Pearson

Using the GPU for speeding up custom models in R

Krzysztof Jędrzejewski
Advanced Computing & Data Science Lab

WhyR, 4 July 2018

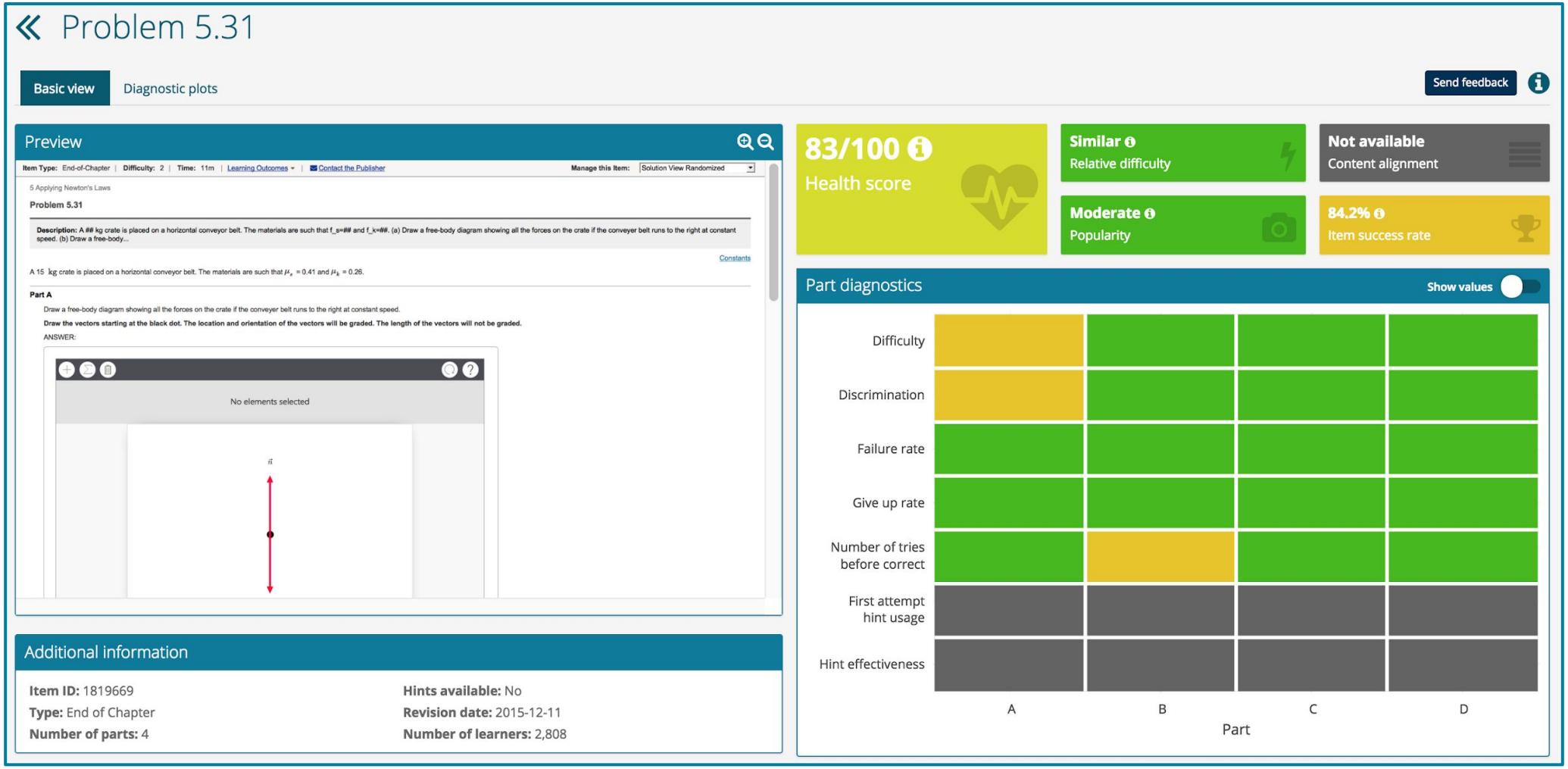
Image by: Tang Yau Hoong

Code samples and slides

https://github.com/kjedrzejewski/WhyR2018

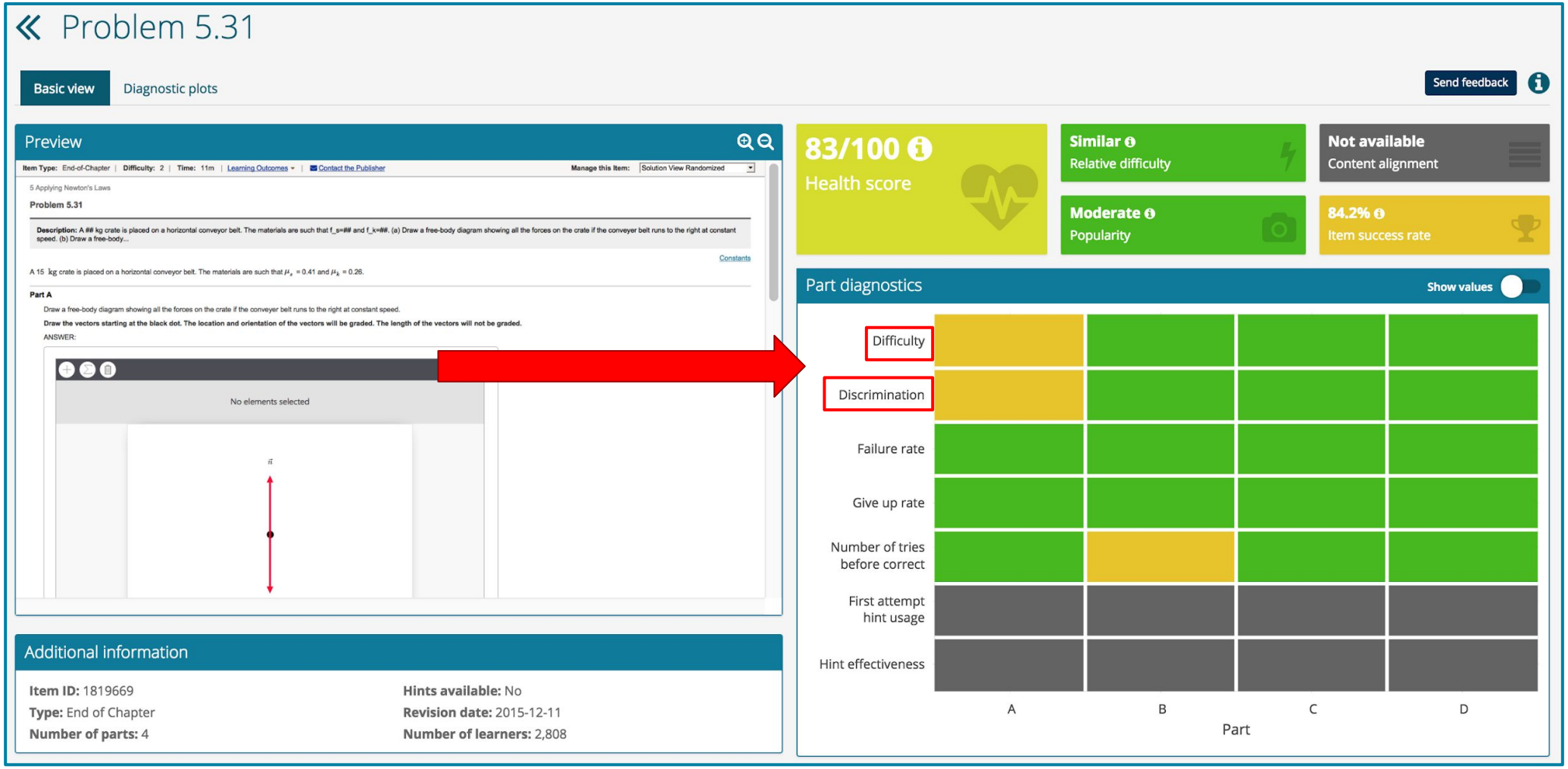
The need

Unified Content Analytics Framework



https://github.com/kjedrzejewski/WhyR2018

Unified Content Analytics Framework



https://github.com/kjedrzejewski/WhyR2018

IRT

- *Item Response Theory*
- Used in psychometrics to estimate the **parameters (e.g. difficulty) of a test question** (and a student's skill level)
- Assumes that the probability that the student would answer a question correctly depends mostly on the **difference between this student skill an the question's difficulty**
- Observed data**:**
  - which **question** was answered?
  - by which **student**?
  - was this answer **correct or incorrect**?
- Can also be used in other areas, e.g. to assess **ad clickability** or to **construct a survey**

https://github.com/kjedrzejewski/WhyR2018

The challenge

The challenge - model

- Our model is a modification of a standard IRT (2PL) model
- it’s no longer in a form supported by IRT libraries

