

Online trajectory analysis with scalable event recognition

E. Ntoulas¹ E. Alevizos^{1 2} A. Artikis^{1 3}
A. Koumparos⁴

¹Institute of Informatics & Telecommunications, NCSR Demokritos, Greece

²Department of Informatics & Telecommunications, National and Kapodistrian University of Athens, Greece

³Department of Maritime Studies, University of Piraeus, Greece

⁴Vodafone Innovus

4th Big Mobility Data Analytics Workshop, EDBT
March 23, 2021

Introduction

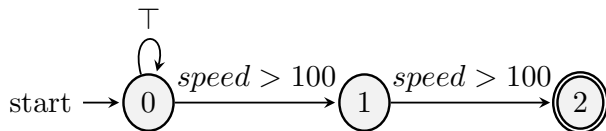
- Monitoring of moving objects:
 - Detect critical situations to avoid accidents and ensure regulations.
 - Detect them as fast as possible to provide margins for action.
- Solution:
 - Use Wayeb, a Complex Event Processing tool, to identify such situations.
 - Implement and compare parallel techniques on top of Wayeb to improve scalability.

What is Complex Event Processing?

vehicle id	78986	78986	78986	78986	78986	...
speed	85	93	99	104	111	...
timestamp	1	2	3	4	5	...

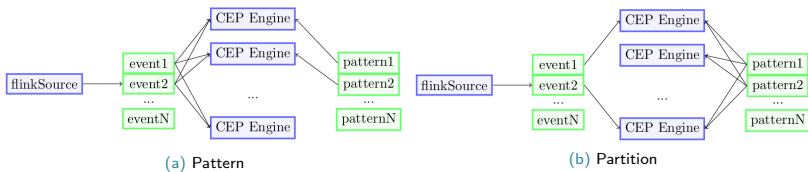
- Events in a stream come in the form of tuples with both numerical and categorical values
- Through the use of patterns on such events we identify complex events.
- e.g. A vehicle exceeding a speed limit of 100 km/h for two consecutive events.

Automata-based Event Recognition



- Wayeb transforms these patterns to symbolic automata as its computational model.
- Transitions use boolean predicates that operate on the attributes of the events.

Event Recognition with Flink 1/2



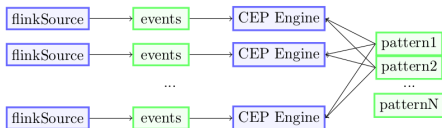
(a) Pattern-based parallelization:

- Patterns distributed in round robin format to engines
- Events are broadcast to all engines

(b) Partition-based parallelization:

- Each engine receives the full set of patterns.
- Each event goes to one engine.
- A partitioner decides in which engine each event is sent to.

Event Recognition with Flink 2/2



(c) Partition Based parallelization with parallel sources

- Flink gives us the option to use parallel input streams.
 - Used to emulate streams of greater event rate.
- (c) Special case of Partition-Based parallelization when one-to-one relation between sources and CEP engines exists.
 - Events are forwarded to the corresponding engine.
 - Each engine receives the full set of patterns.

- Fleet Management: 270M events. Covers a period of 5 months, from June 30,2018 11:00:00 PM to November 30, 2018 11:59:59 PM
- Maritime Monitoring: 18M events from 5K vessels sailing in the Atlantic Ocean around the port of Brest, France, between October 1st 2015 and 31st March 2016 (6 months).

Fleet Management Patterns

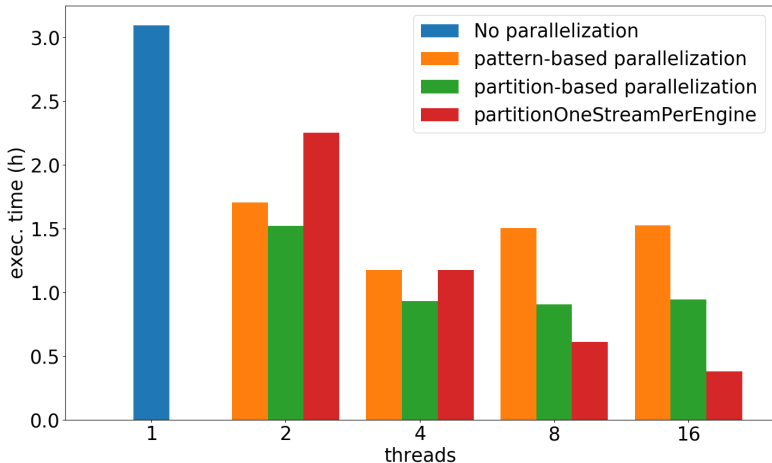
Experts have provided us with patterns:

- Route:
 - Basic element of vehicle management.
 - Defined as the time period between periods of the vehicle being parked.
- Identify erroneous patterns in the data that imply:
 - hardware malfunctions,
 - bad connection during the device installation
 - lack of satellites tracking the vehicle
- Extra: Refuel opportunity and dangerous driving

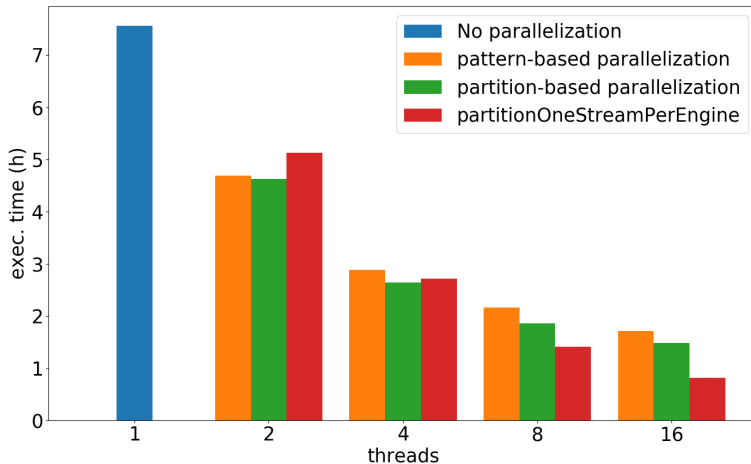
Maritime Patterns

- Patterns deemed of special significance in previous works by domain experts:
 - High speed near coast
 - Anchored
 - Drifting
 - Search and rescue
 - Trawling
 - Loitering.
- Vessels approaching a port:
 - Use the same pattern with different port eachtime
 - Emulate loads of higher magnitude.

