DATA SCIENCE SUMMIT

sponsor diamentowy:

accenture

SAMSUNG

sponsorzy srebrni:

lingaro Nethone VERTICA

Pyt(h)on vs słoń: aktualny stan przetwarzania dużych danych w Python

Jakub Nowacki, SigDelta/Sages

whoami

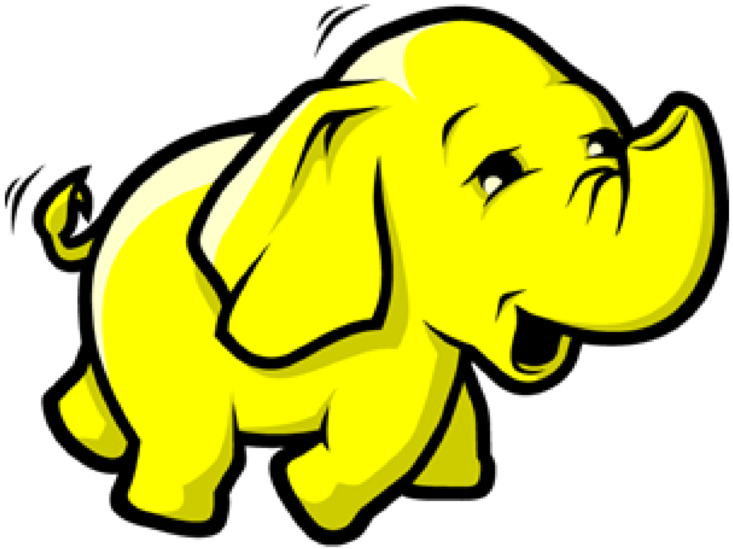
Lead Data Scientist @ SigDelta (sigdelta.com)

Trainer @ Sages (sages.com.pl)

I can code, I do maths

@jsnowacki

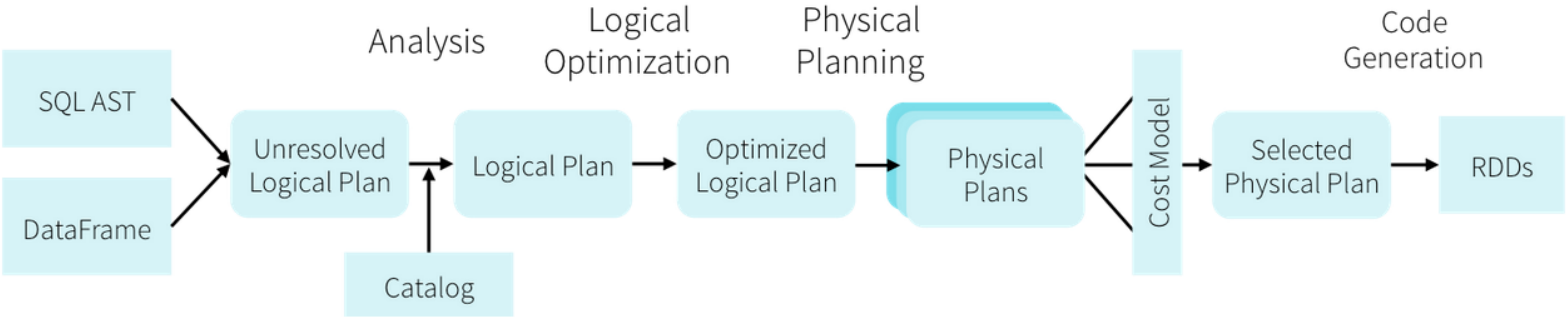
Jak to było kiedyś?



Apache Spark!

