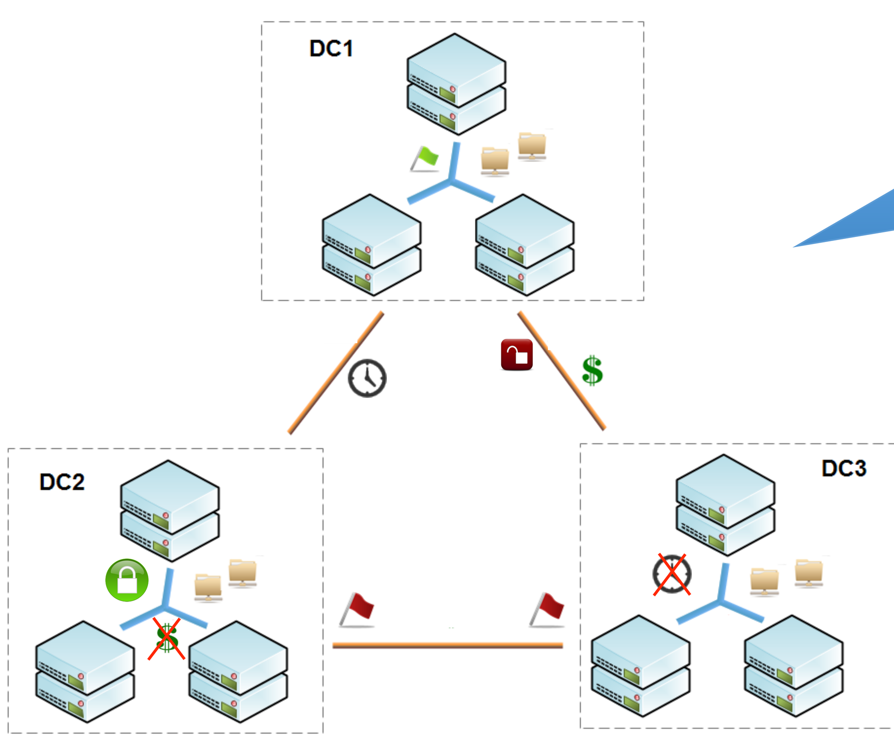
Towards Geo-Distributed Machine Learning

Ignacio Cano, Markus Weimer, Dhruv Mahajan, Carlo Curino, Giovanni Matteo Fumarola
icano@cs.washington.edu, {mweimer,dhrumaha,ccurino,gifuma}@microsoft.com

Motivation

- **Data** is generated and stored **all around the world**.
- **ML** applications require a **global view** of such **data** to achieve the **best results**.

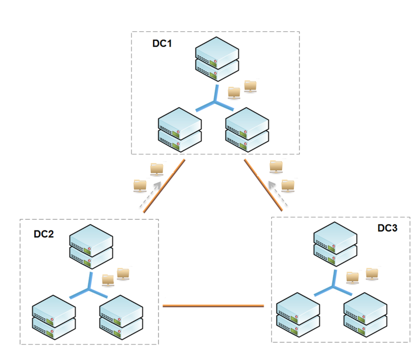
Setup



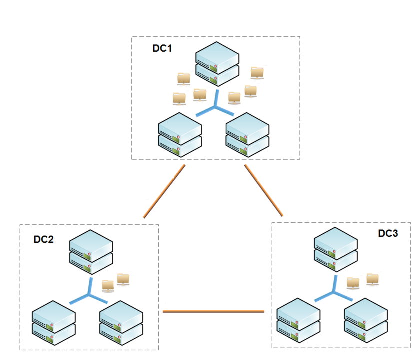
GDML

Current Solutions (Centralized)

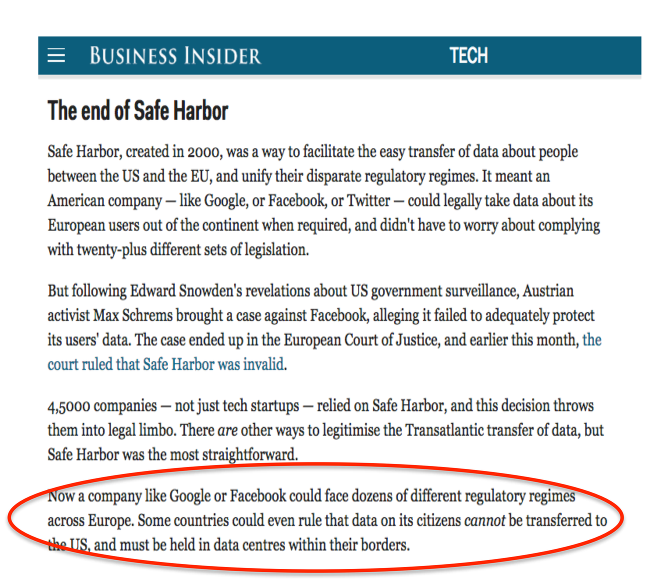
1. **Copy** the **RAW** data into a **single data center**.



