Методы работы с большими данными

Киреев Василий Сергеевич,
к.т.н., доцент

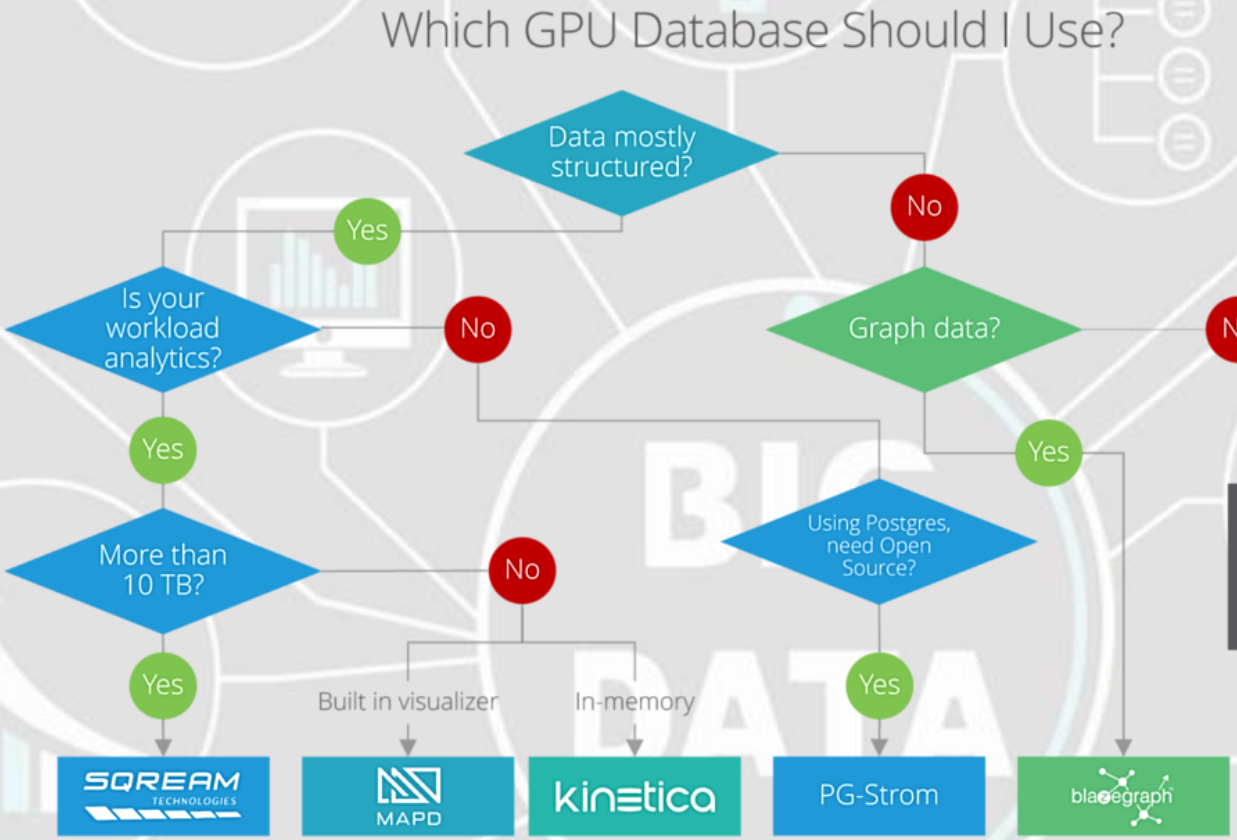
Москва, 2020

GPU и большие данные

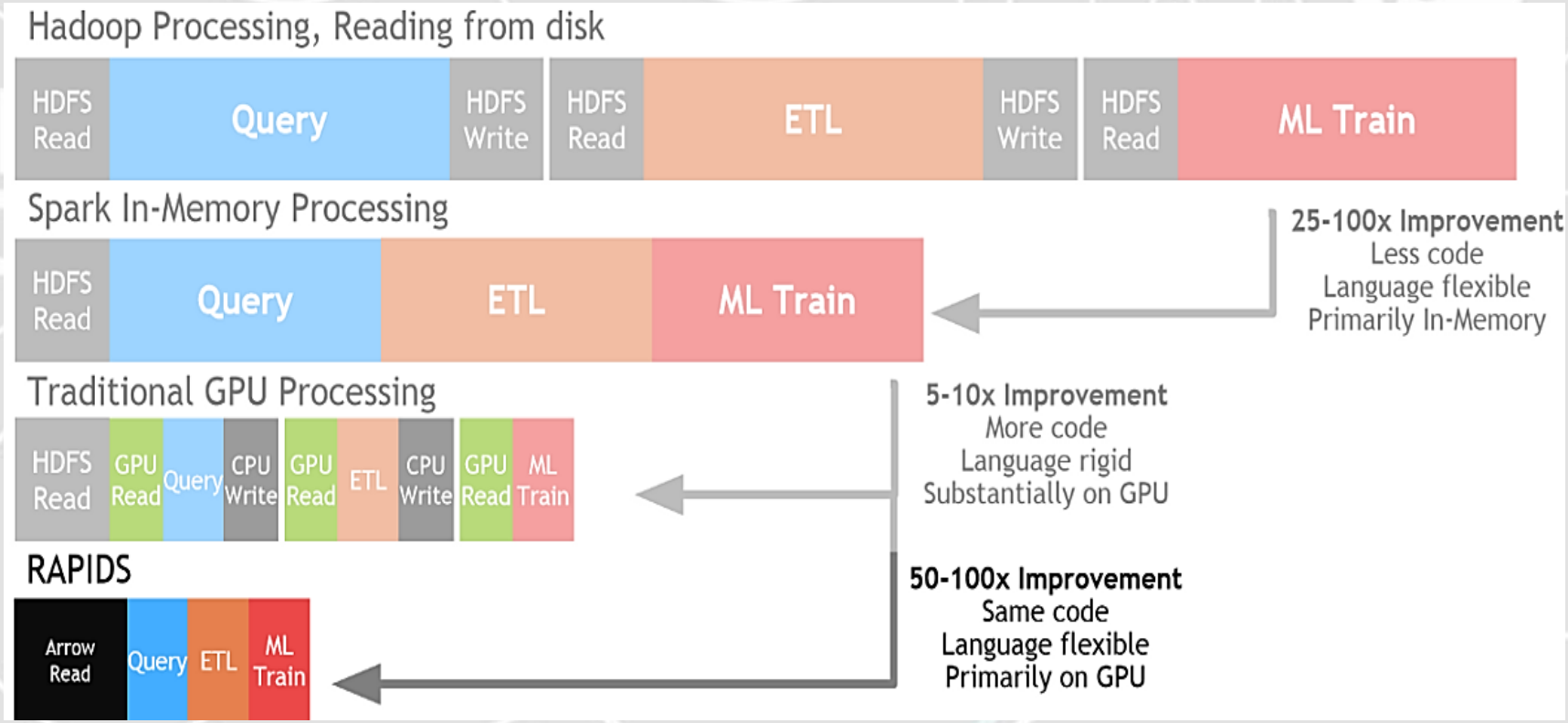
Graphics Processing Unit — это графический процессор широко используемый в настольных и серверных системах. Отличительной особенностью этого устройства является ориентированность на массовые параллельные вычисления. В отличие от графических процессоров архитектура другого вычислительного модуля CPU (Central Processor Unit) предназначена для последовательной обработки данных.

Если количество ядер в обычном CPU измеряется десятками, то в GPU их счет идет на тысячи, что накладывает ограничения на типы выполняемых команд, однако обеспечивает высокую вычислительную производительность в задачах, включающих параллелизм.

