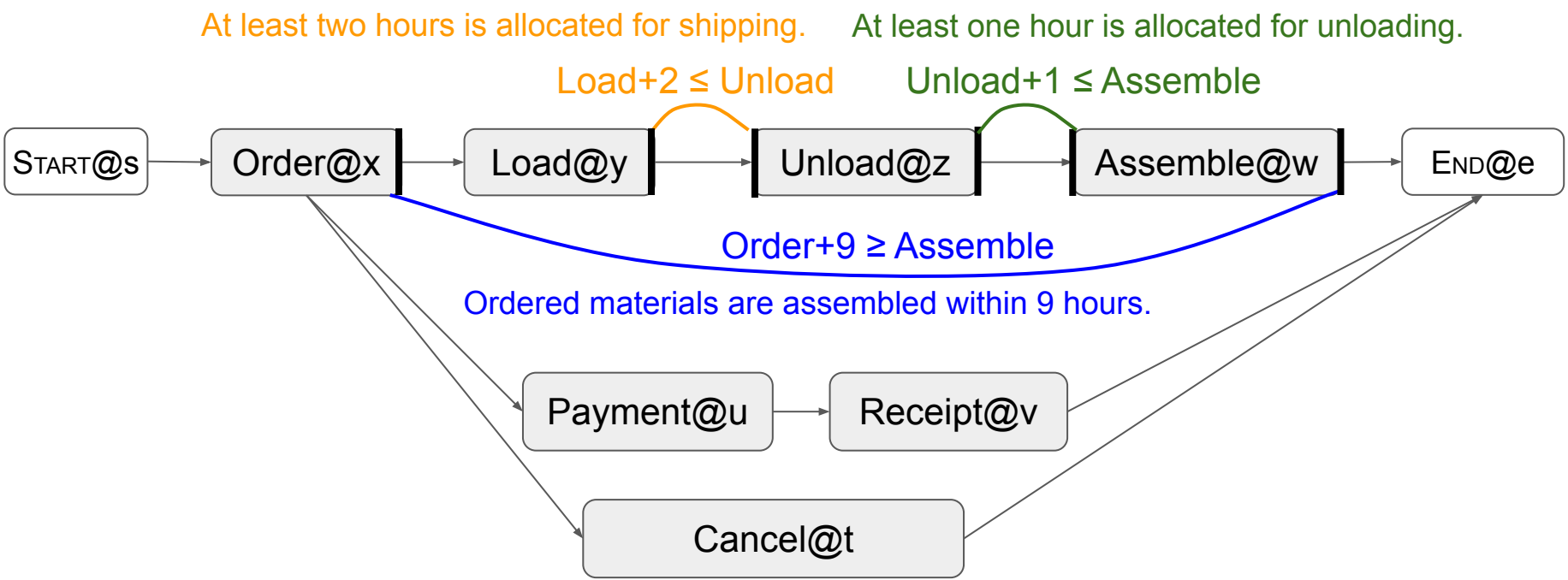
Early Detection of Temporal Constraint Violations

Isaac Mackey, Raghubir Chimni, Jianwen Su
University of California, Santa Barbara

