



Dashboard debugging and perf

David Kaltschmidt

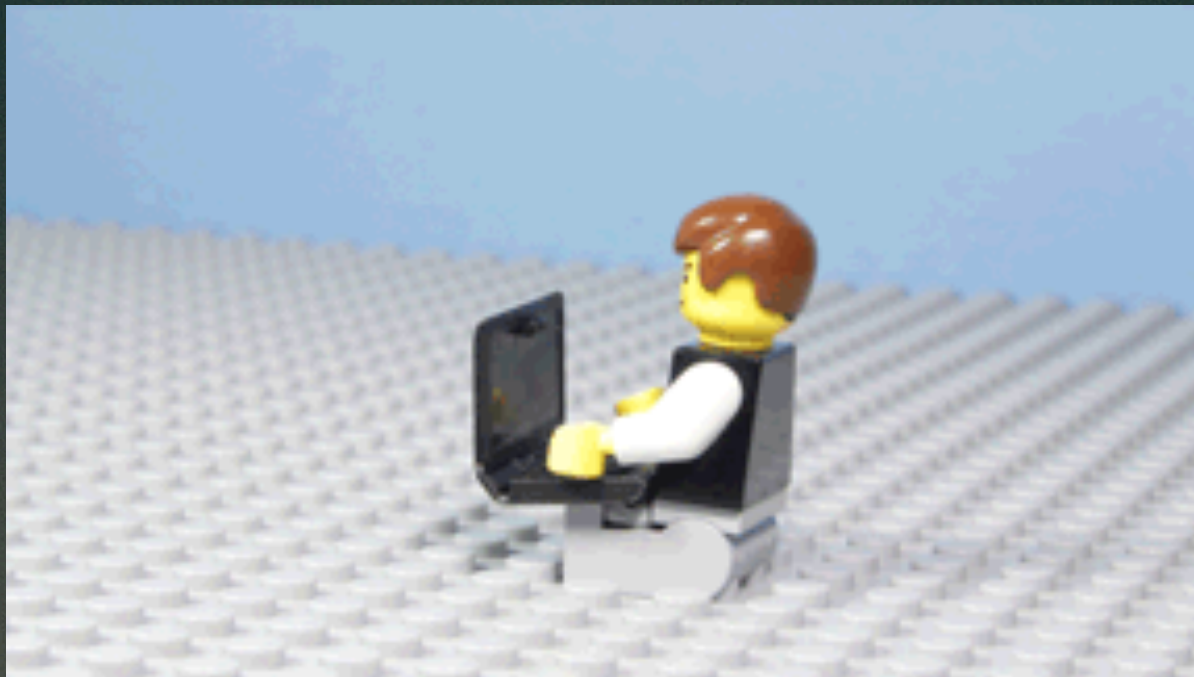
@davkals

Grafana Labs

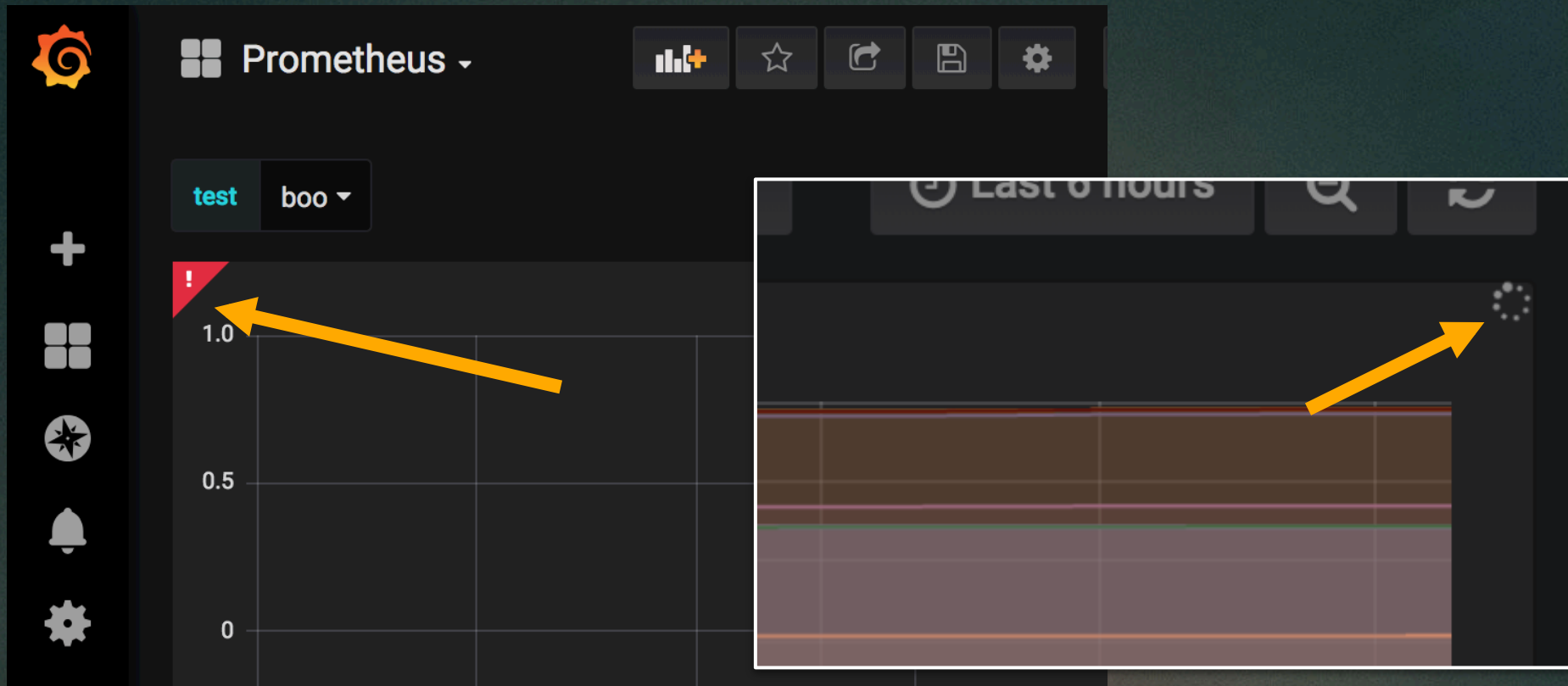
My graphs are not loading!



My graphs are **too slow!**



My graphs are not loading!



Debugging broken panels

[Link to Grafana](#)



Name	Prometheus PA	Default	<input type="checkbox"/>
------	---------------	---------	--------------------------

HTTP

URL	http://localhost:9090/prometheus	i
Access	Server (Default)	Help
Whitelisted Cookies	Add Name	i

Auth

Basic Auth	<input type="checkbox"/>	With Credentials	i	<input type="checkbox"/>
TLS Client Auth	<input type="checkbox"/>	With CA Cert	i	<input type="checkbox"/>
Skip TLS Verify	<input type="checkbox"/>			

Scrape interval	15s	i
Query timeout	60s	i
HTTP Method	GET	i



HTTP Error Not Found

Save & Test


Delete

Back

Debug tooling: Data Source settings test



Debug tooling: Query Inspector

Queries to  Prometheus RB ▼ Add Query Query Inspector ?

Query Inspector ✕

▼ Object ⊕ Expand All 📄 Copy to Clipboard

```
xhrStatus: "complete"
▼ request: Object
  method: "GET"
  url: "api/datasources/proxy/31/api/v1/query_range?query=up%7Binstance%3D%22demo.robustperception.io%3A9090%22%7D&start=1551180840&end=1551202460&step=20"
▼ response: Object
  status: "success"
  ▼ data: Object
    resultType: "matrix"
    ► result: Array[1]
```

Debug tooling: Query Inspector and beyond

The screenshot shows the Grafana Query Inspector interface. At the top, there is a time range selector from 04:00 to 09:30. Below this, the 'Queries to' dropdown is set to 'Prometheus RB'. To the right are buttons for 'Add Query', 'Query Inspector', and a help icon. The main panel is titled 'Query Inspector' and shows 'Loading query inspector...' with a loading spinner. Below this, a query editor shows the query `up{instance=$instance}`. At the bottom, there are tabs for 'Legend', 'legend format', 'Min step', and 'Resolution', with the 'Resolution' tab currently selected, showing '1/1'.

