CSE 544
Principles of Database Management Systems

Magdalena Balazinska

Winter 2009

Lecture 17 - Stream Processing

Announcements

- Remember the final exam this Wednesday
  - During class
  - Open notes and open books
- Next week: project presentations Monday and Thursday
- Final reports due on Friday

References

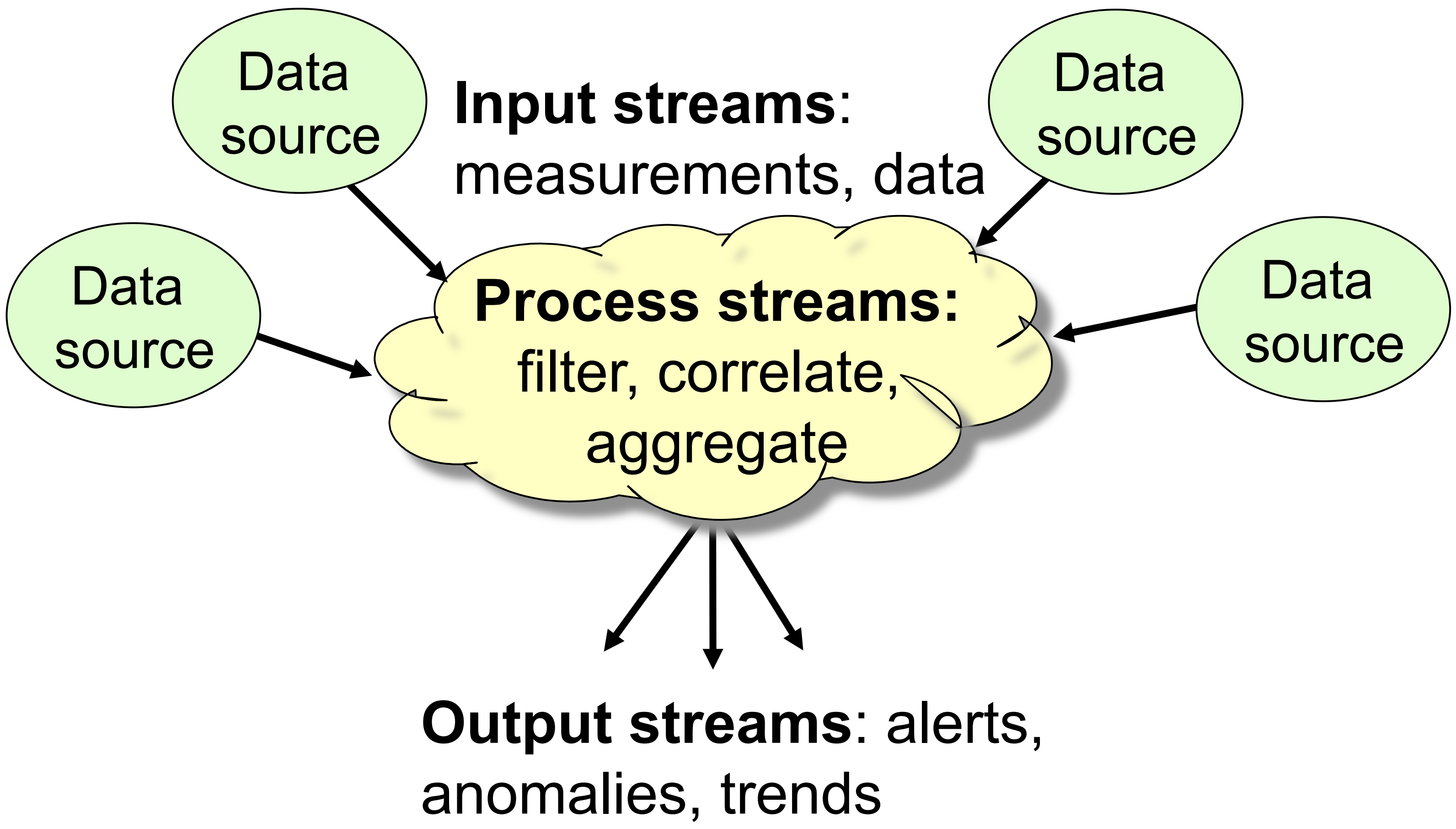
- **Aurora: A New Model and Architecture for Data Stream Management.** Daniel Abadi et. al. VLDB Journal. 12(2). 2003

Outline

- **Stream processing applications**
  - Examples
  - Requirements

- **The Aurora stream processing engine**
  - Stream model and query model
  - Processing model
  - Operators
  - Query examples
  - Other features

Stream Processing


Data source
Data source
Data source
Data source
Input streams: measurements, data
Process streams: filter, correlate, aggregate
Output streams: alerts, anomalies, trends

