Is your Python too slow?

Hardware and software for accelerating data science

Data Science Salon Virtual
September 2020

Aaron Richter

Saturn Cloud

Hi!

Aaron Richter, PhD

Senior Data Scientist @ Saturn Cloud

> *I work to make data scientists faster and happier*

aaron@saturncloud.io
rikturr.com
@rikturr

Saturn Cloud

Saturn Cloud

Bringing together the fastest hardware + OSS

DASK

- Pythonic parallelism
- Rapidly scale PyData

RAPIDS

- Multi-GPU computing
- The future of HPC

PREFECT

- Workflow orchestration
- Flow insight and mgmt

kubernetes

- Fast setup
- Enterprise secure

Data science with Python

Saturn Cloud

Data science with Python

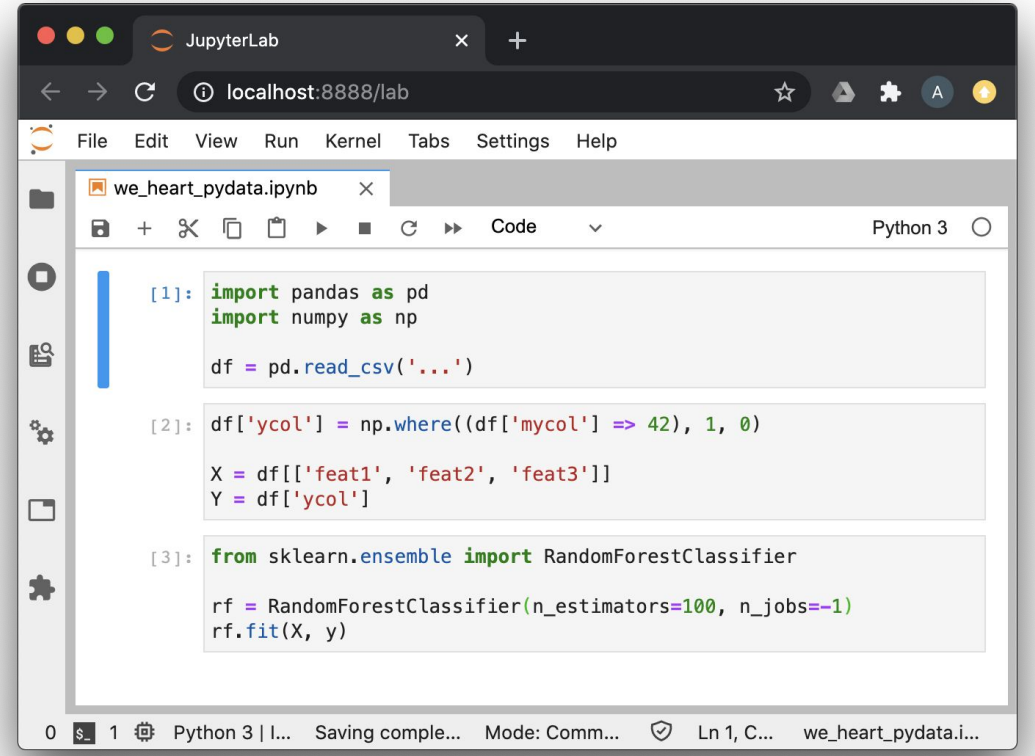
PyData

pandas

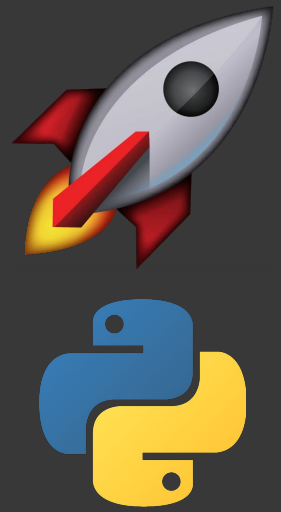
scikit learn

NumPy

jupyter



Accelerating data science in Python



Saturn Cloud

Hardware

Bigger data and compute requires more resources

Laptop / workstation
<64GB RAM
<12 cores
<1 GPU

Up-front cost
Expensive to upgrade

Cloud machine
1+ TB RAM
100+ cores
8+ GPUs

Pay per second
Switch machines easily

Cloud cluster
1000s of machines

Cloud machine, but lots of them!

