When Testing in Production is a Good Idea

Dan Robinson

CTO, Heap

- Joined as Heap's first hire in July, 2013.
- Previously an engineer at Palantir.
- Studied Math & CS at Stanford.

What we'll talk about:

1. What is Heap?
2. Testing in prod and why it works so well for us.
3. Some thoughts on how to generalize this approach.
4. Same concept applied to testing our client side JS.

HEAP

What is Heap?

Heap | Mobile and Web Analyti

Customers - Heap | Mobile and

https://heap.io/customers

HEAP

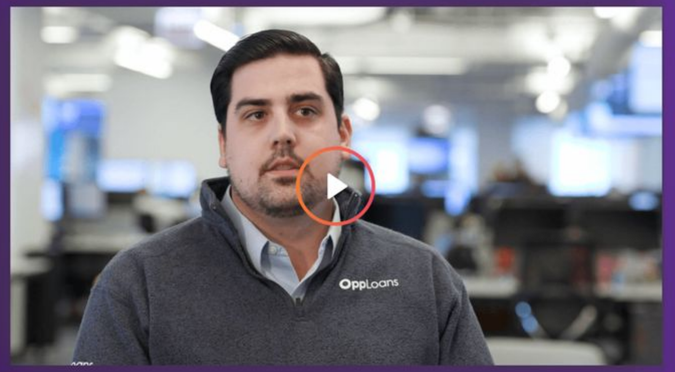
PRODUCT SOLUTIONS CUSTOMERS PRICING RESOURCES PARTNERS

LOGIN SIGN UP

opploans

"Heap is the only data tool I've ever used that's just worked out of the box, which was shocking during our proof of concept."

Matt Gomes, Director of Marketing



WATCH CUSTOMER STORY »

HARRY'S

opploans

twilio

Microsoft

AdRoll

Sur la table

esurance

Casper

```
playButton.addEventListener('click', function() {
    Analytics.track('Watched Video', {customer: 'opploans'});
});
```

HEAP

Web

To get started with Heap, paste the following code snippet before your website's closing `</head>` tag:

```
<script type="text/javascript">
  window.heap=window.heap||[],heap.load=function(e,t){window.heap.appid=e,window.heap.config=t=
  heap.load("236035469");
</script>
```

HEAP

Heap | Mobile and Web Analyti

Customers - Heap | Mobile and

https://heap.io/customers

Back to Heap | Events | Debug in Live View | Define Pageview | "Ctrl + /" to toggle modes | Definition Mode | Normal Mode

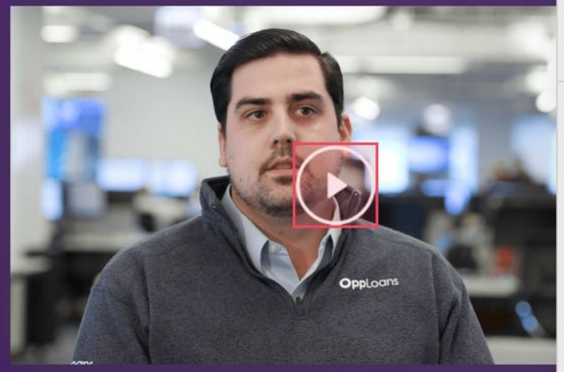
HEAP PRODUCT SOLUTIONS CUSTOMERS PRICING RESOURCES PARTNERS

oploans

"Heap is the only data tool I've ever used that's just worked out of the box, which was shocking during our proof of concept."

Matt Gomes, Director of Marketing

WATCH CUSTOMER STORY »



HARRY'S oploans twilio Microsoft AdRoll Sur la table esurance

Define Click Event

14 clicks in the past week
1 matching element on this page

Definition

#customers-slider × .active ×
[data-vid="https://fast.wistia.ne... ×

Edit Definition

Select elements to broaden or narrow your definition.