Fleet Management Execution Times, 16 patterns



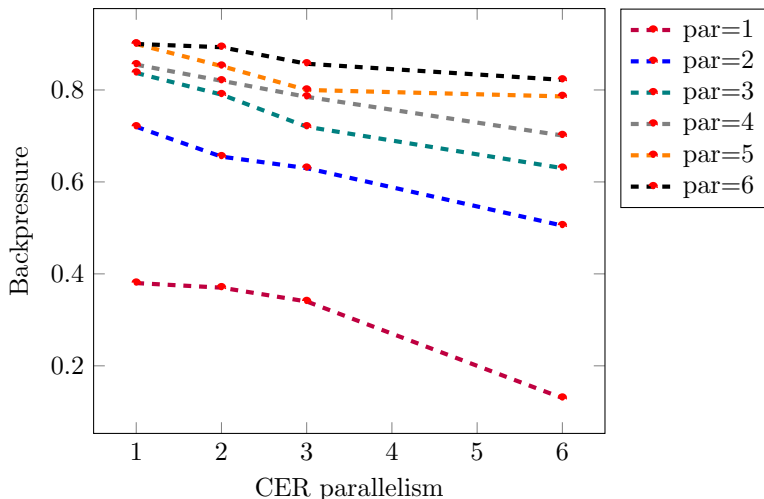
Fleet Management Execution Times, 48 patterns



Defining Backpressure

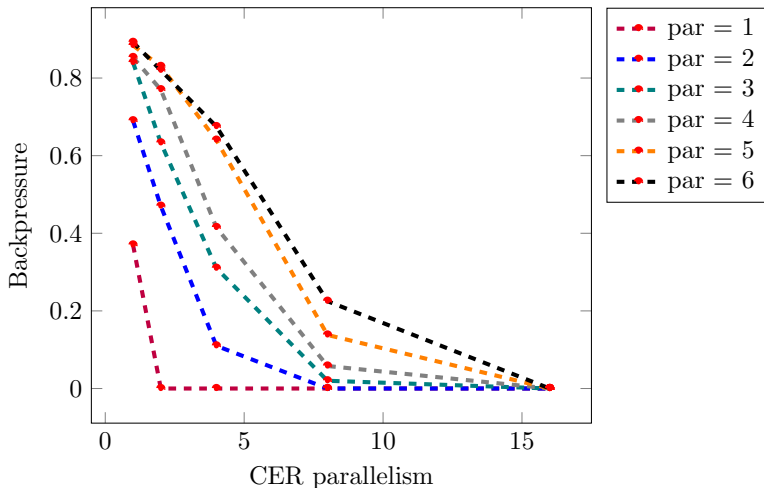
- BackPressure: Flink metric that notifies you if your operators can consume the data from the stream source as fast as they arrive.
 - In our case, the operator is the distributed version of Wayeb.
- Measured as a percentage:
 - 0% pressure means the operators consume the events as fast as they arrive.
 - I.e., their throughput is at least equal to the incoming event rate.

Pattern parallelization - (6 Patterns)



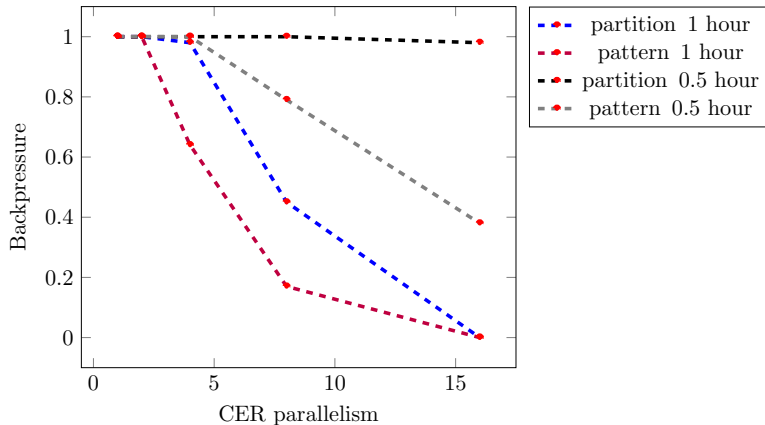
par: parallelism of the source. Black line (6 threads) is 490K events/second. Pink line (1 thread) is 110K events/second

Partition parallelization - (6 Patterns)



par: parallelism of the source. Black line (6 threads) is 490K events/second. Pink line (1 thread) is 110K events/second

BackPressure - Maritime (200 Patterns)



- Simulated the whole stream to run in half and in one hour due to larger load.

Summary

Conclusion:

- effectively process streams of event rate at least 490K events/second
- superiority of partition-based over pattern-based parallelization, when patterns are few.
- pattern-based parallelization is viable for many patterns.

Future Work:

- Combine various distribution techniques
- Construct more patterns for the domains presented.
- Compare our automata-based method against other approaches.