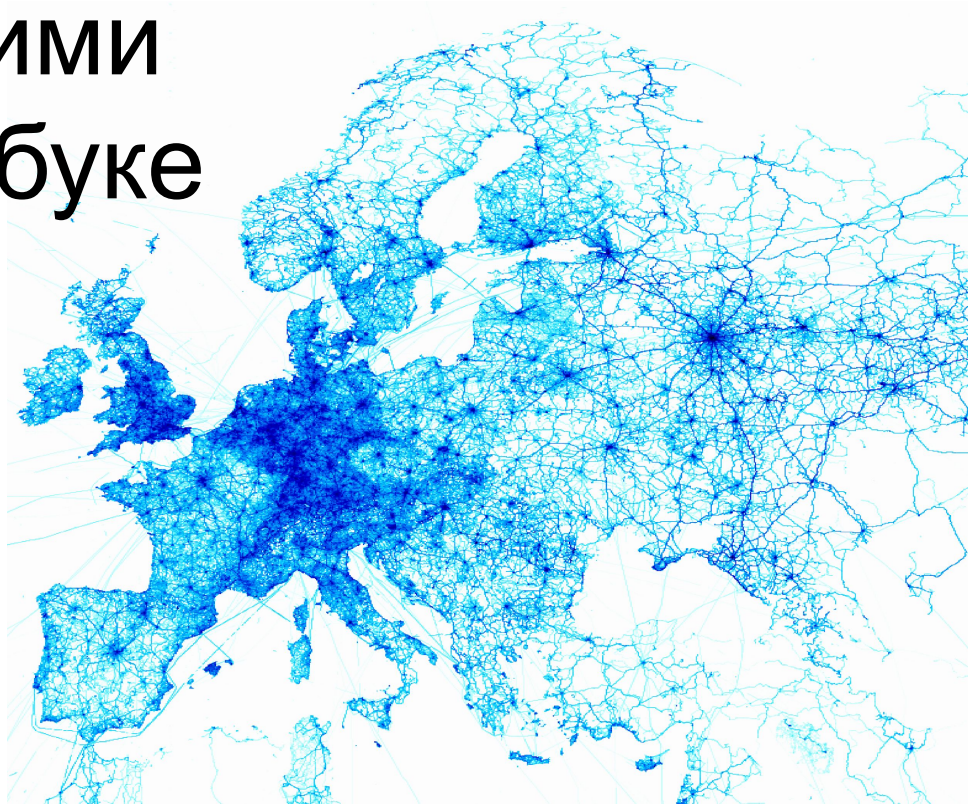
Vaex

Работа с большими данными на ноутбуке



Булат Яминов

Инженер в Uber

Контрибьютор в vaex.io

План презентации

- Что такое Vaex
- Как он работает
- Демо
- Сравнение с другими решениями

Vaex

- vaex-core: библиотека для работы с таблицами данных, как Pandas
- На базе нее:
  - vaex-hdf5 – поддержка формата файлов HDF5
  - vaex-arrow – поддержка Apache Arrow
  - vaex-viz – визуализация (на базе matplotlib)
  - vaex-ml – обертки для популярных алгоритмов машинного обучения: sklearn, annoy, xgboost, lightgbm, catboost
  - vaex-server – работа с удаленными data frames
  - …
- pip install vaex или conda install -c conda-forge vaex

DataFrames

- Pandas
  - dask.dataframe
  - Modin (с использованием Ray)
- SFrame (turi/Apple)
- cudf (nVidia/rapids)
- Spark
  - PySpark
  - Koalas
- Vaex

DataFrames: Vaex

- Эффективная работа с памятью
  - Работа в выражениями, нежели с командами
  - Ленивые вычисления
  - Без копий в памяти
- Использование C++ кода для ускорения и параллелизации
  - Поддержка JIT компиляции через numba/pythran/CUDA
- Работа с memory-mappable форматами файлов
  - Мгновенно открывает файлы размером в терабайт
  - Комфортно работает с миллиардом точек
- Работает на обычном железе
  - Не нужно создавать кластер

Vaex: данные и состояние

```
df = vaex.open('/my_data.hdf5')
```

```
df == {
  'data': {'x': Column(fd='/my_data.hdf5', name='x'), 'y': Column(...)},
  'state': {}
}
```

```
df2 = df[df.y < 5]
```

```
df2 == {
  'data': {'x': Column(fd='/my_data.hdf5', name='x'), 'y': Column(...)},
  'state': {
     'filter': 'y < 5'
  }
}
```

```
df2['z'] = df.x + df.y * 10
```

```
df2 == {
  'data': {'x': Column(fd='/my_data.hdf5', name='x'), 'y': Column(...)},
  'state': {
     'filter': 'y < 5',
     'virtual_columns': {
        'z': 'x + y * 10'
     }
  }
}
```

