VARADA

Big Data Indexing Dramatically Accelerates AWS Athena Performance

By Ori Reshef, VP Products | Varada

Say goodbye to “out of memory” queries, slow performance and inconsistent SLAs. Varda’s data lake query engine runs on your data lake and in your VPC, just like Athena, but delivers dramatically faster queries at a fraction of the cost.

AWS Athena is a Critical Building Block in Any Modern Data Lake Architecture

As an AWS native implementation of Presto, Athena offers both the Presto query engine's versatility and tight integration with other AWS services.

Athena lets data engineering teams create direct access to their AWS data sources such as S3, feed data into other AWS services, as well as share and control ad-hoc access to AWS data.

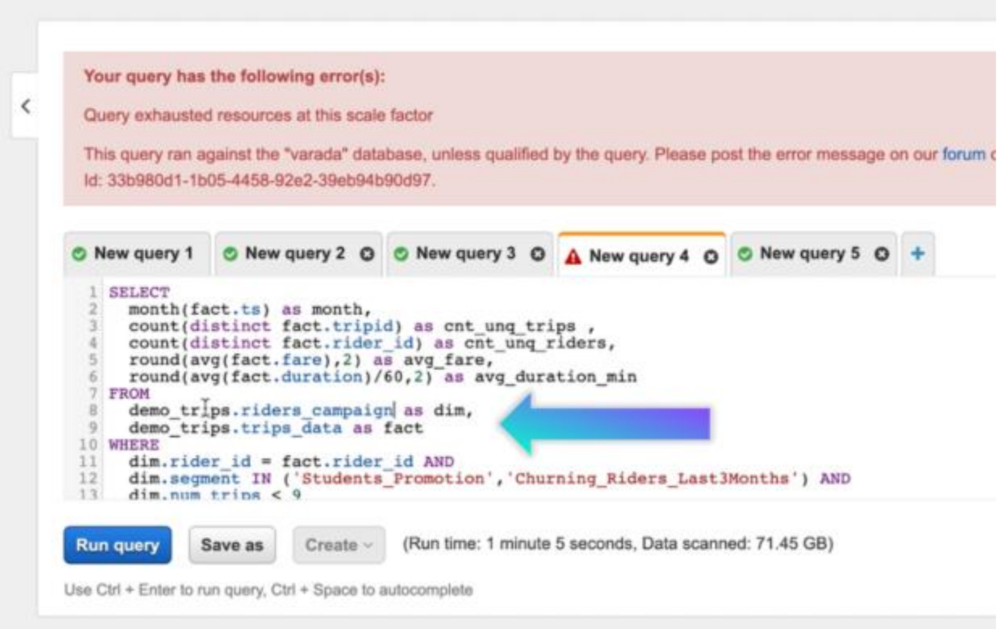
Users can build reporting dashboards against original datasets via Athena's standard Presto endpoint rather than complex and brittle ETL pipelines. Athena also gives administrators centralized control over ad-hoc access to data.

As a serverless Presto-based query engine, Athena offers a wide range of capabilities. It runs directly on an AWS-based data lake, with no data duplication, data movement, and data consistency issues.

As a result, the operational overhead of Athena is incredibly low. What drives users to augment Athena, is lack of control over performance and costs.

Introducing a data warehouse instantly voids all these benefits. Suddenly, data engineering teams need to deal with data migration, consistency, multiple permission models, and users struggling with finding data across multiple data catalogs.

As Athena administrators know, Athena works exceptionally well out of the box until users run into performance issues. Even though core Presto has powerful tools for optimization, a serverless and zero DevOps solution such as Athena doesn't include any tooling for analyzing performance issues.



Your query has the following error(s):

Query exhausted resources at this scale factor

This query ran against the "varada" database, unless qualified by the query. Please post the error message on our forum
Id: 33b980d1-1b05-4458-92e2-39eb94b90d97.

```
SELECT
  month(fact.ts) as month,
  count(distinct fact.tripid) as cnt_unq_trips ,
  count(distinct fact.rider_id) as cnt_unq_riders,
  round(avg(fact.fare),2) as avg_fare,
  round(avg(fact.duration)/60,2) as avg_duration_min
FROM
  demo_trips.riders_campaign as dim,
  demo_trips.trips_data as fact
WHERE
  dim.rider_id = fact.rider_id AND
  dim.segment IN ('Students_Promotion','Churning_Riders_Last3Months') AND
  dim.num_trips < 9
```

(Run time: 1 minute 5 seconds, Data scanned: 71.45 GB)

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

***AWS Athena query fails** because the JOIN was not optimized.*

As a result, when an Athena deployment gains adoption in an organization, users run into roadblocks trying to productionalize the system. For example, some queries incur expensive and time-consuming scans, which means users can't reliably power real time dashboards. Users also run into issues getting predictable query times when issuing complicated joins.

