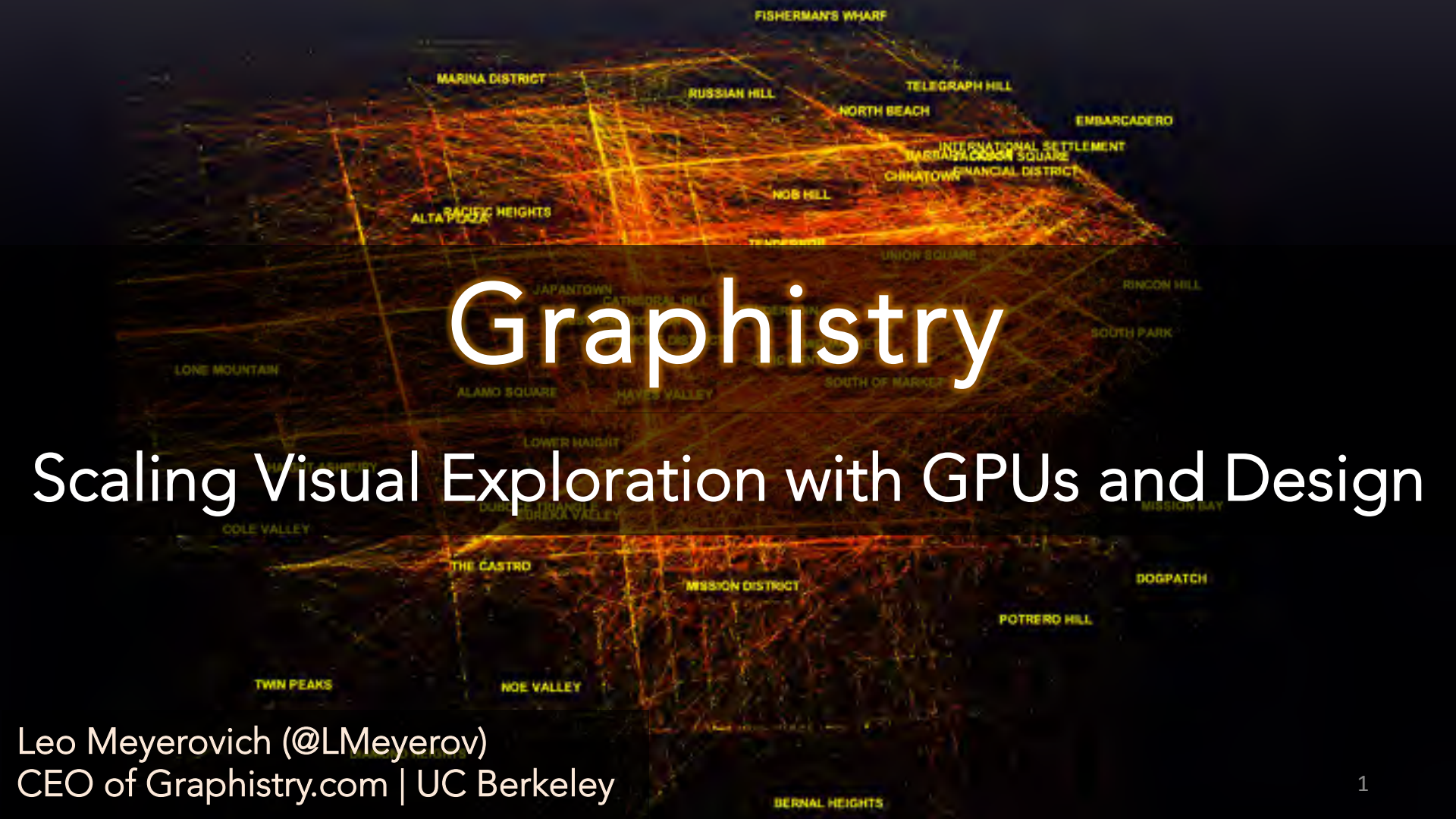
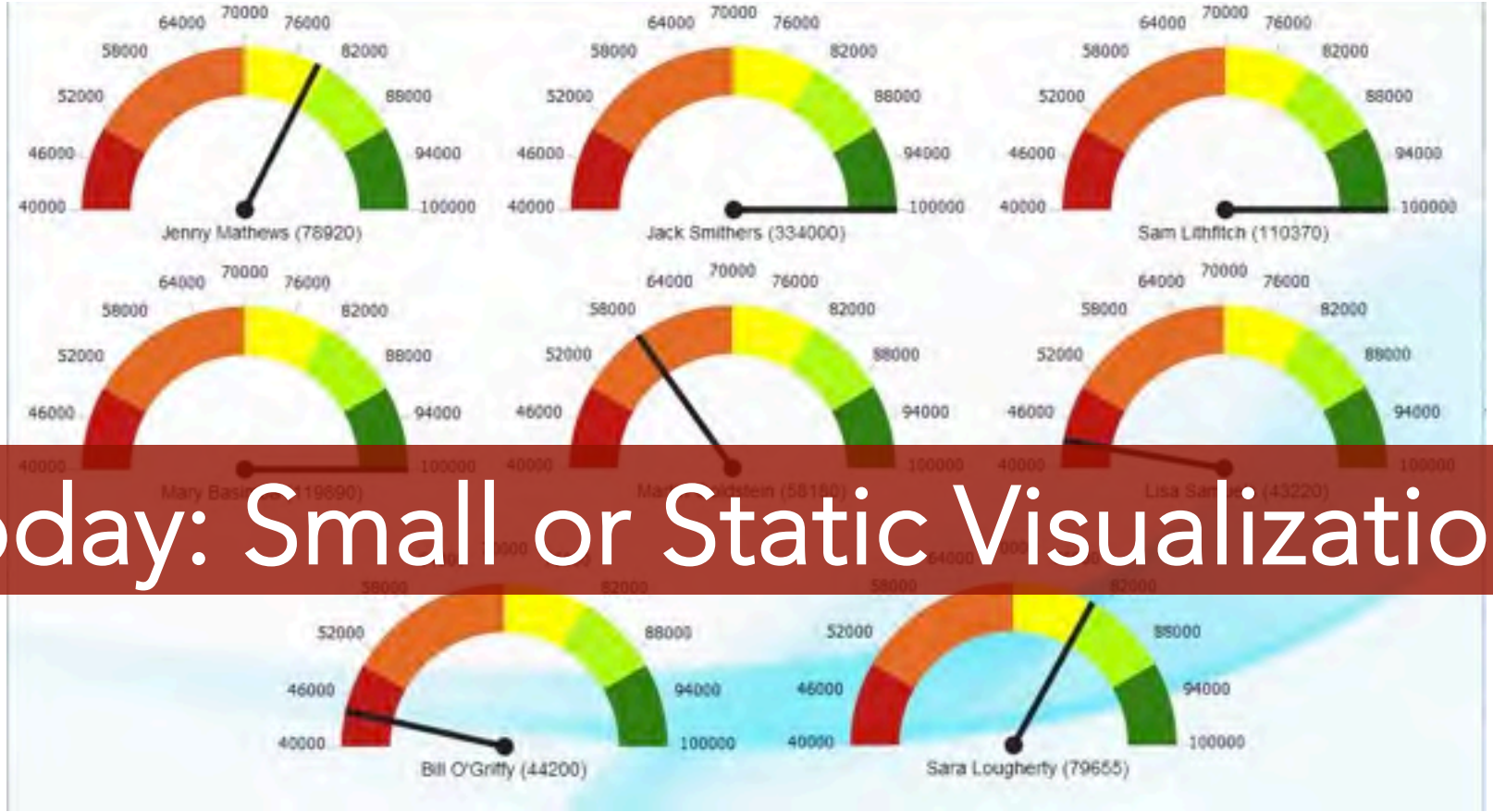


