DS256:Jan17 (3:1)

L9,10:Distributed Stream Processing

Yogesh Simmhan

14 Feb, 2017

Stream are Commonplace (too)

- Web & Social Networks
  - Twitter, Facebook, Internet packets
- Cybersecurity
  - Telecom call logs, financial transactions, Malware
- Internet of Things
  - Smart Transport/Power/Water networks
  - Smart watch/phone/TV/…

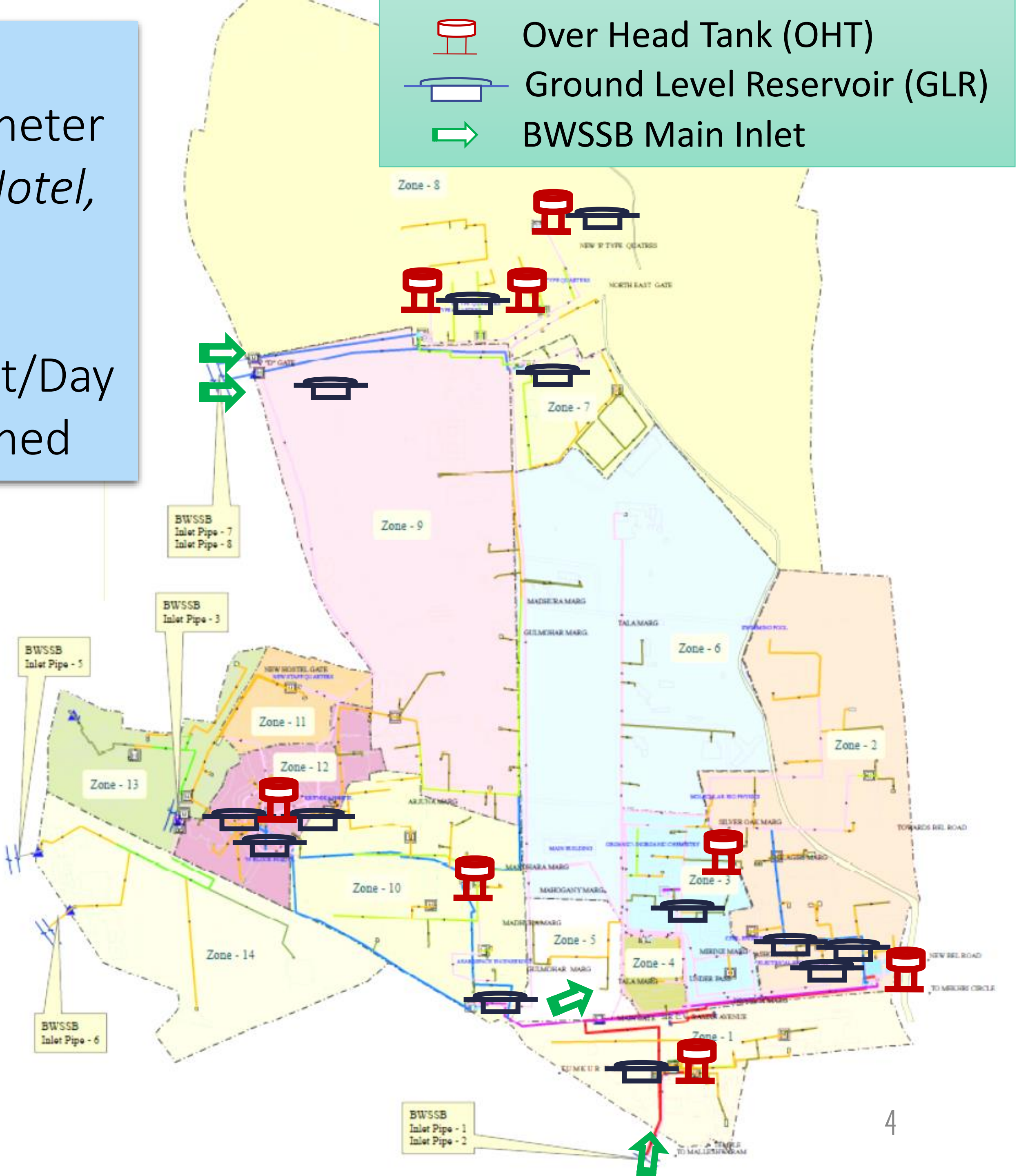
IISc Smart Campus: Water Management

- Plan pumping operations for reliability
  - Avoid underflow/overflow of water
  - 12 hrs to fill a large OHT, scarcity in summer weeks
- Provide safer water
  - Leakages, contamination from decades old network
- Reduce water usage for sustainability
  - IISc average: **400 Lit/day**, Global standard: **135 Lit/day**
  - Lack of visibility on usage footprint, sources
  - Opportunities for water harvesting, recycling
- Lower the cost
  - Reduce cost for water use & electricity for pumping

IISc Campus

- 440 Acres, 8 Km Perimeter
- 50 buildings: *Office, Hotel, Residence, Stores*
- 10,000 people
- Water Use: 40 Lakh Lit/Day
- 10MW Power Consumed

OHT	8
GLR	13
Inlet	4+3



Over Head Tank (OHT)

Ground Level Reservoir (GLR)