https://github.com/kjedrzejewski/WhyR2018

The challenge - amount of data

- Dozens of e-learning platforms
- Hundreds of titles
- Over 100M observations for some titles
- **Terabytes of data**

https://github.com/kjedrzejewski/WhyR2018

What we tried

Logistic regression with mixed effects

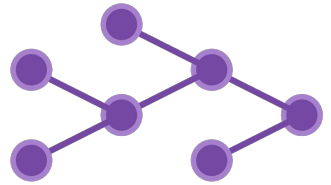
- 1PL model parameters can be estimated as random effects in linear logistic regression for questions and students
- It's fast for small sample sizes, but becomes slow when the amount of data increases
  - For about 2.5 million observations it already takes over 30 minutes

Example code:

github.com/kjedrzejewski/WhyR2018/blob/master/1pl_me.R

Probabilistic programming

+ May be used to build a large variety of models
+ Provides credible intervals of estimated model parameters, which gives us **information about the precision of our estimates**
+ greta can use GPU to speed-up computations, as it's build on top of TensorFlow



- Sampling is **time-consuming**, esp. for big datasets

Example packages: *rstan*, *greta*

Example code:
github.com/kjedrzejewski/WhyR2018/blob/master/2pl_stan.R
github.com/kjedrzejewski/WhyR2018/blob/master/2pl_greta.R