Graphistry

Scaling Visual Exploration with GPUs and Design

Leo Meyerovich (@LMeyerov)
CEO of Graphistry.com | UC Berkeley

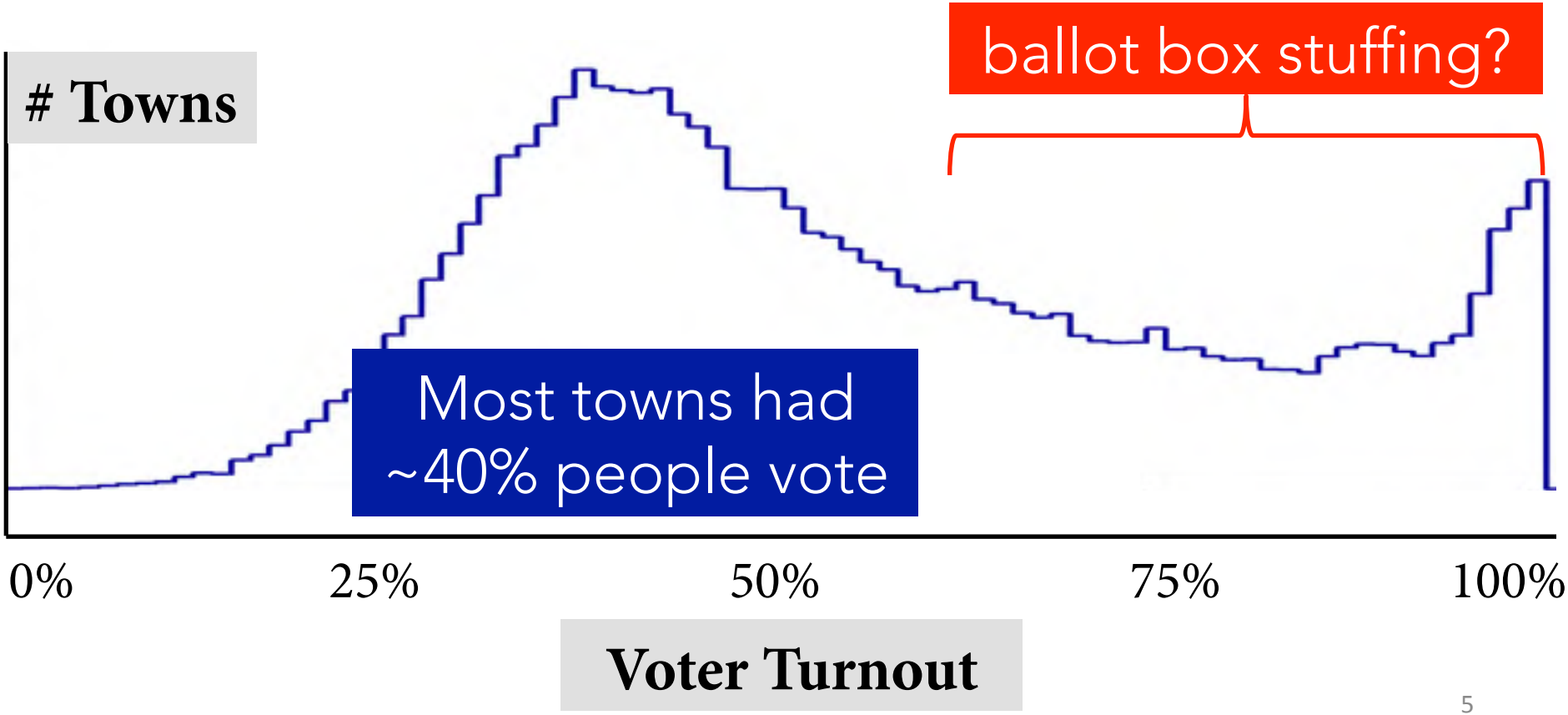

The Future of Data Visualization



Today: Small or Static Visualizations

Ballot Boxes: 100K rows x 30 col CSV

Stack Towns by Voter Turnout



SUPERCONDUCTOR

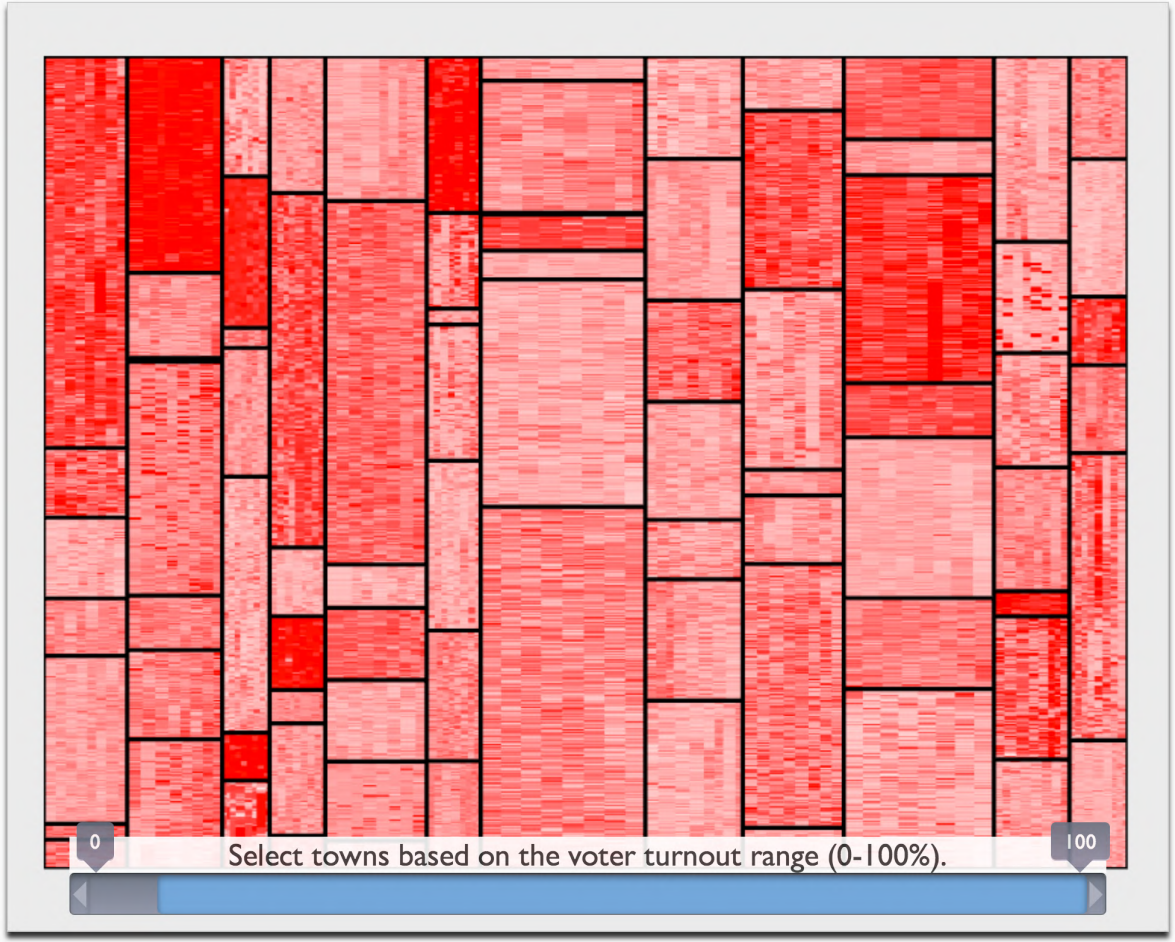
Demo: Voter Precinct by Party & Turnout

Data: Reported Results

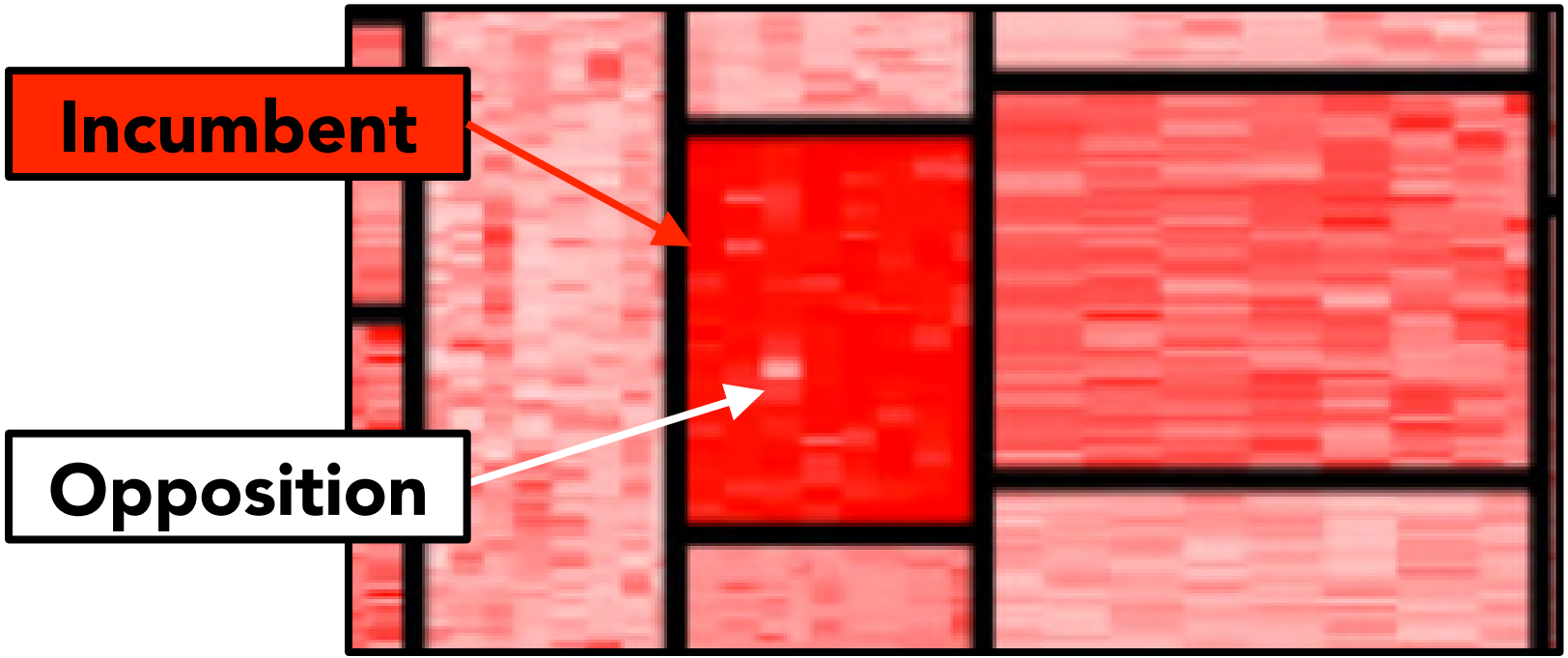
Ruling Party: 50%

Legend

- Small square: voting precinct
- Square size: votes cast
- Red: town voted for the incumbent
- Black borders: different districts



