# Machine Learning on Apache Apex with Apache SAMOA

**Bhupesh Chawda**
Committer, Apache Apex,
DataTorrent Software
bhupesh@apache.org

**Nicolas Kourtellis**
Committer and PMC, Apache SAMOA,
Telefonica Research
nkourtellis@apache.org

## Agenda

- Introduction to Big Data, Stream Processing and Machine Learning
- Apache SAMOA and the Apex Runner
- Apache Apex and relevant concepts
- Challenges and Case Study
- Conclusion with Key Takeaways

## Big Data

## Introduction

- What is Big Data?
  - Search engine queries
  - Facebook posts
  - Emails
  - Tweets
  - etc.
- Volume, Variety, Velocity, Veracity
- Subjective?
- Beyond capability of typical commodity machines

## Stream Processing

## Distributed

- Why?
  - Real time, Low latency processing
  - Big Data, High speed of arrival
  - Potentially infinite sequence of data
- Each data item in the stream passes through a series of computation stages
- Helps in distributing the computation over multiple machines
- Typically, data goes to computation
- Batch - Special case of Streaming, snapshot over an interval of time

## Traditional Machine Learning

## Batch Oriented

- Supervised - most common
    - Training and Scoring
- One time model building
- Data sets
    - Training - Model Building
    - Holdout - Parameter tuning
    - Test - Accuracy of the model
- Training data has to be a representative data set
- Complex algorithms

Apache SAMOA
Scalable Advanced Massive Online Analysis

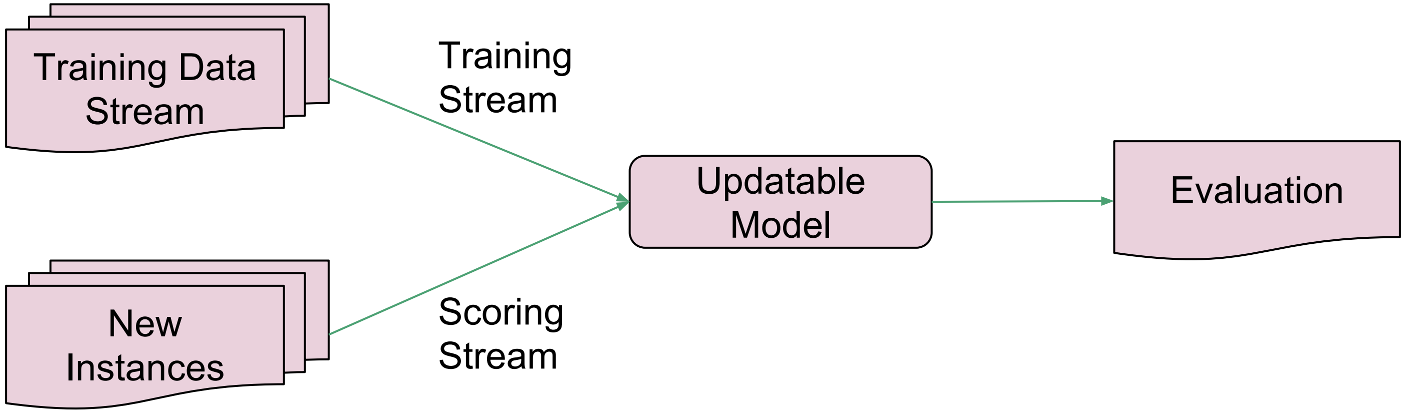
APEX

## Online Machine Learning?

## Streaming!

- Change!
  - Dynamically adapt to new patterns in data
  - Change over time (concept drift)
- Model updates
- Approximation algorithms
  - Single pass - one data item at a time
  - Sub-linear space and time per data item
  - Small error with high probability

