Bringing Data to Life

Data management and Visualization Techniques

Benika Hall

Rob Harrison

Corporate Model Risk

March 16, 2018

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Introduction

- Benika Hall
  Analytic Consultant
  Wells Fargo - Corporate Model Risk

- Rob Harrison
  Analytic Consultant
  Wells Fargo - Corporate Model Risk

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Presentation Outline

- Business Model of the Financial Industry
- About the Data
  - Financial Data Flow Overview
  - Complexities of Data
  - Ingestion
  - Preparation
  - Pre-processing
  - Data Exploration
- Available Solutions
- Regression Analysis with GPU databases
  - Fannie Mae Example

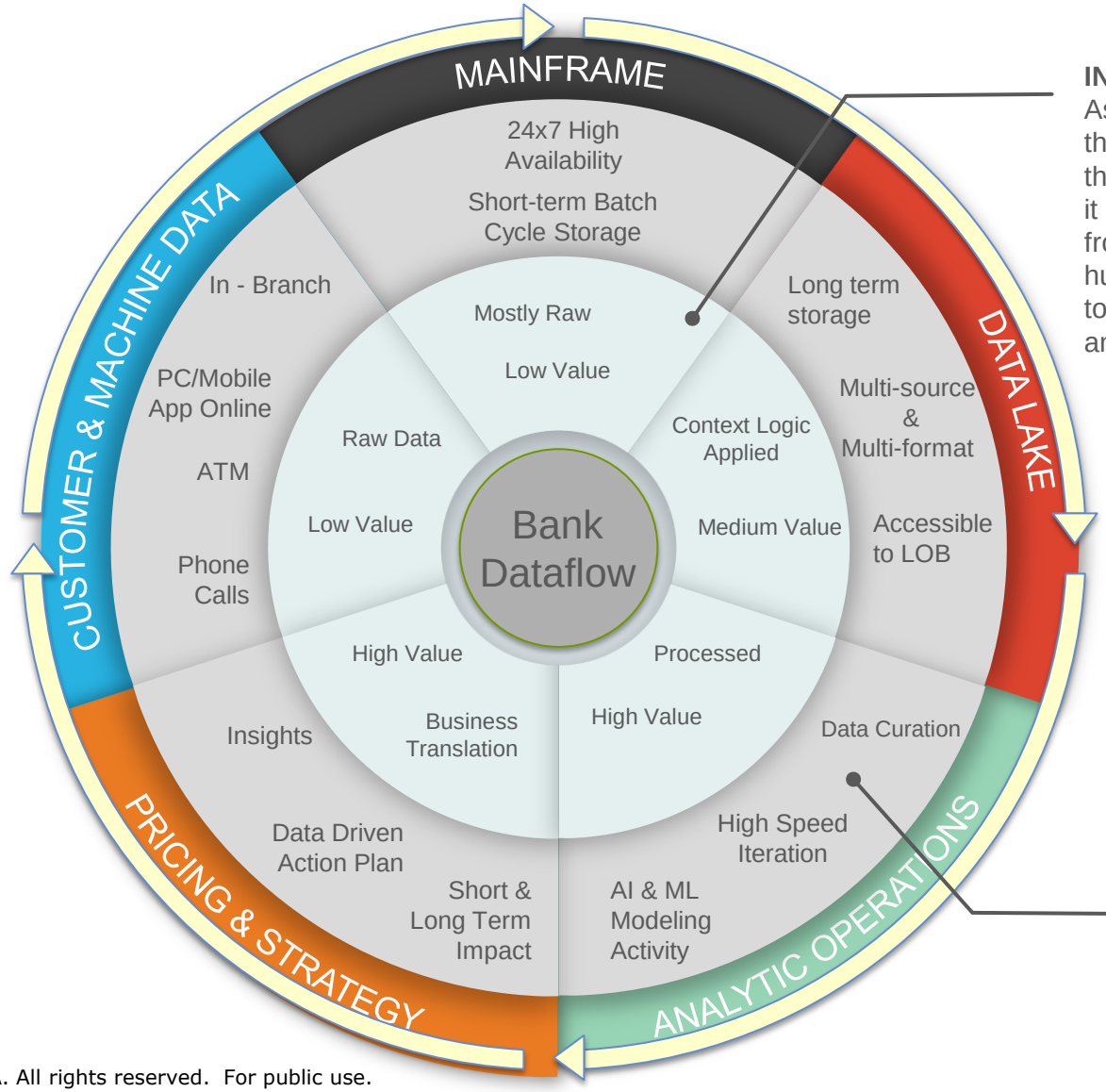
© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Business Model of the Financial Industry

- What do banks do?
- Comparison to other industries
- Financial Industry Business Model
  - Provide financial services
  - Manage your money
  - Working with data!

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Financial Dataflow Overview



INFUSION STATE:
As data progresses through each stage through the lifecycle, it is infused with value from raw low value human/machine level to highly refined analytical output.

It is here that we apply machine learning algorithms to infuse value into the data.

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Complexities of Data

Format

Sensitivity

Volume

Multi-sourced

Variety

Lineage

Velocity

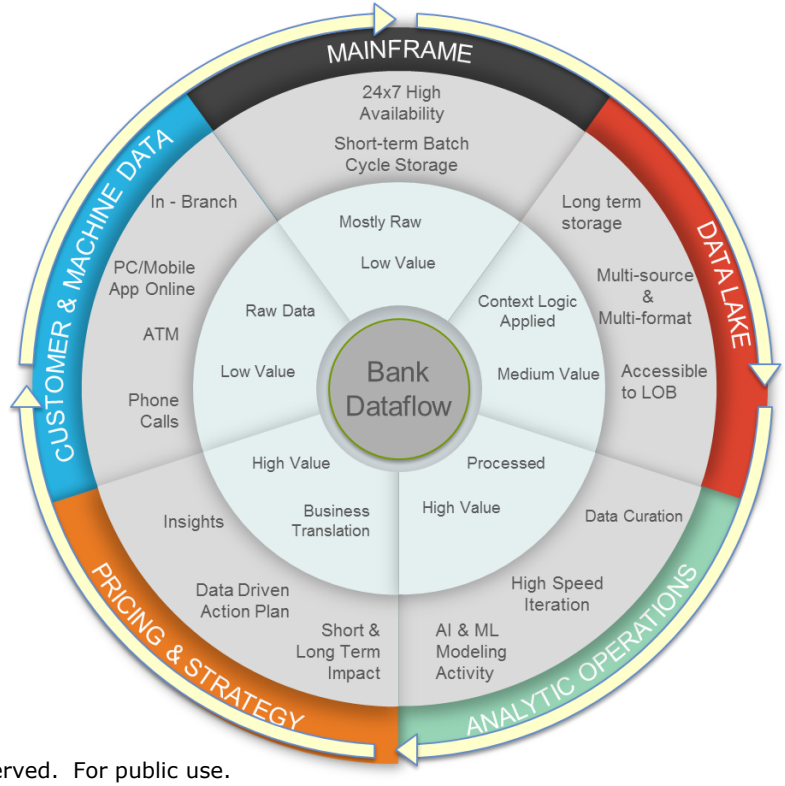
Context

Governance

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Phases of Data Management & Analysis

- Ingestion – pulling data from source system into local storage environment
- Preparation – data merging based on context and analytical needs
- Pre-processing – feature engineering, missing value imputation, outlier detection, categorical feature encoding, binning, etc.



© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Data Ingestion

- Ingestion – pulling data from source system into local storage environment
  - Sqoop
  - Kafka
  - SAS Connectors
  - NiFi

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Sqoop

- Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.

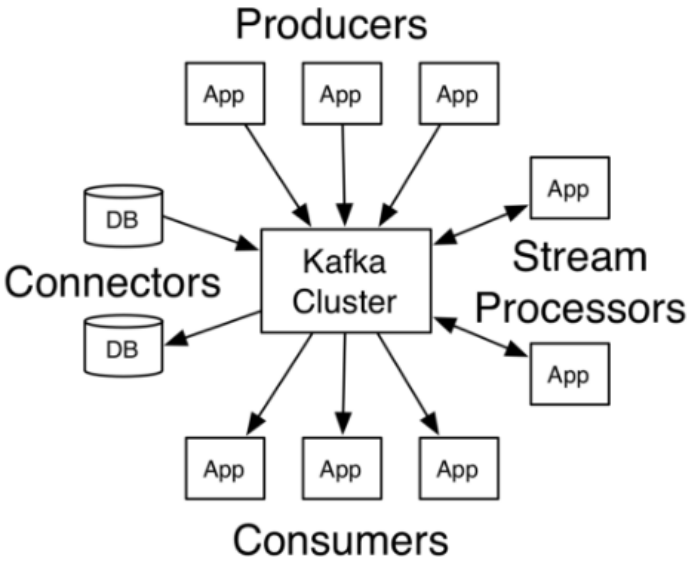
Function	Benefit
Import sequential datasets from mainframe	Satisfies the growing need to move data from mainframe to HDFS
Import direct to ORCFiles	Improved compression and light-weight indexing for improved query performance
Data imports	Moves certain data from external stores and EDWs into Hadoop to optimize cost-effectiveness of combined data storage and processing
Parallel data transfer	For faster performance and optimal system utilization
Fast data copies	From external systems into Hadoop
Efficient data analysis	Improves efficiency of data analysis by combining structured data with unstructured data in a schema-on-read data lake
Load balancing	Mitigates excessive storage and processing loads to other systems

YARN coordinates data ingest from Apache Sqoop and other services that deliver data into the Enterprise Hadoop cluster.

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Kafka

- Apache Kafka is an open-source stream processing software platform developed by the Apache Software Foundation written in Scala and Java. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds.



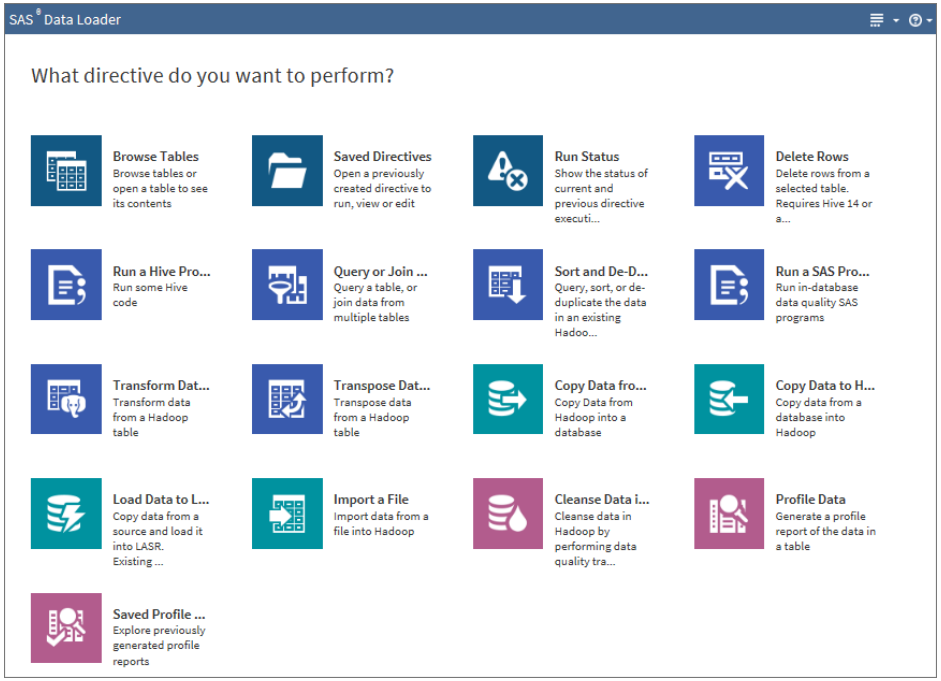
Kafka has four core APIs:

- The Producer API allows an application to publish a stream of records to one or more Kafka topics.
- The Consumer API allows an application to subscribe to one or more topics and process the stream of records produced to them.
- The Streams API allows an application to act as a *stream processor*, consuming an input stream from one or more topics and producing an output stream to one or more output topics, effectively transforming the input streams to output streams.
- The Connector API allows building and running reusable producers or consumers that connect Kafka topics to existing applications or data systems. For example, a connector to a relational database might capture every change to a table.

