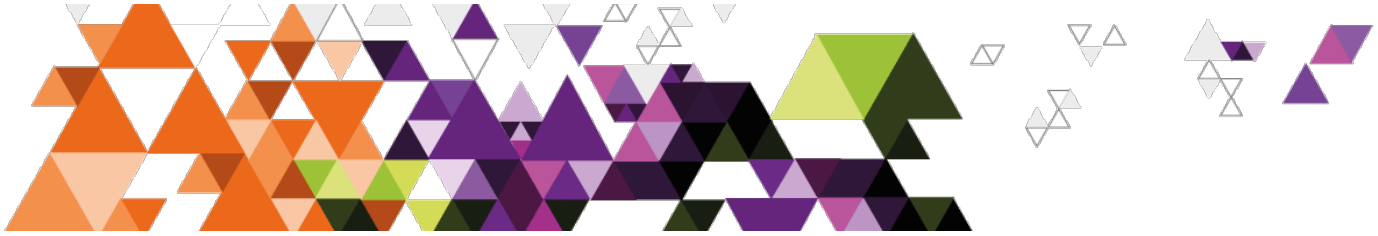


NorCom

Unboxing Engineering Data.

Time Series Analysis with Spark in the Automotive R&D Process

Til Piffl (tpf@norcom.de)
Miha Pelko (@mpelko)

NorCom IT AG, Munich, Germany
www.norcom.de

NorCom IT AG - Facts & Figures

Numbers	• Established 1989 • IPO: 1999 • Turnover 16,5 Mio. € • about 130 Employees
Location	• München • Nürnberg • San Jose
Customer	• Automotive • Public (German) • Media • Finance

Consulting: Design | Build | Run — NorCom EXPERTS

NDOS
(NorCom Data Operating System)

INORMATION MANAGEMENT

News (Broadcast)

Video Management

Enterprise Communication

BIG DATA

Hortonworks
cloudera
MAPR
elasticsearch.

BIG INFRASTRUCTURE

Hewlett Packard Enterprise
ORACLE FUSION MIDDLEWARE
APPLICATION DEVELOPMENT FRAMEWORK
Microsoft

Where is Big Data in Automotive?

NorCom

- Development
  - Few development locations worldwide
  - Some test vehicles (<100)
  - Raw sensor data (camera, radar, lidar, …)
  - Algorithm development (Autonomous driving)
- Testing Phase
  - Many locations worldwide
  - Lots of test vehicles
  - Compressed Data (Video)
  - Verification
- Field
  - All around the world (with many regulators)
  - Hundreds of thousands of connected cars
  - Triggered Data
  - Predictive maintainance

Next Generation

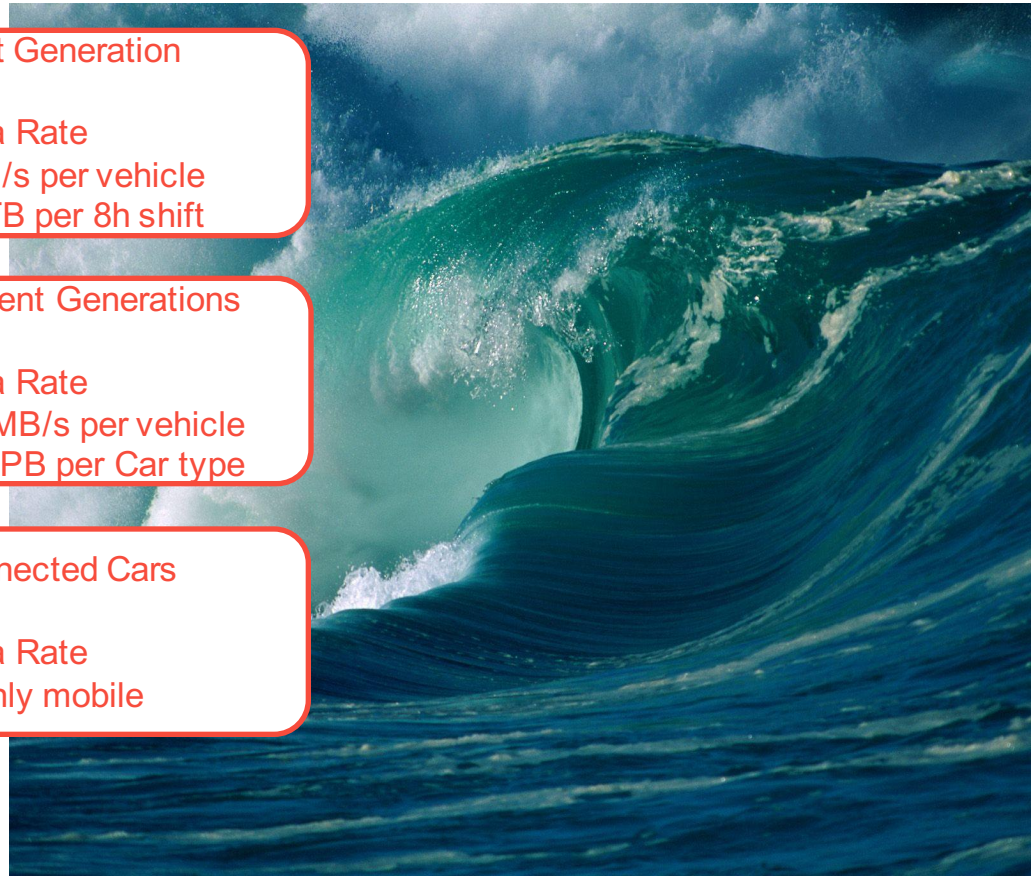
Data Rate
2GB/s per vehicle
60 TB per 8h shift