## Online Machine Learning

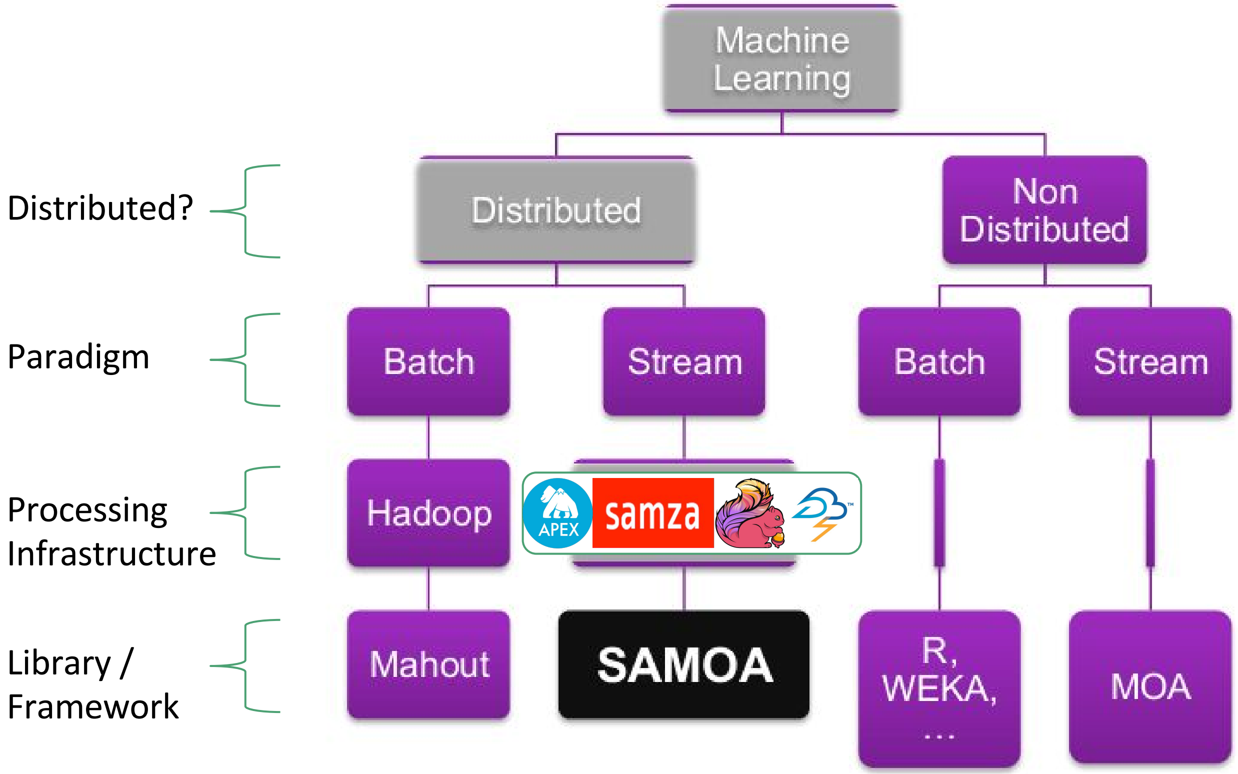


# Apache SAMOA

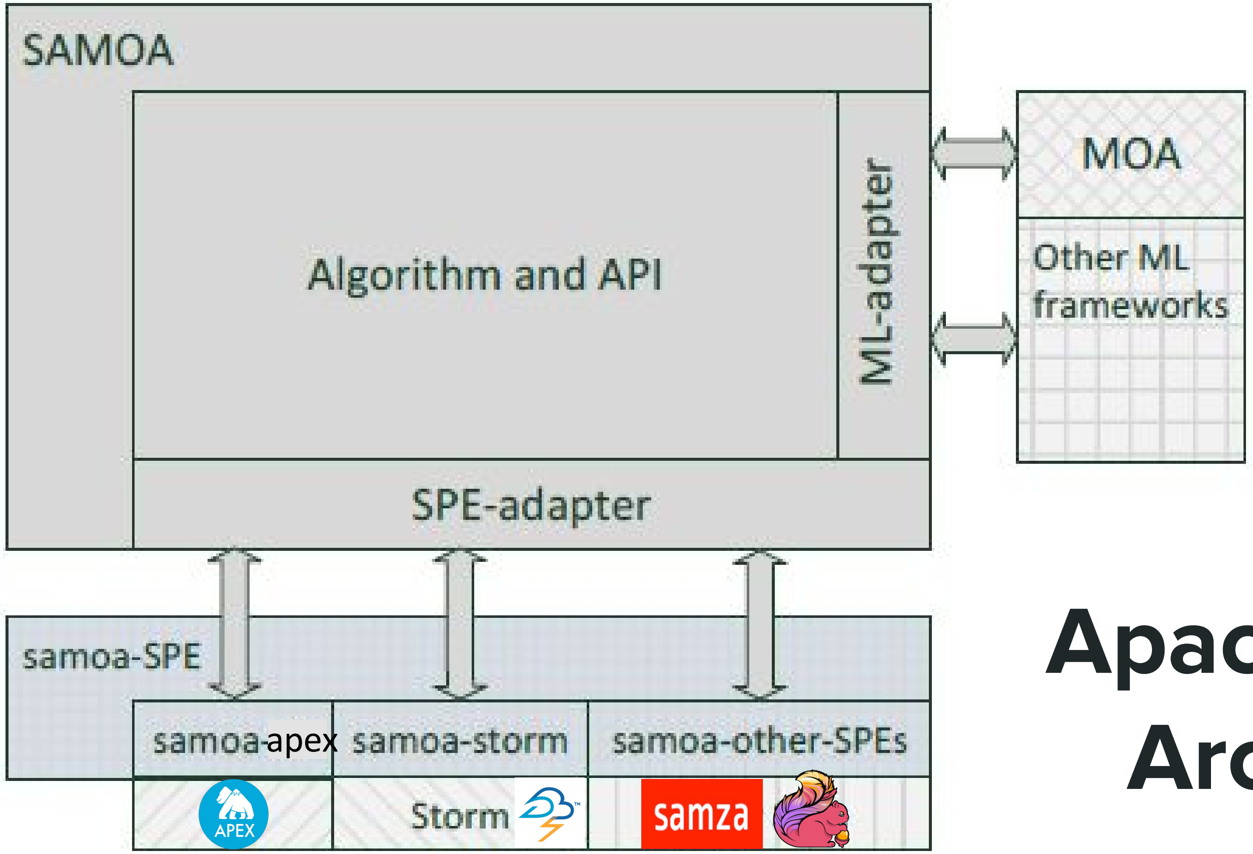
**S**calable **A**dvanced
**M**assive **O**nline **A**nalysis

- What we need
  - Platform for streaming learning algorithms
  - Distributed, Scalable
- A platform for mining big data streams
- Framework for developing new distributed stream mining algorithms
- Framework for deploying algorithms on new distributed stream processing engines
- A library of Streaming Machine Learning Algorithms

# Apache SAMOA - Taxonomy



Apache SAMOA
Scalable Advanced Massive Online Analysis



# Apache SAMOA Architecture

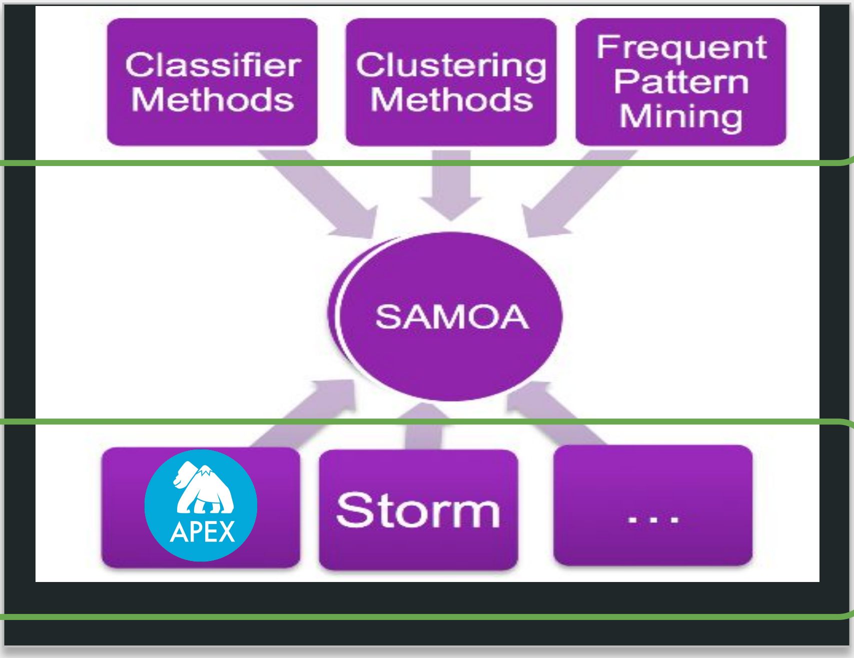
Apache SAMOA
Scalable Advanced Massive Online Analysis