Application Domains

- Network monitoring
  - Intrusion, fraud, anomaly detection, click streams
- Financial services
  - Market feed processing, ticker failure detection
- Sensor-based environment monitoring
  - Weather conditions, air quality, car traffic
  - Civil engineering, military applications, etc.
- Medical applications
  - Patient monitoring, equipment tracking
- Near real-time data analytics

Requirements

- **Input data is pushed continuously**
  - Traditional DBMSs not designed for continuous loading or inserting of individual data items
  - “DBMS-active, human passive” model
- **Users want to execute continuous queries**
  - Traditional DBMSs have no direct support for such queries. Can use triggers, but triggers do not scale
- **Low-latency processing**
  - Need to see results in near real-time
  - Data is possibly high-volume and high-rate

Other Requirements

- Distribution
- Load management and load shedding
- Approximate processing, approximate answers
- Fault-tolerance and revision processing
- Exploiting data archives

Outline

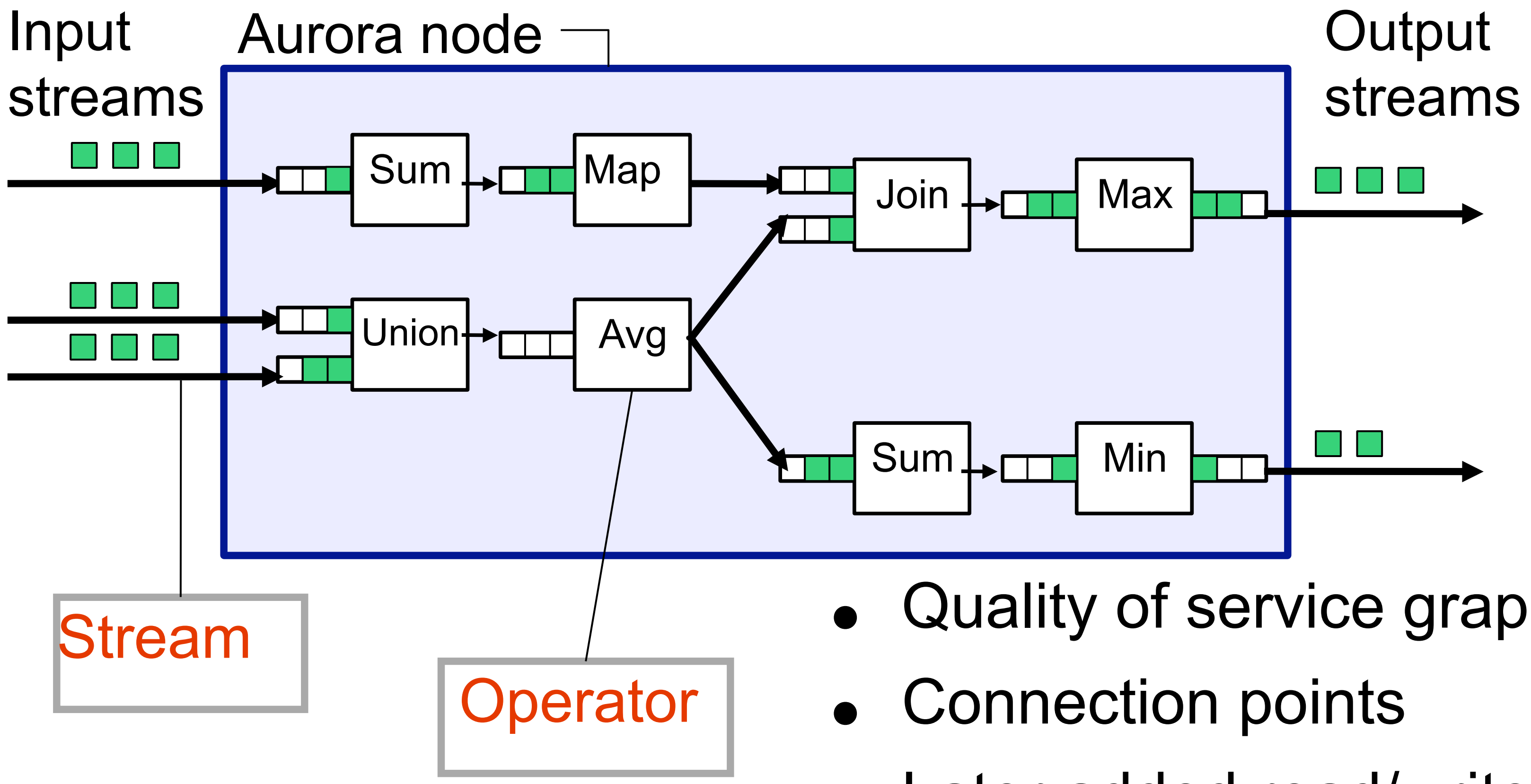
- **Stream processing applications**
  - Examples
  - Requirements
- **The Aurora stream processing engine**
  - Stream model and query model
  - Processing model
  - Operators
  - Query examples
  - Other features