SHOW MORE

```
div .container
  div #customers-slider .owl-carousel .owl-t
  heme .owl-loaded .owl-drag
    div .owl-stage-outer
      div .owl-stage
        div .owl-item .active
          div .single-slide .video
            div .slide-asset
              div .play-cont [data-vid="http
              s://fast.wistia.net/embed/iframe/
              pshbpcgybj?videoFoam=true"]
              img .lazyloaded [data-src="http
              s://heap.io/wp-content/uploads/20
              18/12/Screen-Shot-2018-12-06-...
              [alt="video placeholder"] [src="h
              ttps://heap.io/wp-content/upload
              s/2018/12/Screen-Shot-2018-12-06-
```

Additional Filters

Limit to URL: heap.io/customers

Occurrences on 1 other page...

Next

Heap | Mobile and Web Analyti

Customers - Heap | Mobile and

https://heap.io/customers

Back to Heap | Events | Debug in Live View | Define Pageview | "Ctrl + /" to toggle modes | Definition Mode | Normal Mode

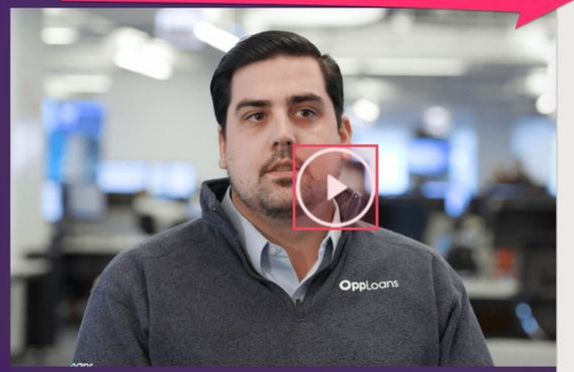
HEAP PRODUCT SOLUTIONS CUSTOMERS PRICING RESOURCES PARTNERS

opploans

"Heap is the only data tool I've ever used that's just worked out of the box, which was shocking during our proof of concept."

Matt Gomes, Director of Marketing

WATCH CUSTOMER STORY »



HARRY'S opploans twilio Microsoft AdRoll Sur la table esurance

Go Back

Create Definition

Name

Watch OppLoans Video

Visibility

Define as a personal event

Event Definition

Click on

#customers-slider .active [data-vid="https://fast.wistia.net/embed/iframe/pshbpc videoFoam=true"]