04:00 04:30 05:00 05:30 06:00 06:30 07:00 07:30 08:00 08:30 09:00 09:30

Queries to Prometheus RB Add Query Query Inspector ?

Query Inspector x

Loading query inspector... ⌂

▼ A ↓ ↑ 📄 👁 🗑

```
up{instance=$instance}
```

Legend legend format ⓘ Min step ⓘ Resolution 1/1 ▼

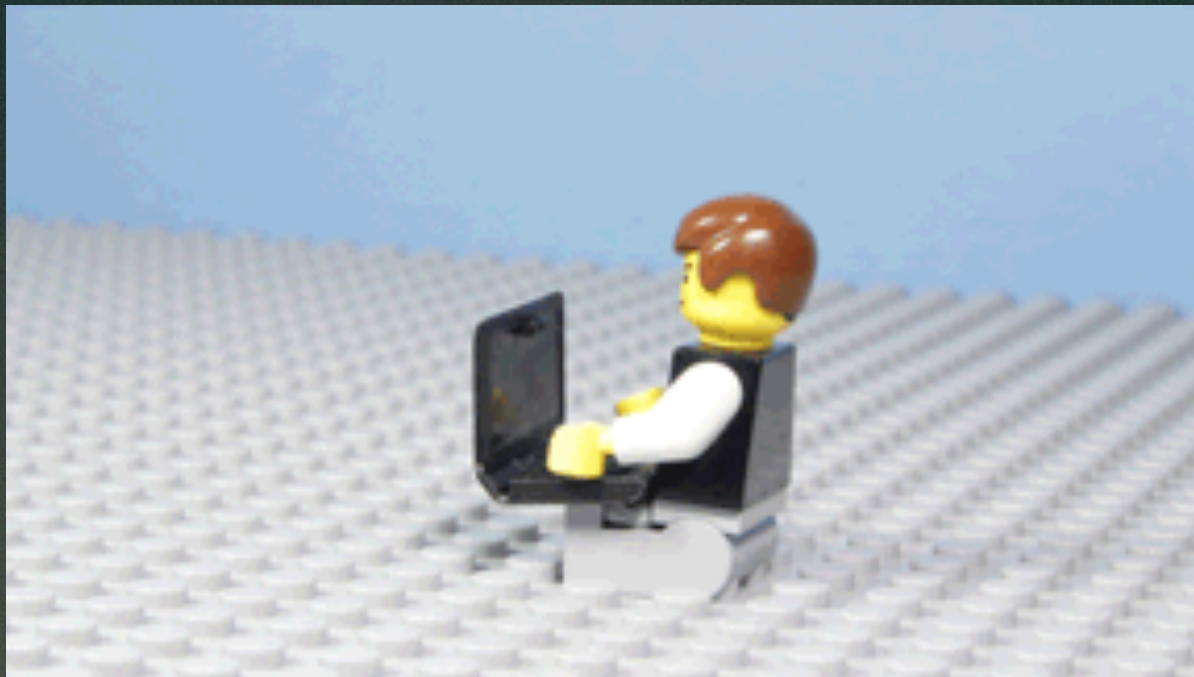
Debug tooling: Chrome dev tools

The screenshot shows the Chrome DevTools interface with the Network tab selected. The top bar indicates the page is at `localhost:3333/datasources/edit/3/`. The Network tab shows a list of requests with a filter set to 'XHR'. The selected request is `search?limit=30&starred=true`. The response preview shows a JSON error object:

