Temporal Streams
Programming Abstractions for Distributed Live Stream Analysis Applications

David Hilley
davidhi@cc.gatech.edu
Advisor: Umakishore Ramachandran
School of Computer Science
College of Computing, Georgia Institute of Technology
What is Live Stream Analysis?

• Live Stream Analysis
  • Surveillance / “Situational Awareness”
  • Traffic Analysis
  • Cargo / Asset Tracking
  • Robotics
  • Disaster Response

• Ubiquitous and increasingly important
What is Temporal Streams?

- Building blocks for stream analysis
- Distributed data structures for streams
- “Glue” for communicating components
- A lower-level substrate
Our Own Experience

- TV Watcher
- Media Broker / MB++
- Streaming Grid
- \( V(A)aaS \) – video-analytics-as-a-service
- \( RF^2ID \)
- ASAP – situational awareness
- IPTV Analytics / Recommender systems
Pain Points

- Time – synchronization, data retrieval
- Scalable data delivery
- Storage of streaming data
- Management of computation? – yes, but vastly different requirements between applications and domains
Solution Space

• MPI, Message-Oriented-Middleware (too low level)

• Stream Data Management Systems

• Event Stream Processing (ESP) / CEP (too high level)

• Temporal Streams (just right)
Programming Model
Channel

- Represents a continuous data stream
- Items ordered by wall-clock timestamp
Channel

- Represents a continuous data stream
- Items ordered by wall-clock timestamp
- Simple time-based operations:
  - `put(item, [timestamp])`
  - `get(lower_bound, upper_bound)`
- **Time variables** to specify time intervals
  - e.g. `now`, `newest-after(ts)`, `oldest`, etc.
- Spans communication & storage
Channel Get Interval Example

Returned Items

oldest

lower bound

upper bound

now
The Premise

- Not just about “plumbing” / transport
- More explicit support for writing stream manipulation code via the data abstractions
- Wall-clock time as a recognized entity
- Time as an indexing mechanism naturally admits synchronization, data persistence
Persistent Channel Get

Get Interval

... Stored Data

Live Data (60 sec.)

On Disk / Backing Store

In RAM
Stream Persistence

- Seamless persistence with same interface
- System automatically manages:
  - moving “live” items to backing store
  - retrieving stored items when necessary
- Control storage representation:
  - User-provided transformation
  - Automatic adaptation
Future

- $V(A)aaS$ – video-analytics-as-a-service
- Live stream analysis in the cloud
That’s all folks

- Questions?