http://mc-stan.org/users/interfaces/rstan https://greta-dev.github.io/greta/

The solution we used

TensorFlow

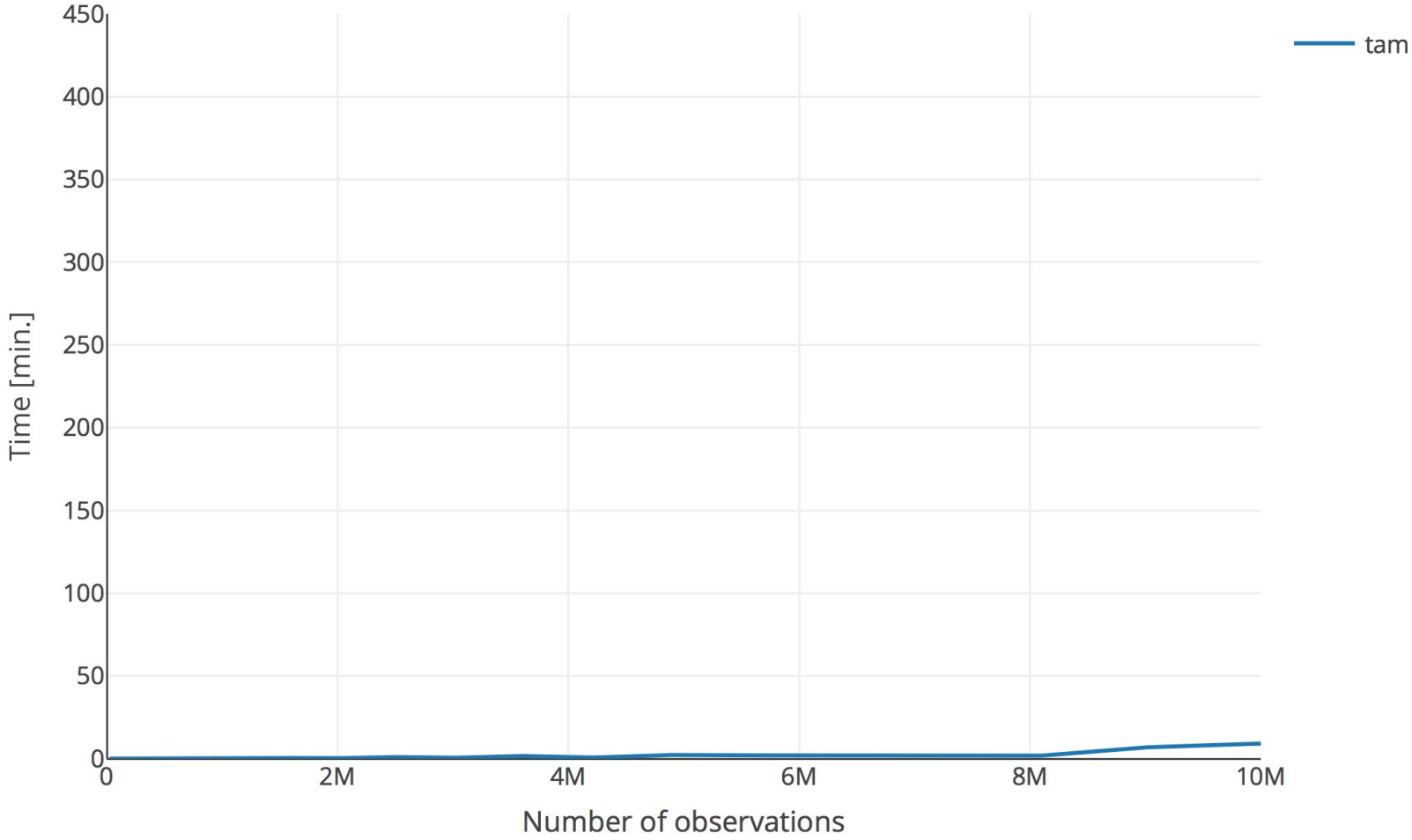
- Dataflow programming library
- Library used commonly for Deep Learning
  - Can be successfully used to build other models too, e.g. **IRT models**, and estimate their parameters based on the Maximum Likelihood
- Provides building blocks that can be used to build a Neural Network
  - tensors, operations, loss functions, optimisers, ...
- User defines the structure of computations using a high-level language, e.g. R or Python
  - C++ backend takes care of the actual computations
- Computations can be significantly **speed up using GPU**

Example code:
github.com/kjedrzejewski/WhyR2018/blob/master/2pl_tf.R

https://tensorflow.rstudio.com

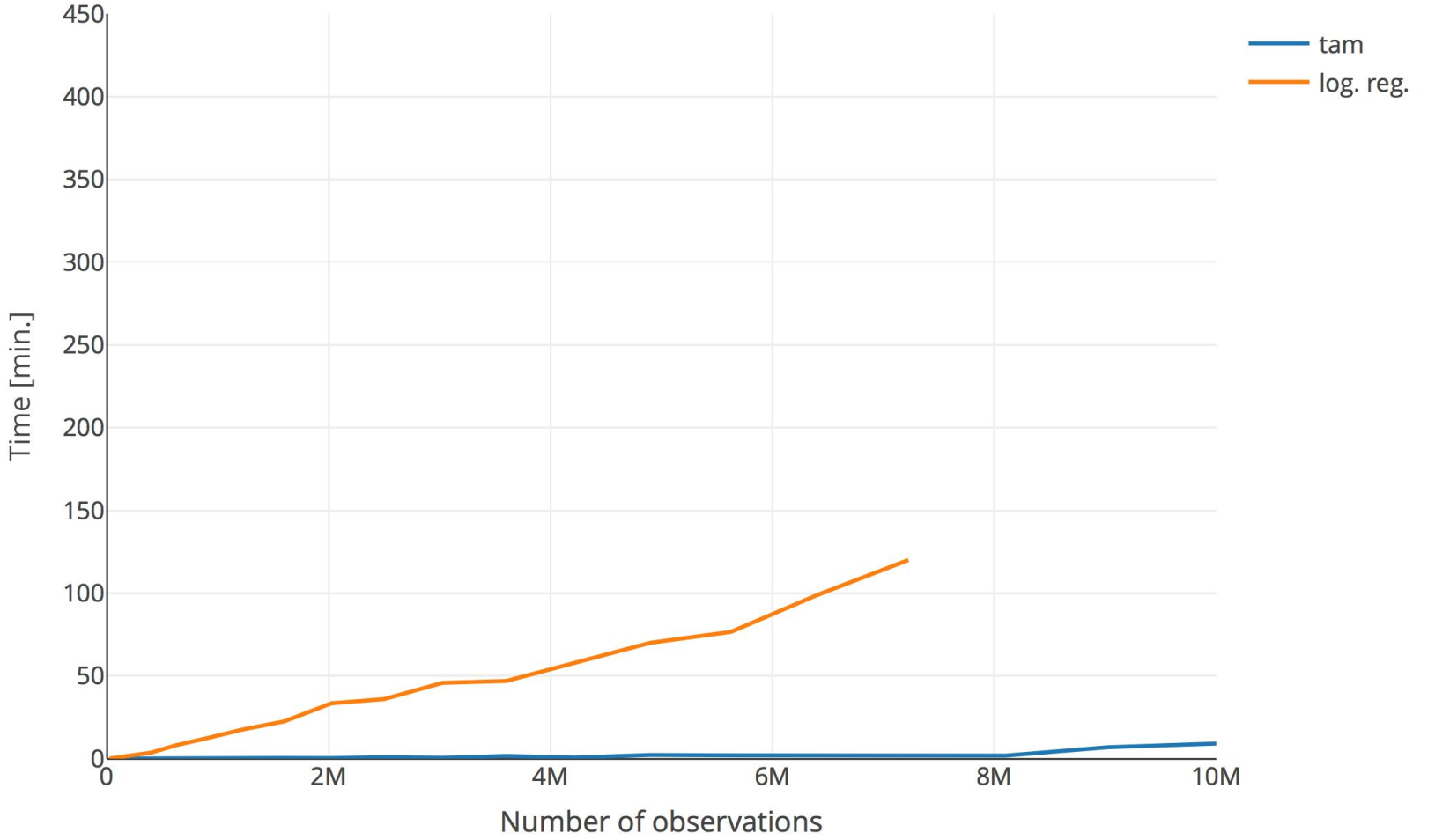
How fast is it?