Stream Data Model

Tuple: $\overbrace{(\texttt{timestamp},}^{\text{header}}\ \overbrace{\texttt{v}_1\texttt{,...,v}_n}^{\text{data}})$

- Stream: append-only sequence of tuples
- All tuples on a stream have same schema
- Timestamp is used for QoS

Query Model



- Quality of service graphs
- Connection points
- Later added read/write ops
- No query language

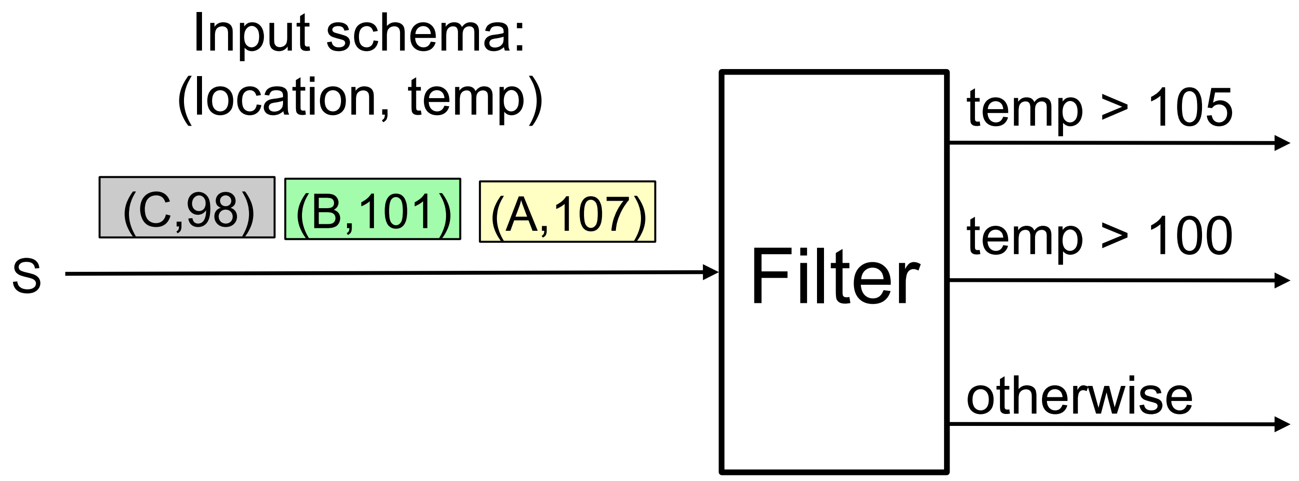
Aurora Operators

- Order-agnostic
  - Filter
  - Map
  - Union
- Order-sensitive
  - Aggregate
  - Join
  - BSort, Resample
- **Why do we need new operators?**
  - Ops cannot block & cannot accumulate state that grows with input