Current Generations

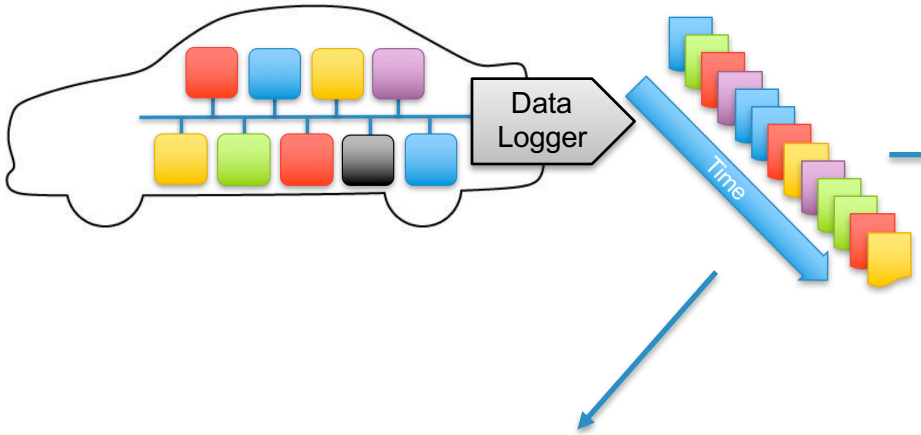
Data Rate
350MB/s per vehicle
~10 PB per Car type

Connected Cars

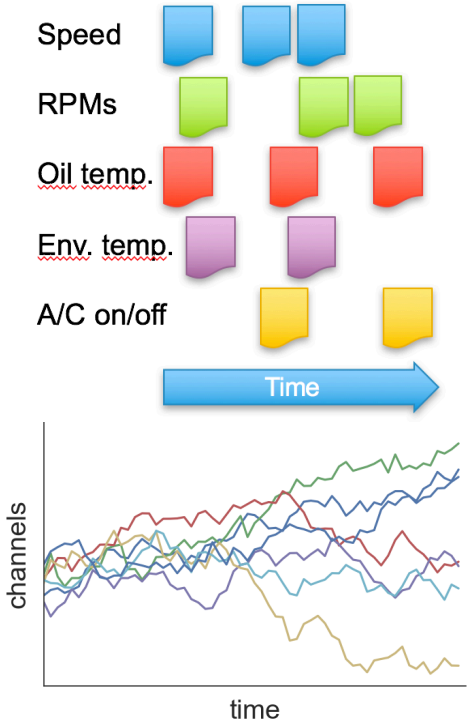
Data Rate
mainly mobile



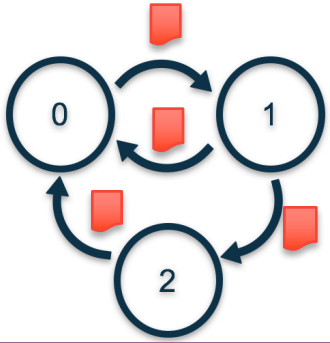
Automotive time series analysis requires parallel processing

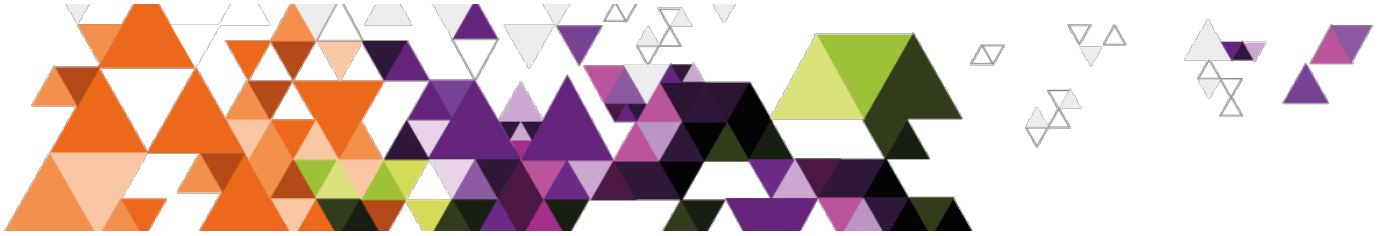


Part 1:
A Spark-API for Multi-Sensor Time Series



Part 2:
State machine analysis with Spark





NorCom

Unboxing Engineering Data.