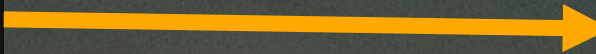
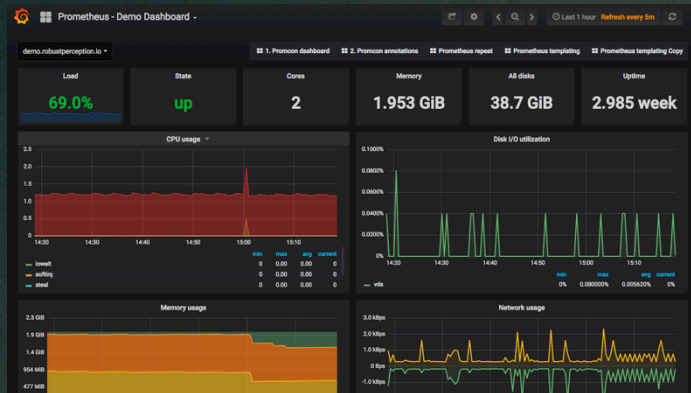
```
{status: "error", errorType: "bad_data",...}
error: "parse error at char 13: unexpected identifier "demo" in label matching, expected string"
errorType: "bad_data"
status: "error"
```

The bottom status bar shows 172 / 195 requests and 1.6 MB / 37.9 MB transferred.

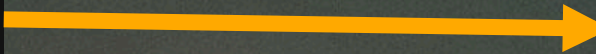
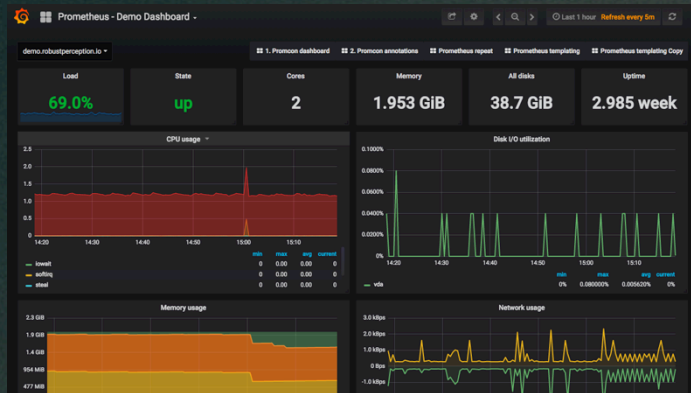
My graphs are **too slow!**



Understanding latency



Understanding latency



Name

Prometheus PA

Default



HTTP

URL

http://localhost:9090/prometheus



Access

✓ Server (Default)

Browser

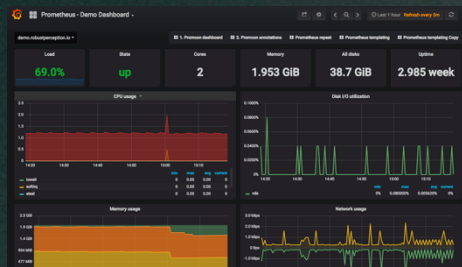
Help ▶

Whitelisted Cookies

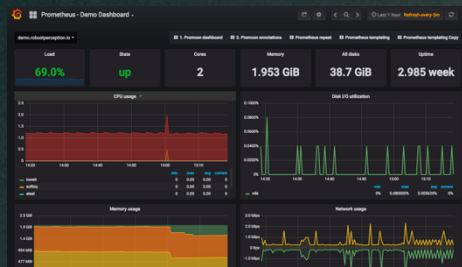
Add Name



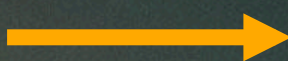
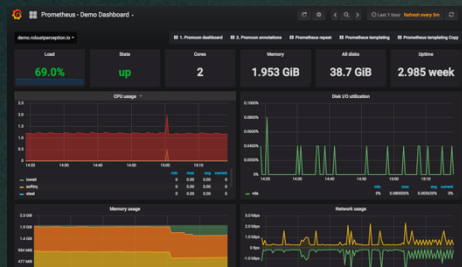
Understanding latency w/ server



