upsolver

Technical Whitepaper:

A Data Lake Approach to Event Stream Analytics

Executive Summary

Events logs are the archetypical source of streaming data - a continuously generated stream of semi-structured data, reflecting the state of a software or hardware system at any given moment - with some of the most common scenarios being IoT sensors, server and network security logs, app activity and advertising data.

While in the past the task of collecting and analyzing logs were associated mainly with IT and networking professionals, the rise of software development, clickstream and app analytics has led to increased demand event for log data analysis by data analytics teams, business units, and other non-IT stakeholders. This article will examine the common architectural challenges of operationalizing log data at scale, and suggest a solution based on a cloud data lake built and managed using Upsolver on Amazon S3.

Limitations of Databases and SQL

Traditionally, log analysis was seen as another analytical workload that could be offloaded to a database, against which SQL and BI tools could be used by data consumers to leverage data and generate insights. ElasticSearch, which was released less than a decade ago, has become the de-facto standard in this regard due to its ability to provide strong out-of-the-box performance for search and analytics.

However, as log data grows in volume and velocity, and with increasing requirements to use this data for new types of analysis - dashboards, operational analytics, ML model training and more - the drawbacks of a database approach become more apparent. These include:

1. **Cost -** databases lack elasticity, making them expensive and cumbersome to scale due to the need to constantly spin up, configure and resize clusters based on changes in data volume and retention.

2. Latency - data isn’t available in real-time because due to the latencies built-in to batch ETL processes.

3. Machine learning roadblocks - the database, which was built for OLAP analytics isn’t a good fit for data scientists that build models and train them against custom datasets. ML datasets require data from various points in time, making them easier to extract from an append-only log, as opposed to a datastore that allows updates and deletes. Additionally, data scientists want to use scripting in Python/Scala to build their datasets and apply functionality which would be very cumbersome to do in SQL.

4. Lock-in - databases rely on a proprietary data format which is optimised for their own query engine. This limitation creates vendor lock-in and forces organizations to choose between two bad options: force a use case on an existing database or replicate the data to another store which introduces consistency and reliability issues.

A Decoupled Architecture Solves Some Problems, Creates New Ones

Due to all of the issues above, the database fell out of favor as the core datastore for log data as organizations began searching for more scalable, cost-effective and agile solutions. Hence, in recent years we have seen the rise of the decoupled architecture, wherein raw data is ingested and stored on inexpensive object storage, with compute resources provided ad-hoc to different analytic services per use case.

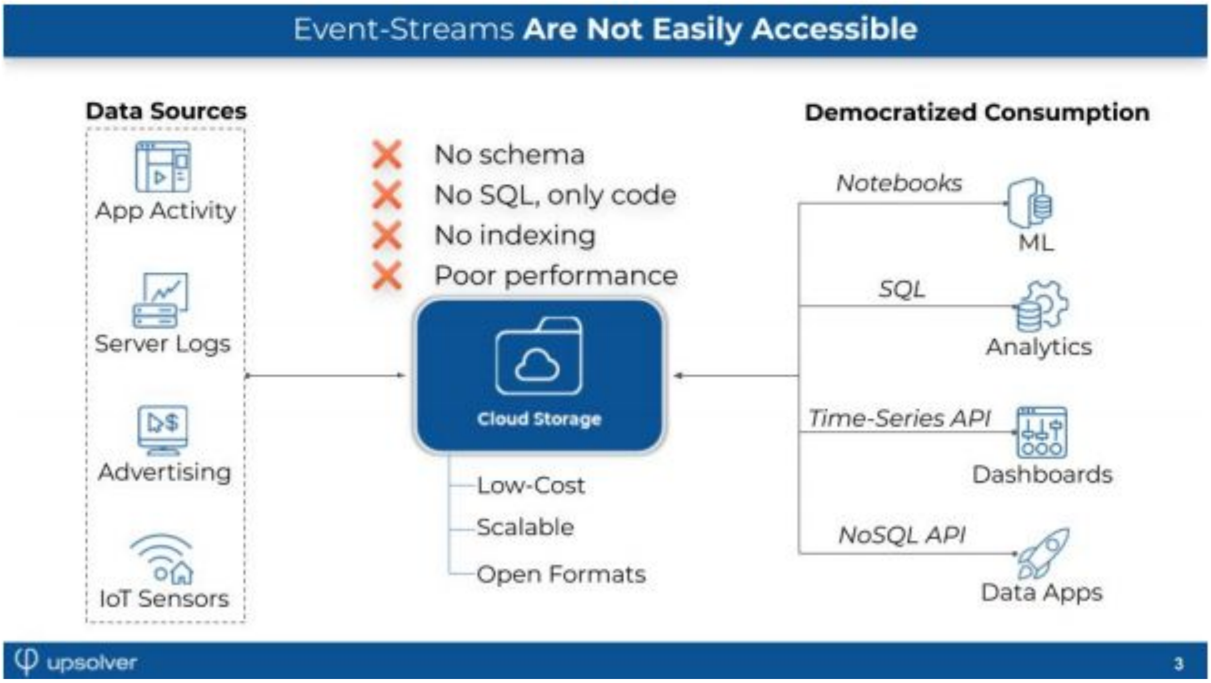
This data lake approach bypasses the roadblocks of traditional databases above by storing data as an append-only log on cloud storage, while enabling consumers to analyze the data with best-of-breed engines per use case. With this architecture, scaling is elastic, cost is low, machine learning is natively supported and data is stored in its raw format without ETL delays. This architecture heavily relies on open file formats (such as Apache Parquet) for storage and metadata stores for data discovery and queries.

