International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V71I5P110

Volume 71 Issue 5, 57-61, May 2023

Original Article

Amazon Athena : Serverless Architecture and Troubleshooting

Amol Kulkarni

SAP Solution Architect, Data-Core Systems Inc, Bristol PA.

Received: 27 March 2023 Revised: 01 May 2023 Accepted: 17 May 2023 Published: 30 May 2023

Abstract *- Data Analysis is of topmost import important for the survival of any organization in today's digital and competitive world. In order to store enormous amounts of data, different OLTP ERP Systems like SAP S/4HANA, Amazon S3, Oracle Net Suite etc. are available, but accessing that data in a simple and easiest way is tricky due to the heavy cost involved in Extraction, transformation and loading ETL operations, maintenance, monitoring and data visualization in data warehousing. Serverless Systems come into the picture for data analysis due to their features like simplicity, ease of operation, automatic scalability and less cost for accessing huge data. There are various serverless systems available in the market, like Amazon Athena, Amazon Glue, Microsoft Azure, and Google Big Query. In all these serverless systems, Amazon Athena is best used for cost savings, easy access to huge amounts of data, tight integration with Amazon S3, scalability, security and simplicity. This paper aims to explain the serverless architecture of Amazon Athena and how to troubleshoot different issues in Amazon Athena while accessing huge data.*

Keywords *- Amazon S3, AWS Glue, Business Intelligence(BI), ETL, Machine Learning(ML), SQL.*

1. Introduction of Amazon Athena

Amazon Athena, introduced in 2016, is a serverless query engine service that does not need any infrastructure to access and analyze the data in the source systems like Amazon S3, Amazon Redshift, SAP HANA etc. It uses SQL or Python to query the data in Amazon S3, which is tightly integrated. Amazon Athena is built on Open-source Trino and Presto Engines and Apache Framework. It is very fast, cost-effective, and easily scalable to access petabytes of data. It is charged only $5 for accessing 1 TB of data which is very cheap compared to other serverless accessing tools. Also, it is not charged for any DML DDL commands or connections. It does not need any ETL operations and is hence easy to manage. Athena, after running and executing a query on the source system data, pushes data to Data Lakes for Amazon Quick Size data visualization. Athena scalability is automated, and we do not have to bother with existing size and future scalability. Using Athena, we can access unstructured, semi-structured, and structured data on Amazon S3. We can use Amazon Athena within a Jupyter Notebook for machine learning purposes. Mobileye Global Inc. uses Amazon Athena to enable autonomous driving.

2. Literature Review

Serverless Architecture is supported by different systems like Amazon Glue, Amazon Athena, Google Big Query, Microsoft Azure etc. All these systems are cloud-based, and choosing the right system is tricky from different factors like cost, maintenance and ease to handle, scalability and elasticity, speed of data access, fetching huge amounts of data, durability and availability, Security, Authorization, encryption, supporting a wide variety of interfaces and data sources. Amazon Athena is the best choice for serverless analytics as it supports all these features.

Not all data we extract into Data-Warehouse via ETL service, but some data we need for getting an insight into the business, and Amazon Athena is best used for cases such as accessing log files, ad-hoc analysis, and data exploration. Athena is tightly integrated with Amazon S3 and can fetch petabytes of data with high speed and minimum cost for scanning the data.

3. Architecture of Amazon Athena

Amazon Athena Architecture is very simple as it is serverless and comprises Source Systems, Amazon Glue, AWS Console, SQL, Data lakes and client applications like Amazon Quick-Sight.

3.1.1. Client Applications

Amazon Athena Integrates Business Intelligence and Analytical tools like Microsoft Power BI, Tableau, Amazon Quicksight etc., by means of different connectors for each tool.

Different clients easily integrate with Athena and fetch the data for analysis and reporting purposes. SQL Clients can be connected to Amazon Athena using JDBC ODBC drivers.

3.1.2. Connections

JDBC drivers are used to connecting Amazon Athena to Amazon S3 and fetch the CSV, JSON, Parquet, Avro, and Orc format of data.

AWS Glue stores the metadata of Amazon S3 and can be easily integrated with Amazon Athena to fetch the data of different tables in Amazon S3. External Hive Metastore can be connected to Amazon Athena by means of AWS Lambda functions.

In order to access data from other source systems other than Amazon S3, we have to use Amazon Athena Federated Query to write the connections to the source system. Athena can invoke ML models deployed on Amazon SageMaker. All these connections provide great speed to fetch huge amounts of data, are secured and are excellent performance-wise.

