Welcome
Welcome to Beam Summit

Pablo Estrada & Danielle Syse
Going since 2018...
Thanks to our speakers!
Thanks to our Sponsors

Google Cloud

GOLD

Spotify
aws
MAVENCODE

SILVER

Adobe
Thanks to our partners
KEYNOTE SESSIONS

Kerry Donny-Clark
Manager Google Cloud Dataflow
10:00 - 10:25 AM
GOOGLE’S INVESTMENT ON BEAM AND ITS INTERNAL USE

Lak Lakshmanan
Operating Executive Silver Lake
10:25 - 10:50 AM
MACHINE LEARNING DESIGN PATTERNS: BETWEEN BEAM AND A HARD PLACE

Rickard Zwahlen
Data Engineer Spotify
10:50 - 11:15 AM
TAILORING PIPELINES AT SPOTIFY

Lohit Vijayarenu
Principal Software Engineer Twitter
11:15 - 11:40 AM
THE ADOPTION, CURRENT STATE, AND FUTURE OF APACHE BEAM
Before anything..!

Please, PLEASE fill our survey:
Monday Schedule

11:40  Break

12:00  Veggie - Scaling Mic.Ops Pipelines at Credit Karma using Apache Beam and Dataflow by Debasish Das & Vishnu Venkatakrishnan

12:30 - 12:50  Houston, we’ve got a problem: 6 principles for pipelines design taken from the Apollo missions by Israel Hernandez & Paul Pahl


13:00  Lunch

14:00  Powering Real-time Data at Intuit: A Look at Golden Signals powered by Beam by Omkar Deshpande, Durja Panig, Nick Hwang & Nagaraja Tantry

14:00 - 14:45  How the sausage gets made: Dataflow under the covers by Pablo Estrada

14:00 - 14:10  State of the Go SDK 2022 by Robert Burke

14:00 - 14:30  How to break Wordle with Beam and BigQuery by Kris San Jose

15:00 - 15:50  BillVoyant: Detecting Security Dumpster Fires on the Internet by Alfredo Gomariz, Adam Najman, Tucker Leavitt & Tyler Flach

15:00 - 15:05  Migration Spark to Apache Beam/Dataflow and hexagonal architecture + DDD by Mahim Tosun

15:00 - 15:30  Introduction to performance testing in Apache Beam by Alexey Romannikov

15:30 - 15:50  From script slums to beam skyscrapers by Shailish Mangal

16:00  Break

16:15 - 16:40  Data integration on cloud made easy using Apache Beam by Parag Ghosh

16:15 - 16:40  How to benchmark your Beam pipelines for cost optimization and capacity planning by Roy Arsan

16:45 - 17:10  Colibrtris Telemetry Backbone: OpenTelemetry and Apache Beam by Alex Van Bael

16:45 - 17:10  Strategies for caching data in Dataflow using Beam SDK by Zeeshan

17:15 - 18:05  New Apex serialization and deserialization in Beam SQL by Talat Uyarer

17:15 - 18:05  Implementing Cloud Agnostic Machine Learning Workflows with Apache Beam on Kubernetes by Charles Adeloyer & Alexander Lemao

17:15 - 18:00  Cloud Spanner change streams and Apache Beam by Haikuo Liu, Nancy Xu & Le Chang

17:15 - 18:00  Cloud Spanner change streams and Apache Beam (continued) by Haikuo Liu, Nancy Xu & Le Chang

18:05  Reception 18:05 - 20:00 hrs

Austin, 2022
Where to Go Next?

- All sessions will take place on this floor across 202-204
  - Keynotes will be held in the Amphitheatre only

- Lunch will take place from 1-2 PM in the Tejas Dining Room
  - Lunch box options include roast beef on ciabatta, chicken salad croissant and falafel fritter wraps

- Session rooms will be noted outside each door as well on each calendar invite/Beam Summit page

- Restrooms located at each end of the hall with elevators to our left next to the Dining Room
About the space...

- We have rooms with whiteboards across the hall. Feel free to use the whiteboards.
  - We also have easel pads in the presentation rooms. Feel free to use in technical convos.
Networking Opportunities

Please join us for networking opportunities while you’re with us:

Reception tonight!
Join us for drinks after the event from 6:00 - 8:00 pm at the AT&T Conference Center Courtyard.

After Party Tuesday
Tuesday at 6:30pm at Lazarus Brewing Co, where beer on the house will be waiting for you! Send the directions to your phone by scanning the following QR code.
Networking Opportunities

Job Openings

Reminder to take a look at the current job openings gathered by our sponsors:
Speakers!