Software

Considerations for adopting OSS data science tools

- Origin / maturity
- Community / usage
- Contributing
- “Native” support
- Ecosystem / compatibility
- Deployment / installation

Software

Data science workloads

- Arrays (multi-dim)
- DataFrames
- Machine learning
  - Classification & regression
  - Hyperparameter tuning
- Deep learning
- Arbitrary (custom) code

Software

PyData world

Software

Big data world

PyData world

Software

Big data world

PyData world

Python + big data!

Spark

- *Distributed computing based on MapReduce*
- U.C. Berkeley AMPLab ~2010
- Built in Scala; APIs for Java, SQL, R, Python
- Mature, big data / data eng communities
- Native DataFrames, streaming, ML, graph
- Integrates with Hadoop ecosystem
  - Python/Pandas limited with UDFs
- Cluster deployment is complex

http://spark.apache.org/

Dask

- *Distributed computing based on dynamic task scheduling*
- Anaconda, ~2015
- Built in Python; Python API
- Mature, scientific computing communities
- Low-level task library
- High-level libraries for DataFrames, arrays, ML
- Integrates with PyData ecosystem
- Runs on laptop, scales to clusters

https://dask.org/

Ray

- *Distributed computing: task/actor based*
- U.C. Berkeley RISELab, ~2016
- Built in Python/C++; APIs for Python, Java
- Low-level task library similar to Dask
- High-level libraries focused on reinforcement learning, hyperparameter tuning
- Runs on laptop, scales to clusters
  - Windows support in alpha

https://github.com/ray-project/ray

RAPIDS

- *GPU accelerated data science*
- NVIDIA, ~2018
- Built in C++(CUDA), Python; Python API
- Large dev team, support from NVIDIA
- Native DataFrames, arrays, ML, graph, streaming, spatial
- Integrates with PyData ecosystem
- Scales to clusters with Dask integration

https://rapids.ai/

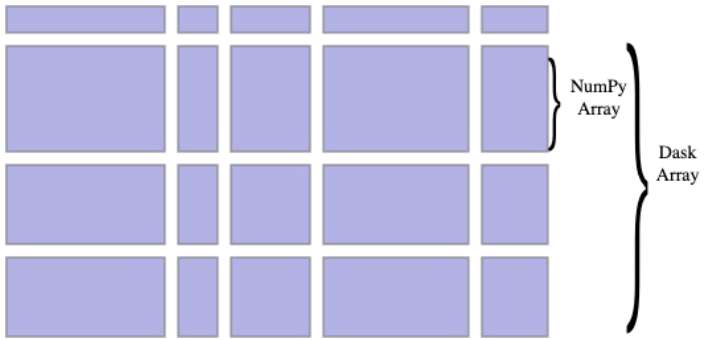
DataFrames

- PySpark DataFrame
- Dask DataFrame
- Koalas (Spark backend)
- Modin (Ray or Dask backend)
- Vaex
- RAPIDS (cuDF)

https://koalas.readthedocs.io/en/latest/
https://modin.readthedocs.io/en/latest/
https://vaex.readthedocs.io/en/latest/

Multi-dimensional arrays

- PySpark: not supported
- Dask: dask.array, xarray
- Ray: possible with task interface
- cuPy (GPU/CUDA)



Machine learning

- PySpark: ML (linear, tree, etc)
- Ray: distributed DL training, reinforcement learning, hyperparameter tuning
- Dask: extend scikit-learn, hyperparameter tuning, XGBoost
- RAPIDS (cuML): scikit-learn parity, XGBoost

Unofficial guide to accelerating Python

- Use “traditional” PyData tools on your laptop until you can’t

Unofficial guide to accelerating Python

- Use “traditional” PyData tools on your laptop until you can’t
- **Then, use parallel computing / out of core tools *on your laptop***
  - Dask, Ray, Modin, Vaex
  - RAPIDS if you have a GPU

Unofficial guide to accelerating Python

- Use “traditional” PyData tools on your laptop until you can’t
- Then, use parallel computing / out of core tools *on your laptop*
  - Dask, Spark, Ray, Modin, Vaex
  - RAPIDS if you have a GPU
- **Then, get a bigger machine in the cloud**