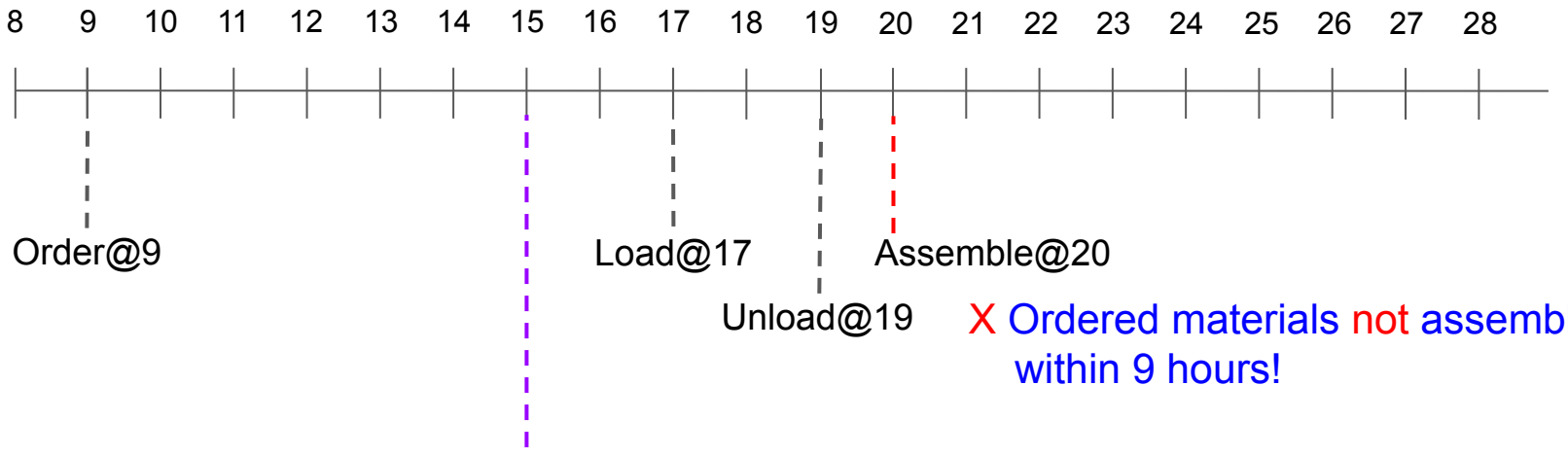
Collecting Materials for Construction

Workflow to manage projects' activities



A Violation of Time Constraints

Rule: Order → Load+2 ≤ Unload, Unload+1 ≤ Assemble, Assemble ≤ Order+9



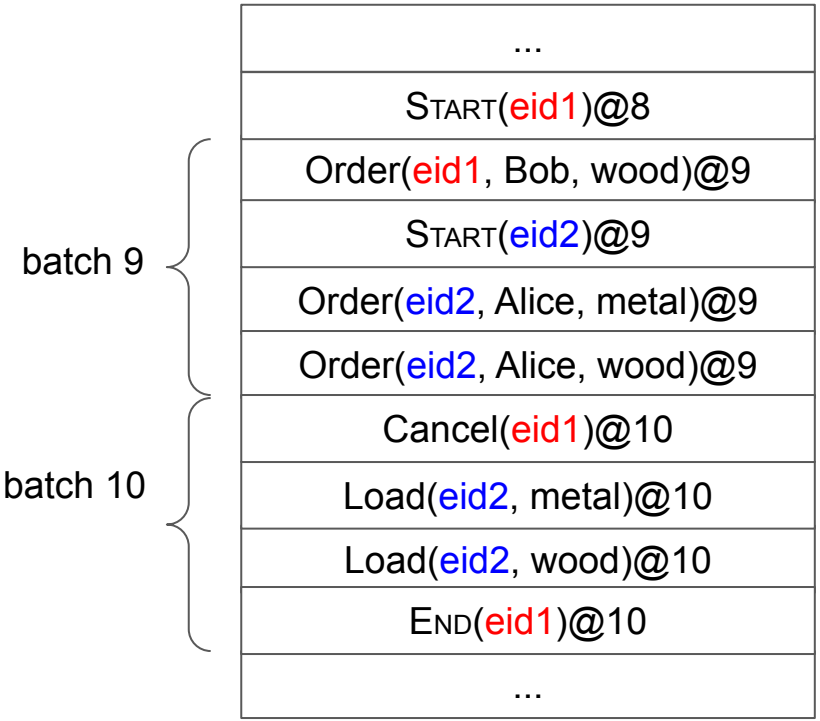
A violation is guarenteed at time 15.
Early detection can prevent costly shipping, waste of labor and materials, etc.

