Apache Arrow

A High Perfomance Interoperable In-Memory Data Format

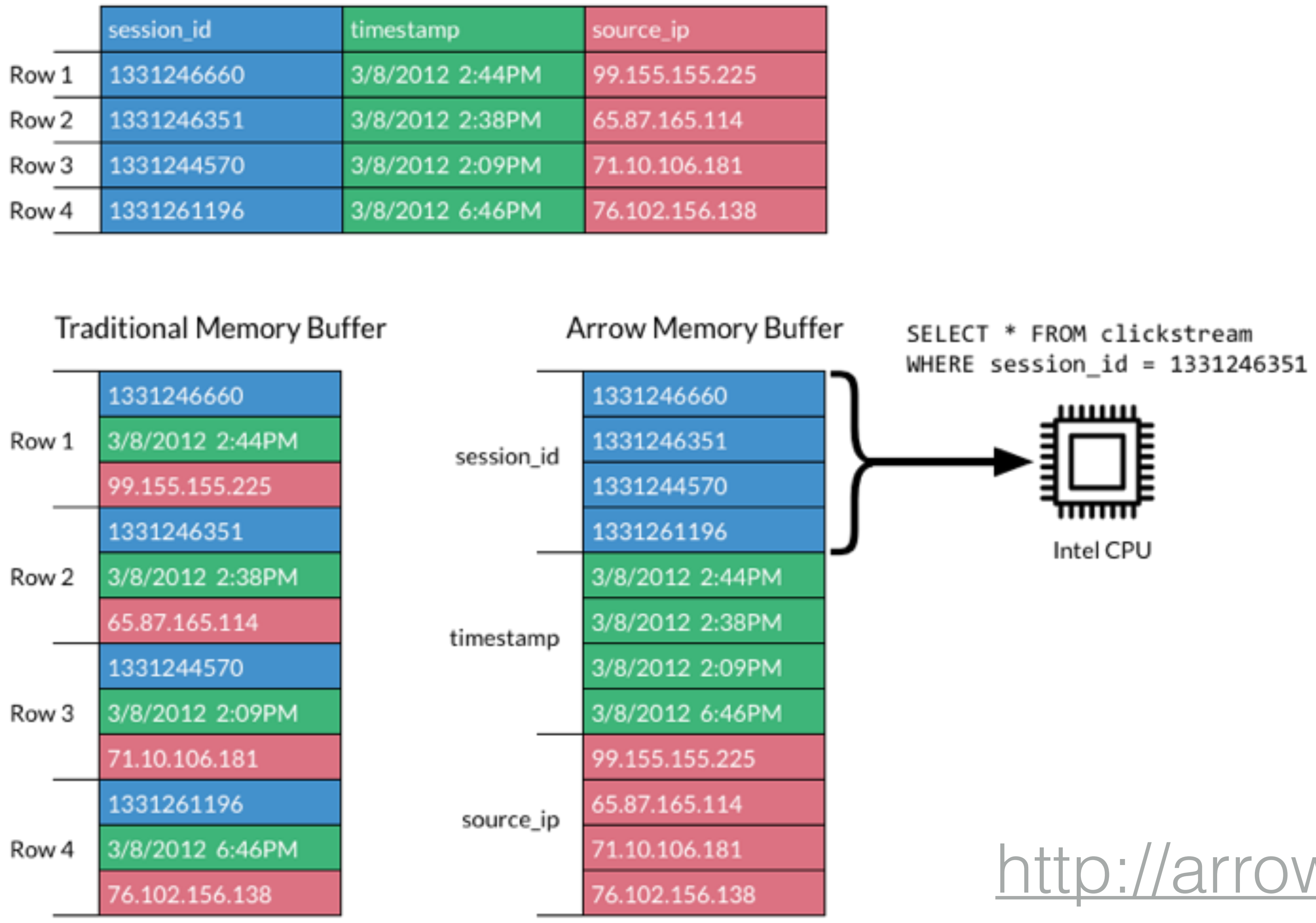
Chenglin Zhang
2016-06-23

A New Data Format

- Top level Apache project 2016-02-17
- Designed for columnar in-memory analytics and data interchange
- High performance on modern CPU, cache, and memory
- Nested data as first-class
- Common data format between languages and systems
- Backed-up by major open source data projects, including Calcite, Cassandra, Drill, Hadoop, HBase, Ibis, Impala, Kudu, Pandas, Parquet, Phoenix, Spark, Storm, R