Демо

Анализ миллиарда поездок такси в Нью-Йорке

Сравнение

На базе статьи [Beyond Pandas: Spark, Dask, Vaex and other big data technologies battling head to head]

Библиотеки:

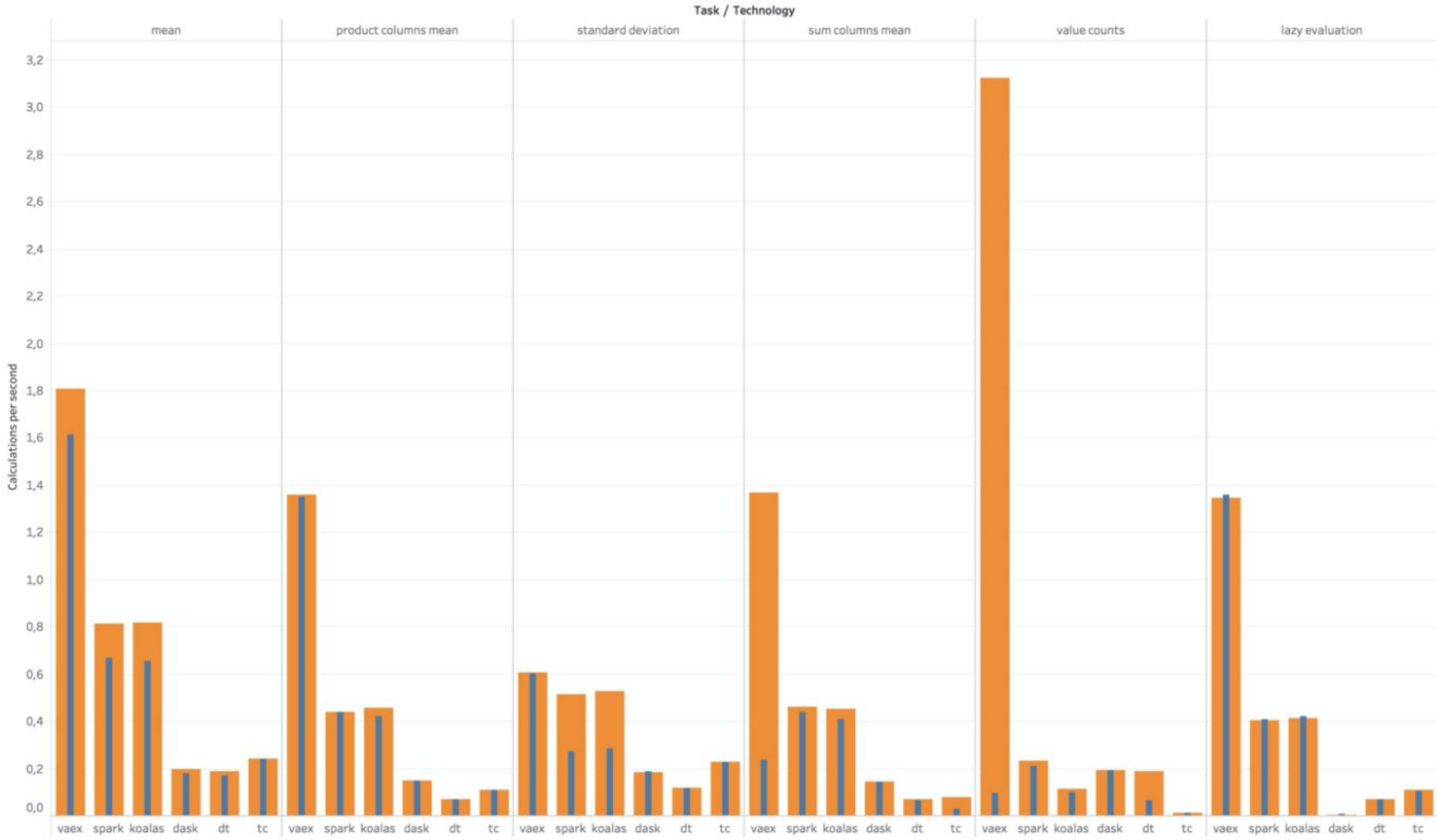
- dask.dataframe
- PySpark
- Koalas
- Vaex
- Turicreate
- Datatable (H2O)

