Data Consumers are better Data Producers

Etsy

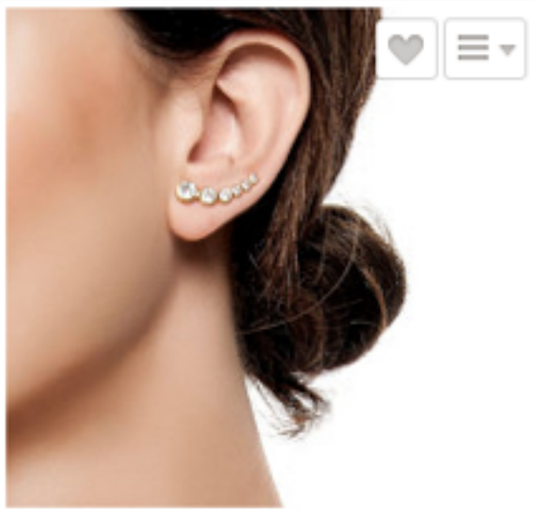
Etsy

Etsy is the world's handmade marketplace.

Our mission is to empower people to change the way the global economy works. We see a world in which very-very small businesses have much-much more sway in shaping the economy, local living economies are thriving everywhere, and people value authorship and provenance as much as price and convenience. We are bringing heart to commerce and making the world more fair, more sustainable, and more fun.

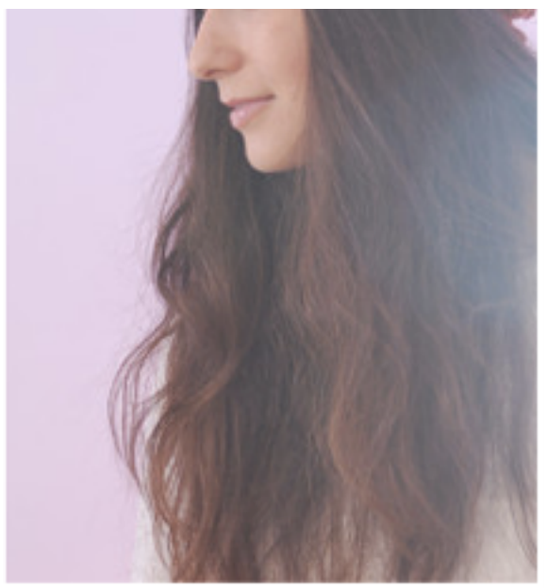
Abimbola - Felt Giraffe. Art Puppet, ...
TwoSadDonkeys $86.00 USD

Kingfisher painting PRINT of acrylic ...
LouiseDeMasi $13.39 USD



Ear Cuff ~ Delicate Two Swarovski E...
PersonalNecklace $35.00 USD

See similar items →



Chunky headband, orange wool hea...
SexyCrochetBy... $18.57 USD

She's "Consumed" Watercolor Painti...
ABitofWhimsyArt $30.00 USD

Signature Bracelet, Handwriting Brac...
capucinne $169.00 USD

Knit fingerless gloves, mittens, wome...
PetiteIdasCreati... $19.00 USD

Knitted Cable Boot Cuffs. Braids with...
VividBear $21.90 USD

Deer Cufflinks. Vintage Woodland C...
JujuTreasures $25.00 USD

Data is for Everyone

- Every person in product is a data producer
- Every person in the company CAN BE a data consumer

This is Brittany.

She’s a product manager for our Shipping products team.

She likes to use data to make decisions.