⇨ BWSSB Main Inlet

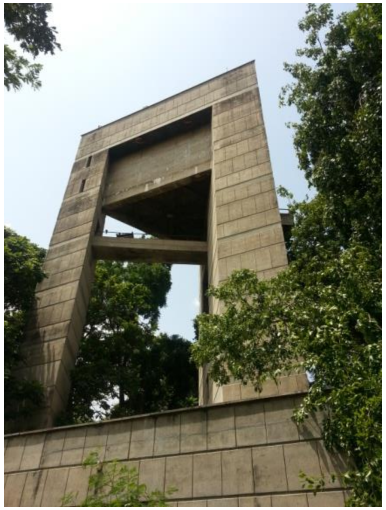
Over Head Tanks (OHT)

TPH (near Mechanical)

JNT Auditorium

Chemical Stores

Opposite to CENSE



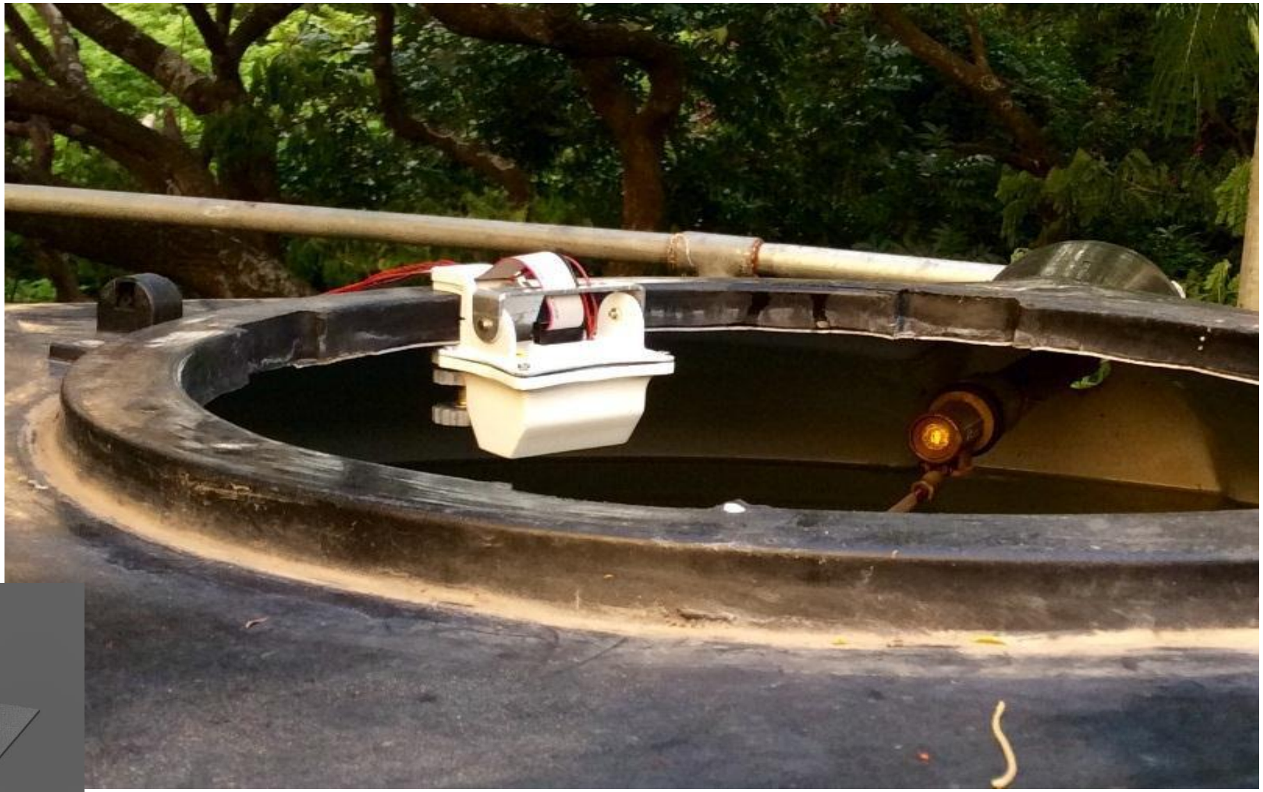
Opposite to NESARA

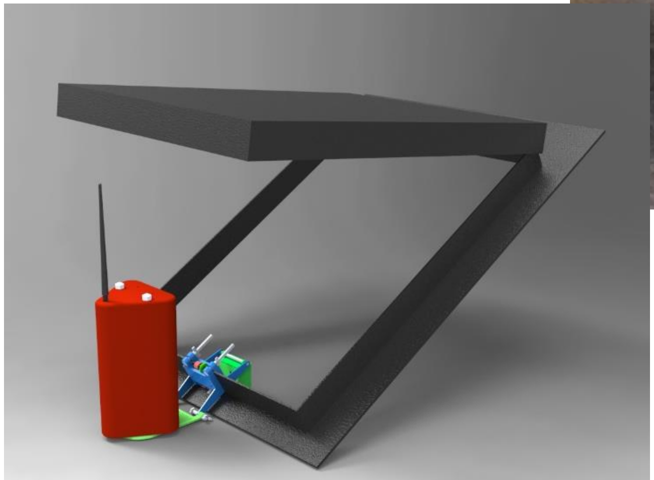
Behind old C-Mess

Opposite to Cense (new)