Benchmark, 2PL, IRT with TAM



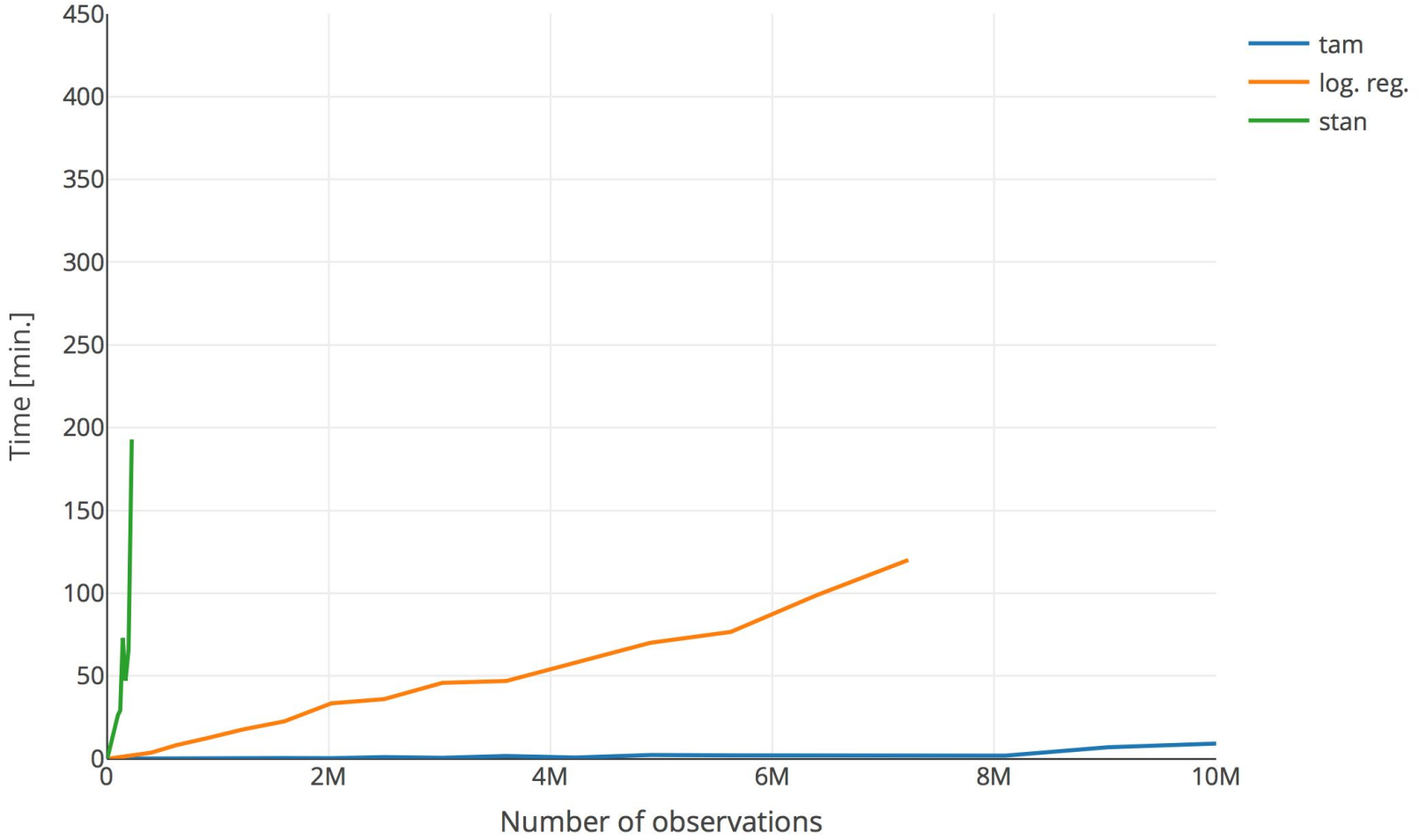
https://github.com/kjedrzejewski/WhyR2018/blob/master/tests_2pl.R