● Please arrive **a little early** to your room for setting up
Thank You
Google’s investments in Beam

By Dr. Kerry Donny-Clark, Google Engineering Manager for Beam
Hello!
I’m Kerry

Me at work
Me at home
My old job
Apache Beam

- Unified Model:
  - Batch and Streaming
- Many SDKs
  - Java, Python, Go, TS*
- Portability
  - Dataflow, Flink, Spark, Hazelcast, Ray*, Dask*, etc

*Experimental or in progress
Apache Beam used in Google
The Beam Team at Google

Googlers working full time on Beam

Austin, 2022
The Beam Team at Google

And more!
Beam going in new directions: Multi Language pipelines
A Rebus Riddle

Beam 2.40, Dataflow GA 7/20
Beam **Going in new directions**

14:00 - 14:25.
State of the Go SDK 2022
by Robert Burke
Room: 202.

16:15 - 16:40.
Writing a native Go streaming pipeline
by Danny McCormick & Jack McCluskey
Room: 203

Austin, 2022
Beam going in new directions: RunInference in Beam Python
Beam going in new directions: RunInference in Beam Python

PYTORCH

TensorRT

TensorFlow

scikit learn
Beam going in new directions: RunInference in Beam Python

RunInference in Beam 2.40, GA on Dataflow 7/20

https://beam.apache.org/documentation/sdks/python-machine-learning/
Beam going in new directions: Typescript SDK
Beam going in new directions: Typescript SDK

Programming, scripting, and markup languages

JavaScript completes its ninth year in a row as the most commonly used programming language. For most developers, programming is web programming. Python traded places with SQL to become the third most popular language.

<table>
<thead>
<tr>
<th>Language</th>
<th>All Respondents</th>
<th>Professional Developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript</td>
<td>68.62%</td>
<td>95.9%</td>
</tr>
<tr>
<td>HTML/CSS</td>
<td>95.9%</td>
<td>69.73%</td>
</tr>
<tr>
<td>SQL</td>
<td>41.53%</td>
<td>41.53%</td>
</tr>
<tr>
<td>Python</td>
<td>34.42%</td>
<td>36.19%</td>
</tr>
<tr>
<td>TypeScript</td>
<td>36.19%</td>
<td>34.51%</td>
</tr>
<tr>
<td>Node.js</td>
<td>34.51%</td>
<td></td>
</tr>
<tr>
<td>Java</td>
<td>36.19%</td>
<td></td>
</tr>
</tbody>
</table>
Beam going in new directions:
Typescript SDK

https://github.com/apache/beam/tree/master/sdks/typescript

… to contribute!
A better way to learn Beam: Beam Playground

https://play.beam.apache.org/
A better way to learn Beam: Beam Playground

11:00 - 11:10.
Beam Playground: discover, learn and prototype with Apache Beam
by Daria Malkova
Room: 201

Wednesday
A better way to learn Beam: Cloud notebooks

https://cloud.google.com/dataflow/docs/guides/interactive-pipeline-development
A better way to learn Beam: A Tour of Beam

Coming in late 2022!
Contributing to Beam has never been easier: Github Issues

<table>
<thead>
<tr>
<th>Inbox</th>
<th>Apache</th>
<th>Contributor permission for JIRA - Beam connecto</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Contributor permission for JIRA - org/jira/browse,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Contributor permission for Beam Jira tickets - ado,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Contributor permission for Beam Jira tickets - org,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Jira contributor permission - request &gt; contributo,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Jira contributor permission - you to Jira. Thanks,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Jira - contributor permission - you to Jira. Thanks,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>RE: Re: Contributor permission for Jira tickets - is,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Contributor permission for Jira tickets - is your jir,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Jira Contributor Permission Request - Please gran,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Contributor permission for Beam Jira tickets - as,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Contributor permission for Beam Jira tickets - My,</td>
</tr>
<tr>
<td>Apache</td>
<td></td>
<td>Contributor permission for Beam Jira tickets - as a contr,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Re: Contributor permission for Beam Jira Tickets,</td>
</tr>
<tr>
<td>Inbox</td>
<td>Apache</td>
<td>Re: Contributor permission for Beam Jira Tickets,</td>
</tr>
</tbody>
</table>
Contributing to Beam has never been easier: PR-bot

Turn pr-bot on for whole repo #21421

damccorm opened this issue on 4 Jun · 0 comments · Fixed by #22257

damccorm commented on 4 Jun

Right now, the pr-bot is only enabled for prs in the Go area - once its proven to be working, we should turn it on for the rest of the repo.