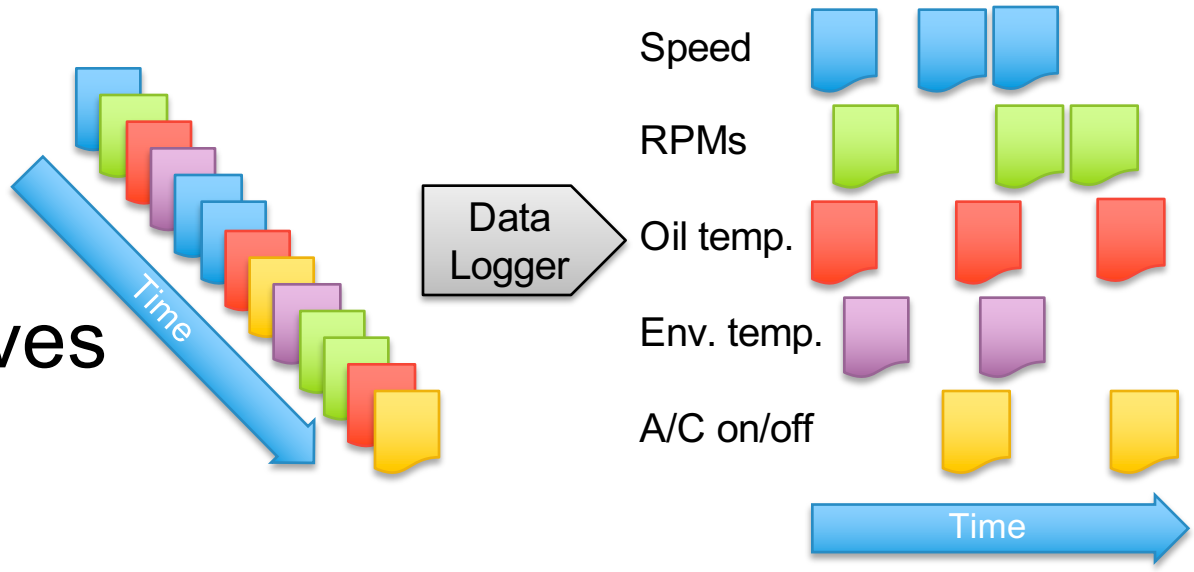
DaSense: A Spark-API for Multi-Sensor Time Series

SPARK SUMMIT EUROPE 2016

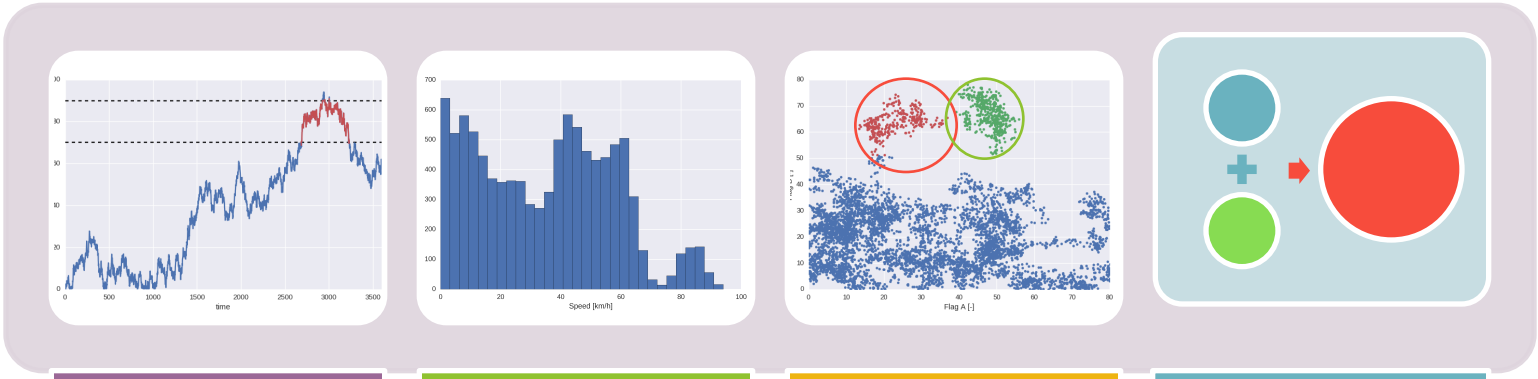
NorCom Information Technology AG

Multi-sensor time series

- Bus communication is filtered and sorted
- Thousands of signal types
- Time series with millions of entries
- Hundreds of measurement drives