3.1.3. Source Systems of Amazon Athena

Amazon Athena can query on AWS Systems like Amazon S3, Amazon Redshift, Amazon CloudWatch, Amazon OpenSearch, Amazon Neptune etc.

Amazon Athena can query directly on third-party systems like MySQL, PostgreSQL, Oracle, Microsoft Azure, Microsoft SQL Server, Google Big Query, Snowflake, and SAP HANA.

Athena can also connect to various other source systems by means of creating custom connections via AWS Lambda.

3.1.4. AWS Glue

AWS Glue stores the metadata of the Amazon S3 and is connected to Amazon Athena. This way, we do not have to define table structures in Athena. AWS Glue Crawlers automatically scan, and catalog data in Amazon S3, and Athena SQL queries can call the structures in AWS Glue and then fetch the data from Amazon S3. This way, the table creation in Athena can be avoided. With AWS Glue connection to Athena, all table names and their columns appear automatically into Athena to use them while querying.

3.1.5. Amazon Athena Federated Query

In order to fetch data from NON-Amazon S3 data sources, we can use Athena Federated Query to query the data in the source. AWS Lambda provides necessary custom connections to connect data sources like Amazon Redshift, MySQL, Amazon DocumentDB, and Amazon RDS to Athena, Azure Data Lakes, and SAP HANA.

3.1.6. Data Visualization Tools for Amazon Athena

Amazon Quicksight is a data visualization tool connected to Amazon Athena or Athena Federated Query to fetch data for reporting needs. QuickSight is also a serverless and easy to manage and cost-effective tool. We can also use Tableau or other third-party tools like PowerBI as a front-end tool to access Amazon Athena data for dashboard development. Tableau is connected to Athena via the JDBC driver. The performance of Dashboards built on Athena data in front-end tools like Amazon QuickSight and Tableau is excellent.

4. Amazon Athena Features and Performance Improvement

4.1. Athena Features

Amazon Athena is serverless, and there is no infrastructure and no administration. We just have to build the schema in Athena or fetch the schema from Amazon Glue and then start querying the data of Amazon S3 directly into Athena using Standard SQL. The data fetched is secure and encrypted. Athena automatically scales as per the requirements, and we do not have to scale it manually. The cost is only for fetching the data, not if there are failed sql statements, create, delete, or alter SQL commands. It charges $5 for 1 TB of data scanned. The cost is optimized by partitioning the source table data, columnar storage of the table and compression techniques. Athena supports parallel query execution on the source table data; hence, performance is very fast. Athena can invoke Machine Learning models located at Amazon SageMaker in SQL query for predictive analysis.