GPU в хранении больших данных

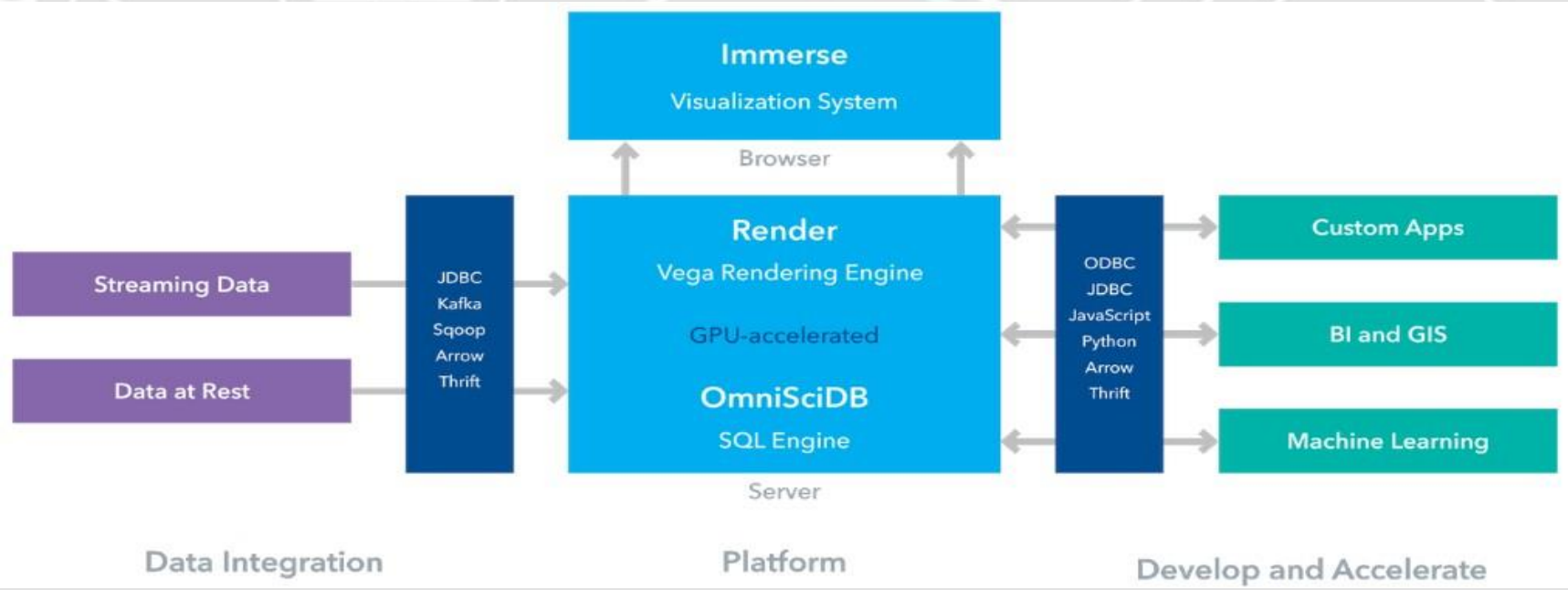


GPU в обработке больших данных

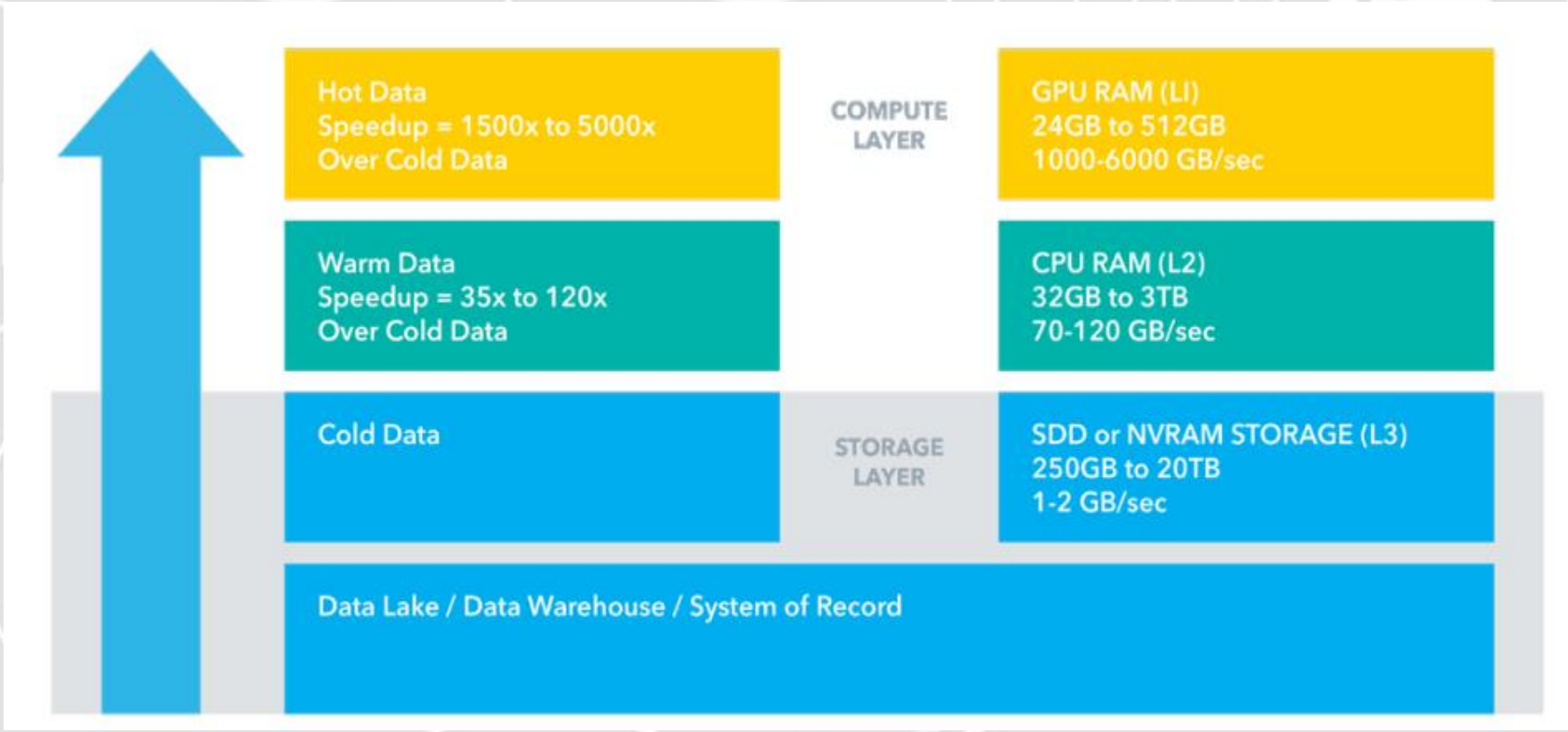


OmniSciDB

OmniSciDB является основой платформы OmniSci. OmniSciDB основана на SQL, является реляционной, столбчатой и специально разработана для использования массового параллелизма современного оборудования CPU и GPU. OmniSciDB может запрашивать до миллиардов строк за миллисекунды и таким образом удовлетворяет требованиям обработки больших, высокоскоростных данных.