Tiny square shows town size (area) and vote (color)



SUPERCONDUCTOR

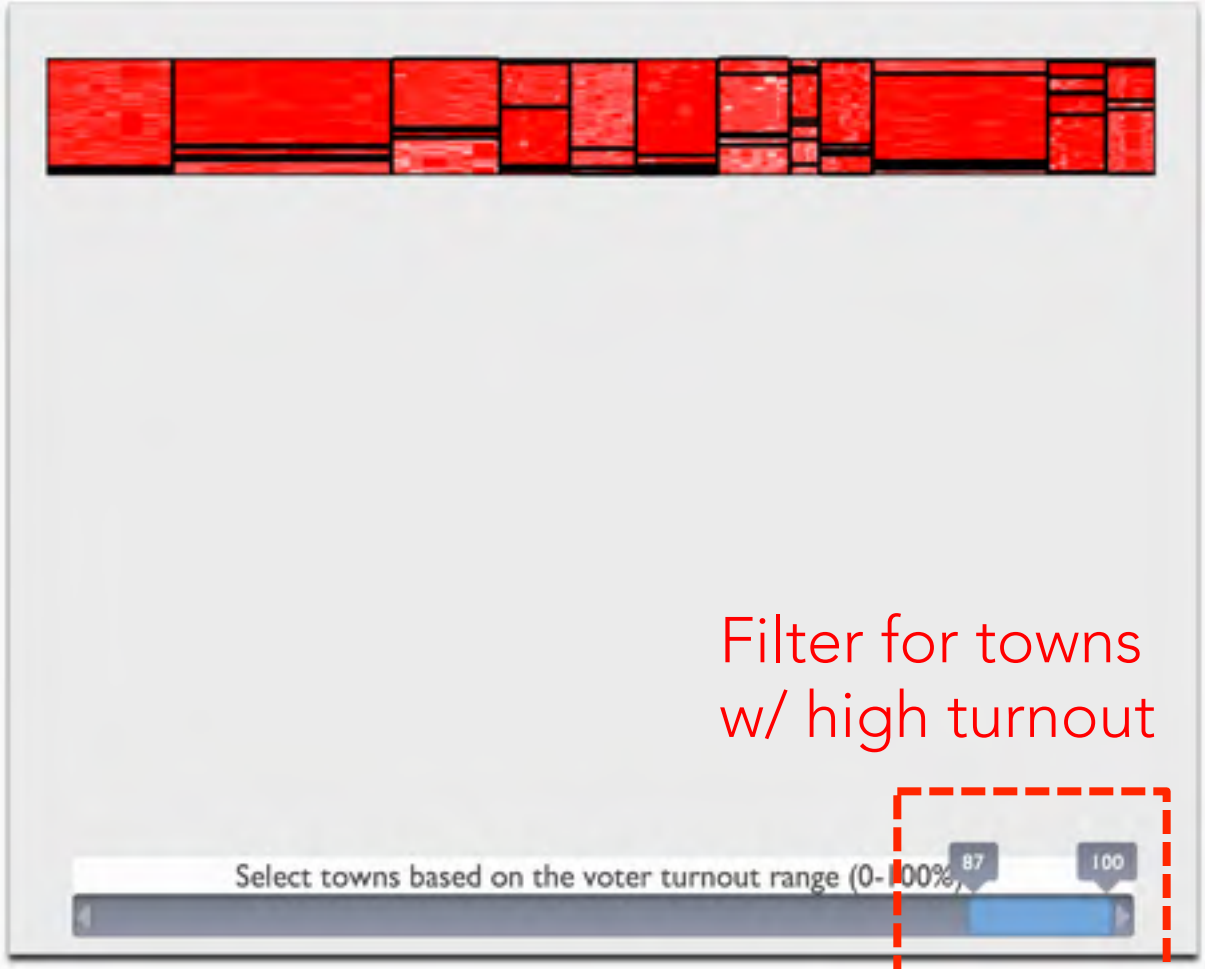
Demo: Voter Precinct by Party & Turnout

Data: Reported Results

Ruling Party: 81%

Legend

- Small square: voting precinct
- Square size: votes cast
- Red: town voted for the incumbent
- Black borders: different districts



Filter for towns w/ high turnout

SUPERCONDUCTOR

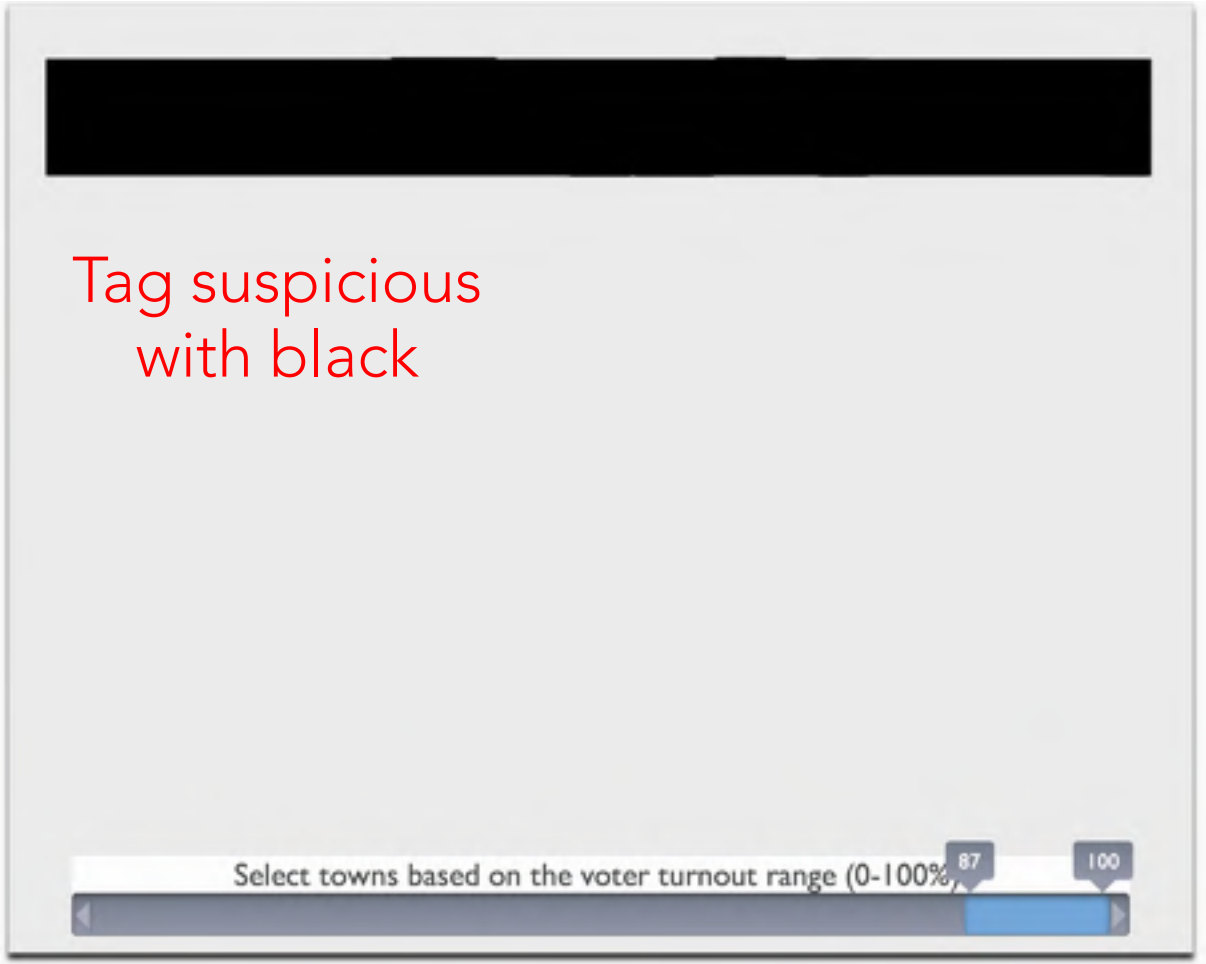
Demo: Voter Precinct by Party & Turnout

Data: Suspect Results Highlighted

Ruling Party: 81%

Legend

- Small square: voting precinct
- Square size: votes cast
- Red: town voted for the incumbent
- Black borders: different districts