Spark SQL - DataFrame

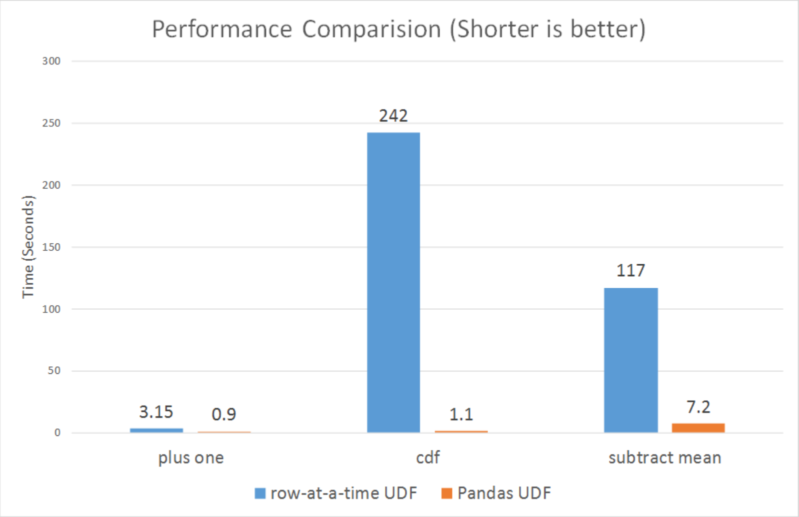
```
from pyspark.sql.functions import *

spark.read.text('hdfs://...’) \
        .select(explode(split('value', '\W+')).alias('word')) \
        .groupBy('word') \
        .count() \
        .orderBy(desc('count')) \
        .write.parquet('hdfs://...')
```



Źródło: https://databricks.com/blog/2015/03/24/spark-sql-graduates-from-alpha-in-spark-1-3.html

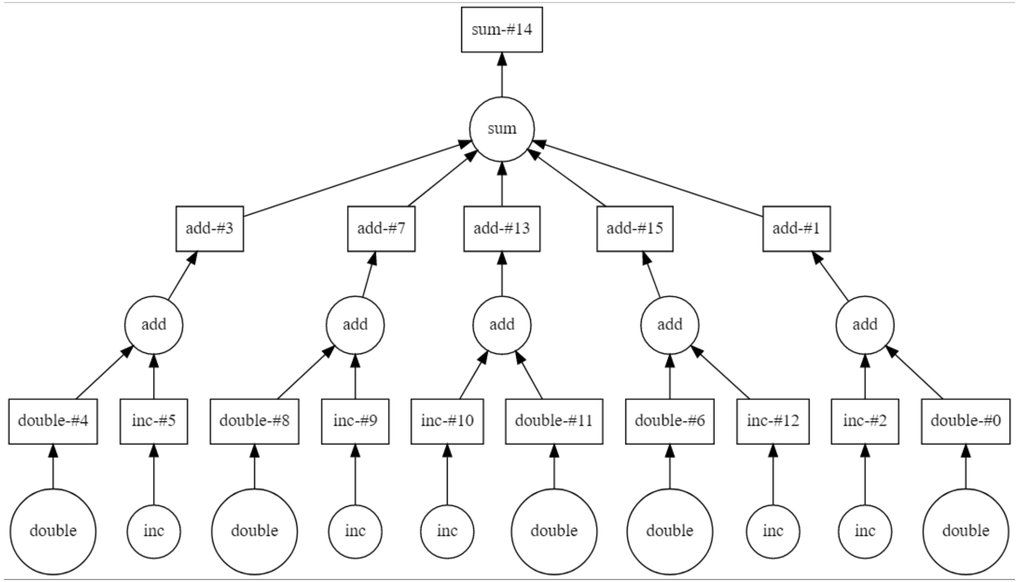
Vectorized UDFs

```
import pandas as pd
from pyspark.sql.types import LongType
def multiply_func(a, b):
      return a * b
multiply = pandas_udf(multiply_func,
                      returnType=LongType())
pdf = pd.DataFrame([1, 2, 3], columns=["x"])
print(multiply_func(pdf.x, pdf.x))
# 0 1
# 1 4
# 2 9
# dtype: int64
df = spark.createDataFrame(pdf)
df.select(multiply(col("x"), col("x"))).show()
# +-------------------+
# |multiply_func(x, x)|
# +-------------------+
# | 1| # | 4| # | 9|
# +-------------------+
```