4.2. Athena Performance Improvement

4.2.1. Compression of the Data

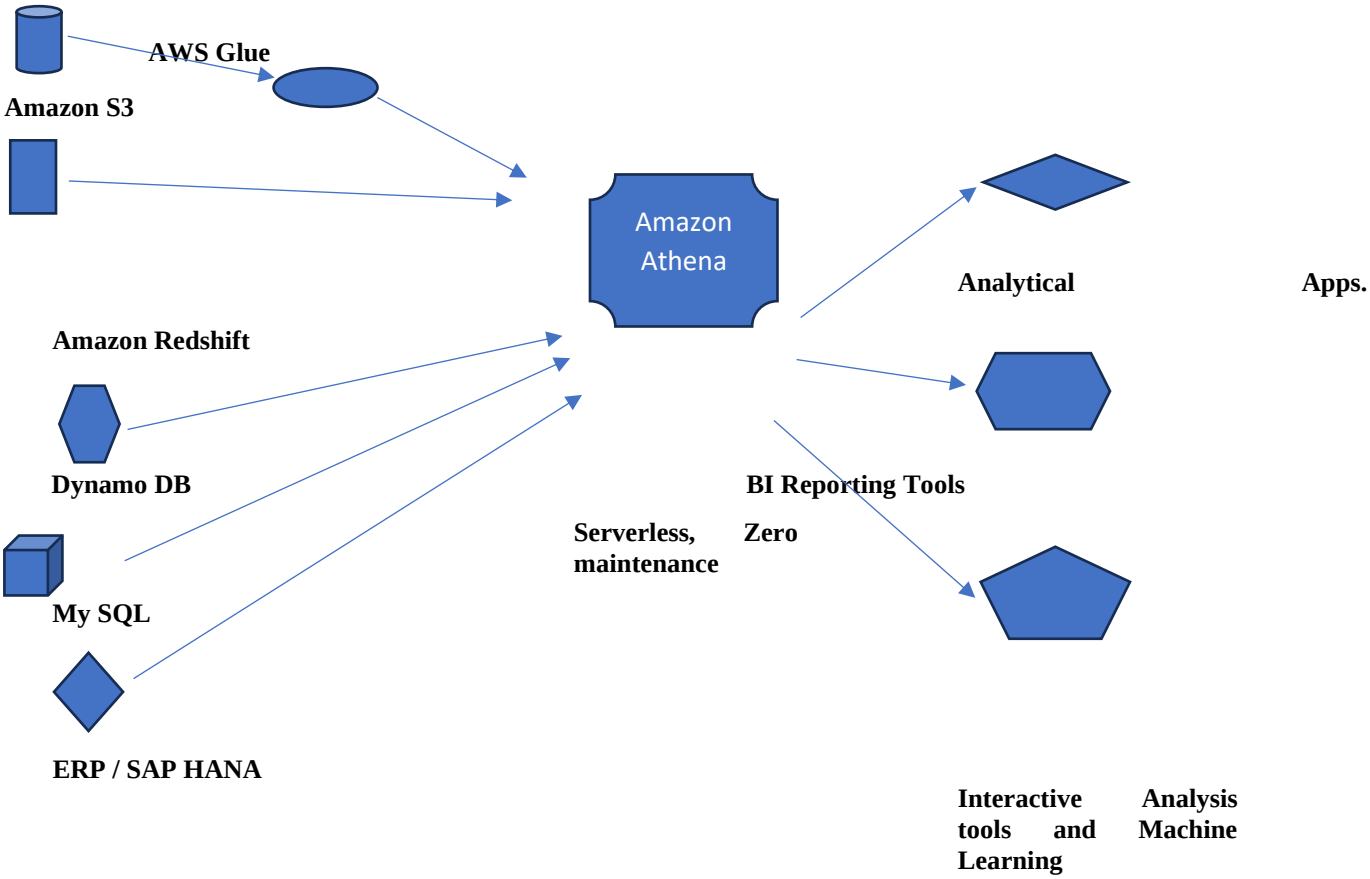
Compressing the files means splitting the data of the files to perform parallel query execution. The file formats supported for compression are text files, Parquet, Orc, JSON, CSV etc. Parquet and Orc files are, by default, compressed and splitable.

4.2.2. Columnar Storage

Data stored in columnar format is easily compressed and fetched easily via parallel processing. Apache Parquet and Orc file formats have the ability to fine-tune block size, which supports compression by default.

4.2.3. Partition of the Data

Partitioning of the data reduces the number of records scanned as data is stored in a particular partition only is read. Athena supports the Hive partitioning in which we mention the value of the column to be partitioned.



Amazon Athena Architecture Diagram as above

Source Systems -→ Amazon Athena -→ Frontend Reporting Tools

5. Amazon Athena Troubleshooting

5.1. Tables return ZERO Records

Zero records are fetched while accessing the tables in Amazon S3 if the complete table location paths with extensions like. CSV, TXT or. JSON is not defined, or if the path itself is not correctly defined or in AWS glue, all table partitions are not defined in metadata. In order to fix this issue verify the table path and make sure partitions are present in AWS glue metadata.

5.2. Access Denied Error

Access Denied error can occur if the permission for read or write is not defined in AWS Identity and Access Management. Access Denied Error also occurs if the Amazon S3 and Athena tables are in different buckets. The object's owner is not the same as that of the S3 bucket owner. AWS glue does not allow access to users. Encrypted data access key permission issue also creates access-denied issues. Fix for these issues to take care of permissions and access to different objects and definitions.

5.3. Error while Reading Data from JSON

Error while reading data from JSON file occurred in Athena mainly due to malformed JSON data.

HIVE_CURSOR_ERROR message occurred if the data in the JSON is incorrect. In order to fix this issue, we have to create a new table with SERDEPROPERTIES (ignore.malformed.json' = 'true') and try and if this does not fix, then try to convert the multiline JSON records to single line via SQL create table function and then fetch the table then call the table path into Athena SQL. It will give the correct output of the valid JSON data.

5.4. Stale View Error in Amazon Athena

Stale view refers to the tables which are modified in the source system Amazon S3. If you query on a view that uses a modified table, then we get the error SYNTAX_ERROR: View is Stale; it must be recreated. In order to fix this issue, we have to create the view again or replace it with a new view and then use it in the Athena query.