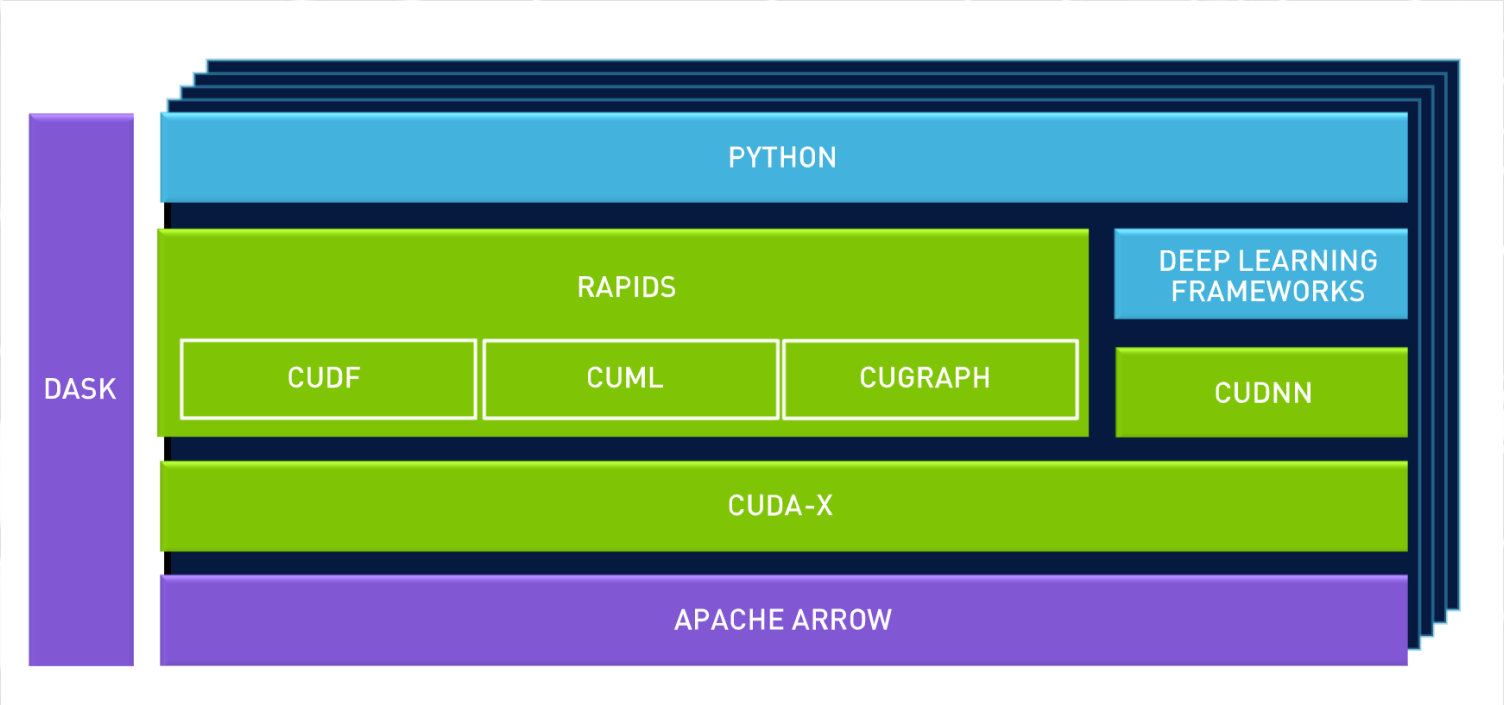


Структура и характеристики OmniSciDB



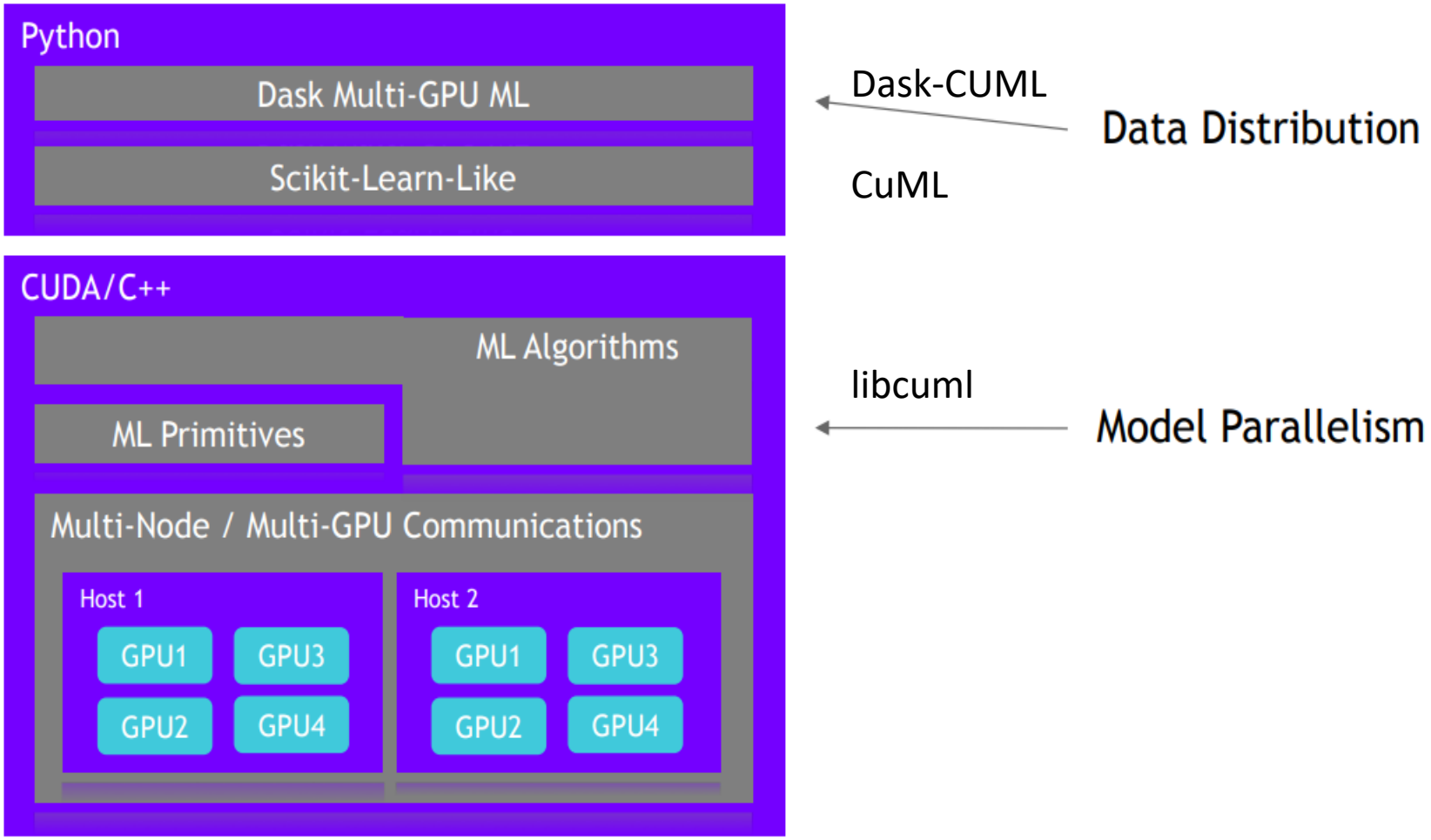
Rapids

RAPIDS включает набор открытых библиотек для анализа, машинного обучения и, визуализации данных с GPU-ускорением. Эта платформа разрабатывалась инженерами NVIDIA в сотрудничестве с ключевыми разработчиками открытого ПО.