Athena administrators find they need domain expertise to understand the data that users are querying and to optimize users' workflow.

Administrators also struggle to help users because there is no good way to analyze what Athena is doing under the covers. Since data sets and use cases change quickly, any hard-earned gains through manual optimization goes out the window.

Therefore, there are practical limits on how broadly users can adopt Athena for access to data in the data lake. While the benefits of using Athena still outweigh the limitations, these issues are what ultimately lead organizations to abandon the pure data lake strategy and adopt a hybrid approach, duplicating data into a data warehouse.

Instead of being the unifying analytics platform, Athena and the data lake ends up being relegated to yet another data silo.

VARADA

Varada's data lake acceleration platform delivers dramatic query performance and resource utilization uplift, enabling data teams to reduce AWS Athena query cost while meeting business requirements.

Augmenting Data Lake Architecture To Address Performance and TCO Challenges

Varada delivers the new standard data lake analytics and enables data architects to seamlessly accelerate and optimize workloads, using dynamic analysis and adaptive indexing, resulting in optimal control over performance and cost.

Varada offers a Presto-based query engine that gives data lake administrators the power to optimize their Athena based data virtualization architecture.

Varada lets you run a full-scale production grade interactive analytics solution without needing to resort to an add-on data warehouse or hand optimize every query.

Best of all, Varada runs directly in your VPC, with an easy initial deployment through AWS Marketplace. Users can access everything in the data lake via Varada through the shared catalog using AWS Glue or the Hive metastore. Administrators simply need to make Varada available to users via a standard Presto endpoint.

Any SQL consumer to easily query any data source out-of-the-box without any need for optimizations or query re-writes.

Our secret sauce is our ability to automatically and dynamically index relevant data, at the structure and granularity of the source. Varada enables any query to meet continuously evolving performance and concurrency requirements for users and analytics API calls, while keeping costs predictable and under control.

Zero DataOps Solution

Varada automatically accelerates queries according to workload behavior and automatic detection of hot data and bottlenecks. The platform also enables data teams to define business priorities and accordingly adjust performance and budgets, eliminating the need to build separate silos for each use case.

The platform seamlessly chooses which queries to accelerate and which data to index. Varada elastically adjusts the cluster to meet demand and optimize cost and performance.

Leverage the Power of Big Data Indexing to Expand Data Lake Analytics

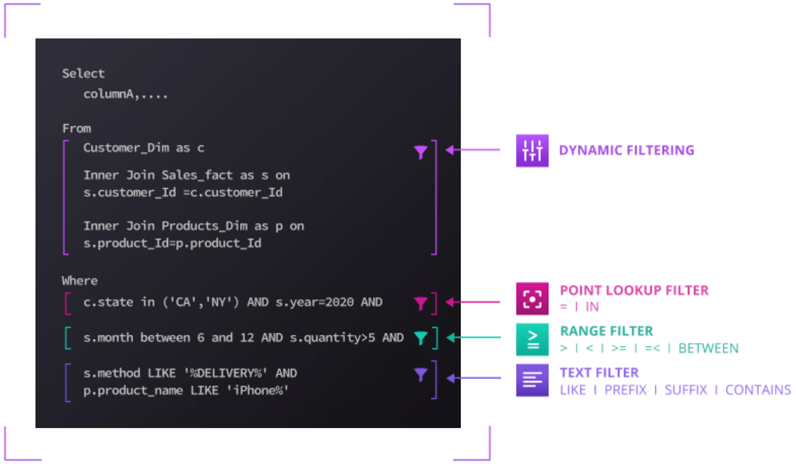
Varada's unique indexing efficiently indexes data directly from the data lake across selected columns so that every query is optimized automatically. Varada indexes adapt to changes in data over time, taking advantage of Presto SQL's vectorized columnar processing by splitting columns into small chunks, called nanoblocks™.

Based on the data type, structure, and distribution of data in each nanoblock, Varada automatically creates an optimal index. To ensure fast performance for every query and each nanoblock, Varada automatically selects from a set of indexing algorithms and indexing parameters that adapt and evolve as data changes to ensure best fit index any data nanoblock.

At query time when running through the Varada endpoint, users see transparent performance benefits when filtering, joining and aggregating data. Varada transparently applies indexes to any SQL WHERE clause, on indexed columns, within a SQL statement. Indexes are used for point lookups, range queries and string matching of data in nanoblocks. Varada automatically detects and uses indexes to accelerate JOINs using the index of the key column.

Varada indexes can be used for dimensional JOINs combining a fact table with a filtered dimension table, for self-joins of fact tables based on time or any other dimension as an ID, and for joins between indexed data and federated data sources. SQL aggregations and grouping is accelerated using nanoblock indexes as well, resulting in highly effective SQL analytics.