Given timing constraints and an updating enactment, identify rule violations *as early as possible.*

Outline

- Motivating early detection
- Events, enactments, and constraints
- Violation detection algorithm
- Experimental evaluation
- Conclusions

Events, Enactments, Logs, Batches



event: reports activities' completion
- integer timestamps
- data attributes

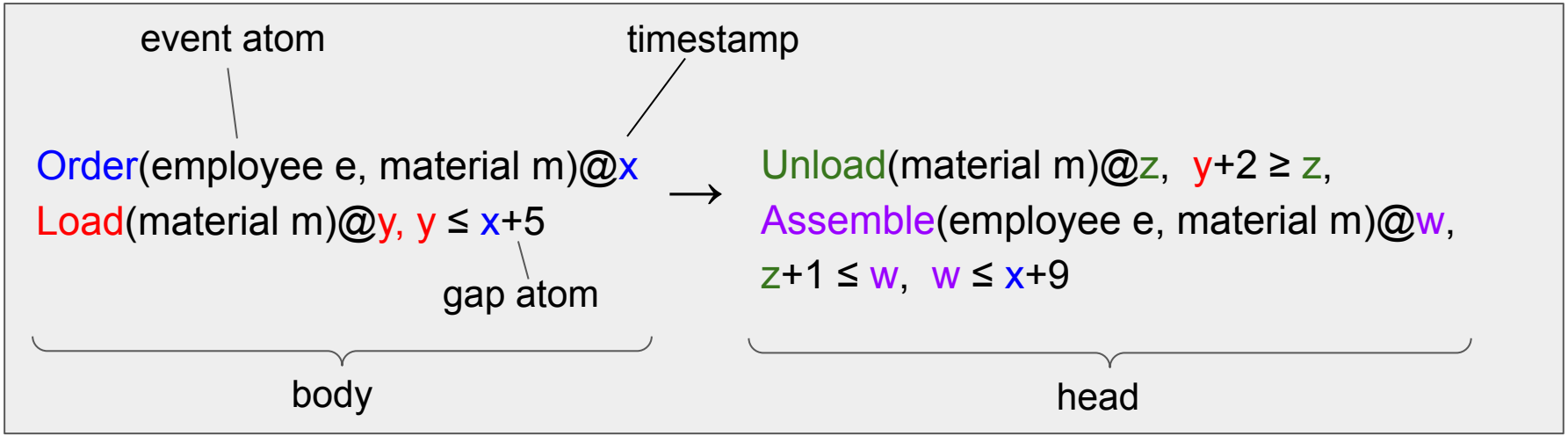
enactment: events from one workflow instance; related by enactment id (eid1)

log: time-ordered events in multiple enactments

batch: events in a log with same timestamp

A Rule-Based Temporal Constraint Language

Ex. In each enactment, for every Order by a employee for a material Loaded with 5 hours, the material is Unloaded after 2 hours, and Assembled by the employee after at least 1 hour from Unloading and within 9 hours of Ordering.



Rule satisfaction for an enactment: natural first-order logic semantics $\eta \models \phi \rightarrow \psi$

Single enactment semantics | Order(*eid1*, Bob, wood)@9

Technical Problem: Early Detection

Given a rule $\phi \rightarrow \psi$ and batches of an enactment η, can we report rule violations at the earliest possible time?

A body assignment σ is a *violation* of $\phi \rightarrow \psi$ for η if

- $\eta \models \phi[\sigma]$, and
- For all extensions η' of η, no head assignment $ß$ extends σ with $\eta' \models \psi[ß]$

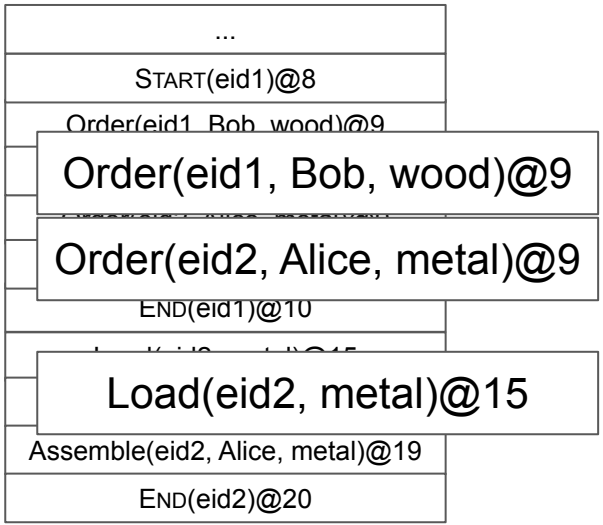
Report unmatched body assignments that can't be matched later

Key Steps

1) Generate body and head assignments
2) Match body and head assignments
3) Detect body assignments that can't be extended

