

Amazon Athena

Julien Simon

Principal Technical Evangelist

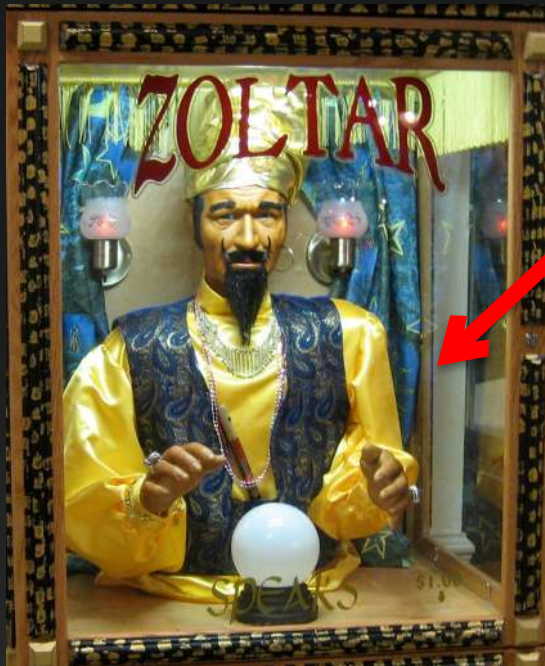
julsimon@amazon.fr

@julsimon



Big Data the way it should be

Questions
(not data!)



We shouldn't have to
care about how this
really works !



Answers

We shouldn't have to
mess with this at all



Data

Want to build it yourself? You need to master this

- Planning **capacity** for storage and compute
- Handling different **data formats**, structured and unstructured (CSV, JSON, Parquet, Avro, etc.)
- Learning complex **programming models** and **languages** (Map Reduce, Spark, Scala, etc.)
- Keeping **costs** under control
- **Availability**, **performance**, **security** and a few more



THERE IS ANOTHER

Amazon Athena

- New service announced at re:Invent 2016
- Run read-only SQL queries on S3 data
- No data load, no indexing, no nothing
- No infrastructure to create, manage or scale
- Availability: us-east-1, us-east-2, us-west-2
- Pricing: \$5 per Terabyte scanned

Athena queries

- Service based on **Presto** (already available in Amazon EMR)
- **Table creation**: Apache Hive Data Definition Language
 - CREATE EXTERNAL TABLE
- ANSI SQL **operators** and **functions**: what Presto supports
- **Unsupported operations**
 - User-defined functions (UDF or UDAFs)
 - Stored procedures
 - Any transaction found in Hive or Presto

Data formats supported by Athena

- **Unstructured**
 - Apache logs, with customizable regular expression
- **Semi-structured**
 - delimiter-separated values (CSV, OpenCSV)
 - Tab-separated values (TSV)
 - JSON
- **Structured**
 - Apache Parquet <https://parquet.apache.org/>
 - Apache ORC <https://orc.apache.org/>
 - Apache Avro <https://avro.apache.org/>
- **Compression** : Snappy, Zlib, GZIP, LZO
- **Partitioning**
- **Encryption** : integration with KMS



Demo #1

Athena vs Redshift: data set

Caveat: this isn't a huge data set and it doesn't have any joins

- 1 table
- 1 billion lines of “e-commerce sales” (43GB)
- CSV format, 10 columns
- 1000 files in S3, compressed to 12GB (bzip2)

```
Lastname, Firstname, Gender, State, Age, DayOfYear, Hour, Minutes, Items, Basket  
YESTRAMSKI, KEELEY, F, Missouri, 36, 35, 12, 21, 2, 167  
MAYOU, SCOTTIE, M, Arkansas, 85, 258, 11, 21, 9, 106  
PFARR, SIDNEY, M, Indiana, 59, 146, 22, 21, 3, 163  
RENZONI, ALLEN, M, Montana, 31, 227, 13, 49, 10, 106  
CUMMINS, NICKY, M, Tennessee, 50, 362, 1, 33, 1, 115  
THIMMESCH, BRIAN, M, Washington, 29, 302, 20, 41, 2, 95
```

Athena vs Redshift: table creation

```
CREATE EXTERNAL TABLE athenatest.sales (  
  lastname STRING,  
  firstname STRING,  
  gender STRING,  
  state STRING,  
  age INT,  
  day INT,  
  hour INT,  
  minutes INT,  
  items INT,  
  basket INT  
)  
ROW FORMAT SERDE  
'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'  
WITH SERDEPROPERTIES (  
  'serialization.format' = ',',  
  'field.delim' = ','  
) LOCATION 's3://jsimon-redshift-demo-us/data/'
```

```
CREATE TABLE sales(  
  lastname VARCHAR(32) NOT NULL,  
  firstname VARCHAR(32) NOT NULL,  
  gender VARCHAR(1) NOT NULL,  
  state VARCHAR(32) NOT NULL,  
  age INT NOT NULL,  
  day INT NOT NULL,  
  hour INT NOT NULL,  
  minutes INT NOT NULL,  
  items INT NOT NULL,  
  basket INT NOT NULL)  
DISTKEY(state)  
COMPOUND SORTKEY (lastname,firstname);  
  
COPY sales FROM 's3://jsimon-redshift-demo-us/data/'  
REGION 'us-east-1' CREDENTIALS ... DELIMITER ',' bzip2  
COMPUUPDATE ON;
```