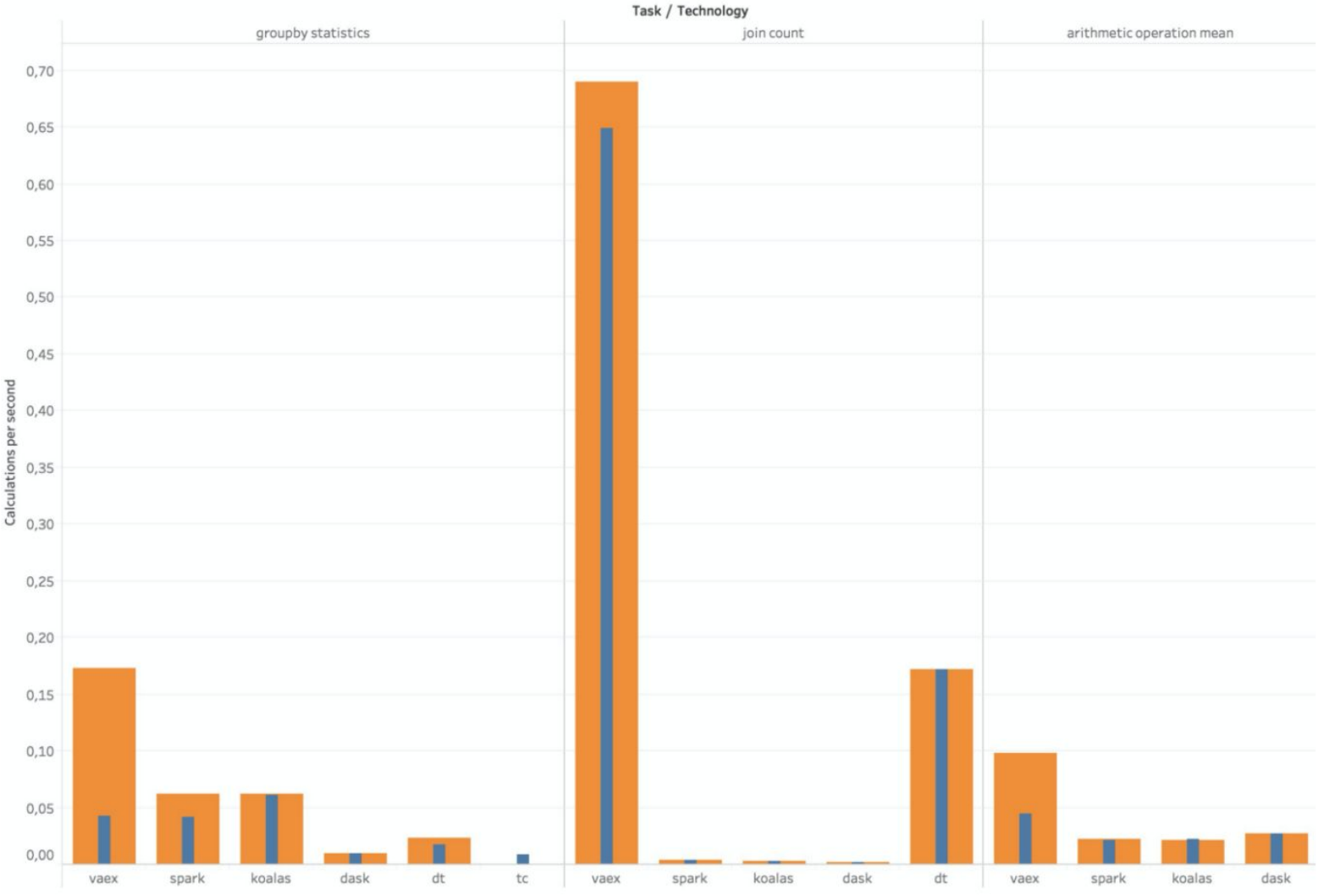
Железо (AWS Sagemaker):