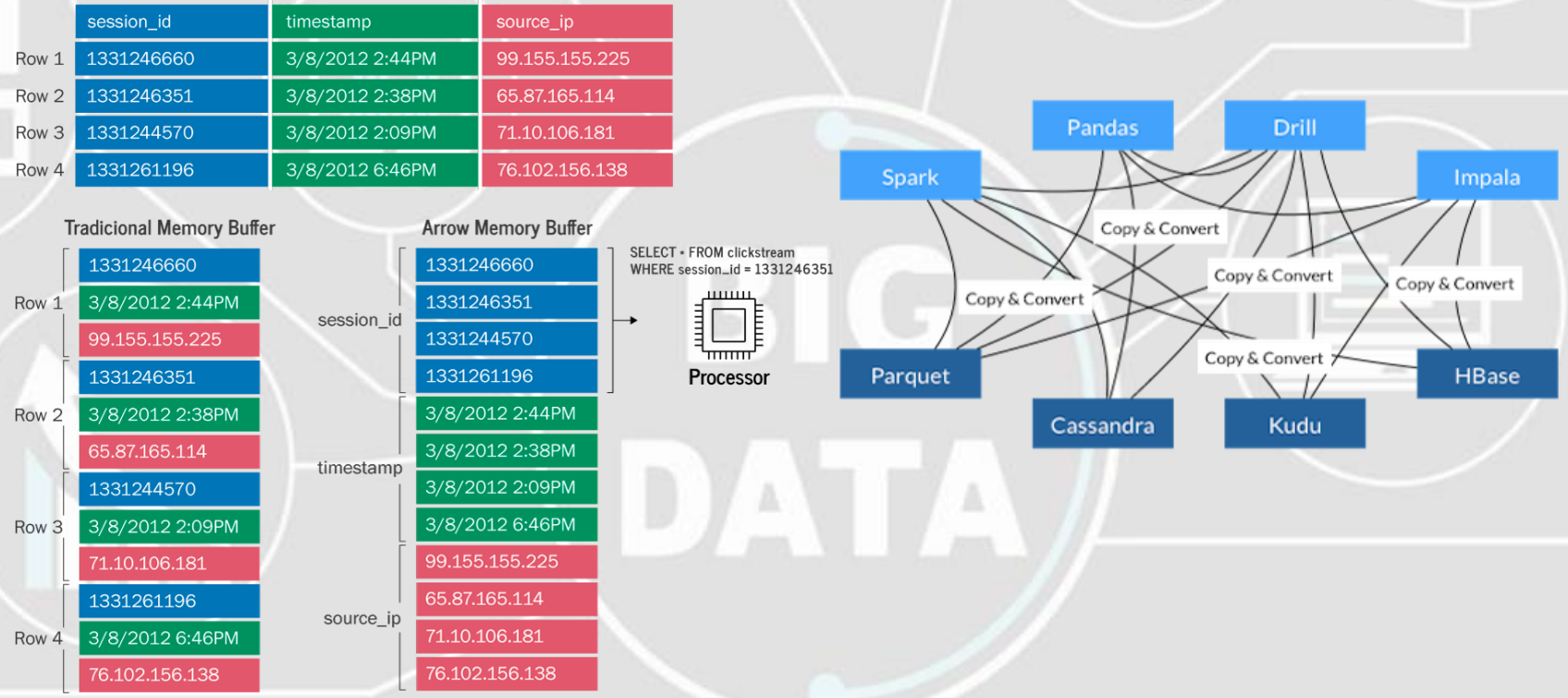
https://rapids.ai/

Высокоуровневые API



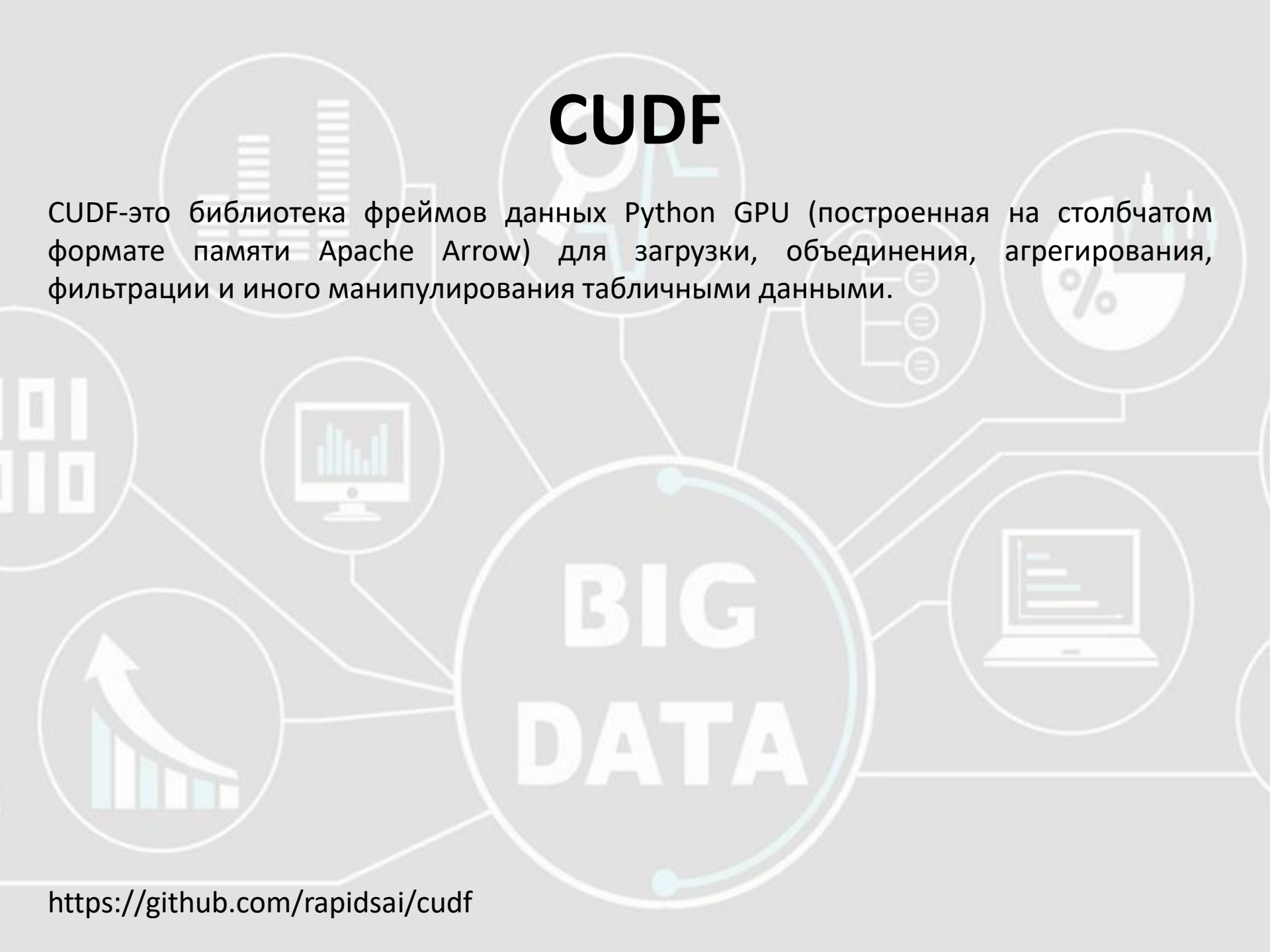
Arrow

Apache Arrow - это многоязычная платформа для разработки данных в памяти. Она определяет стандартизированный независимый от языка формат колоночной памяти для плоских и иерархических данных, организованных для эффективных аналитических операций на современном оборудовании.



CUDF

CUDF-это библиотека фреймов данных Python GPU (построенная на столбчатом формате памяти Apache Arrow) для загрузки, объединения, агрегирования, фильтрации и иного манипулирования табличными данными.



https://github.com/rapidsai/cudf

Типы данных CUDF

Kind of Data	Data Type	Scalar	String Aliases
Integer		np.int8, np.int16, np.int32, np.int64, np.uint8, np.uint16, np.uint32, np.uint64	`'int8'`, `'int16'`, `'int32'`, `'int64'`, `'uint8'`, `'uint16'`, `'uint32'`, `'uint64'`
Float		np.float32, np.float64	`'float32'`, `'float64'`
Strings		str	`'string'`, `'object'`
Datetime		np.datetime64	`'datetime64[s]'`, `'datetime64[ms]'`, `'datetime64[us]'`, `'datetime64[ns]'`
Timedelta (duration type)		np.timedelta64	`'timedelta64[s]'`, `'timedelta64[ms]'`, `'timedelta64[us]'`, `'timedelta64[ns]'`
Categorical	CategoricalDtype	(none)	`'category'`
Boolean		np.bool	`'bool'`