Benchmark, 1PL, logistic regression with lme4



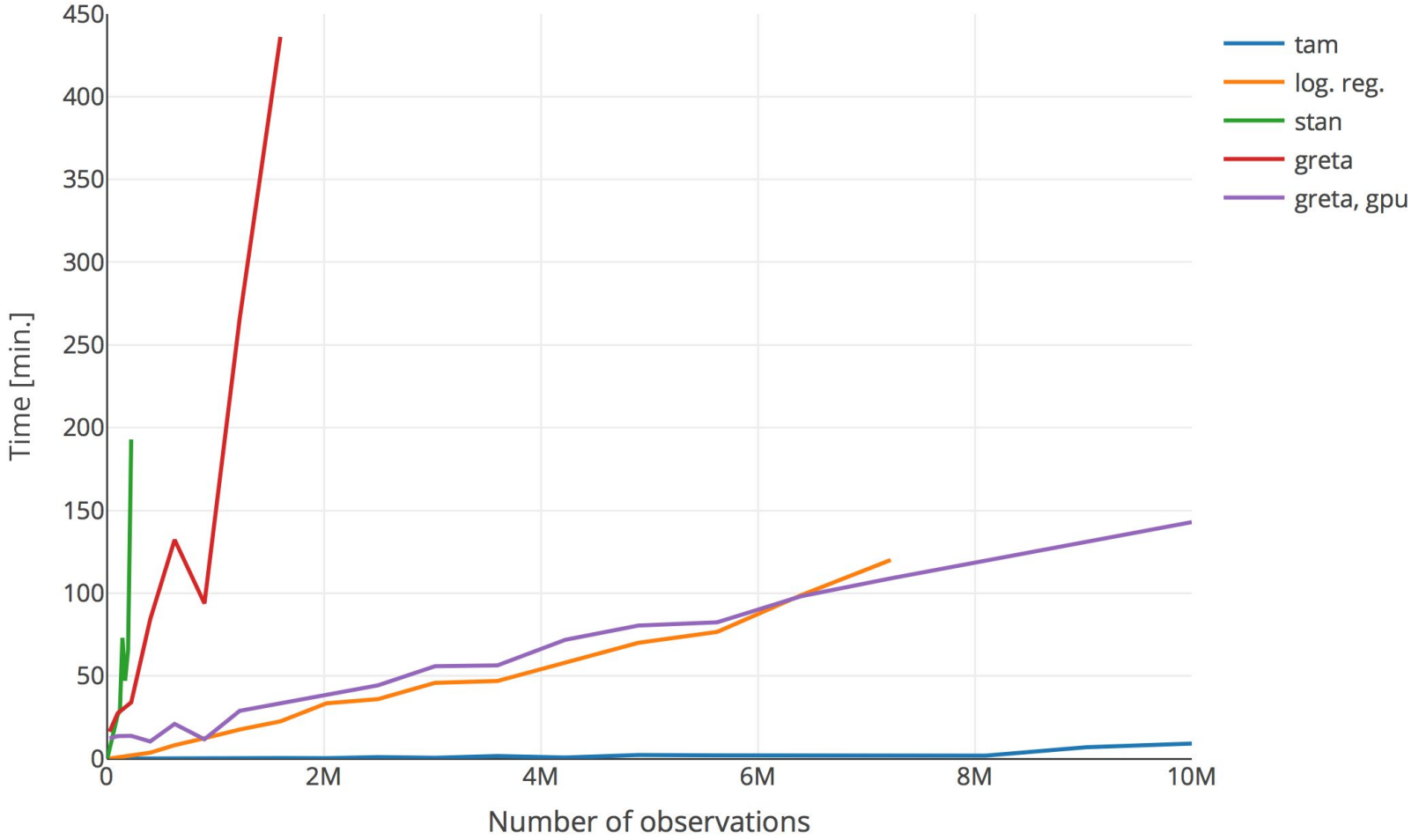
https://github.com/kjedrzejewski/WhyR2018/blob/master/tests_2pl.R

