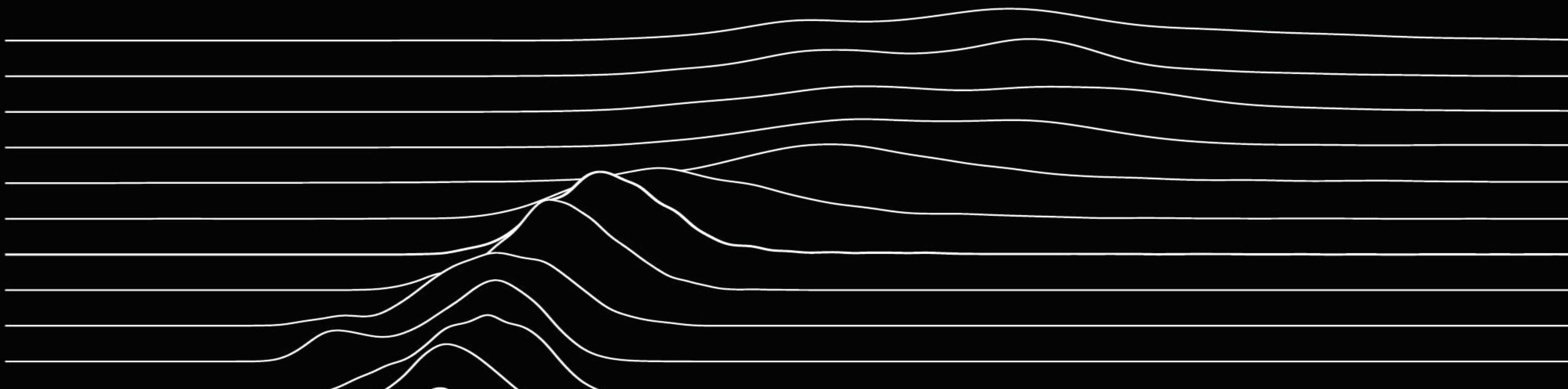
MLAB

10TH ANNIVERSARY
AUGUST 7-8, 2018

M-Lab 2.0

Architecture + demo | Peter Boothe, Stephen Soltesz



M-Lab's new Platform

M-Lab is building a new version of it's platform

More narrowly focused than Planet Lab

Built on control systems and abstractions not invented in 2008

Docker runs the code
Kubernetes (k8s) orchestrates docker across the fleet
Prometheus monitors everything

We had to build some pieces

ePoxy is our secure remote boot system.

index2ip configures the network for each experiment container.

pusher saves all experiment data to Google Cloud Storage.

fast-sidestream provides network instrumentation as a service, replacing web100.

All of these are open source!
http://github.com/m-lab/{epoxy,index2ip,pusher,tcp-info}

Kind of a lot of pieces...

epoxy-images is a system for building kernels, filesystems
prometheus-nagios-exporter to enable transition from Nagios to Prometheus
prometheus-bigquery-exporter to monitor all the way through the parsing step
inotify-exporter because IOPS are the achilles heel of cloud systems
gcp-service-discovery allows Prometheus to monitor more parts of Google Cloud
alertmanager-github-receiver turns Prometheus alerts into GitHub issues
ndt-cloud an NDT server with monitoring that does not depend on web100