Unofficial guide to accelerating Python

- Use “traditional” PyData tools on your laptop until you can’t
- Then, use parallel computing / out of core tools *on your laptop*
  - Dask, Spark, Ray, Modin, Vaex
  - RAPIDS if you have a GPU
- Then, get a bigger machine in the cloud
- **Then, use a cluster in the cloud**
  - Dask, Spark, Ray

Unofficial guide to accelerating Python

- Use “traditional” PyData tools on your laptop until you can’t
- Then, use parallel computing / out of core tools *on your laptop*
  - Dask, Spark, Ray, Modin, Vaex
  - RAPIDS if you have a GPU
- Then, get a bigger machine in the cloud
- Then, use a cluster in the cloud
  - Dask, Spark, Ray
- ***Use the best tool for each workload!***

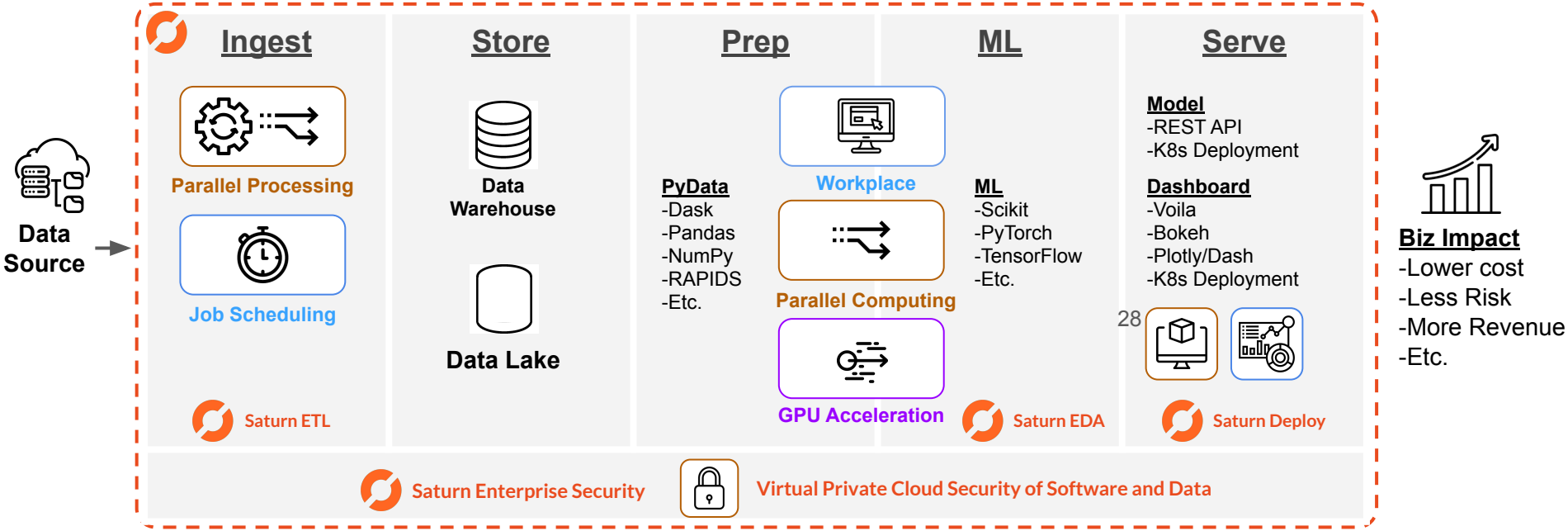
Resources

- Videos
  - Is Spark still relevant? Multi-node CPU and single-node GPU workloads
  - High-Performance Data Science at Scale with RAPIDS, Dask, and GPUs
  - Ray: A System for Scalable Python and ML
  - Scaling Interactive Pandas Workflows with Modin
  - Dask: A Pythonic Distributed Data Science Framework
- Articles
  - Ray comparison to Dask
  - Scaling Pandas: Comparing Dask, Ray, Modin, Vaex, and RAPIDS

Saturn Cloud

Saturn Cloud

Saturn enables end-to-end DS and ML in Python



Saturn Cloud

https://saturncloud.io

[7 day free trial!]

@saturn_cloud