Step 1: Generate Assignments

Log



Order(employee e, material m)@x, Load(material m)@y, y ≤ x+6

Body assignments

aid	eid	e	m	x	y	gaps
b1	eid1	Bob	wood	9		y ≤ 15
b2	eid2	Alice	metal	9		y ≤ 15
b3	eid2		metal		15	9 ≤ x
b4	eid2	Alice	metal	9	15	

Unload(material m)@z, y+2 ≥ z,
Assemble(employee e, material m)@w, z+1 ≤ w, w ≤ x+9

Head assignments

aid	eid	e	m	z	w	gaps
h1	eid3		metal	17		y ≥ 15, 18 ≤ w, w ≤ x+9
h2	eid4	Alice	metal		19	y+2 ≥ z, z ≤ 18, 10 ≤ x
h3	eid5	Alice	metal	17	19	y+2 ≥ 15, 10 ≤ x

Step 2: Match Assignments

Body assignments

aid	eid	e	m	x	y	gaps
b1	eid1	Bob	wood	9		y ≤ 15
b2	eid2	Alice	metal	9		y ≤ 15
b3	eid2	Alice	wood	9		y ≤ 15
b4	eid2		metal		15	9 ≤ x
b5	eid2	Alice	metal	9	15	

Head assignments

aid	eid	e	m	z	w	gaps
h1	eid3		metal	17		y ≥ 15, 18 ≤ w, w ≤ x+9
h2	eid4	Alice	metal		19	y+2 ≥ z, z ≤ 18, 10 ≤ x
h3	eid5	Alice	metal	17	19	y+2 ≥ 15, 10 ≤ x

Extension table

body aid	head aid	gaps	deadline
b5	Ø	17 ≥ z, z+1 ≤ w, w ≤ 18	17
b5	h1	18 ≤ w, w ≤ 18	18

Step 3: Detect Assignments That Can't Be Extended

Body assignments

aid	eid	e	m	x	y	gaps
b5	eid2	Alice	metal	9	15	-

Head assignments

aid	eid	e	m	z	w	gaps
h1	eid3		metal	17		y ≥ 15, 18 ≤ w, w ≤ x+9

Extension table

body aid	head aid	gaps	deadline
b5	Ø	17 ≥ z, z+1 ≤ w, w ≤ 18 (⇒ z ≤ 17)	17
b5	h1	18 ≤ w, w ≤ 18	18

The deadline for extending a body assignment... can be effectively determined!

c1	Ø	...	12
c1	d1	...	17
c1	d2	...	20
c1	d3	...	14

Algorithm Yields Earliest Detection

Detection Algorithm

When each batch arrives:

1) Generate body and head assignments
2) Match body and head assignments
3) Detect body assignments that can't be extended

Theorem. Given a rule and batches of an enactment, the algorithm reports each violation at the earliest possible time.

Outline

- Motivating early detection
- Events, enactments, and constraints
- Violation detection algorithm
- Experimental evaluation
- Conclusions

Experimental Evaluation

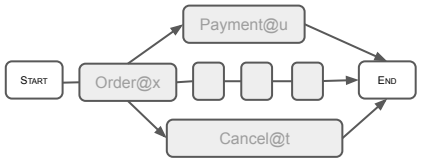
Q1. How beneficial is early detection?

Q2. How costly is early detection overhead?

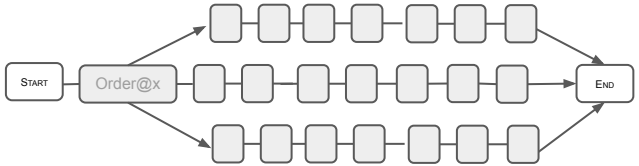
Q3. When is early detection feasibile?