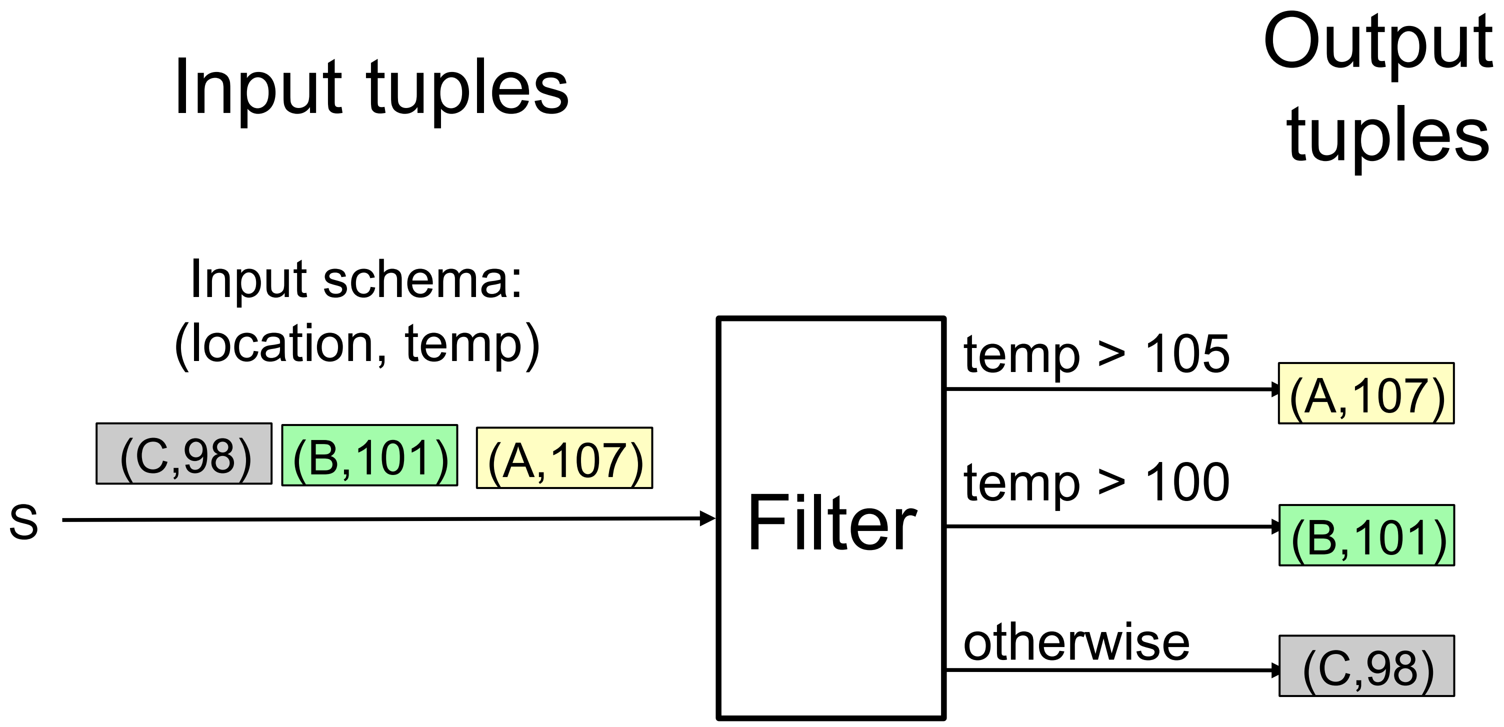
Filter Example

Input tuples

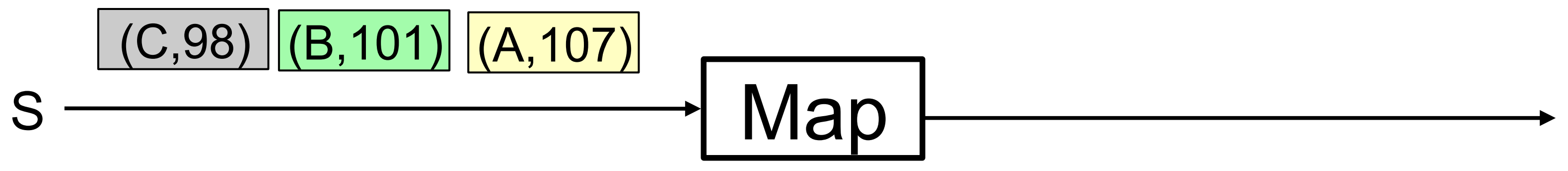
Output tuples



Filter Example

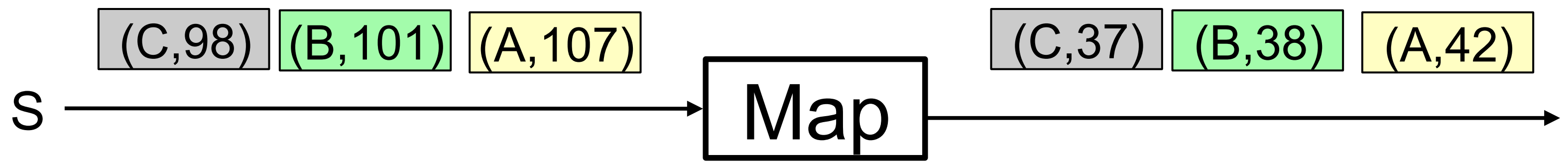


Map Example