State-of-the-art implementations for distributed machine learning on streams

## Adapter Layers

**ML Algorithms**

**Distributed Stream Processing Engines**



Minimal API to cover all modern DSPEs

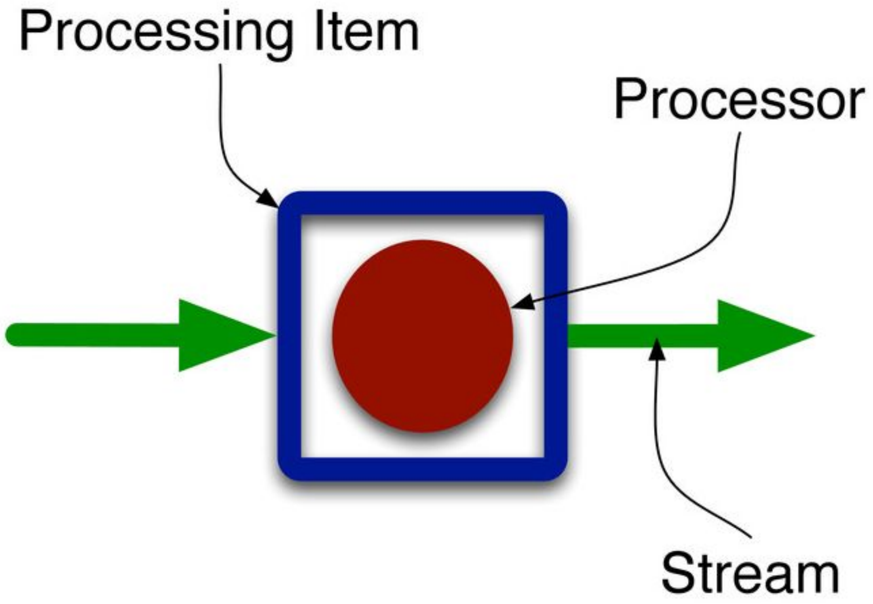
## Why is SAMOA important?

- Program once, run everywhere
- Avoid deploy cycles
    - No system downtime
    - No complex backup/update process
    - No need to select update frequency

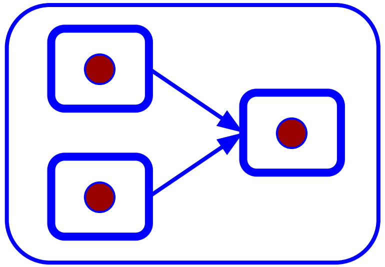
Apache SAMOA
Scalable Advanced Massive Online Analysis

APEX

## Logical Building Blocks



## Apache SAMOA Developer API



```
TopologyBuilder builder;
Processor sourceOne = new
SourceProcessor();
builder.addProcessor(sourceOne);
Stream streamOne =
builder.createStream(sourceOne);
Processor sourceTwo = new
SourceProcessor();
builder.addProcessor(sourceTwo);
Stream streamTwo =
builder.createStream(sourceTwo);
Processor join = new JoinProcessor();
builder.addProcessor(join)
    .connectInputShuffle(streamOne)
    .connectInputKey(streamTwo);
```

## SPE Adapter Layer

- Component Factory
  - ApexComponentFactory
    - `createTopology`
    - `createEntrancePi`
    - `createPi`
    - `createStream`
- Topology -
  - `Apex Topology - DAG`
    - `addEntranceProcessingItem`
    - `addProcessingItem`
    - `addStream`
- Other interfaces for functionality
  - `EntranceProcessingItem`
  - `ProcessingItem`
  - `Stream`

## Build and Run

- Get SAMOA

```
$   git clone https://github.com/apache/incubator-samoa.git
$   cd incubator-samoa
```

- Build for a DSPE

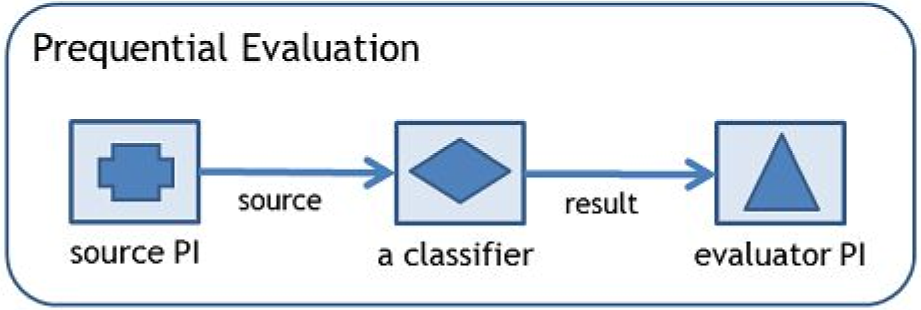
```
$   mvn -Papex package
$   mvn -Pstorm package
$   mvn -Pflink package
```

- Run

```
$   bin/samoa apex ../SAMOA-Apex-0.4.0-incubating-SNAPSHOT.jar "PrequentialEvaluation
    -d /tmp/dump.csv
    -l (classifiers.trees.VerticalHoeffdingTree -p 2)
    -s (org.apache.samoa.streams.ArffFileStream
         -s HDFSFileStreamSource
         -f /tmp/bhupesh/input/covtypeNorm.arff)"
```

## Prequential Evaluation Tasks in SAMOA

- Interleaved test-then-train
- Evaluates performance for online classifiers
  - Basic - Overall
  - Sliding Window Based - Most recent



## Apache Apex DSPE

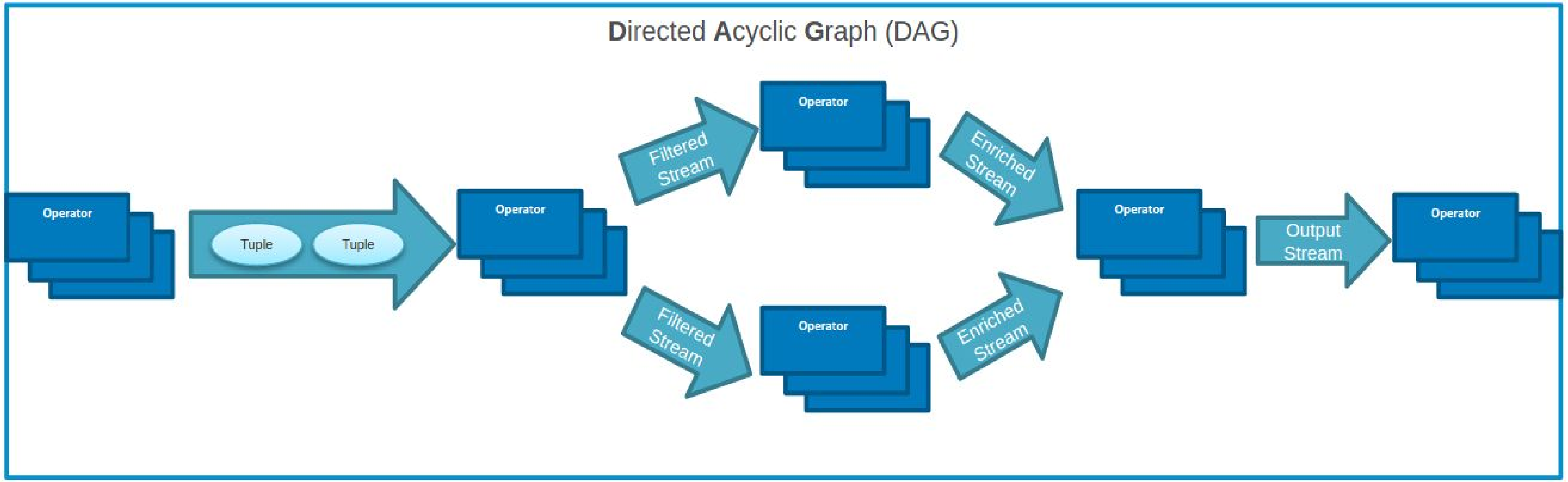
- Highly Scalable
- Highly Performant
- Fault Tolerant
- Stateful Recovery
- Built-in Operability