2. **Run** the **ML** algorithm "**locally**".



- Intuitive as ML is:
  - Iterative.
  - Communication intensive.
- Problems:
  - X-DC transfers are costly.
  - Data sovereignty issues.
  - Security threats.
  - X-DC high latency.



BUSINESS INSIDER TECH

The end of Safe Harbor

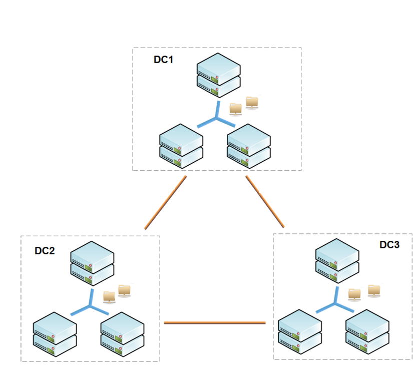
Safe Harbor, created in 2000, was a way to facilitate the easy transfer of data about people between the US and the EU, and unify their disparate regulatory regimes. It meant an American company — like Google, or Facebook, or Twitter — could legally take data about its European users out of the continent when required, and didn't have to worry about complying with twenty-plus different sets of legislation.

But following Edward Snowden's revelations about US government surveillance, Austrian activist Max Schrems brought a case against Facebook, alleging it failed to adequately protect its users' data. The case ended up in the European Court of Justice, and earlier this month, the court ruled that Safe Harbor was invalid.

4,5000 companies — not just tech startups — relied on Safe Harbor, and this decision throws them into legal limbo. There *are* other ways to legitimise the Transatlantic transfer of data, but Safe Harbor was the most straightforward.

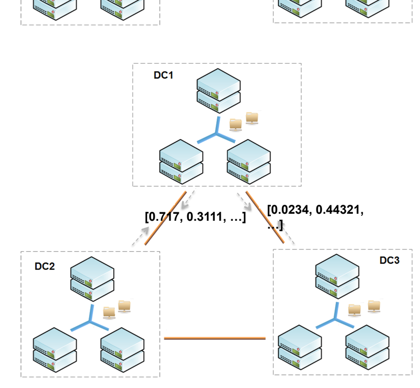
Now a company like Google or Facebook could face dozens of different regulatory regimes across Europe. Some countries could even rule that data on its citizens *cannot* be transferred to the US, and must be held in data centres within their borders.

Our Approach



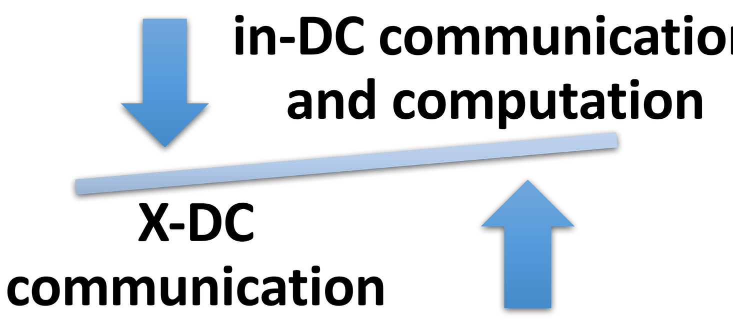
1. **Leave** the **data in place**.

2. **Train** in a **geo-distributed fashion**.



- **Key Challenges**:
  - **Algorithm:** reduces X-DC communication of centralized and achieves same accuracy.
  - **System:** realizes benefits of algorithm, and make it robust to network failure.

1. Algorithm: Mahajan et al., 2015



Objective Function

$$f(w) = \frac{\lambda}{2}||w||^2 + L(w) = \frac{\lambda}{2}||w||^2 + \sum_{p=1}^{P} L_p(w)$$