SUPERCONDUCTOR

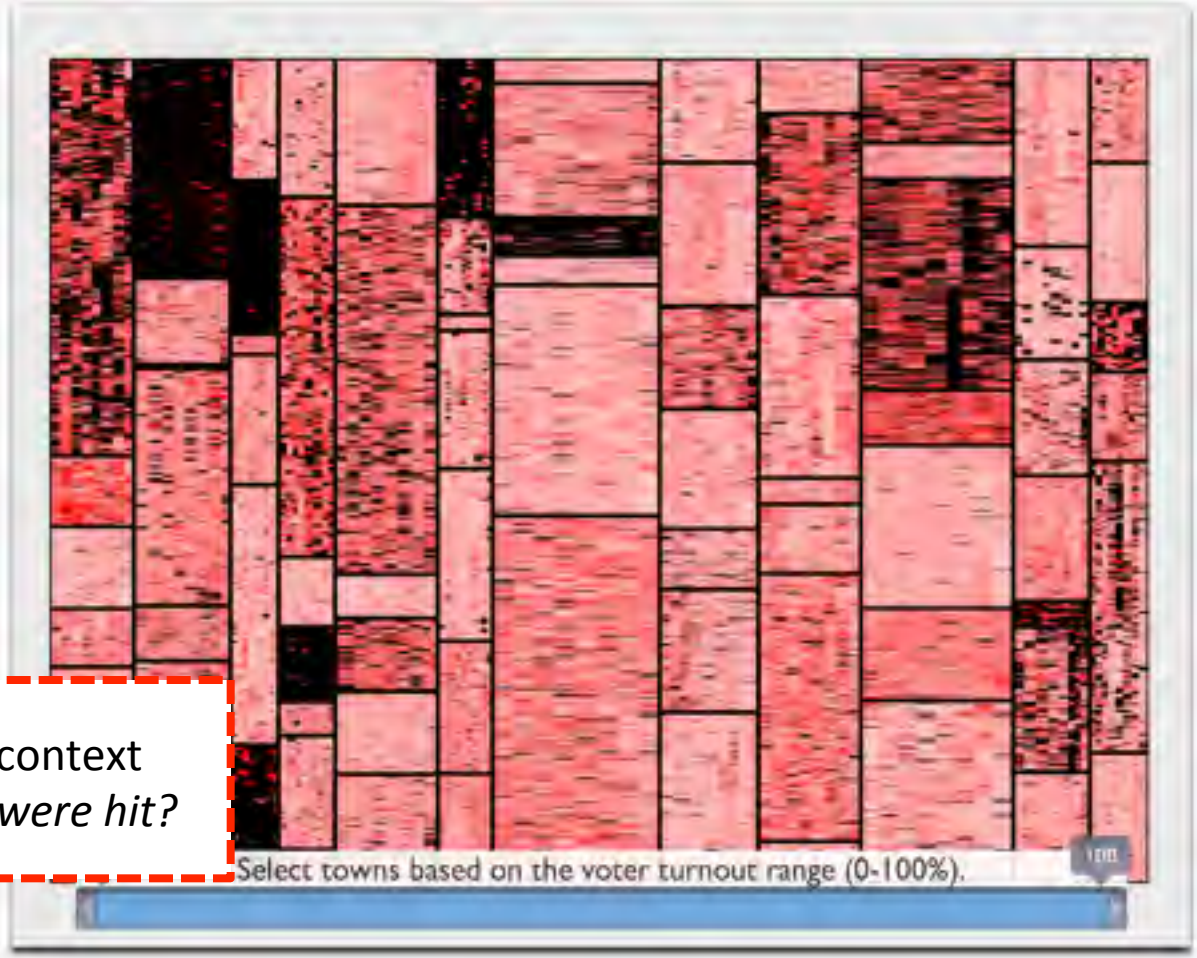
Demo: Voter Precinct by Party & Turnout

Data: Suspect Results Highlighted

Ruling Party: 50%

Legend

- Small square: voting precinct
- Square size: votes cast
- Red: town voted for the incumbent
- Black borders: different districts



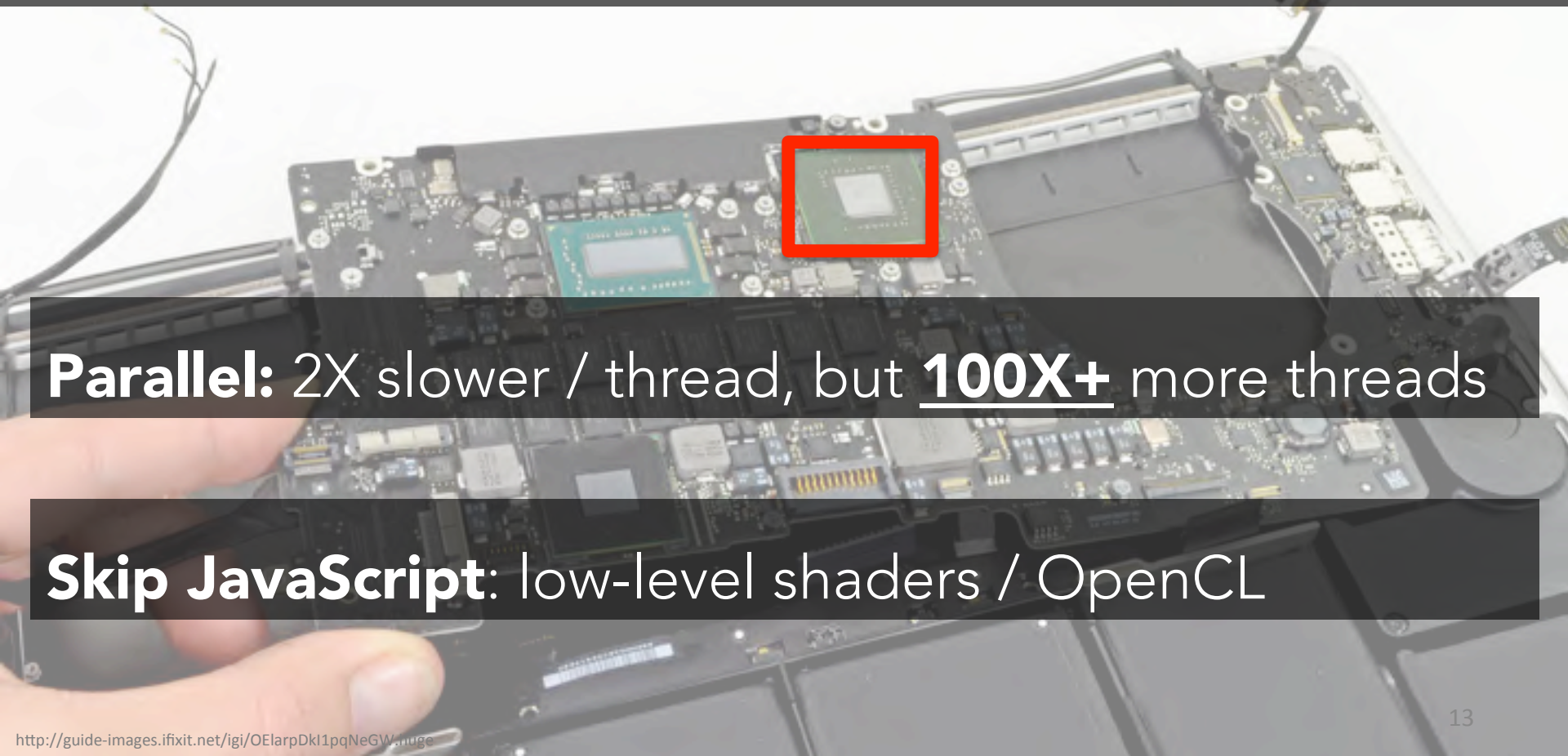
Analyze suspicious activity in context
What parts of the supply chain were hit?

A slider is worth a hundred queries.

H/s Hypotheses per second

100X more visibility
over space and time?