**D**istributed **S**tream **P**rocessing **E**ngine

## Project History

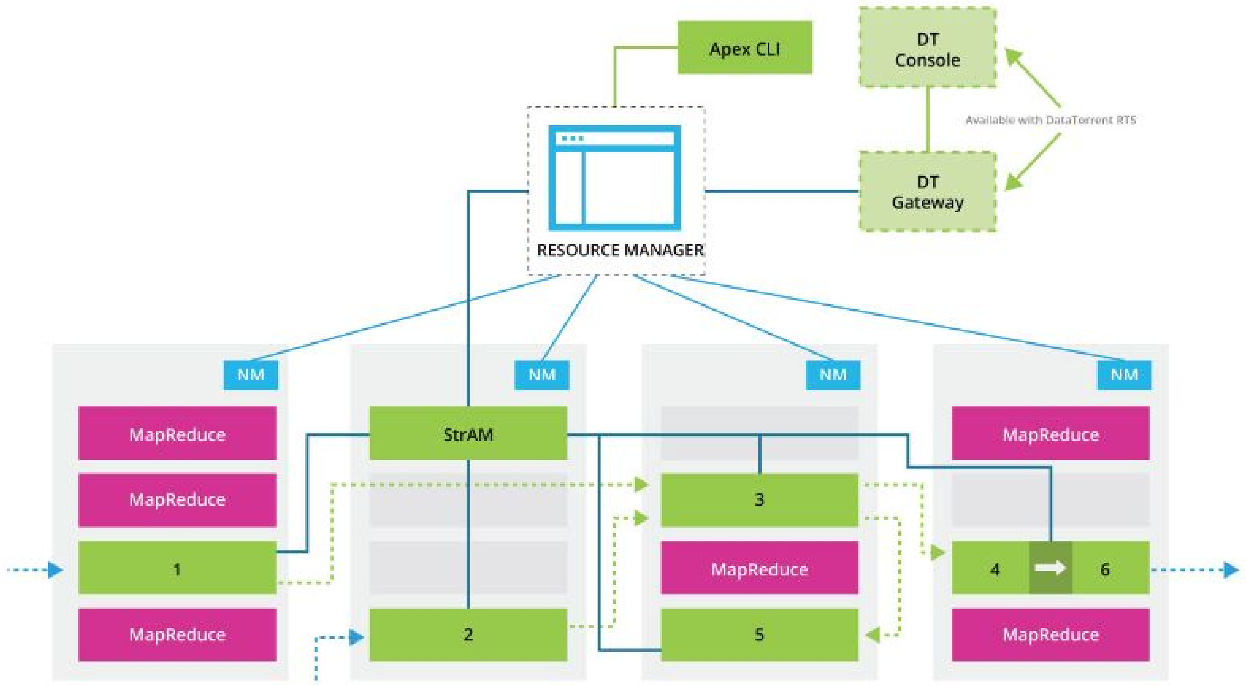
- Project development started in 2012 at DataTorrent
- Open-sourced in July 2015
- Apache Apex started incubation in August 2015
- Top Level Apache Project in April 2016

# Apex Application - DAG



- A DAG is composed of vertices (***Operators***) and edges (***Streams***).
- A ***Stream*** is a sequence of data tuples which connects operators at end-points called ***Ports***
- An ***Operator*** takes one or more *input streams*, performs computations & emits one or more *output streams*
    - Each operator is USER's business logic, or built-in operator from the Apache Apex Malhar library
    - Operator may have multiple instances that run in parallel

# Apex - As a YARN Application



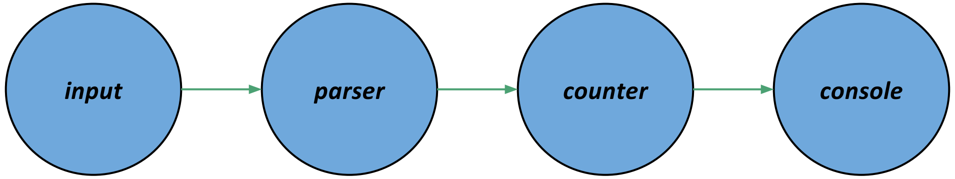
# Apache Apex API

**D**irected **A**cyclic **G**raph

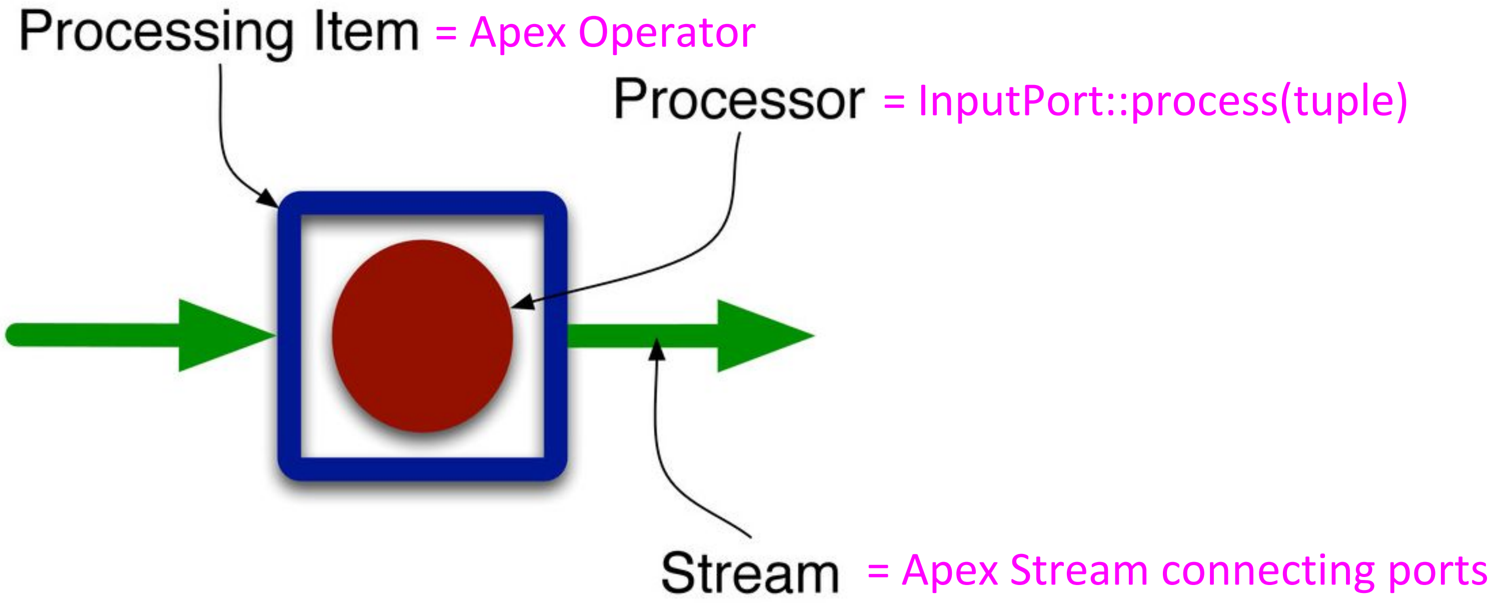
*populateDag()*

```
LineReader input = dag.addOperator("input", new
LineReader());
Parser parser = dag.addOperator("parser", new
Parser());
UniqueCounter counter = dag.addOperator("counter", new
UniqueCounter());
ConsoleOutputOperator out = dag.addOperator("console",
new ConsoleOutputOperator());

dag.addStream("lines", input.out, parser.in);
dag.addStream("words", parser.out, counter.data);
dag.addStream("counts", counter.count, out.input);
```