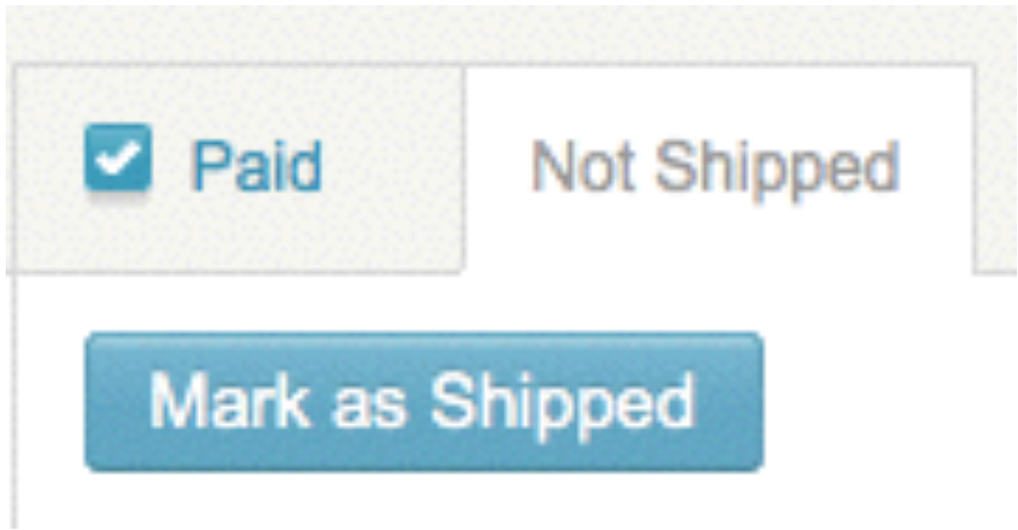
Decision for Shipping

Hypothesis: Once someone marks an item as shipped, if they do make a mistake and take it back, they will do so quickly.

If our hypothesis is true, then we can send email shipping notices.

Question about ‘Mark as Shipped’ Behavior

Orders Page:



Learned to get data

- Wrote a scalding job to get the data
- Looked at a full month of data to check for consistency

Learned to get data

- Wrote a scalding job to get the data
- Looked at a full month of data to check for consistency

And then…

- Next decision: new version of shipping labels flow
- Didn’t distinguish the versions in the event
- Stopped the push until logging was fixed

Zipcode to find City & State

Our goal with this experiment was to improve overall address quality on Etsy without materially impacting conversion, since the changes are part of the checkout flow. Our results showed an insignificant lift in conversion for those bucketed into our test group.

However, we knew that just because someone was bucketed into the test group did not mean they would actually interact with the auto-suggest features. We ran a scalding job to determine conversion rate specifically for those who interacted with the features we were testing, and found the following:

- The conversion rate for buyers not in our test group was [redacted]
- The conversion rate for buyers in our test group who manually selected a city/state suggestion was [redacted]
- The conversion rate for buyers in our test group who had automatic completion of city/state based on zip was [redacted]

In addition, we also saw that **buyers were getting through the shipping address form 3.35 seconds faster, on average**, which is about a 10% time savings.

Data in Further Decisions: Address Verification

[Launched to 100%] Suggested USPS Addresses in Checkout Review
1 message

Brittany Fri, Oct 31, 2014 at 1:35 PM
Reply-To
To: "shipping-labels-dev@etsy.com" <shipping-labels-dev@etsy.com>, ShipShape <shipshape@etsy.com>
Bcc: product-news@etsy.com

tl;dr: After running an experiment to verify US buyer addresses against USPS postal records in the checkout review page, we've now ramped it up to 100% of buyers that edit or add an address from that page.

We saw no significant changes in key checkout metrics in conjunction with overall improved address quality.

