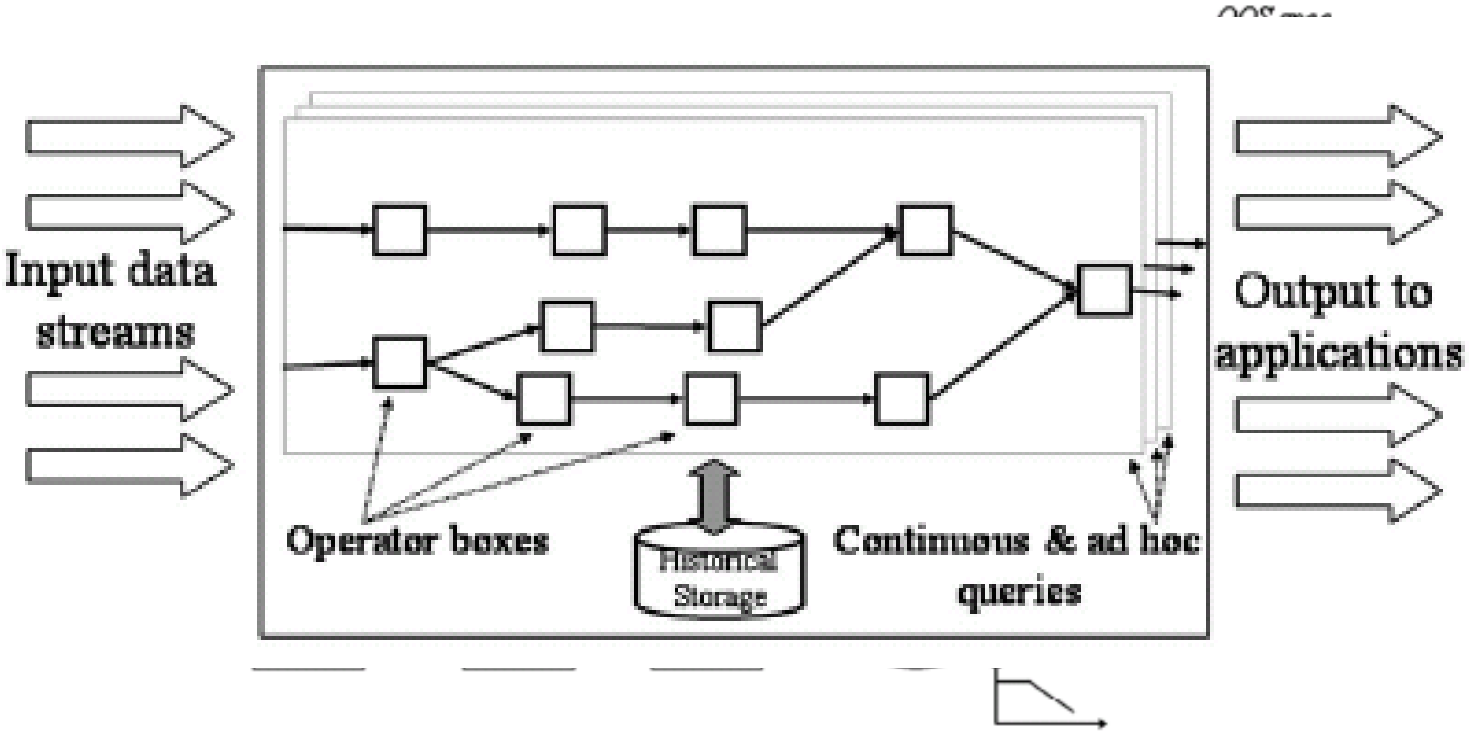
Data Stream Processing

Mengmeng Liu
Svilen Mihaylov

Aurora: Centralized Stream Processing Engine

- Why not Database Management System (DBMS)?
  - Data sources: external sources (e.g. sensors) vs. human issuing transactions
  - Data support: history data vs. current data
  - Query precision: approximate vs. accurate
  - Query response: real-time vs. offline
  - Others: large scale of triggers vs. few triggers

