Cayuga: An Event Monitoring System

Data Centric Networking: Open Source Project Study
Cayuga: Introduction

- Real-time processing of event streams, handles *stateful* subscriptions that involve more than a single event.

  "For a company trade’s volume > 10,000, notify me when the stock price is monotonically decreasing for at least 10 minutes, and after the decrease, the stock rebounds at least 5%"

- Cayuga Event Language
  - Uses operators with formal semantics: Filter, project, join (correlate), aggregate events from multiple streams:
    
    ```sql
    SELECT Name, MaxPrice, MinPrice, Price AS FinalPrice
    FROM FILTER{DUR > 10min}(
      (SELECT Name, Price_1 AS MaxPrice, Price AS MinPrice
       FROM FILTER{Volume > 10000}(Stock)
    NEXT{$2.Name = $1.Name AND $2.Price > 1.05*$1.MinPrice}
    Stock)
    ```

- Applications: automated stock analysis, RSS feed monitoring
- Distinguishing feature: Scalability via effective multi-query optimisation
  - Throughput of tens of thousands of events per second for hundreds of thousands of active queries

Custom Tweet Notifications

“Notify me when there is a tweet on CNN about US troop morale with positive sentiment and this is followed by an article on BBC about US troop morale with negative sentiment”

“Notify me when there are more than 10 tweets on the launching of iPhone 5 within the last 10 minutes”

“Notify me when there are more than 5 distinct persons who re-tweeted my tweets related to the jobs that I offered”

Real-time / offline processing

Performance evaluation: Event complexity vs. Speed vs. Scalability
I’ve tried ...

- Setup the system (Visual Studio, C++, Java)
- Explored examples
- Generated tweets for offline processing
- Testing new queries & tried the visualiser

**Fig. Cayuga Automata Visualiser**
Challenges

- Collecting a large set of tweets
  - Max 3200 tweets per export
  - Continuous query simulation can last at least 53 minutes
- Pre-processing the stream data into a format that Cayuga can understand
- Predicting query behavior
  - Complex query logic and uncertainty in input stream(s) make it difficult to predict outcome