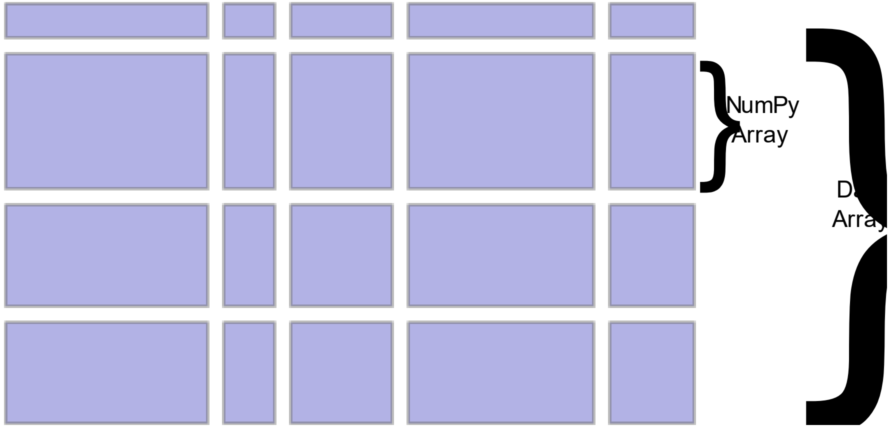
Źródło:
https://databricks.com/blog/2017/10/30/introducing-vectorized-udfs-for-pyspark.html

Dask!



Źródło: https://dask.pydata.org/

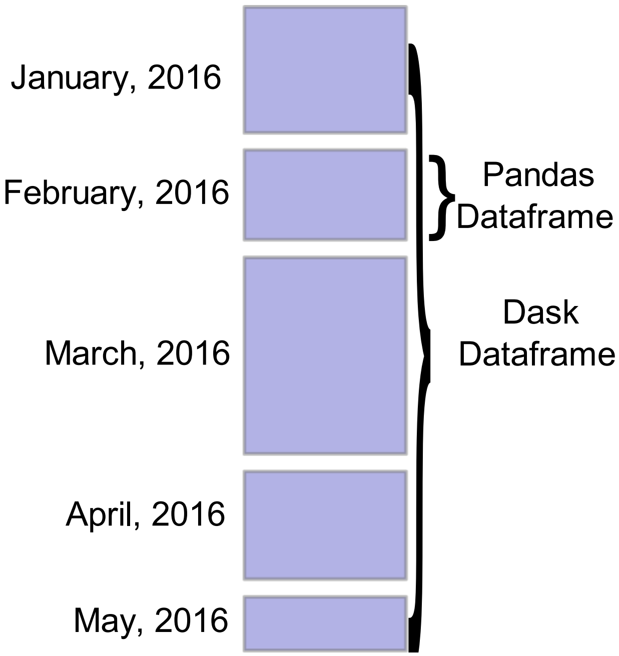
Dask Array



```
import dask.array as da
import numpy as np

x = da.ones(10, chunks=(5,))
y = np.ones(10)
z = x + y
print(z)
# dask.array<add, shape=(10,),
#  … dtype=float64, chunksize=(5,)>
```

Źródło: https://dask.pydata.org/

Dask DataFrame



```
import dask.dataframe as dd

posts = dd.read_parquet('data/posts_tags.parq’)\
       .set_index('id’)
posts_count = posts.creation_date.dt.date\
       .value_counts()

posts_count_df = posts_count.compute()
posts_count_df.head()

# 2017-08-23 9531
# 2017-07-27 9450
# 2017-08-24 9366
# 2017-08-03 9345
# 2017-03-22 9342
# Name: creation_date, dtype: int64
```

