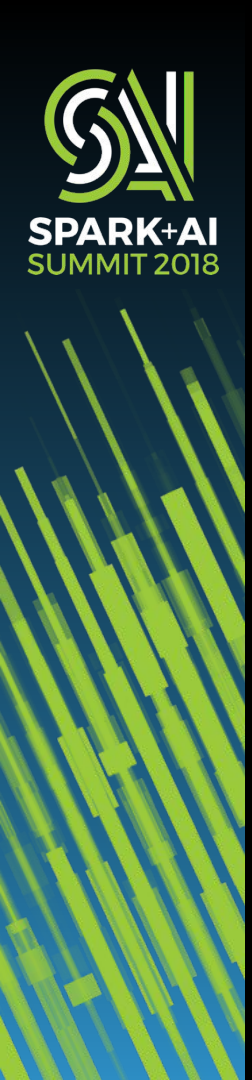



# Large-Scaled Telematics Analytics in Apache Spark

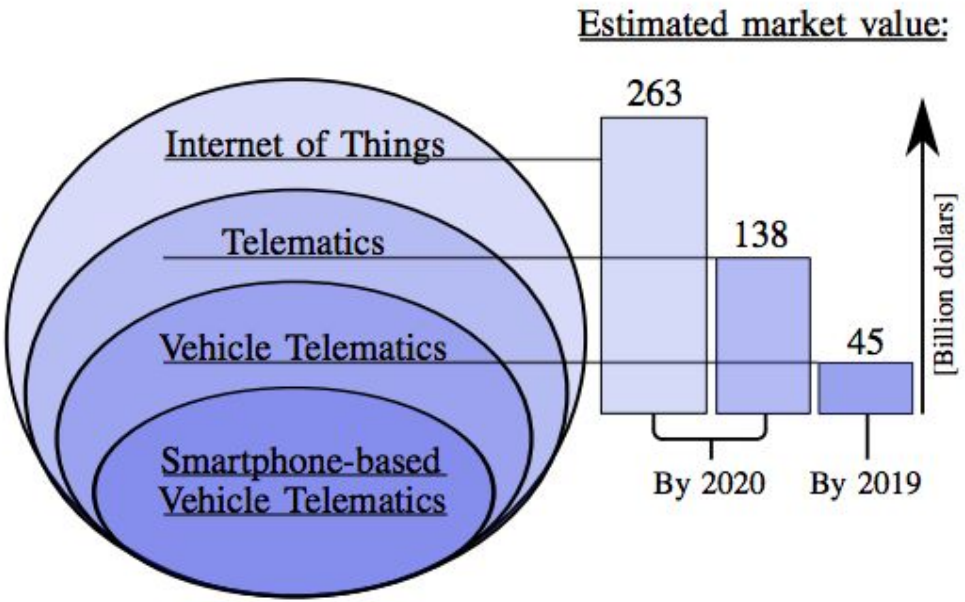
Wayne Zhang, Uber
Neil Parker, Uber

#DS3SAIS

# Agenda

- Telematics introduction
- Eng pipeline

# Telematics



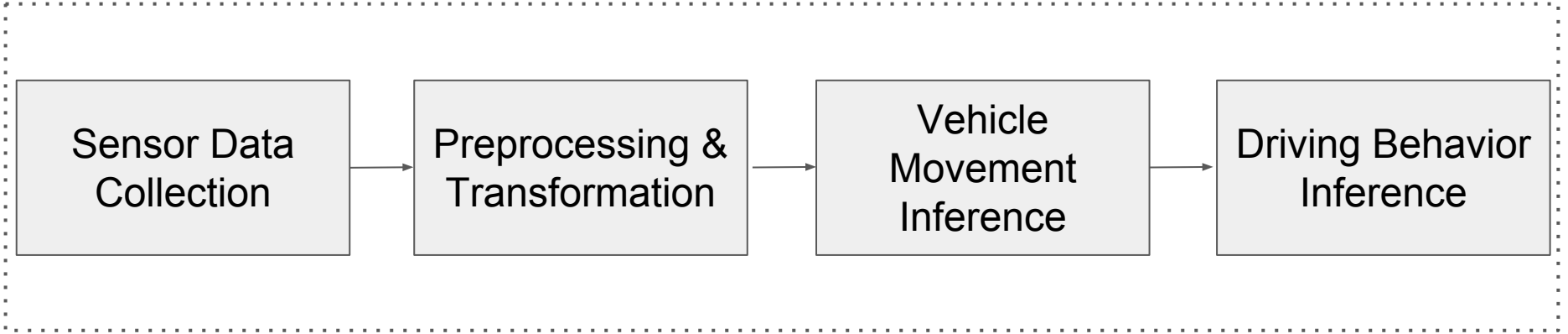


- Wide availability
- Cheap
- Short upgrade cycle

- Lower quality
- Measure phone motion

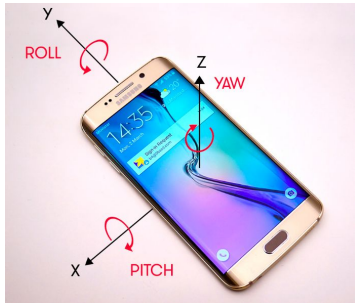
Source: [Smartphone-based Vehicle Telematics - A Ten-Year Anniversary]