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

SAS Connectors

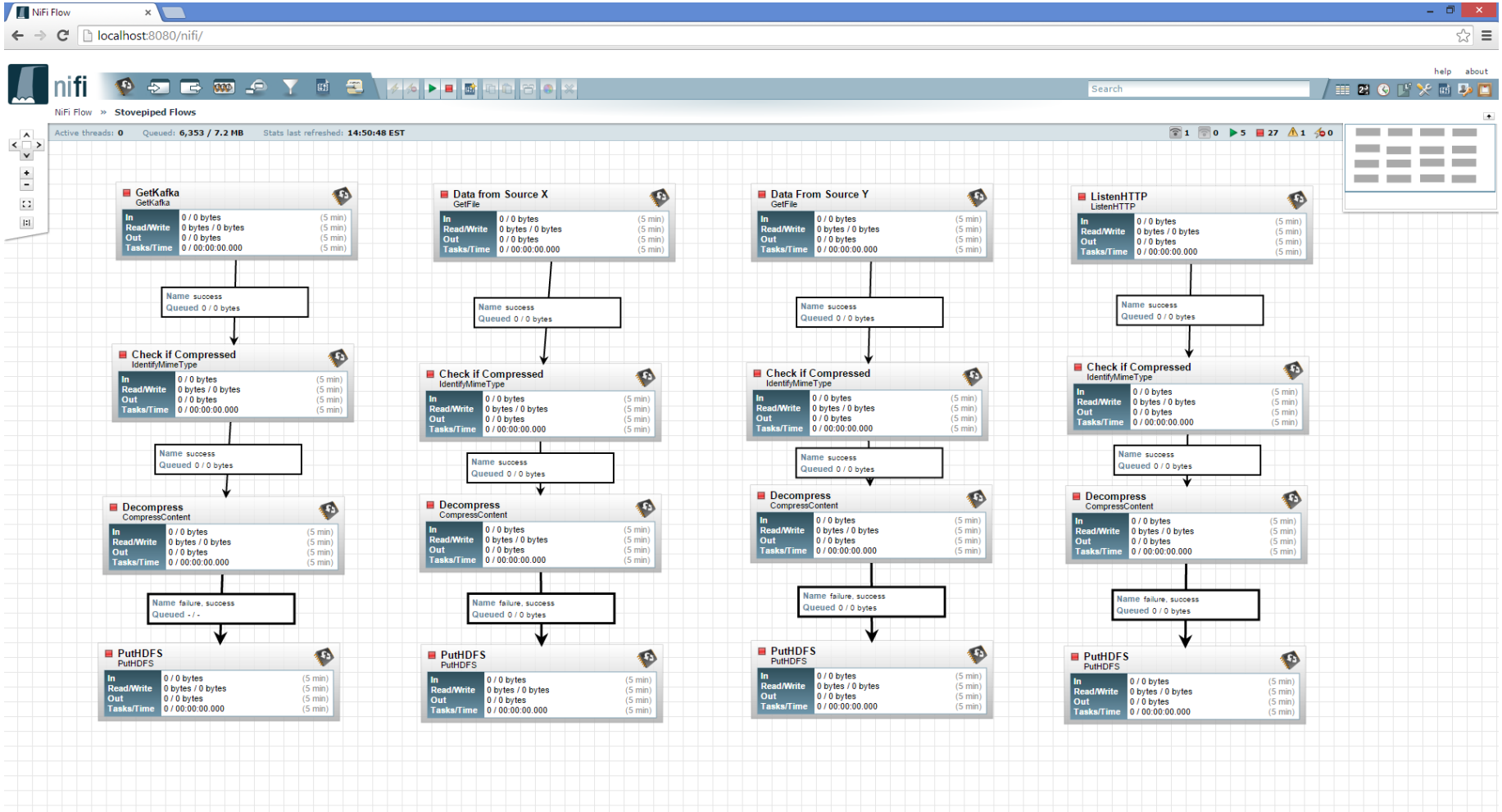
- SAS/ACCESS Interface to Hadoop
- SAS Data Loader



© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

NiFi

Apache NiFi supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic.



© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

NiFi *(continued)*

- **Systems fail:**
  Networks fail, disks fail, software crashes, people make mistakes.

- **Data access exceeds capacity to consume:**
  Sometimes a given data source can outpace some part of the processing or delivery chain - it only takes one weak-link to have an issue.

- **Boundary conditions are mere suggestions:**
  You will invariably get data that is too big, too small, too fast, too slow, corrupt, wrong, or in the wrong format.

- **What is noise one day becomes signal the next:**
  Priorities of an organization change - rapidly. Enabling new flows and changing existing ones must be fast.

- **Systems evolve at different rates:**
  The protocols and formats used by a given system can change anytime and often irrespective of the systems around them. Dataflow exists to connect what is essentially a massively distributed system of components that are loosely or not-at-all designed to work together.

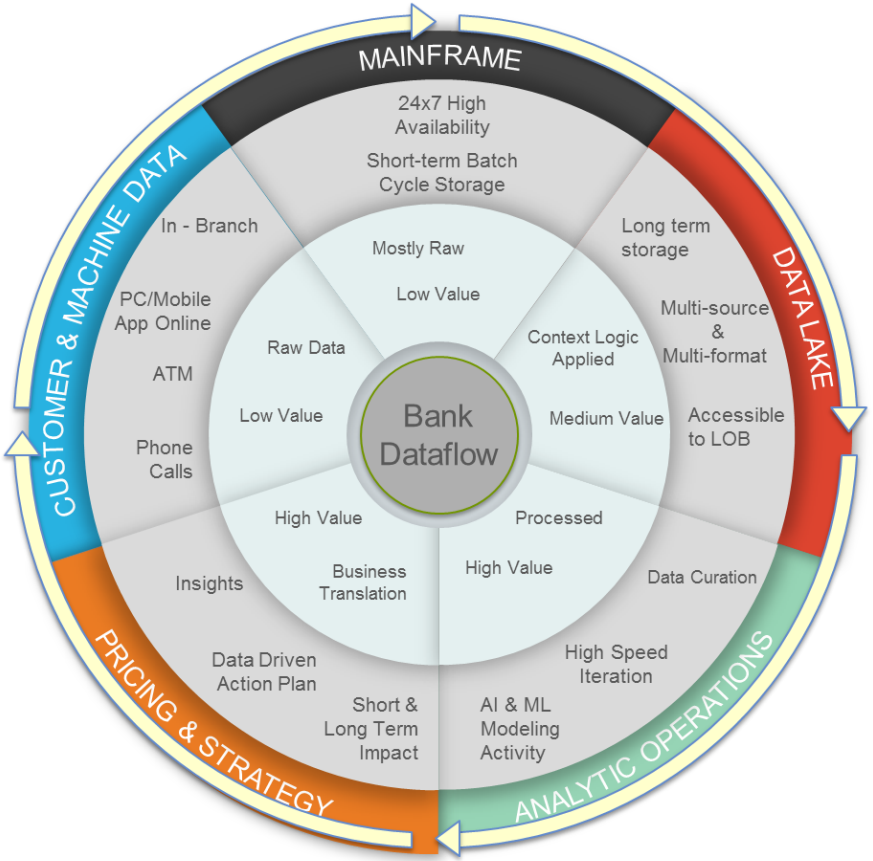
- **Compliance and security:**
  Laws, regulations, and policies change. Business to business agreements change. System to system and system to user interactions must be secure, trusted, accountable.

