Models and Issues in Data Stream Systems

Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, Jennifer Widom

Presented by Christian Valdemar Mathiesen
cmath@cs.brown.edu
March 9, 2015
STREAM*

*STanford StREam DatA Manager
STREAM

• Query language
• Query processing
• Conclusion
“In the STREAM project, we have chosen to use a modified version of SQL as the query interface to the system […]. SQL is a well-known language with a large user population.”
```
SELECT AVG(V.minutes) 
FROM (SELECT S.minutes 
FROM Calls S, Customers T 
WHERE S.customer_id = T.customer_id 
AND T.tier = 'Gold') 
V [ROWS 1000 PRECEDING]
```

VS.

**Figure 1: Tweet word count topology**

Source: “Storm @Twitter”, Toshniwal et al.
Which is easier to understand?

STREAM

Aurora


** Source: The Aurora and Borealis Stream Processing Engines, Cetintemel et al.
“Formally we say that a data stream consists of a set of (tuple, timestamp) pairs[...] — all that is required is that [the timestamp] comes from a totally ordered domain with a distance metric.”
What if tuples arrive from multiple sources?

In other words, how do we guarantee a totally ordered domain?
Query processing

Paper uses same notation for queries and queues!? 

\[ Q_1: \text{SELECT notifyoperator(sum(len)) FROM } B \text{ GROUP BY getminute(time) HAVING sum(len) > t} \]
Query processing

How are query plans generated?
How does the system scale (i.e. it only has one central scheduler)?
Conclusion

- Paper presents a series of relevant issues for OLTP systems
- STREAM tries to solve these issues, but reasoning behind design decisions are sometimes unclear
- Algorithmic issues should be put in separate paper