Filters

URL: heap.io/customers

Snapshots

None

Notes

Define Event

Heap | Mobile and Web Analyti

Graph - Heap | Mobile and Wel

https://heapanalytics.com/app/graph/13071279

Main
Production

Dashboard

Reports

Analyze

Graph

Funnel

Users

Retention

Influence

Paths

Queries

Define

Capture

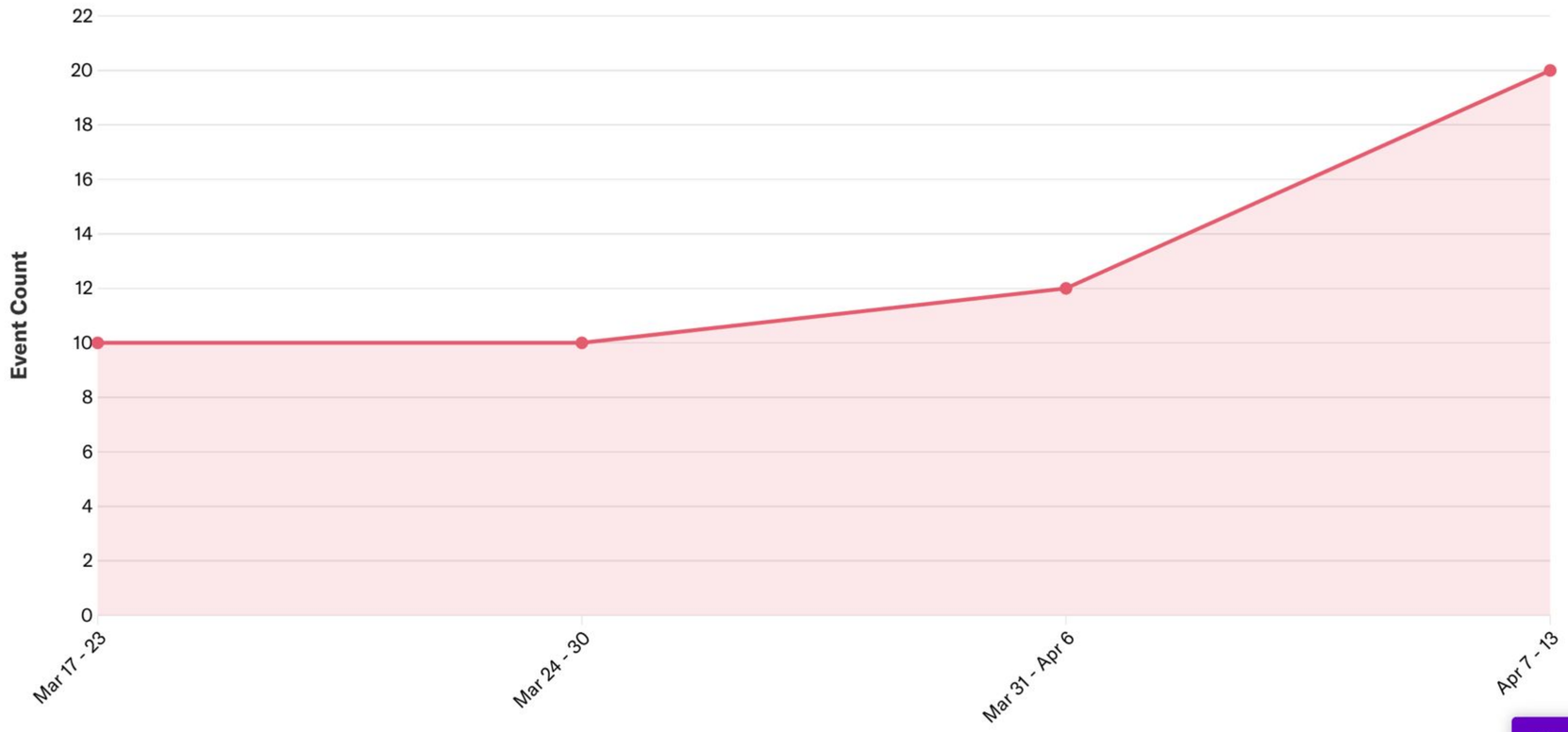
Activate

Account