Typical tasks



Criteria tests & Pattern search

Aggregation & Reporting

Classification & Machine Learning

Correlations & Root-Cause Analysis

Time series API

Python-based

Reduces complexity by focusing on time series

Preserves lazy evaluation

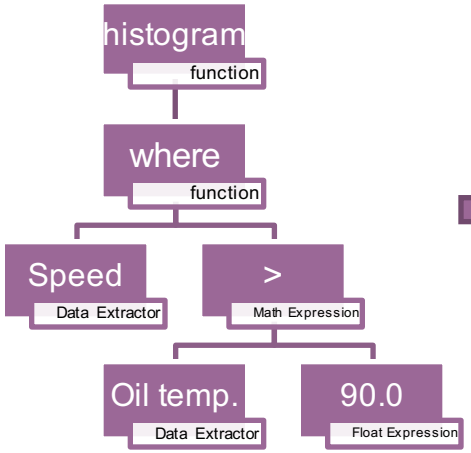
Important concepts:

Expressions

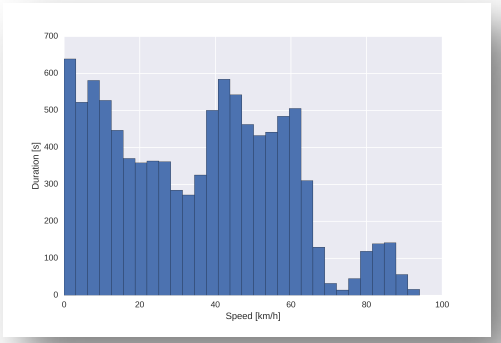
Data Extractors

Concepts - expressions

```
context = Context()
all_data = ChannelData("/mapr/norcom_cluster/test_data/test_drive.parquet", context)

OilTemp = ChannelExtractor('t_oil', unit='deg C')
Speed   = ChannelExtractor('Speed', unit='km/h')

hist_expression = Speed.where(OilTemp > 90.0).histogram(bins=100)

result = all_data.evaluate({'Speed_histogram' : hist_expression})
```

