

real-time analytics for
small data teams



matt helm

head of analytics

materialize





shopify

merchant analytics

11 years



materialize

internal analytics

1 year

12 years building analytics

analytics team



matt

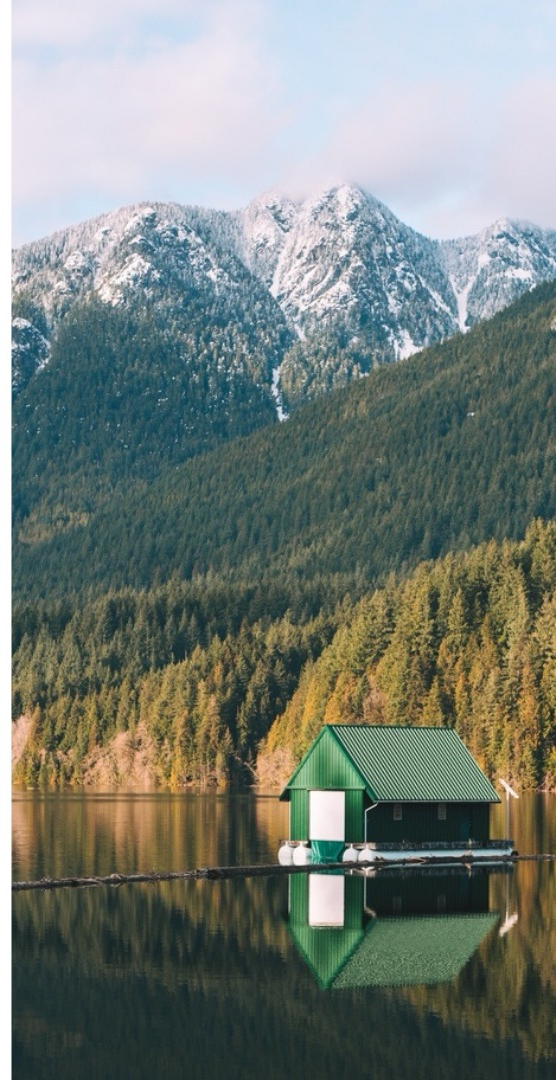


jonathan

2%



our team is a fraction the size of Shopify's analytics team



batch
8 hours

hot-loop
15 minutes

real-time
2 seconds



“i am an enthusiast,
not a sponsor”

