Managing Dynamic Data

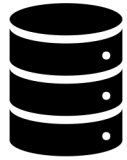
Alessandro Margara

Politecnico di Milano

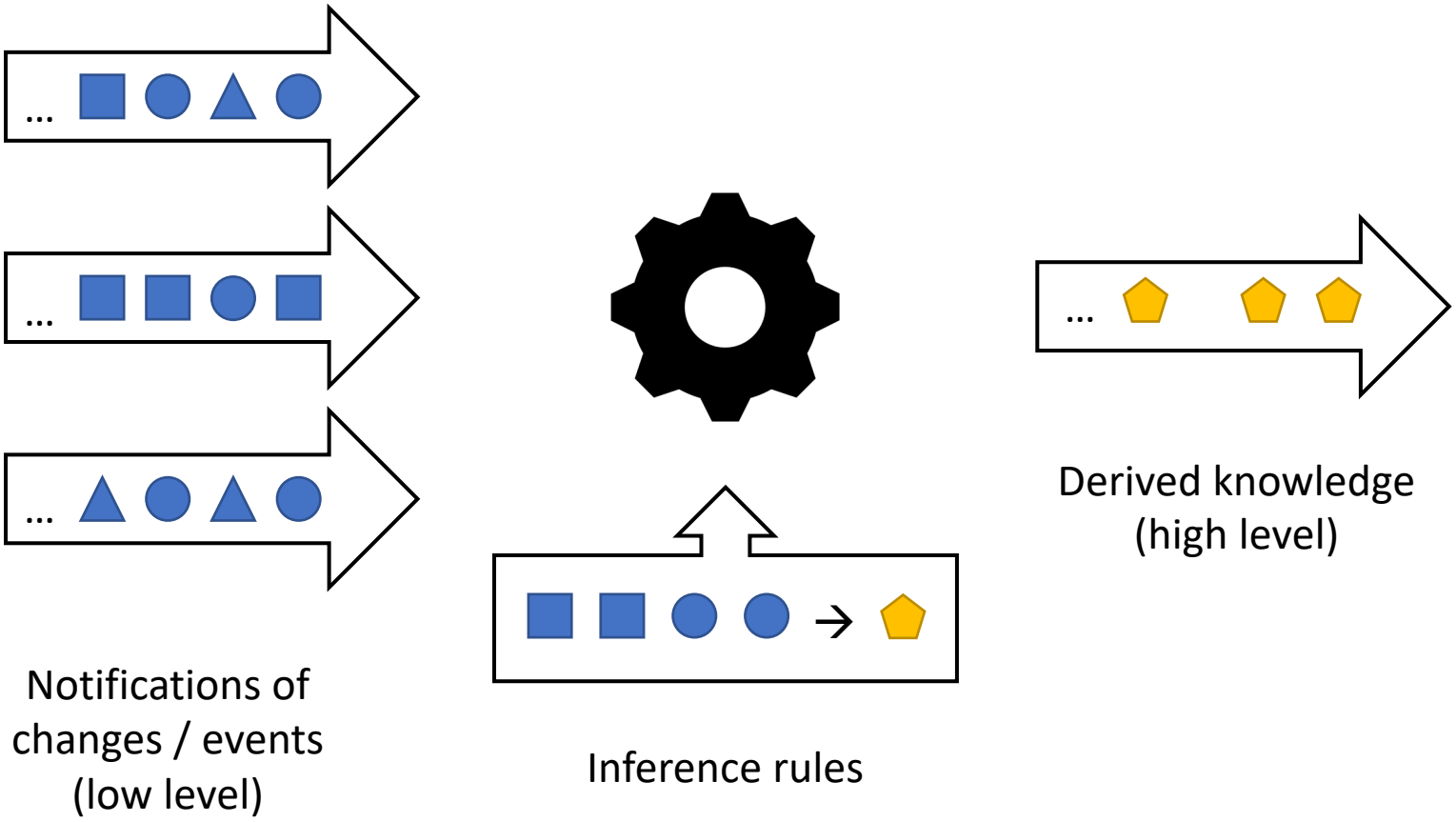
Research focus

1. Analyze dynamic data
   - Timely extract useful knowledge from streams of data
   - E.g., to support decision making

1. Propagate changes across components
   - Update the state of the (distributed) system …
   - … providing guarantees of correctness / consistency