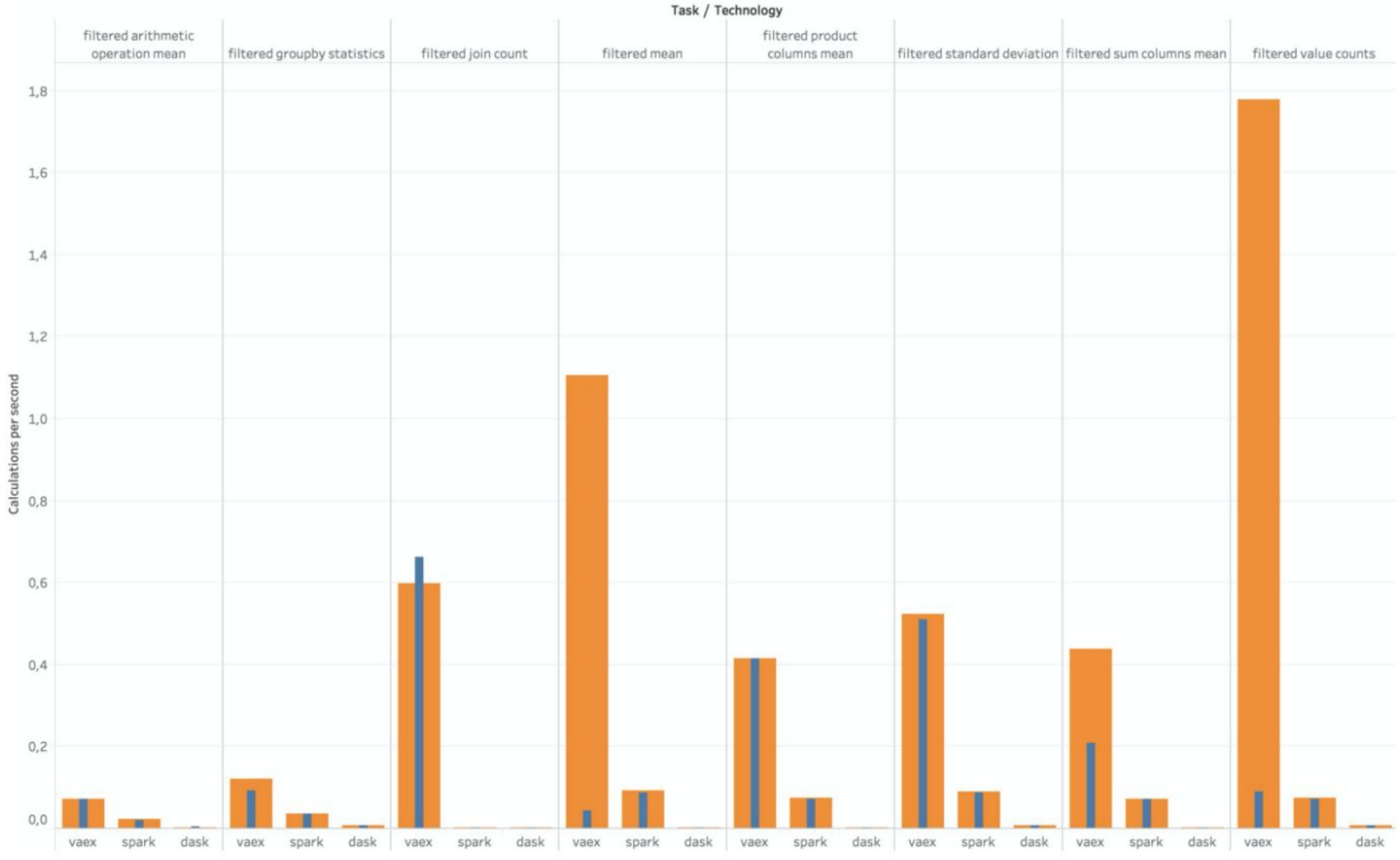
- ml.c5d.4xlarge
- 16 vCPUs
- 32 GB RAM
- 500 GB SSD

Аспекты:

- Простота кода
- Функциональность
- Пайплайны
- Ленивые вычисления
- **Скорость**


Task / Technology
mean
product columns mean
standard deviation
sum columns mean
value counts
lazy evaluation
Calculations per second
3,2
3,0
2,8
2,6
2,4
2,2
2,0
1,8
1,6
1,4
1,2
1,0
0,8
0,6
0,4
0,2
0,0
vaex spark koalas dask dt tc


Task / Technology
groupby statistics
join count
arithmetic operation mean
Calculations per second
0,70
0,65
0,60
0,55
0,50
0,45
0,40
0,35
0,30
0,25
0,20
0,15
0,10
0,05
0,00
vaex
spark
koalas
dask
dt
tc
vaex
spark
koalas
dask
dt
vaex
spark
koalas
dask


Task / Technology
filtered arithmetic operation mean
filtered groupby statistics
filtered join count
filtered mean
filtered product columns mean
filtered standard deviation
filtered sum columns mean
filtered value counts
Calculations per second
1,8
1,6
1,4
1,2
1,0
0,8
0,6
0,4
0,2
0,0
vaex
spark
dask

Заключение

- Vaex позволяет работать с большими данными на ноутбуке
  - 1 миллиард строк в таблице
  - 1 TB данных
- Эффективно работает с памятью
- Быстрее других библиотек

Ресурсы:

- docs.vaex.io / vaex.io
- github.com/vaexio/vaex
- github.com/vaexio/vaex-talks