Imported from Jira BEAM-14045. Original Jira may contain additional context.
Reported by: damccorm.
Subtask of issue #21417
Conclusion

Beam is growing
- Multi Language
- Beam Go SDK
- RunInference in Python
- TypeScript SDK

Learn Beam
- Beam Playground
- Beam Notebooks
- A Tour of Beam

Contribute to Beam
- Github Issues
- PR-bot
Machine Learning Design Patterns: Between Beam and a Hard Place

Lak Lakshmanan

@lak_luster
Formalized best practices to solve common problems

- Preface
- The Need for ML Design Patterns
- Data representation design patterns
  - #1 Hashed Feature
  - #2 Embedding
  - #3 Feature Cross
  - #4 Multimodal Input
- Problem representation design patterns
  - #5 Reframing
  - #6 Multilabel
  - #7 Ensemble
  - #8 Cascade
  - #9 Neural Class
  - #10 Rebalancing
- Patterns that modify model training
  - #11 Useful overfitting
  - #12 Checkpoints
  - #13 Transfer Learning
  - #14 Distribution Strategy
  - #15 Hyperparameter Tuning
- Resilience patterns
  - #16 Stateless Serving Function
  - #17 Batch Serving
  - #18 Continuous Model Evaluation
  - #19 Two Phase Predictions
  - #20 Keyed Predictions
- Reproducibility patterns
  - #21 Transform
  - #22 Repeatable Sampling
  - #23 Bridged Schema
  - #24 Windowed Inference
  - #25 Workflow Pipeline
  - #26 Feature Store
  - #27 Model Versioning
- Responsible AI
  - #28 Heuristic benchmark
  - #29 Explainable Predictions
  - #30 Fairness Lens
- Summary

Vallappa Lakshmanan, Sara Robinson & Michael Munn

Austin, 2022
ML flavors of the same problems that arise in all software

### Maintainability
How do you represent categorical data when the vocabulary increases over time?

### Reusability
How do you avoid having to relearn relationships between categorical variables used in related ML problems?
Beam is widely used in a few design patterns
CREATE OR REPLACE MODEL ch09edu.bicycle_model
OPTIONS(input_label_cols=['duration'],
    model_type='linear_reg')
AS
SELECT
    duration,
    start_station_name,
    CAST(EXTRACT(dayofweek from start_date) AS STRING) as dayofweek,
    CAST(EXTRACT(hour from start_date) AS STRING) as hourofday
FROM
    `bigquery-public-data.london_bicycles.cycle_hire`
The Transform pattern: the model graph should include transformations

```
inputs = keras.Input(shape=input_shape)
x = preprocessing_layer(inputs)
outputs = rest_of_the_model(x)
model = keras.Model(inputs, outputs)
```
tf.transform provides reuse and efficiency

def main(output_dir):
    with tft_beam.Context(temp_dir=tempfile.mkdtemp()):
        transformed_dataset, transform_fn =(
            (raw_data, raw_data_metadata) | tft_beam.AnalyzeAndTransformDataset(
                preprocessing_fn))
        transformed_data, transformed_metadata = transformed_dataset

    # Save the transform_fn to the output_dir
    _ = (transform_fn
         | 'WriteTransformFn' >> tft_beam.WriteTransformFn(output_dir)
    )
    return transformed_data, transformed_metadata

class ExportModel(tf.Module):
    def __init__(self, trained_model, input_transform):
        self.trained_model = trained_model
        self.input_transform = input_transform

    @tf.function
    def __call__(self, inputs, training=None):
        x = self.input_transform(inputs)
        return self.trained_model(x)
Other patterns that Beam supports well

---

```
# Time window into 2 hour windows, triggered every minute
WINDOW_INTERVAL = 2 * 60 * 60. # 2 hours, in seconds
PANE_INTERVAL = 10*60 # 10 minutes, in seconds
windowed = (data | 'window' >> beam.WindowInto(
    beam.window.SlidingWindows(WINDOW_INTERVAL, PANE_INTERVAL),
    accumulation_mode=beam.trigger.AccumulationMode.DISCARDING))
model_state = (windowed | 'model' >> beam.transforms.CombineGlobally(ModelFn()).without_defaults())
anomalies = (windowed | 'latest_slice' >> beam.FlatMap(is_latest_slice) | 'find_anomaly' >> beam.Map(is_anomaly, beam.pvalue.AsSingleton(model_state)))
```
Other patterns that Beam supports well