GPUs are Chainsaws



Parallel: 2X slower / thread, but **100X+** more threads

Skip JavaScript: low-level shaders / OpenCL

http://guide-images.ifixit.net/igi/OElarpDkI1pqNeGW.huge

amazon web services

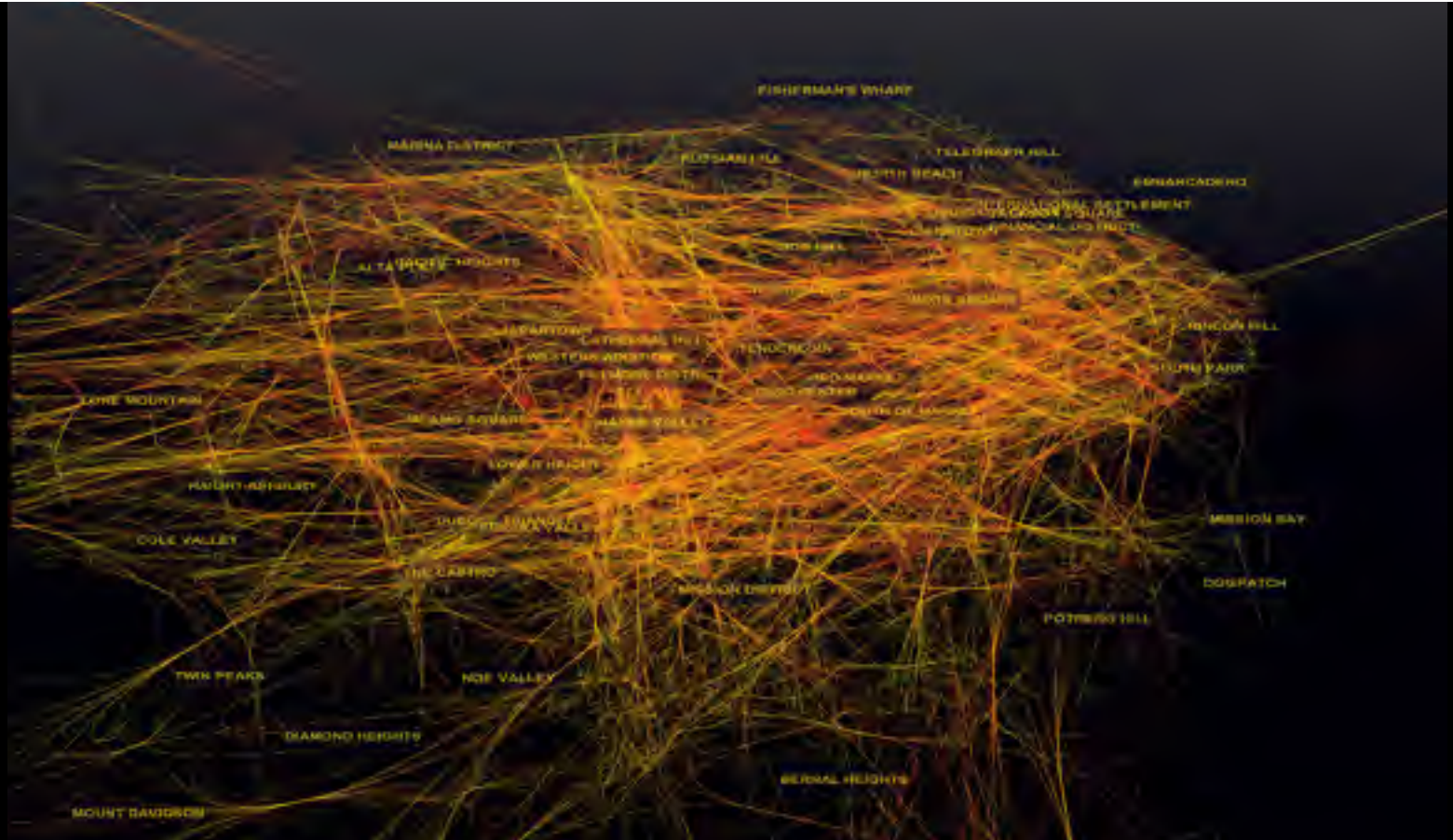
Announcing New Amazon EC2 GPU Instance Type

Posted On: Nov 4, 2013

Uber: Trip Start to End

Direct Edge Placement: Overplotting

Speed → Design → Edge Bundling



Distributed Browser w/ GPU Cloud Streaming

Cloud

WebCL Layout
filter, physics, …

GPU
1,536 cores
4GB memory

Browser

IE, Chrome, iOS, …

WebGL Renderer
paint pixels

GPU
384 cores
1GB memory

Distributed Browser w/ GPU Cloud Streaming

Cloud

15 fps

WebCL Layout
filter, physics, …

GPU
1,536 cores
4GB memory

shapes

commands

The internet is now the interconnect!

Browser

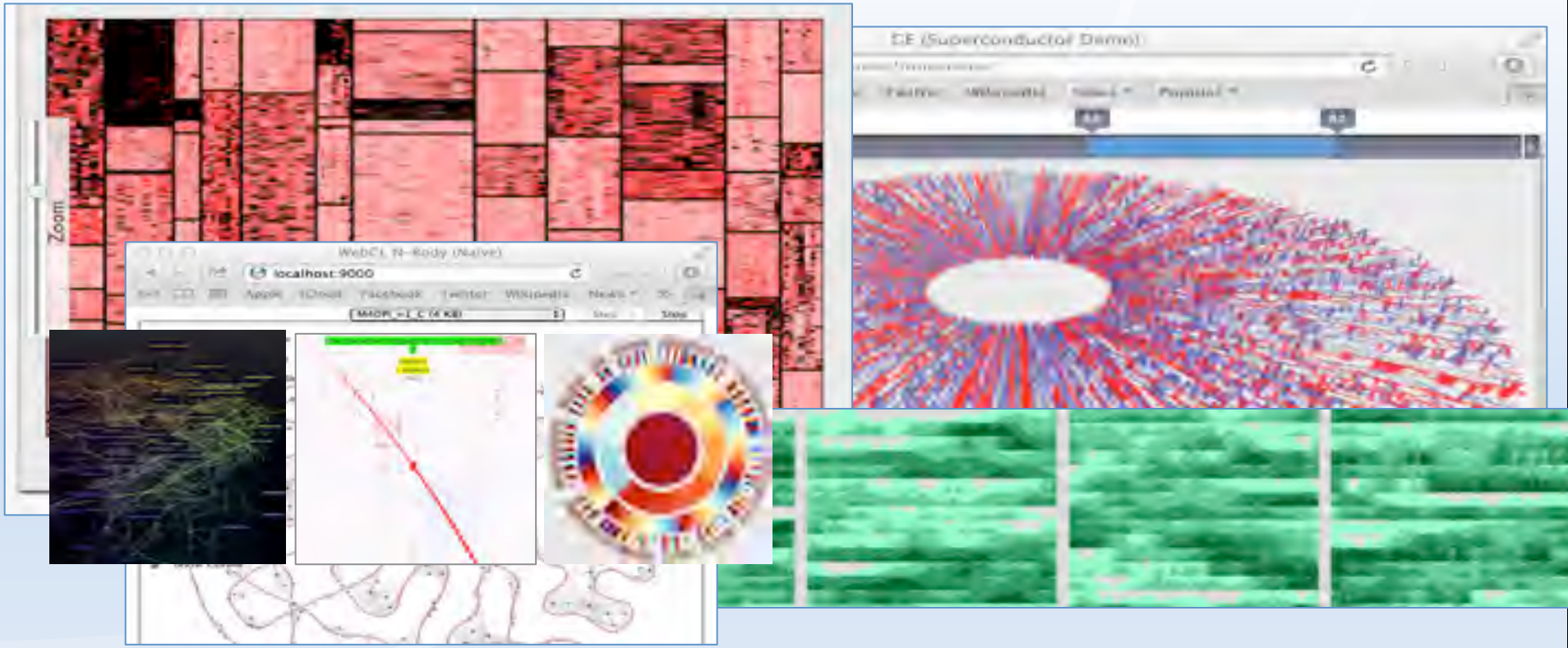
IE, Chrome, iOS, …

60 fps

WebGL Renderer
paint pixels

GPU
384 cores
1GB memory

Toolchain for Creating & Customizing Visualizations



Superconductor language, graph layout engine, OpenGL streaming runtime, …


The Future of Data Visualization
Increase visibility over space and time
by 100X using GPU cloud streaming