Benchmark, 2PL, stan



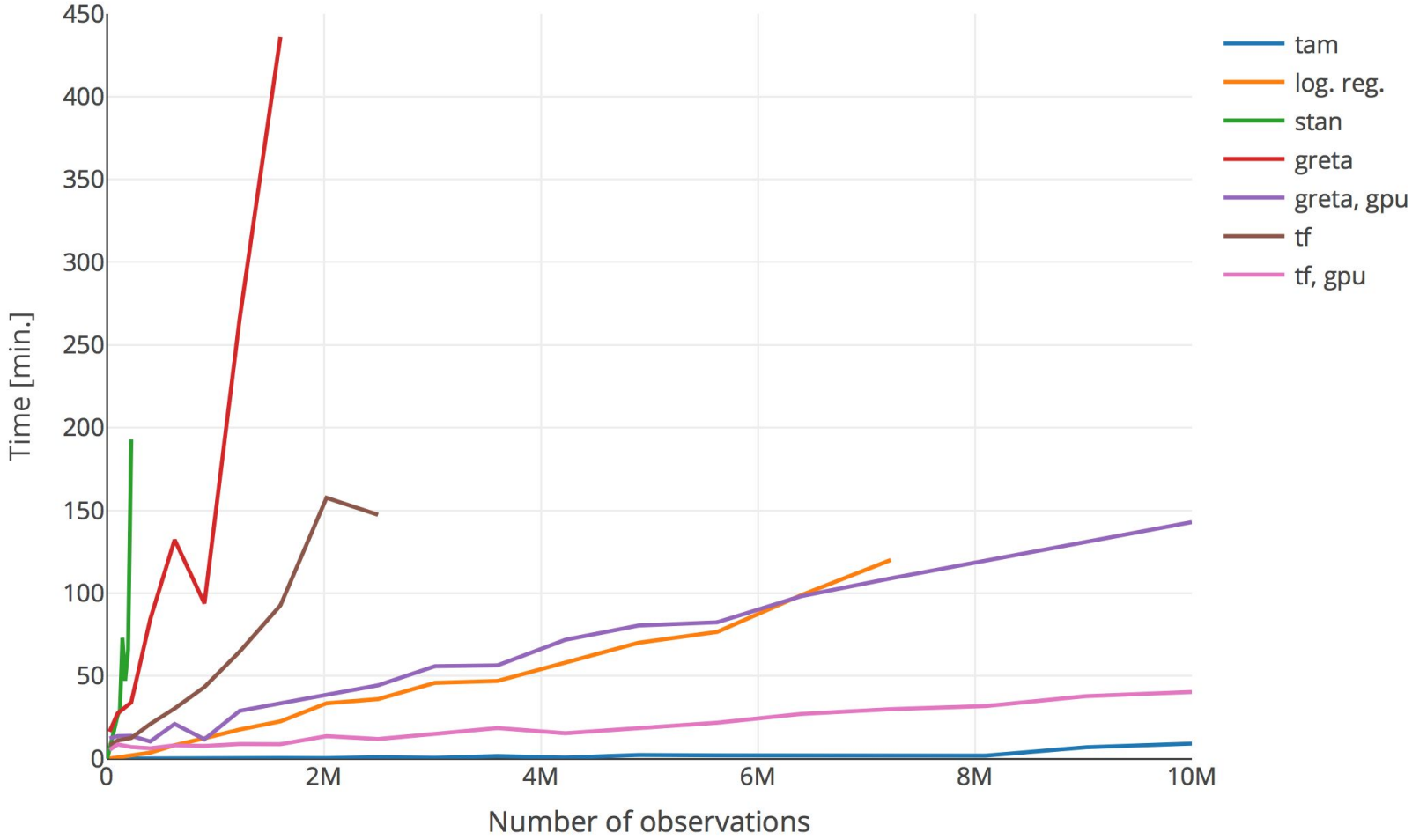
https://github.com/kjedrzejewski/WhyR2018/blob/master/tests_2pl.R

Benchmark, 2PL, greta



https://github.com/kjedrzejewski/WhyR2018/blob/master/tests_2pl.R

Benchmark, 2PL, TensorFlow



https://github.com/kjedrzejewski/WhyR2018/blob/master/tests_2pl.R

Running it for thousands of titles

Amazon EC2

- Virtual Machines hosting service
  - Machines are billed per second
- P3 Instances
  - equipped with NVIDIA Tesla V100, a general purpose GPU

https://aws.amazon.com/ec2/

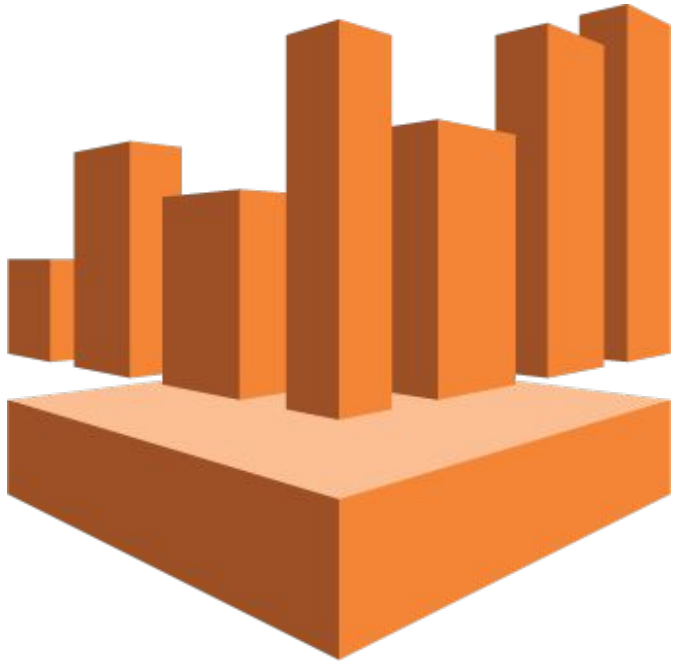
Deep Learning AMI

- A configured environment for using GPU for computations
- We just need to install R and packages we need
- But we can create our own custom AMI afterwards