- **Batch Serving**
- **Windowed Inference**

### Time window into 2 hour windows, triggered every minute

```
WINDOW_INTERVAL = 2 * 60 * 60. # 2 hours, in seconds
PANE_INTERVAL = 10*60 # 10 minutes, in seconds
windowed = (data
    | 'window' >> beam.WindowInto(
        beam.window.SlidingWindows(WINDOW_INTERVAL, PANE_INTERVAL),
        accumulation_mode=beam.trigger.AccumulationMode.DISCARDING))
model_state = (windowed
    | 'model' >> beam.transforms.CombineGlobally(ModelFn()),without_defaults())
anomalies = (windowed
    | 'latest_slice' >> beam.FlatMap(is_latest_slice)
    | 'find_anomaly' >> beam.Map(is_anomaly, beam.pvalue.AsSingleton(model_state)))
```
There are other patterns where Beam could be used, but isn’t

Cascade
Transfer Learning
Continuous Evaluation
Two Phase Predictions
Multimodal Input
Workflow Pipeline

What’s common to these?

Cascade
Transfer Learning
Continuous Evaluation
Two Phase Predictions
Multimodal Input
Workflow Pipeline

Why?

- **Training, evaluation:** One-off, rare task
- **Online serving:** On-demand to millions
- **Artifact Management among multiple ML models:** Orchestration
What if Beam could:

- scale from zero to millions of QPS
- consume/produce HTTP, cloud events
- be GPU-accelerated
- be run on-demand (start instantaneously)?
Imagine ...

A Beam Runner that runs on Cloud Run

Portable way to run Java/Go/Python across serverless container options on AWS, GCP, Azure

Scales to zero, suitable for rare ETL, scales on-demand code

Portable ML code across training, inference, evaluation
Thank you!
Tailoring pipelines at Spotify

By Rickard Zwahlen
Rickard Zwahlen
Data engineer @ Spotify
@rickardzwahlen
(Mel: ABBA - Super Trouper)
Super Deduper runs as fast as lightning
Handles massive skew
Gets events to you
But only one of each, not two
Smörgåsbord of data
How it started
How it’s going
Why Scala?

- Productivity + performance
- Functional & type safe
- Large software ecosystem for data
The love triangle
val sc = ScioContext()

sc
  .textFile("shakespeare.txt")
  .flatMap { _ .split("[^a-zA-Z\']+")
  .filter(_ nonEmpty) }
  .countByValue
  .saveAsTextFile("wordcount.txt")

sc.run()
val sc = ScioContext()
sc
  .avroFile[Artist](args("artists"))
  .keyBy(_.getArtistId)
  .hashJoin(musicLabels)
  .map { case (artistId, (artist, label)) =>
    (artistId,
     businessLogic(artist, label)
    )
  }
sc.run()
Bread & butter pipelines

A large majority of pipelines are written in Scio
Cake mix pipelines

Just add water
Data profiling

schedule: hourly
docker_image: grc.io/data-profiling/1.2.3@sha256:foo
docker_args:
  - wrap-luigi
  - --module
  - luigi_tasks
  - ProfileRunner
  - --input-dataset
  - Impressions.gcs
  - --partitioning
  - hours
  - --project
  - my-cloud-project
Data profiling (pt 2)

<table>
<thead>
<tr>
<th>Top Values (Show Percentages)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8.7.4</td>
<td>2.6G</td>
</tr>
<tr>
<td>8.7.4</td>
<td>1.3G</td>
</tr>
<tr>
<td>11.88</td>
<td>163M</td>
</tr>
<tr>
<td>8.7.4</td>
<td>120M</td>
</tr>
<tr>
<td>8.7.4</td>
<td>98M</td>
</tr>
<tr>
<td>8.6.4</td>
<td>61M</td>
</tr>
<tr>
<td>11.88</td>
<td>51M</td>
</tr>
<tr>
<td>8.7.4</td>
<td>43M</td>
</tr>
<tr>
<td>8.6.4</td>
<td>35M</td>
</tr>
<tr>
<td>8.7.4</td>
<td>34M</td>
</tr>
<tr>
<td>2022</td>
<td>27M</td>
</tr>
<tr>
<td>web</td>
<td>25M</td>
</tr>
<tr>
<td>8.7.3</td>
<td>26M</td>
</tr>
</tbody>
</table>
Data profiling (pt 3)

**Historical Profiling**
Analyze metrics for a field over time by selecting a data range below.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>element_detail_hash</code></td>
<td>2022-07-06 20:00 UTC</td>
<td>2022-07-13 20:00 UTC</td>
</tr>
</tbody>
</table>

**Calculations**
- Approx. Top K
- Approx. Distinct
- Empty String
- Non-Empty String
- Max Length
- Min Length

**Approx. Distinct**

![Graph showing data profiling metrics over time]
Anomaly detection

Partition Selected
2022-06-15T10:00:00Z

Predicted Range
-47.201 - 149.682

Actual
246

Out of range values are shown relative to latest run.
The difficult stuff

Scale, complexity, edge cases
Case by case

or

Scio
Spotify Wrapped

You listened to 1,528 total artists this year, but things got serious with one...