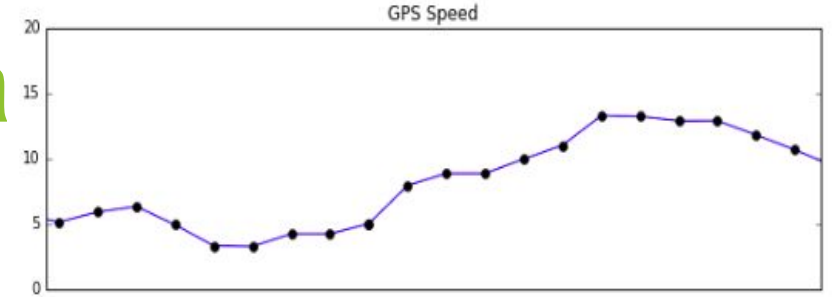
# Core Pipeline

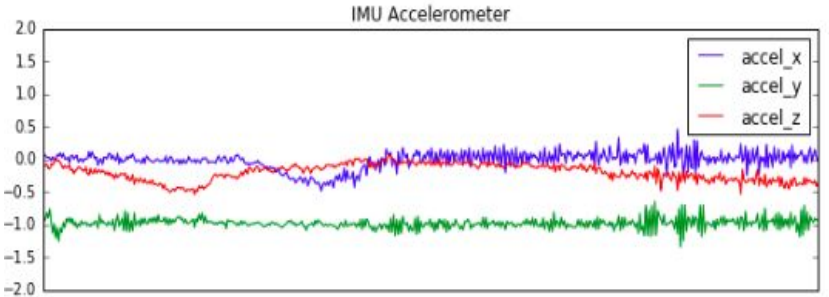


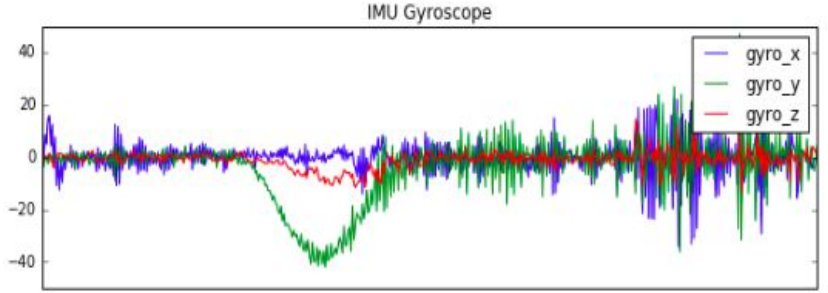
# Phone Sensor Data

- GPS
  - Absolute location, velocity and time
  - Low frequency (<= 1 point per second)
- IMU
  - Relative motion of phone
    - Accelerometer: 3D linear acceleration
    - Gyroscope: 3D angular velocity
  - High frequency (>20 points per second)



