Percolator: Scalable Pattern Discovery in Graph Streams

Sutanay Choudhury[1], Sumit Purohit[1], Peng Lin[2], Yinghui Wu[2], Lawrence Holder[2], Khushbu Agarwal[1]

Pacific Northwest National Laboratory[1], Washington State University[2]

{sutanay.choudhury, sumit.purohit, khushbu.agarwal}@pnnl.gov, {plin1, yinghui, holder}@eecs.wsu.edu

Motivation

- Discovering emerging events from massive graph streams is a critical problem in a wide range of applications. E.g., Web, social media and cyber networks are often represented as graph patterns.

- **New challenges:**
  - ✓ Conventional event detection is mostly over item sets.
  - ✓ ***Complex events*** are characterized as ***graph patterns***.
  - ✓ Pattern matching/mining is expensive.
  - ✓ Online maintenance and parallel detection of graph streams.

- **Contributions**:
  - Feasible detection of events as graph patterns over graph streams
  - Integrate incremental pattern mining and parallel pattern mining for massive graph data

Events as Graph Patterns.

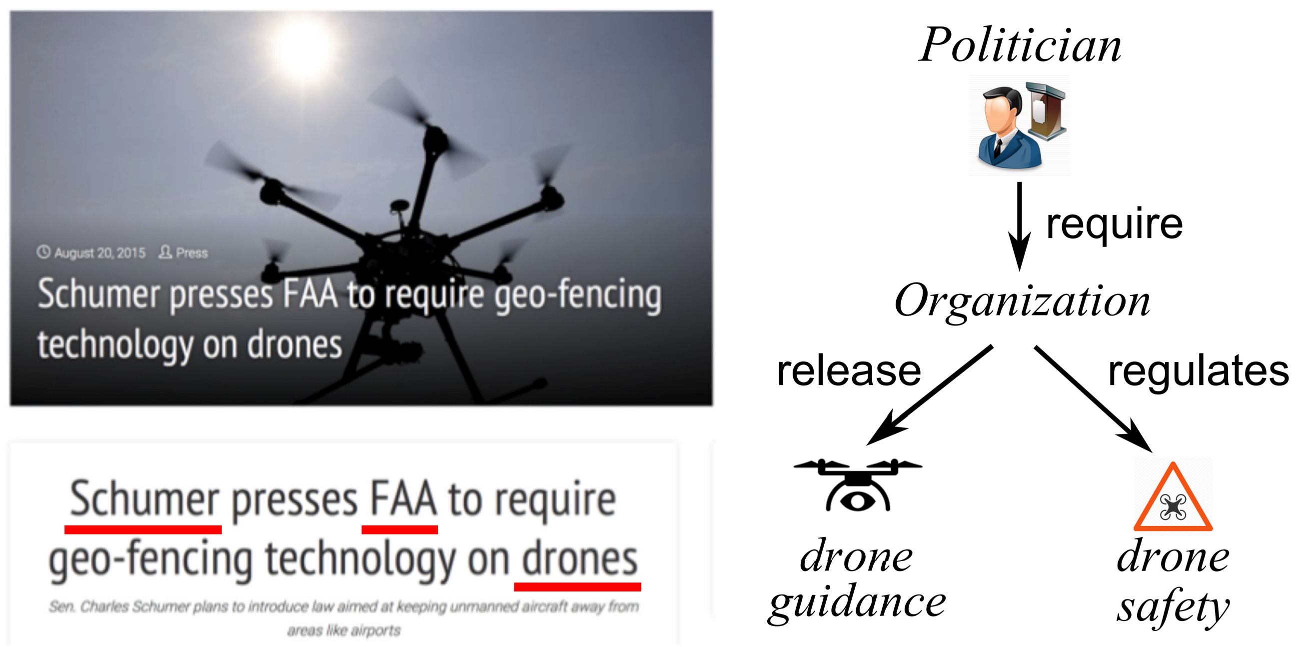


Fig 1. Emerging events from News data as triples. The event is a ***graph pattern*** that verifies a news that politicians (*e.g.,* "Schumer") are pressing organizations (*e.g.,* "Federal Aviation Administration (FAA)") to regulate drones and provide guidance. The event is detected by mining and tracking frequent graph patterns continuously in news reports as a stream of RDFs

Graph Stream Discovery Problem

Interestingness Measure

- Active patterns in G_i:

$$Act(P, G_i) = \min_{u \in V_P} |\varphi_l(u)|$$

- If $Act(P, G_i) \geq \theta$, a user-defined threshold, the event is active.

