The goal of this paper is to bridge the gap between stream processing and ad-hoc queries in SPEs by sharing computation and resources.

We define three main requirements for ad-hoc shared stream processing:

- **Integration**: Ad-hoc query processing should be a composable layer which can extend stream operators, such as join, aggregation, and window operators.
- **Consistency**: Ad-hoc query creation and deletion must be performed in a consistent manner and ensure exactly-once semantics and correctness.
- **Performance**: In contrast to state-of-the-art SPEs, ad-hoc SPE should not only maximize data throughput but also query performance via incremental computation and resource sharing.

AStream is the first system that supports distributed ad-hoc stream processing. We design AStream based on the requirements listed above.

### AStream vs Naive Data Model

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
<td>Q4</td>
<td>Q5</td>
<td>Q6</td>
</tr>
</tbody>
</table>

Q1 and Q1 are created. Q2 deleted, Q3 created. Q2 deleted, Q3 created (naive data model)

AStream: Ad-hoc Shared Stream Processing

We design AStream based on the requirements listed above.

### System Architecture

- **Incremental query processing**
- **Memory efficient dynamic slice data structure**
- **Reduced data copy and shuffling**

### Experiments

**Workload Scenario 1**

**Workload Scenario 2**

**Query deployment latency**

Blue boxes indicate join operation between two slices and red boxes show deleted slices

**Abstract**

The goal of this paper is to bridge the gap between stream processing and ad-hoc queries in SPEs by sharing computation and resources. We define three main requirements for ad-hoc shared stream processing:

- **Integration**: Ad-hoc query processing should be a composable layer which can extend stream operators, such as join, aggregation, and window operators.
- **Consistency**: Ad-hoc query creation and deletion must be performed in a consistent manner and ensure exactly-once semantics and correctness.
- **Performance**: In contrast to state-of-the-art SPEs, ad-hoc SPE should not only maximize data throughput but also query performance via incremental computation and resource sharing.

AStream is the first system that supports distributed ad-hoc stream processing. We design AStream based on the requirements listed above.