream slicing is generally applicable while offering better performance than existing approaches.

Performance Evaluation

Key findings
- General slicing outperform alternative concepts with respect to throughput and scales to large numbers of concurrent windows.
- Stream slicing and Buckets scale with constant throughput to large fractions of out-of-order tuples and are robust against high delays of these tuples.
- On time-based windows, stream slicing performs diverse distributive and algebraic aggregations with similarly high throughputs. Considering count-based windows and out-of-order tuples, invertible aggregations lead to higher throughputs than not invertible ones.