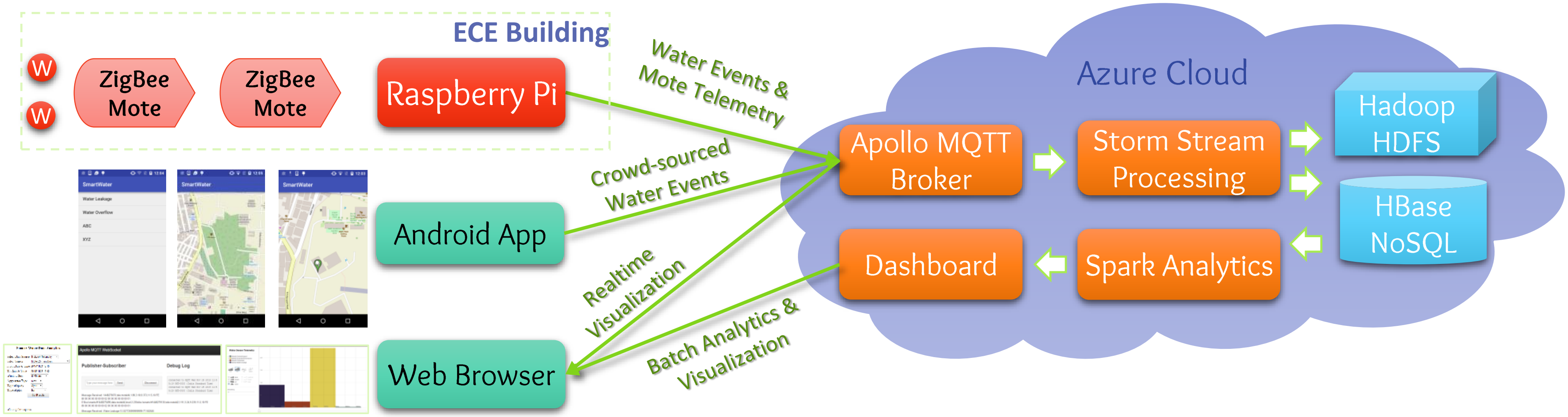
E Type Quarters

Custom Level + Quality Sensor





Backend



ECE Building

W

W

ZigBee Mote

ZigBee Mote

Raspberry Pi

Water Events & Mote Telemetry

Android App

Crowd-sourced Water Events

Web Browser

Realtime Visualization

Batch Analytics & Visualization

Azure Cloud

Apollo MQTT Broker

Storm Stream Processing

Hadoop HDFS

HBase NoSQL

Dashboard

Spark Analytics

Aadhaar Enrolment

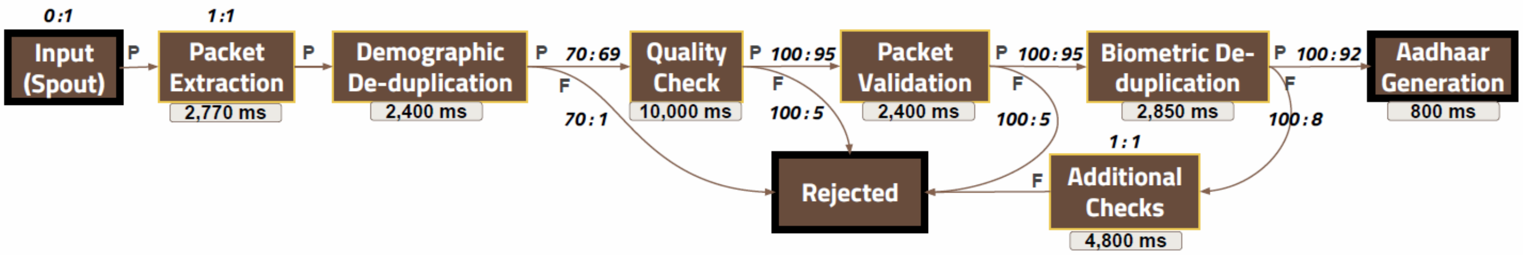


Fig. 1. *Enrollment dataflow.* Tasks are labeled with the average latency time in milliseconds. The selectivity is given for each outgoing edge. "P" edges are taken by events that pass the check at a task, while "F" edges are taken by events that fail a check.

- Input is stream of identity enrolment packets
- Output is a *UIDAI ID (success)* or *rejection*
- Each task tagged with Latency (ms)
- Each edge tagged with Selectivity
  - Input:output rate, probability of path taken

Benchmarking Fast Data Platforms for the 'Aadhaar' Biometric Database, Shukla, et al, Workshop on Big Data Benchmarking (WBDB), 2015