https://aws.amazon.com/ec2/

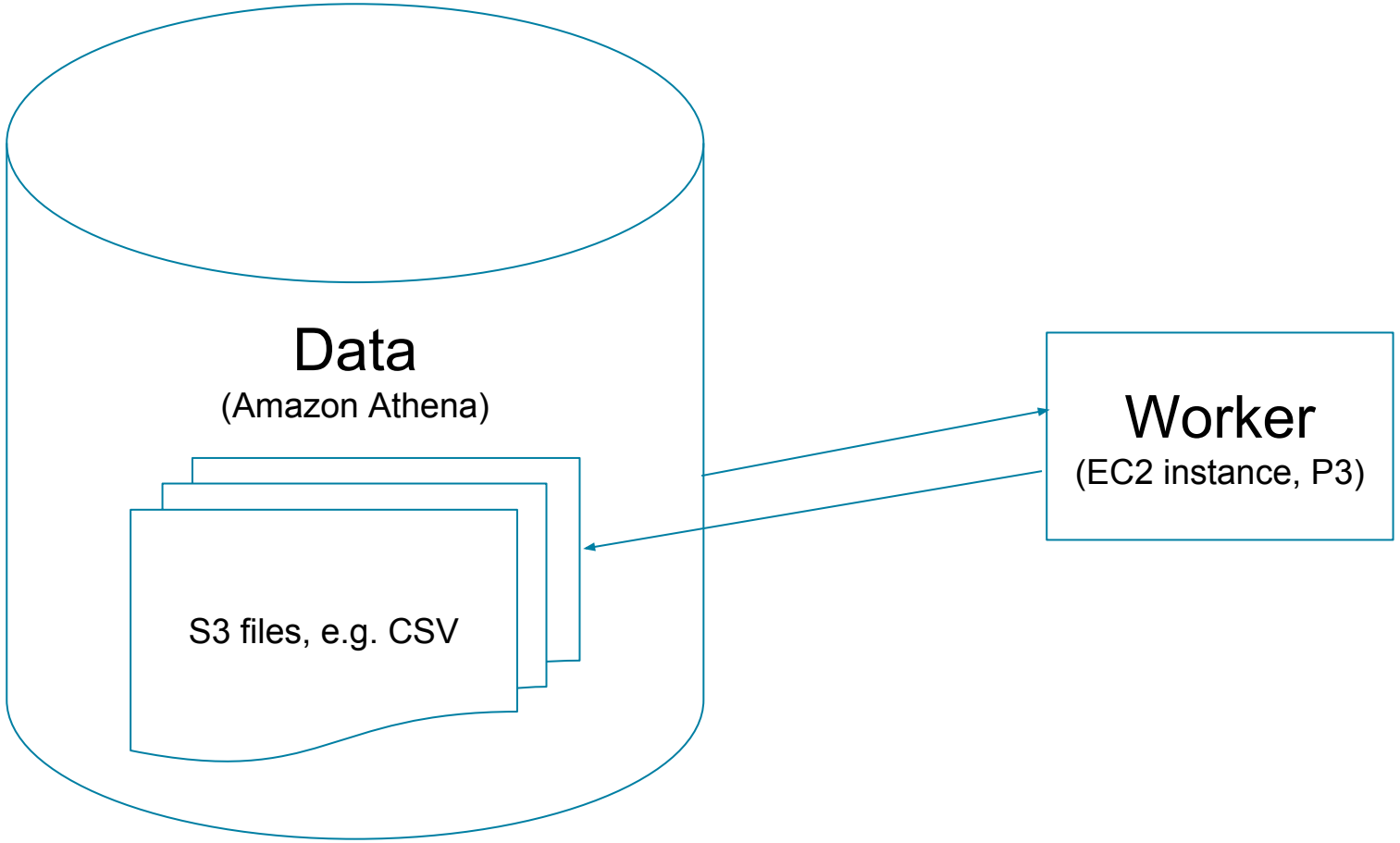
Amazon Athena

- SQL-based service for querying data stored in files on S3
- It’s relatively cheap when compared with a classic SQL-based data warehouses