# GPS Map-Matching

altitude: 0

course: 171.2255096435547

epoch: 1518909650

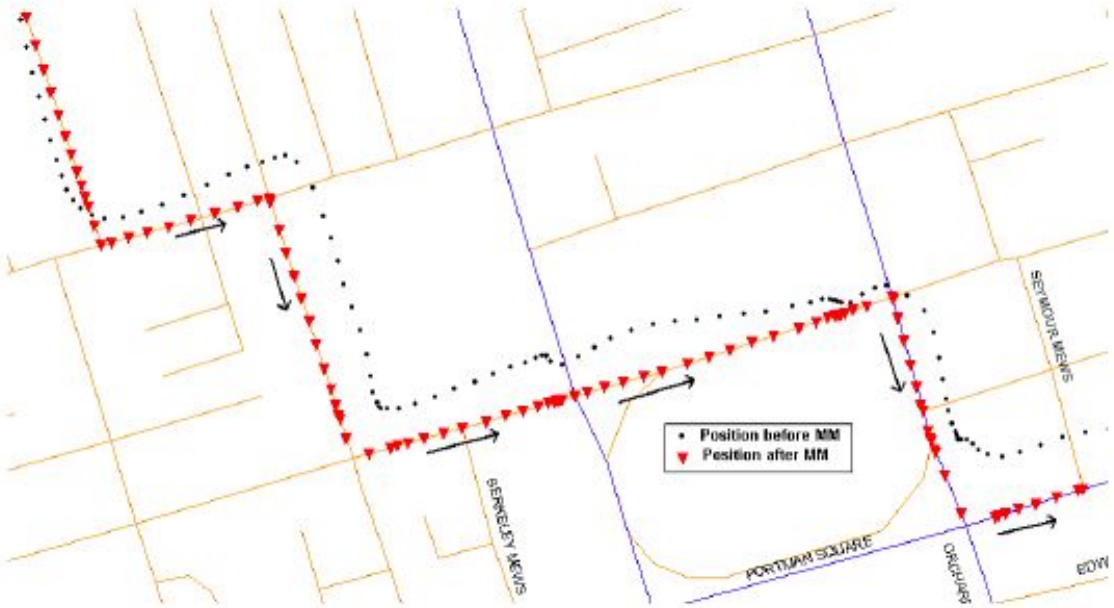
horizontalAccuracy: 10

latitude: 37.79969442636411

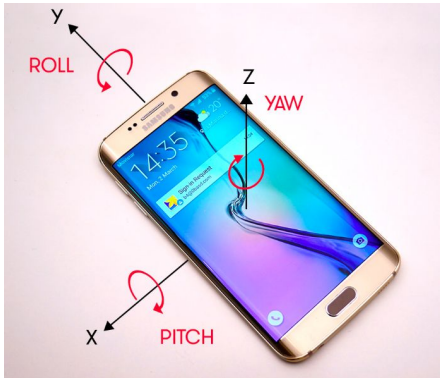
longitude: -122.43939904970063

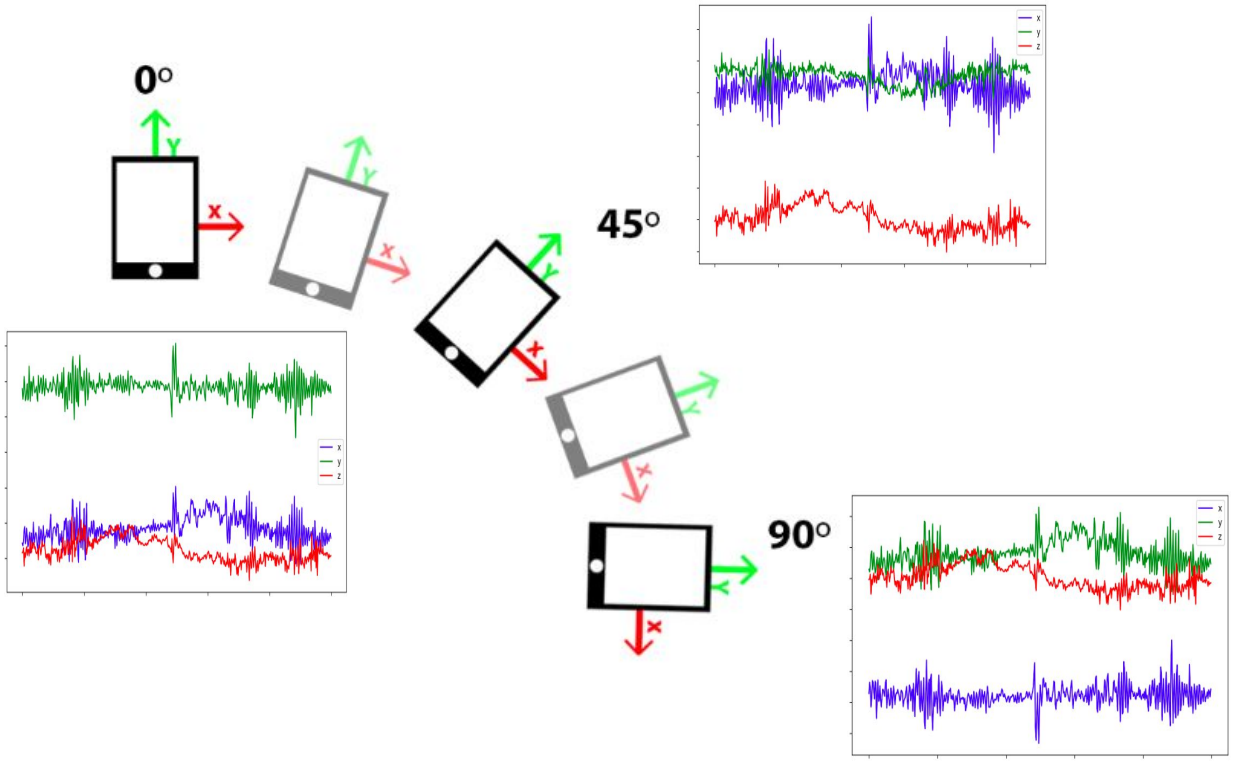
speed: 9.76000022888184

verticalAccuracy: 3

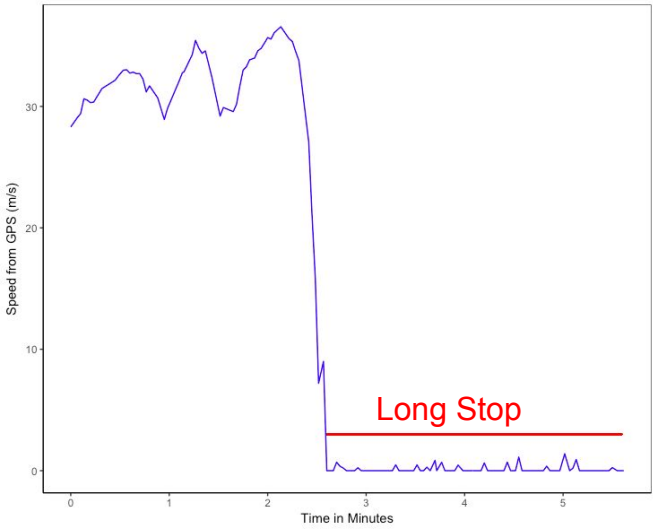


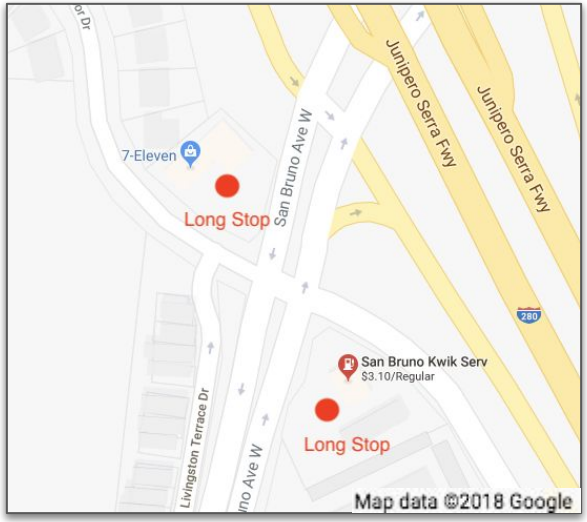