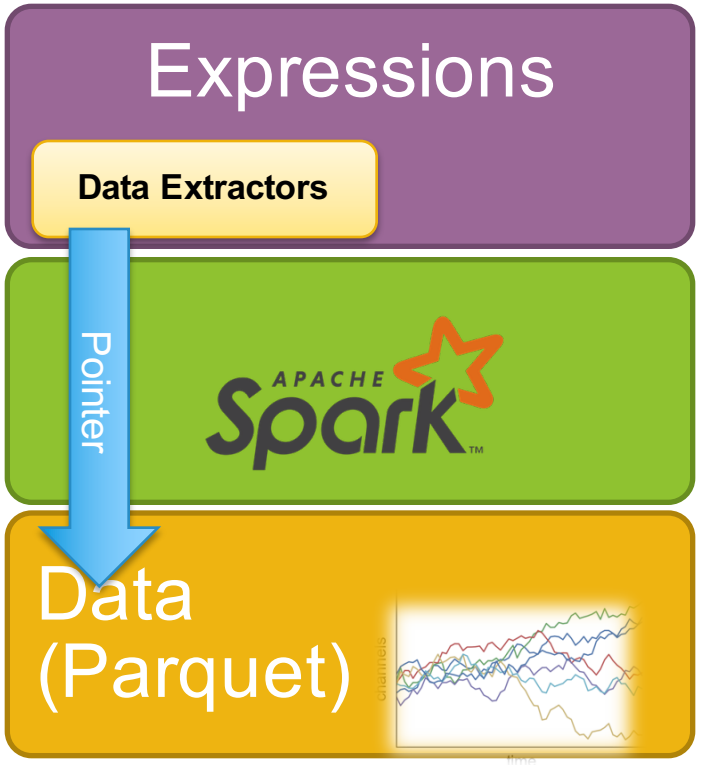


Data

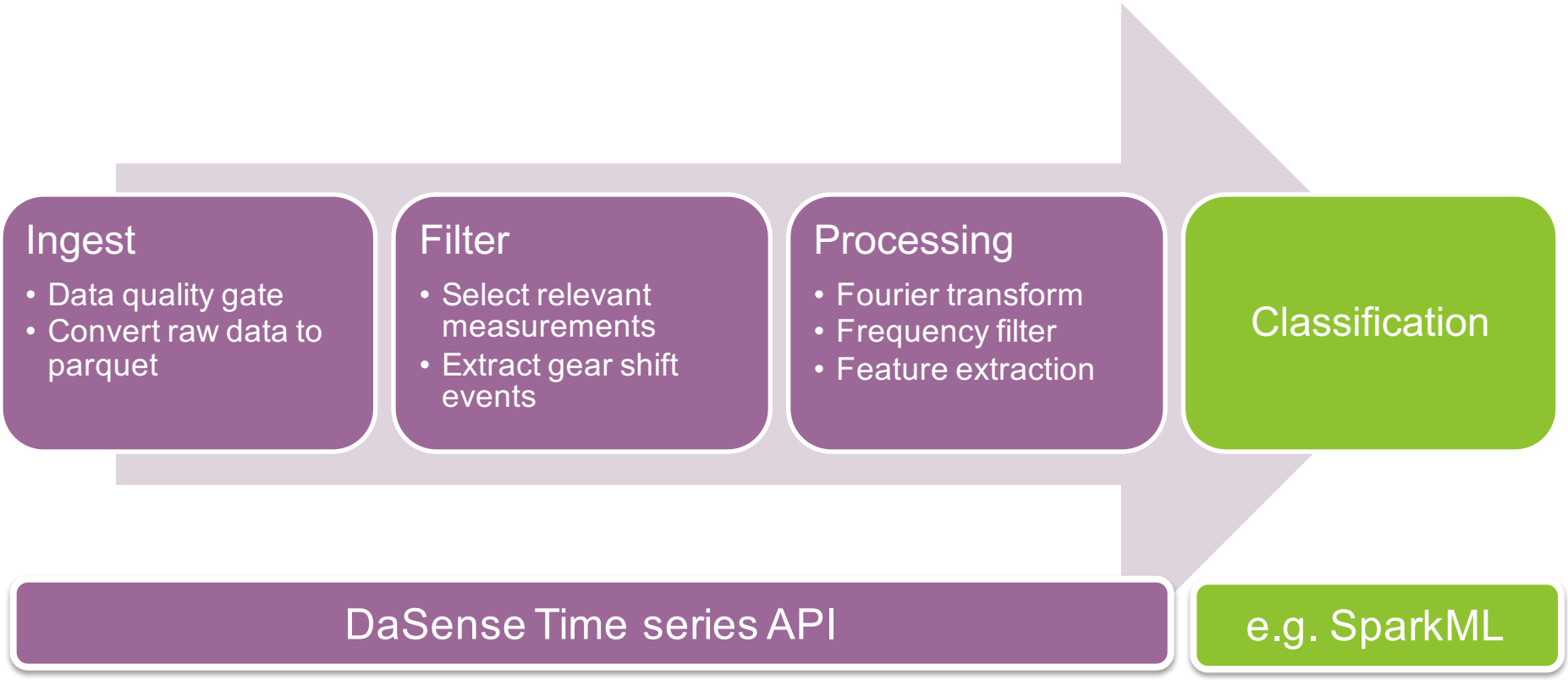


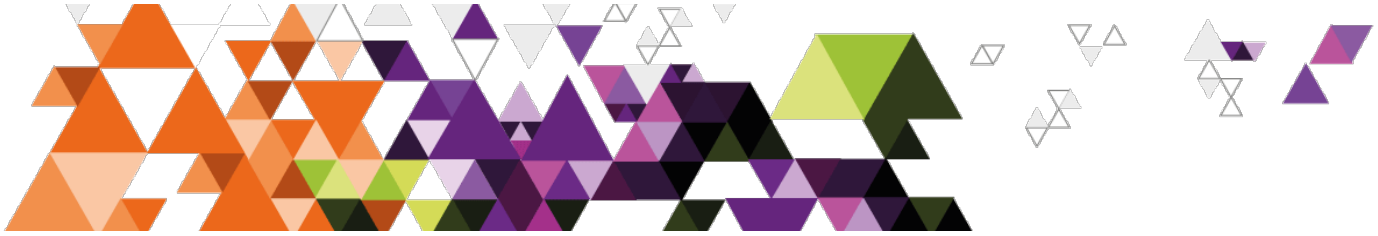
Concepts – data extractors

- Basic time series expression
- Interface to actual data
- Handles
  - channel name aliases (Speed or VehV_v?)
  - units and conversions (mph to km/h)
  - interpolation requirements (linear, zero-order,…)



Workflow example





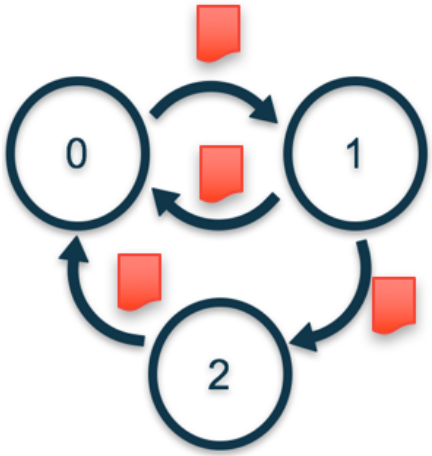
NorCom

Unboxing Engineering Data.

