AMLGO LABS

Big Data Engineering Massive Parallel Processing

AMLGO LABS

Executive Summary

One of the biggest challenges in managing high-volume transactional systems is processing large data workloads within acceptable timeframes. A leading banking client approached Amlgo Labs with the need to redesign its rules engine for event-based accounting, where the system was taking nearly 6 to 8 hours to generate journal lines. This delay was impacting operational efficiency and downstream processes.

To address this, Amlgo Labs implemented a PolyBase-based architecture that enabled massive parallel processing, reducing processing time to just 30–40 minutes. The new design seamlessly integrated data from diverse sources and significantly improved the performance of the bank’s core accounting system.

AMLGO LABS

Background & Goal

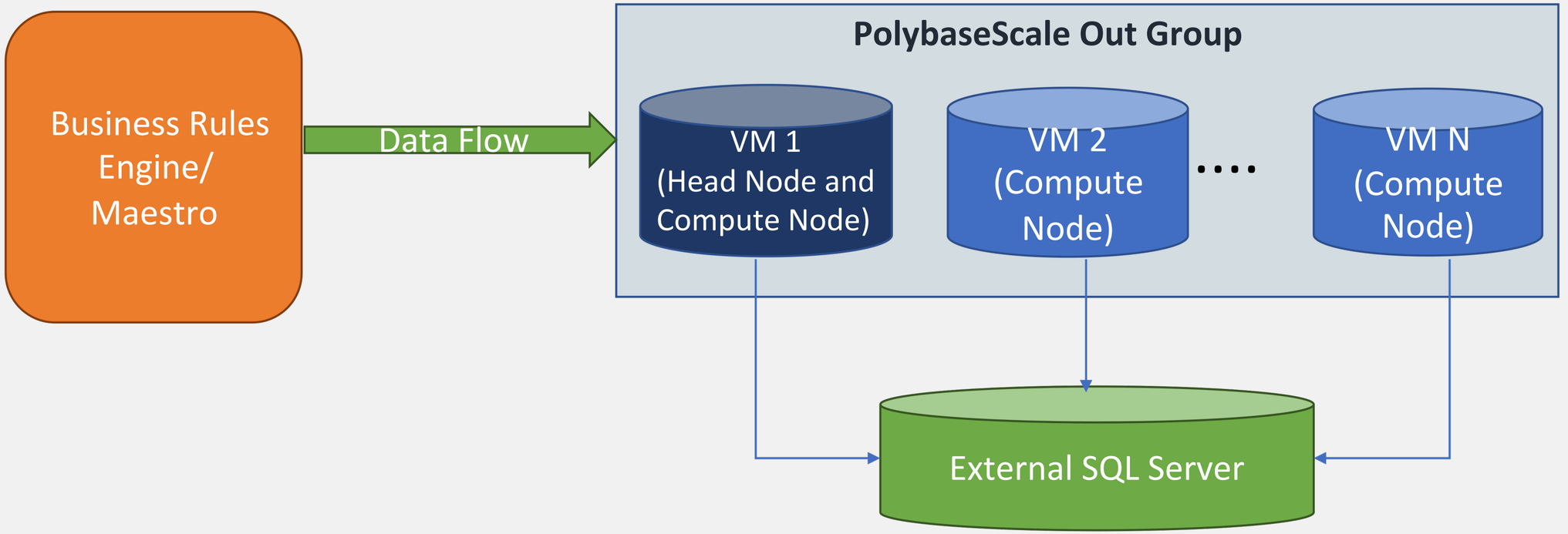
The client’s existing system processed high volumes of accounting events across heterogeneous data sources, including Hadoop, Oracle, Teradata, DB2, MongoDB, and flat files. These data sets fed into a rules engine that generated journal lines based on complex accounting logic and calculations.

The goal was to:

- Reduce the processing time from hours to minutes
- Enable scalable, high-performance data processing
- Ensure seamless integration of structured and unstructured data
- Maintain accuracy and reliability in financial outputs

AMLGO LABS

Solution Proposed : Parallel Processing using Polybase



A TSQL is hitting for instance VM1 and we have VM2 and VM3 as well with an External SQL Server. Now the query hitting the VM1 gets break down into 3 parts for VM1, VM2 and VM3 and they will individually compute them and the results will be forwarded to the Head Node in this case VM1, resulting in more computing power, hence increasing performance.

AMLGO LABS

ImplementationSteps

Study Current Database Architecture

- Test environment and Infrastructure study of existing application.
- Maestro/FSDB HLD analysis, i.e. table structure and normalization study.
- Analysis of partitions and indexes on big tables.
- Locking and transactions set up analysis.

Design and Implementation of Polybase Clusters

- Installing and Setting up polybase cluster on sql server – in master and slave model
- Restructuring and reconfiguring the load balancing mechanism with different nodes
- Creating scripts for all the modules (views, stored procedures, functions)

Identifying Other Improvement Areas

- Optimization of existing stored procedures and long running queries.
- Implementing column structured indexes and techniques like sharding, partitions, load balancing, parallel processing etc
- Improvement areas for Locking and transactions set up.

Regression Testing and Conclusions

- End to end testing of complete set up along with unit testing.
- Measuring performance and Monitoring Data Insertion, String Manipulation, query plans, Analysis of partitions and statistics.
- Monitor Performance and Create Metrics

AMLGO LABS

Key Insights & Highlights

Massive Performance Gains

Processing time for journal line generation reduced from **6–8 hours to just few minutes**

Scalable Data Architecture

PolyBase allowed data to remain in its native system while still being processed efficiently.

Efficient Workload Distribution

Compute-intensive tasks were split across **multiple compute nodes**, allowing for near-linear performance scaling.

Support for Heterogeneous Sources

Enabled seamless querying and integration from a wide range of **structured and unstructured sources**.

Business Impact

AMLGO LABS

Impact Area	Outcome
Processing Speed	Reduced journal line generation time by over 80%
Operational Efficiency	Faster processing led to quicker downstream reconciliation and reporting
Infrastructure Scalability	Architecture supports increased volumes without performance loss
Data Flexibility	Enabled integration across various legacy and modern data platforms
Cost Savings	Reduced processing time resulted in lower resource consumption and operational costs

AMLGO LABS

Conclusion

The newly created design integrates data coming from heterogenous data sources like like Hadoop, Oracle, Teradata, DB2, MongoDB, Flat files, any other unstructured data seamlessly. In our Use case of Event Based Accounting where we are generating the data set based on business rules with high calculations and all the data sits at different place, Polybase implementation was very fruitful reducing time from hours to few minutes in many scenarios.

This project is a clear example of how thoughtful data engineering strategies and the right use of modern technologies can deliver tangible results in complex, high-volume enterprise environments.

Amlgo Labs: Engineering Scalable Solutions for High-Performance Data Processing.

AMLGO LABS

Contact US

+91 85120 01385

info@amlgolabs.com

www.amlgolabs.com

AMLGO Labs