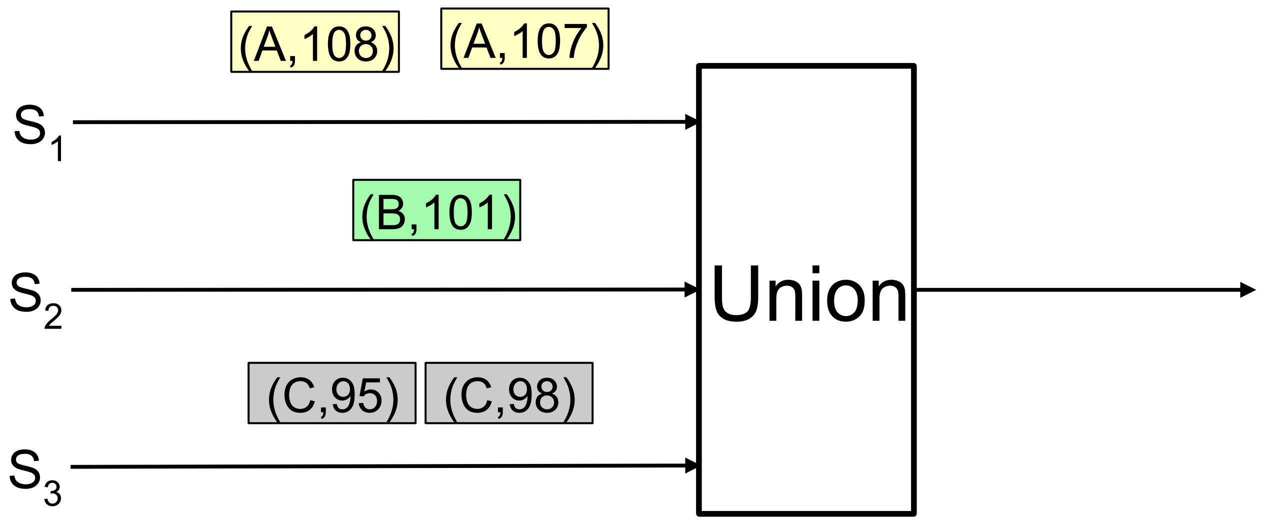
new.location = old.location
new.temp_celcius = 5/9*(old.temp - 32)

Map Example

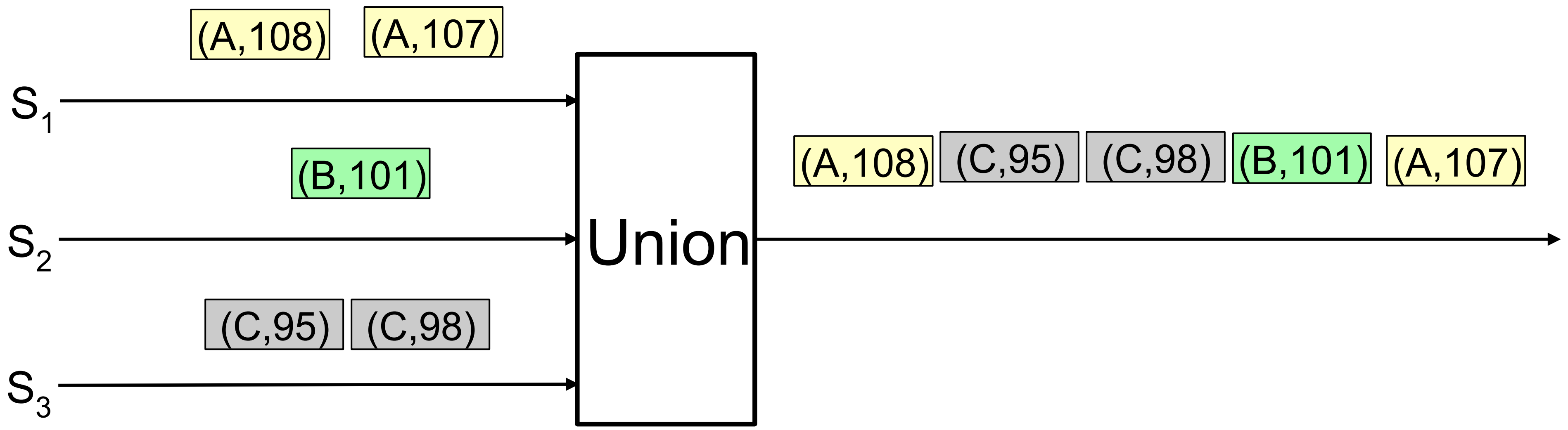


new.location = old.location
new.temp_celcius = 5/9*(old.temp - 32)

