To edge, or not to edge, that’s the question –

*an outsider’s view*

Ion Stoica

UC Berkeley, Director of RISELab

October 3, 2019
Caveats

Really, an outsider when it comes to edge

Intentionally, this is a controversial talk

Cloud outposts (i.e., “edge” managed by cloud providers), not edge in this talk
Why “not to edge”?

Huge heterogeneity:
- Hard to develop
- Hard to test

Deployment nightmare:
- Cannot deploy when you want unless you own devices
- Can take weeks, even months to upgrade!
**Edge**

Huge heterogeneity:
- Hard to develop
- Hard to test

Deployment nightmare:
- Cannot deploy when you want unless you own devices
- Can take weeks, even months to upgrade!

**Cloud**

Homogeneous:
- Same hardware, same infrastructure, same tools
- Test on same infrastructure

Anytime deployment:
- Can deploy daily
- No need to handle multiple versions
Conviva’s story

Phase 1: peer-to-peer video distribution
  • Most functionality at edge
Conviva’s story

Phase 2: Split functionality; multi-CDN delivery

• Backend: select CDN and bitrate
• Peer: metrics computation, downgrade bitrate

![Diagram of data & measurements, CDN selection, and bitrate downgrades.](image-url)
Conviva’s story

Phase 3: dumb edge

- Backend: chose CDN and bitrate; compute metrics
- Edge: collect raw measurements & execute commands
Conviva’s story

Phase 3: dumb edge
• Backend: chose CDN and bitrate; compute metrics
• Edge: collect raw measurements & execute commands

Use JavaScript whenever possible to simplify upgrade

**Tradeoff:** trade performance for simplicity and flexibility

**Side benefit:** can compute new metrics, not available at the collection time as we have raw data!
Another example: Video on Demand

Download & watch (2000-2010):
Limited bitrate couldn’t sustain playing rate
- Complex DRM software

Netflix (2007-): advances in bitrate and network infrastructure allowed streaming
Another example: CDNs

Akamai (2000s): deployed servers at hundreds of sites collocated with ISPs to minimize latency and maximize aggregate bandwidth

- Hard to manage, upgrade

CDN dominant design today: relatively few datacenters peering with many ISPs
Yet another example: Wireless App Protocol (WAP)

WAP (1999): Make it possible for a mobile (bwdth constrained) device to display HTML content

Fully featured HTML mobile clients (2007- )
Why “to edge”?

Performance (latency, bandwidth)
Availability
Security
Latency

If human interaction, we are talking about 100s ms

Increasing Speed of Processing With Action Video Games, Matthew W. G. Dye, C. Shawn Green, Daphné Bavelier, Current directions in psychological science 2009

click. They claim that Formula One driver Lewis Hamilton has a reaction time of an approximate 200 milliseconds, or one fifth of a second. I am comfortable

https://www.thedrive.com/accelerator/8916/is-your-reaction-time-faster-than-lewis-hamiltons
Latency

If wearable cognitive assistance, we are talking about ~33ms (assuming 30 fps)
Latency

If humans involved, we are talking about at least ~33ms
But: 5G + close by datacenter < 10ms RTT

So, even for most interactive tasks, cloud probably ok

If not latency, then what?
Bandwidth

Too much data to send to the backend, e.g., video, sensor measurements \(\rightarrow\) too expensive
But…

Saying it’s too expensive to push data to cloud/cluster…
... is “equivalent” with saying **much of data is not valuable**!

True in some cases (e.g., traffic video monitoring)…
... but not others (e.g., video surveillance)

If not latency and bandwidth, then what?
Availability

For mission critical apps where human life is at stake cannot get disconnected!
But…

… both bandwidth and availability might grow rapidly

Could be good enough for almost all apps
Security

Process personal identifiable information locally → strong privacy guarantees

Lots of resources going into this at Apple, Google, Microsoft, etc!
The challenge

**Train** models preserving user privacy

**Serve** models preserving user privacy
Federated training

Learn without revealing data user’s data

https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
Transfer learning

Train model on lots of public data
Refine it on each edge device
One Challenge

Training largest models: doubles every 3.5 months (*35x* over 18 months)!

Need specialized hardware and algorithms
Promising directions

SqueezeNet\textsuperscript{1}: 100x smaller

New network architectures

Use sketching to reduce communication\textsuperscript{2}

\textsuperscript{1}“SqueezeNet: AlexNet-level Accuracy with 50X Fewer Parameters and < 0.5MB Model Size”, Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer (https://arxiv.org/pdf/1602.07360.pdf)

\textsuperscript{2}“Communication-efficient distributed SGD with Sketching”, Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, Raman Arora, NeurIPS 2019
What about development?

**Automatic optimization** for given platform
- E.g., Auophase\(^1\), NeuroVectorizer\(^2\)

**Program synthesis**: generate programs from high level specifications or input-output examples:
- E.g., Autopandas\(^3\)

---


\(^3\)“AutoPandas: Neural-Backed Generators for Program Synthesis”, Rohan Bavishi, Caroline Lemieux, Roy Fox, Koushik Sen, Ion Stoica, OOPSLA 2019
Summary

The edge is more exciting than ever: key drivers

- Security and availability for mission-critical apps
- Bandwidth cost prohibitive in some situations
- Latency (not sure)

However:

- Keep in mind technology trends (e.g., 5G, satellites)

Put functionality at the edge, *only* if you cannot put it in the cloud.