# Phone Re-Orientation

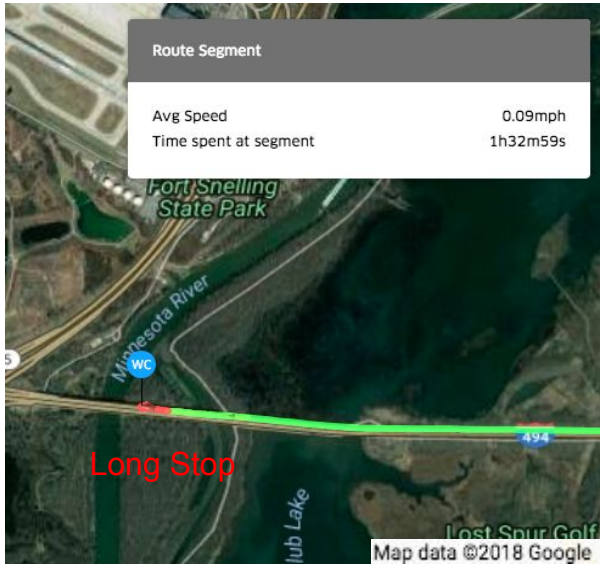


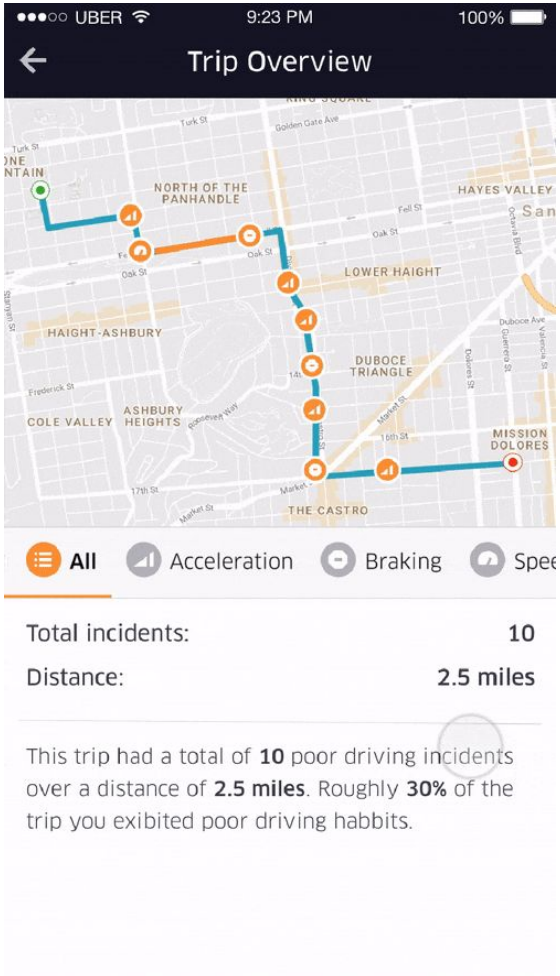
# Long Stop Detection

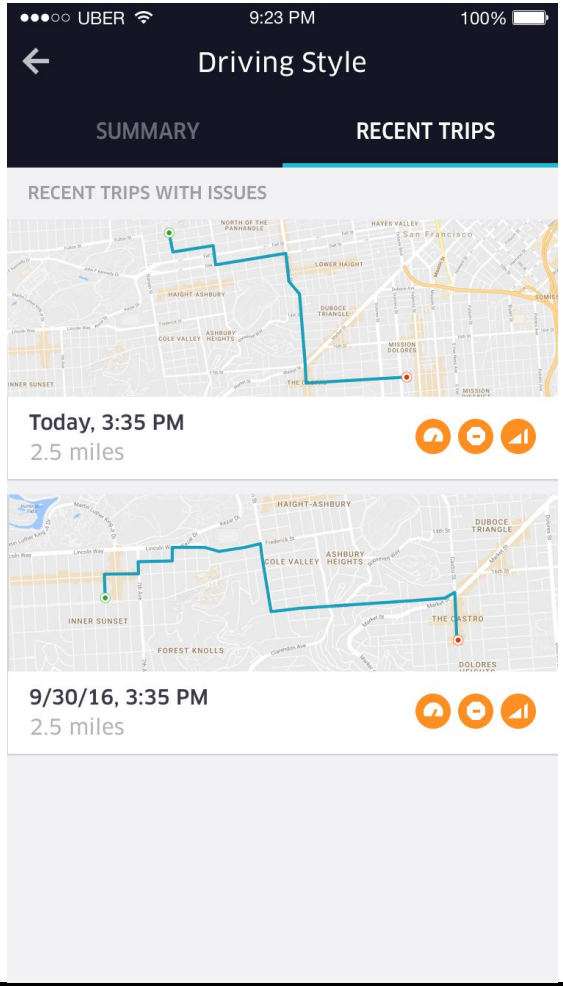


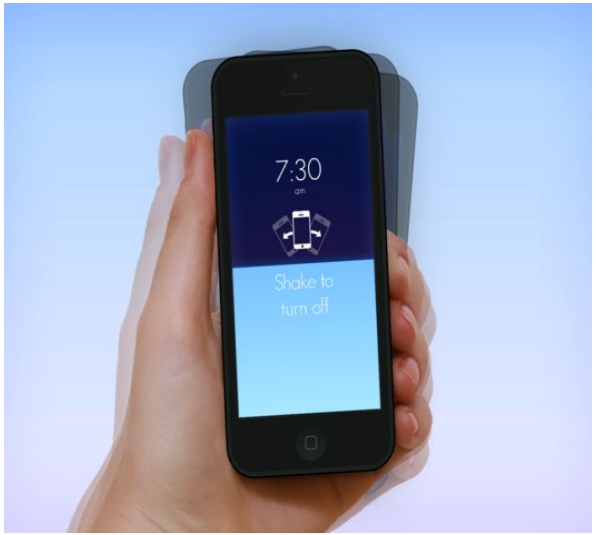



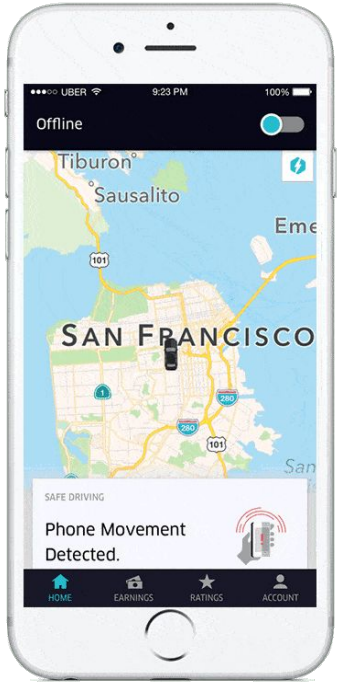
# Vehicle Movement



# Phone Mounting

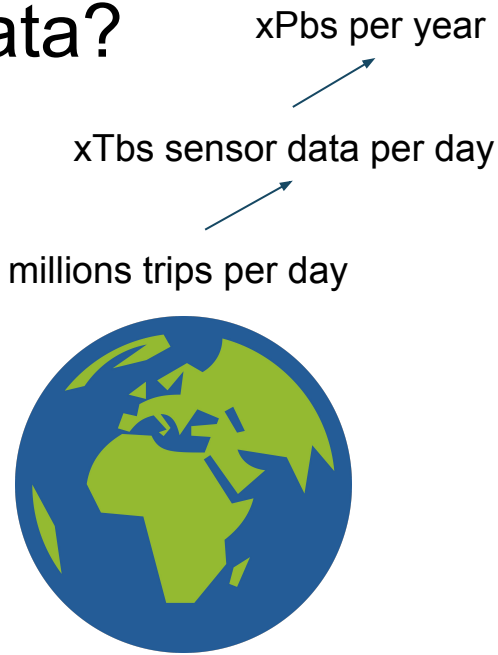


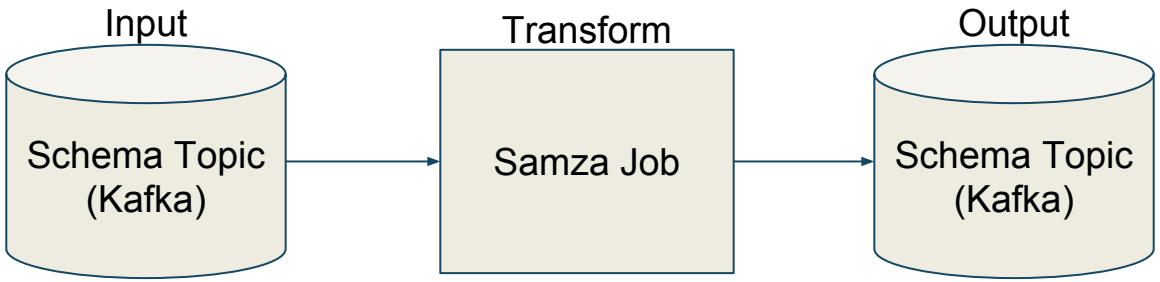
# Engineering

- Pipeline
  - Past
  - Present
- Problems

How big is our data?