Run Query

Chart Type

Watch OppLoans Video



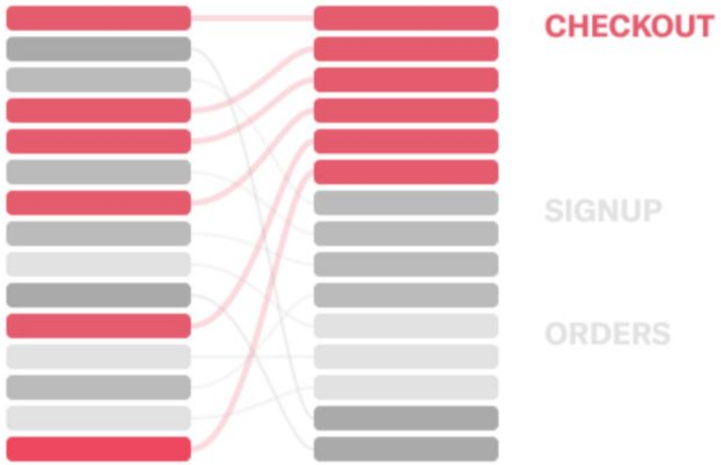
Need Help?

Challenges

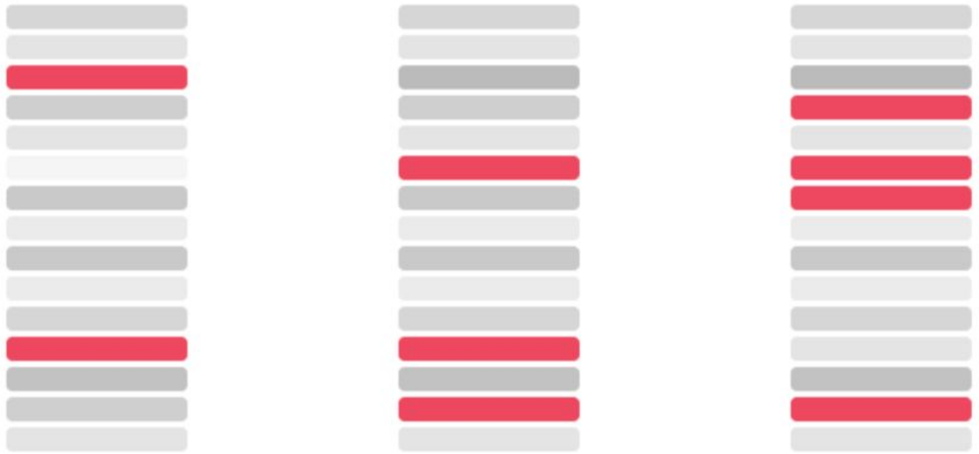
1. Capturing 10x to 100x as much data as a traditional analytics tool. Will never care about 95% of it.
2. Enormous variability in usage. Every query is unique.
3. Fundamental "indirection" in the dataset.