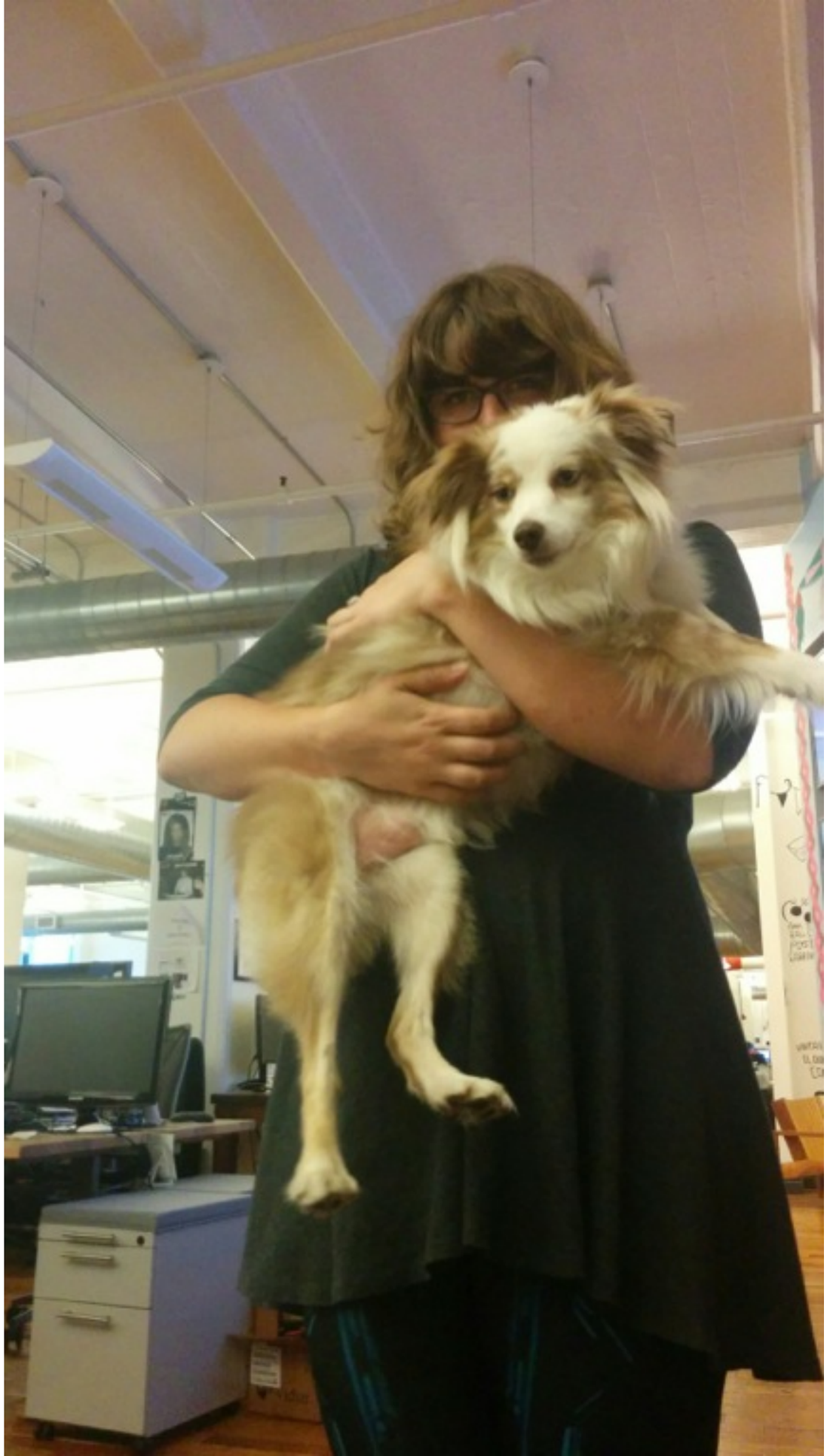
The Findings

In the experiment we saw no significant change in conversion for members bucketed into the test variant within catapult.

We also ran a scalding job to verify conversion rates for buyers who specifically interacted with the verification features, since not all members who land on checkout review will necessarily add or edit an address from that page. Through this job, we once again saw no significant change in conversion rates.

In addition to having an insignificant impact on conversion, we also measured how address quality was changing as a result and saw:

- Of buyers whose addresses were run through verification, **% were automatically cleaned**, requiring no action from the member and not interrupting the checkout flow.
- For the remaining buyers that did experience a choice between a USPS verified address and their originally entered address, **% chose the USPS suggested address.**



What are we doing that makes this possible?

- making it simple to get started and learn
- tools to make it easy to view data

Mission to teach anyone interested

- Offered tutoring in-person or online
- Specifically asked people in product management if they were interested
- Sent out notes and scheduled _1 hour_ to get from question to answer.

