Architecting for change: LinkedIn's new data ecosystem

Shirshanka Das, Principal Staff Engineer, LinkedIn
Yael Garten, Director of Data Science, LinkedIn

Sept 28, 2016

@shirshanka, @yaelgarten
Design for change. Expect it. Embrace it.
The Product Change:
Launch a completely rewritten LinkedIn mobile app
What does this impact?

Data driven product
Tracking data records user activity

InvitationClickEvent()
Tracking data records user activity

Scale fact:
~ 1000 tracking event types,
~ Double-digit TB per day,
hundreds of metrics & data products
Tracking Data Lifecycle

<table>
<thead>
<tr>
<th>Produce</th>
<th>Transport</th>
<th>Consume</th>
</tr>
</thead>
</table>

- User engagement tracking data
- Metrics scripts
- Production code
- Member facing data products
- Business facing decision making
Tracking Data Lifecycle & Teams

<table>
<thead>
<tr>
<th>Produce</th>
<th>Transport</th>
<th>Consume</th>
</tr>
</thead>
</table>

Product or App teams: PMs, Developers, TestEng

Infra teams: Hadoop, Kafka, DWH, ...

Data teams: Analytics, Relevance Engineers, ...

Member facing data products

Production code

Metric scripts

Business facing decision making

user engagement tracking data
How do we calculate a metric: ProfileViews

PageViewEvent

Record 1:
{"header": {
  "memberId": 12345,
  "time": 1454745292951,
  "appName": {
    "string": "LinkedIn",
    "pageKey": "profile_page"
  },
},
"trackingInfo": {
  ["vieweeID": "23456"],
  ...
}
}

Metric:
ProfileViews = sum(PageViewEvent)
where pageKey = profile_page or new_profile_page
Evolution as we mature and grow...

PageViewEvent
Record 1:
{
    "header" : {
        "memberId" : 12345,
        "time" : 1454745292951,
        "appName" : {
            "string" : "LinkedIn",
            "pageKey" : "profile_page"
        },
        "pageKey" : "profile_page"
    },
    "trackingInfo" : {
        "vieweeID" : "23456",
        ...}
}

CASE
WHEN trackingInfo["profileIds"] ...
WHEN trackingInfo["profileid"] ...
WHEN trackingInfo["profileId"] ...
WHEN trackingInfo["url\$profileIds"] ...
WHEN trackingInfo["11"] LIKE '%profileIds=%' THEN SUBSTRING(trackingInfo["11"],9,60)
WHEN trackingInfo["12"] LIKE '%priceIds=%' THEN SUBSTRING(trackingInfo["12"],9,60)
ELSE NULL
END AS profile_id

Metric: ProfileViews = sum(PageViewEvent
where
pagekey = profile_page and
memberID != trackinginfo[vieweeID] )
Eventually... unmaintainable

get_tracking_codes = foreach get_domain_rolled_up generate
    ..entry_domain_rollup,
        ( (tracking_code matches 'eml-ced.*' or tracking_code matches 'eml-b2\_content\_ecosystem\_digest.*'
            or referer is not null and (referer matches '.*touch\.linkedin\.com.*trk=eml-ced.*'
                or referer matches '.*touch\.linkedin\.com.*trk=eml-b2\_content\_ecosystem\_digest.*')) ? 'Email - CED':
            (tracking_code matches 'eml-.*' or (referer is not null and referer matches '.*touch\.linkedin\.com.*trk=eml-.*')
                or entry_domain_rollup == 'Email' ? 'Email - Other':
                (tracking_code == 'hp-feed-article-title-hpm' and entry_domain_rollup == 'Linkedin' ? 'Homepage Pulse Module':
                    (tracking_code matches 'hp-feed-.*' and entry_domain_rollup == 'Linkedin') or (std_user_interface matches '(phone app|tablet app|phone browser|tablet browser)' and tracking_code == 'v-feed')
                or (tracking_code == 'Organic Traffic' and entry_domain_rollup == 'Linkedin' and (referer == 'https://www.linkedin.com/nhome' or referer == 'http://www.linkedin.com/nhome')) ? 'Feed':
                    (tracking_code matches 'hb_ntf_MEGAPHONE_.*' and entry_domain_rollup == 'Linkedin' ? 'Desktop Notifications':
                        (tracking_code == 'm_sim2_native_reader_swipe_right' ? 'Push Notification':
                            (tracking_code == 'pulse_dexter_stream_scroll' and entry_domain_rollup == 'Linkedin' ? 'Pulse - Infinite Scroll on Dexter': --infinite scroll on dexter
                                ((tracking_code == 'pulse_dexter_nav_click' or tracking_code == 'pulse-det-nav_art') and entry_domain_rollup == 'Linkedin' ? 'Pulse - Left Rail Click on Dexter': --left rail click on dexter
                                    (tracking_code == 'Organic Traffic' and referer is not null and referer matches '.*linkedin\.com\/pulse\/article.*' ? 'Publishing Platform':
                                        'None Found Yet')))))))))) as entry_point;