HEAP

DATASET PARTITIONED BY EVENT TYPE



HOWEVER WITH VIRTUALIZATION, WHAT DATA CONSTITUTES A CHECKOUT MAY CHANGE OVER TIME



HEAP

How do you make this fast?

Ground Rules

1. Need to make large, system-wide improvements.
2. Need to do so on a predictable cadence.
3. Low tolerance for breaking the product.

Case Study: Rolling out ZFS

ZFS Backstory

- We wanted filesystem-level compression.
- We built a benchmarking suite, evaluated our product extensively.
- We decided to roll it out.

HEAP

- **Weeks** into the rollout, we ran into serious problems.
- We couldn’t ingest incoming data fast enough.
- Resolving the issues took **weeks**!

This was the most thoroughly vetted analysis-layer change we had ever made.

What went wrong?

Our benchmarking had holes that are clear in retrospect.

- We were testing with disks that were less full than in prod.
- Our benchmark was a scaled-down test on a smaller machine, but the scaled up workload on a larger machine didn’t perform the same way.

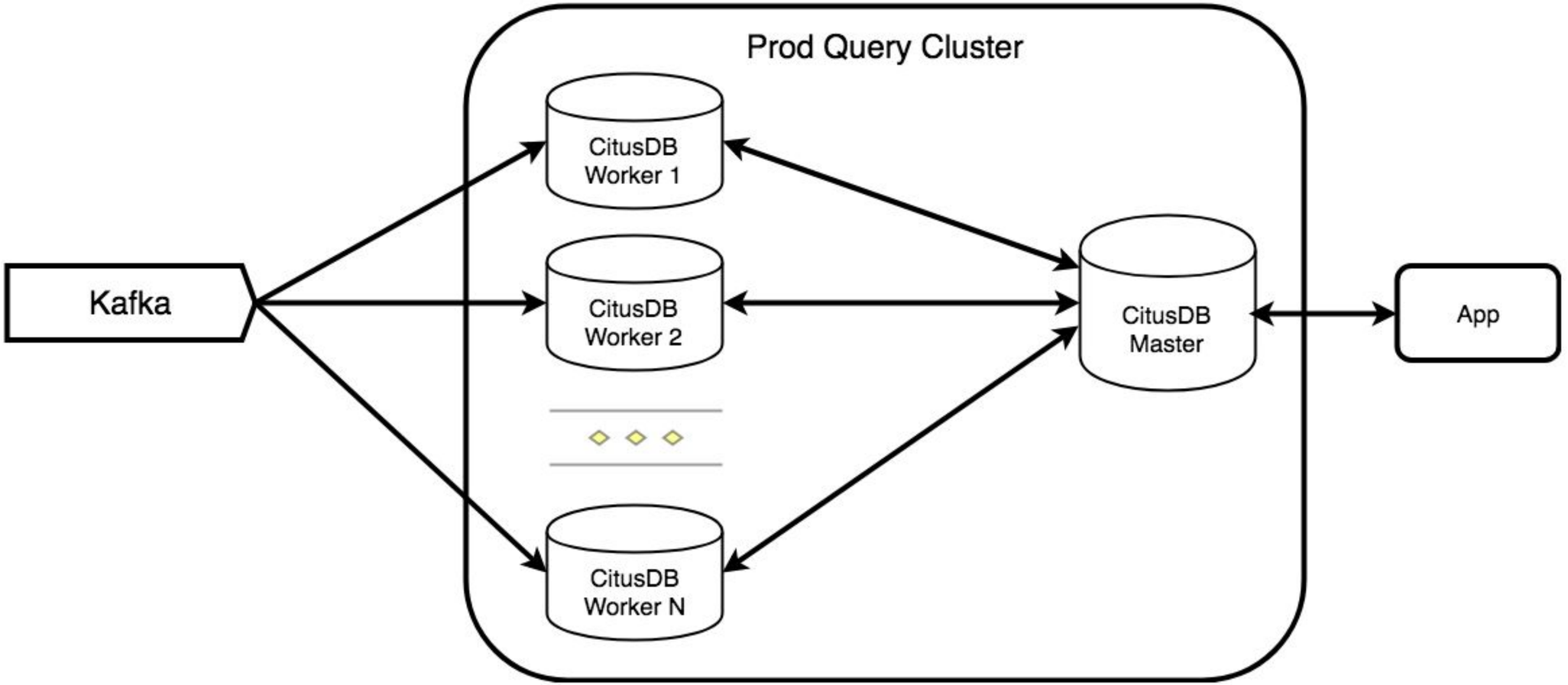
Any way your testing differs from prod is surface area for surprises in prod.