matt helm

historical reporting



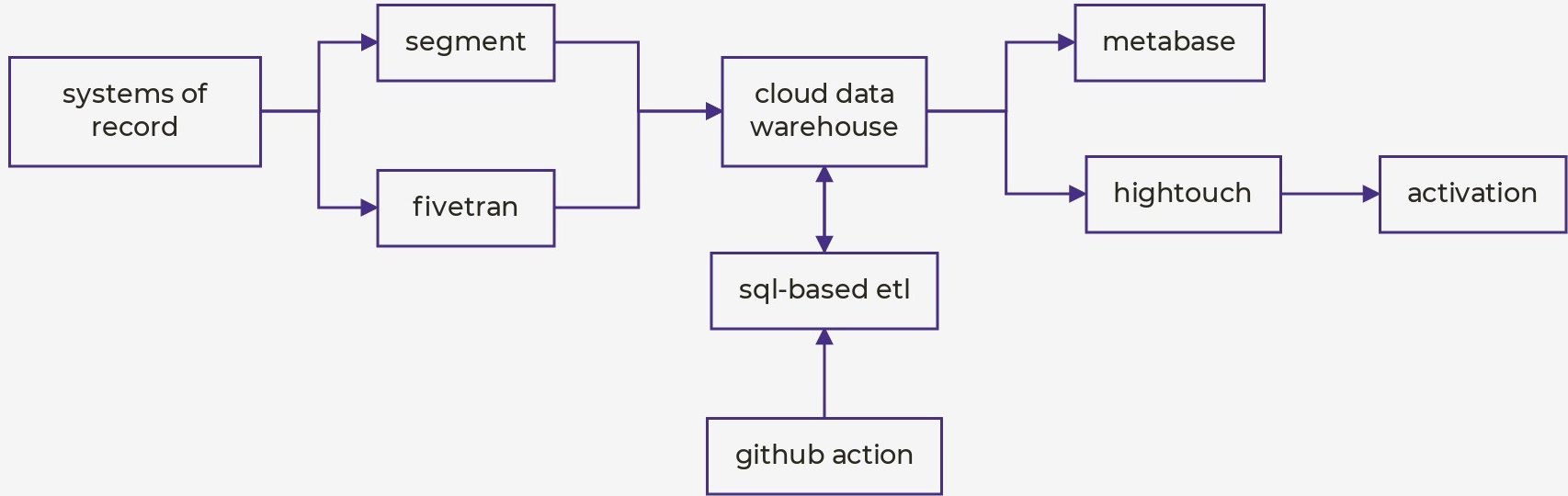
day-to-day metrics



business reviews

it's okay if these are a few hours out-of-date

architecture



sql-based etl

```
-- purchases.sql
CREATE OR REPLACE TABLE purchases AS (
  ...
)
```

```
-- run.yml
on:
  schedule:
    - cron: '0 */8 * * *' -- every 8 hours
```

operational workflows



authentication
expiration dates



LaunchDarkly

access control
resource limits

PagerDuty

alerting
customer support



marketing
product marketing

we'd like these to be real-time

batch
8 hours

hot-loop
15 minutes

real-time
2 seconds

sql-based etl

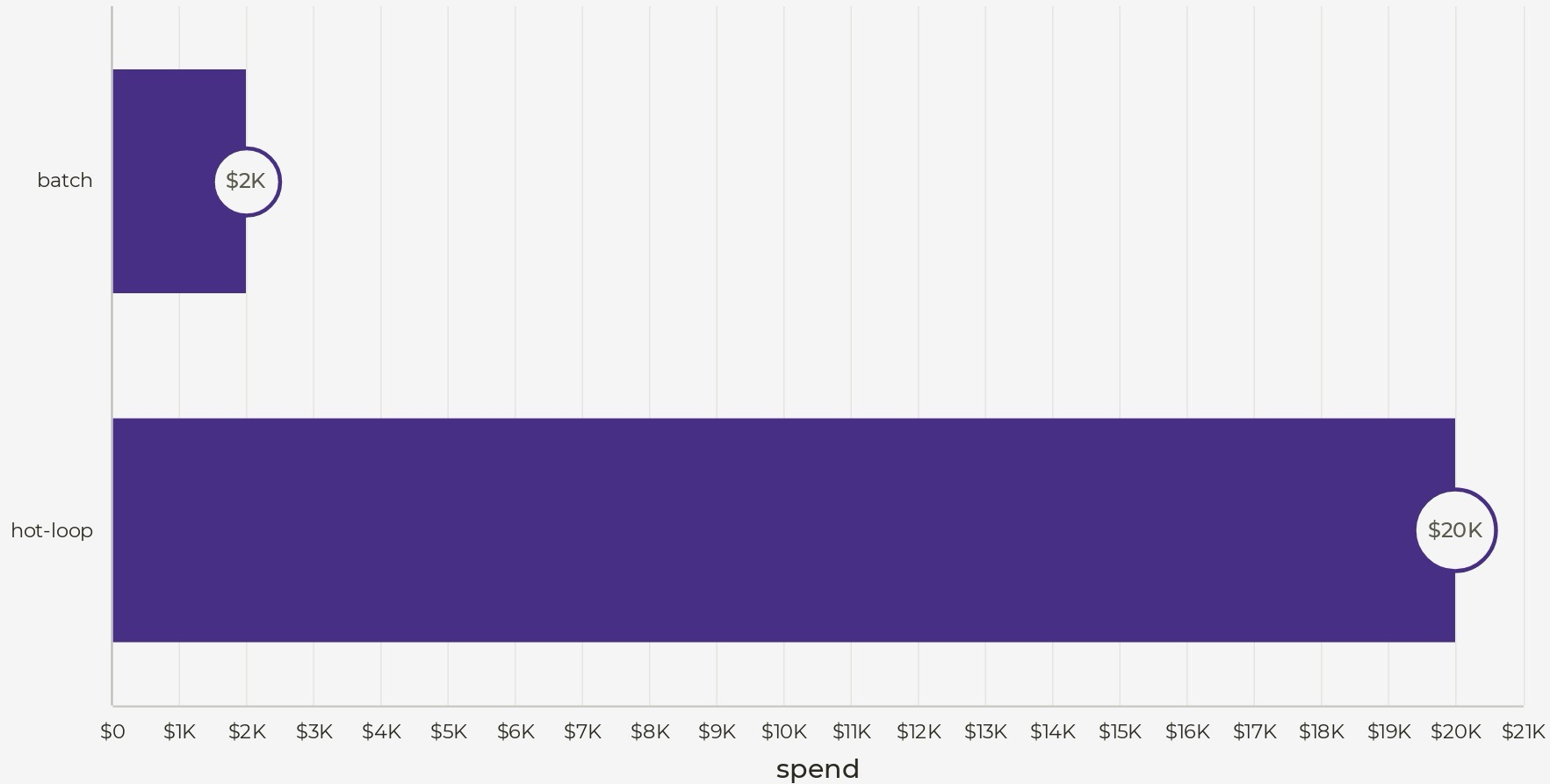
```
-- purchases.sql
CREATE OR REPLACE TABLE purchases AS (
  ...
)
```

```
-- run.yml
on:
  schedule:
    - cron: '* /10 * * * *' -- every 10 minutes
```