Getting Brittany Started

Hadoop references/notes for Thursday

Melissa Santos <msantos@etsy.com> Mar 4 ☆ Reply

to Calia, Brittany

Calia, do you have a VM? ask in #devtools if you don't - you'll need it. I am pretty sure we got Brit one last time I threatened to teach her all the things.

here is my horrible conglomeration of notes that I should really put on the wiki or something:

Running your first microscope job

In case you forgot or in case this is your first hadoop job
sit in #hadoop while you run hadoop jobs (and add "doopers" as an alert word to your irc setup)

go to the jobtracker
http://jobtracker.doop.ny4.etsy.com:50030/jobtracker.jsp (you may need to be on the prod vpn)
if there are jobs running from user oozie, ask in #hadoop if it is ok to run adhoc jobs
(try ?jtstatus)

if it's been a little while (weeks or more)
cd ~/development/BigData
git rpull
ant rebuild

if it's been at least a couple of days
cd ~/development/BigData
git rpull
ant

Define the Problem

Question: How often do people change a shipment to unshipped, and how many seconds do they go between marking as shipped and marking as unshipped?

Both of these actions are events in our data stack

order_shipped

and

order_unshipped

Get the Data with Hadoop

1. Find all the visits in a day that have ‘order_shipped’ followed by ‘order_unshipped’ for the same receipt id
2. Count them
3. Calculate the time between the two events and find the average for the day
4. Look over several days to make sure the numbers are fairly stable.

Help HER write the code

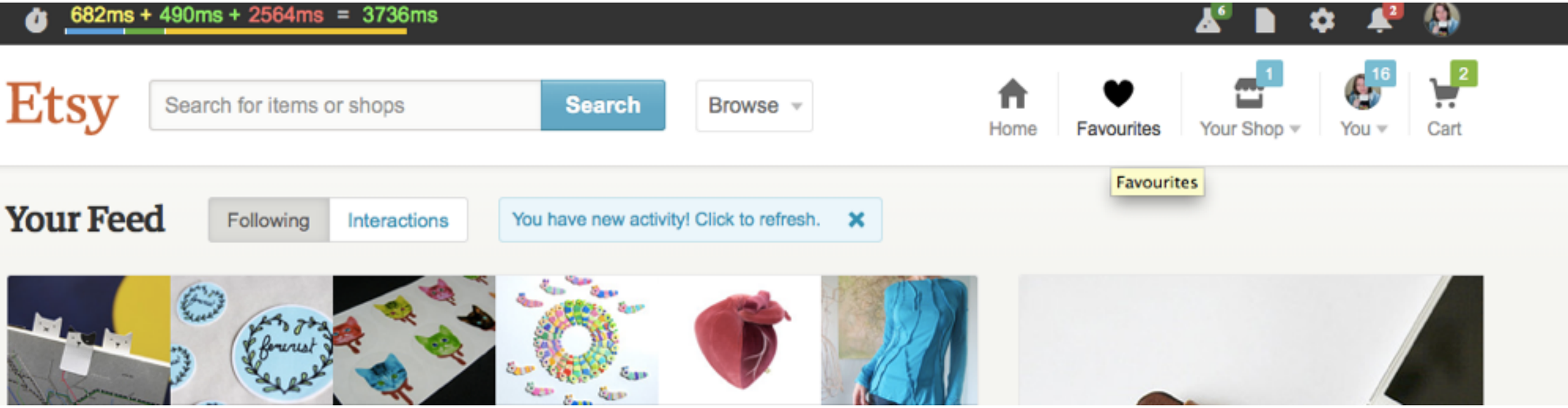
```
1   package com.etsy.scalding.jobs
2
3   import com.twitter.scalding._
4   import com.etsy.scalding._
5   import analytics.sequence.MatchPredicates._
6   import analytics.sequence._
7   import analytics._
8   import com.etsy.cascading.flow._
9
10
11  class OrderShippedUnshipped(args : Args) extends AnalyticsJob(args) {
12      val shipped_orders = VisitLog()
13        .filter('visit) { visit: Visit => visit.eventTypeExists("order_shipped") }
14
15   shipped_orders.mapTo('visit -> 'shipped_ct) { visit: Visit =>
16   visit.eventTypeCount("order_shipped")
17    }
18    .groupAll{_.sum('shipped_ct)}
19    .write(Tsv("orders_shipped_" + dateRange.start.toString("yyyy_MM_dd")))
20
21  shipped_orders.flatMapTo('visit -> 'time_between) {visit: Visit =>
22  val query = List(EventType("order_shipped"), EventType("order_unshipped") & propMatches("receipt_id"))
23  EventSequenceScanner.scan(query, visit).filter( _.size == query.size)
24  .map( matches => matches(1).epochMs - matches(0).epochMs)
25  }
26  .groupAll{_.size.sum('time_between)}
27  .write(Tsv("orders_unshipped_" + dateRange.start.toString("yyyy_MM_dd")))
28  }
29
```