- **Continuous improvement occurs in production:**
  It is often not possible to come even close to replicating production environments in the lab

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Data Preparation

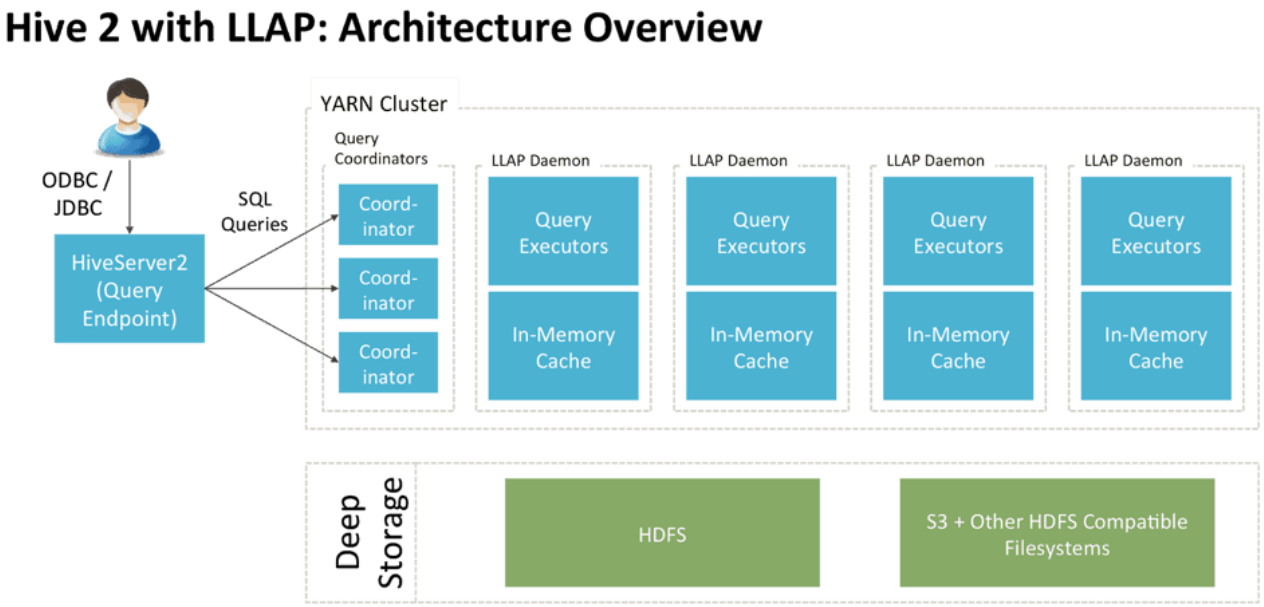
- Data Preparation –> requires data exploration!
  - GPU Databases
  - Hive
  - Hive LLAP
  - Spark
  - Alluxio
  - Druid



© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Hive & LLAP

- Live Long and Process, LLAP provides a hybrid execution model. It consists of a long-lived daemon which replaces direct interactions with the HDFS DataNode, and a tightly integrated DAG-based framework.
  Functionality such as caching, pre-fetching, some query processing and access control are moved into the daemon. Small/short queries are largely processed by this daemon directly, while any heavy lifting will be performed in standard YARN containers



© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Spark

- Apache Spark is a lightning-fast **unified analytics engine** for big data and machine learning. It was originally developed at UC Berkeley in 2009

- It provides in-memory computing capabilities to deliver speed, a generalized execution model to support a wide variety of applications, and Java, Scala, and Python APIs for ease of development.