Aurora: System Model



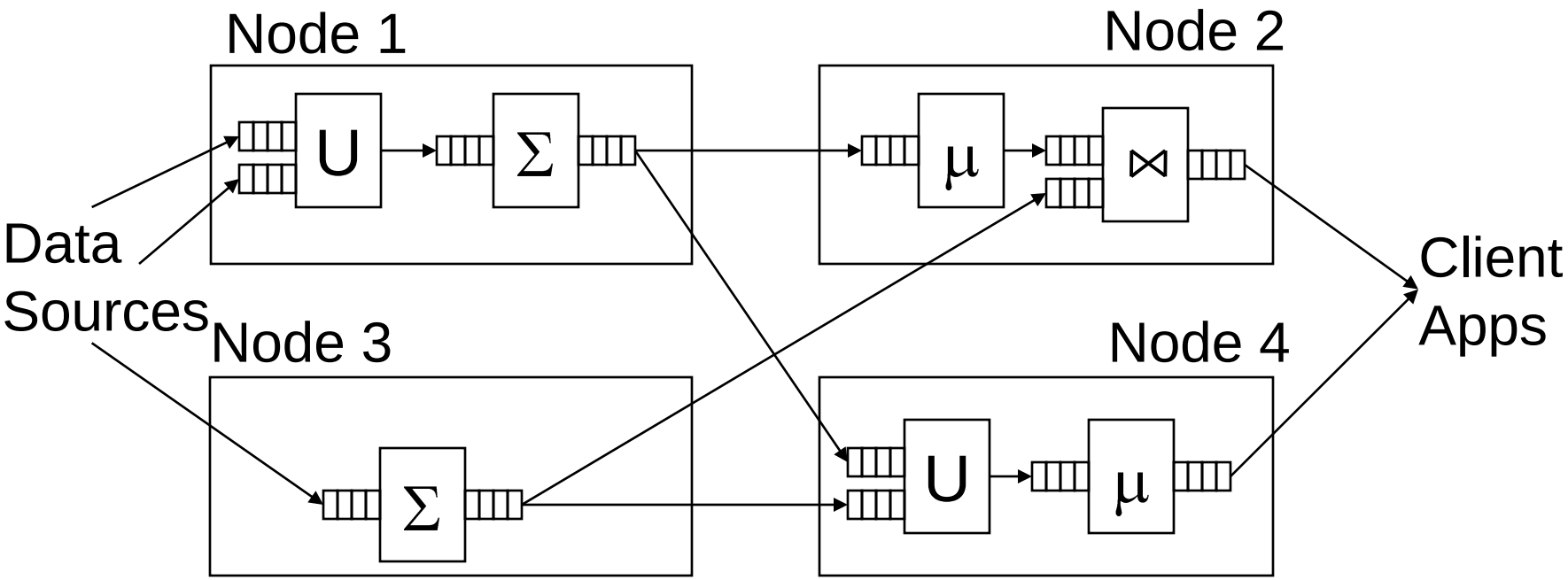
Aurora: Load Shedding

- Load shedding by dropping tuples
  - Improves overall performance
  - Drop randomly selected tuples
  - Delay-based
- Load shedding by filtering tuples
  - Identifies semantic importance
  - Converts lowest utility interval to filters
  - Value-based

Aurora: Lessons Learned

- Support for historical data
  - Historical queries
  - Hybrid queries involving both streaming and historical data
  - Aggregate summaries
- Resilience to unpredictable stream
  - No assumptions that a data stream arrive in any particular order
  - The arrival rate of streams might be irregular
- Tolerate imprecision

Borealis: Distributed Stream Processing Engine



- Data sources push tuples continuously
- Operators process windows of tuples

Borealis: generalized data model

header data

(time,**type**,**id**,a1,...,an)