Źródło: https://dask.pydata.org/

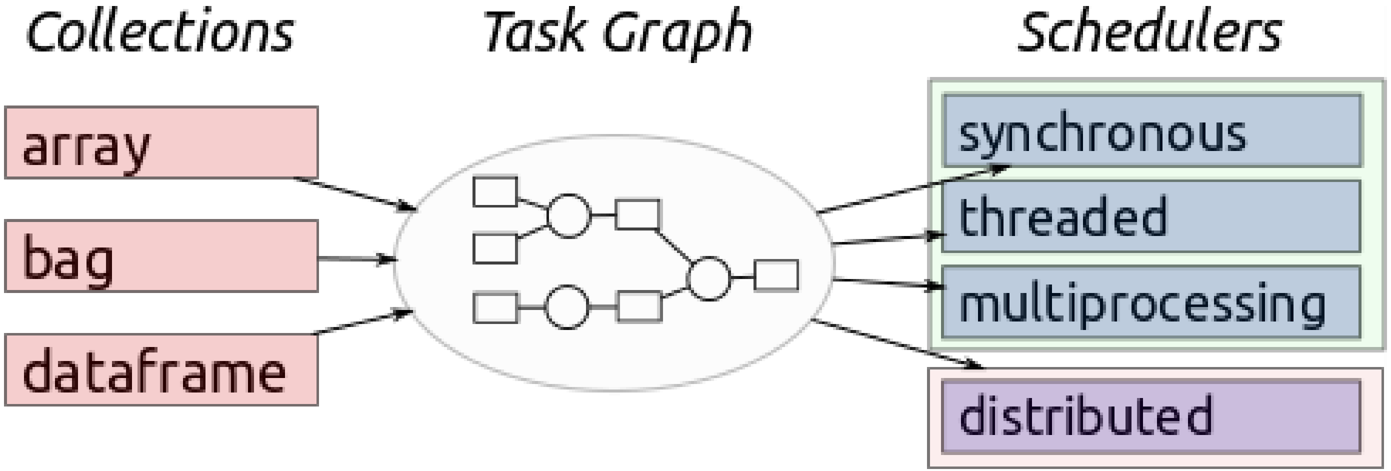
Przykład: http://sigdelta.com/blog/stackpverflow-tags-with-dask/

Dask Bag

```
import dask.bag as db

tags_xml = db.read_text('data/Tags.xml', encoding='utf-8’)
tags_xml.take(5)
# ('\ufeff<?xml version="1.0" encoding="utf-8"?>\n’,
#  '<tags>\n’,
#  ' <row Id="1" TagName=".net" Count="257092" … />\n’,
#  ' <row Id="2" TagName="html" Count="683981" … />\n’,
#  ' <row Id="3" TagName="javascript" Count="1457944" … />\n’)
tags_rows = tags_xml.filter(lambda line: line.find('<row') >= 0)
tags_rows.take(5)
# (' <row Id="1" TagName=".net" Count="257092" … />\n’,
#  ' <row Id="2" TagName="html" Count="683981" … />\n’,
#  ' <row Id="3" TagName="javascript" Count="1457944" … />\n’,
#  ' <row Id="4" TagName="css" Count="490198" … />\n’,
#  ' <row Id="5" TagName="php" Count="1114030" … />\n’)
tags = tags_rows.map(extract_tags_columns).to_dataframe()
```