- Supported languages:
  - R
  - SQL
  - Python
  - Scala
  - Java

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

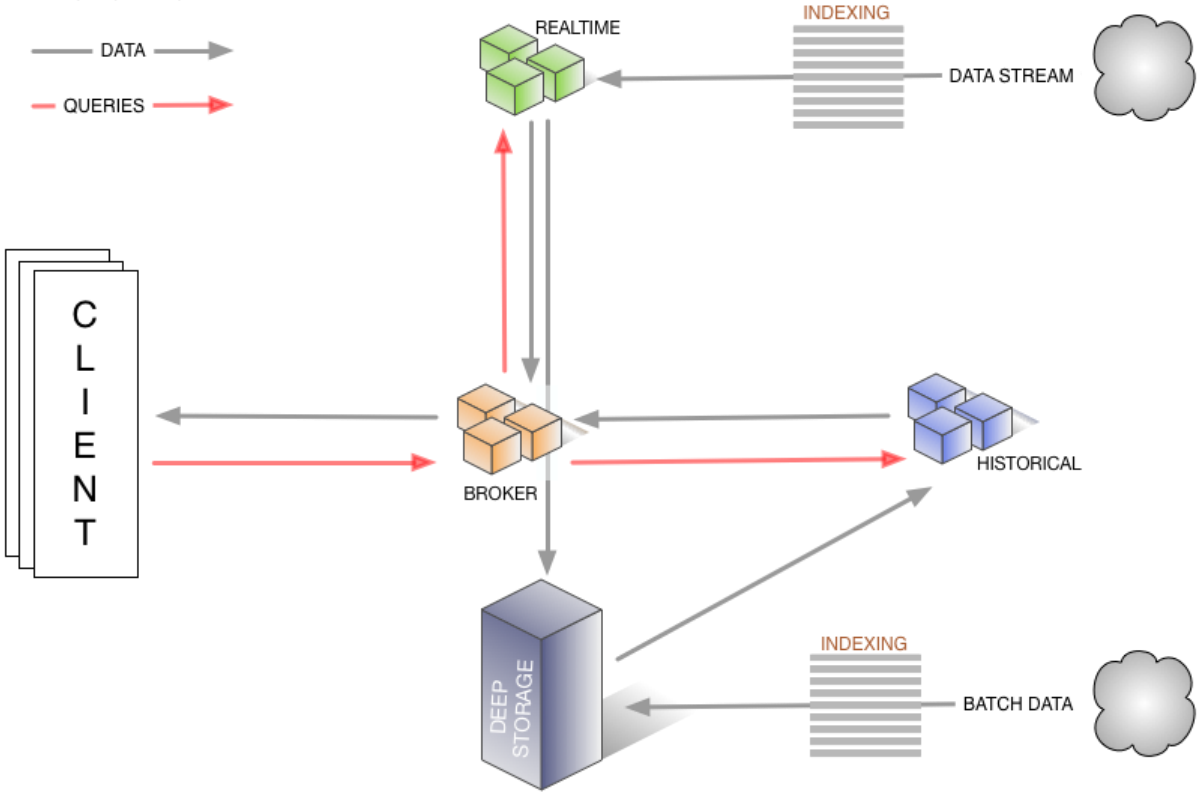
Alluxio

- Alluxio holds a unique place in the big data ecosystem, residing between **storage** systems such as Amazon S3, Apache HDFS or OpenStack Swift and **computation** frameworks and applications such as Apache Spark or Hadoop MapReduce and provides the central point of access with a memory centric design.

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Druid

- Druid is an open-source data store designed for sub-second queries on real-time and historical data.



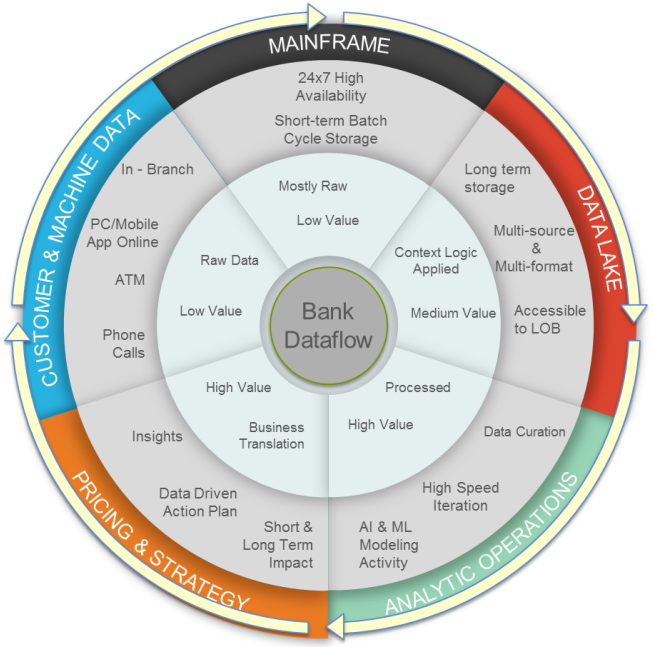
© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Data Pre-processing

- Feature engineering, missing value imputation, outlier detection, categorical feature encoding, binning, etc.

- Languages:
  - Python
  - R
  - SAS
  - Scala

- Why PySpark?
  - Leverage python
  - Big data
  - Flexibility

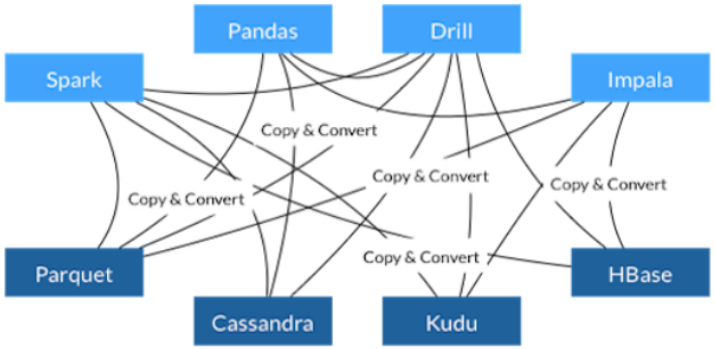


© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

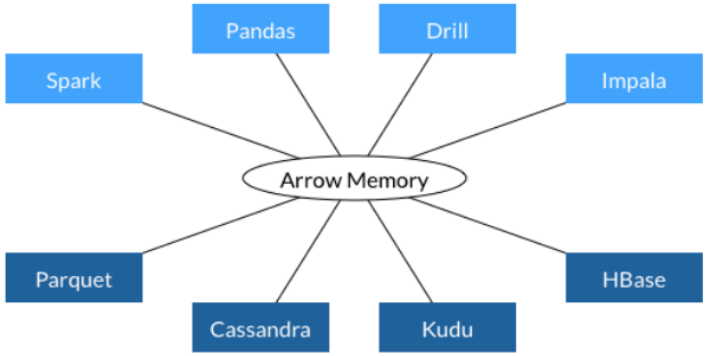
Apache Arrow - GOAI

- What is it?

Advantages of a Common Data Layer



- Each system has its own internal memory format
- 70-80% computation wasted on serialization and deserialization
- Similar functionality implemented in multiple projects



- All systems utilize the same memory format
- No overhead for cross-system communication
- Projects can share functionality (eg, Parquet-to-Arrow reader)

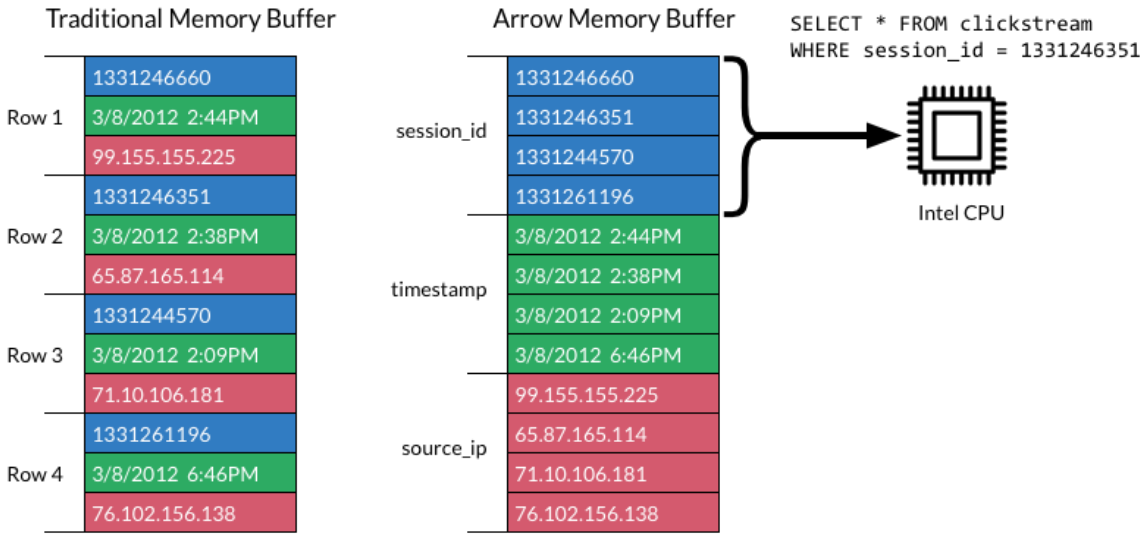
- How does this differ from Parquet? Alluxio?