latency

15 minutes ↓ 97%

vs 8 hours with batch



hot-loop

more complex than batch processing



noisy errors in slack

running dbt 4,000 times per month meant rare but flaky errors created meaningful noise in slack



orchestrator

an increasing number of dbt pipelines meant we couldn't run our whole dbt project each time



sql-based etl



drop and re-build

new data on each refresh



deploy once

always up to date

the existing approach seems wasteful



page

discussion

view source

history

Want to edit, but don't see an edit button when logged in? [Click here.](#)

Incremental View Maintenance

This page describes Incremental View Maintenance (IVM) proposed in pgsq-l-hackers.

Contents [\[show\]](#)

Overview

PostgreSQL has supported materialized views since 9.3. This feature is used to speed up query evaluation by storing the results of specified queries. One problem of materialized view is its maintenance. Materialized views have to be brought up to date when the underlying base relations are updated.

Incremental View Maintenance (IVM) is a technique to maintain materialized views which computes and applies only the incremental changes to the materialized views rather than recomputing the contents as the current REFRESH command does. This feature is not implemented on PostgreSQL yet. Implementing this into PostgreSQL core was proposed firstly at [PgCon 2013 Developer Meeting](#) and [\[1\]](#) [\[2\]](#) [\[3\]](#). There were also other discussions on IVM [\[3\]](#) [\[4\]](#) [\[5\]](#). The first patch was submitted in 2019 [\[5\]](#) [\[6\]](#).

Basic Theory of IVM

IVM computes and applies only the incremental changes to the materialized views. Suppose that view V is defined by query Q over a state of base relations D . When D changes $D' = D + dD$, we can get the new view state V' by calculating from D' and Q , and this is re-computation performed by REFRESH MATERIALIZED VIEW command. On the other hand, IVM calculates the delta for view (dV) from the base tables delta (dD) and view definition (Q), and applies this to get the new view state, $V' = V + dV$.