Przykład: http://sigdelta.com/blog/dask-introduction/

Na jednej maszynie lub wielu



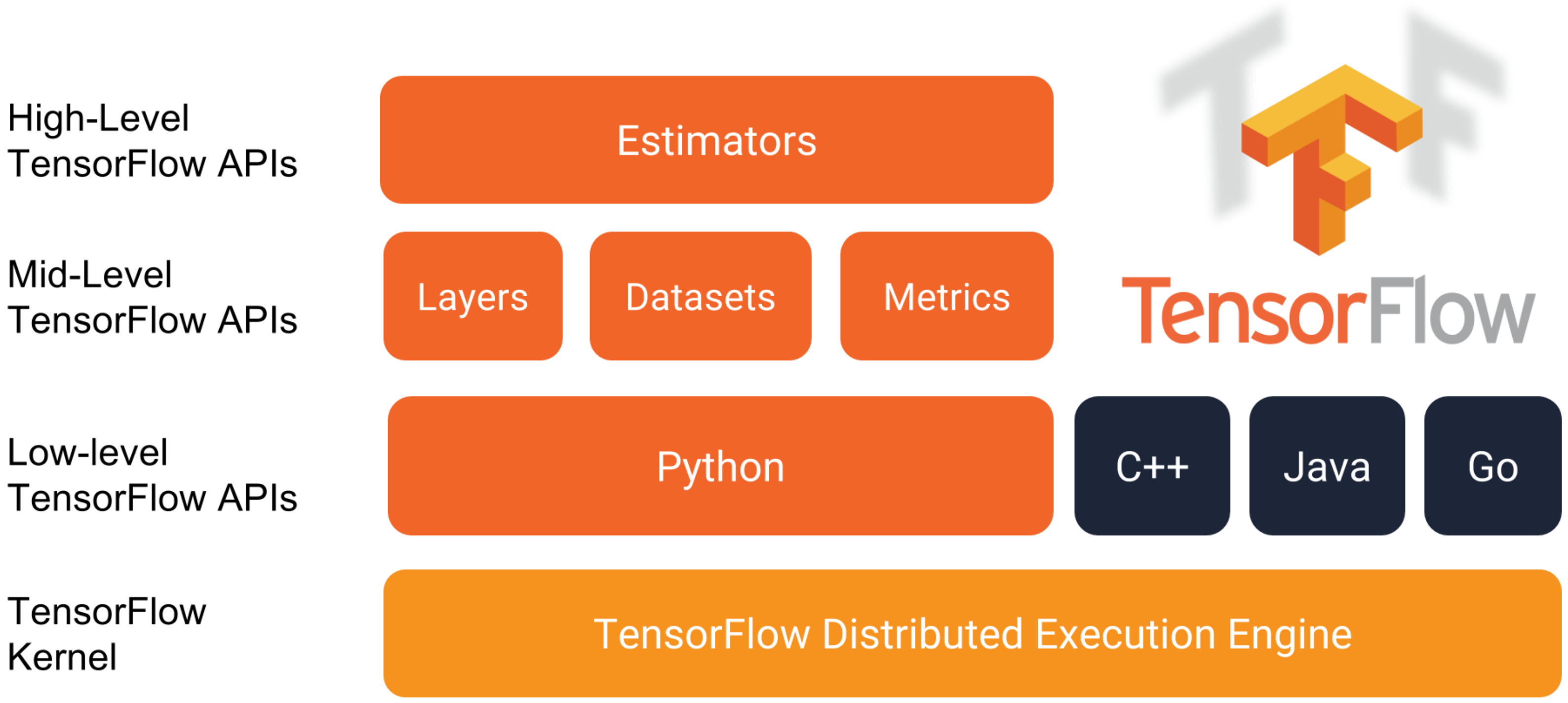
Źródło: https://dask.pydata.org/

Dask?!

```
...

t.reset_index().head()

# ---------------------------------------------------------------------------
# ValueError Traceback (most recent call last)
# <ipython-input-100-e6186d78fb03> in <module>()
# ----> 1 t.reset_index().head()
#
# ...
#
# ValueError: Length mismatch: Expected axis has 3 elements, new
#  values have 2 elements
```

Źródło: https://github.com/dask/dask/issues/3038 (naprawione)

TensorFlow



High-Level TensorFlow APIs

Estimators

Mid-Level TensorFlow APIs

Layers

Datasets

Metrics

TensorFlow

Low-level TensorFlow APIs

Python

C++

Java

Go

TensorFlow Kernel

TensorFlow Distributed Execution Engine

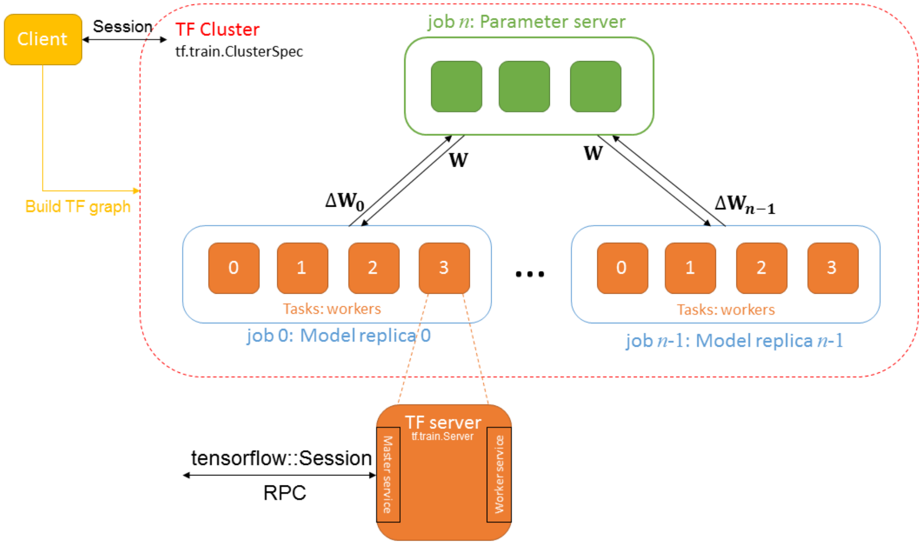
Źródło: https://www.tensorflow.org/get_started/premade_estimators

