Towards Cascading Reasoning for Generic Edge Processing

Pieter Bonte, Femke Ongenae
Limitations of Cloud Computing for the IoT

- Environmental impact
- High latency
- Network congestion
- Privacy concerns

Streams

Cloud

Edge

Sensor Gateway

Sensor Gateway

Sensor Gateway

Sensor Gateway

CO2
Temperature
Loudness

IoT Devices

Processing

Real-time data

High Velocity
Potential of Edge and Fog Computing for the IoT

1) More data is being produced than can be meaningfully processed

2) Limited resources, heterogeneity, scale and complexity of the infrastructure

3) Data integration problem
Streams / SensORS

Expressive Reasoning

Continuous Information Integration

Mediation

Rewriting

Selection

Annotation

Information Need (IN)

Stream Processing

Devices / Sensors

Increasing complexity of processing

Decreasing volume & velocity data stream

Cascading Reasoning

1) Less complex layers at the edge

2) Split up IN and map different layers to available resources

3) Reasoning capabilities at different levels to solve data integration problem
How can we rewrite queries and optimize the intermediate reasoning steps to offload them to the Edge and filter data early on?
Running Example

What is the status of room X

Retrieve the comfort scores for each room

Building manager
Running Example

{ "sensorID": "1234", "observationID": "5674", "value": 765 }
Running Example

Cloud

- Processing
- Annotation

Edge

- Sensor Gateway
- Sensor Gateway
- Sensor Gateway

Streams

CO2

Temperature

Loudness

Devices / Sensors

mappings:
sensor:
sources:
s: iot:$(observationID)
po:
- [a, ssn:Observation]
- [ssn:madeBySensor, iot:$(sensorID)]
- [sosa:hasSimpleResult, $(value)]
... :sensorX a sosa:Sensor;  
sosa:observes :temp, :loudness, :co2, humidity;  
:s hasLocation :officeY.  
:officeY :connectedTo :officeZ, officeQ;  
:hasName "200.009"^^xsd:string.  
...
Running Example

Cloud

Processing

Annotation

Edge

Sensor Gateway

Sensor Gateway

Sensor Gateway

CO2

Temperature

Loudness

Devices / Sensors

Rules

R1: {?x a :TempObservation} => {?x a :ComfortObservation}
R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}
R3: {?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor}
    => {?x a :TempObservation}
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor}
    => {?x a :LoudnessObservation}
    => {?s a :TempSensor}
    => {?s a :LoudnessSensor}
Q1:
?obs a Observation; // stream
  hasSimpleResult ?value; // stream
  madeBySensor ?sensor. // stream
?sensor hasLocation ?loc. // static
?loc hasName "200.009" // static

Q2:
?obs a ComfortObservation; // stream + domain knowledge
  hasSimpleResult ?value. // stream
How do we get there?

a) Cloud paradigm

Cloud

Processing

Annotation

Queries

Rules

Static data

YARRML mapping files

Edge

JSON data

Sensor Gateway

Sensor Gateway

Sensor Gateway

Streams

High Velocity

CO2
Temperature
Loudness

b) Edge paradigm

Processing

Annotation

Sensor Gateway

Sensor Gateway

Sensor Gateway

Rewritten queries/rules

YARRML mapping files

VoCaLS stream description

Devices / Sensors

CO2
Temperature
Loudness
How do we get there?

Information Need (Queries)

Domain knowledge

Stream Annotation
Shape Extraction

```
{?x a NodeShape. ?x targetClass ?cls} => {_:t a ?cls}
{?x a NodeShape. ?x sh:property [ sh:path ?prop]} => {_:s ?prop _:o}
...
```

mappings:
sensor:
sources:
s: iot:$(observationID)
po:
- [a, ssn:Observation]
- [ssn:madeBySensor, iot:$(sensorID)]
- [sosa:hasSimpleResult, $(value)]
Query Rewriting

Rewritten Query:

1. ?obs a Observation;
2. hasSimpleResult ?value;
3. madeBySensor ?sensor.
4. ?sensor hasLocation ?loc.
5. ?loc hasName "200.009"

Event BluePrint

Static data

Rewritten Query:

1. ?obs a Observation; // stream
2. hasSimpleResult ?value; // stream
3. madeBySensor :sensorX. // stream

<table>
<thead>
<tr>
<th>?obs</th>
<th>?value</th>
<th>?sensor</th>
<th>?loc</th>
</tr>
</thead>
<tbody>
<tr>
<td>_t</td>
<td>_z</td>
<td>:sensorX</td>
<td>:officeY</td>
</tr>
<tr>
<td>origin</td>
<td>Event</td>
<td>Event</td>
<td>Event/Static</td>
</tr>
</tbody>
</table>
Reasoning-enabled Rewriting

Query

\(?obs \ a :\)ComfortObservation;
  :hasSimpleResult \(?val\).