5.5. Unable to Verify/create Output Bucket Error in Athena

Unable to Verify/Create output bucket error occurred in Athena if the query bucket access roles are not defined properly in Amazon S3. In order to fix this issue go to S3 roles for queries and confirm that permissions are as per the attached policy to run the query bucket. Ensure the IAM statement does not contain the denial statement for the IP address used to access the query. If there is an issue, then use the example policy given to provide access to the role, which gives query access to the query in Athena and will fix the issue.

5.6. Error Failed ParseException Line 1: X missing EOF

Sometimes the database mentions having a dash - between 2 words, and if you use the show CREATE TABLE statement, it will throw an error message as 'Failed ParseException Line 1: X missing EOF. In order to fix this issue, there should create a database with an underscore between 2 words and then use the show table statement.

5.7. Error Opening Hive Split-503 S3 Slow Down

503 S3 slow-down error occurred when Athena sent many query fetch requests per second to S3. There is a limit per second per prefix in the Amazon Storage bucket. If that exceeds, then we get the 503 S3 to slow down the error. This issue can be fixed by partitioning the S3 tables data and merging small files into larger ones using the s3distcp tool to reduce the number of requests made. Another way to fix this issue is to use cloud watch to find other applications if they are using the same S3 prefix that Amazon Athena query is using. If there is a discrepancy, then change the schedule to access the S3 prefix.

5.8. Incorrect Syntax in SQL Statements of the Query

While using AWS glue for metadata of the table and calling the table in Athena, use the statement table type attribute as EXTERNAL TABLE at the end; else, the error message FAILED: NullPointerException name is null occurred. Always use the functions predefined by Athena in the SQL; else, we get a function not registered error message. Sometimes various GENERIC_INTERNAL_ERROR exceptions error occurred as syntax errors due to different reasons like schema mismatch, columns of data type array which is not supported, data type INT with numeric value very high is not supported, partitions and filters does not match,

5.9. Troubleshooting Workgroups

Amazon Athena workgroups separate workloads, users, and applications and manage the cost of queries. In order to troubleshoot workgroups check the access for users to fetch the source table location and make sure the workgroup is not disabled or deleted else, we get INVALID_INPUT; while using API I, the query the output location of the query results is not defined properly then error message InvalidRequestException will occur, if the workgroup prepared statement limit of 1000 is exceeded then an error message is cannot create prepare the statement and one has to DEALLOCATE PREPARE to remove one or more prepared statements from the workgroup to fix the issue, the query in the workgroup should not mention external location else it will fail, so remove external location statement from query to fix the issue.

5.10. Troubleshoot Athena Network API calls using Cloud Trail

In order to troubleshoot Athena Network API calls, examine the cloud trail logs discrepancies for start sessions, terminate sessions, import notebook etc. This will provide the actions taken by users and then fix them accordingly.

6. Miscellaneous Topics of Amazon Athena

6.1. Athena ACID Transaction use

ACID transactions are database transactions with properties such as atomicity, consistency, isolation and durability, ensuring data integrity. Users can perform add and delete objects one at a time by means of ACID transactions which provide support for SQL DML Commands. Athena ACID transactions are used to modify the data in Amazon S3.

6.2. Interactive Data Analytics using Apache Spark

Amazon Athena supports Apache Spark for interactive data analytics. Spark code is used in Athena to retrieve the results of the query for interactive data analytics without the need for additional configuration.

6.3. Handling Unstructured Data in Amazon Athena

Unstructured data is stored in Amazon S3 in the form of JSON files. These JSON format files can be accessed in Amazon Athena easily. Unstructured data is either a struct data type or an Array data type; a typical example of unstructured data is Application Log files can be fetched in Athena.

6.4. Machine Learning with Amazon Athena

Amazon Athena integrates Amazon SageMaker to access different Machine Learning ML models using the SQL Query of Amazon Athena. USING EXTERNAL FUNCTION Clause ML can be used in Athena. Federated Query in Athena accesses the data from different source systems like Amazon Redshift, Dynamo DB etc., via Jupiter notebooks and sends it to Amazon SageMaker and then to S3, and it is further read by Athena via AWS Glue by means of Machine Leaning Query.

7. Conclusion

Amazon Athena is evolving, and new and new features are added periodically to serve better and eliminate any issues. It is easy to use, has zero maintenance, has no infrastructure, and has very cheap features, making it the first

choice for a lot of scenarios where we do not want to perform costly ETL operations. It is easy to integrate with ML and AWS Glue and can fetch data from various source systems with ETL operations, which is why different customers widely use across it.