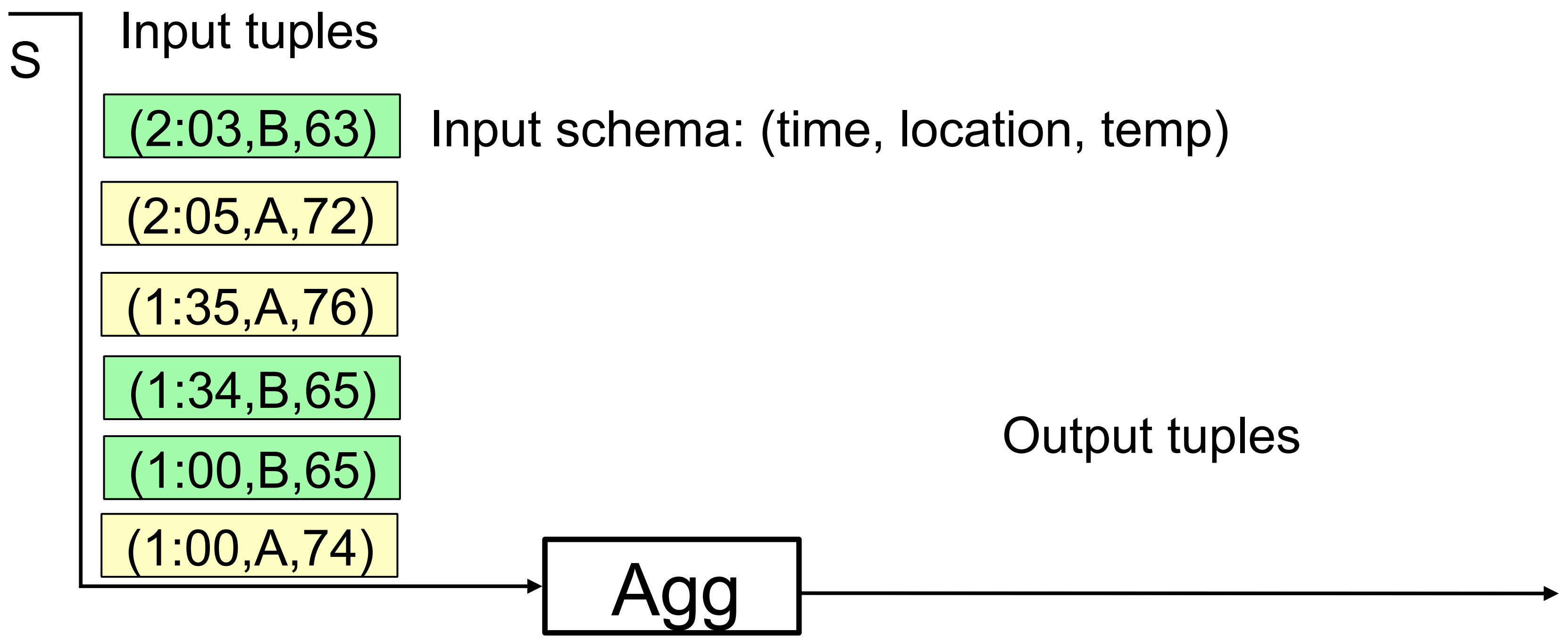
Union Example



Union Example



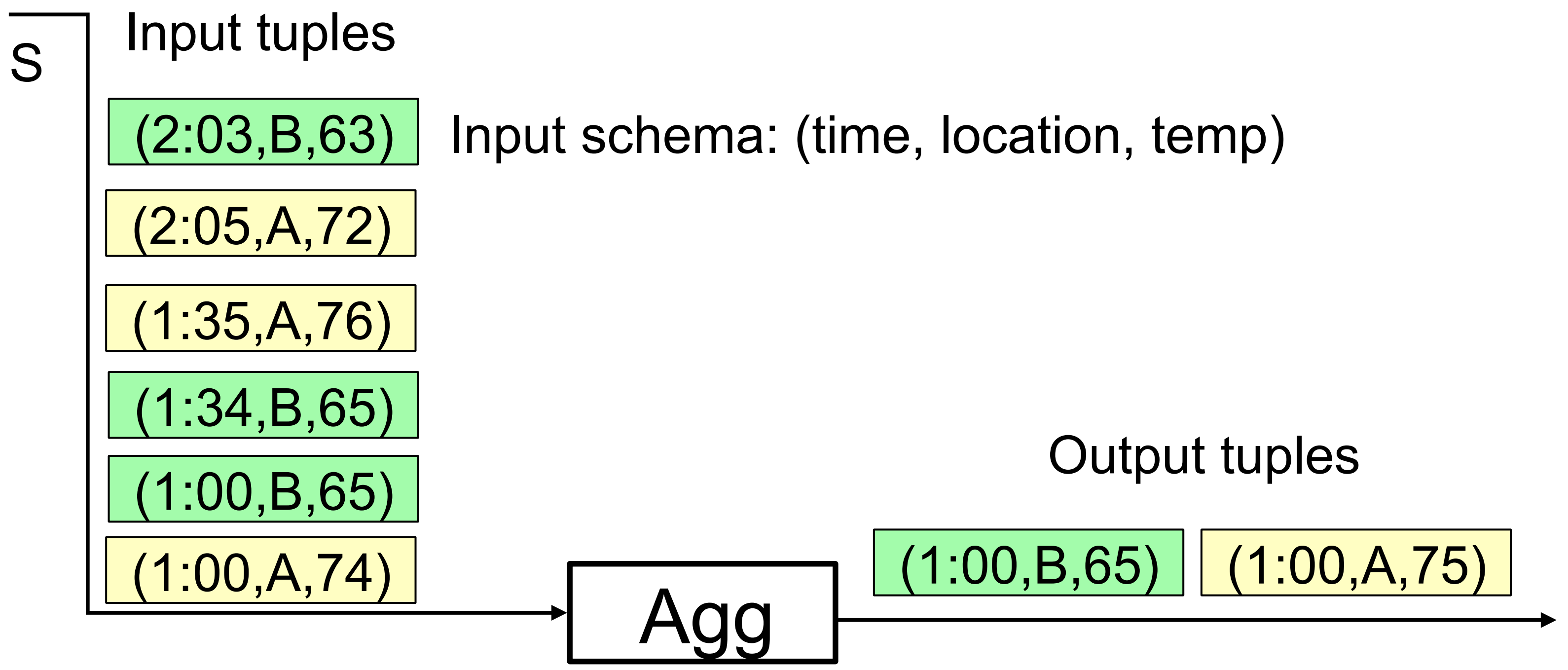
Aggregate Example



Operator parameters:

- group by location