In theory, the view definition is described in a relational algebra (or bag algebra) form. For example, a (inner) join view of table R and S is defined as $V = R \bowtie S$.

When table R is changed in a transaction, this can be described as $R \leftarrow R - \nabla R + \Delta R$, where ∇R and ΔR denote tuples deleted from and inserted into R , respectively. (To be accurate, instead of $-$ and $+$, \setminus and \cup are used by tradition in bag algebra.) In this condition, the deltas of the view are calculated as $\nabla V = \nabla R \bowtie S$ and $\Delta V = \Delta R \bowtie S$, then the view can be updated as $V \leftarrow V - \nabla V + \Delta V$.

Multiple Tables Modification

For example, suppose that we have a view V joining table R, S and T , and new tuples are inserted to each table, dR , dS , and dT respectively.

```
V = R*S*T
R_new = R + dR
S_new = S + dS
T_new = T + dT
```

navigation

- [Main Page](#)
- [Random page](#)
- [Recent changes](#)
- [Help](#)

tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Page information](#)

search

Go

Search

Blog > [Company News](#) Feb 18, 2020

Introducing Materialize: the Streaming Data Warehouse

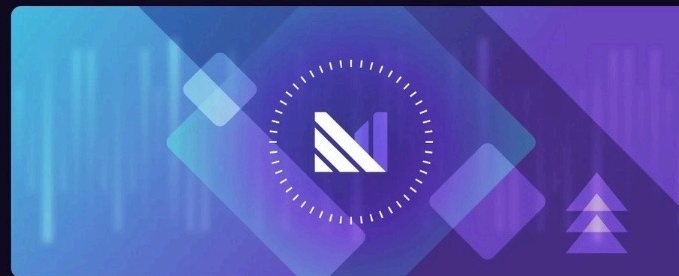
Materialize offers a streaming data warehouse for real-time analytics & interoperability with millisecond latency, revolutionizing data handling.



Frank McSherry
Chief Scientist



Arjun Narayan
CEO



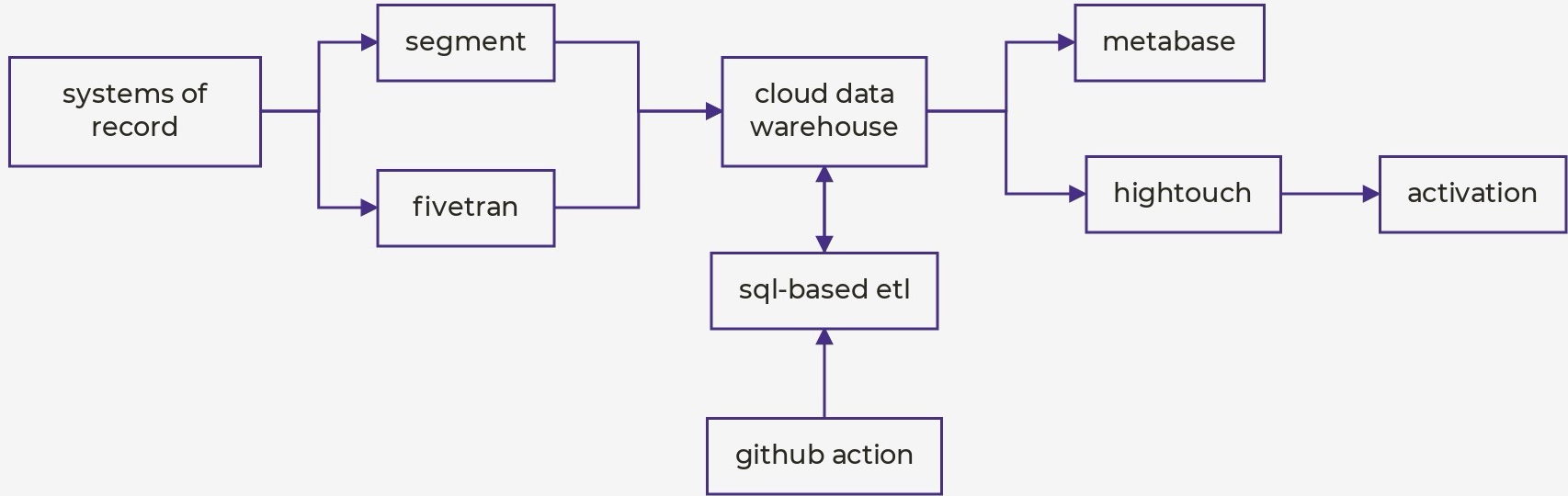
Databases, and data infrastructure generally, have made substantial progress over the years.