Columnar In-Memory

- Vectorization, SIMD, pipelining, cache locality



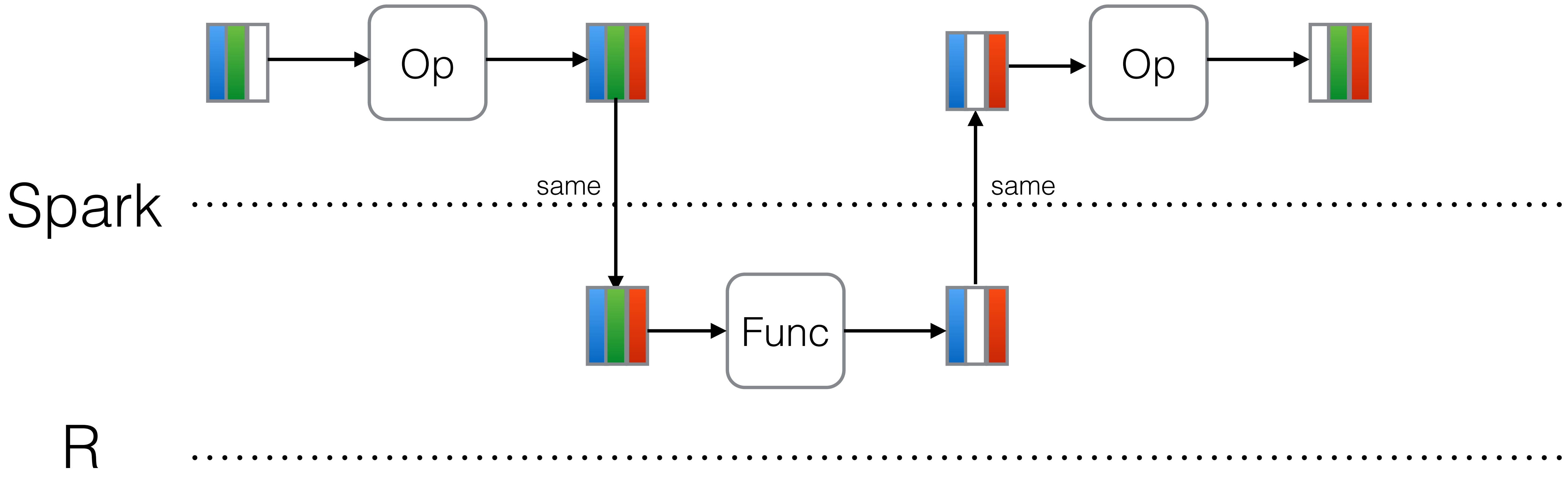
	session_id	timestamp	source_ip
Row 1	1331246660	3/8/2012 2:44PM	99.155.155.225
Row 2	1331246351	3/8/2012 2:38PM	65.87.165.114
Row 3	1331244570	3/8/2012 2:09PM	71.10.106.181
Row 4	1331261196	3/8/2012 6:46PM	76.102.156.138

```
SELECT * FROM clickstream
WHERE session_id = 1331246351
```