- average temp,
- order on time, window size 1 h, advance 1h

Aggregate Example

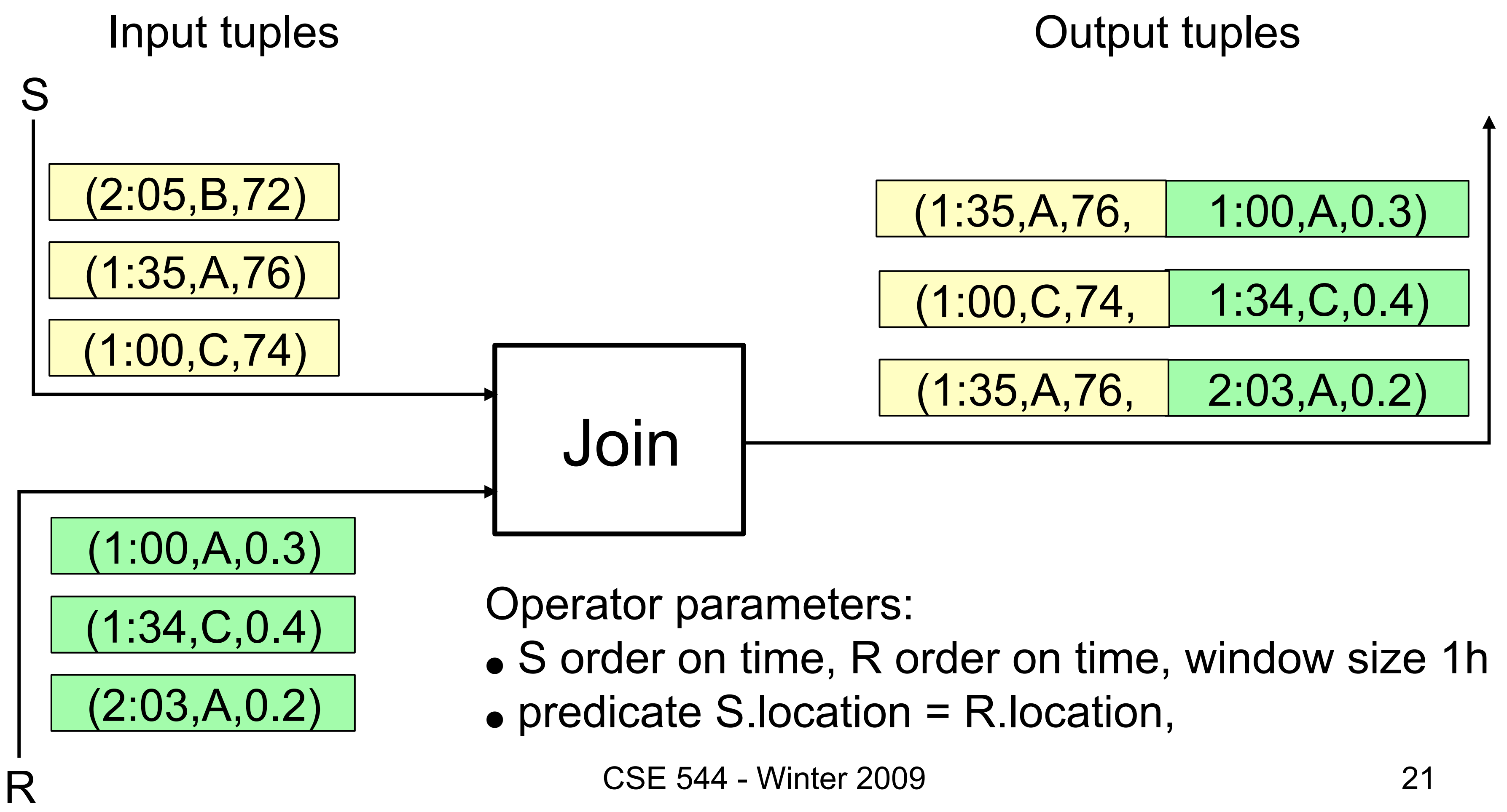


Input schema: (time, location, temp)