Athena provides support for interactive analysis for fast, scalable queries on petabytes of data, different BI tools for data visualization, Analytical Apps for business insights, and Machine Learning for combining data from different source systems.

It automatically scaled, robust, secured and encrypted data; using wide SQL statements to fetch data makes it an alternative to different data warehouses, which needs ETL operations and maintenance.

References

[1] Alexander S Gills, Amazon Athena . [Online] Available: https://www.techtarget.com/searchaws/definition/Amazon-Athena

[2] Anthony Virtuoso, Mert Turkay Hocanin, and Aaron Wishnick, Serverless Analytics with Amazon Athena, 2021. [Online] Available: https://www.oreilly.com/library/view/serverless-analytics-with/9781800562349/

[3] Mert Hocanin, and Pathik Shah, Top 10 Performance Tuning Tips for Amazon Athena. [Online] Available: https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/

[4] Dhiraj Thakur, Building AWS Data Lake visualizations with Amazon Athena and Tableau. [Online] Available: https://aws.amazon.com/blogs/big-data/building-aws-data-lake-visualizations-with-amazon-athena-and-tableau/

[5] Amazon QuickSight. [Online] Available: https://aws.amazon.com/quicksight/

[6] Amazon Athena features. [Online] Available: https://aws.amazon.com/athena/features/

[7] Ernesto Marquez, How to use Athena to troubleshoot AWS operations issues. [Online] Available: https://www.techtarget.com/searchaws/tip/How-to-use-Athena-to-troubleshoot-AWS-operations-issues

[8] Noreen Hasan, Amazon Athena, [Online] Available: https://acloudguru.com/blog/engineering/amazon-athena-explained-what-is-it-and-when-should-i-use-it

[9] Amazon RedShift Spectrum vs Amazon Athena. [Online] Available: https://digitalcloud.training/amazon-redshift-spectrum-vs-amazon-athena/

[10] Troubleshooting in Athena. [Online] Available: https://docs.aws.amazon.com/athena/latest/ug/troubleshooting-athena.html

[11] Workgroups in Athena. [Online] Available: https://docs.aws.amazon.com/athena/latest/ug/troubleshooting-athena.html#troubleshooting-athena-workgroups

[12] Using Athena ACID transactions. [Online] Available: https://docs.aws.amazon.com/athena/latest/ug/acid-transactions.html

[13] Data Architecture for AWS Athena. [Online] Available: https://www.upsolver.com/blog/data-architecture-aws-athena-examples

[14] Amazon Athena Architecture, Why Athena with QuickSight. [Online] Available: https://www.xenonstack.com/blog/amazon-athena-quicksight

[15] Invoking Machine Learning Models with Amazon Athena using SQL Queries, Machine Learning with Amazon Sage Maker Cookbook. [Online]. Available: https://subscription.packtpub.com/book/data/9781800567030/4/ch04lvl1sec43/invoking-machine-learning-models-with-amazon-athena-using-sql-queries

[16] Work with Amazon Athena Data in Apache Spark Using SQL. [Online] Available: https://www.cdata.com/kb/tech/athena-jdbc-apache-spark.rst

[17] Amazon Athena for Apache Spark. [Online] Available: https://aws.amazon.com/athena/spark/

[18] Blokdyk Gerardus, Amazon Athena a Clear and Concise Reference.

[19] Haritha Chatradipalli, Blog Amazon Athena.

[20] Integration with AWS Glue. [Online] Available: https://docs.aws.amazon.com/athena/latest/ug/glue-athena.html

[21] AWS Athena and Glue: Querying S3 data. [Online] Available: https://towardsdatascience.com/aws-athena-and-glue-querying-s3-data-ce83f1ba9f9f

[22] Building a DynamoDB to Athena Data Pipeline with AWS Glue and CDK. [Online] Available: https://docs.getcommandeer.com/docs/Glue/building-a-dynamodb-to-athena-data-pipeline-with-aws-glue-and-cdk/#the-problem

[23] Athena SQL basics – How to Write SQL against Files. [Online] Available: https://www.obstkel.com/amazon-athena-sql

[24] Amazon Athena Connection with Tableau [Online] Available: https://help.tableau.com/current/pro/desktop/en-us/examples_amazonathena.htm

[25] Configure and Optimize Performance of Amazon Athena Federation with Amazon Redshift. [Online] Available: https://aws.amazon.com/blogs/big-data/configure-and-optimize-performance-of-amazon-athena-federation-with-amazon-redshift/