Aadhaar Authentication

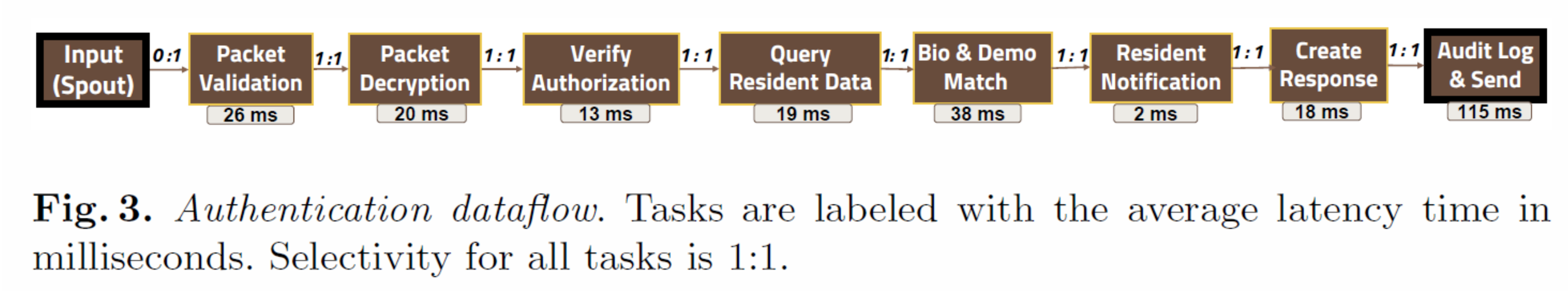
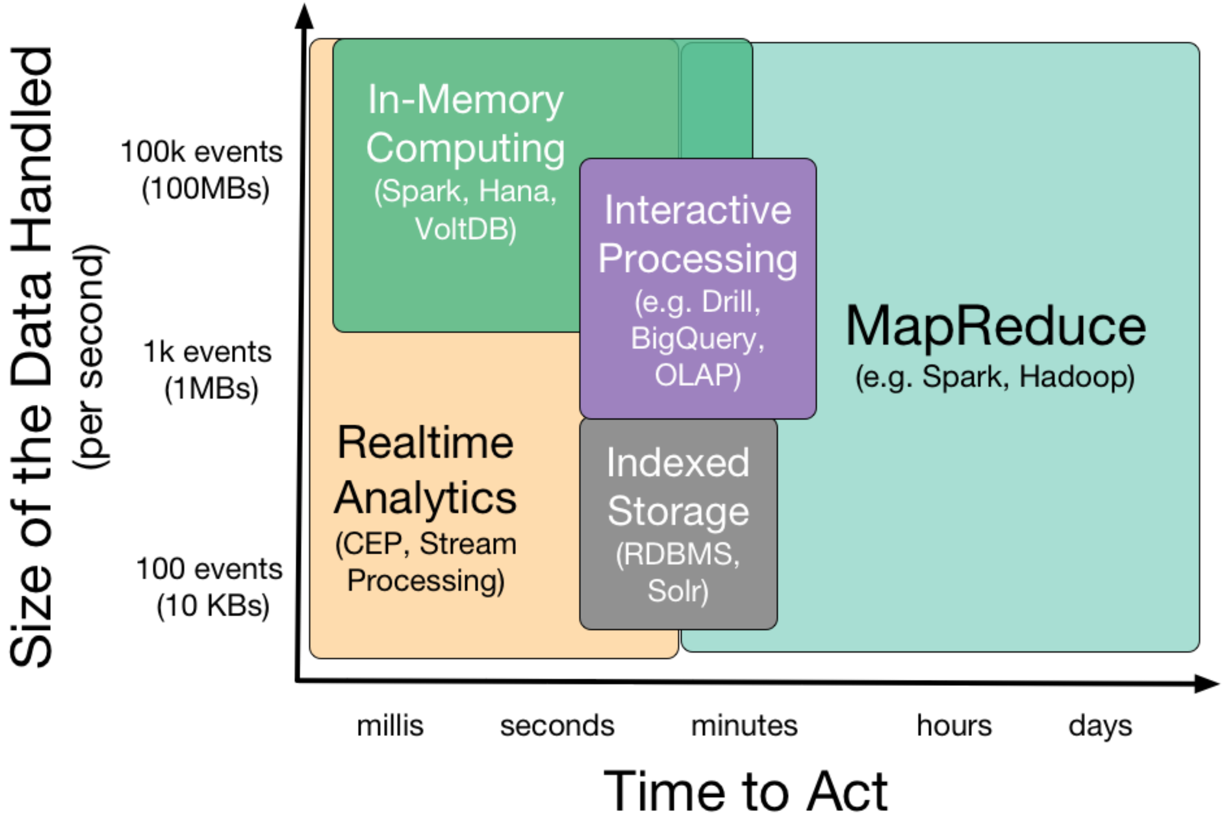


Fig. 3. *Authentication dataflow.* Tasks are labeled with the average latency time in milliseconds. Selectivity for all tasks is 1:1.

Benchmarking Fast Data Platforms for the 'Aadhaar' Biometric Database, Shukla, et al, Workshop on Big Data Benchmarking (WBDB), 2015

Fast Data Processing

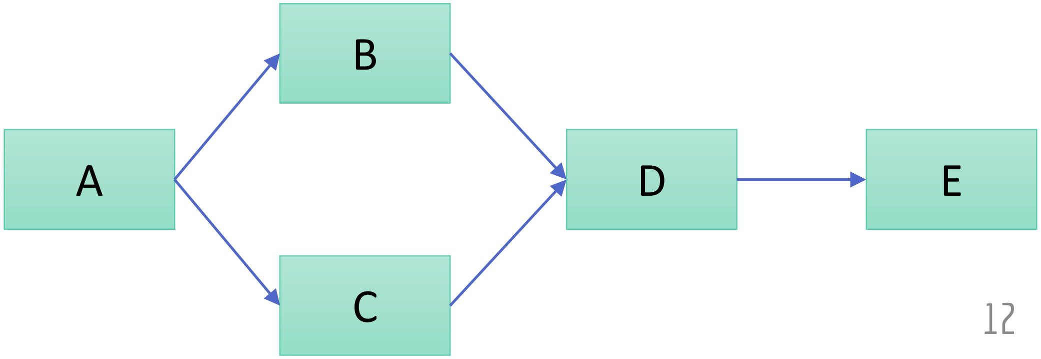
Size vs. Latency



Big Data Analytics Platforms for Real-time Applications in IoT, Yogesh Simmhan & Srinath Perera, Big Data Analytics: Methods and Applications, Eds. Saumyadipta Pyne, B.L.S. Prakasa Rao, S.B. Rao, 2016