However, and despite these benefits, a data lake is still missing something - namely, all the advantages of a database! Object storage is often messy, complex and slow to show value, with the main reasons including:

- Data stored on cloud storage can be compared to a bunch of files on disk. Databases were invented exactly in order to give users easier access to data by letting them run SQL instead of querying files directly, and this functionality is lacking in data lakes.
- On cloud storage, there is no schema or statistics to guide consumers to insights - instead, data discovery requires manual ETL efforts to pull samples of the data.
- Query latency can be 1000X slower compared to databases that include indexing, an optimized file system

and fast access to disk (much faster than the latency you get with cloud storage).

Addressing these data preparation and management challenges forces companies to spend hundreds of engineering hours on operations like [compacting small files](), which are code and effort-intensive but do not deliver direct value from a business point of view.



Event-Streams **Are Not Easily Accessible**

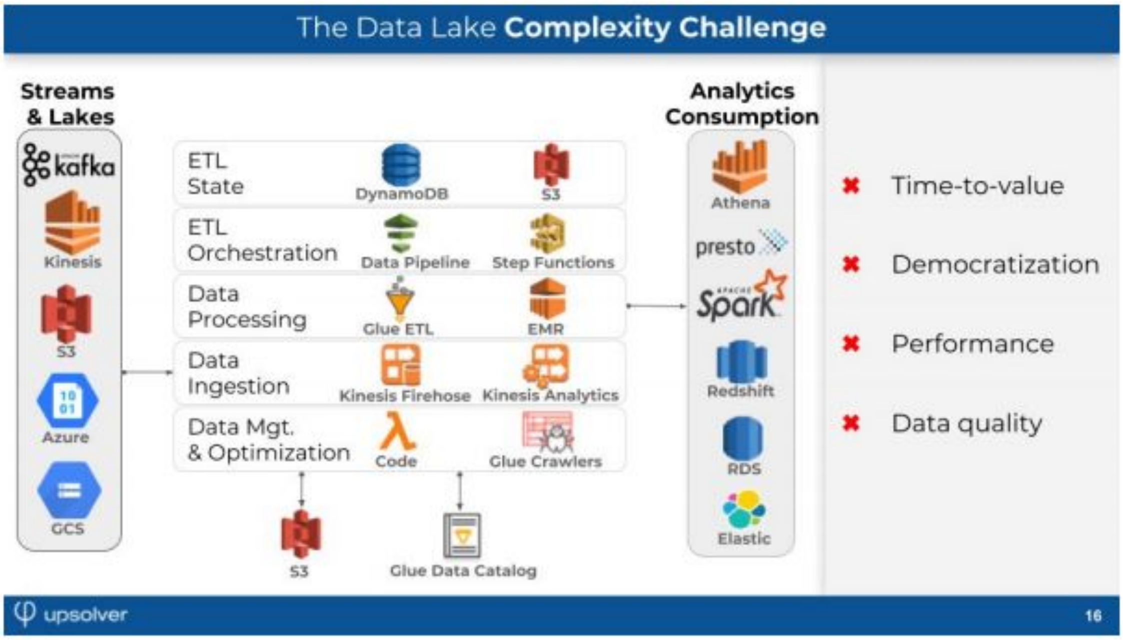
Closing the Gap Between Data Lake and Data Consumers

Up to this point we've explained why companies want to analyze logs and the challenges they encounter when trying to do this at high scales: databases are fragile and expensive, data lake architectures are complex and effort-intensive.

The significant gap between the data lake and its consumers created the need for a **data platform** which would operationalize data in cloud object stores. Companies can't compromise on parameters like elasticity or readiness for machine learning, so they seek a platform to preserve the functional advantages of databases. This data platform should offer simple APIs for enabling data consumers and automation for reducing IT overhead.

Query performance and data quality are dependent on companies' ability to implement a data platform effectively.

However, this is often easier said than done. Today, companies use a variety of cloud services and open source projects to build. their own data platform. The end result is often closer to the vision of self-service, democratized analytics, but the data-to-insight process is still coding- and IT-intensive, with over 80% of the time spent on building, orchestrating and scaling data pipelines. Services like AWS EMR or AWS Glue help with the IT overhead but the technical barrier of entry in order is still higher compared to writing SQL.