We now have access to cloud-native infrastructure that allows just about anyone to set up, maintain, and query databases at substantial scale. This is a serious

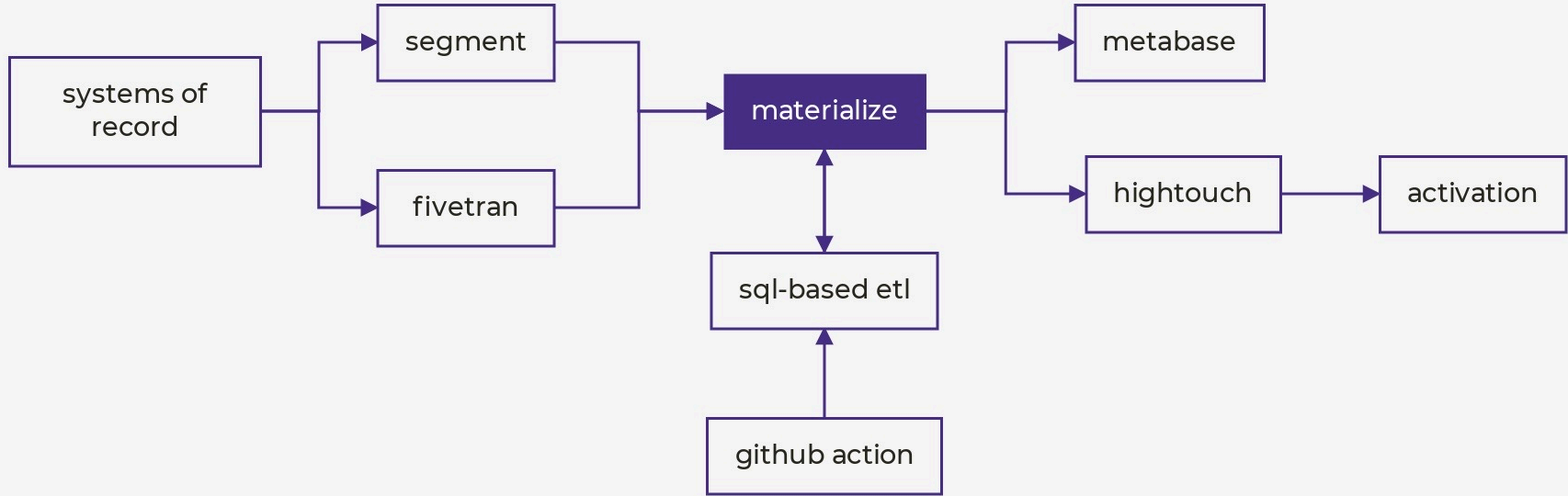
SEE MATERIALIZE IN ACTION

[SIGN UP FOR A LIVE DEMO >](#)

