Integrating Heterogeneous Data

Presented by
Joe Caserta

#DAMADay

@joe_caserta
Caserta Timeline

- **1986**: Began consulting database programming and data modeling
- **1996**: Data Analysis, Data Warehousing and Business Intelligence since 1996
- **2001**: Founded Caserta Concepts in NYC
- **2004**: Launched Big Data practice
- **2009**: Launched Data Science, Data Interaction and Cloud practices
- **2012**: Laser focus on extending Data Analytics with Big Data solutions
- **2013**: Dedicated to Data Governance Techniques on Big Data (Innovation)
- **2014**: Top 20 Most Powerful Big Data consulting firms
- **2016**: Awarded Fastest Growing Big Data Companies 2016
- **2016**: Awarded Top 20 Big Data Companies 2016
About Caserta Concepts

- **Consulting** Data Innovation and Modern Data Engineering
  - Award-winning company
  - Internationally recognized work force
  - Strategy, Architecture, Implementation, Governance

- **Innovation** Partner
  - Strategic Consulting
  - Advanced Architecture
  - Build & Deploy

- **Leader** in Enterprise Data Solutions
  - Big Data Analytics
  - Data Warehousing
  - Business Intelligence
  - Data Science
  - Cloud Computing
  - Data Governance
Caserta Client Portfolio

Finance, Healthcare & Insurance

Retail/eCommerce & Manufacturing

Digital Media/AdTech

Education & Services

#DAMADay @joe_caserta
Awards & Recognition

- Inc. 5000
  - Gartner Market Guide to Advanced Analytics Service Providers
- Gartner
- Silicon Review
  - Top 10 Fastest Growing Big Data Companies 2016
- Silicon Review
  - Top 20 Big Data Companies 2015
- CIO Story
  - Top 20 Most Powerful Big Data Solution Providers 2014
- Healthcare Tech 
  - Top 20 Healthcare Solutions Providers 2015
- CIO Data Analytics Consulting
  - Top 20 Data Analytics Consulting Companies 2013
<table>
<thead>
<tr>
<th>Amazon Web Services</th>
<th>Hortonworks</th>
<th>Cloudera</th>
<th>IBM</th>
<th>MapR</th>
<th>Collibra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucidworks</td>
<td>Tableau</td>
<td>Basho</td>
<td>TIBCO Jaspersoft</td>
<td>Talend</td>
<td>Snowflake</td>
</tr>
<tr>
<td>DataStax</td>
<td>Neo4j</td>
<td>Actian</td>
<td>Syncsort</td>
<td>Datameer</td>
<td>Alation</td>
</tr>
<tr>
<td>Pitney Bowes</td>
<td>Cisco</td>
<td>DataguiSE</td>
<td>Revolution Analytics</td>
<td>Databricks</td>
<td>Microsoft Azure</td>
</tr>
</tbody>
</table>
Aligning Heterogeneous Data Sources

Digital Touchpoints

Awareness

Consideration

Purchase

Service

Loyalty Expansion

Physical Touchpoints

#DAMADay

@joe_caserta
Why do we Care?

Attribution Type

- Single Touch
  - Assign the credit to the first or last exposure
  - Last touch only
  - Ignores bulk of customer journey
  - Undervalues other interactions and influencers
  - Ad-Click: 100%

- Rules-Based
  - Assign the credit to each interaction based on business rules
  - Subjective
  - Assigns arbitrary values to each interaction
  - Undervalues other interactions and influencers
  - Mailing → E-mail → Ad-Click: 33% → 33% → 33%

- Statistically Driven
  - Assign the credit to interactions based on data-driven model
  - Looks at full behavior patterns
  - Considers all touch points
  - Can apply different models for best results
  - Use data to find correlations between touch points (winning combinations)
  - Mailing → E-mail → Ad-Click: 27% → 49% → 24%
**Onboarding New Data**

**Business:** “I need to analyze some new data”
- IT collects requirements
- Creates normalized and/or dimensional data models
- Profiles and conforms and the data
- Sophisticated ETL programs and quality standards
- Loads it into data models
- Builds a BI semantic layer
- Creates dashboards and reports

**IT:** “You’ll have your data in 3-6 months to see if it has value!
- Onboarding new data is difficult!
- Rigid Structures and Data Governance
- Disconnected/removed from business
The New Data Paradigm

OLD WAY:
- Structure Data → Ingest Data → Analyze Data
- Fixed Capacity
- Monolith