Operator parameters:

- group by location
- average temp,
- order on time, window size 1 h, advance 1h

Join Example



Operator parameters:

- S order on time, R order on time, window size 1h
- predicate S.location = R.location,

Sample Query

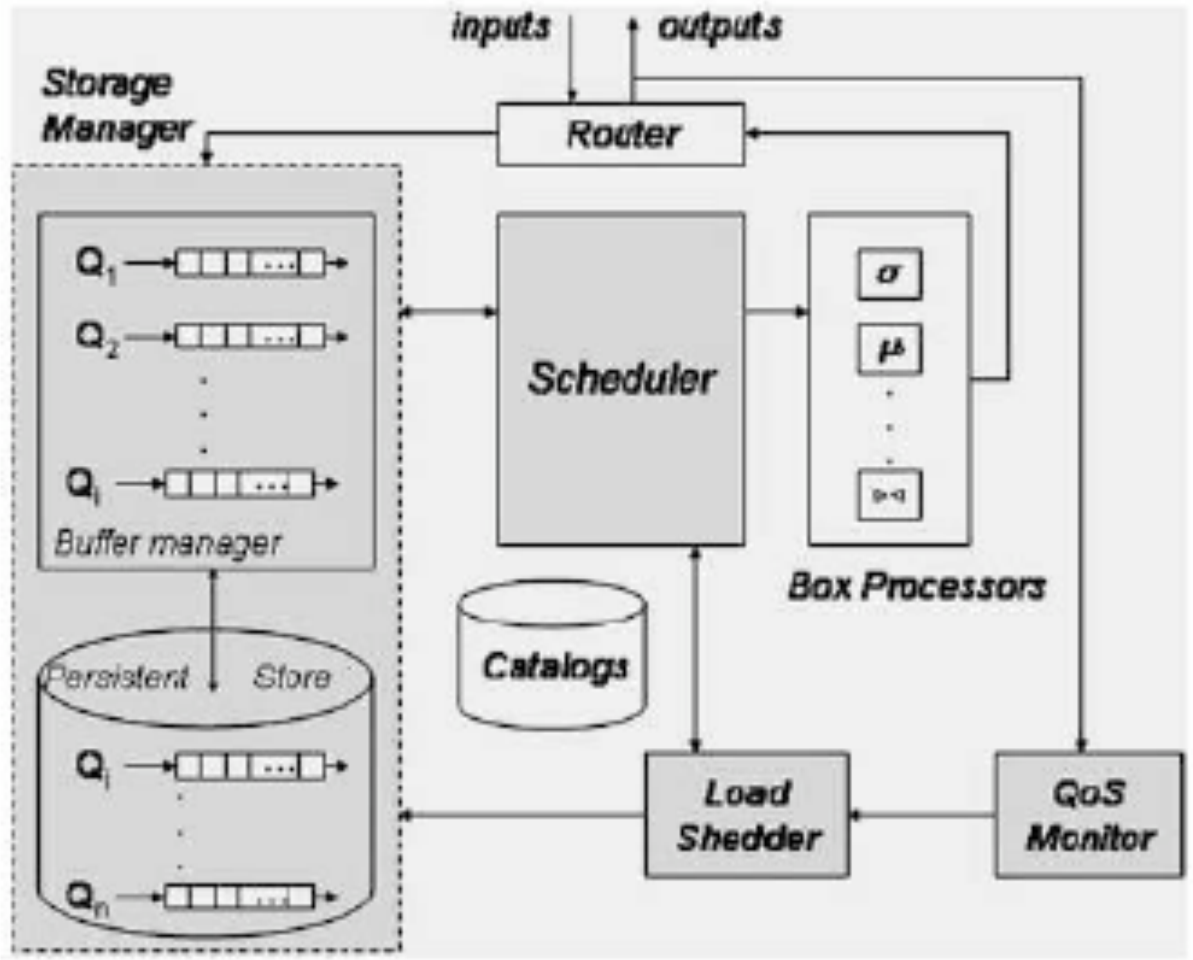
- Application: network intrusion detection

- Schema of input stream

  `(src_ip,src_port,dst_ip,dst_port,time)`

- **Query**
  - Alert me if an IP address establishes more than 100 connections per minute
  - and within 30 seconds of that event
  - the IP tries to connect to more than 10 distinct ports within a minute

Processing Model



[Figure 3 from Abadi:03]

Additional Features

- **Load management**
  - What happens when system is overloaded?
- **Fault-tolerance**
  - What happens if a node fails?
  - What happens if the network fails?
  - What happens if input data is wrong?
- **Exploiting data archives**
  - Historical queries, ad-hoc queries
  - Integrating push-based processing with pull-based
- …