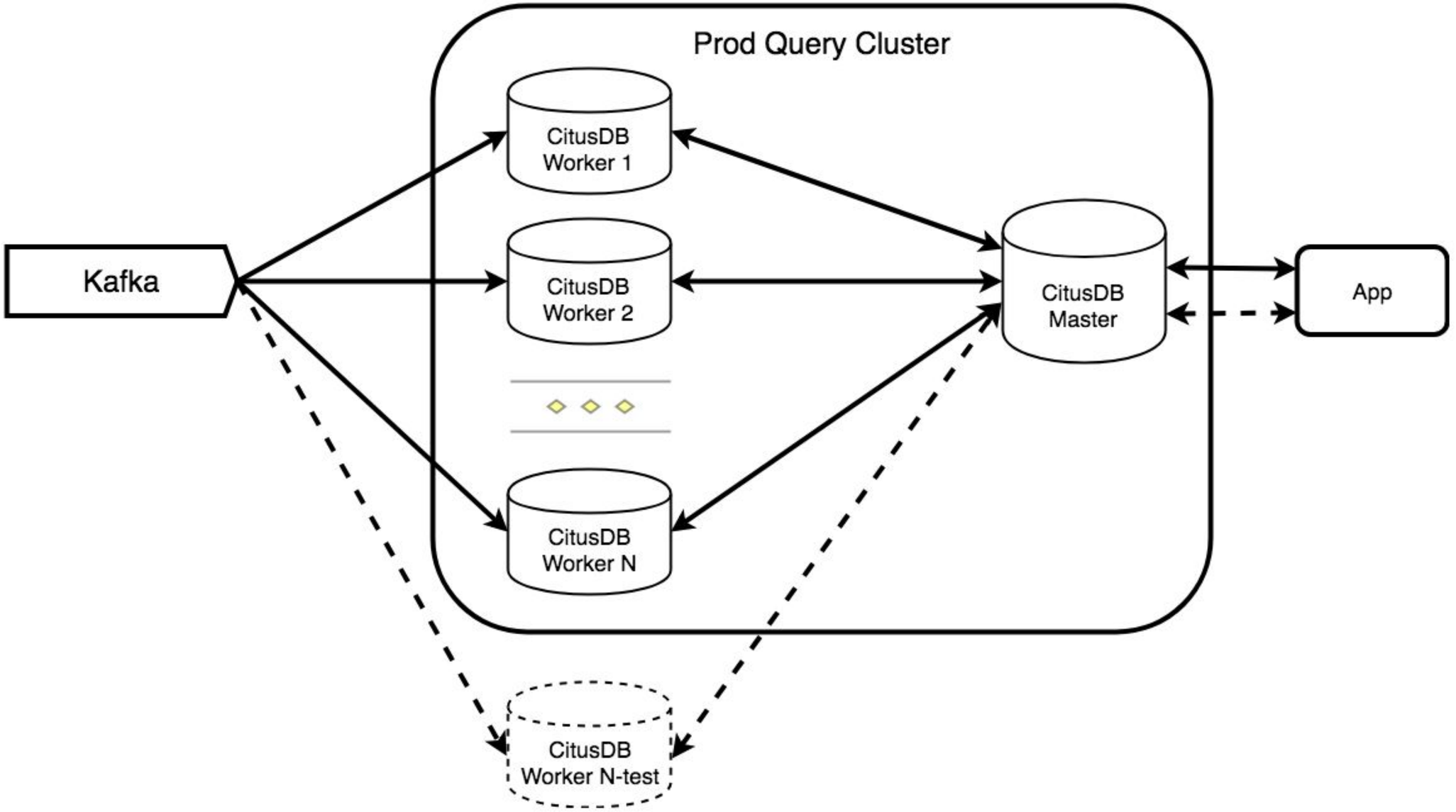
Instead of starting from a synthetic benchmark and making it increasingly sophisticated, why not build a way to test your idea in prod, without the risk?