This example highlights the different techniques Varada leverages to optimize and accelerate SQL queries, including existing Presto queries:



```
Select
   columnA,....

From
   Customer_Dim as c

   Inner Join Sales_fact as s on
   s.customer_Id =c.customer_Id

   Inner Join Products_Dim as p on
   s.product_Id=p.product_Id

Where
   c.state in ('CA','NY') AND s.year=2020 AND
   s.month between 6 and 12 AND s.quantity>5 AND
   s.method LIKE '%DELIVERY%' AND
   p.product_name LIKE 'iPhone%'
```

Let’s Define “Fast”...

In this benchmarking analysis, we ran three queries on AWS Athena and compared them to a Varada cluster. We used different types of queries to illustrate the performance uplift data teams can expect over a wide range of workloads.

For this benchmarking analysis, we used data and queries based on a popular ride sharing application. Queries illustrate use cases used by marketing and product teams for segmentation, user behavior, etc.

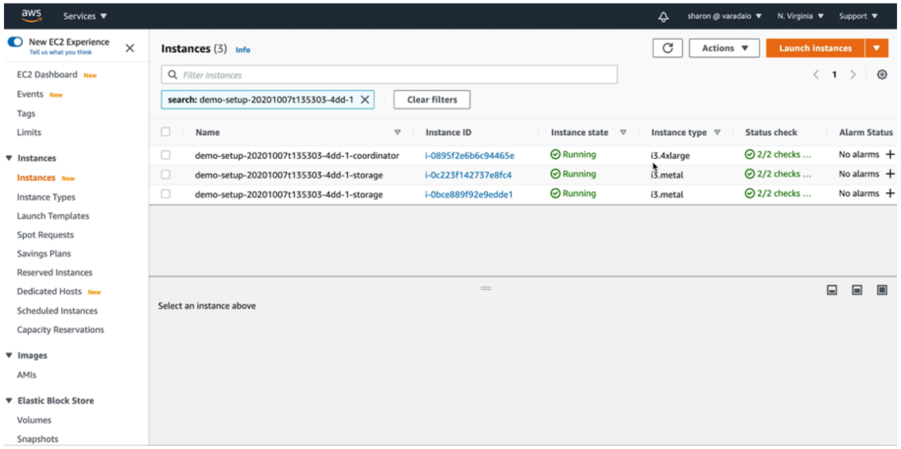
Running Queries Directly on the Data Lake

Queries are leveraging the data lake as the data source, accessing files in parquet format (S3) with snappy compression. The data set includes 27bn rows / 1.5TB (compressed).

Cluster Overview

AWS Athena is using 3 months of data, partitioned daily. As a serverless solution the actual size of the cluster and the type of EC2 machines used is not transparent to users.

Varada is using a 3-node cluster including two i3.metal workers and one i3.4xlarge coordinator.