Problem Statement

- **Online pattern discovery over graph stream**

Given a graph stream G_i over the time range $[1, t]$ and a support threshold θ, Percolator detects and maintains the maximal active patterns Σ^i *w.r.t* θ at any time i in $[1, t]$, upon the ad-hoc queries.

Percolator System

A distributed pattern discovery tool over graph streams

- **Events as Graph Patterns**
  - ✓ Support online analytic add-hoc queries
  - ✓ Offline trend analysis and anomaly detection

- **Fast**
  - ✓ Incremental pattern mining
  - ✓ Parallel pattern mining: "think like a pattern"

- **Easy-to-use**
  - ✓ A user-friendly GUI
  - ✓ Graph visualization.

- **Open Source**
  - ✓ https://github.com/streaming-graphs/NOUS

System Overview

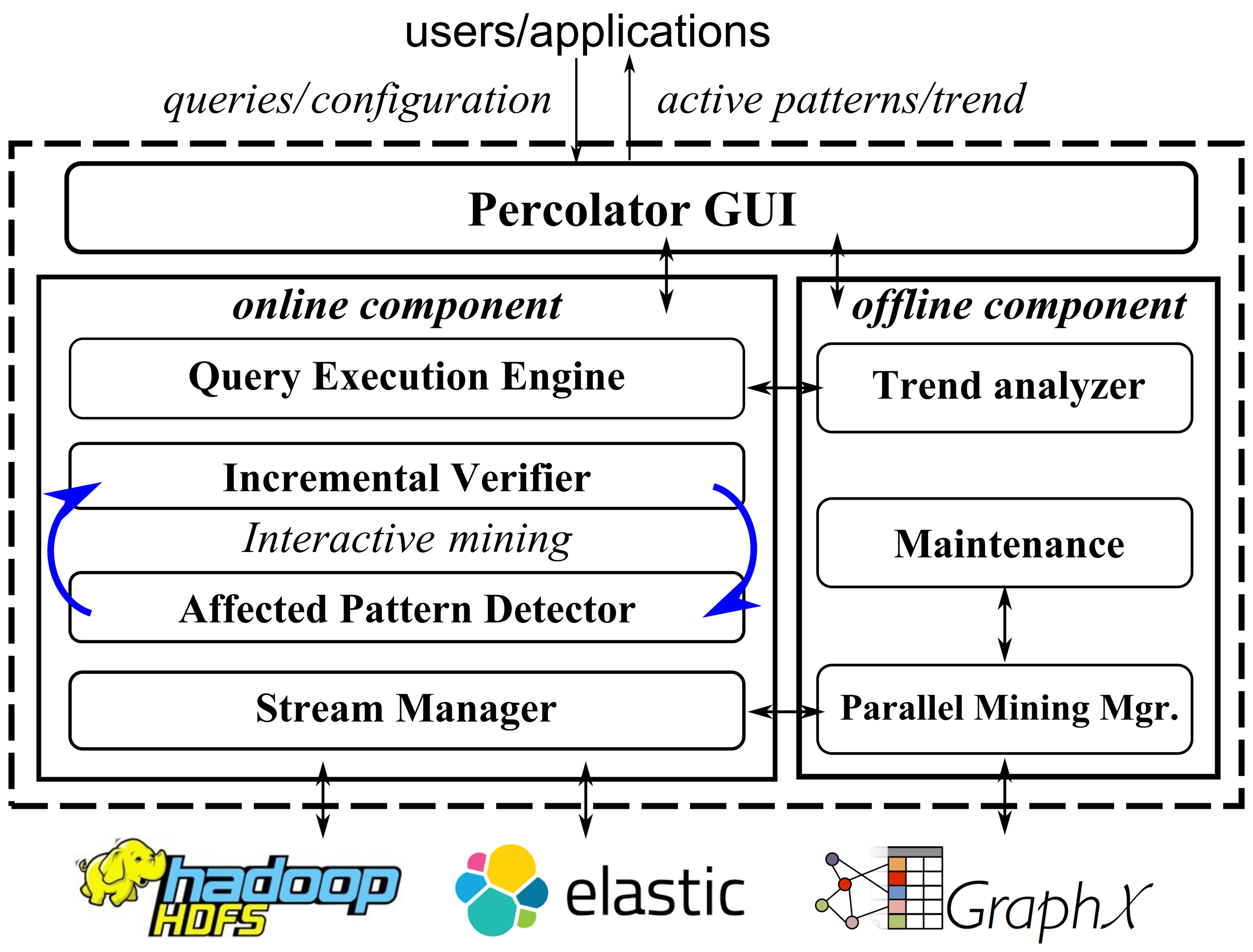


Fig 2. Architecture of Percolator.