NEW WAY:
- Ingest Data → Analyze Data → Structure Data
- Dynamic Capacity
- Ecosystem

RECIPE:
- Cloud
- Data Lake
- Holistic Architecture & Framework
Ingest Raw Data
Organize, Define, Complete
Munging, Blending Machine Learning
Arbitrary/Ad-hoc Queries and Reporting

Usage Pattern

Data Lake – Integrated Sandbox
Data Science Workspace
Big Data Warehouse

Data Governance

Fully Governed (trusted)
Data Quality and Monitoring
Metadata, ILM, Security
Data Catalog
Data Integration
Metadata, ILM, Security

Landing Area – Source Data in “Full Fidelity”

Corporate Data Pyramid (CDP)

#DAMADay @joe_caserta
Why Use Spark?

Spark has become our *default* processing engine for a data engineering & science

- Development local or distributed is identical
- Beautiful high level API’s
- Full universe of Python modules
- Open source and Free
- Blazing fast!
# The Data Lake on the Cloud

- Remove barriers between data ingestion and analysis
- Democratize data with Just Enough Data Governance (JEDG)

<table>
<thead>
<tr>
<th>Cloud Component</th>
<th>AWS</th>
<th>Google</th>
<th>Microsoft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalable distributed storage</td>
<td>S3</td>
<td>GCS</td>
<td>Azure Storage</td>
</tr>
<tr>
<td>Pluggable fit-for-purpose processing</td>
<td>EMR</td>
<td>DataProc</td>
<td>HDInsight</td>
</tr>
<tr>
<td>Compute Services</td>
<td>EC2</td>
<td>GCE</td>
<td>VMs</td>
</tr>
<tr>
<td>Consistent extensible framework</td>
<td>Spark</td>
<td>Spark</td>
<td>Spark</td>
</tr>
<tr>
<td>Dimensional MPP Data Warehouse</td>
<td>Redshift</td>
<td>BigQuery</td>
<td>Azure SQL Data Warehouse</td>
</tr>
<tr>
<td>Data Streaming</td>
<td>Kenesis</td>
<td>PubSub</td>
<td>Azure Stream</td>
</tr>
<tr>
<td>Common Interface</td>
<td>Jupyter</td>
<td>DataLab</td>
<td>Azure Notebook</td>
</tr>
</tbody>
</table>
The Notebook is the ETL Tool

Here's some code in Scala:

```scala
val df = sqlContext.sql("Select col_1, date_time, date(date_time) as date from ommiturelogs")
df.write.mode("overwrite")
.partitionBy("date")
.parquet("/mnt/ommiture_parquet")
```

Here's the same code in Python:

```python
df = sqlContext.sql("Select col_1, date_time, date(date_time) as date from ommiturelogs")
df.write.mode("overwrite").partitionBy("date").parquet("/mnt/ommiture_parquet")
```

Now let's try SQL:

```sql
CREATE TABLE ommiturelogs_parquet LIKE ommiturelogs STORED AS PARQUET;
SET PARQUET_COMPRESSION_CODEC=snappy;
INSERT INTO ommiturelogs_parquet select * from ommiturelogs;
```
Data Quality and Monitoring

• BUILD a robust data quality subsystem:
• Metadata and error event facts
• Orchestration
• Based on Data Warehouse ETL Toolkit
• Each error instance of each data quality check is captured
• Implemented as sub-system after ingestion
• Each fact stores unique identifier of the defective source row
Unifying the Customer Across Channels

**Customer Data Integration (CDI):**
Match and manage customer information from all available sources
Marketing channels: DMP, Salesforce, Adobe, Social, Direct Mail, Call Center, CRM

In other words...

We need to figure out how to **LINK** people across systems!
Mastering Master Data is Still MDM

Validate

Standardize

Match

Survivorship
Standardization and Matching

Cleanse and Parse:
• Names
  • Resolve nicknames
  • Create deterministic hash, phonetic representation
• Addresses
• Emails
• Phone Numbers

Spark map operations:
• Data cleansing, transformation, and standardization
  – Address Parsing: usaddress, postal-address, etc
  – Name Hashing: fuzzy, etc
  – Genderization: sexmachine, etc

Matching:
Join based on combinations of cleansed and standardized data to create match results:

```sql
create table matches
as
select c1.id as xid, c2.id as yid, "phone" as match_type
from customer_cleansed c1
join customer_cleansed c2
on c1.name_hash = c2.name_hash
and c1.phone_std = c2.phone_std
where c1.id <> c2.id
```
Mastering Unmanageable Source Data