Analyze dynamic data (CEP)



Analyze dynamic data (CEP)

Language/abstractions to define the inference rules

- Open problem
  - Several proposals
  - Different trade-offs between expressivity and complexity of pattern detection
  - Dagstuhl seminar on the foundations of CEP

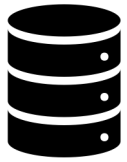
- Contribution [TESLA, DEBS 2010]
  - Language that grounds on temporal logic

Analyze dynamic data (CEP)

Efficient inference algorithm

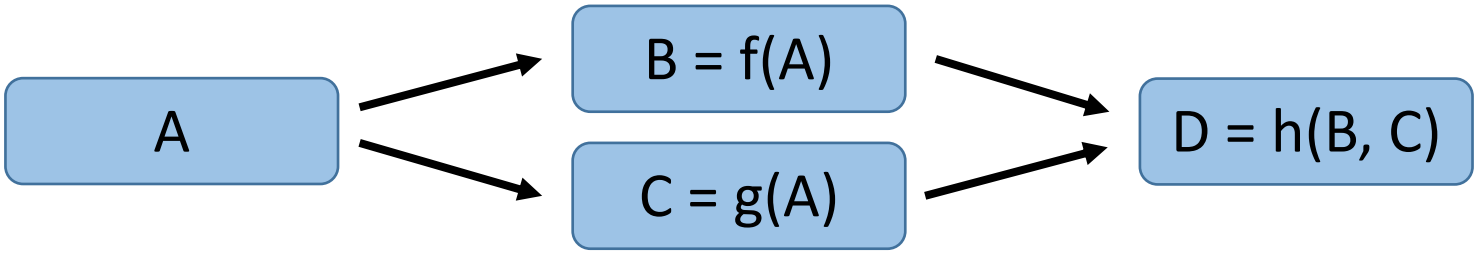
- Contributions [T-REX: JSS 2012, JPDC 2014, TPDS 2014]
  - Event evaluation algorithm that defers complex computations as much as possible
  - Parallel evaluation
    - Within individual rules and across multiple rules
    - Can exploit hardware accelerators (GPUs)
  - Distributed evaluation
    - Adaptive techniques that trade latency for network efficiency

Propagate changes (RP)

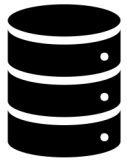


Reactive programming

- Promotes changes to first-class concepts
  - Time-changing variables
  - Automated propagation of changes



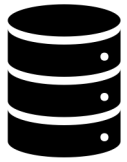
Propagate changes (RP)



Propagation across distributed components

- Contributions [DREAM, DEBS 2014, TSE 2018]
  1. Define the semantics
     - Guarantees related to isolation of concurrent propagations and accesses to time changing variable
  2. Study the cost to ensure the above guarantees
     - In some cases, it requires costly coordination across components
  3. Provide configurable propagation algorithms
     - Different trade-offs between guarantees and cost

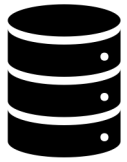
Propagate changes (SP)



- Big Data stream processing systems are becoming the tools of reference to handle dynamic data at scale
  - E.g., Apache Spark Streaming, Apache Flink

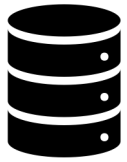
- Designed for distributed processing in cluster environment
  - Dataflow model
    - Data and task parallelism across machines
  - Fault tolerance

Propagate changes (SP)



- Often used as “intelligent communication channels” …
- … that transform streaming data …
  - E.g., cleaning, enriching with background data, …
- … and feed their results into external datastores to make them accessible

Propagate changes (SP)



- Contribution [FlowDB, DEBS 2016]
  - Extend the dataflow model with queryable state
  - Unified system to process dynamic data and store/expose the results of processing
    - Similar to materialized views
    - Constantly and consistently updated when new data becomes available
  - Offers transactional guarantees
    - Customizable scope / boundaries
    - Customizable isolation levels / concurrency control
    - Developers trade guarantees for performance depending on the application requirements

Questions / problems

- Abstractions to analyze / transform data as it gets produced
  - Can serve as a smart communication channel across components
- Abstractions to propagate changes
  - In a homogeneous environment (single programming model: RP / SP)
- How to adapt these ideas to open environments?
  - E.g., microservices