I²: Interactive Real-Time Visualization for Streaming Data

Jonas Traub jonas.traub@tu-berlin.de
Nikolaas Steenbergen nikolaas.steenbergen@dfki.de
Philipp Grulich philipp.grulich@dfki.de
Tilmann Rabl rabl@tu-berlin.de
Volker Markl volker.markl@tu-berlin.de

I² - Two Types of Interactivity

Interactive Development

Change your program and deploy your updates with just one click. Develop real-time data visualizations while operating on live data.

Adaptive Data Stream Processing Pipelines in Apache Flink .

Development and Live-Visualization in Apache Zeppelin Notebooks.

Flink Job:

One click deployment

```
%flink
debsDataStream.soccerFieldAggregation()
              .addSink(i2SoccerField)
```

Visualization code:

Update ui element

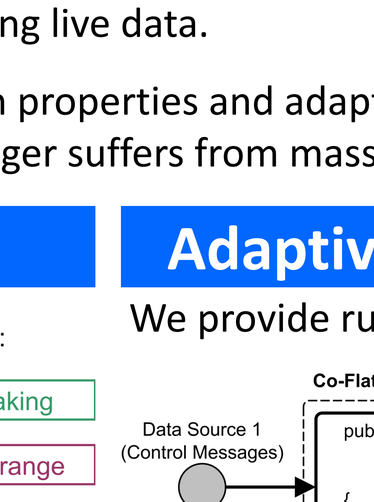
```
%angular
<script>
zeppelin.addComponent(
    "i2SoccerField",
    "<template>",
    function(scope, element) {
        ...
        function updatePoints(event) {
            ...
        }
        function getColor(event){
            if(isBallEvent(event))
                return "white"
            else
                return "black"
        }
        scope.$watch('events', function(newEvent) {
            updatePoints(newEvent);
        });
    })
</script>
```

Interactive Visualization

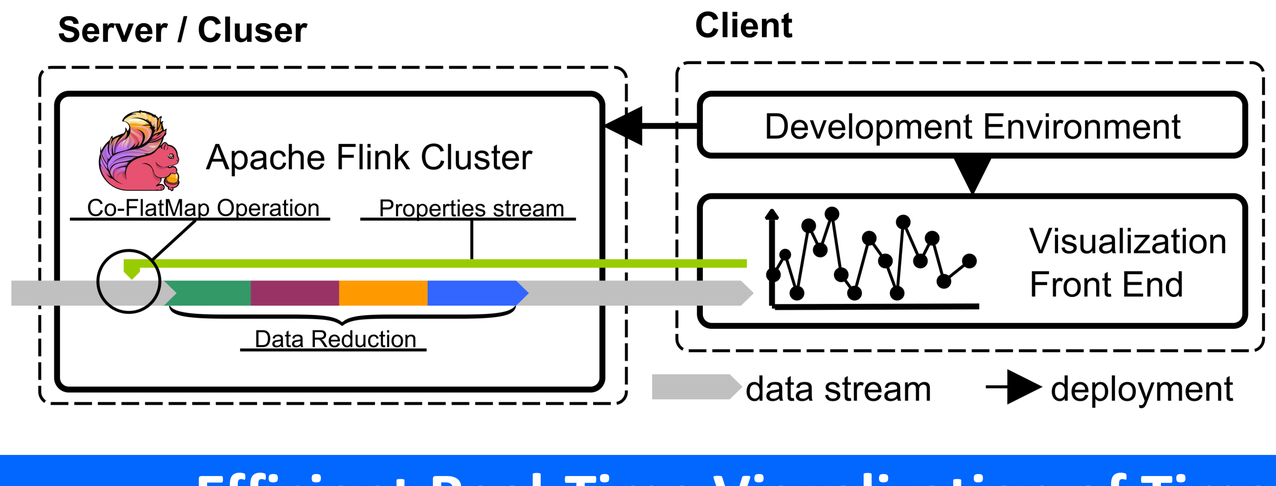
Explore live data in visualizations. The underlying cluster job adapts at runtime to your settings and sends the required data to the dashboard.

Example Dashboard:

- Sensor data from a football match.
- Adaptive Flink job.
- Interactivity:
  - player selection.
  - different metrics.
  - range of the depicted history.



Architecture Overview

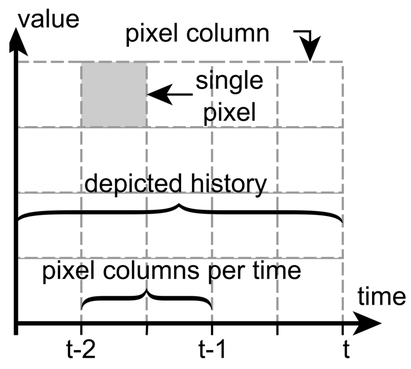


I² seamlessly connects live data visualization with the development of analysis pipelines for streaming data.

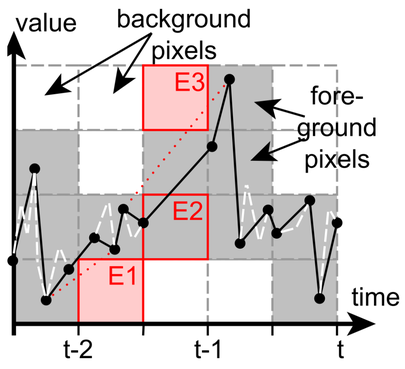
1. Develop stream analysis pipelines and visualizations.
2. Deploy your code with just one click.
3. Discover the incoming live data.