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Apache Arrow

- Columnar In-Memory Storage

	session_id	timestamp	source_ip
Row 1	1331246660	3/8/2012 2:44PM	99.155.155.225
Row 2	1331246351	3/8/2012 2:38PM	65.87.165.114
Row 3	1331244570	3/8/2012 2:09PM	71.10.106.181
Row 4	1331261196	3/8/2012 6:46PM	76.102.156.138



© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Apache Arrow Example

- Example of usage:

```
from pyspark import SparkConf
from pyspark.sql import SparkSession
from pyspark.sql import Row
import time

#Configure spark session
conf = SparkConf().setAppName("Spark Arrow 032318")
conf.setMaster("local")
spark = SparkSession.builder.enableHiveSupport().config(conf=conf).getOrCreate()
```

```
#Load data
df_spark_default = spark.sql("select * from default.fannie_mae_mgd_1mm_txt limit 600000")
```

```
%time df_original = df_spark_default.toPandas()
```

```
CPU times: user 6.56 s, sys: 624 ms, total: 7.18 s
Wall time: 11.2 s
```

```
#Read into Arrow
table = pa.Table.from_pandas(df_original)

#Write into into parquet table format in Arrow
pq.write_table(table, 'example.parquet')

#Read into arrow dataframe
df_parquet_arrow = pq.read_table('example.parquet')
```

```
%time df_arrow = df_parquet_arrow.to_pandas()
```

```
CPU times: user 289 ms, sys: 75.6 ms, total: 365 ms
Wall time: 323 ms
```

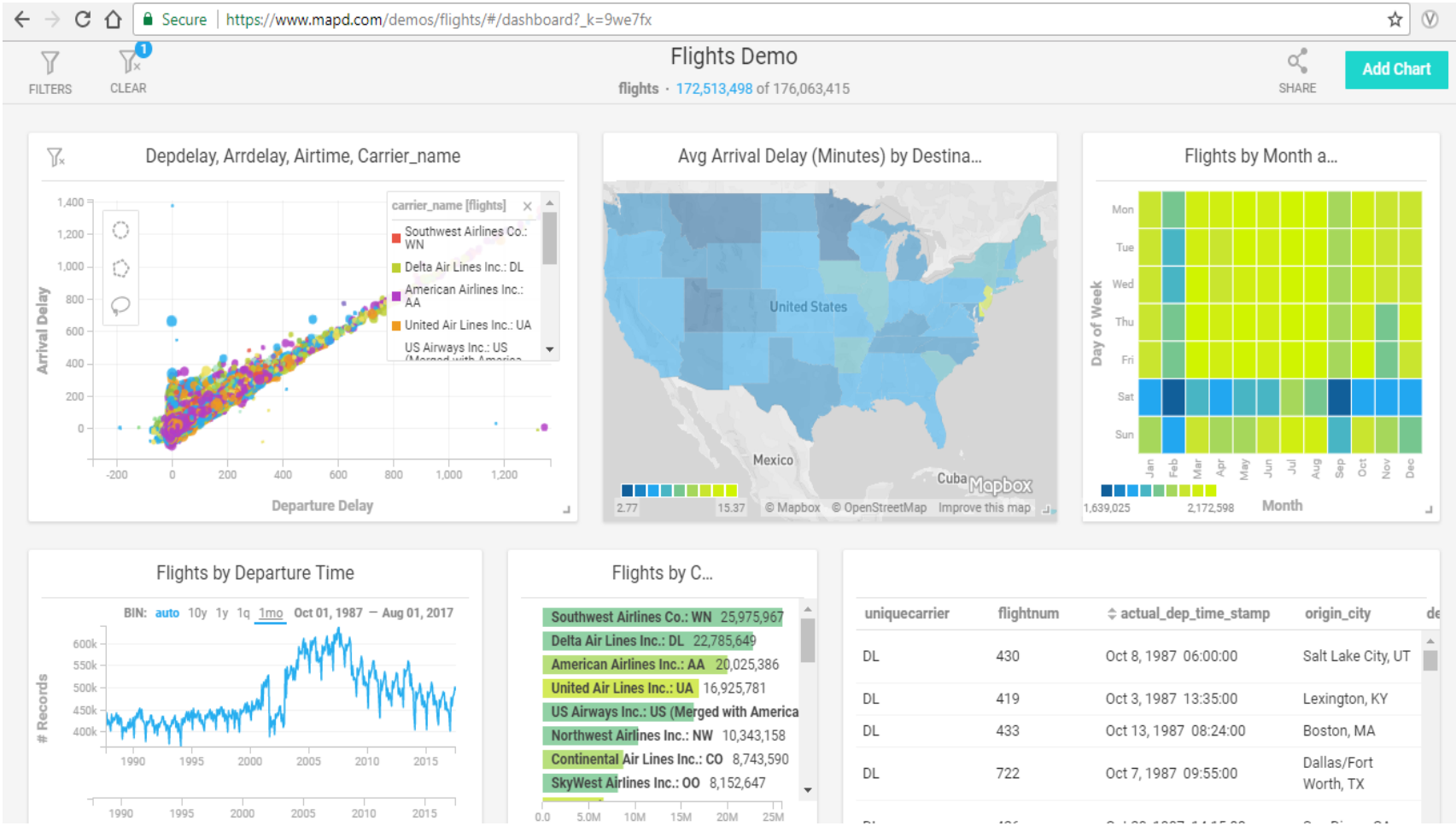
© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Data Exploration w/GPU Technology

- Overview and examples of each of the following:
  - Immerse
  - Superset
  - Reveal