Примеры работы в CUDF

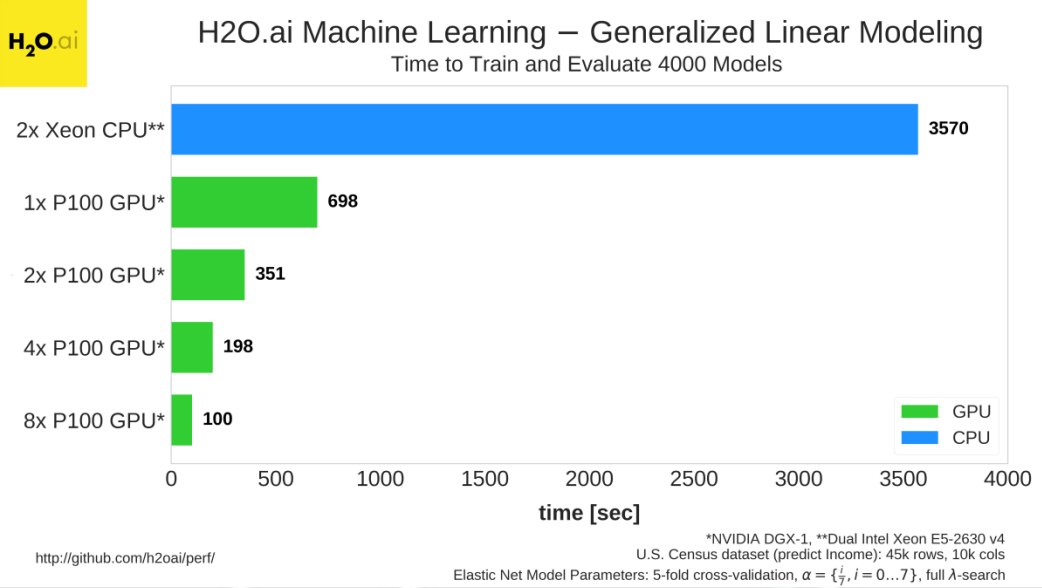
```
import cudf, io, requests
from io import StringIO

url = "https://github.com/plotly/datasets/raw/master/tips.csv"
content = requests.get(url).content.decode('utf-8')

tips_df = cudf.read_csv(StringIO(content))
tips_df['tip_percentage'] = tips_df['tip'] / tips_df['total_bill'] * 100

# display average tip by dining party size
print(tips_df.groupby('size').tip_percentage.mean())
```

Output:

```
size
1    21.729201548727808
2    16.571919173482897
3    15.215685473711837
4    14.594900639351332
5    14.149548965142023
6    15.622920072028379
Name: tip_percentage, dtype: float64
```

H2O4GPU

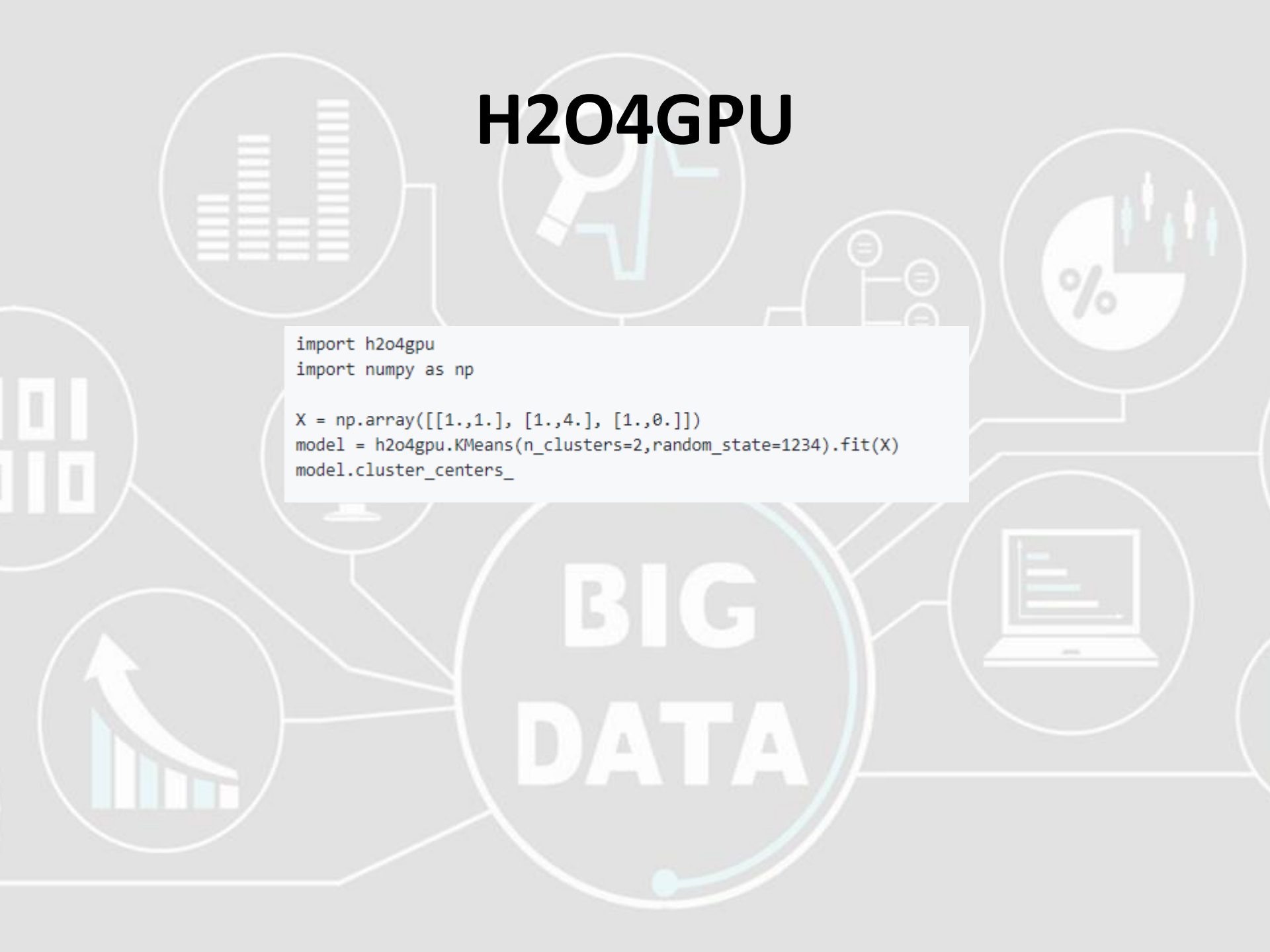
H2O4GPU-это пакет машинного обучения с открытым исходным кодом, ускоренный GPU, с API на Python и R, который позволяет любому пользователю использовать преимущества графических процессоров для создания передовых моделей машинного обучения. Существует множество популярных алгоритмов, включая машины градиентного бустинга (GBM), обобщенные линейные модели (GLM) и кластеризацию K-средних.