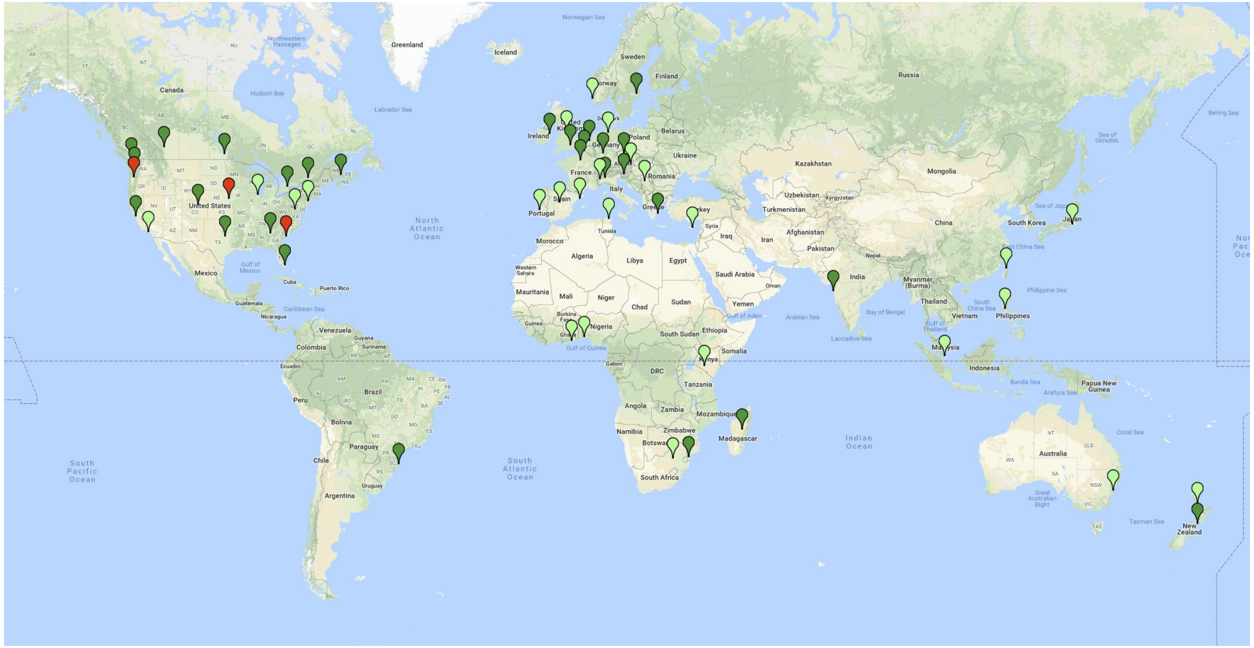
All of these are open source too!
http://github.com/m-lab/${NAME}

DevOps/SRE pieces, too...

prometheus-support scripts for the deployment of monitoring
k8s-support scripts for the deployment of kubernetes masters and nodes
snmp-exporter-support to enable snmp monitoring via Prometheus
script-exporter-support to enable custom health-checks
travis to automate testing and deployment
git-hooks to automatically enforce best-practices in code

All of these are also open source!
http://github.com/m-lab/${NAME}

Across our worldwide fleet



It works today!

It is not (yet) widely deployed

Today, we will perform a live demo of a continent-wide deployment

Hopefully we can also demonstrate some of the capabilities unlocked by this new architecture

LIVE DEMO
HUBRIS ALERT!

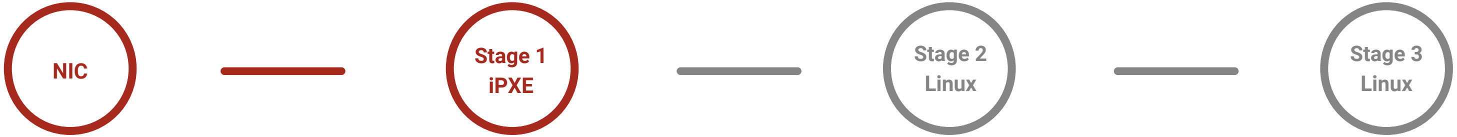
It works today!

It is not (yet) widely deployed

Today, we will perform a live demo of a continent-wide deployment

Hopefully we can also demonstrate some of the capabilities unlocked by this new architecture

ePoxy boots the node...