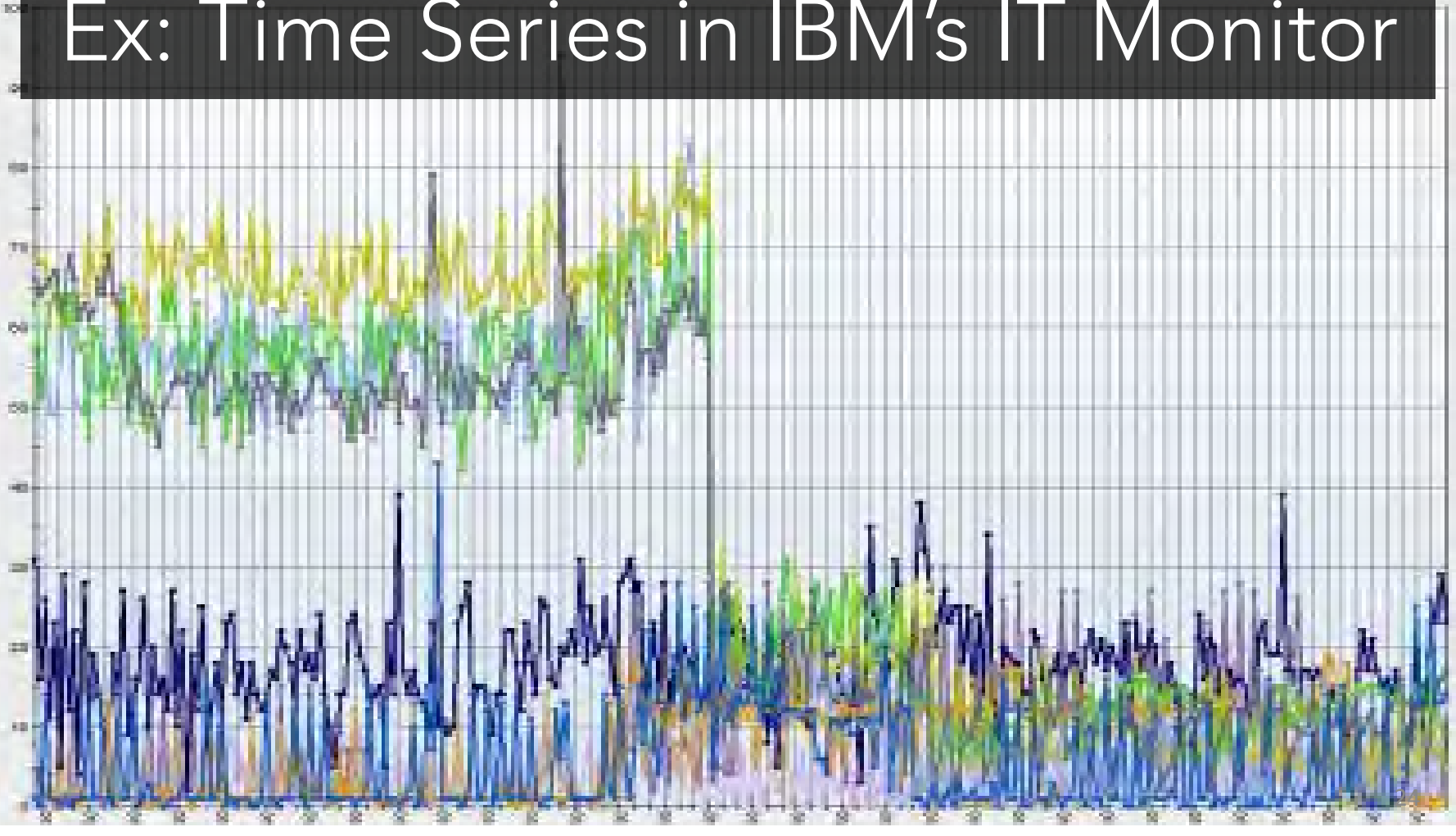
Graphistry

We're hiring: web infoviz, data/ops eng
(& come chat about embedding!)

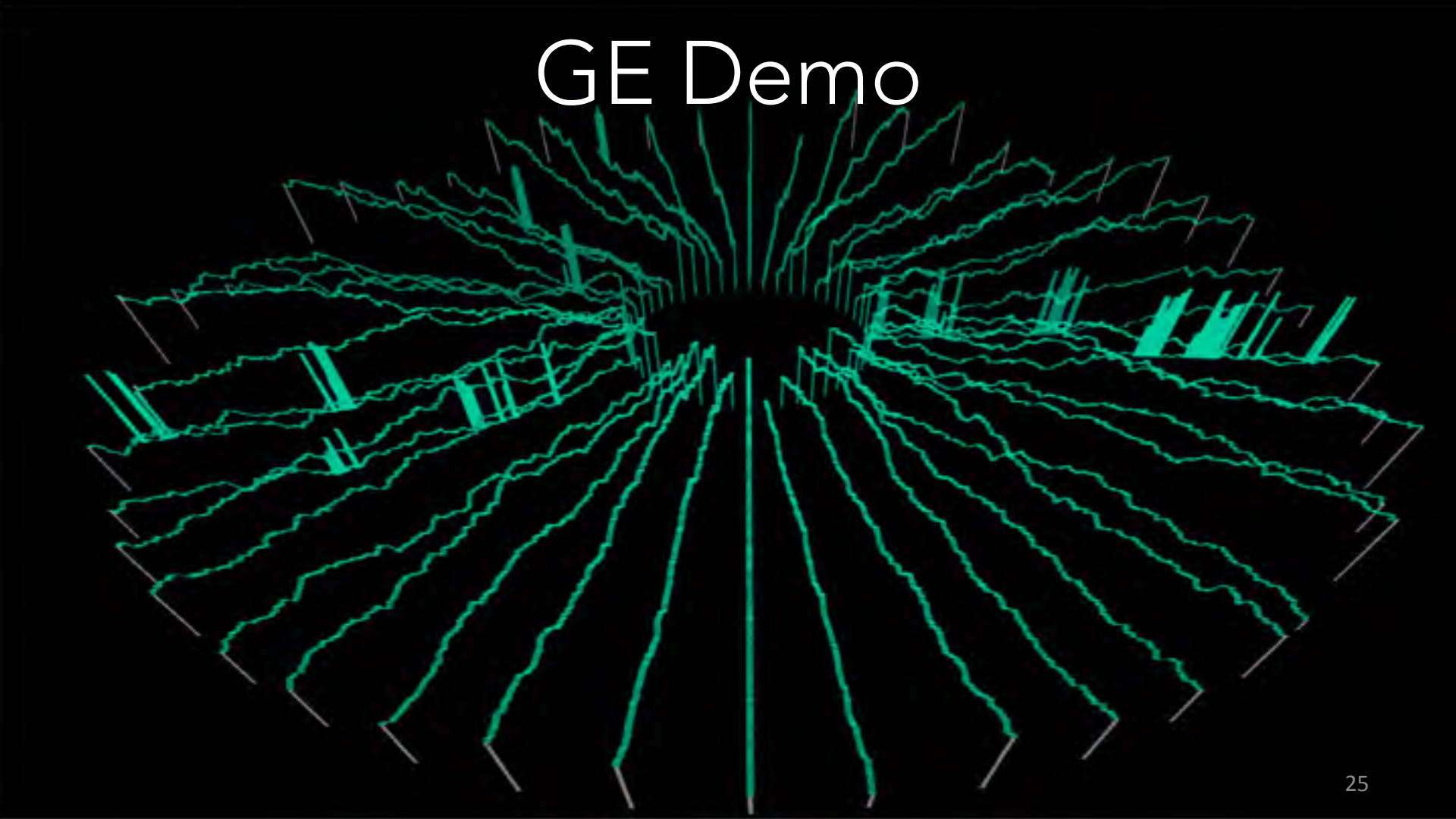
Leo Meyerovich (@LMeyerov)
CEO of Graphistry.com | UC Berkeley