Application dimensions

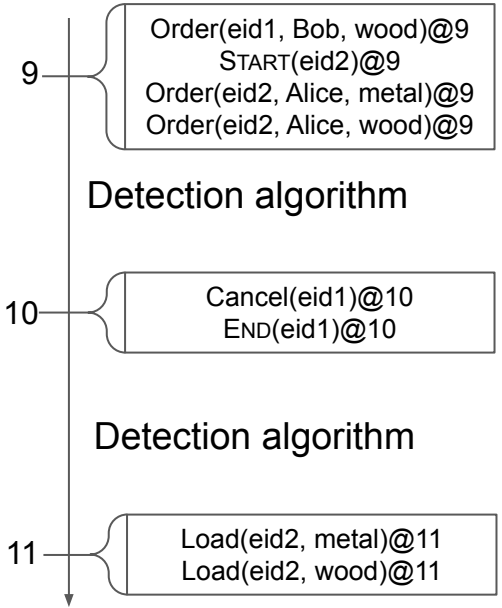
1. Length of enactments



Normal (~10 events)



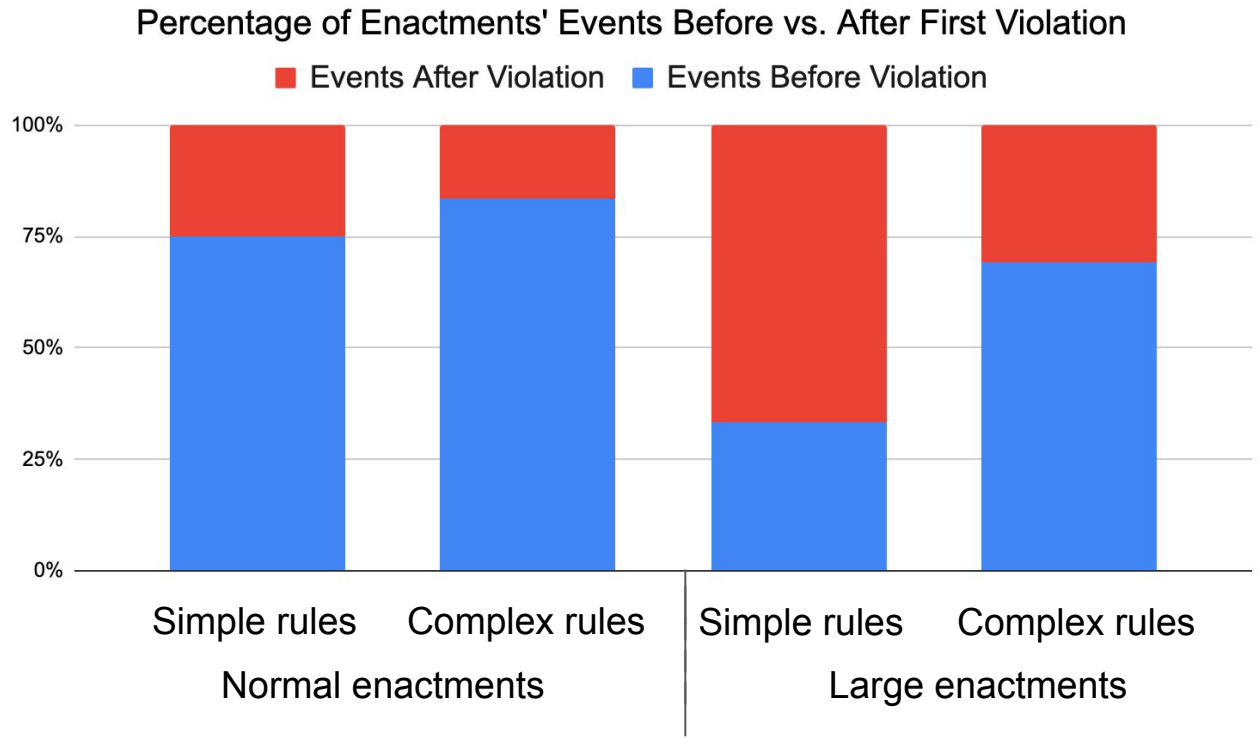
Large (~100 events)