Request latency with dev tools

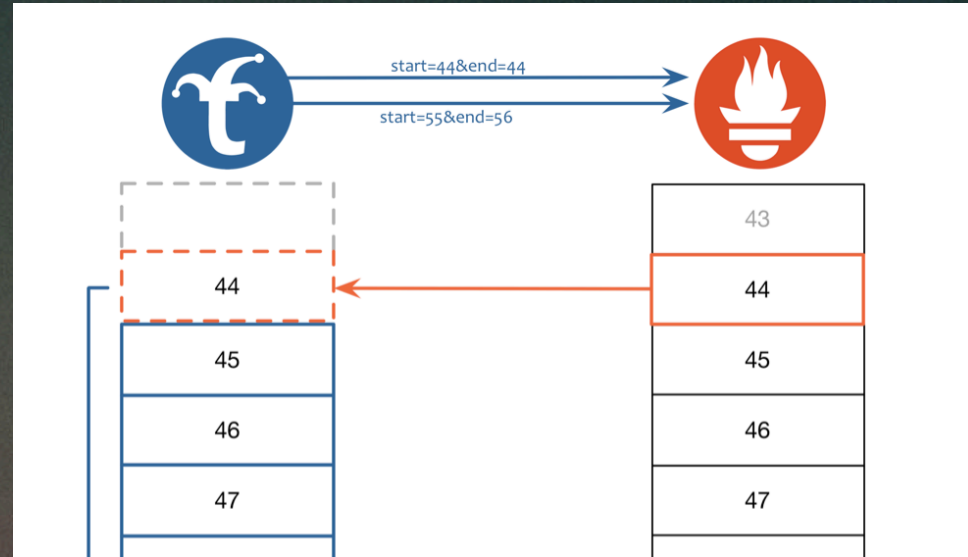
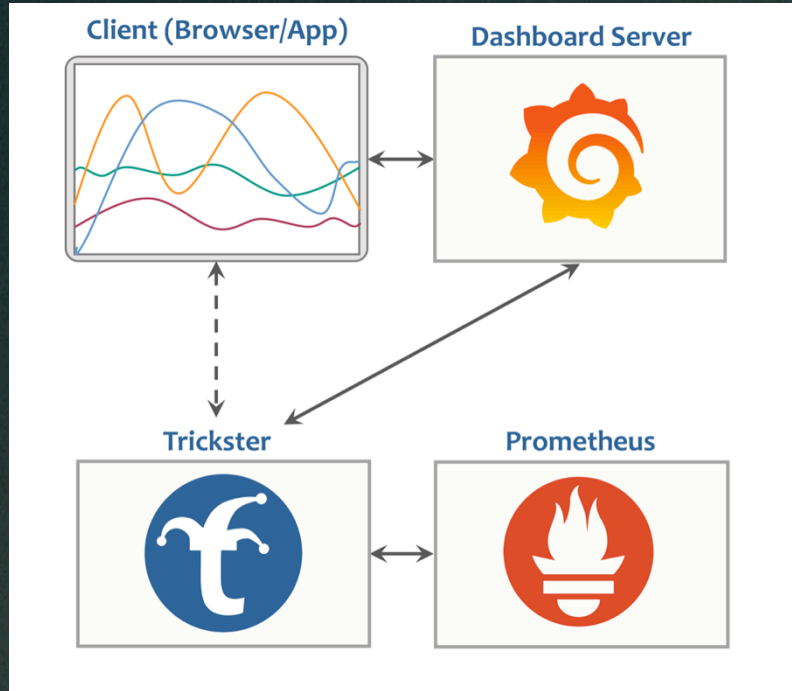