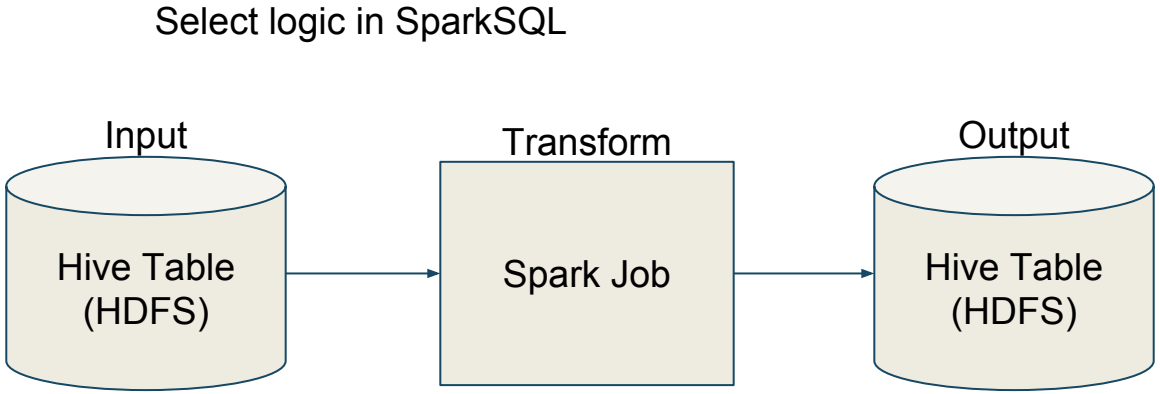


# Data Pipeline - Past (Streaming)



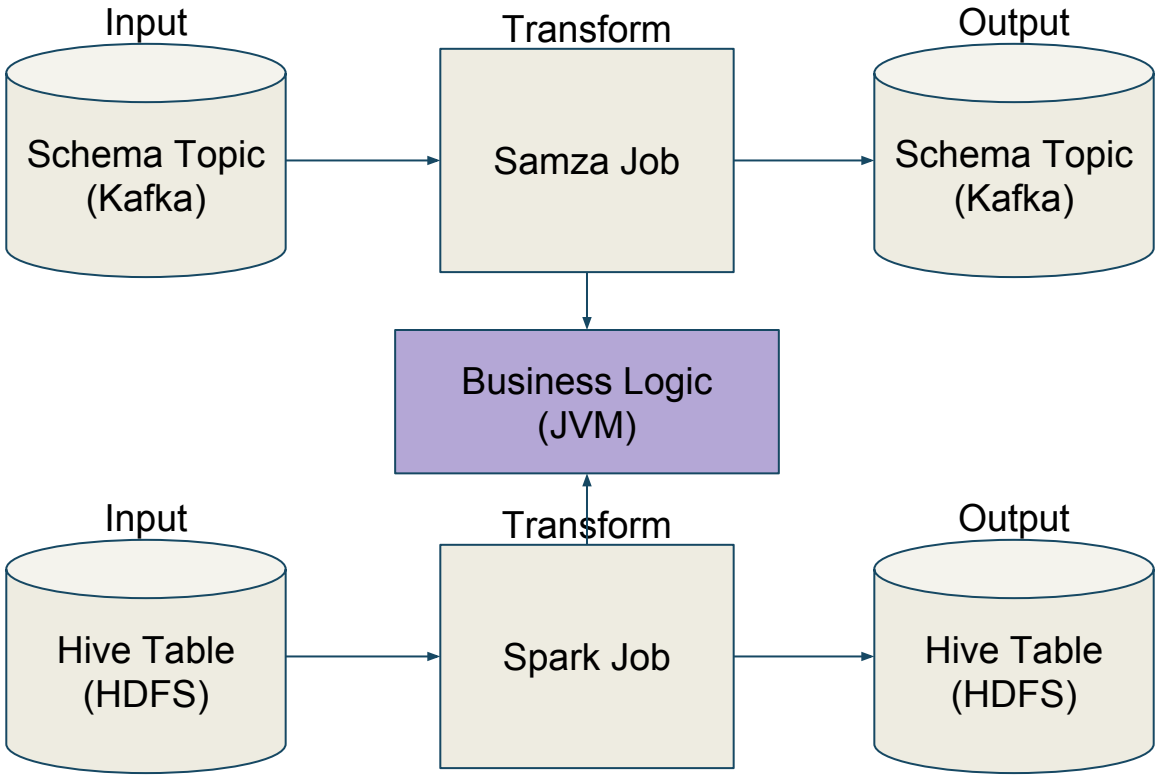

- Realtime

# Data Pipeline - Present (Batch)



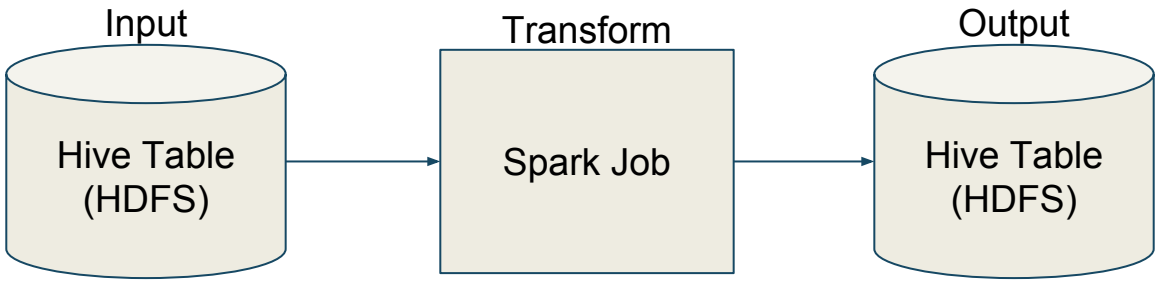

- Flexible

# Data Pipeline - Actuality (λ)

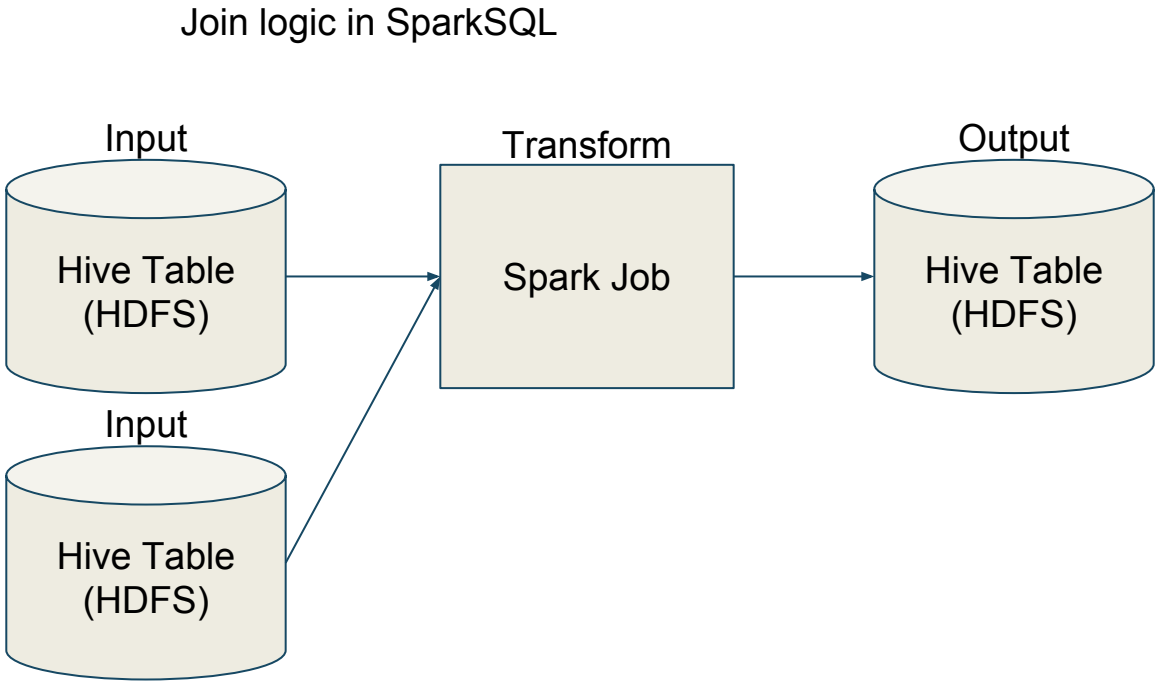


# Data Pipeline - Present

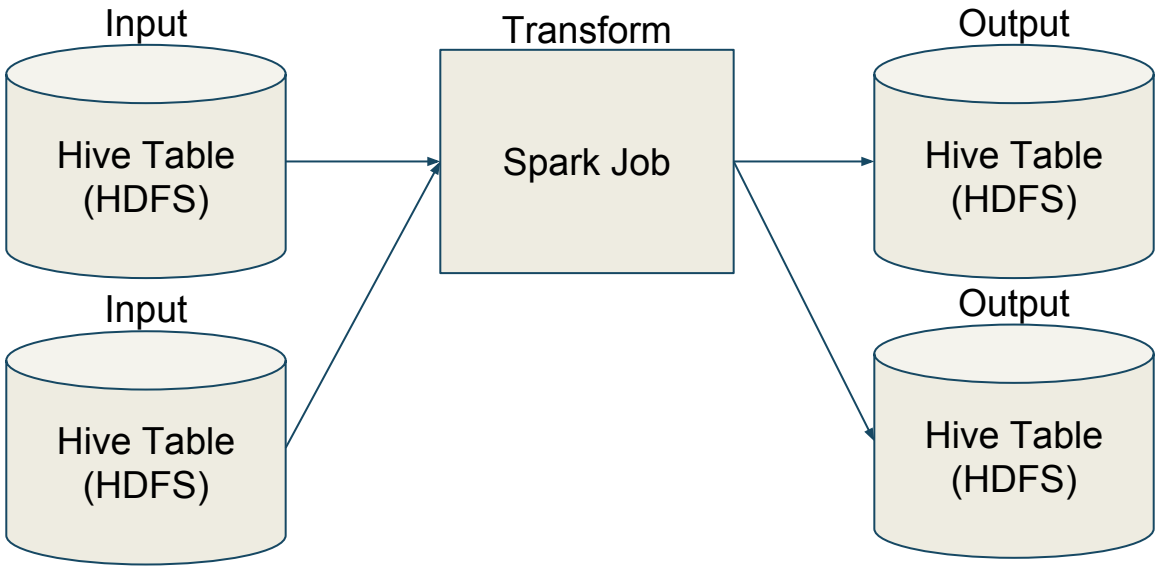
Select logic in SparkSQL



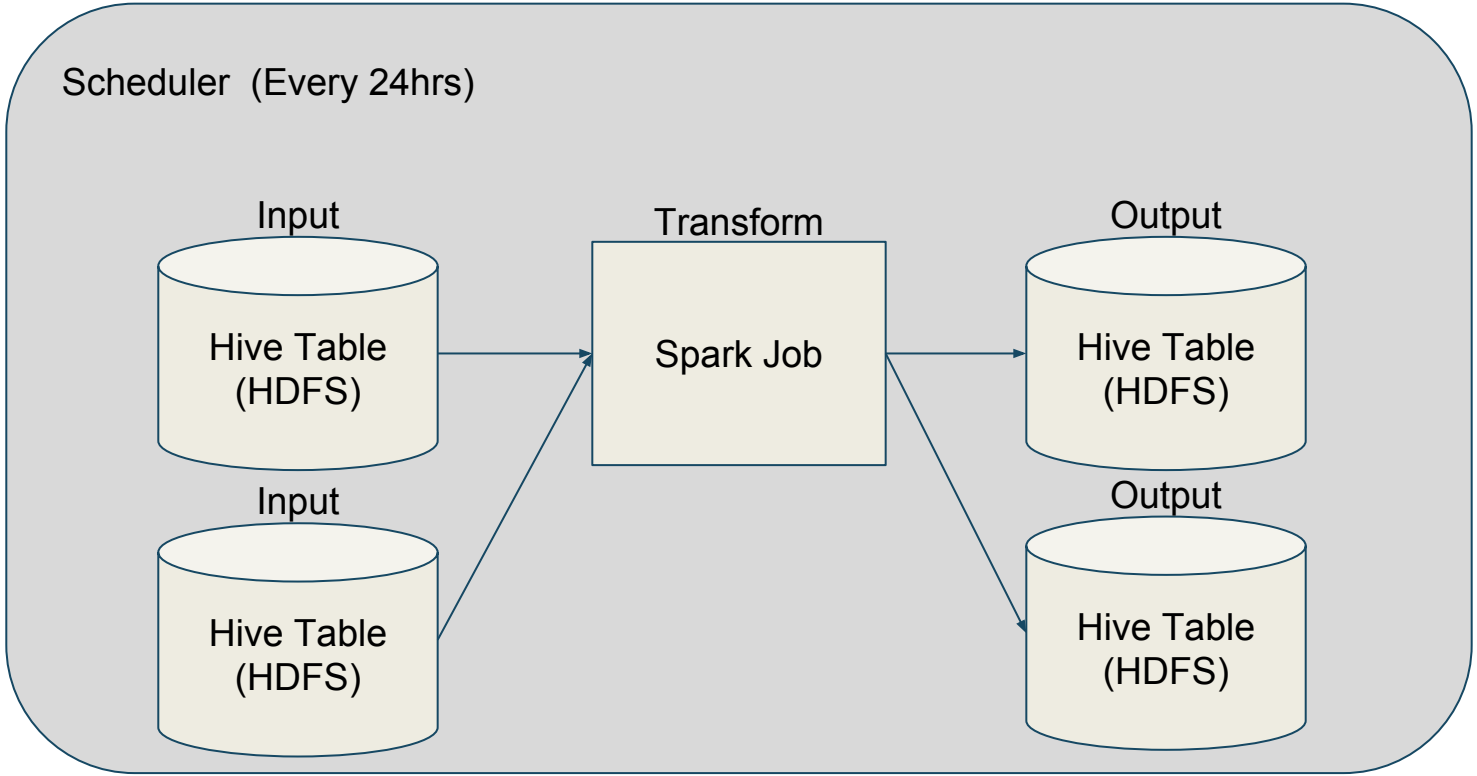
# Data Pipeline



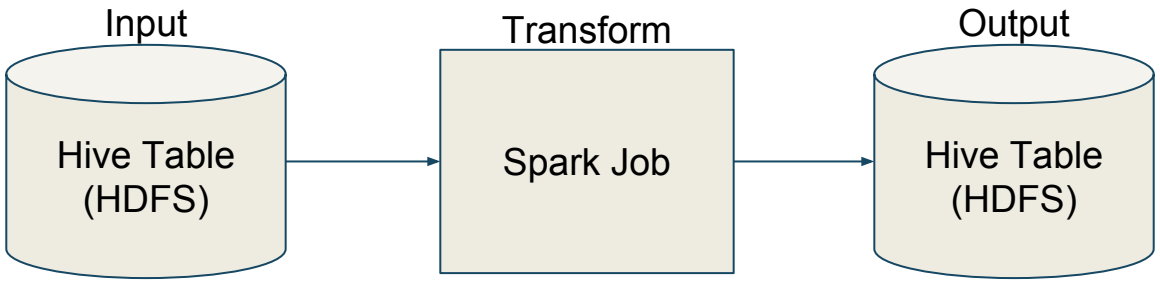