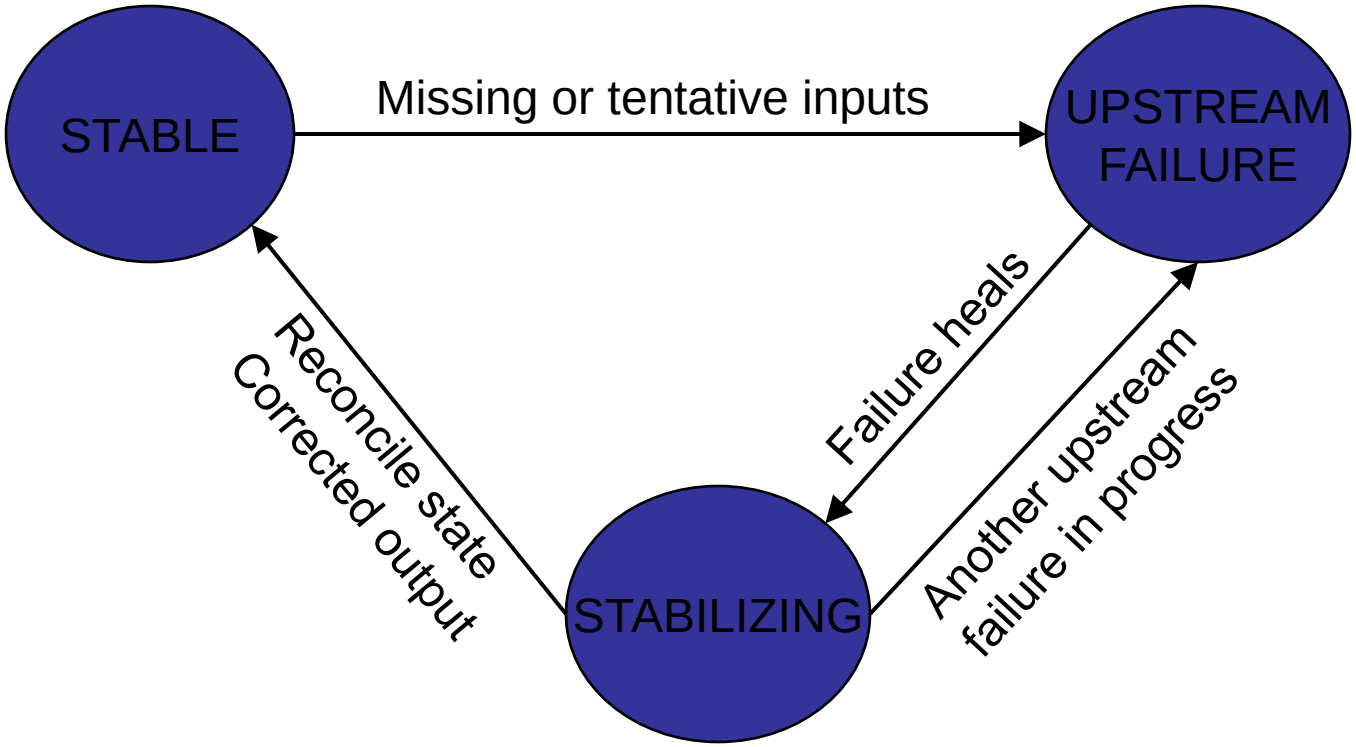
- time: tuple timestamp
- **type**: revision type
  - insertion, deletion, replacement
- **id**: unique identifier of tuple on stream

Borealis: System Features

- Three fundamental requirements
  - Dynamic revision of query results
    - Revision on input streams
    - Fault-tolerance
  - Dynamic query modification
    - Control lines
    - Time travel
  - Dynamic and distributed query optimization

Borealis: Fault-Tolerance Approach

- If an input stream fails, find another replica
- No replica available, produce tentative tuples
- Correct tentative results after failures