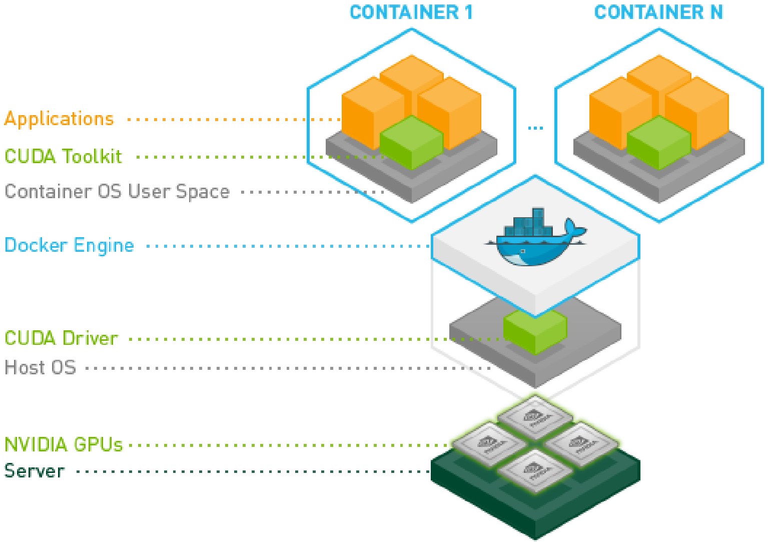
TensorFlow Data

```
dataset2 = tf.data.Dataset.from_tensor_slices(
       (tf.random_uniform([4]),
        tf.random_uniform([4, 100], maxval=100, dtype=tf.int32)))
print(dataset2.output_types) # ==> "(tf.float32, tf.int32)"
print(dataset2.output_shapes) # ==> "((), (100,))"

dataset3 = tf.data.Dataset.zip((dataset1, dataset2))
print(dataset3.output_types) # ==> (tf.float32, (tf.float32, tf.int32))
print(dataset3.output_shapes) # ==> "(10, ((), (100,)))"

dataset1 = dataset1.map(lambda x: ...)
dataset2 = dataset2.flat_map(lambda x, y: ...)
dataset3 = dataset3.filter(lambda x, (y, z): ...)
```