architecture



architecture



business impact

compared to **hot-loop**

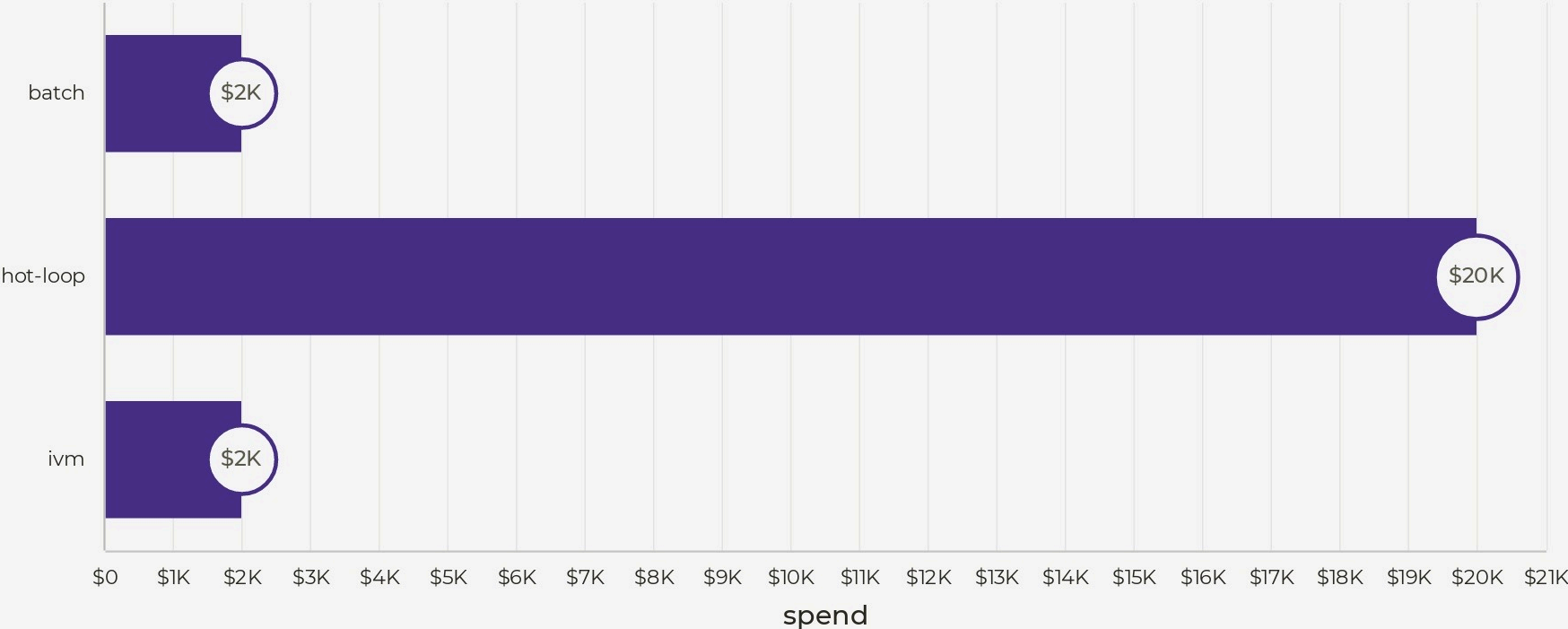
latency

2 seconds ↓ 99.8%

vs 15 minutes with hot-loop

business impact

compared to **hot-loop**



developer experience

compared to **hot-loop**



less noisy

no longer hitting flaky errors



less complex

no longer need complex orchestration

developer experience

compared to **stateful stream** processing



streaming data lake

easily ingest, process, and hold your company's streaming data



automatic backfills

new views are backfilled automatically



sub-second query engine

connect BI tools and run ad-hoc queries with consistent results

business impact