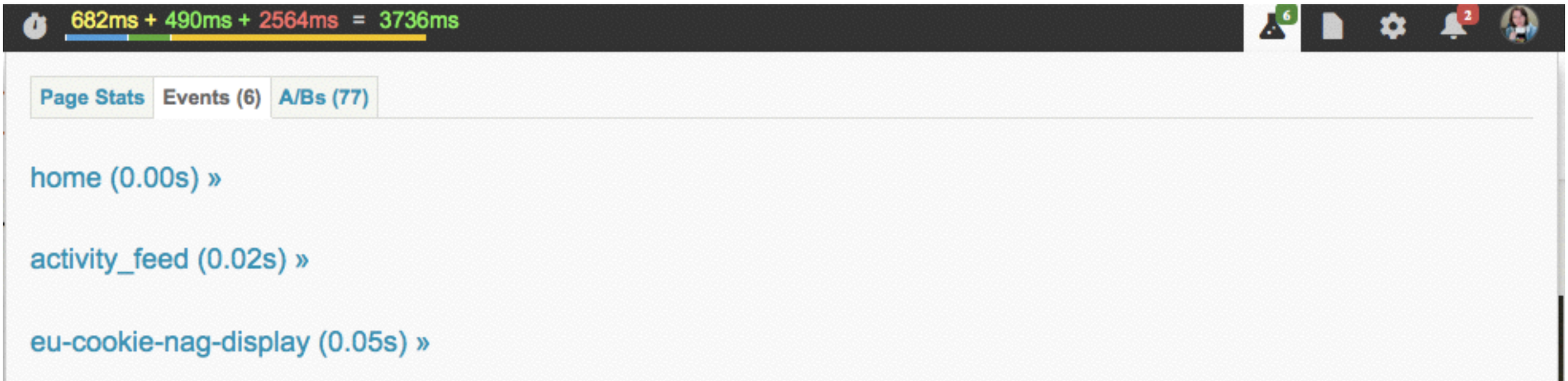
Tools

- Admin toolbar
- EventHorizon
- Scalding REPL
- Example code
- Codelab

How do I view events?

Admins can see many events on the website:





What is an Event?

order_shipped (30.50s) »

.event_logger	frontend
.event_source	web
.loc	https://www.msantos.vm.ny4dev.etsy.com/your/orders/sold?page=4
.page_guid	cdf7b814095.33c5069a63b4a7a1f660.00
.ref	https://www.msantos.vm.ny4dev.etsy.com/your/orders/sold?ref=hdr_shop_menu
.user_id	5586073
.version	0
accept-languages	en-US,en
buyer_user_id	9056254
cdn-provider	
detected_currency_code	USD
detected_language	en-US
detected_region	US
encrypted_user_id	m61MECe1j84=
etala_override	
isAdmin	false
isChromeInstantRequest	0
isEtsyApp	0