Parallelization of a State Machine

State machines in automotive industry

States and transitions



Examples of states:

- Engine on / off / ready to start
- Current Gear
- States on the communication bus

Example of analytical use-case:
Analyze / Validate the communication protocol from the logs.

Need for parallel Big Data solutions

Current approach – sequential:

- Sequential replay of messages used for analysis
- No way of scaling within the single log

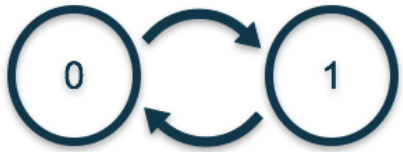
Desired approach – parallel:

- Split the log in partitions and analyze in parallel
- Enables scaling within the single log

The density of messages is increasing

What is the status at the beginning of the partition?

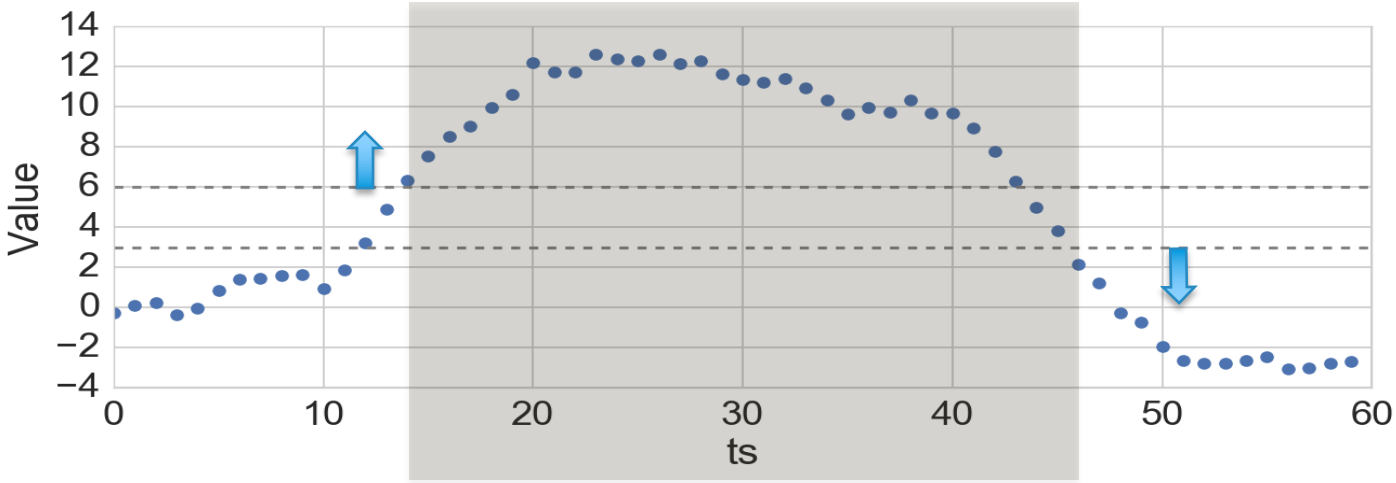
Various encodings of state machine transitions