Borealis: Optimized Resource Allocation

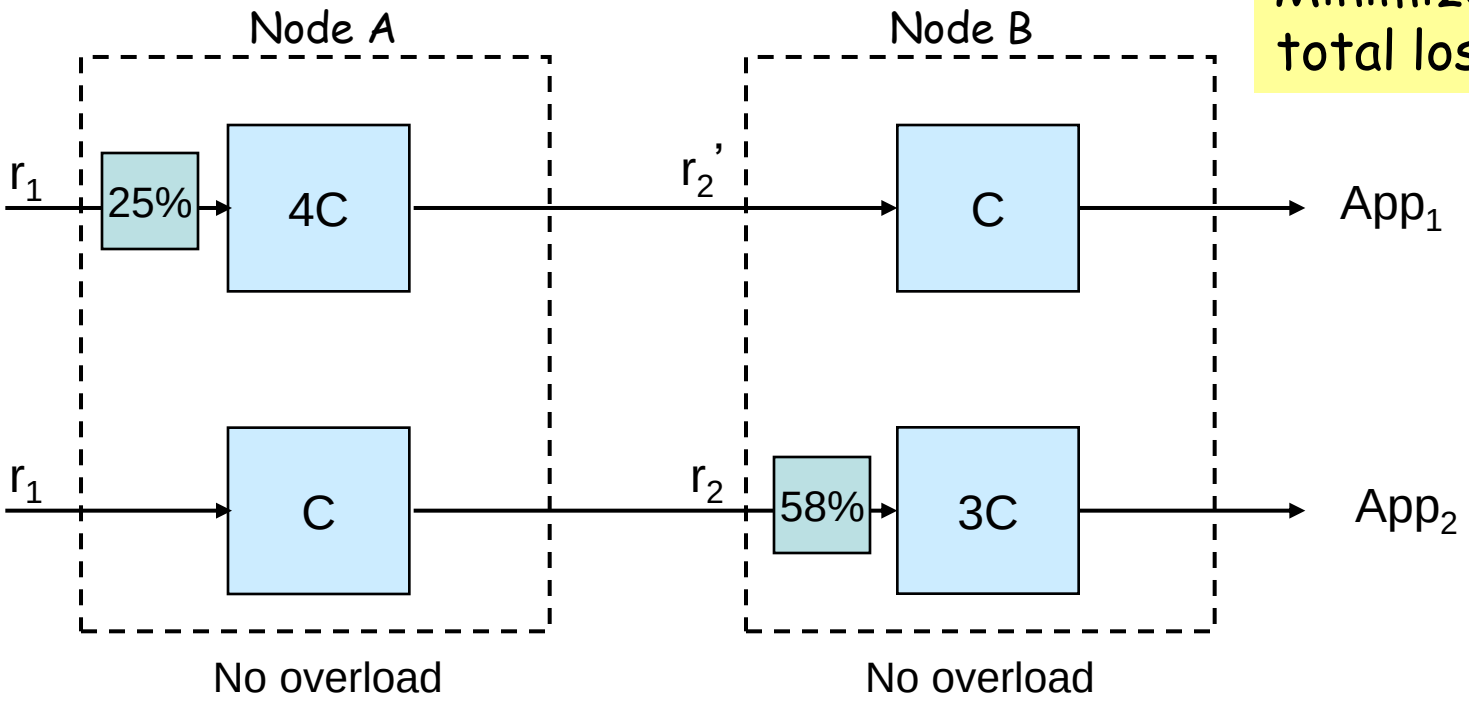
- Challenges:
  - Wide variation in resources
    - High-end servers vs. tiny sensors
  - Multiple resources involved
    - CPU, memory, I/O, bandwidth, power
  - Dynamic environment
    - Changing input load and resource availability
  - Scalability
    - Query network size, number of nodes

Borealis: Load Shedding

- Goal: Remove excess load at all nodes and links
- Shedding at node A relieves its descendants
- Distributed load shedding
  - Neighbors exchange load statistics
  - Parent nodes shed load on behalf of children
  - Uniform treatment of CPU and bandwidth problem
- Load balancing or Load shedding?