The Data Lake **Complexity Challenge**

Streams & Lakes

kafka
Kinesis
S3
Azure
GCS

ETL State — DynamoDB, S3

ETL Orchestration — Data Pipeline, Step Functions

Data Processing — Glue ETL, EMR

Data Ingestion — Kinesis Firehose, Kinesis Analytics

Data Mgt. & Optimization — Code, Glue Crawlers

S3
Glue Data Catalog

Analytics Consumption

Athena
presto
Spark
Redshift
RDS
Elastic

- ✖ Time-to-value
- ✖ Democratization
- ✖ Performance
- ✖ Data quality

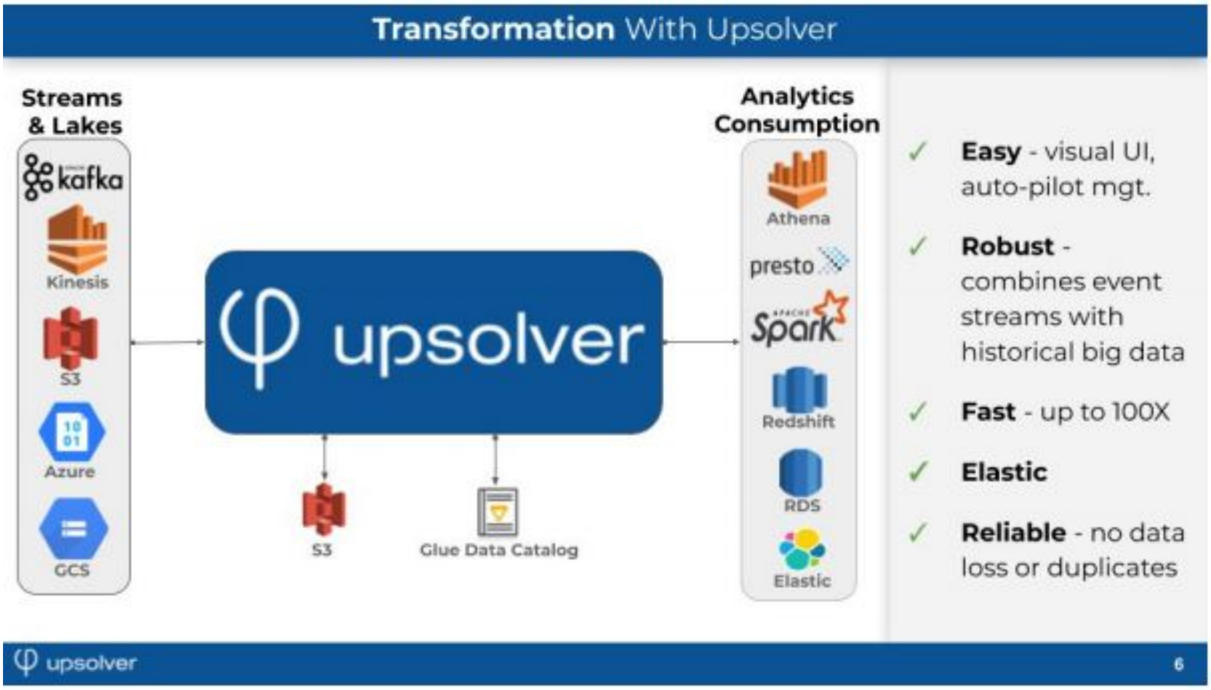
upsolver 16

As the big data landscape matures, three types of approaches have emerged:

- **Templates -** AWS promotes a best-of-breed approach by releasing a service per technical need and there are already 100s of services. To make it easier for customers to build solutions from multiple services, AWS released Data Lake Formation (in preview) which utilizes pre-defined templates in order to save some of the time spent on writing code and integrating services.
- **Databases that utilize a decoupled architecture -** Google BigQuery, Snowflake and DataBricks Delta are data warehouse services that decouple compute from storage. They scale elasticity and are priced by consumption of compute and query resources. This is definitely a big improvement compared to traditional RDBMS but the issue of vendor lock-in remains - these

services use from their query engine, while introducing new analytical engines requires replication of the data to a new store.

- **Self-service platforms over data lakes -** when building a long-term strategy, companies want to use data lakes for their cost-effectiveness while data consumption tools will come and go. To accomplish such a vision, companies will need open storage and metadata layers and a platform to keep their chaos organized and optimized. For example, Upsolver is a service located between the AWS data storage layer (S3+Glue Data Catalog) and analytical services such as Athena, Redshift and RDS. Upsolver enables companies to only use a visual interface to operationalize an AWS data lake so it will be accessible and useful for all data consumers. Upsolver does this by adding capabilities like data discovery, indexing, self-serve data preparation and performance optimizations to Amazon S3.


Transformation With Upsolver

- ✓ **Easy** - visual UI, auto-pilot mgt.
- ✓ **Robust** - combines event streams with historical big data
- ✓ **Fast** - up to 100X
- ✓ **Elastic**
- ✓ **Reliable** - no data loss or duplicates

Log data and event streams play a more prominent part than ever in modern big data architectures. The need to access and operationalize this data for analytics and machine learning is challenging even for massive organizations, and certainly for companies that have limited in-house data engineering resources.

In this whitepaper, we’ve covered several strategies for taming event log data - from databases to data lakes to modern data platforms. To see value from streaming data, most companies will have to transition from the former to the latter.