First boot

The NIC on the node securely contacts the ePoxy server to download a script and tokens for all subsequent stages

iPXE script

The iPXE script securely downloads, using its token, a minimal kernel image, arguments for the kernel, and an initram filesystem

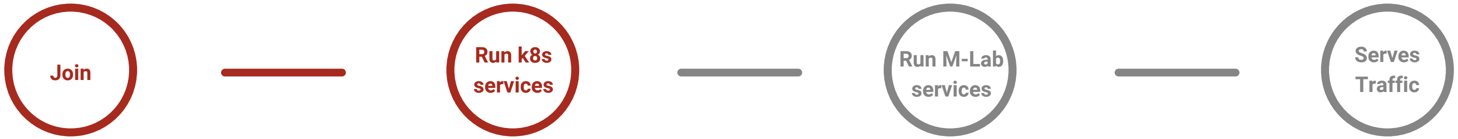
From minimal to full

The minimal linux kernel+args+fs boots up (iPXE is a very limited language) and uses its token to download a complete kernel, complete args, and a full initram filesystem

Boot up, join the cluster

The full kernel boots up with a complete filesystem and complete set of arguments. It then uses its token to download and run a shell script that joins the kubernetes cluster

It becomes a k8s node...



Join

Run k8s services

Run M-Lab services

Serves Traffic

The node joins the cluster

The final ePoxy script causes the node to contact the cluster master and join the cluster

Becomes healthy for k8s

The node starts up the base containers needed to become a healthy kubernetes node

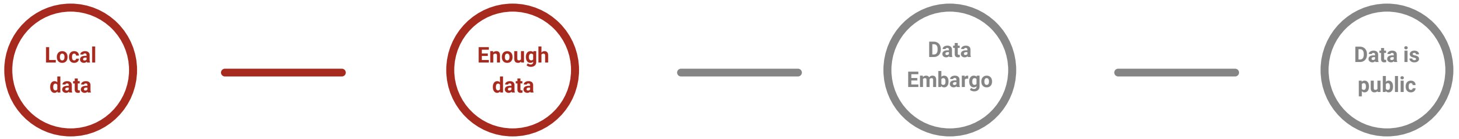
Becomes healthy for M-Lab

Once healthy in the eyes of the kubernetes scheduler, the node is assigned a set of containers to run as part of M-Lab.

Becomes a production node

Once the M-Lab containers are running and M-Lab monitoring reports everything is healthy, the load-balancer begins to assign traffic to the node

It uploads data...



The containers save data

As each experiment serves production traffic, data is written to disk

Data hits the threshold

Once the data is large enough or old enough, a tarfile of the data is uploaded and, after upload, the local copy is deleted.

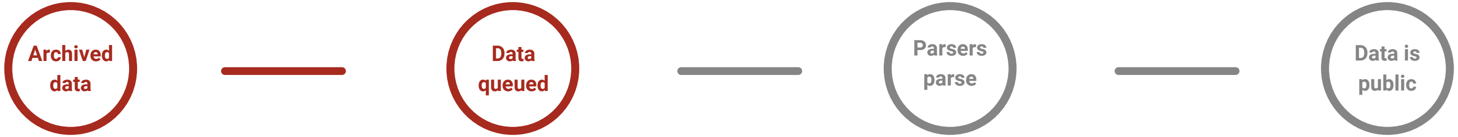
Uploaded data is (possibly) embargoed

Some uploaded data is, for new experiments and for the Measuring Broadband America data, embargoed for up to a year.

Data is made public

After the data passes through embargo (which is of zero time for most experiments), the data is moved to a public bucket and made freely available for download

The data is parsed...



Archived data

Data queued

Parsers parse

Data is public

Cloud functions alert

Newly archived data causes a cloud function to schedule that data for parsing

Data is queued

The data file is queued up for parsing

Each parser grabs its data

A set of parsers divides up the data in the queue, parses it, and streams the results to BigQuery

Data is made public

After the data passes through embargo (which is of zero time for most experiments), the data is moved to a public bucket and made freely available for download