I² observes visualization properties and adapts the Flink job at runtime. The visualization no longer suffers from massive ingestion rates.

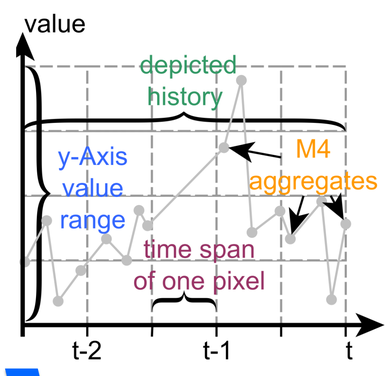
Efficient Real-Time Visualization of Time Series Data

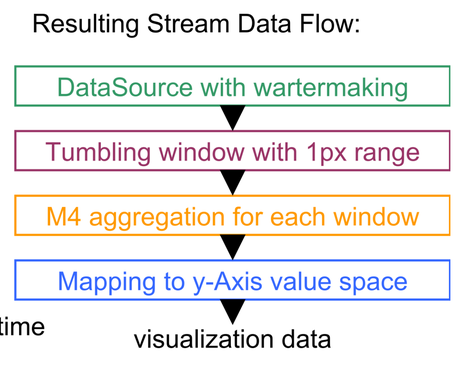


1. There is a trade off between the length of the depicted history and visualization precision (pixel columns per time).



2. We need exactly four data points per pixel column to provide a loss-free plot of time series data. [M4, Jugel et al., VLDB'14]

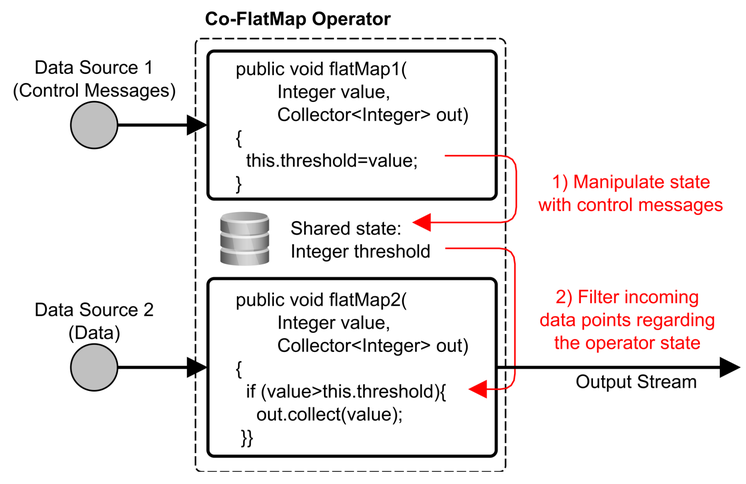




- Transfer four values per pixel column.
- Constant workload at the front end.
- The front end is independent from the ingestion rate at the Flink cluster.

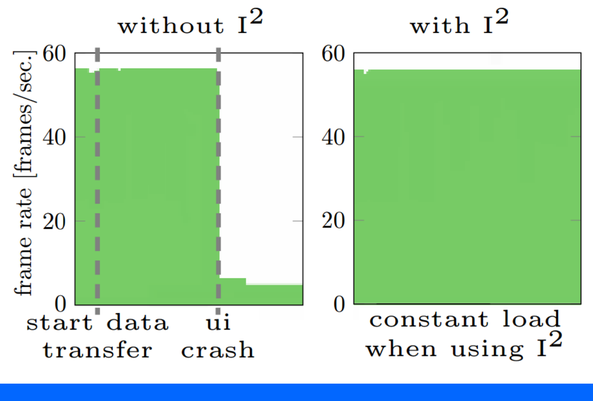
Adaptive Flink Operators

We provide runtime adaptive operators.



Example: A runtime adaptive filter operator for variable thresholds.

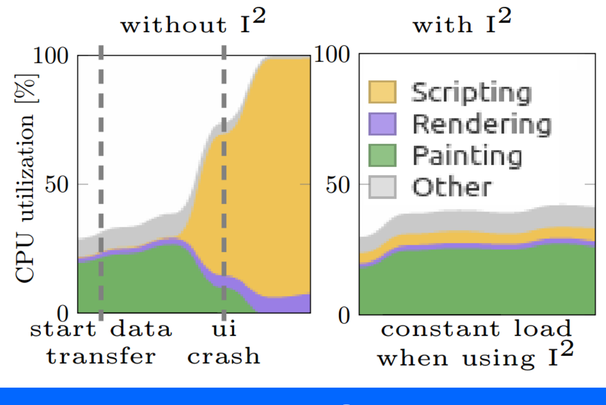
Performance Evaluation



Frame Rate

Without I²:
Unresponsive dashboard shortly after start-up (CPU overload).

With I²:
Constant 60Hz frame rate.



CPU Utilization

Without I²:
CPU cannot keep up with the massive ingestion rates.

With I²:
Reduced and constant CPU load.

Try it! - It's all open source!

Apache Flink – flink.apache.org

Apache Zeppelin – zeppelin.apache.org

I2 github.com/TU-Berlin-DIMA/i2

hub.docker.com/r/tuberlindima/i2

Technische Universität Berlin (dima.tu-berlin.de)
German Research Center for Artificial Intelligence (dfki.de)

STREAMLINE.
This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 688191.
Bundesministerium für Bildung und Forschung
Software Campus (01IS12056) & BBDC (01IS14013A)