Event Horizon

Search Event Properties

Save Visible Events as Json Visit

getAllPersonalizedInfo	backend api 15:30:24
invites_buyer_giftcards_access_header	backend web 15:30:24
findAllCollections	backend api 15:30:23
activity_feed	frontend web 15:30:23
home	primary frontend web 15:30:23
.event_logger	frontend
.event_source	web
.guid	ce0a1c43952.1bee874518798bc98d7f.00
.loc	https://www.msantos.vm.ny4dev.etsy.com/
.np	2
.p	2
.page_guid	ce0a1c43952.59707f16d892ea4da8df.00
.ref	
.user_agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:33.0) Gecko/20100101 Firefox/33.0
.user_id	5586073
.version	0
accept-languages	en-US,en
cdn-provider	

Scalding REPL

```
    __   _________  __   ___  _________
   / _ )/  _/ ___/ / _ \ / _ /_  __/ _ |
  / _  |/ // (_ / / // / __ |/ / / __ |
 /____/___/\___/ /____/_/ |_/_/ /_/ |_|

Welcome to the Big Data REPL. Interested in exploring visits? Try running:
val visits = VisitExplorer(100)

If you need something more specific you can run a filter job on the cluster:
val visits = VisitFilterJob(_.newVisitor)

Or if you are interesed in Events you ca do:
val events = EventExplorer(100)
```

```
big-data > val events = EventExplorer(1000)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/msantos/development/BigData/lib/operators/target/compile/jars/slf4j-simple-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/msantos/development/BigData/lib/operators/target/compile/jars/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
HEADS UP! Using the latest logs at path: /logs.etsy.com/event_logs/2014_11_11
events: Stream[analytics.Event] = Stream(Tuple3(invites_buyer_giftcards_access_header,1415710794000,Map(php_ab_selector_names -> x;x;x;x;x;x;w;x;x;x;x;x;x;x;x;x;x;x;x;x;
x;x;x;x;x;x;x;x;e;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;x;w;x;x;x;x;x;x;x;x;e;x;e;e;w;w;x;w;w;w;w;w;w;x;x;w, .ref -> https://www.etsy.com/people/
akeller1/favorites/items-i-love?ref=favs_index_1, .page_guid -> ce03563c026.356e7dd9eaaa44eeefae.00, anchor -> null, invite_type -> 2, scheme -> http, .version -> 0, .us
 35362480, .guid -> ce03563ca88.1a3ac5007cfcd7433ea8.00, isMobileSupported -> 0, pref_language -> en-US, isChromeInstantRequest -> 0, .event_source -> web, .np -> 1, php
ate -> 11/Nov/2014:07:59:53, isMobileDevice -> 0, http_status -> 204, isMobileRequestIgno...
```

Example Code

```
package com.etsy.scalding.jobs;

import com.twitter.scalding._;
import com.etsy.scalding._;
import com.twitter.scalding.mathematics.{Histogram => HIST}
import analytics.Event;
import analytics.sequence.Visit;

class SimpleVisitAndEventExamples(args: Args)
  extends AnalyticsJob(args) {

  EventLog()
    .map('event -> 'query) {
      (e:Event) =>
        e.prop("query", "null")
    }
    .groupBy('query){
      _.size
    }
    .limit(100)
    .write(Tsv("queries"));


  VisitLog()
    .map('visit-> 'visit_length) {
      (v:Visit) =>
        v.collection.size
    }
    .groupBy('visit_length){
      _.size
    }
    .limit(100)
    .write(Tsv("visit_length"));
}
~
~
~
~
~
"SimpleVisitAndEventExamples.scala" 34L, 672C
```

CodeLab: Big Data Jobs on Etsydoop

What You'll Be Doing In This CodeLab

- Learn to run an analytics job locally and on the Hadoop cluster
- Read the job results
- Analyze the Visit Logs to see the search query terms
- Store job results into Vertica database

Getting Started

Get this CodeLab

If you haven't already cloned the `CodeLabs` repository to your VM, do this now:

```
cd ~/development # or wherever you want CodeLabs to live
git clone git://github.etsycorp.com/Engineering/CodeLabs.git
```

Set up BigData

Go to the BigData directory:


BEANIE

Data Abstractions

- Events - we've talked about these a bit
- Visits - strung together Events that share a browser id
- Searching for Events in Visits - MatchPredicate

What's a Visit?