Rules

R1: \{?x \ a :\)TempObservation\} \implies \{?x \ a :\)ComfortObservation\}
R2: \{?x \ a :\)LoudnessObservation\} \implies \{?x \ a :\)ComfortObservation\}

R3: \{?x \ a :\)Observation, \?x :\)madeBySensor \s, \?s \ a :\)TempSensor\} \implies \{?x \ a :\)TempObservation\}
R4: \{?x \ a :\)Observation, \?x :\)madeBySensor \s, \?s \ a :\)LoudnessSensor\} \implies \{?x \ a :\)LoudnessObservation\}
R5: \{?s \ a :\)Sensor, \?s :\)observes \p, \?p \ a :\)Temperature\} \implies \{?s \ a :\)TempSensor\}
R6: \{?s \ a :\)Sensor, \?s :\)observes \p, \?p \ a :\)Loudness\} \implies \{?s \ a :\)LoudnessSensor\}
Reasoning-enabled Rewriting

Query

\[ ?\text{obs} \text{ a :ComfortObservation;} \]
\[ :\text{hasSimpleResult} \ ?\text{val}. \]

Inferred:

:sensorX a :TempSensor
:obsX a :TempObservation
:obsX a :ComfortObservation

Real Example Event

Static data

R3: \{?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor\} => {?x a :TempObservation}
R1: \{?x a :TempObservation\} => {?x a :ComfortObservation}
Reasoning-enabled Rewriting

Query

?obs a :ComfortObservation;
:hasSimpleResult ?val.

R1: {?x a :TempObservation} => {?x a :ComfortObservation}
R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}

R3: {?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor}
  => {?x a :TempObservation}
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor}
  => {?x a :LoudnessObservation}
**Reasoning-enabled Rewriting**

**Query**

```
?obs a :ComfortObservation; :hasSimpleResult ?val.
```

R1: `{?x a :TempObservation} => {?x a :ComfortObservation}`
R2: `{?x a :LoudnessObservation} => {?x a :ComfortObservation}`
R3: `{?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor} => {?x a :TempObservation}`
R4: `{?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor} => {?x a :LoudnessObservation}`
Reasoning-enabled Rewriting

Query

?obs a :ComfortObservation; :hasSimpleResult ?val.

R1: {?x a :TempObservation} => {?x a :ComfortObservation}
R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}
R3: {?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor} => {?x a :TempObservation}
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor} => {?x a :LoudnessObservation}

Event BluePrint

Static data

<table>
<thead>
<tr>
<th></th>
<th>?s</th>
<th>?p</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>:sensorX</td>
<td>:propY</td>
</tr>
<tr>
<td>origin</td>
<td>Static</td>
<td>Static</td>
</tr>
<tr>
<td>head</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

20
Reasoning-enabled Rewriting

Query

?obs a :ComfortObservation;
:hasSimpleResult ?val.

R1: {?x a :TempObservation} \Rightarrow {?x a :ComfortObservation}
R2: {?x a :LoudnessObservation} \Rightarrow {?x a :ComfortObservation}
R3': {?x a :Observation, ?x :madeBySensor :sensorX} 
\Rightarrow {?x a :TempObservation}
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor} 
\Rightarrow {?x a :LoudnessObservation}

Event BluePrint

Static data

<table>
<thead>
<tr>
<th>?x</th>
<th>value</th>
<th>origin</th>
<th>head</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_:t</td>
<td>Stream</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Reasoning-enabled Rewriting

Query

?obs a :ComfortObservation; :hasSimpleResult ?val.

Identified Rules

R1: {?x a :TempObservation} => {?x a :ComfortObservation}
R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}

R3': {?x a :Observation, ?x :madeBySensor :sensorX} => {?x a :TempObservation}
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor} => {?x a :LoudnessObservation}
Reasoning-enabled Rewriting

Query

\[ \text{obs a :ComfortObservation; :hasSimpleResult ?val.} \]

Identified Rules

R1: \{?x a :TempObservation\} \implies \{?x a :ComfortObservation\}

R3’: \{?x a :Observation, ?x :madeBySensor :sensorX\} \implies \{?x a :TempObservation\}

Prune hierarchies

R3’’: \{?x a :Observation, ?x :madeBySensor :sensorX\} \implies \{?x a :ComfortObservation\}
Reasoning-enabled Rewriting

Query

?obs a :ComfortObservation;
   :hasSimpleResult ?val.

Rewritten Query

?obs a :Observation;
   :madeBySensor :sensorX;
   :hasSimpleResult ?val.

Optimized Rules

R3'': {?x a :Observation, ?x :madeBySensor :sensorX} => {?x a :ComfortObservation}
Implementation

RoXi

Come check out the poster during the poster & demo session!

SHAROQ Github

RoXi Github
Evaluation

Static Data Increase

- cloud
- edge
Evaluation

Sensor Observations Increase

- **cloud**
- **edge**

![Graph showing time (nano seconds) vs. #observations for cloud and edge sensor observations.](Image)
Evaluation
Conclusion

Top-down and bottom-up optimization to rewrite rules/queries in order to enable efficient and privacy-aware edge analytics

a) Cloud paradigm
- Processing
- Annotation
- Queries
- Rules
- Static data
- YARRML mapping files

b) Edge paradigm
- Processing
- Annotation
- Sensor Gateway
- Sensor Gateway
- Sensor Gateway
- Rewritten queries/rules
- YARRML mapping files

Streams:
- CO2
- Temperature
- Loudness

Devices / Sensors:
- CO2
- Temperature
- Loudness

VoCaLS stream description

Pieter.Bonte@UGent.be