search: demo-setup-20201007t135303-4dd-1

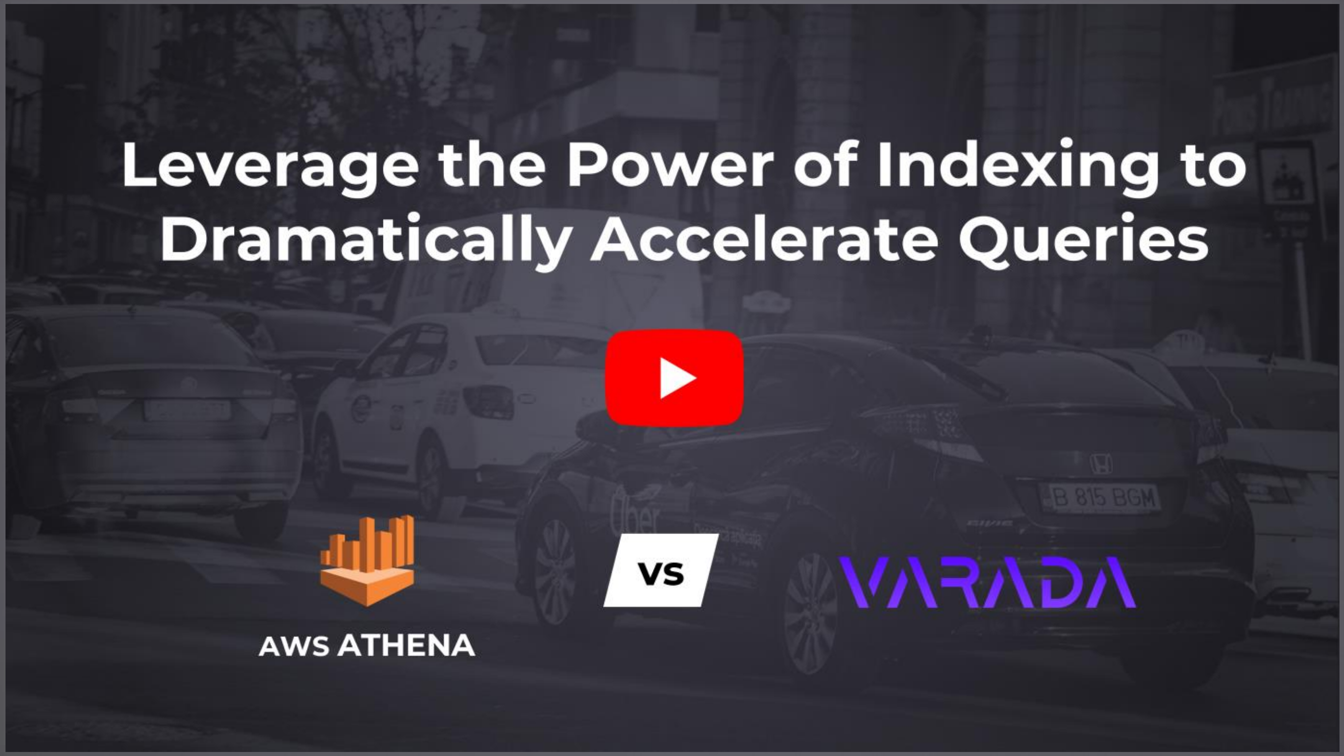
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm Status
□	demo-setup-20201007t135303-4dd-1-coordinator	i-0895f2e6b6c94465e	Running	i3.4xlarge	2/2 checks ...	No alarms +
□	demo-setup-20201007t135303-4dd-1-storage	i-0c223f142737e8fc4	Running	i3.metal	2/2 checks ...	No alarms +
□	demo-setup-20201007t135303-4dd-1-storage	i-0bce889f92e9edde1	Running	i3.metal	2/2 checks ...	No alarms +

Select an instance above

Varada Cluster on AWS

Varada runs directly on AWS data lake within the customer’s VPC so there is absolutely no modeling or partitioning and there is no need to move / duplicate data to an isolated environment.

Here are the results...


Leverage the Power of Indexing to Dramatically Accelerate Queries
AWS ATHENA
VS
VARADA

Benchmarking Results

Query	**AWS Athena**		**Varada**	
	Query Runtime	**Data Scanned**	**Query Runtime**	**Data Scanned**
Cohort Analysis Filter by: age group (18-19), day (Thursday), month (September)	11.39 secs	49.28GB ***Estimated cost: $0.25***	1.18 secs ***10x faster***	895MB ***98% decrease***
Highly Selective (needle in a haystack) Filter by: rider ID, at rush hour (7-10am)	41.43 secs	131.79GB ***Estimated cost: $0.66***	0.64 secs ***65x faster***	235KB ***99.9% decrease***
Join Acceleration (with star schema like model) **> Take #1** Filter by: marketing campaign and number of rides in the last 3 months. ** Tables were ordered correctly per Athena best practices*	103 secs	194.59GB ***Estimated cost: $0.97***	0.67 secs ***154x faster***	100KB ***99.9% decrease***
Join Acceleration (with star schema like model) **> Take #2** Filter by: marketing campaign and number of rides in the last 3 months. **** Tables were ordered so that the smaller table is first***	"out of resources"	NA	0.67 secs	100KB

But what happens if joins are not optimized for Athena

- Common mistake by inexperienced users
- Often occurs when queries are generated automatically by BI tools

Eliminate Manual Join Optimizations for SQL Order of Operations

Based on AWS Athena best practices, SQL optimization for joins requires to manually SQL order joins: larger table on the left side of join and the smaller table on the right side. Otherwise, queries often result in “timeouts”.

This is a common mistake by inexperienced users, and often occurs when queries are generated automatically by BI tools.

Varada leverages CBO and dynamic filtering to accelerate joins and integrates advanced CBO with tables statistics. Dynamic filtering accelerates joins by orders of magnitude. These advancements enable to avoid timeouts and manual query re-writes to support a wide range of SQL consumers.

Predictable Cost Structure for Queries on the Data Lake

As a serverless solution, AWS Athena delivers a true zero DevOps deployment, reducing all barriers to entry. But when more and more data consumers onboard the platform, the full-scan nature of query execution and the fact pricing is based solely on data scanner, often spiral out of control. This unpredictable and often very pricey service can prevent many business units from leveraging the benefits of the data lake architecture.

Varada introduces a very simple pricing model, which is based on the size of the cluster deployed. By introducing a smart big data indexing technology, **Varada eliminates the need for full scans in many queries, reducing data scanned from 100s of GBs to 100s of KBs.** Overall Varada can reduce total cost of ownership (TCO) by 40%-60%.