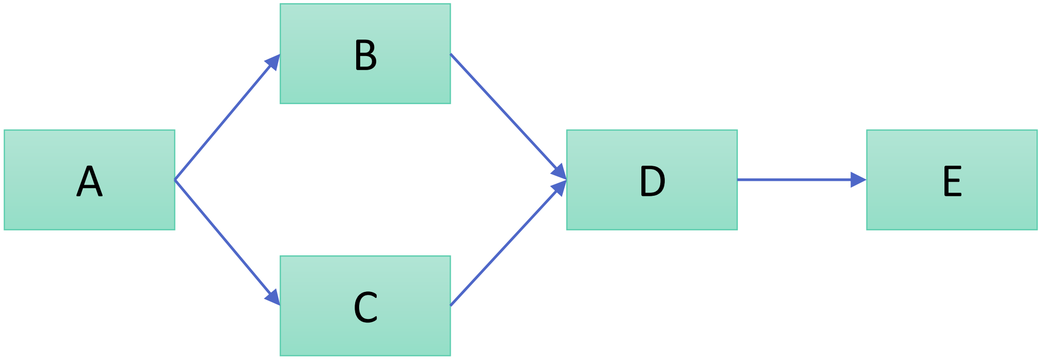
Application Model

- Application composed as a Directed Acyclic Graph (DAG)
- Tasks are user-defined logic, vertices of the DAG
- Streams are channels between Tasks, edges of DAG
- Streams carry “infinite” number of tuples
  - Also called events, messages
- Tuples may be opaque, or have Name/Value/Type



Application Model

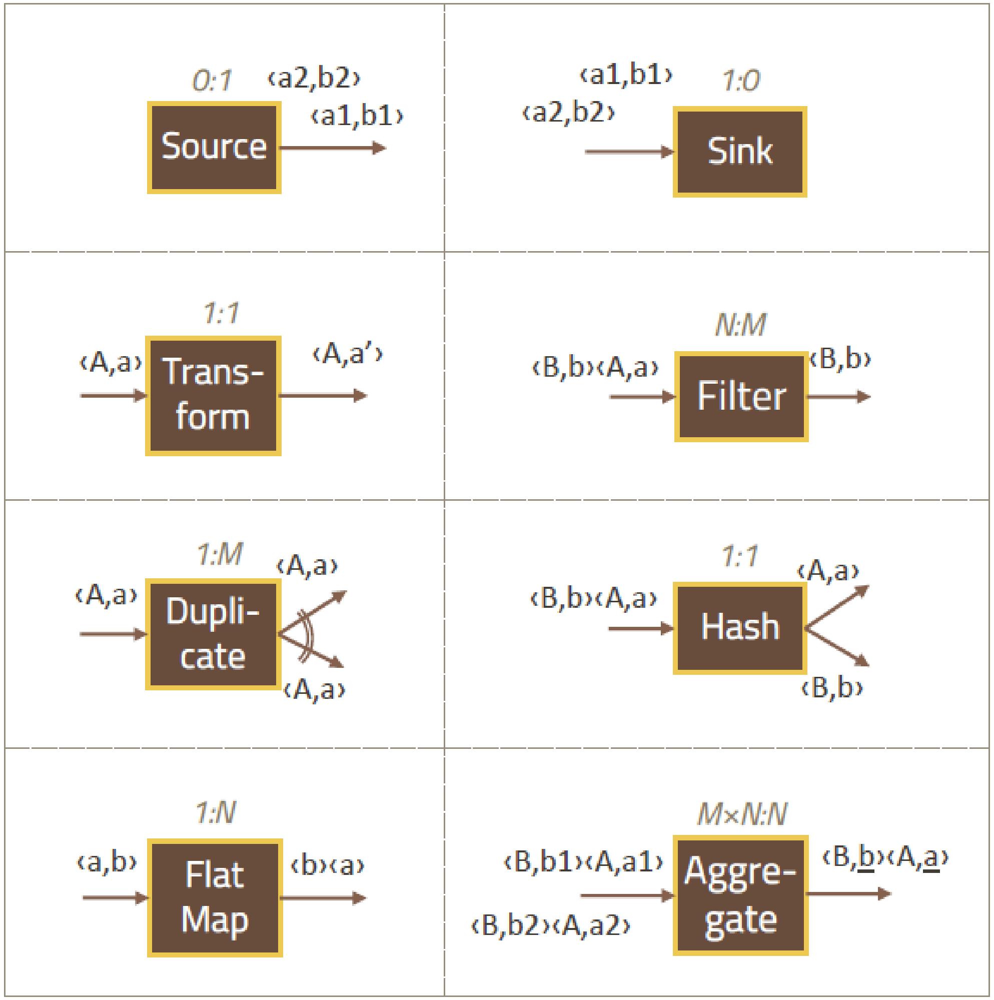
- Tasks executed *once* for each input tuple
  - Tasks may emit *zero or more tuples* for each input
- Latency: Time taken to process a single tuple by a task.
- Selectivity: Ratio of average number of output tuples expected for each input tuple ($\boldsymbol{in{:}out}$ or $\boldsymbol{\frac{out}{in}}$)
- Can be used to calculate input rate at each task, given DAG input rate
  - e.g. i_d = i*s_a*s_b + i*s_a*s_c