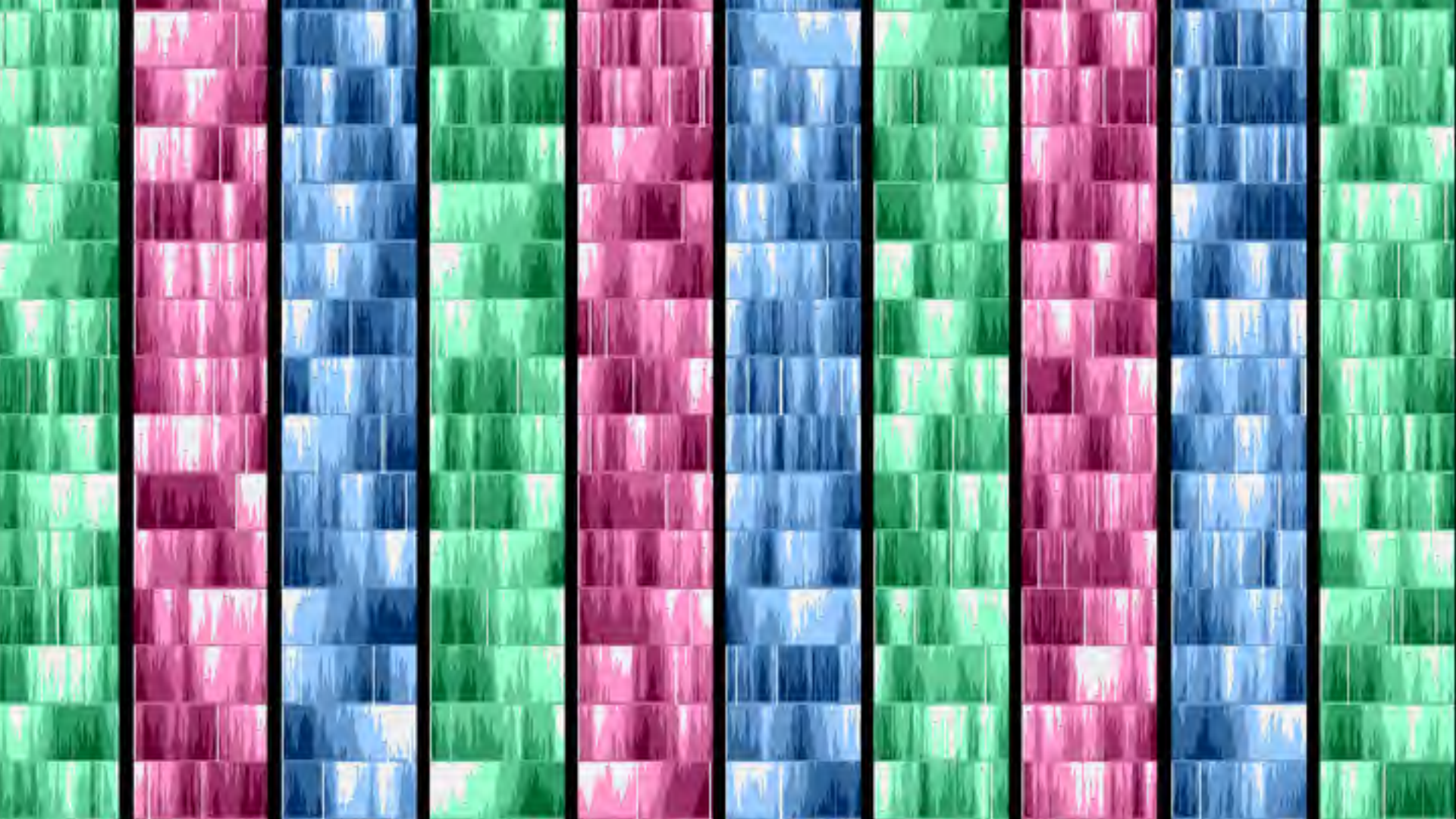
BACKUP

Ex: Time Series in IBM's IT Monitor



GE Demo

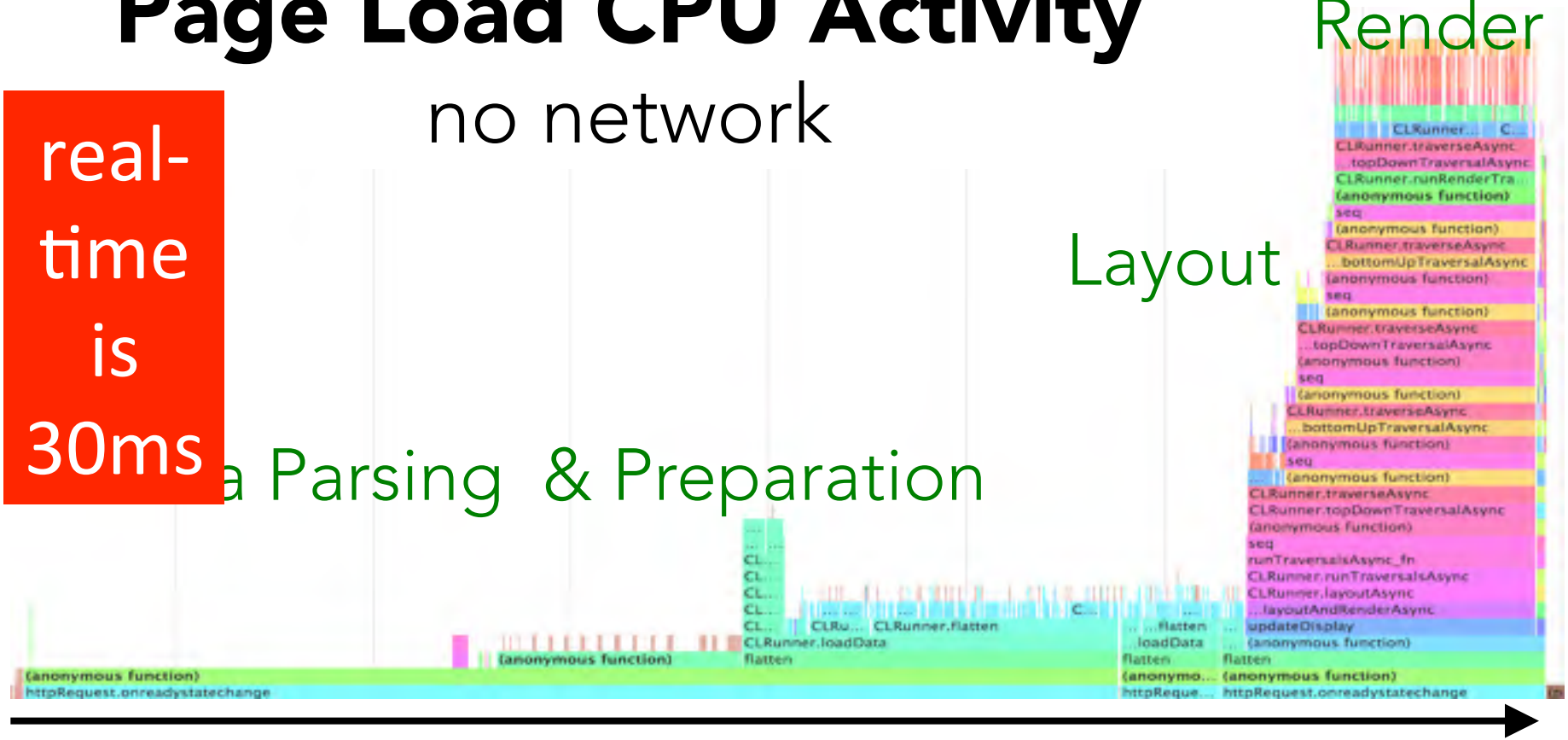




Page Load CPU Activity

no network

real-time is 30ms



0ms

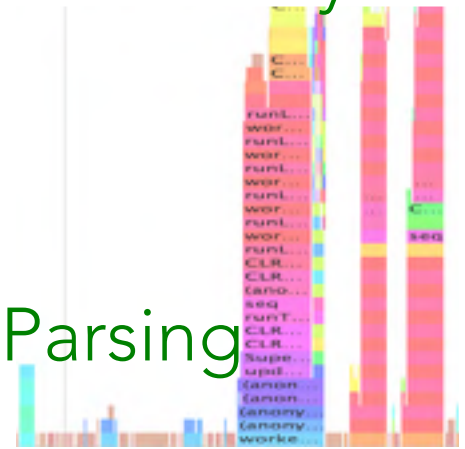
1500ms

event loop is now
80ms & non-blocking

Multicore + GPU Acceleration

Render

Layout



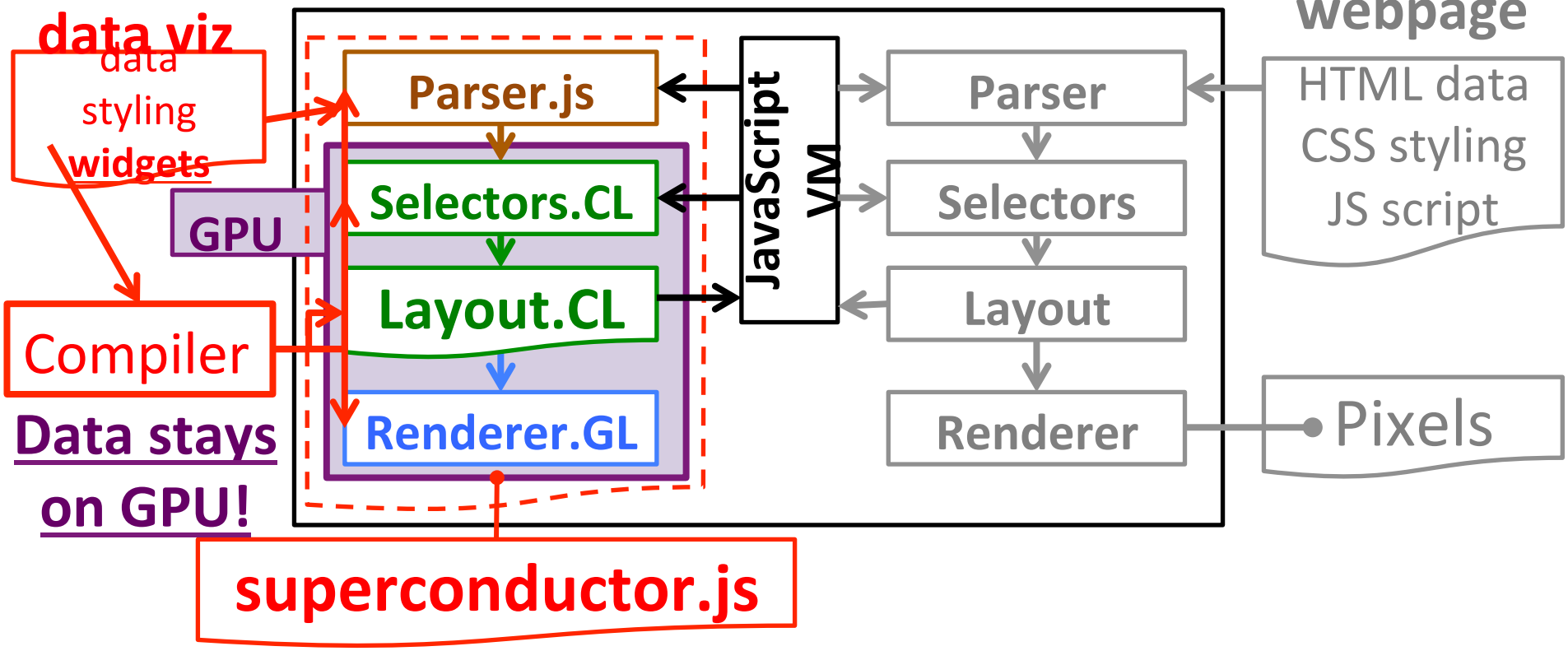
Parsing

0ms

1500ms

SUPERCONDUCTOR

Client Only Version

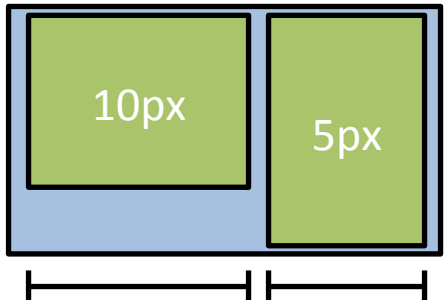


Grammar 1/2: *Schema* of Attributed Tree

[Kastens 1980, Saraiva 2003] [WWW 2010, PPOPP 2013]