Źródło: https://www.tensorflow.org/programmers_guide/datasets

TensorFlow GPU & Distributed

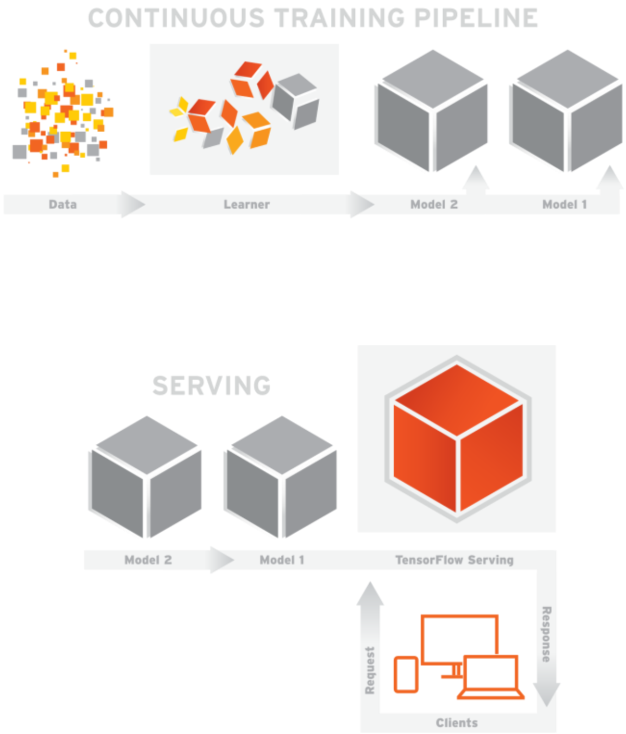




Źródło: http://www.pittnuts.com/2016/08/glossary-in-distributed-tensorflow/

Źródło: https://towardsdatascience.com/using-docker-to-set-up-a-deep-learning-environment-on-aws-6af37a78c551