"Shadow Prod"

- Our query cluster has a master and N workers. (N = 70 right now.)
- We built a system that picks a worker and creates a “shadow” copy of it, with our desired change.
  - We duplicate the dataset exactly on the shadow machine.
  - We mirror all reads and writes.
- This machine is in prod, except that we ignore reads from it.

HEAP


Prod Query Cluster
CitusDB
Worker 1
CitusDB
Worker 2
CitusDB
Worker N
Kafka
CitusDB
Master
App


Prod Query Cluster
CitusDB
Worker 1
CitusDB
Worker 2
CitusDB
Worker N
CitusDB
Master
Kafka
App
CitusDB
Worker N-test

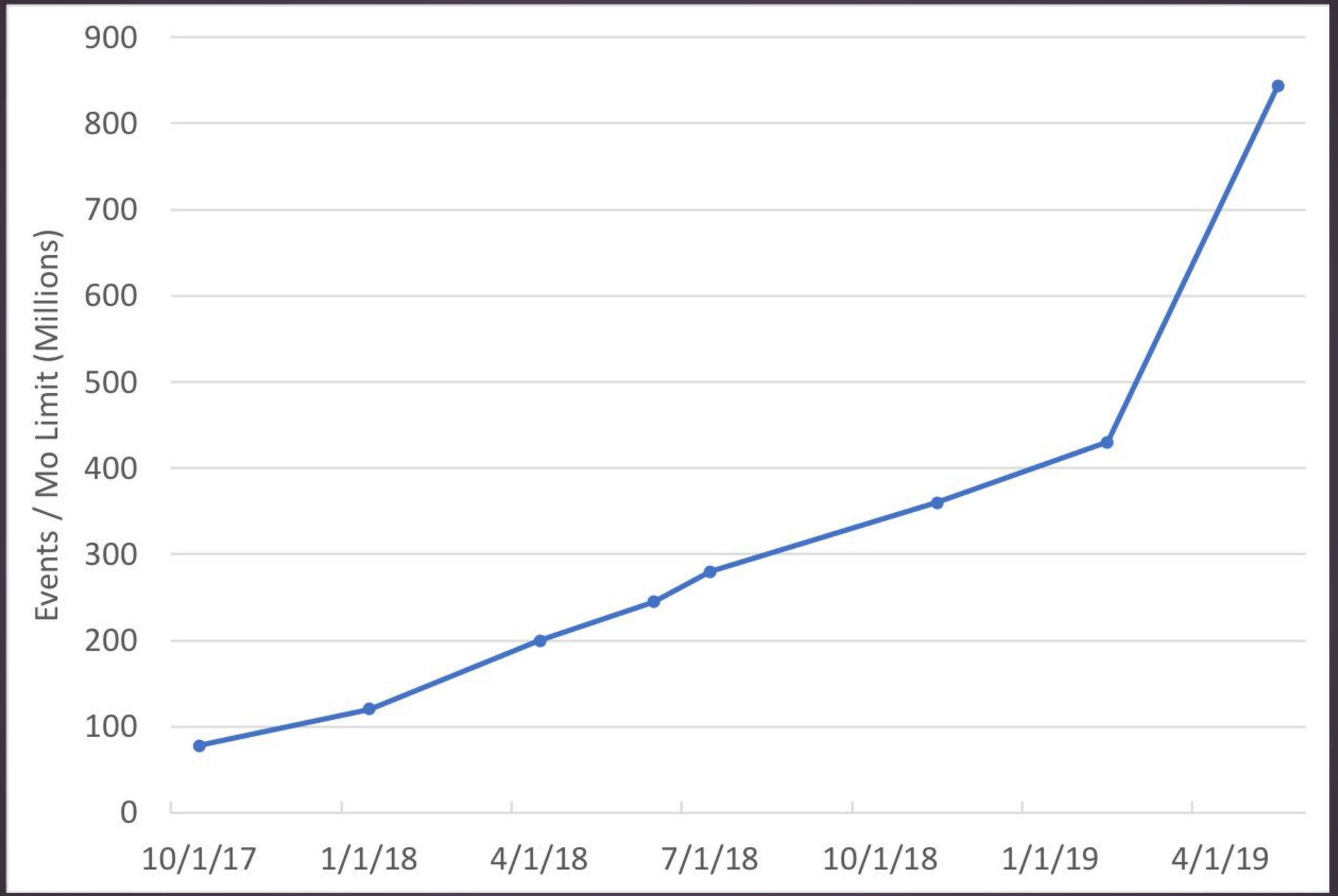
"Shadow Prod" Results

- Evaluating a change takes 2-4 weeks of wall time, most of which is passive.
- We’re improving query perf by 20% to 40% per quarter, reliably.
- We're up 11x in the last 18 months.
- We have a two person database team.

HEAP

System Level	*Example*	*Result*
Hardware	i3.16xlarge vs i3.metal	41% p95 improvement
OS Config	Clock Source xen vs tsc	30% p95 improvement
Filesystem Config	ZFS Recordsize 8kb vs 64kb	2.4x reduction in disk footprint
DB Schema	Partitioning event table by top-level type	22% p95 improvement
Indexing Strategy	Including user IDs in event indexes	20% p95 improvement

HEAP


900
800
700
600
500
400
300
200
100
0
Events / Mo Limit (Millions)
10/1/17
1/1/18
4/1/18
7/1/18
10/1/18
1/1/19
4/1/19

"Shadow Prod" Results

- Easy to be confident that a change is safe for prod, because it's already in prod.
- Bonus: this system tests the rollout process for free, because you use it to create shadow nodes.

HEAP

Protips

Protip: use A/A tests to expose confounding variables.

Protip: the ability to align specific atoms in your experiment between prod and shadow prod is key.

Protip: build a sanity checker to make sure the improvements you're getting make sense.