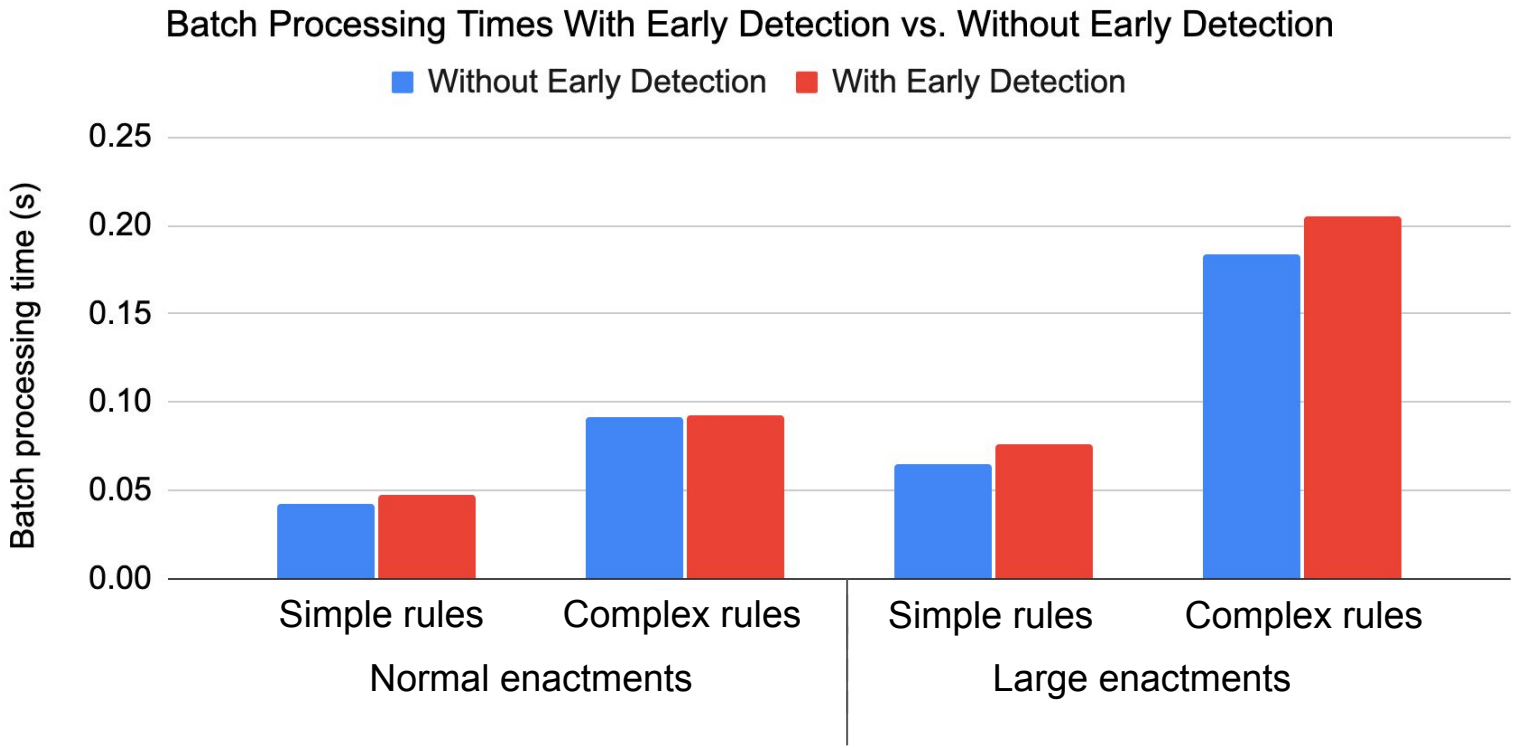
2. Size of rules

- simple (2-4 atoms) vs. complex (4-8 atoms)

~25% of events in normal violating enactments and up to 70% of events in large violating enactments can be prevented.



Early violation detection increases processing time by ≤15%.



Monitoring is feasible for medium-sized applications & a batch interval of 1 second.

	enactment length			
	normal	large	normal	large
batch size	simple rules		complex rules	
100	0.0005	0.0006	0.0008	0.0013
1,000	0.0043	0.0062	0.0075	0.0135
10,000	0.0424	0.0608	0.0749	0.1352

Batch processing times (seconds)

Conclusions

- We study a **violation detection problem** for timed rules on event streams.
  and provide an algorithm for detecting violations **as early as possible**.

Also,

- The algorithm incorporates optimizations (data expiration, semi-niave updates).
- We evaluate the benefits and feasibility of monitoring wrt application dimensions.

Future Work

- Detecting violations for multiple rules
- Monitoring rules with aggregation functions