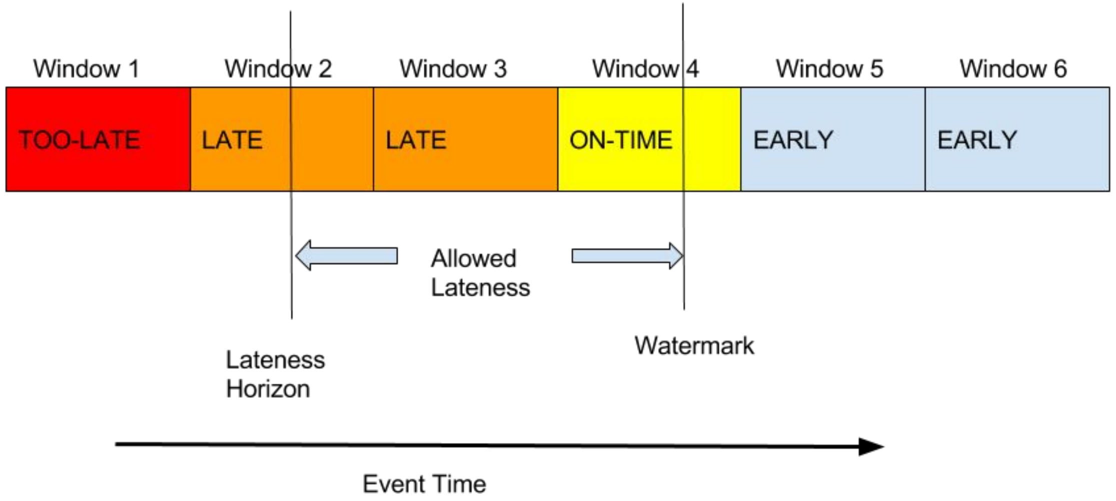


## Logical Building Blocks - Integration


Processing Item = Apex Operator
Processor = InputPort::process(tuple)
Stream = Apex Stream connecting ports

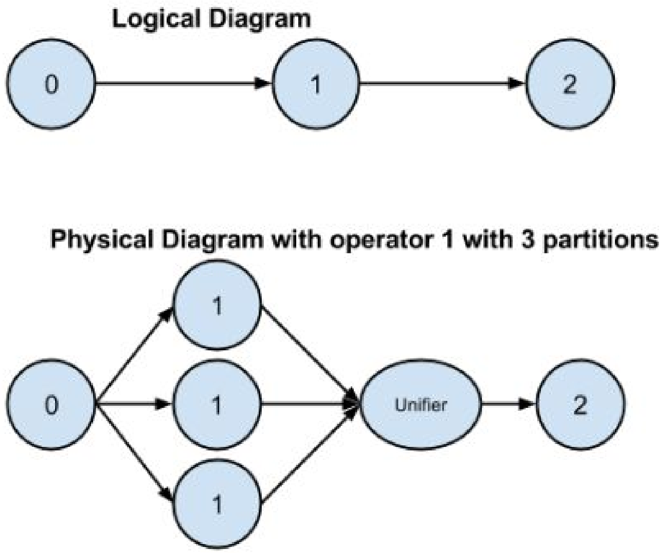
## Support for Windowing

- ***Streaming Windows* -** Finite time sliced windows - Bookkeeping in the engine
- ***Event-time windows-*** Supports concepts like watermarks, triggers and accumulators and sessions - Application level windowing
- ***Checkpoint Windows*** - Governs automatic periodic checkpointing of the operator state by the engine