compared to **stateful stream** processing



ready out-of-the-box

no need to build your own data platform



easy to build

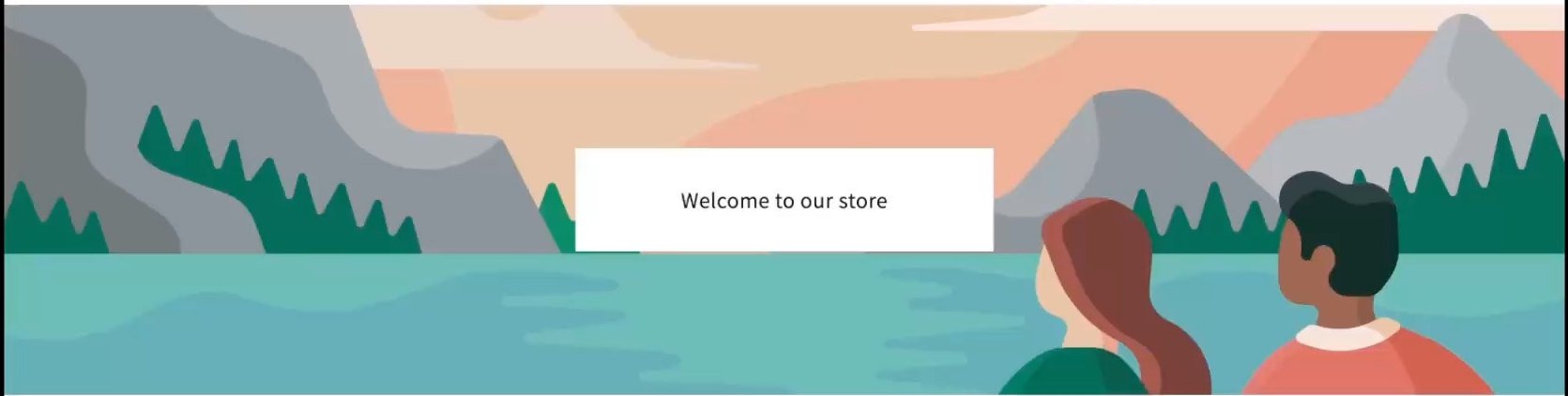
prototype, iterate on, and ship new analytics in an afternoon



easier to hire

build real-time datasets as easily as batch

let's take a look



Collections



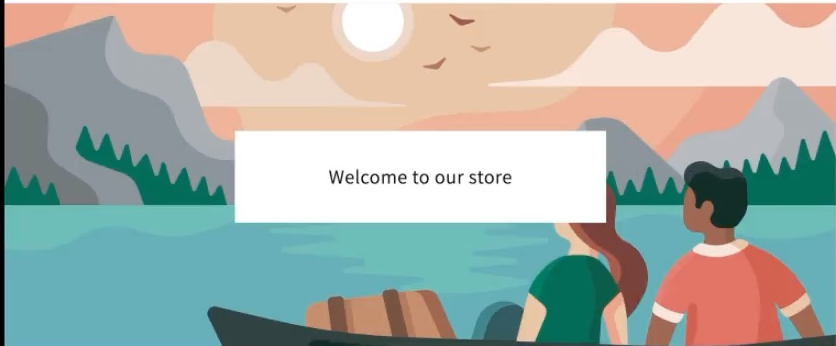
Clothing →

Accessories →

My Store x +


85dcb8-de.myshopify.com

My Store Home Catalog Contact




Welcome to our store

Collections



Clothing →



Accessories →

© 2024, My Store Powered by Shopify

My Sto x My Sto x Orders x Orders x Inbox x +

metabase.dev.materialize.com/dashboard/119-my-store...