Homepage team

Email team

Push Notification team

Long form post team
We wanted to move to better data models

PageViewEvent
Record 1:
{
    "header" : {
        "memberId" : 12345,
        "time" : 1454745292951,
        "appName" : {
            "string" : "LinkedIn",
            "pageKey" : "profile_page"
        },
    },
    "trackingInfo" : {
        "vieweeID" : "23456",
        ...
    }
}

LI_ProfileViewEvent
Record 1:
{
    "header" : {
        "memberId" : 12345,
        "time" : 4745292951145,
        "appName" : {
            "string" : "LinkedIn",
            "pageKey" : "profile_page"
        },
    },
    "entityView" : {
        "viewType" : "profile-view",
        "viewerId" : "12345",
        "vieweeId" : "23456",
    }
}
Two options:

1. Keep the old tracking:
   a. Cost: producers (try to) replicate it (write bad old code from scratch),
   b. Save: consumers avoid migrating.

2. Evolve.
   a. Cost: time on data modeling, and on consumer migration,
   b. Save: pays down data modeling tech debt
How much work would it be?

Two options:

1. Keep the old tracking:
   a. Cost: producers (try to) replicate it (write bad old code from scratch),
   b. Save: consumers avoid migrating.

2. Evolve:
   a. Cost: time on data modeling and on consumer migration,
   b. Save: pays down data modeling tech debt

5000 days

Estimated cost for producers to attempt to replicate old tracking

2000 days

Estimated cost to update consumers to new tracking with clean, committee-approved data models

#AnalyticsHappiness
The Task and Opportunity

Must do: So we will do the data modeling, and rewrite all the metrics to account for the changes happening upstream… but…

Extra credit points: How do we make sure that the cost is not this high the next time?

How do we handle evolution in a principled way?
Product Change

Technology

Culture & Process

Learnings
Metrics ecosystem at LinkedIn: 3 yrs ago

Operational Challenges
Diminished Trust due to multiple sources of truth
Data Stages

Create  Ingest  Process  Serve  Visualize
Create

Ingest

Process

Serve

Visualize

Tracking

Kafka

Espresso

...
Components

SDKs in different frameworks (server, client)

Tracking front-end

Monitoring Tools
Data Stages

Create  Ingest  Process  Serve  Visualize
Unified Ingestion with Gobblin

Ingest

Hundreds of TB / day
Thousands of datasets
80+% of data ingest
In production @ LinkedIn, Intel, Swisscom, NerdWallet, PayPal
Create
Ingest
Process
Serve
Visualize

Hadoop
Processing engines @ LinkedIn
Pinot

Distributed Multi-dimensional OLAP
Columnar + indexes
No joins
Latency: low ms to sub-second
Serve

In production @ LinkedIn, Uber

Site-facing Apps

Reporting dashboards

Monitoring
Unified Metrics Platform (UMP)

Create  Ingest  Process  Serve  Visualize

Hadoop  Pinot  Raptor
Unified Metrics Platform

Incremental
Aggregate
Backfill
Auto-join
Create
Kafka
Espresso
...
Ingest
Process
Hadoop

Serve
Pinot

Visualize
Raptor

Tracking
Unified Metrics Platform (UMP)
How do we handle old and new?

