Dismantling data silos through cloud integration

Pythian
About myself

- Danil Zburivsky
- Director of Engineering at Pythian
- Author of Hadoop Cluster Deployment book
- Author of Designing Cloud Data Platforms
- @zburivsky 🍀
Helping businesses transform and win in the data economy
ETL is not a new idea. Why is data integration still a challenge?
- Volume
- Variety
- Velocity
- Veracity
ETL Tool

RDBMS Data Warehouse

- storage
- processing
- SQL
Volume: storage and processing are tightly coupled
Variety: ETL tools and SQL offer limited functionality
Velocity: Warehouse design is batch oriented. Hard to implement real time analytics
Veracity: “Bring your own dataset” raises data quality questions.
Monolith design
Data lake or data warehouse?
Hadoop didn’t quite play out as a unified data platform
Modern data platform requires modular design
And it requires Cloud!
● Multiple services that can be combined together
● Elasticity
● PaaS
Cloud Data Platform: best of both data lake and a data warehouse
The ingestion layer connects to source systems and brings data into the data lake, preserving original data formats.

---

**Build your ingestion layer to support batch and stream ingests as first class citizens for maximum flexibility.**

---

**Fast message store. Has data expiry policies**

**Real time analytics**

**Direct data lake access**

**“Slow” storage. Permanent data retention**

**ETL tools overlay**
Fast storage is a message bus for data that is coming in streams, message by message.
A distributed data processing engine is a must. You might need one for batch and one for real time.
Fast message store. Has data expiry policies

Real time analytics

Direct data lake access

“Slow” storage. Permanent data retention

ETL tools overlay

Warehouse is a destination for curated data sets. It’s role is to provide results of the queries back to users. fast.
Not everyone wants access to curated data sets. Reading raw files from storage may be preferred by many ML applications.
ETL overlay tools are optional, but can boost productivity and also sometimes come with metadata tracking capabilities.

- **Stream**
  - Ingestion
  - Processing
  - Direct data lake access

- **Batch**

- **Fast message store. Has data expiry policies**
- **Real time analytics**
- **Metadata**
- **Warehouse**
- **“Slow” storage. Permanent data retention**

**Data consumers**
Consumers are not just people anymore. Some may need to consume streams directly, some daily batch extracts.

- **Fast message store. Has data expiry policies**
- **Real time analytics**
- **Direct data lake access**
- **“Slow” storage. Permanent data retention**
- **ETL tools overlay**

- **stream**
- **ingestion**
- **batch**
- **processing**
- **metadata**
- **warehouse**
- **Data consumers**
Case Study: A global fashion brand implements a Cloud Data Platform on GCP
Key Challenges:

- Data variety
- Data quality
- Self service
```json
{
  reports: [
    {
      columnHeader: {
        dimensionsHeader: ["ga:campaign", "ga:country", "ga:device"],
        metricsHeader: ["ga:pageViews", "ga:sessions", "ga:bounces"]
      },
      data: {
        rows: [
          {
            dimensions: ["some campaign 1", "Canada", "mobile"],
            metrics: ["112", "100", "10"]
          },
          {
            dimensions: ["some campaign 2", "Ukraine", "web"],
            metrics: ["80", "60", "20"]
          }
        ]
      }
    }
  ]
}
```

<table>
<thead>
<tr>
<th>ga:campaign</th>
<th>ga:country</th>
<th>ga:device</th>
<th>ga:pageViews</th>
<th>ga:sessions</th>
<th>ga:bounces</th>
</tr>
</thead>
<tbody>
<tr>
<td>some campaign 1</td>
<td>Canada</td>
<td>mobile</td>
<td>112</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>some campaign 2</td>
<td>Ukraine</td>
<td>web</td>
<td>80</td>
<td>60</td>
<td>20</td>
</tr>
</tbody>
</table>
Apache Spark for data transformations

```python
file_location = "gs://landing/incoming/GA"
file_type = "json"

df = spark.read.format(file_type).load(file_location)

df.createOrReplaceTempView("GA")

spark.sql("SELECT COUNT(*) FROM GA WHERE GA.Country='Canada'")
```
GCP Data Fusion for self-service ETL
GCP Cloud Composer (Apache Airflow) for scheduling and orchestration
What is possible today:
- Adding new source
- Self service analytics via Tableau
- Self service ETL
- Data quality process
LOVE YOUR DATA

Pythian