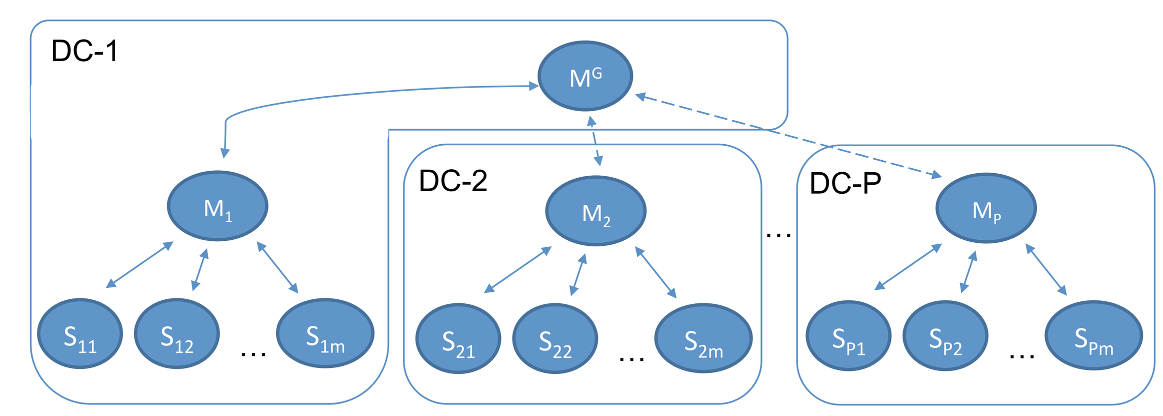
Local approximations

$$\hat{f}_p(w) = \frac{\lambda}{2}||w||^2 + \tilde{L}_p(w) + \hat{L}_p(w)$$

Approximation of $L_p(w)$

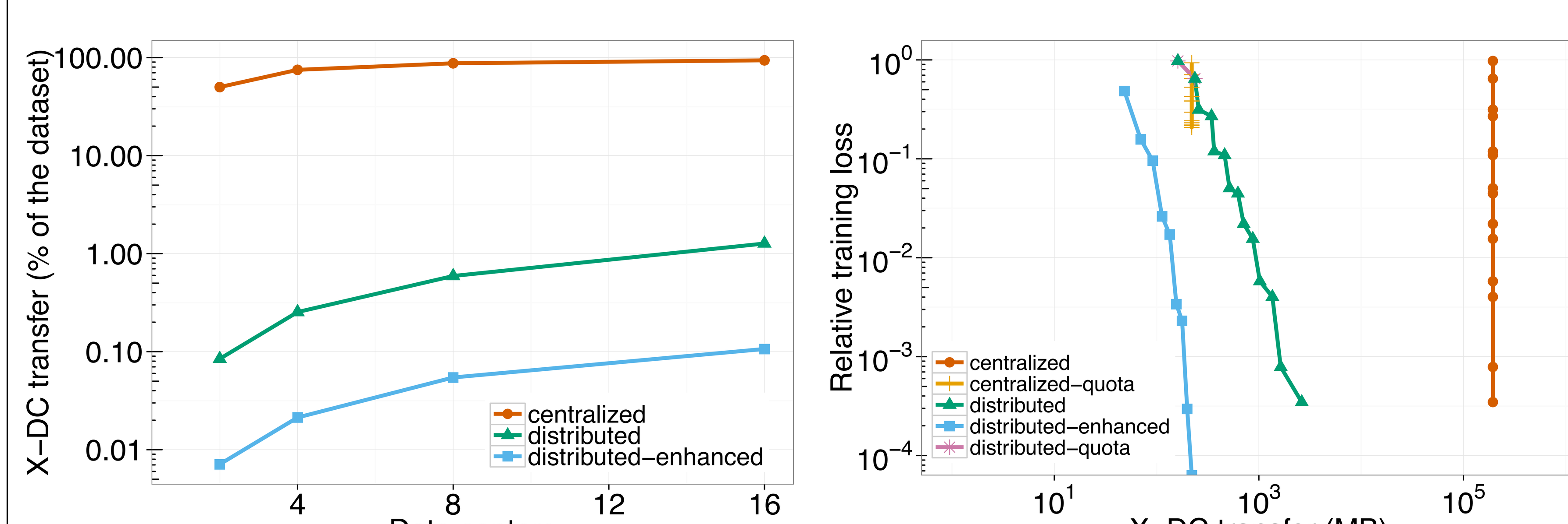
Approximation of the losses of the other DCs $\sum_{q \neq p} L_q(w)$

2. System: Apache **REEF**[1] application on top of a federated **YARN** cluster.



Preliminary Results

- Splice dataset for Human Splice Site Recognition.
  - 50M examples, 47K features, 200GB on disk.
- Simulation of 2, 4, 8 and 16 DCs in large centralized cluster.
- L2 regularized Logistic Regression with TRON.



AUC

Method	2DC	4DC	8DC	16DC
Centralized	0.6660174	0.6660174	0.6660174	0.6660174
Centralized-Quota	0.6652307	0.6642873	0.6557136	0.6417300
Distributed	0.6660174	0.6660174	0.6660174	0.6660174
Distributed-Quota	0.5696233	0.5696233	0.5422752	0.5686233
Distributed-Enhanced	0.6661202	0.6661884	0.6661213	0.6662581

Conclusions & Future Work

- Introduced a **new kind of learning problems** that need to deal with **geo-distributed datasets** (**GDML**).
- Implemented an initial **system** for **X-DC training**.
- Empirical results show **orders of magnitude improvements** in terms of **X-DC transfers** while achieving **same accuracy**.
- Next: Fault-Tolerance, Latency, Privacy, Scheduling…

[1] http://reef.apache.org/