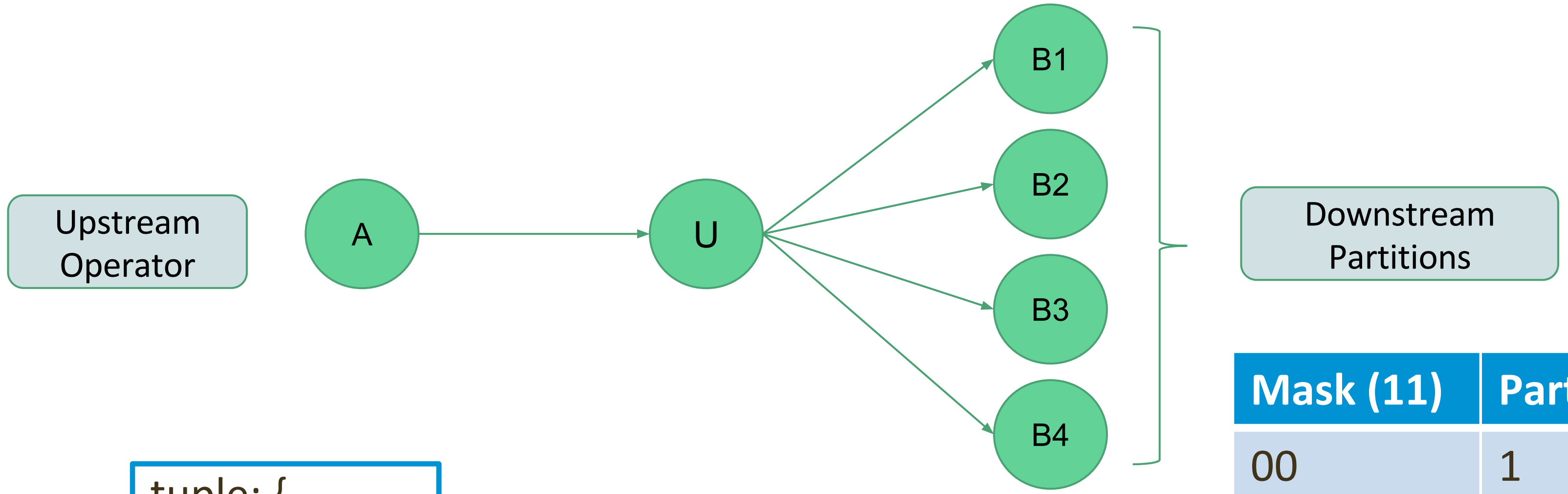


## Scalability - Partitioning

- Requirement: Low latency and high throughput for High Speed Input Streams
- Replicate (Partition) Operator Logic
- Specified at launch time
- Control the distribution of tuples to downstream partitions.
- Automatic pass through unifier or custom unifier to merge results
- Dynamic scaling!



# Stream Codec - Distribution of tuples



tuple: {
Name,
24204842,
San Jose
}

→

Hashcode:
00101010001010**1**

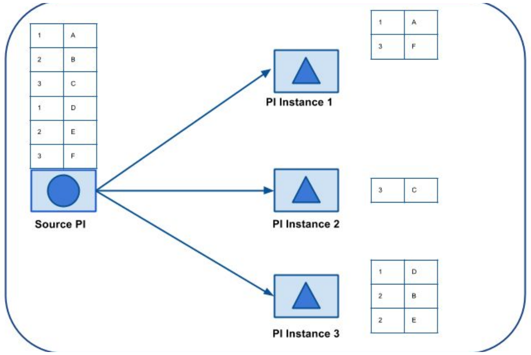
→

| Mask (11) | Partition |
|---|---|
| 00 | 1 |
| 01 | 2 |
| 10 | 3 |
| 11 | 4 |

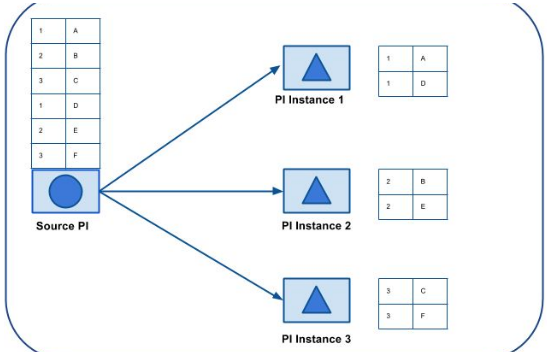
Apache SAMOA
Scalable Advanced Massive Online Analysis

APEX

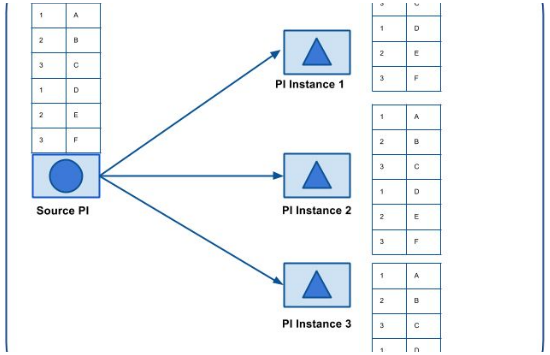
## Stream Connections - Distribution of tuples