Reveal
- Wait for the customer to “reveal” themselves
- Create link between anonymous self and known profile

Vector
- May need behavioral statistical profiling
- Compare use vectors

Rebuild
- Recluster all prior activities
- Rebuild the Graph
The Matching Process

The matching process output gives us the relationships between customers:

<table>
<thead>
<tr>
<th>xid</th>
<th>yid</th>
<th>match_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>4849</td>
<td>phone</td>
</tr>
<tr>
<td>4849</td>
<td>5499</td>
<td>email</td>
</tr>
<tr>
<td>5499</td>
<td>1235</td>
<td>address</td>
</tr>
<tr>
<td>4849</td>
<td>7788</td>
<td>cookie</td>
</tr>
<tr>
<td>5499</td>
<td>7788</td>
<td>cookie</td>
</tr>
<tr>
<td>4849</td>
<td>1234</td>
<td>phone</td>
</tr>
</tbody>
</table>

Great, but it’s not very useable, you need to traverse the dataset to find out 1234 and 1235 are the same person (and this is a trivial case)

And we need to cluster and identify our survivors (vertex)
Graph to the Rescue

Don’t think table... think Graph!

These matches are actually communities

We just need to import our edges into a graph and “dump” out communities
Connected Components

Connected Components algorithm labels each connected component of the graph with the ID of its lowest-numbered vertex.

```
val graph = Graph(nodes_rdd, edges_rdd)
val cc = graph.connectedComponents().vertices
```

This lowest number vertex can serve as our “survivor” (not field survivorship).

<table>
<thead>
<tr>
<th>xid</th>
<th>yid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>4849</td>
</tr>
<tr>
<td>1234</td>
<td>5499</td>
</tr>
<tr>
<td>1234</td>
<td>1235</td>
</tr>
<tr>
<td>1234</td>
<td>7788</td>
</tr>
<tr>
<td>1234</td>
<td>7788</td>
</tr>
<tr>
<td>1234</td>
<td>1234</td>
</tr>
</tbody>
</table>
Identity Resolution Process

Identity Management

Matching
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>4849</td>
</tr>
<tr>
<td>4849</td>
<td>5499</td>
</tr>
<tr>
<td>5499</td>
<td>1235</td>
</tr>
<tr>
<td>4849</td>
<td>7788</td>
</tr>
<tr>
<td>5499</td>
<td>7788</td>
</tr>
</tbody>
</table>

Clustering

Survivorship

Parent | Child
-------|-------
1234   | 1234  
1234   | 1235  
1234   | 4849  
1234   | 5499  
1234   | 7788  

Matches are actually communities representing an individual
### The BDW is still Dimensional

<table>
<thead>
<tr>
<th>Phase</th>
<th>Source</th>
<th>Business Process</th>
<th>Date</th>
<th>Individual/ Members/ Household</th>
<th>Prospects</th>
<th>Employees</th>
<th>Anonymous Users</th>
<th>Locations</th>
<th>Campaigns</th>
<th>Channels</th>
<th>Service Providers</th>
<th>User Actions</th>
<th>Event Type</th>
<th>Call Category</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IMAX</td>
<td>Advocacy</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>?</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>IMAX, KONNEX</td>
<td>Renewal</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PMAX</td>
<td>Acquisition</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>IMAX</td>
<td>Service Provider Purchases</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Salesforce, Adobe</td>
<td>Digital Marketing</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Unica,</td>
<td>Marketing</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Unica,</td>
<td>Volunteering</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>IMAX</td>
<td>Events</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>IMAX</td>
<td>AARP Contributions</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>IMAX</td>
<td>Service Provider Agreements</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>Call Center Data?</td>
<td>Customer Service</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Adobe</td>
<td>Web Activity</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>Konnex</td>
<td>Billing/Payments</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Use Graph for Data Lineage

# DAMADay
@joe_caserta
Sample Solution Architecture
Thank You

Joe Caserta
President, Caserta Concepts
joe@casertaconcepts.com
@joe_Caserta

Data is not important, it’s what you do with it that’s important!

- Award-winning company
- Transformative Data Strategies
- Modern Data Engineering
- Advanced Architecture
- Innovation Partner
- Strategic Consulting
- Advanced Technical Design
- Build & Deploy Solutions
- BDW Meetup
- New York City
- 3,000+ members
- Knowledge sharing

#DAMADay

@joe_caserta