Explicitly in a message

ts	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
state	0	0	1	1	1	0	0	1	1

Implicitly via message value

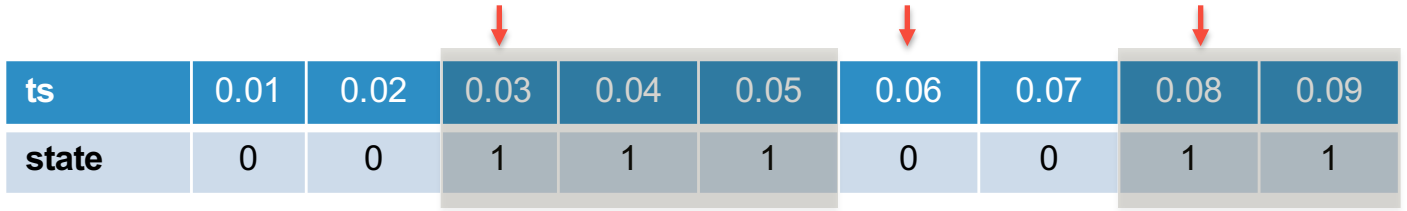


Implicitly via message timing

ts	0.01	0.02	0.03	0.07	0.08	0.09	0.15	0.16	0.17

Basic parallel solution

Original log



ts	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
state	0	0	1	1	1	0	0	1	1

Basic parallel solution

Original log

ts	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
state	0	0	1	1	1	0	0	1	1

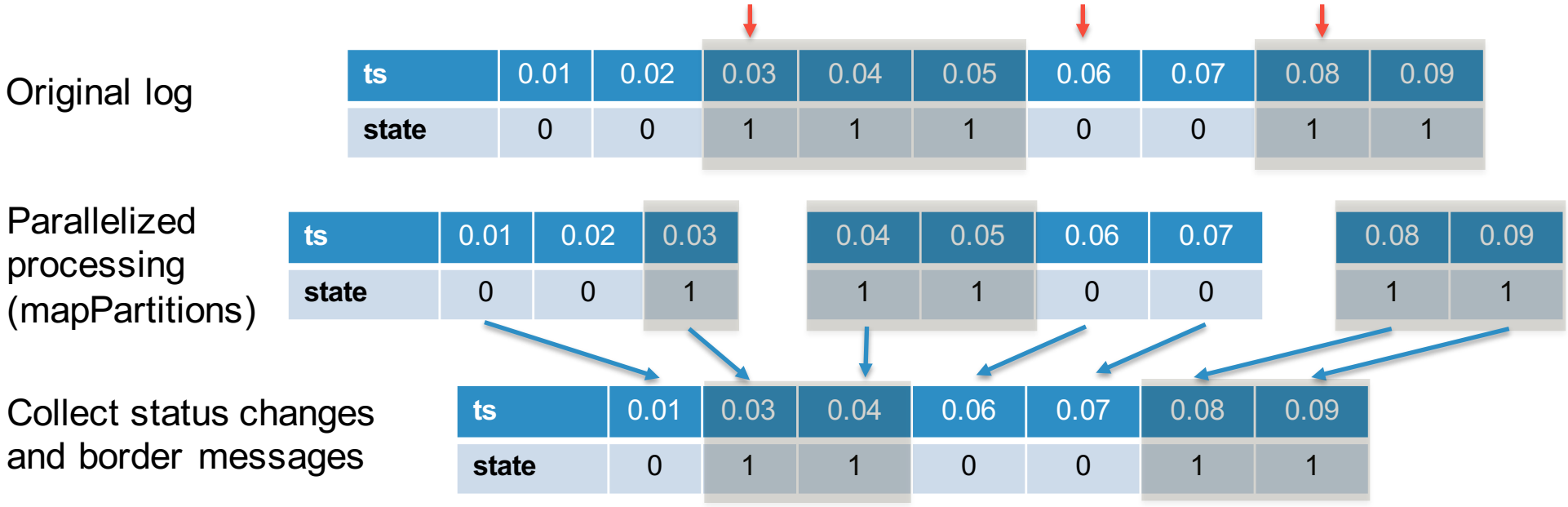
Parallelized processing (mapPartitions)