Network latency vs ds latency vs plotting latency

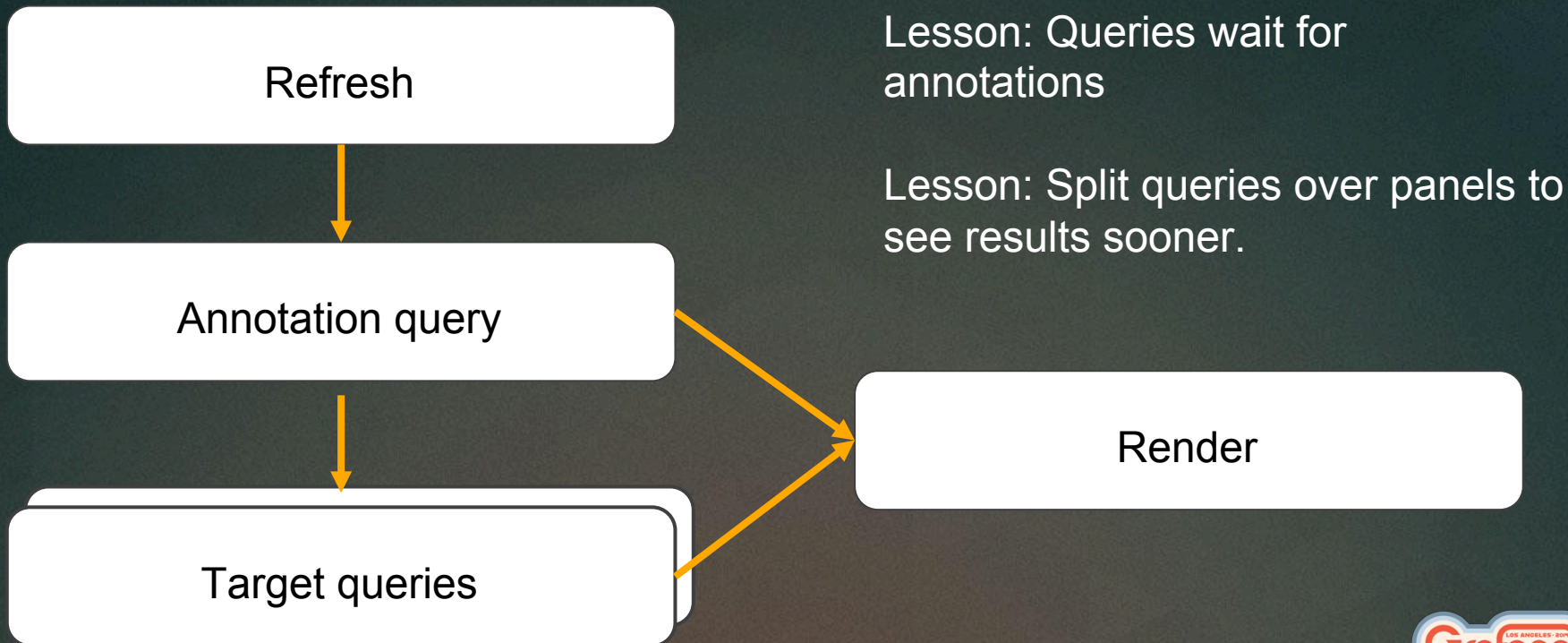


<http://localhost:3333/explore>

Intermission: Trickster, a reverse proxy cache



Graph panel requests: queries wait for all



Lesson: Tweaks with little impact

- Time range
 - Effect on data source latency/has to load more data
 - Some data sources have explicit step parameter for resolution
 - Limited effect on render times because data is sampled down