Our analytics Search... + New

My Store

Part 1 Part 2

0
Page Views

0
Sessions





ACME

Hat

~~\$22.99 USD~~ \$17.99 USD **Sale**

Title

Grey

Quantity

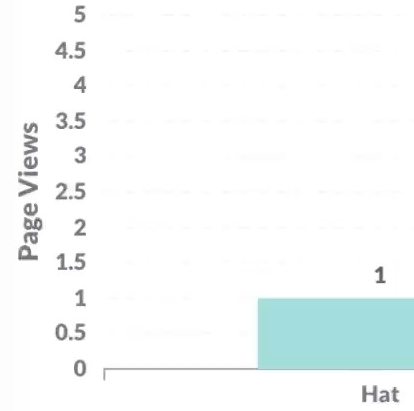
- 1 +

Add to cart

Buy it now

Share

Top Products



Top Product Types

analytics today

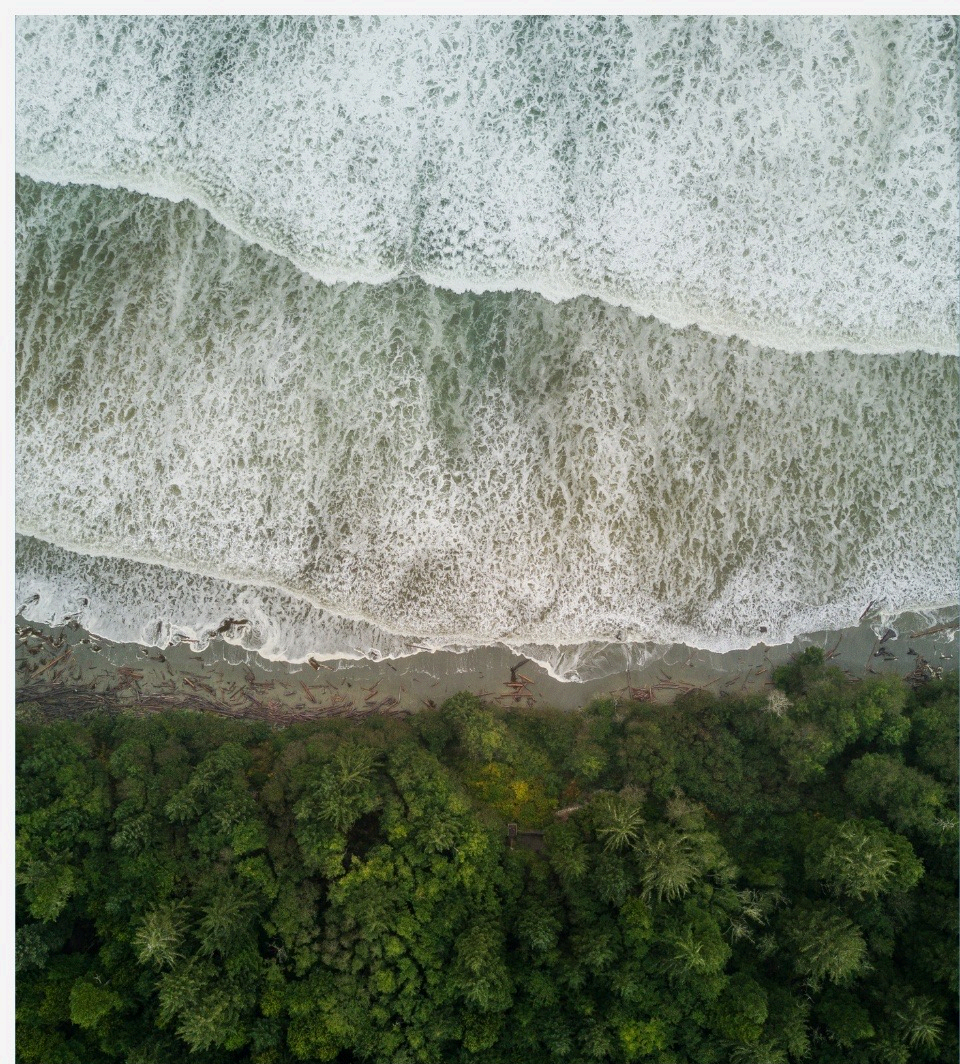
- sql-based etl is as prevalent as ever
- operational workloads benefits from real-time
- real-time has been inaccessible

what's changing

- incremental view maintenance is emerging as an alternative to full drop and re-builds
- materialize makes real-time far more accessible than stateful stream processing

opportunity

- any of us can try real-time analytics, including small teams of analytics engineers
- enables analytics teams to enable more operational workflows



thank you