Foreseeable Issues

Unforeseen Issues

HEAP

Foreseeable Issues

Unforeseen Issues

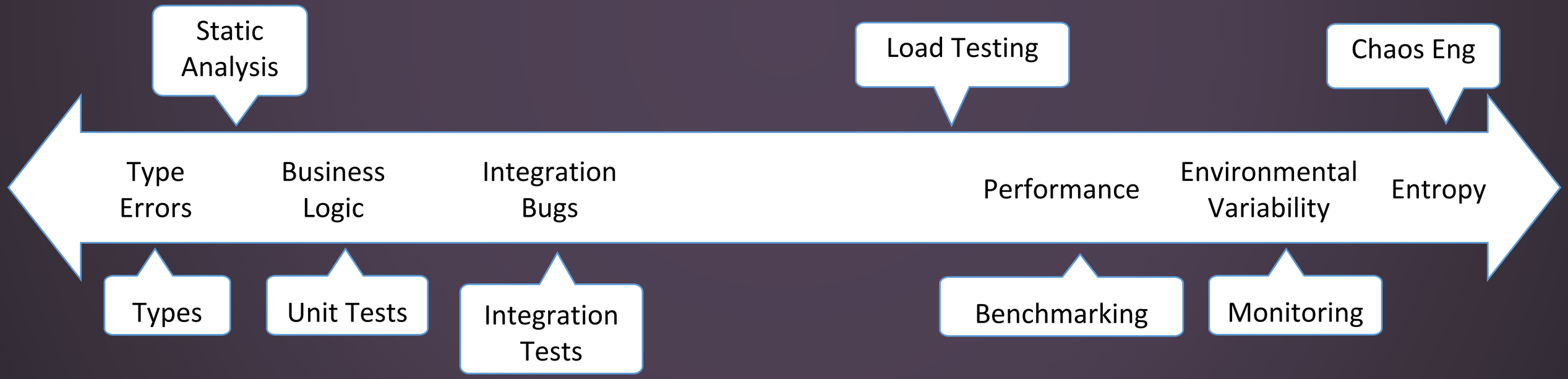
HEAP

Local Tests

Foreseeable Issues

System Tests

Unforeseen Issues



HEAP

- The problem of query perf at Heap has enormous variability.
- Trying to predict all this variability is very difficult, let alone reproducing it in a benchmark.

What would a perfect benchmark handle?

- Sequences of queries typically use the same events repeatedly.
- Different shapes of dataset for different customers.
- People generally use new events right after they define them.
- Intra-week patterns, intra-month patterns.
- Bursty usage – log into your account once a week but run 30 queries.
- Drilldown / pivot workflows, e.g. "compute my funnel, now show me example users who dropped off at step 3."
- The visualizer has its own specific usage pattern.
- Writes for 1b events / day are intermingled in all of this.

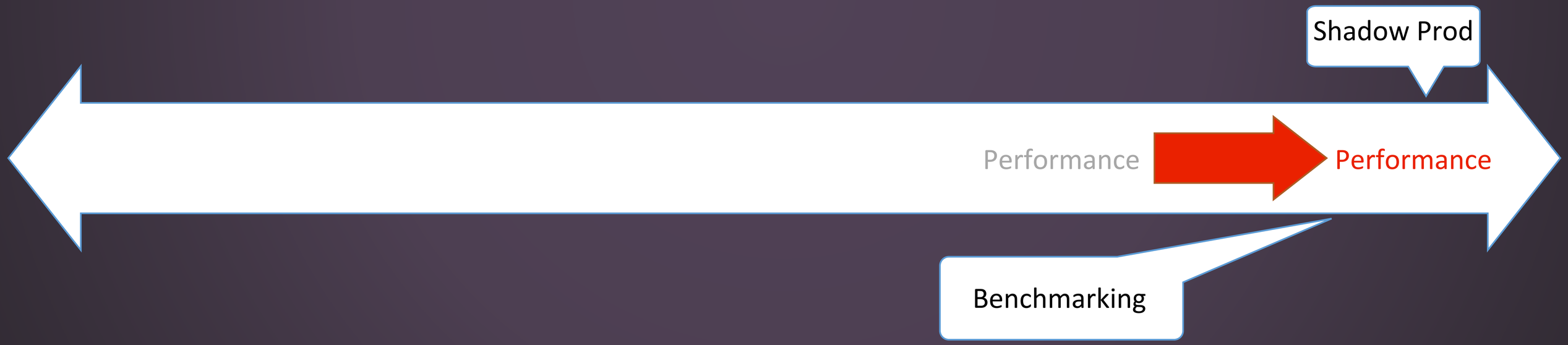
HEAP

Local Tests

Foreseeable Issues

System Tests

Unforeseen Issues



HEAP

In a context with very large variability, you might be better off finding a way to test safely in prod, so as to expose your code to that variability, rather than trying to capture it in tests or benchmarks.

If you have a lot of variability, think "test in prod?"

Testing Client Side JS

- Powering our product is a javascript snippet that runs on every customer's website.
- This javascript is very sensitive – can break a customer's dataset *or* their website!

HEAP

Testing Client Side JS

- We've built an extensive integration test suite to test across browsers, OSes, different website designs...
- But the variability is endless.

HEAP

We’re building out a “shadow heap.js” with the same principle:

capture the variability by getting new code into prod in a safe way.

Testing Client Side JS

- The basic principle is to load two versions of heap.js on select customers' sites.
- We can correspond the events each version captures and compare for any diffs.
- Similarly, we can discard data from the “shadow heap.js” version.

HEAP

Geoff

Kent

Michael

Dan

Enoch

Gediminas

Andrew

HEAP

Questions?

Or, ask me on twitter: @danlovesproofs