- group the events by browser id
- all one visit until:
  - 30 minutes of inactivity
  - utm source changes (this is for marketing attribution)
- max events hit (mostly a hack for performance)


wrote MatchPredicate
to search within Visits

How do I search within a Visit?

```
1  package com.etsy.scalding.jobs
2
3  import com.twitter.scalding._
4  import com.etsy.scalding._
5  import analytics.sequence.MatchPredicates._
6  import analytics.sequence._
7  import analytics._
8  import com.etsy.cascading.flow._
9
10
11 class OrderShippedUnshipped(args : Args) extends AnalyticsJob(args) {
12     val shipped_orders = VisitLog()
13       .filter('visit) { visit: Visit => visit.eventTypeExists("order_shipped") }
14
15  shipped_orders.mapTo('visit -> 'shipped_ct) { visit: Visit =>
16  visit.eventTypeCount("order_shipped")
17   }
18   .groupAll{_.sum('shipped_ct)}
19   .write(Tsv("orders_shipped_" + dateRange.start.toString("yyyy_MM_dd")))
20
21 shipped_orders.flatMapTo('visit -> 'time_between) {visit: Visit =>
22 val query = List(EventType("order_shipped"), EventType("order_unshipped") & propMatches("receipt_id"))
23 EventSequenceScanner.scan(query, visit).filter( _.size == query.size)
24 .map( matches => matches(1).epochMs - matches(0).epochMs)
25 }
26 .groupAll{_.size.sum('time_between)}
27 .write(Tsv("orders_unshipped_" + dateRange.start.toString("yyyy_MM_dd")))
28 }
29
```

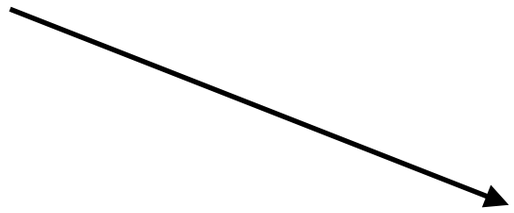
Attributing Sales

```
val purchaseQuery = List(
  EtsyHome,
  locContains("aref"),
  Payment & Purchased
)
```

```
val EtsyHome = EventType("home")
```

Attributing Sales

```
val purchaseQuery = List(
  EtsyHome,
  locContains("aref"),
  Payment & Purchased
)
```



```
def locContains(s : String): EventPredicate = EventPredicate(e => e.prop(".loc").contains(s))
```

view_listing (0.00s) »

.event_logger	frontend
.event_source	web
.loc	https://www.etsy.com/uk/listing/130274889/pineapple-mania-print-removable?ref=fp_item&atr_uid=6914690&aref=18502642101

Attributing Sales

```
val purchaseQuery = List(
  EtsyHome,
  locContains("aref"),
  Payment & Purchased
)
```

```
val Payment = EventType("backend_cart_payment")
```

```
def purchasedHelper(notReversed: Boolean): MatchPredicate =
propListContains("purchased_listing_ids", "listing_id", notReversed) |
propListContains("sold_listing_ids", "listing_id", notReversed) |
propListContains("purchased_listing_ids", "added_listing_id", notReversed) |
propListContains("sold_listing_ids", "added_listing_id", notReversed)

val Purchased = purchasedHelper(true)
```

It's not what data types —
It's how easy it is to use them

- Availability
- Visibility
- Usability

Why does this matter?

- Supports a more inclusive culture, welcoming people from all over the company
- If you can answer your own questions, you are more free to ask questions than if you rely on others
- Empowers product managers, developers, and designers, marketers, merchandisers, etc to be data-driven

Data Bonus

When you let people be part of using the data, they see what it is really for, and they care enough about it to put in better data: win-win for everyone

#dogsofetsy