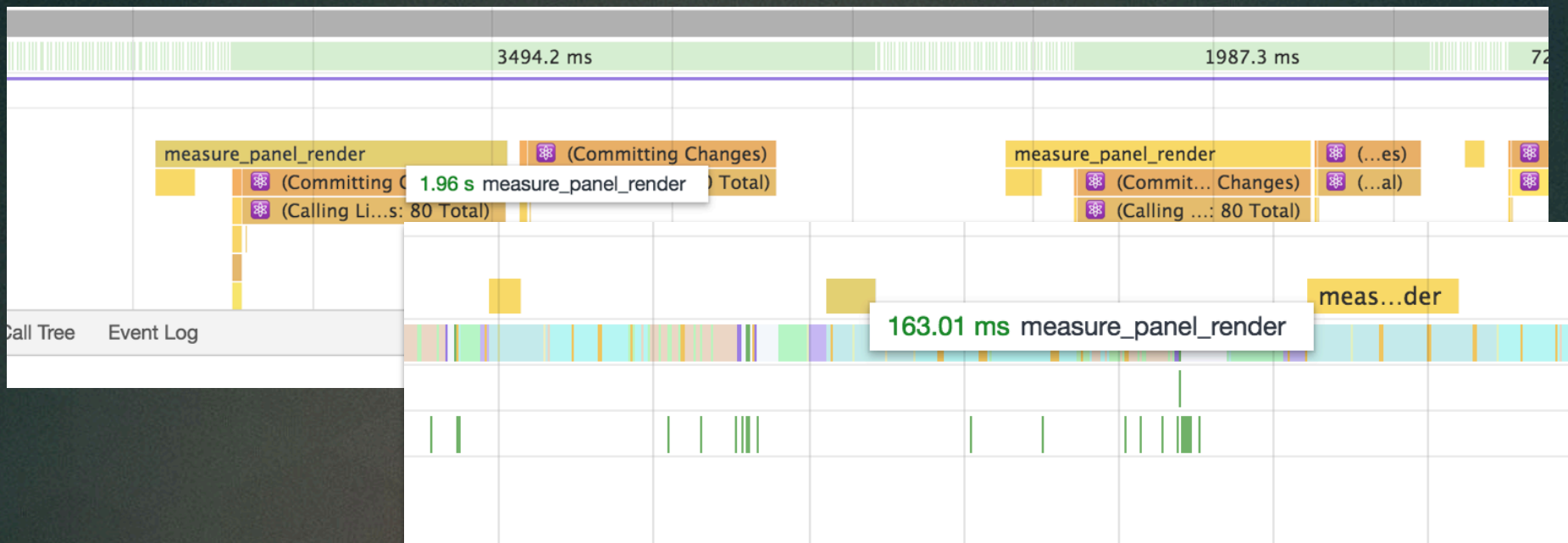
Lesson: Tweaks with more impact

- Query limits that reduce number of time series
 - Graphs render times are similar:
everything is rendered on canvas
 - BUT: legend items are expensive

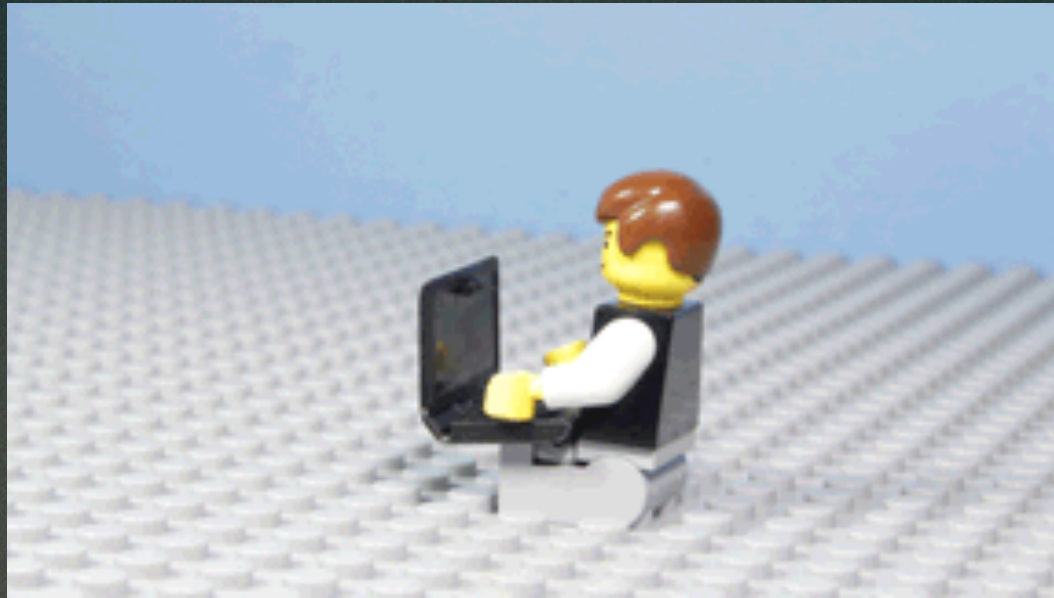


Lesson: high impact tweak

- Hide legend items (rely on hover instead)



THANKS!



David Kaltschmidt

@davkals