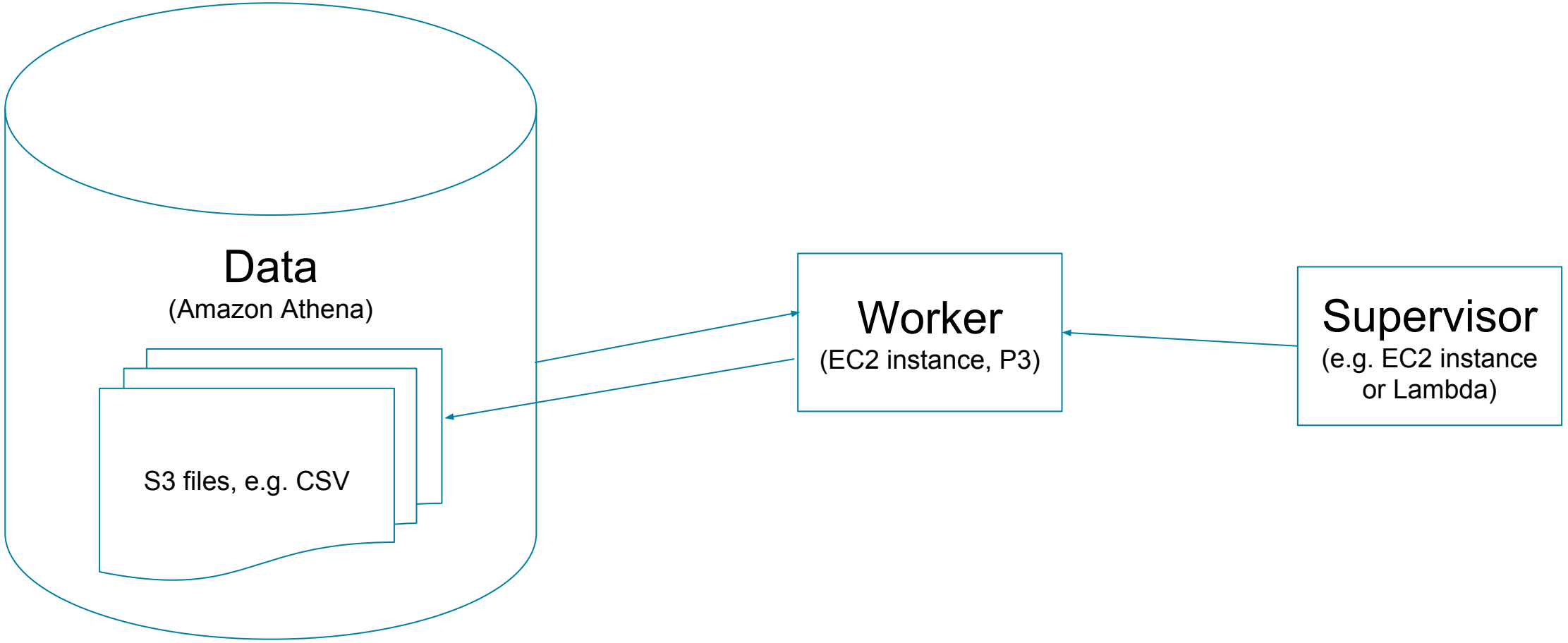


https://aws.amazon.com/athena/

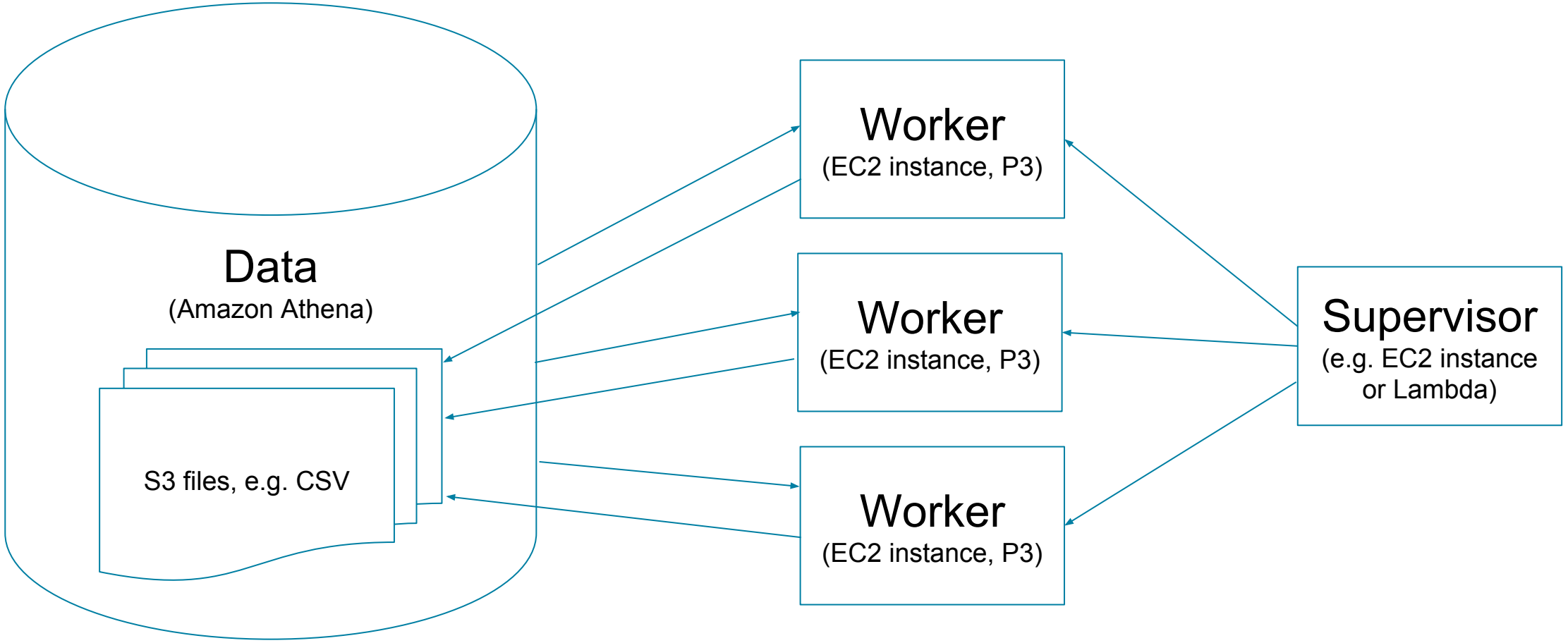
Athena + EC2



Athena + EC2


Data
(Amazon Athena)
S3 files, e.g. CSV
Worker
(EC2 instance, P3)
Supervisor
(e.g. EC2 instance
or Lambda)

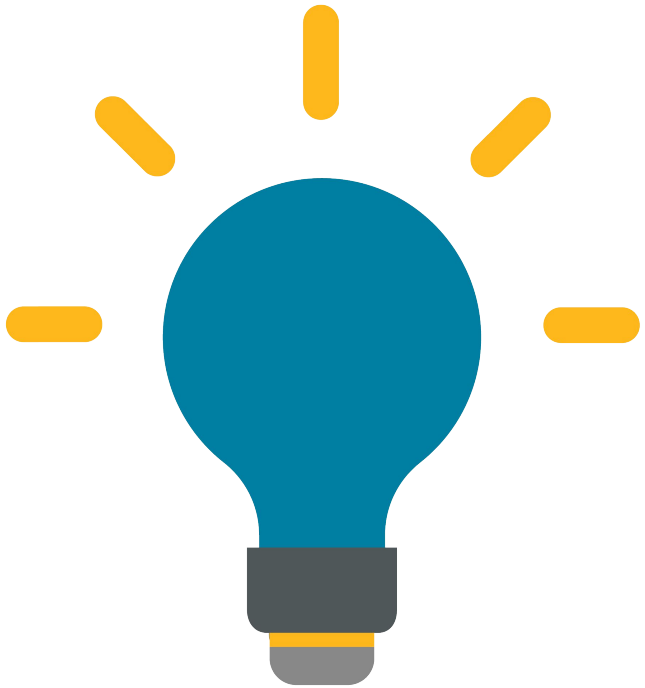
Athena + EC2


Data
(Amazon Athena)
S3 files, e.g. CSV
Worker
(EC2 instance, P3)
Worker
(EC2 instance, P3)
Worker
(EC2 instance, P3)
Supervisor
(e.g. EC2 instance
or Lambda)

Takeaways

Takeaways

- We can use GPUs to speed up calculation of model parameters for large variety of models
- Using cloud services, it's possible to process terabytes of data in a short time
- *TensorFlow* is useful for other tasks than deep learning
- When dealing with a smaller amount of data, *greta* is a good way to go



https://aws.amazon.com/athena/

Useful links

- Code samples and slides: github.com/kjedrzejewski/WhyR2018
- Our team blog: ioki.pl/category/data-science/

ALWAYS LEARNING