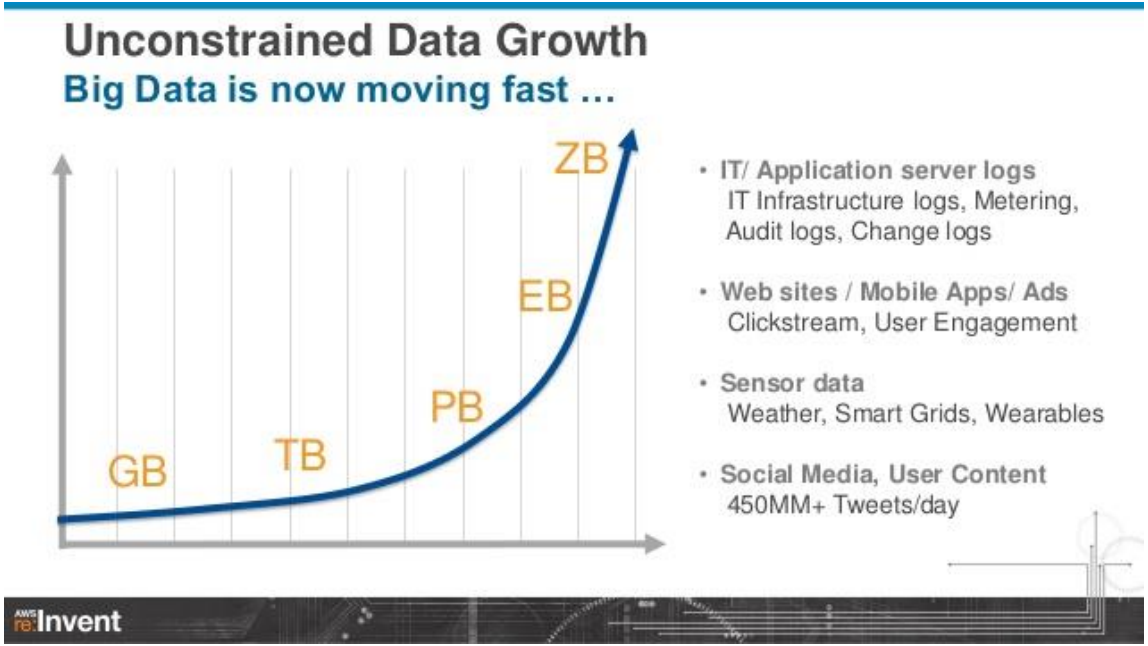
TensorFlow Serving



Źródło: https://www.tensorflow.org/serving/

Źródło: https://cloud.google.com/products/machine-learning/

Co przyniesie przyszłość? ¯_(ツ)_/¯



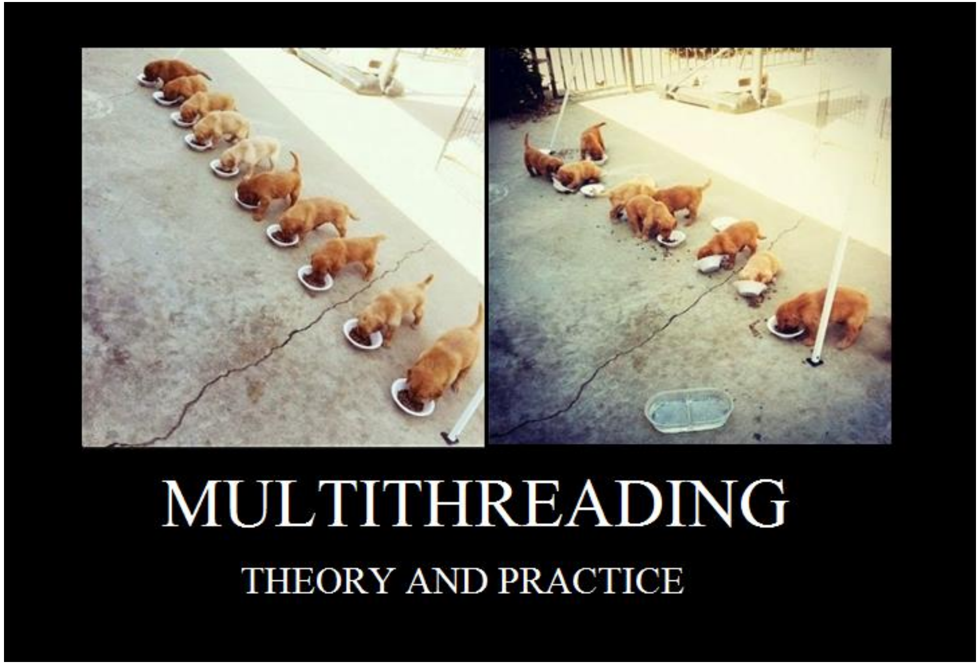
Unconstrained Data Growth

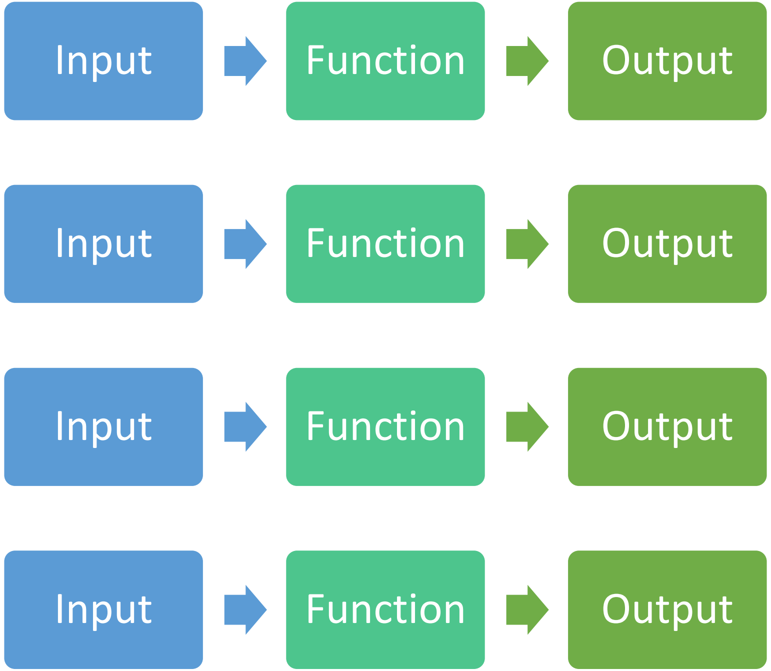
Big Data is now moving fast …

- **IT/ Application server logs**
  IT Infrastructure logs, Metering, Audit logs, Change logs
- **Web sites / Mobile Apps/ Ads**
  Clickstream, User Engagement
- **Sensor data**
  Weather, Smart Grids, Wearables
- **Social Media, User Content**
  450MM+ Tweets/day

Źródło: https://www.slideshare.net/AmazonWebServices/introducing-amazon-kinesis-realtime-processing-of-streaming-big-data-bdt103-aws-reinvent-2013

Programowanie funkcyjne





SQL

APACHE Spark™

HIVE

presto

aws

Azure

Google Cloud

pandas

$y_it = \beta' x_{it} + \mu_i + \epsilon_{it}$

DASK

DATA SCIENCE SUMMIT

sponsor diamentowy:

accenture

sponsorzy srebrni:

SAMSUNG lingaro Nethone VERTICA

Dziękuję za uwagę!

Pytania?