ts	0.01	0.02	0.03
state	0	0	1

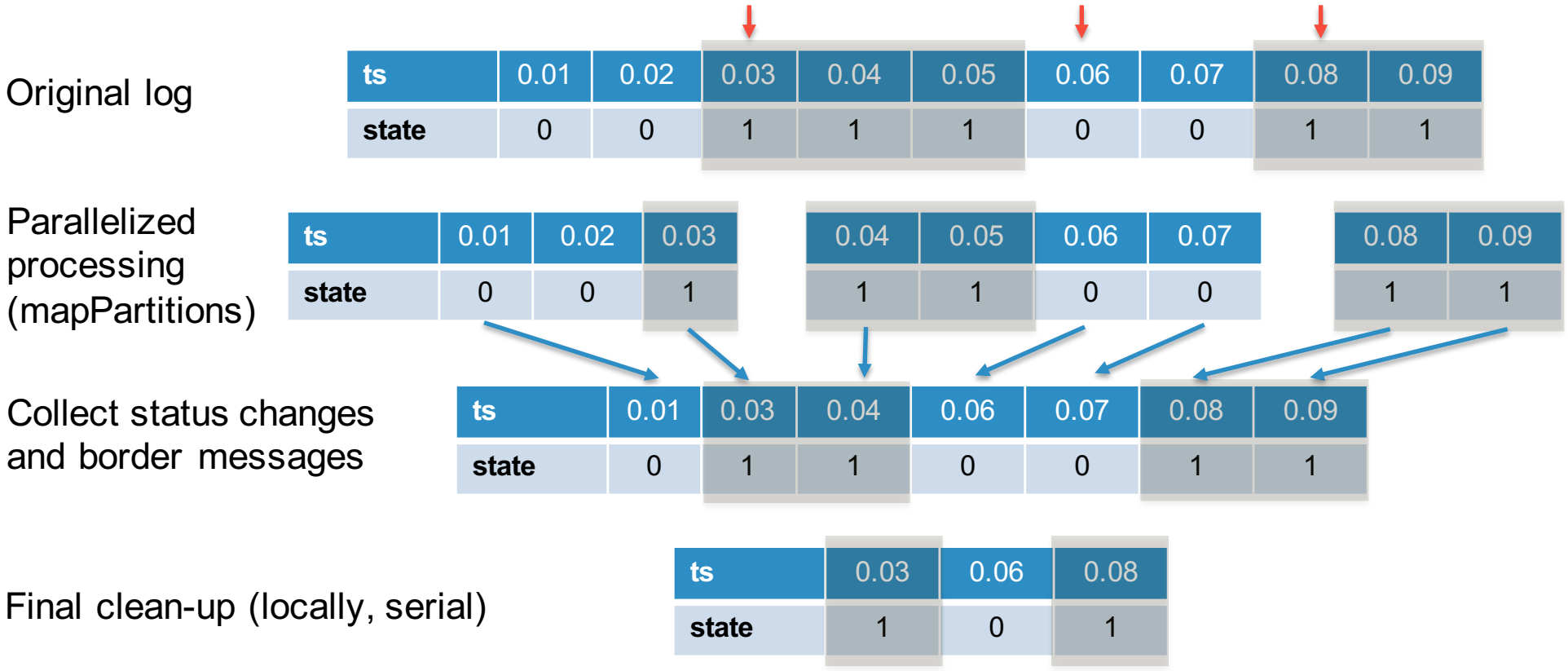
0.04	0.05	0.06	0.07
1	1	0	0

0.08	0.09
1	1

Basic parallel solution



Basic parallel solution



Alternatives

NorCom

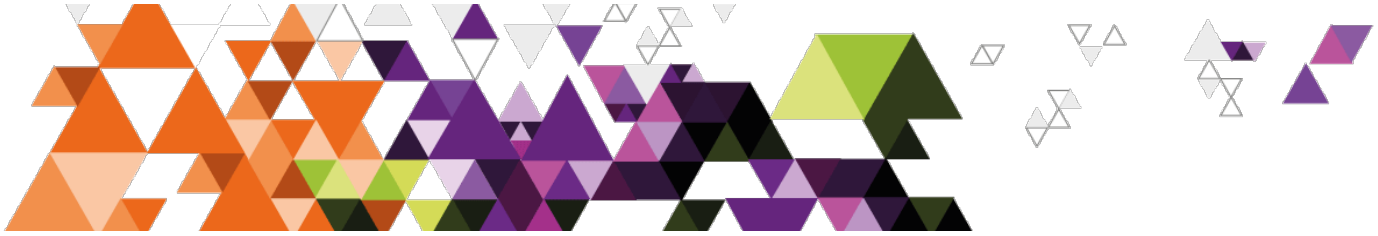
- Use windowing functions
  - Window size unknown, could span the full time series
- "Broadcast" the borders from neighboring partitions (mapPartition → groupByKey)
  - groupByKey expensive, does not generalize well
- mapPartitionWithIndex → reduceByKey
  - Needs complex data structure to handle associativity and commutativity requirement

In our experience

NorCom

keep it simple.

Sample code available at:
http://github.com/dasense/state_machine_analysis_with_spark



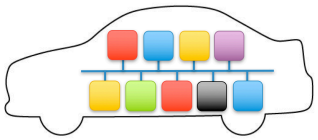
NorCom

Unboxing Engineering Data.

Summary

Summary

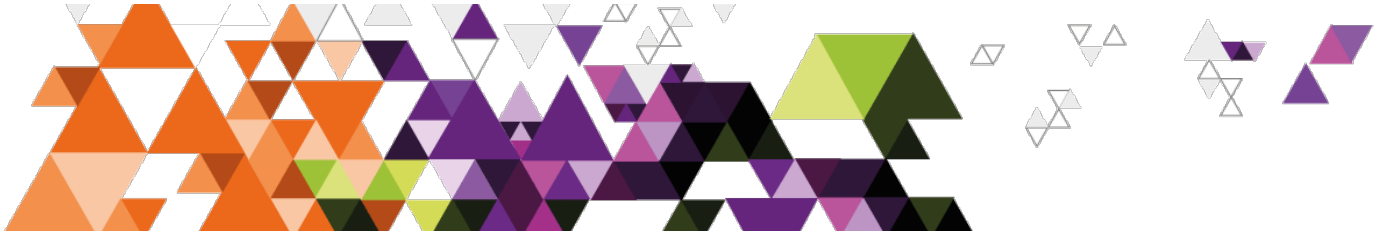
Automotive Industry is a major data producer



Data & problems are somewhat specific, but fun!

DaSense

We are bringing Spark into production in Automotive R&D



NorCom

Unboxing Engineering Data.

THANK YOU.

We are hiring!

Til Piffl (tpf@norcom.de)
Miha Pelko (@mpelko)

NorCom IT AG, Munich, Germany
www.norcom.de

SPARK SUMMIT EUROPE 2016