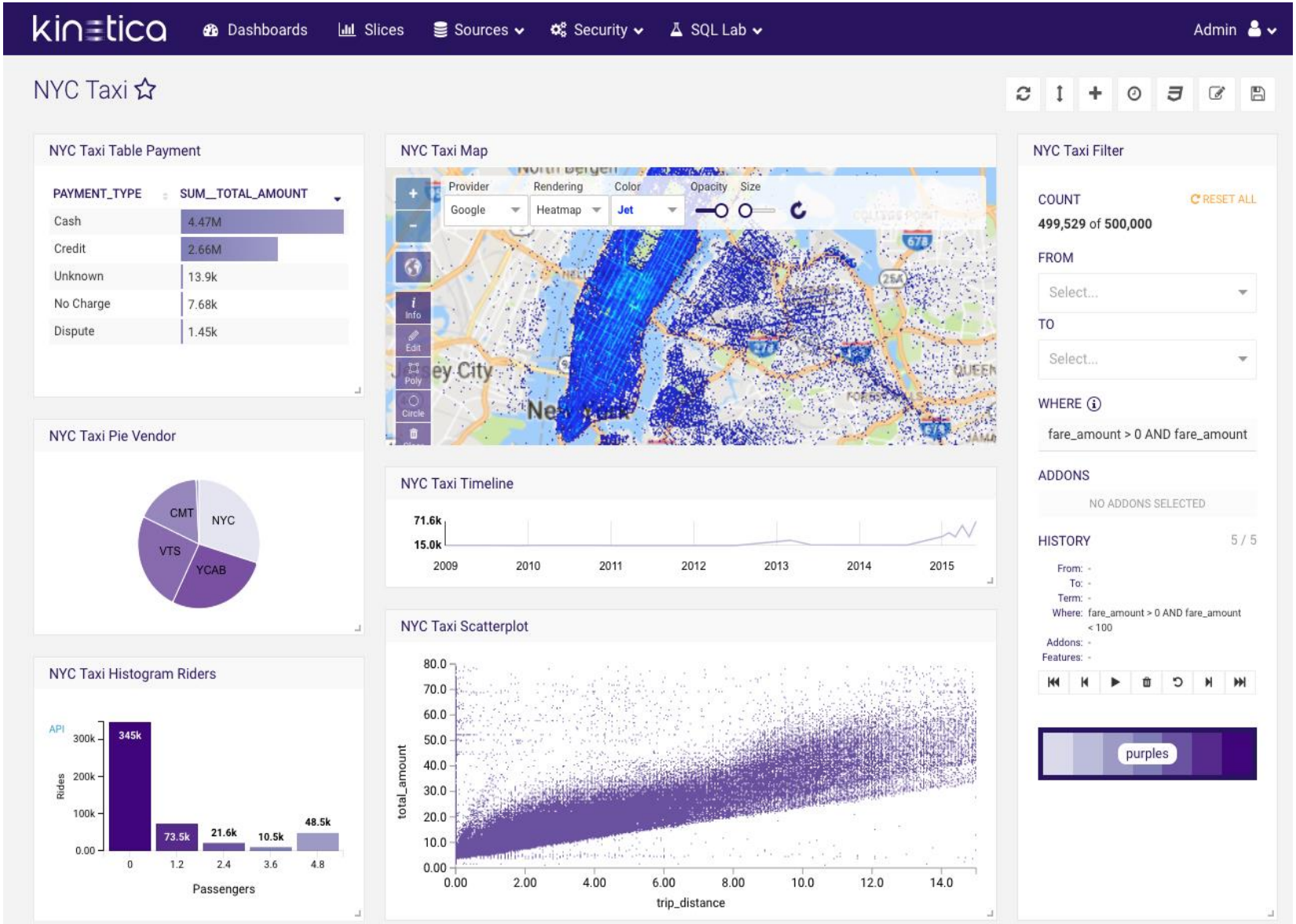
© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Immerse



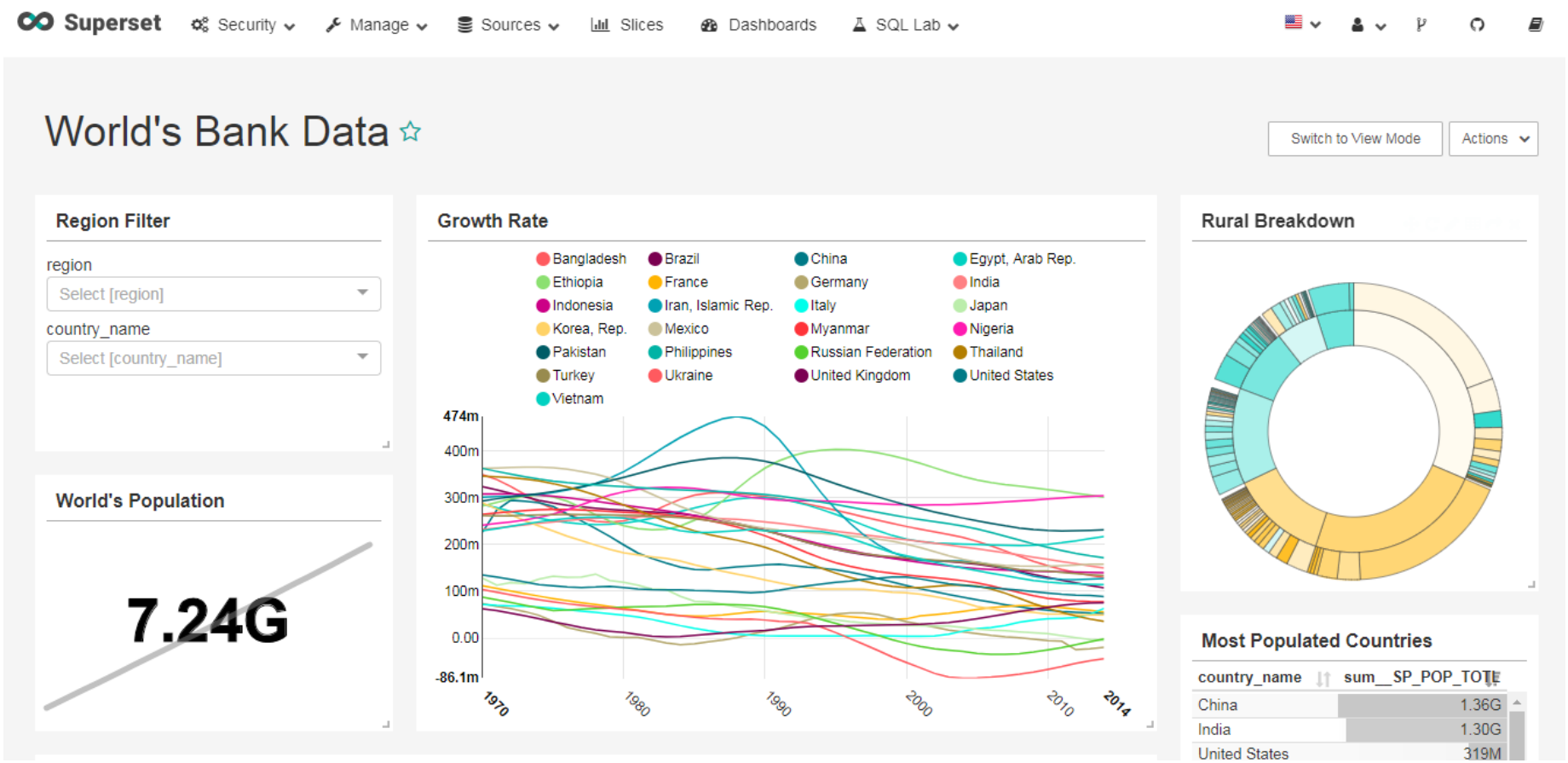
© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Reveal



© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Superset



© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Regression Analysis using GPU

- Example of running regression analysis using GPU for large-scale data and calculating variable importance:
  - Fannie Mae data of 1.8 billion rows
  - Gradient Boosting Trees
  - Hyper-parameter tuning
  - Determine Variable Importance

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

GPU Specifications

```
 NVIDIA-SMI 384.81                 Driver Version: 384.81                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-SXM2...  Off  | 00000002:01:00.0 Off |                    0 |
| N/A   29C    P0    30W / 300W |     10MiB / 16276MiB |      0%      Default |
```

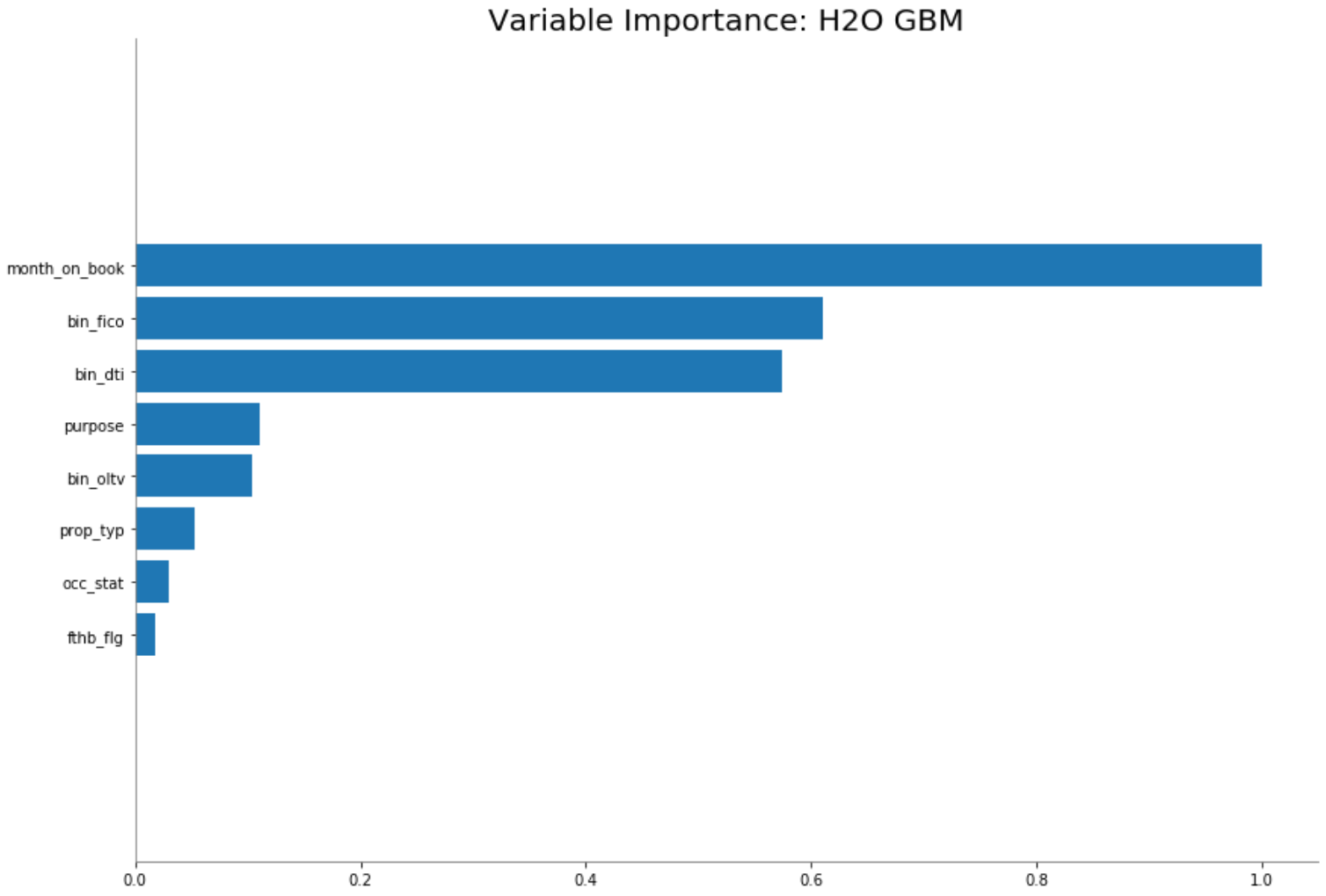
© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Variable Importance using GBM

variable	relative_importance	scaled_importance	percentage
month_on_book	67.9615631	1.0	0.4004045
bin_fico	41.4487762	0.6098856	0.2442009
bin_dti	38.9857483	0.5736441	0.2296896
purpose	7.5061255	0.1104466	0.0442233
bin_oltv	7.0943146	0.1043872	0.0417971
prop_typ	3.5422773	0.0521218	0.0208698
occ_stat	2.0115166	0.0295979	0.0118511
fthb_flg	1.1819650	0.0173917	0.0069637

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Visualization of Variable Importance



© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Performance comparison using GPU vs. CPU

	CPU	GPU
Time to load data (1.5 million rows)	1.5 hours	10 secs
Time to run model	2 hours	50 secs
Hyper-parameter tuning	4.5 hours	200 secs
Total	8 hours	260 secs

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Summary

What have we covered?

- Financial Data Overview
- Data Ingestion
- Data Preparation
- Current Solutions
- Data visualization
- Examples of Machine Learning using GPUs

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.

Thank you!

© 2017 Wells Fargo Bank, N.A. All rights reserved. For public use.