Application Model

- Different degrees of parallelism helps exploit multiple resources to complete the execution faster, reduce latency
- Task parallelism due to multiple tasks composed and executing in parallel (B&C || D)
  - Two tuples can be concurrently executed on two different tasks that are independent of each other
  - Different from data parallelism (later...)
- Pipelining due to streaming execution
  - Different parts of the infinite stream can be executing at the same time on different tasks
  - All tasks can execute at the same time, once pipeline filled
- Orthogonal concepts, can have one without other

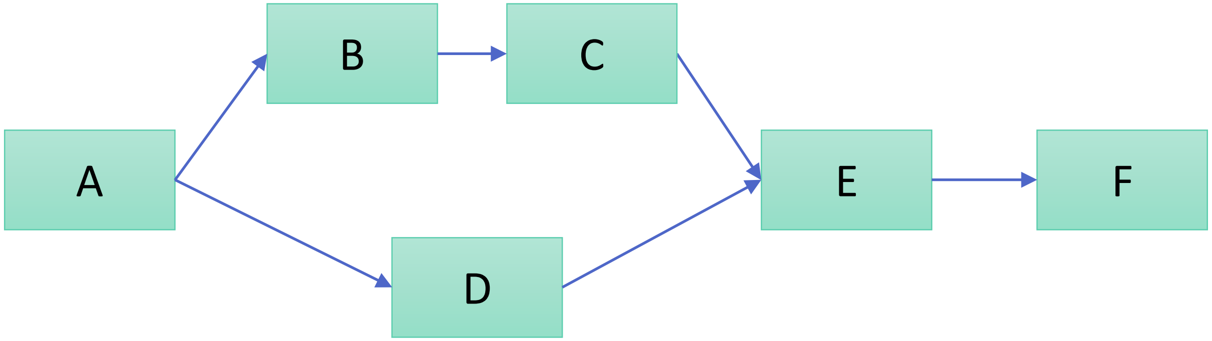
Common Task Patterns



Benchmarking Distributed Stream Processing Platforms for IoT Applications, Anshu Shukla and Yogesh Simmhan, TPCTC, 2016

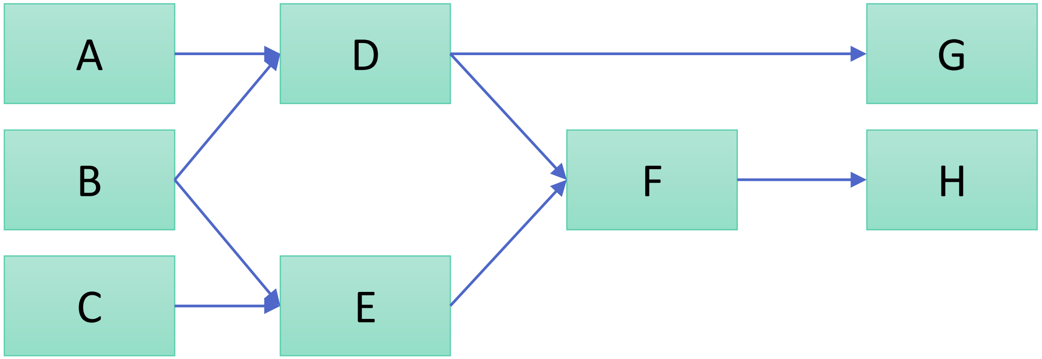
Routing Semantics

- Multiple outgoing edges
  - Duplicate, Round-robin or Hash
- Multiple input edges
  - Interleave: Does a union of tuples entering an input queue. Number of tuples is the sum of number of tuples from each input stream.
  - Join: Merges one tuple from each input stream into a single tuple, that is given to the task. Number of tuples is the minimum of all tuples that enter on any input stream.