http://arrow.apache.org

Zero-Overhead Interchange

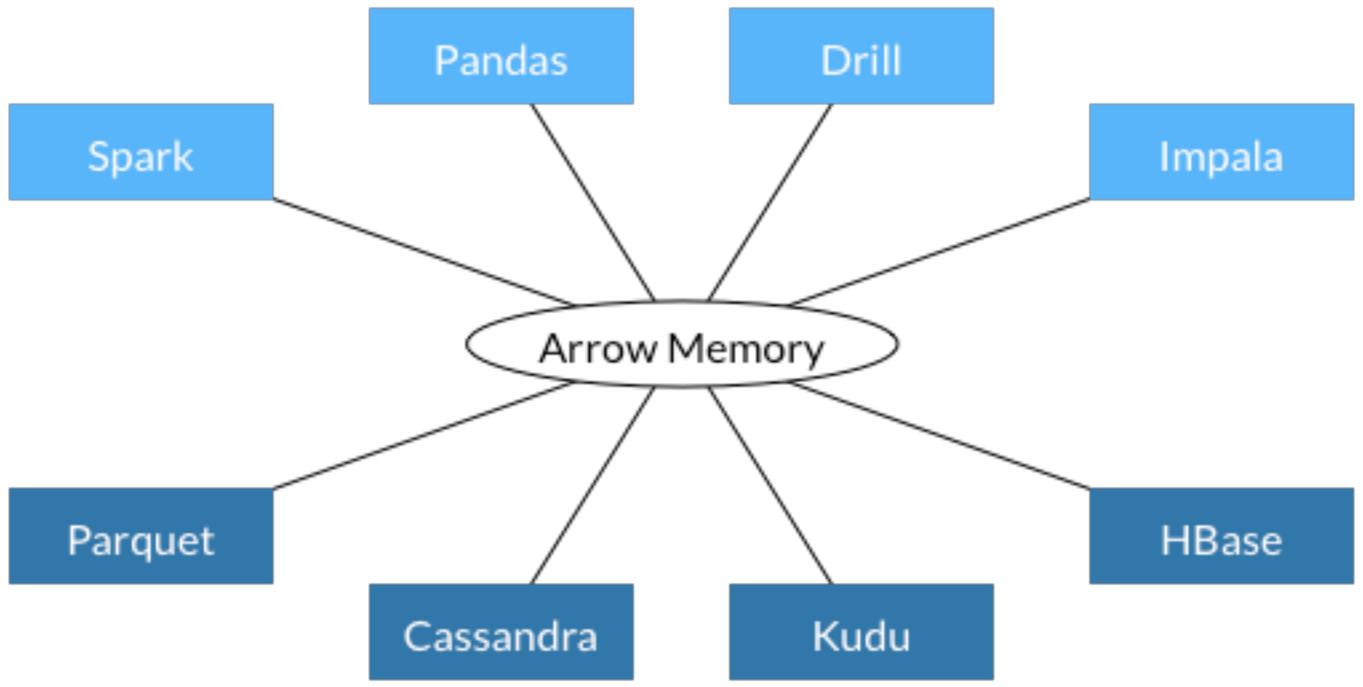
- No serialization and deserialization



immutable arrow batch

Standard Data Format

- Same memory format among languages and systems

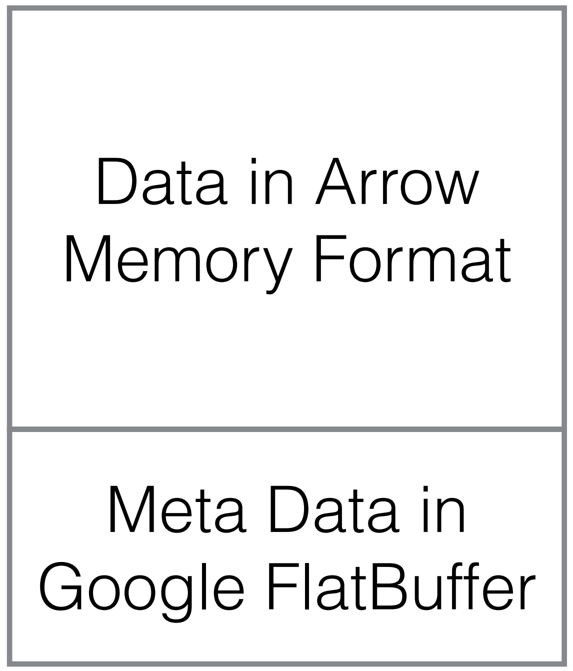


Calcite
Cassandra
Drill
Hadoop
HBase
Ibis
Impala
Kudu
Pandas
Parquet
Phoenix
Spark
Storm
R

http://arrow.apache.org

Arrow in Action: Feather

- Data frame file format powered by Arrow
- Meta data in Google Flatbuffer
- For R and Python
- High performance
- By Wes McKinney (Python) & Hadley Wickham (R)



Installation

- R CRAN

```
install.packages("feather")
```

- R GIT

```
devtools::install_github("wesm/feather/R")
```

- Python

```
pip install feather-format
```

- Conda Python

```
conda install feather-format -c conda-forge
```

- Python on MAC, add

```
export MACOSX_DEPLOYMENT_TARGET=10.9
```

R Demo

```
library(feather)

x <- runif(1e7)

x[sample(1e7, 1e6)] <- NA # 10% NA

df <- as.data.frame(replicate(10, x))

object.size(df) # memory size

system.time(write_feather(df, "test.feather")) # write

system.time(read_feather("test.feather")) # read

data <- read_feather("test.feather")

head(data)
```

Python Demo

```
import feather

import pandas as pd

import numpy as np

arr = np.random.randn(10000000)

arr[::10] = np.nan # 10% nulls

df = pd.DataFrame({'column_{0}'.format(i): arr for i in range(10)})

%time feather.write_dataframe(df, 'test.feather') # Python notebook

%time df = feather.read_dataframe('test.feather')

df.head()
```

Spark Demo

```
import numpy as np

import pandas as pd

import feather

N = 1000000

arr = np.random.randn(N)

df = pd.DataFrame({'data{0}'.format(i): arr

  for i in range(10)

})

sdf = sqlContext.createDataFrame(df)

%time df2 = sdf.toPandas() # Python notebook

%time sdf.write.parquet('test.parquet', mode='overwrite')

%time sdf_parquet = sqlContext.read.parquet('test.parquet')

%time feather.write_dataframe(df, 'test.feather')

%time feather.read_dataframe('test.feather')

%time df.to_csv('test.csv', index=False)

%time df_csv = pd.read_csv('test.csv')

df.head()
```

Readings

- Arrow Project: http://arrow.apache.org
- Arrow Spec Work in Progress: https://github.com/apache/arrow/blob/master/format/Layout.md
- Arrow Source: https://github.com/apache/arrow
- Feather: https://github.com/wesm/feather

Q/A

Thanks