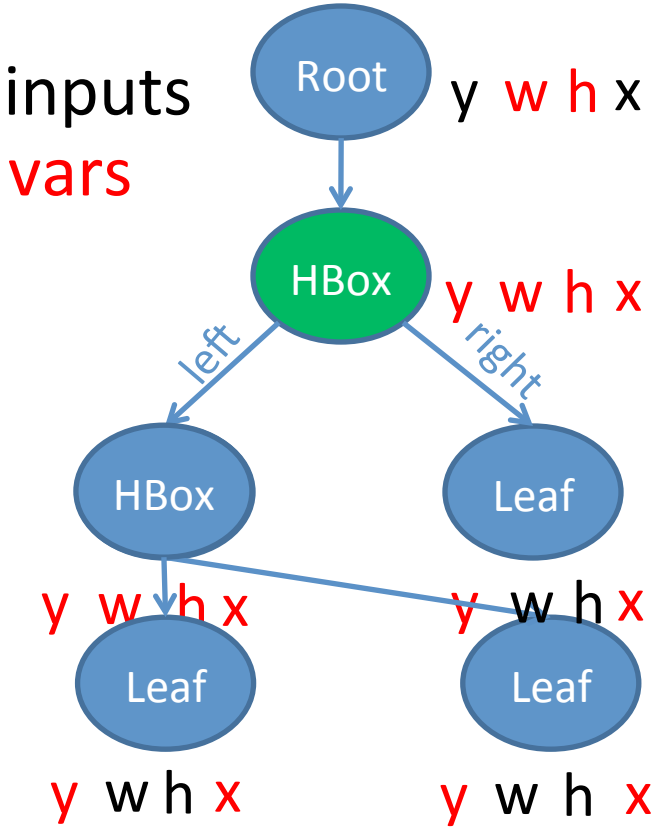
Grammar 2/2: Schema Constraints

1. Local




```
HBox → left=HBox right=Leaf
  w := left.w + right.w
  …
```

2. Single-assignment



Linguistic Extensions to Attribute Grammars

Productivity

Nominal typing (Objects):

```
   interface Node { var x }
class HBox : Node { x := 1 }
```

Traits for code reuse:

```
   trait Paint { x := 1 }
class HBox(Paint) : Node { }
```

Macro-expandable

Expressivity

Declarative loops:

```
w := id + 1
loop c in children:
    c[i].id := c[i-1].id + 1
    c[i].x := c[i].w
```

[Karp 67]

Reduce to AG scheduling:

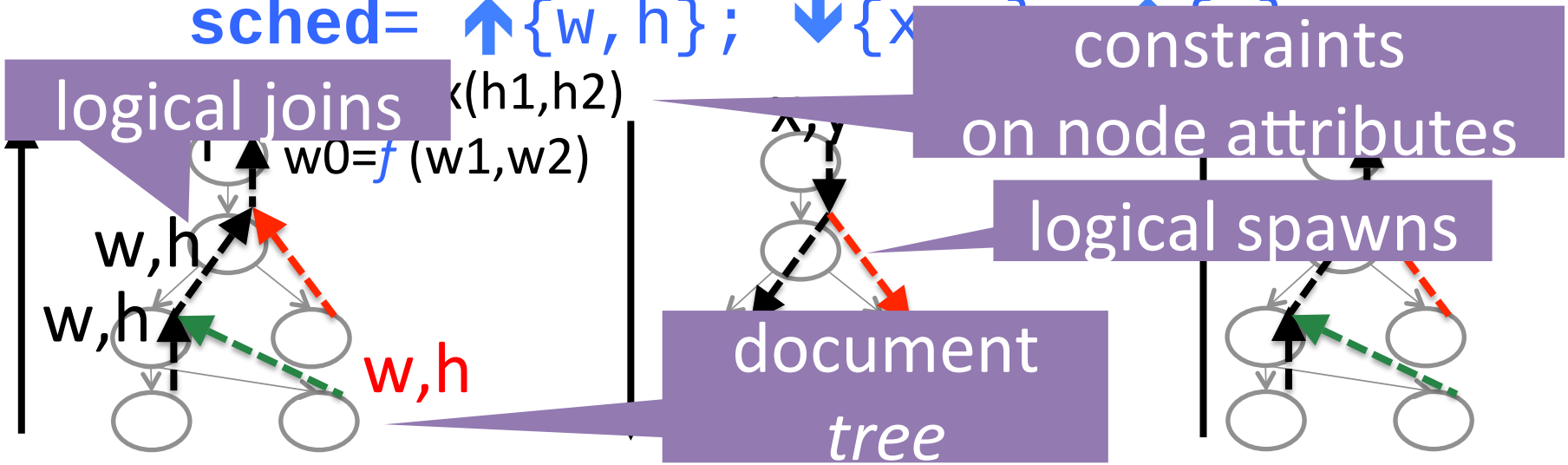
1. unroll: iteration 0, 1, 2, 3, 4
2. schedule
3. contract: loop body == iteration 2

Partial behavioral specification:
Section 3 of talk ☺

Solution: Layout as Tree Traversals

Parallel

[WWW 2010]



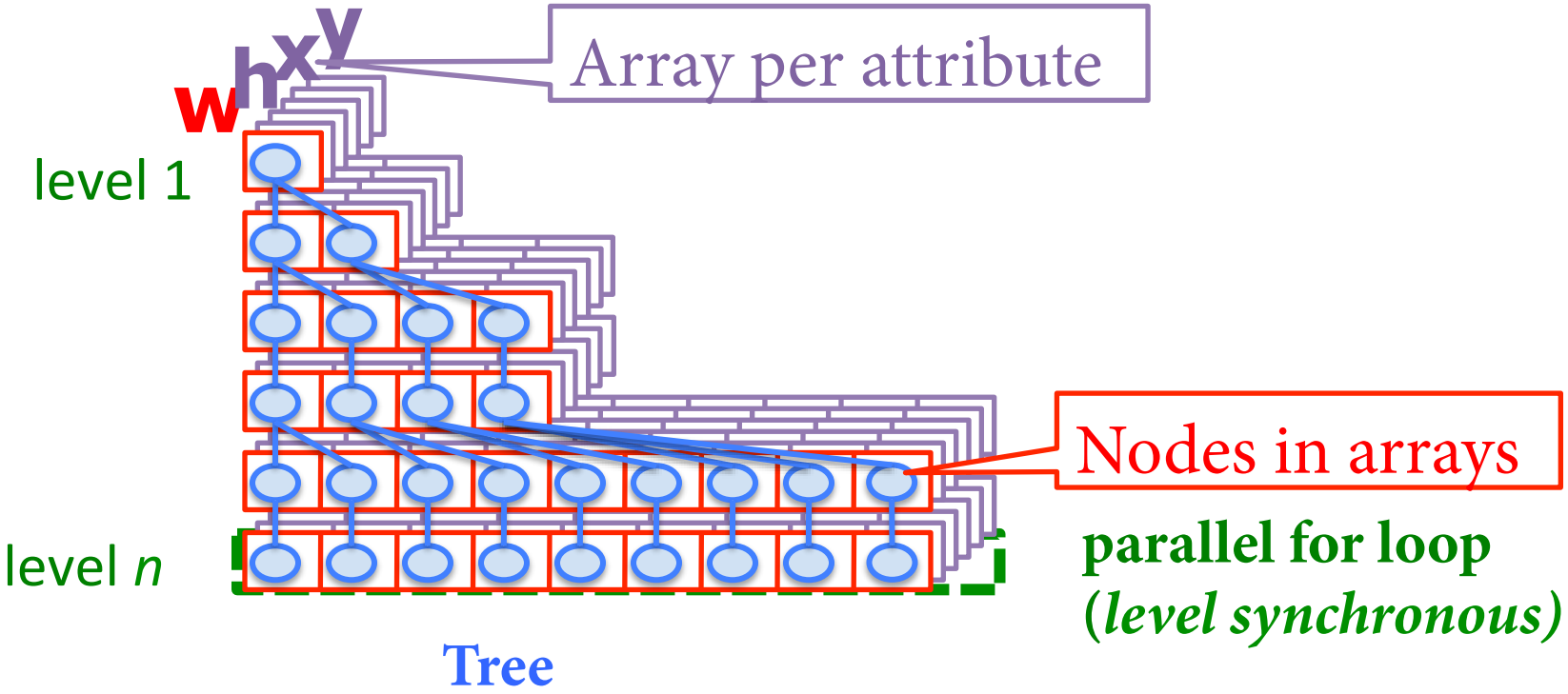
w,h w,h **Can rearrange into 9 parallel passes!**

sched valid for all webpages (in subset):

Schedule is static -- used to build the browser!

[Blelloch 93]

GPU Traversals: Flattened & Level-Synchronous



Compiler automates code + data transformations.