Local Load Shedding

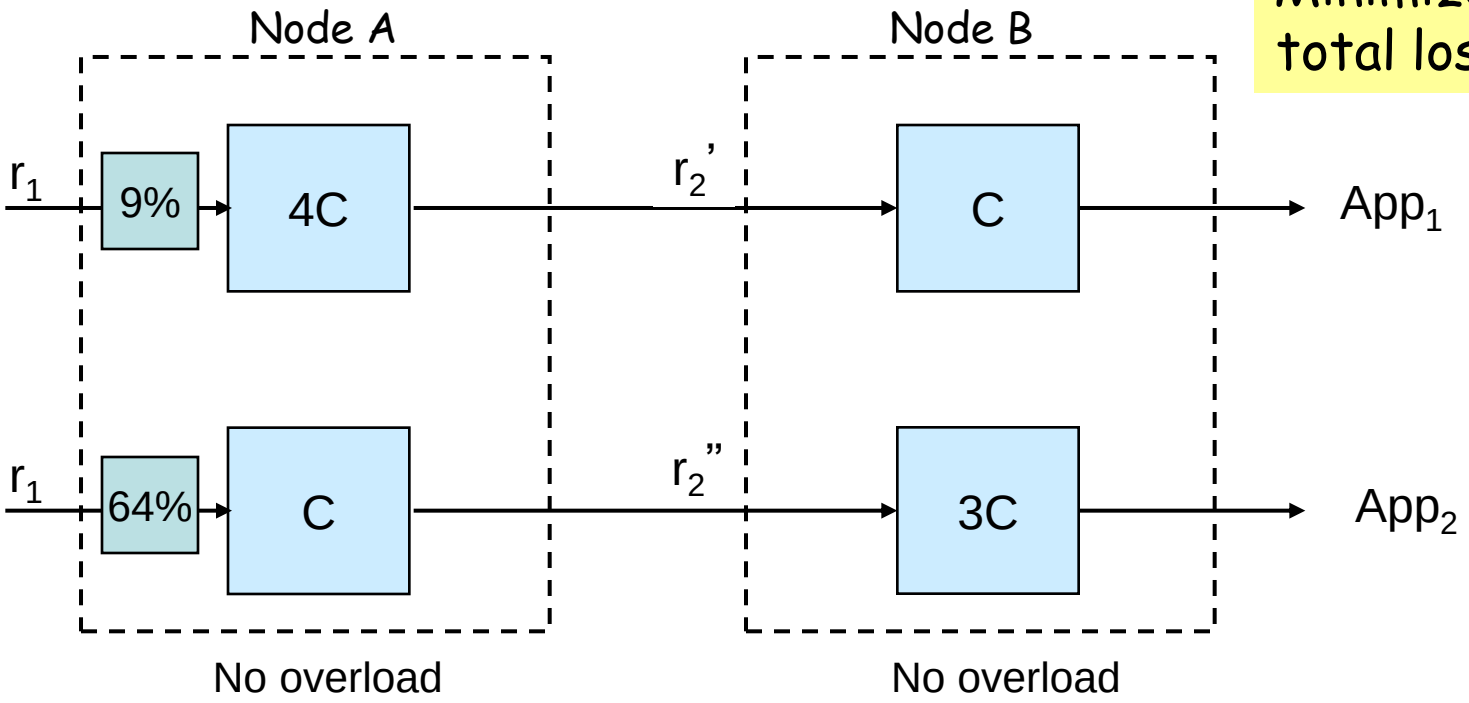
Goal: Minimize total loss



Plan	*Loss*
Local	App_1: 25% App_2: 58%

Distributed Load Shedding

Goal: Minimize total loss



Plan	*Loss*
Local	App_1: 25% App_2: 58%
Distributed	App_1: 9% App_2: 64%

← smaller total loss!

Conclusions

- Main points here
- References:
  - http://www.cs.brown.edu/research/borealis
  - http://www.cs.brown.edu/research/aurora/
  - http://infolab.stanford.edu/stream/

Discussion