Your top artist was Polo G
You were in the top 1% of their listeners this year.

You spent a total of 3,456 minutes with them and just couldn't get enough of RAPSTAR.

My Top Artist
Polo G

Minutes streamed
3,456
Most played song
RAPSTAR
Scio + Sort Merge Bucket

PCollection\(<T>\) → Assign Bucket ID \(\text{hash(key) \% numBuckets}\) → Sort Bucket Groups → GroupByKey + Sort

bucket-0-of-n.avro → ...

bucket-m-of-n.avro

Austin, 2022
We use Beam at the highest level of abstraction that fits the use case

- Beam SDK
- Scio Scala API
- Plug-and-play images
Thanks!

Check out the Scio workshop on Wednesday
Beam @Twitter Evaluation, Adoption, Migration and future.
Beam: Adoption, Current state and future @Twitter

Lohit VijayaRenu
@lohitvijayarenu
Data Processing
@Twitter

Twitter Timelines
Recommendations
Analytics
Ad products
Trends, Search, Explore
Many more... or
Everything
Technology

Streaming: Apache Storm, Apache Heron
Batch: Scalding, Spark, Apache Tez, Apache Hadoop
SQL: Presto, HIVE
Cloud: Google Cloud Platform (BQ, DF, GCS...)

Open Source

Twitter initiated projects: TwitterOSS
Contributions & Adoption: Apache Software, Linux Foundation, Python Foundation, Scala Center...
Every day challenges

- Data pipelines: 50k+
- Data Volume processed: 200+PB
- Data across storage systems: 1+EB
- Events processed: 7+Tri
Continuous improvement

Data Processing requirements

- Stream vs Batch (Unification)
- Modern execution framework
- Newer technologies (Spark, Tez, Flink, Beam)
- Newer API (Scio, Beam, Spark, SQL, Streaming SQL)
- Easier adoption (Metrics, configuration, debugging tools, deployment and support)
Data Processing Evaluation

Dimensions

API
Unified and modern API, API Support, Language Support, Conversion tools from existing to new API.

Platform Offering
Platform availability, support and stability. Evaluation of different runners.

Platform Integration
Integrate with other tools, SQL, tabular, Data formats, Industry adoption.

Twitter Integration

Model use cases
Production vs Ad hoc stream and batch processing, ML workloads, Analytics. Right tool for customer use case.
Why Beam is attractive

- Unified API, Modern Execution frameworks
- Flexibility of different runners and how it affects company strategy
- Attractive for multi cloud support
- Different programming languages
- Strong open source community and support
Streaming Adoption
Ad Engagement Analytic Platform

- Ad Engagement pipelines built on **lambda architecture**
- Stream processing **millions of events per second**
- Migrate **Apache Heron pipelines to Apache Beam**
- Utilize **same API** for both batch and streaming components
- Increased **developer velocity** and cleaner abstraction
Batch Adoption
Experimentation Pipelines

- Modernizing Twitter Experimentation Pipelines
- Scalding based hard to maintain, debug and scale
- Easier programming paradigm
- Increase developer productivity
- Pipeline runtime from days to hours
Challenges

- **Language**: SCIO, Java, Python
- **Migration**: Variety and Scale
- **Custom libraries**: Use case specific logic
- **Long term support**: Compare against other APIs
- **Twitter Integration**: Metadata, deployment, monitoring…
Current Use cases

- Machine Learning & Feature Engineering pipelines
- Curated data and metrics calculation
- Data Replication and Ingestion framework
- Real Time Analytics and Monitoring
- Ad Analytics platform
- Twitter Health monitoring pipelines
- Product learning platform
Future for Beam @Twitter

- Migration of all pipelines to Apache Beam
- Unifying streaming/batch and increase streaming use cases
- Integration tooling for data delivery, metadata and monitoring
- Self serve deployment and management
- Excited about community engagement and contributions
More at Beam Summit

- Talk to us about opportunities
- Tuesday, 19 14:00
  *Log ingestion and data replication at Twitter* by Praveen Killamsetti & Zhenzhao Wang
- Tuesday, 19 17:15
  *Apache Beam backend for open source Scalding* by Navin Viswanath
Thank you!