Producers

PageViewEvent

ProfileViewEvent

Consumers

Relevance

Analytics
The Big Challenge

Our scripts were doing ....
load “/data/tracking/PageViewEvent” using AvroStorage()
We need “microservices” for Data
The Database community solved this decades ago...

Views!
We had been working on something that could help...

A Data Access Layer for Linkedin

Abstract away underlying physical details to allow users to focus solely on the logical concerns

Logical Tables + Views

Logical FileSystem
Solving
With
Views
Views ecosystem

Producers

LinkedInProfileView

LinkedInApp

iOs

Android

JSAProfileView

Job Seeker App (JSA)

JSAProfileView

UnifiedProfileView

Consumers

Relevance (Models)

Analytics

(Reports)
Dali: Implementation Details in Context

DALI CLI

Data Catalog + Discovery (DALI)

View Defs + UDFs (Artifactory, Git)

Dataflow APIs (MR, Spark, Scalding)

Query Layers (Hive, Pig, Spark)

DALI Datasets (Tables + Views)

Processing Engine (MapReduce, Spark)

DaliFileSystem Client

Data Source (HDFS)

Data Sink (HDFS)
One small step for a script

From

load '/data/tracking/PageViewEvent'
using AvroStorage();

To

load 'tracking.UnifiedProfileView'
using DaliStorage();
A Few Hard Problems

Versioning
  Views and UDFs
  Mapping to Hive metastore entities
Development lifecycle
  Git as source of truth
  Gradle for build
  LinkedIn tooling integration for deployment
Early experiences with Dali views

How we executed
   Lots of work to get the infra ready
   Closed beta model
   Tons of training and education (hand holding) for all
   Governance body

Feedback from analysts is overwhelmingly positive:
   + Much simpler to share and standardize data cleansing code with peers
   + Provides effective insulation to scripts from upstream changes
   - Harder to debug where problems are due to additional layer
State of the world today

~100 producer views
~200 consumer views
~30% of UMP metrics use Dali data sources
~80 unique tracking event data sources

ProfileViews
MessagesSent
Searches
InvitationsSent
ArticlesRead
JobApplications
...

Diagram of interconnected nodes.
What’s next for Dali?

Real-time Views on streaming data

Selective materialization

Hive is an implementation detail, not a long term bet

Open source

Data Quality Framework
Product Change → Technology → Culture & Process → Learnings
Infrastructure enables, but culture really preserves
For a great data ecosystem that can handle change:

1. Standardize core data entities

2. Create clear maintainable contracts between data producers & consumers

3. Ensure dialogue between data producers & consumers
1. Standardize core data entities

- Event types and names: Page, Action, Impression
- Framework level client side tracking: views, clicks, flows
- For all else (custom) - guide when to create a new Event or Dali view
2. Create clear maintainable contracts

1. Tracking specification with monitoring: clear, visual, consistent contract

2. Dali dataset specification with data quality rules

Need tooling to support culture shift
3. Ensure dialogue between Producers & Consumers

- Awareness: Train about end-to-end data pipeline, data modeling
- Instill communication & collaborative ownership process between all: a step-by-step playbook for who & how to develop and own tracking
  PM → Analyst → Engineer → All3 → TestEng → Analyst
Product Change → Technology → Culture & Process → Learnings
Our Learnings

Culture and Process
- Spend time to identify what needs culture & process, and what needs tools & tech
- Big changes can mean big opportunities
- Very hard to massively change things like data culture or data tech debt; never a good time to invest in "invisible" behind-the-scenes change
  → Make it non-invisible -- try to clarify or size out the cost of NOT doing it
  → needed strong leaders, and a village

Tech
- Must build tooling to support that culture change otherwise culture will revert
- Work hard to make any new layer as frictionless as possible
- Virtual views on Hadoop data can work at scale! (Dali views)
For a great data ecosystem that can handle change:

1. Standardize core data entities

2. Create clear maintainable contracts between data producers & consumers

3. Ensure dialogue between data producers & consumers

Design for change. Expect it. Embrace it.
Did we succeed? We just handled another huge change!

#AnalyticsHappiness
Thank you.

@shirshanka, @yaelgarten