# Data Pipeline



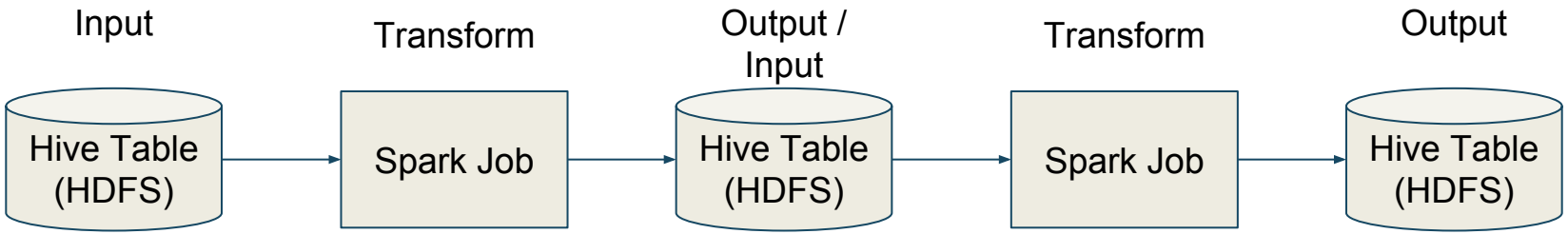
# Data Pipeline



# Data Pipeline



# Data Pipeline



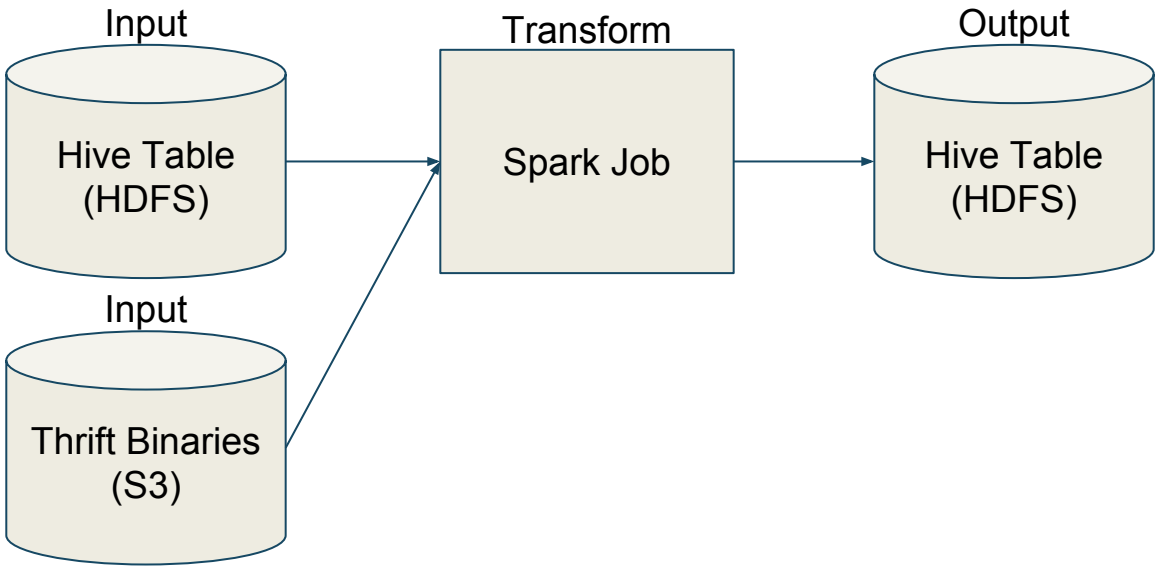
# Data Pipeline - Actuality

# Eng Problems

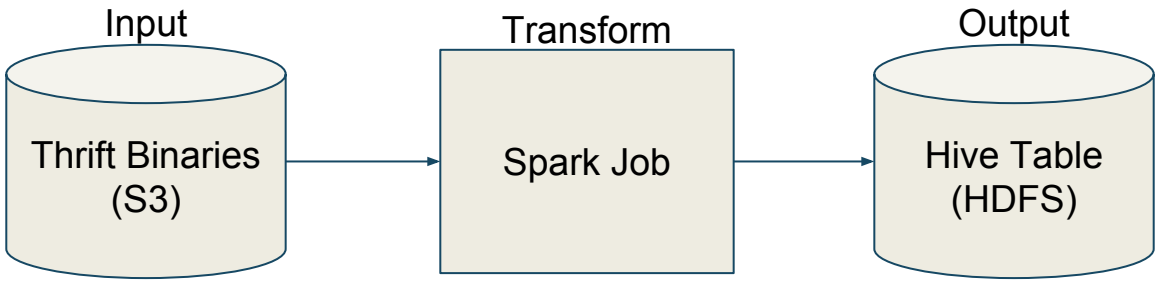
- Data Sources
- OOM Errors
- Too many Namenodes

# Eng Problems - Data Sources

Join logic in SparkSQL
**Doesn't work**

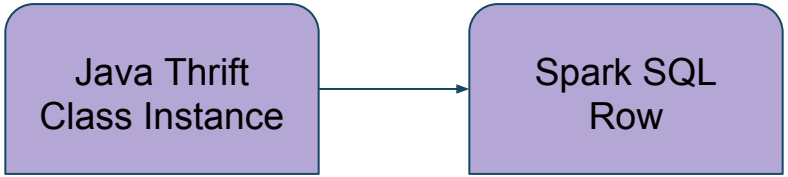


# Eng Problems - Data Sources

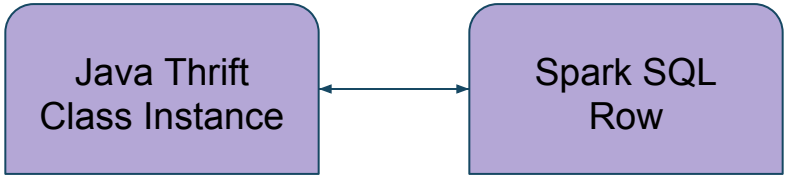