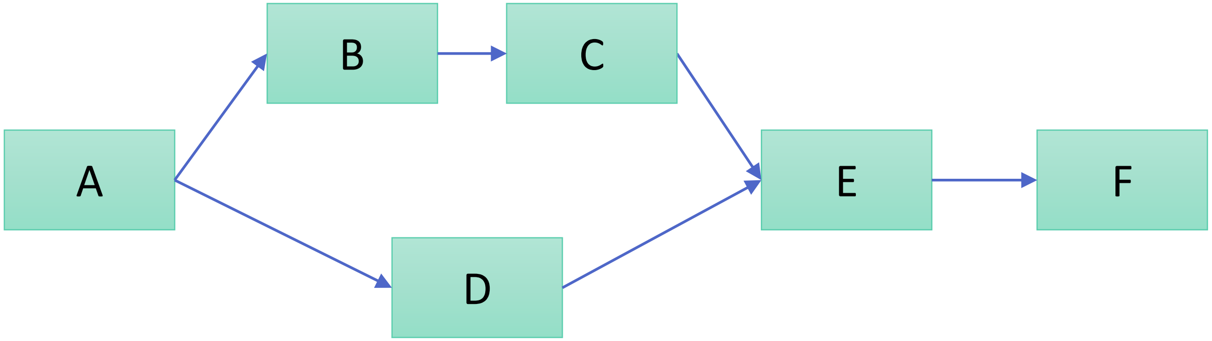
Application Model

- *Multiple* Source/Sink tasks can be present
- Count: Number of tasks in the DAG. Determines resource needs.
- Width: Widest number of parallel tasks. Task parallelism. (e.g. 3)
- Length: Longest number of tasks from a source to a sink (e.g. 4). Similar to Critical path…
- Average Edge Degree: Hotspots, affects selectivity



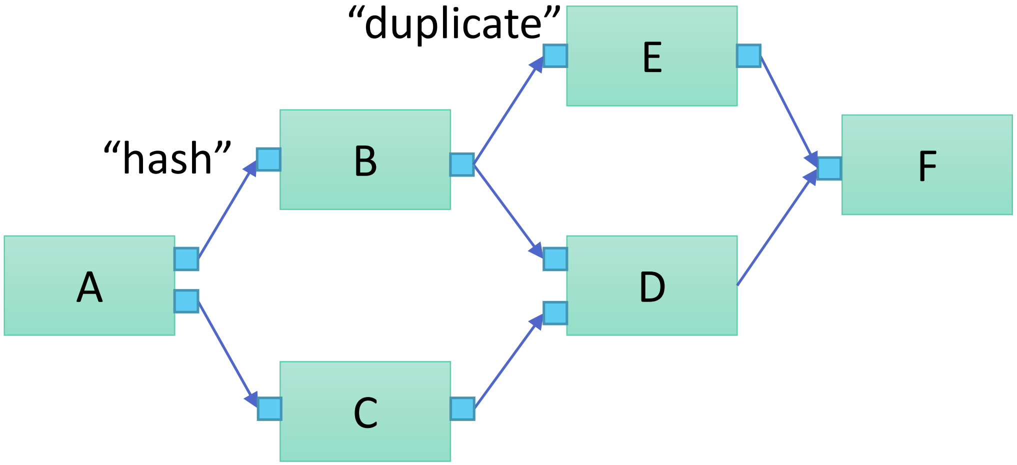
Application Model

- Causally Dependent Messages: Set of output tuple generated as a result of an input tuple.
  - What happens with aggregation? Sliding window?
- Critical Path: Longest latency from the source to the sink.
  - Determines the slowest causally dependent output, for a given input.
  - Includes task latency, I/O queue delay, and NW time



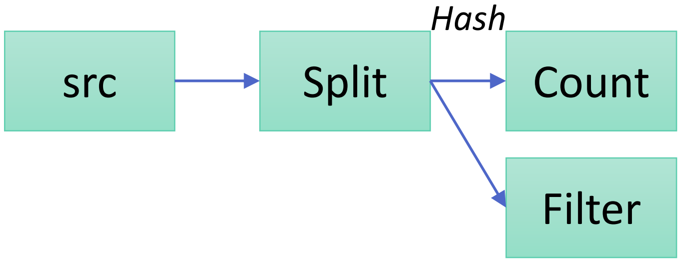
Application Model

- Tasks may have one or more input and output "ports"
- Makes the routing semantics explicit in the composition
    - *Join* between tuples on different ports
    - *Hash* by writing to explicit output port