https://github.com/h2oai/h2o4gpu

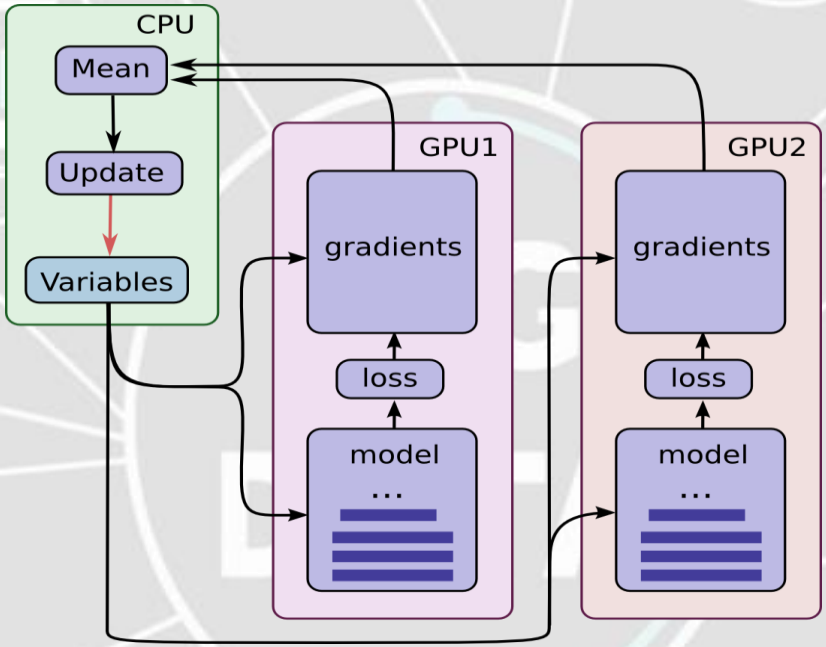
H2O4GPU

```
import h2o4gpu
import numpy as np

X = np.array([[1.,1.], [1.,4.], [1.,0.]])
model = h2o4gpu.KMeans(n_clusters=2,random_state=1234).fit(X)
model.cluster_centers_
```



Tensorflow-GPU

TensorFlow-это библиотека программного обеспечения с открытым исходным кодом для высокопроизводительных численных вычислений. Ее гибкая архитектура позволяет легко развертывать вычисления на различных платформах (процессоры, графические процессоры, TPU), а также на настольных компьютерах, кластерах серверов и мобильных и периферийных устройствах.



https://www.tensorflow.org/

Tensorflow-GPU

```
!pip -q install tensorflow==1.13.1
!pip install tensorflow-gpu==1.13.1
!pip -q install opencv-python
!pip -q install keras==2.3.0
!pip -q install ImageAi==2.1.5
!wget https://github.com/OlafenwaMoses/ImageAI/releases/download/1.0/resnet50_coco_best_v2.0.1.h5
!python -c 'from tensorflow.python.client import device_lib;print(device_lib.list_local_devices());'
```

```
from imageai.Prediction.Custom import ModelTraining
model_trainer = ModelTraining()
model_trainer.setModelTypeAsResNet()
model_trainer.setDataDirectory("/content/odezhda")
model_trainer.trainModel(num_objects=6, num_experiments=100, enhance_data=True, batch_size=32, show_network_summary=True)
```