Shuffle



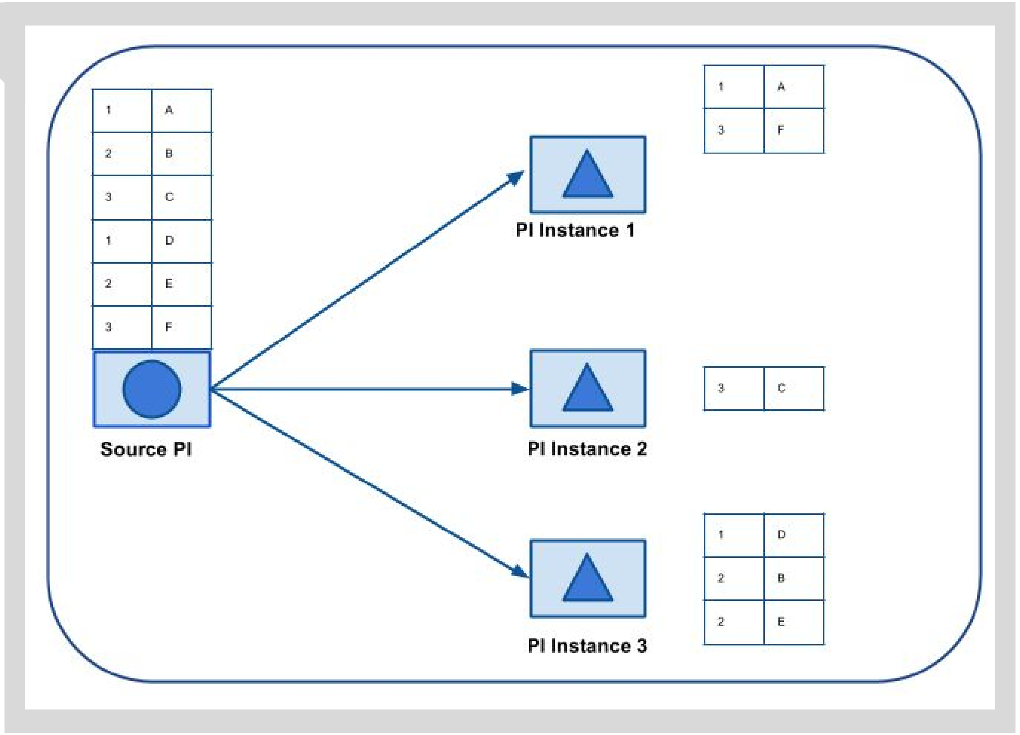
Key



All

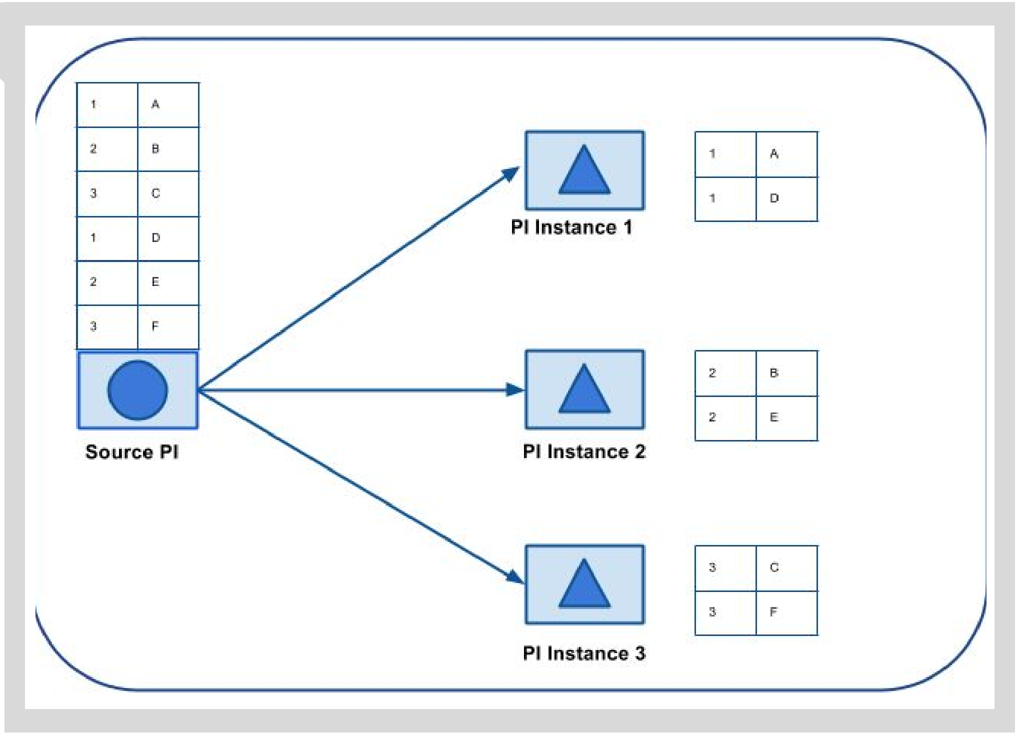
## Message Shuffling

Tuple based Hashcode for Stream codec



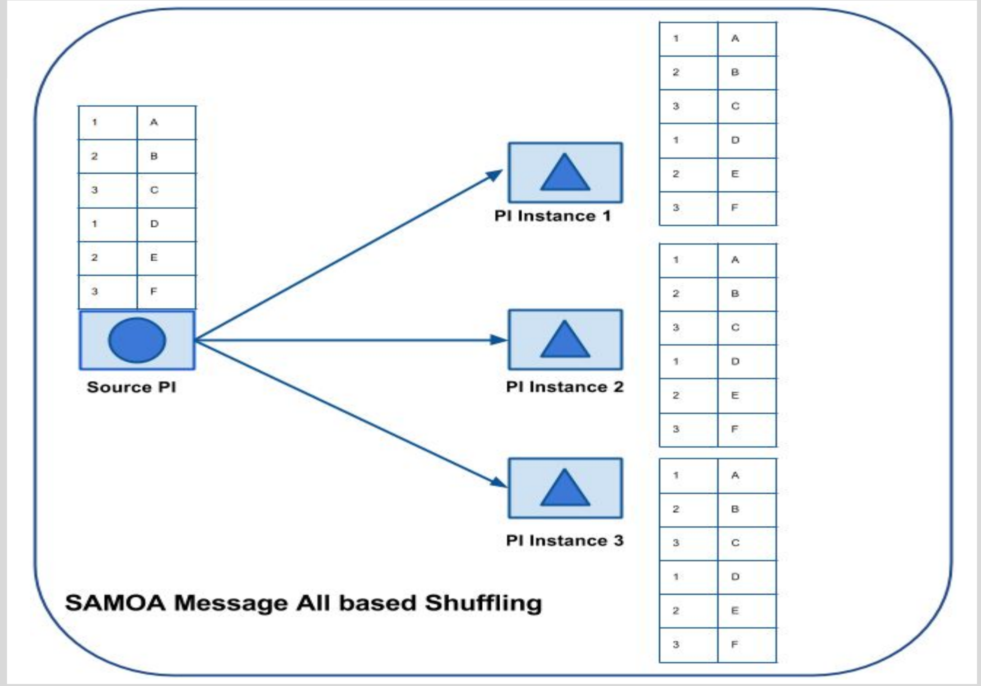
## Key Based Shuffling

Key based Hashcode for Stream codec



## All Based Shuffling (Broadcast)

Custom Partitioner to send all tuples to all downstream partitions



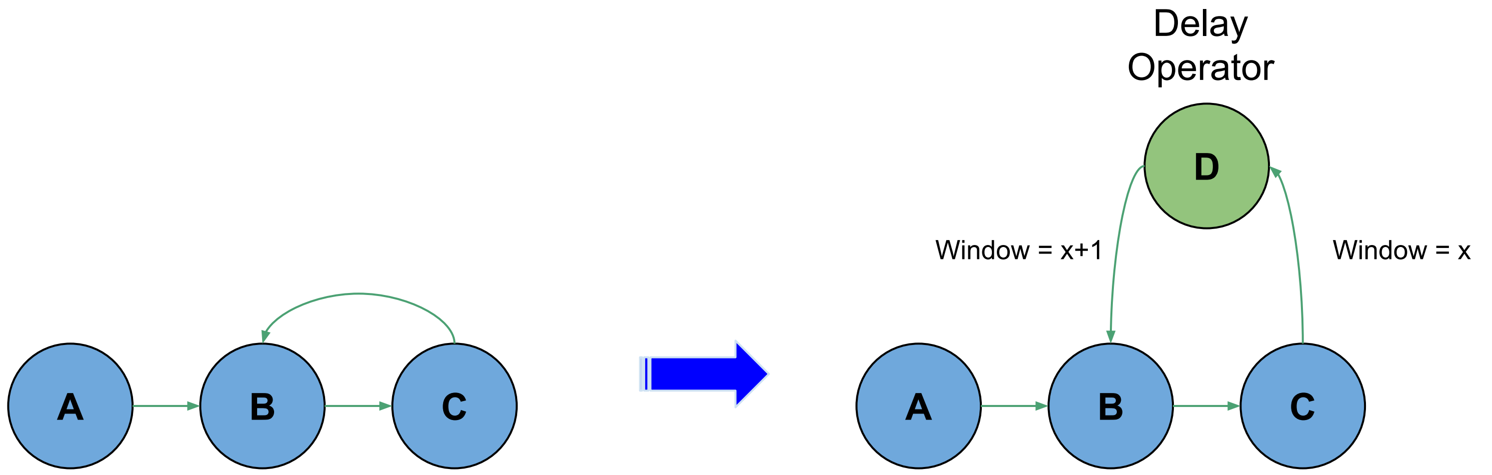
## Iteration support in Apex

- Machine learning needs iterations
  - At the very least, a feedback loop. Example - VHT
- Apex Topology - Predominantly Acyclic - DAG
- Iteration support implemented -
  - Core challenge was fault tolerance and correctness
- Apex maintains the DAG nature of the topology.
  - Cycles, although seemingly present in the logical DAG, maintain the DAG nature while execution.