Application Model

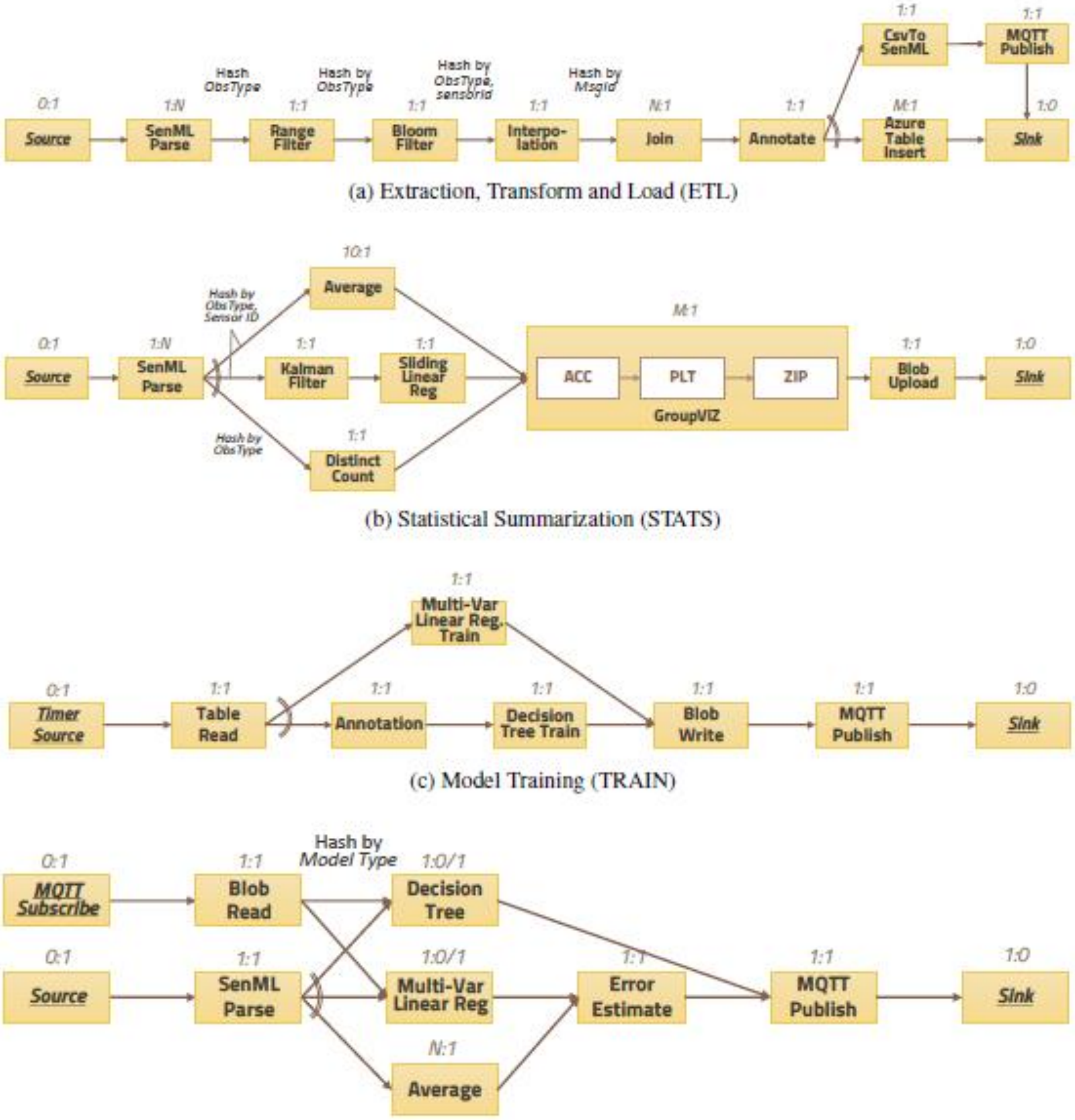
- Visual DAG composition
- Programming abstractions
  - Task centric view (Storm)
  - Data centric view (Spark)



```
val wordsDs = src.flatMap(value => value.split("\\s+"))
val wordsPairDs = wordsDs.groupByKey(value => value)
val wordCountDs = wordsPairDs.count()

filteredDS = wordsDs.filter(value => value =="hello")
```

http://blog.madhukaraphatak.com/introduction-to-spark-two-part-3/



(a) Extraction, Transform and Load (ETL)

(b) Statistical Summarization (STATS)

(c) Model Training (TRAIN)

(d) Predictive Analytics (PRED)

Figure 3. Application benchmarks composed using the micro-benchmark tasks.

Execution Model

- Input and output queue for each task
  - Buffers tuples
- Multiple Threads for same Task
  - Allows and controls data parallel execution
- Each thread can operate on one of the tuples in input queue
  - What happens to ordering?

Execution Model

- Tuple Ordering
- Can we guarantee that tuples are processing in specific order?
  - Difficult
  - Needs logical timestamps
  - Physical time-stamps vs. time skew
- Guarantee at the source vs. each task in the DAG

Execution Model

- Stateful vs. Stateless
- Do tasks retain state?
- Is state shared across threads?
- What is the impact on aggregation operations? Hash keys

Execution Model

- Delivery Guarantees
- Best effort
- At least once delivery
- Exactly once delivery
- Need to keep track of progress. Replay if necessary.

Distributed Stream Processing Systems

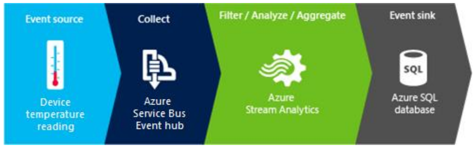
- Aurora – Early Research System
- Borealis – Early Research System
- **Apache Storm**
- Apache S4
- Apache Samza
- Google MillWheel
- Amazon Kinesis
- LinkedIn Databus
- Facebook Puma/Ptail/Scribe/ODS
- Azure Stream Analytics
- Apache Flink
- FlumeJava
- NiFi
- Google Dataflow
- Spark Streaming
- Apache Beam

Storm

S4 distributed stream computing platform

Amazon Kinesis

samza


Event source
Collect
Filter / Analyze / Aggregate
Event sink
Device temperature reading
Azure Service Bus Event hub
Azure Stream Analytics
SQL
Azure SQL database

Event Processing vs. Stream Processing

- Tuples are “transparent”
  - Columns, values
- Query Based
  - Complex Event processing
  - SQL like languages over continuous tuples
  - Tasks are operators with have specific semantic meaning
- Time operators included with
  - window, sequence, group, merge, trigger

The Dataflow Model, Akidau, et al., VLDB 2015

Reading

- Ankit Toshniwal, et al. Storm@twitter. In *ACM SIGMOD,* 2014
- Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters, Zaharia, et al, *USENIX HotCloud*, 2012, https://www.usenix.org/conference/hotcloud12/workshop-program/presentation/zaharia
- Leonardo Neumeyer, et al, S4: Distributed Stream Computing Platform. In *ICDMW* 2010