Athena vs Redshift

- Redshift

- Start 4-node cluster: dc1.large, 160GB SSD, low I/O, \$0.25/hr
- Start 4-node cluster: dc1.8xlarge, 2.56TB SSD, v.high I/O, \$4.80/hr
- Create table
- Load data from S3 (COPY operation)
- Run some queries

- Athena

- Create table
- Run the same queries

Athena vs Redshift: start your engines!

- Athena
 - Initialization : < 5s (table creation)
 - Cost: \$0.0025 for a full scan (12GB + a few thousand S3 requests)
 - Unlimited storage in S3
- Redshift (dc1.large)
 - Initialization: 6mn (create cluster) + 40mn (data load)
 - \$1/hr (\$0.36 with 3-yr, 100% upfront RIs)
 - Maximum storage: 640GB (about 2TB with compression)
- Redshift (dc1.8xlarge)
 - Initialization: 6mn (create cluster) + 4mn (data load)
 - \$19.20/hr (\$6 with 3-yr, 100% upfront RIs)
 - Maximum storage: 10TB (about 30TB with compression)

Athena vs Redshift: SQL queries

Identical queries
on both systems

; Q1: count sales (1 full scan)

```
SELECT count(*) FROM sales
```

; Q2: average basket per gender (1 full scan)

```
SELECT gender, avg(basket) FROM sales GROUP BY gender;
```

; Q3: 5-day intervals when women spend most

```
SELECT floor(day/5.00)*5, sum(basket) AS spend FROM sales WHERE gender='F' GROUP  
BY floor(day/5.00)*5 ORDER BY spend DESC LIMIT 10;
```

; Q4: top 10 states where women spend most in December (1 full scan)

```
SELECT state, sum(basket) AS spend FROM sales WHERE gender='F' AND day>=334 GROUP  
BY state ORDER BY spend DESC LIMIT 10;
```

; Q5: list the top 10000 female customers in the top 10 states (2 full scans)

```
SELECT lastname, firstname, spend FROM (  
    SELECT lastname, firstname, sum(basket) AS spend FROM sales WHERE gender='F'  
    AND state IN(  
        SELECT state FROM sales WHERE day>=334 GROUP BY state  
        ORDER BY sum(basket) DESC LIMIT 10  
    ) AND day >=334 GROUP BY lastname,firstname  
) WHERE spend >=500 ORDER BY spend DESC LIMIT 10000;
```

Athena vs Redshift: SQL queries

YMMV, standard disclaimer applies ☺

; Q1: count sales

Athena: 15-17s, Redshift: 2-3s, Redshift 8xl: <1s

; Q2: average basket per gender

Athena: 20-22s, Redshift: 15-17s, Redshift 8xl: 4-5s

; Q3: 5-day intervals when women spend most

Athena: 20-22s, Redshift: 20-22s, Redshift 8xl: 4-5s

; Q4: top 10 states where women spend most in December

Athena: 22-25s, Redshift: 10-12s, Redshift 8xl: 2-3s

(courtesy of the 'state' distribution key)

; Q5: list the top 10000 female customers in the top 10 states

Athena: 38-40s, Redshift: 34-36s, Redshift 8xl: 7-9s



Demo #2

GDELT Data set

- Global Database of Events, Language and Tone Database
 - 300 categories of political & diplomatic activities around the world
 - Georeferenced to the city
 - Dating back to January 1, 1979
 - <http://www.gdeltproject.org/>
 - <http://blog.julien.org/2017/03/exploring-gdelt-data-set-with-amazon.html>
- 1582 CSV files in S3 (146 GB)
- 1 table (+ reference tables), 58 columns, 448M lines
- <https://aws.amazon.com/public-datasets/gdelt/>

Using columnar formats for fun and profit

- Hive makes it easy to convert from CSV to Parquet
<https://docs.aws.amazon.com/athena/latest/ug/convert-to-columnar.html>
- Large request
 - CSV uncompressed : 26 seconds, 136GB scanned, \$0.13
 - Parquet compressed : 4 seconds, 2.2GB scanned, \$0.002

Athena in a nutshell

- Run SQL queries on S3 data
- No infrastructure
- Multiple input formats supported
- Pretty fast!
- A simple, very cost-efficient option for ad-hoc analysis

Thank You

Julien Simon

julsimon@amazon.fr

@julsimon

**Your feedback
is important to us!**



Pop-up Loft
TEL AVIV