github.com/airbnb/airbnb-spark-thrift

# Aside: Encode Decode Invariant

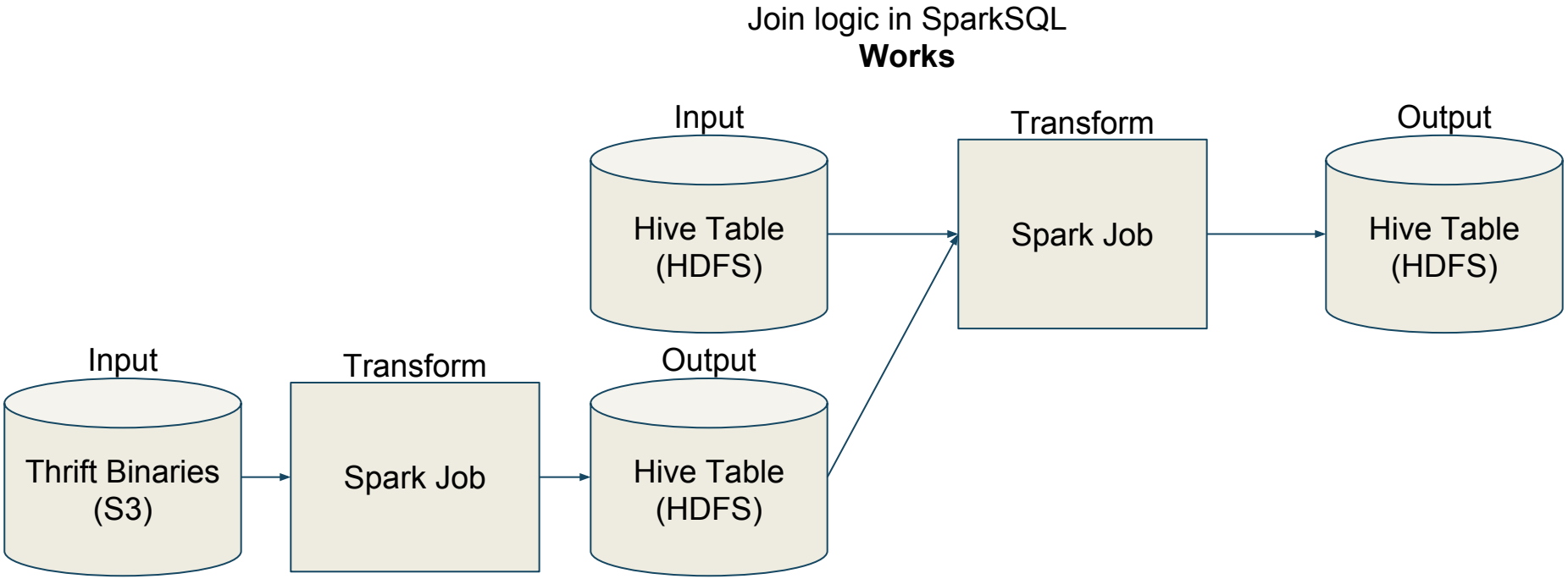


# Aside: Encode Decode Invariant




Generate random data to test
(ScalaCheck Library)

# Eng Problems - Data Sources



# Eng Problems - OOM Errors

- If hitting OOM related issues, usually increasing partitions help
  - `spark.sql.shuffle.partitions=X`
    - Where x > 200 (default)
- Might also play with executor-cores and memory settings

# Eng Problems - # Namenode

- If we have more partitions, we create more files
- Solution: Merge file after job run
  - github.com/apache/parquet-mr/tree/master/parquet-tools

# Looking Towards the Future


low
high

*Fraction of hard braking events*

*Filtered for segments >= 100 traversals and more than 10 hard braking events*

# Thank You!

**(And thanks for everyone @ Uber who helped us)**

Email us if you have any questions:

*actuaryzhang@uber.com*

*nwparker@uber.com*