## Delay Operator

Iteration support

- Increment window id for all outgoing ports
- A note on Fault tolerance -
    - Fabricates the control tuples at the start and at recovery
    - Must replay the first window data tuples at recovery



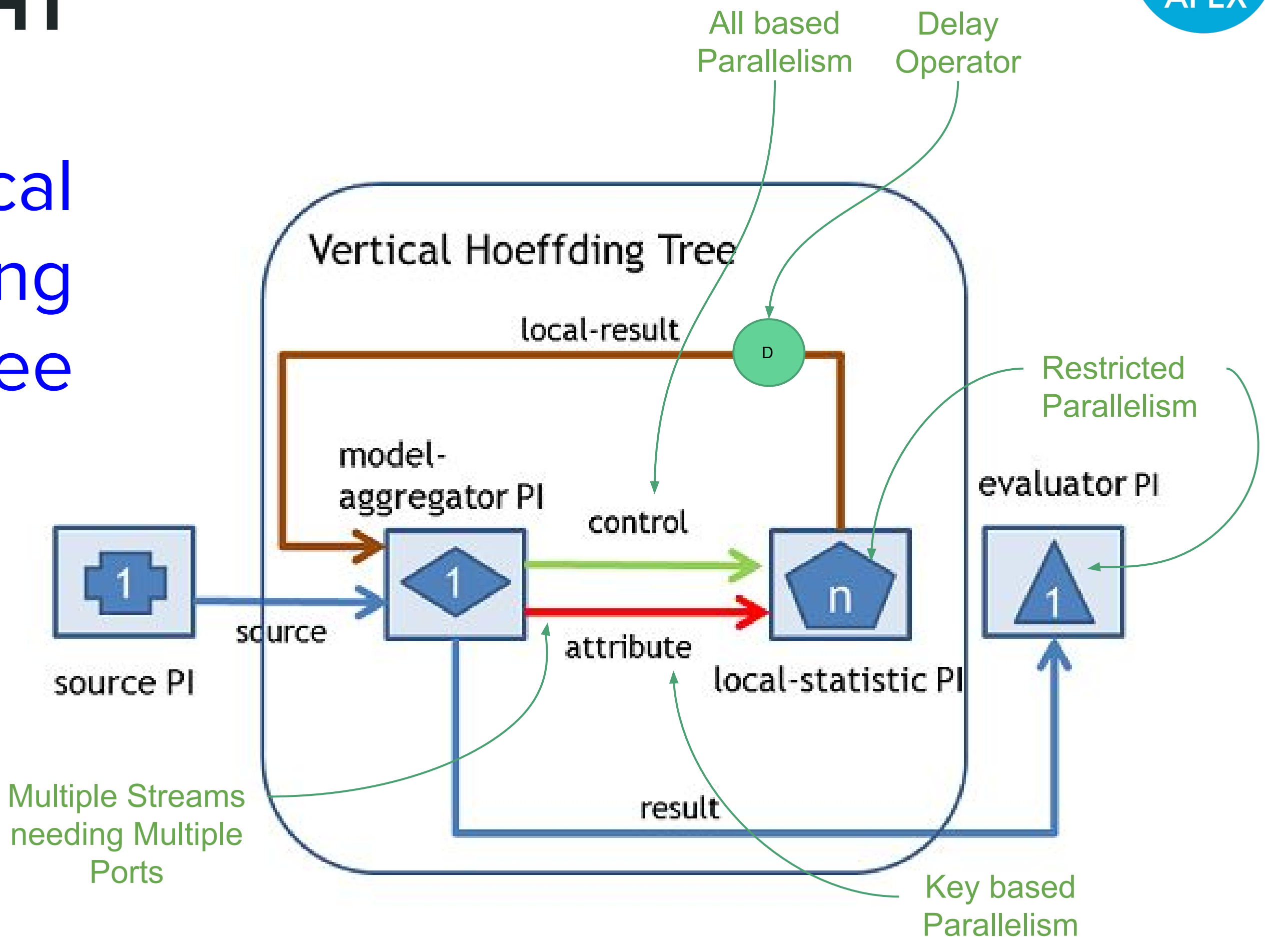
## Challenges

### Adding Runner for Apache Apex

- Differences in the topology builder APIs of SAMOA and Apex
- No concept of Ports in SAMOA
- On demand declaration of streams in SAMOA
- Cycles in topology - Delay Operator
- Serialization of Processor state during checkpointing. Also serialization of tuples.
- Number of tuples in a single window - Affects number of tuples in future windows coming from the delay operator

# Case Study - VHT

**V**ertical
**H**oeffding
**T**ree



Apache SAMOA
Scalable Advanced Massive Online Analysis

APEX

## Roadmap

**SAMOA**

- Stochastic Gradient Descent
- Adaptive + Boosting VHT
- Regression Tree + Gradient Boosted Decision Tree
- Distributed Data Stream Mining using Coresets
- Distributed Data Stream Mining using Sketches

Apache SAMOA
Scalable Advanced Massive Online Analysis

APEX

## Roadmap

## Apex

- SQL support using Apache Calcite
- Apache Beam runner
- Enhanced support for Batch Processing
- Encrypted streams
- Support for Mesos
- Python support for operator logic and API
- Replacing running operators at runtime
- Dynamic attribute changes

Apache SAMOA
Scalable Advanced Massive Online Analysis

APEX

## Key Takeaways

- Samoa brings in a new set of Streaming Machine Learning Algorithms.
- Iterative processing enables Machine Learning on Apache Apex with fault tolerance, maintaining correctness of the workflow.
- Apex as another runner for Apache SAMOA

## Resources

- Apache SAMOA - https://samoa.incubator.apache.org
- Apache Apex - http://apex.apache.org/
- Apache Apex Subscribe - http://apex.apache.org/community.html
- Apache Apex Presentations - http://www.slideshare.net/ApacheApex/presentations
- Apache Apex Download - https://apex.apache.org/downloads.html
- Twitter
  - @ApacheSamoa Follow - https://twitter.com/apachesamoa
  - @ApacheApex Follow - https://twitter.com/apacheapex
- Apache Apex Meetups - http://www.meetup.com/topics/apache-apex
- Apache Apex Webinars - https://www.datatorrent.com/webinars/
- Apache Apex Videos - https://www.youtube.com/user/DataTorrent

Apache SAMOA
Scalable Advanced Massive Online Analysis

APEX

## Questions ?

Apache SAMOA
Scalable Advanced Massive Online Analysis

APEX

## Thank You!