Performance

- **Deployment:** 16 nodes cluster (1 coordinator, 15 workers), each of which has 16 cores of 2.33GHz Intel Xeon CPU and 128GB memory.

- **Efficiency:** 245 seconds to process 10 million updates per batch of edges with 8 workers in parallel.

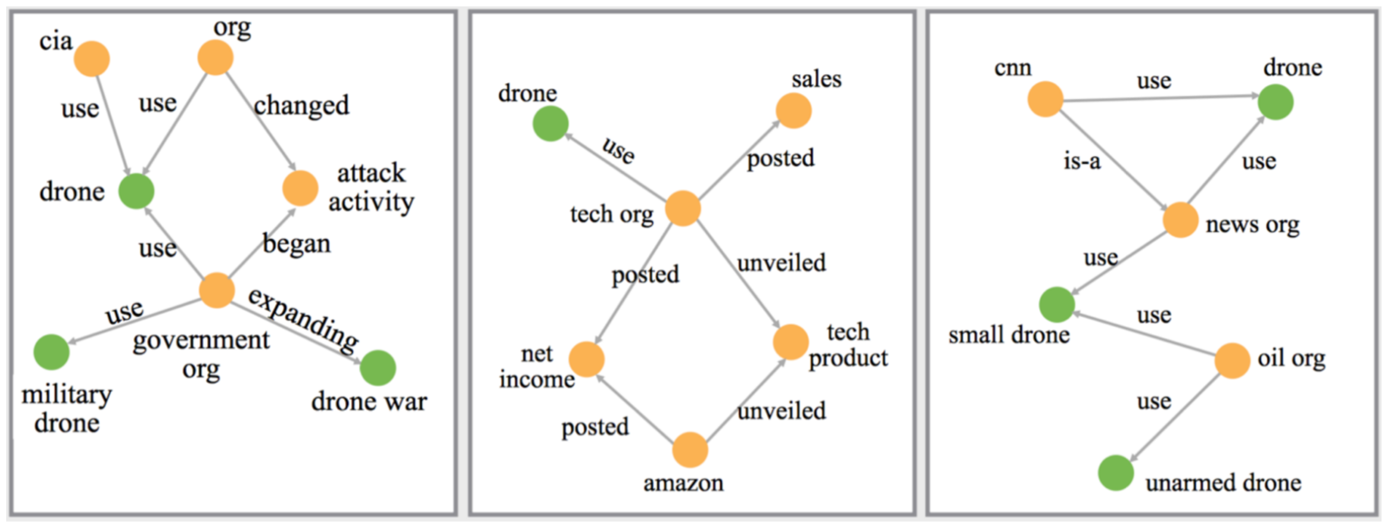
- **Scalability:** 2.1 times faster when the number of workers varies from 2 to 8 over the **MAG** dataset.

Discovery Algorithms

- **Online Incremental Mining**
  - * **Stream Manager**
    - An edge buffer B, a pattern lattice $\mathcal{T}^i$ associated with $Act(\cdot)$
  - * **Affected Pattern Detector**
    - Identify the minimal set of patterns which change status.
  - * **Incremental Verifier & Mining**
    - Incrementalize subgraph isomorphism and update $Act(\cdot)$

- **Parallel Incremental Mining**
  - * "Think like a pattern": call BSP model in each superstep.
  - * Only verify the incurred affected patterns on each worker.

Application: Trend Analysis



2010-10-13	NATO **launches** unmanned drone
2013-11-12	CyPhy Works Inc. **makes** drones
2014-10-05	Amazon **unveiled** drone market
2015-01-09	GoPro **developing** drones
2015-01-13	CNN **use** drones

Fig 3. Real world trend of drone usage: 2010-2015.

- Percolator first found early reports in 2010 that organizations use drones for military activities.

- As time passes, it observed emerging patterns related to high-tech companies such as "Amazon" in 2013.

- Later patterns (2015) suggested that drones started to be used in various industries including oil, news, and constructions.

Acknowledgment

The research is sponsored by the Analysis in Motion Initiative at Pacific Northwest National Laboratory. Wu and Lin are supported in part by NSF IIS-1633629 and Huawei HIRP grant.