ADVANCED
DATABASE
SYSTEMS

Amazon Redshift

22

Andy Pavlo
CMU 15-721
Spring 2024

Carnegie Mellon University

ADMINISTRIVIA

Project:

→ Final Presentations: **Thursday May 2nd @ 9:00am**

→ See Piazza@59 for more information.

Final Exam:

→ New Due Date: **Saturday May 4th @ 11:59pm**

→ Send to Andy via email

Cource Evaluation:

→ https://cmu.smartevals.com/

LAST CLASS

Yellowbrick (aka the unikernel DBMS)
→ An example of what you can do in a DBMS if you hate the OS with a passion and dream big…

HISTORICAL CONTEXT

Since the 2000s, there is a long history of people trying to make distributed versions of Postgres.

→ **OLTP**: Postgres-XC, StormDB, TransLattice, Postgres-XL

→ **OLAP**: Greenplum, Citus, Vertica, $\boxed{\text{ParAccel}}$

In 2010, Amazon was looking to add a new data warehouse service. Instead of writing it from scratch, they bought a license to ParAccel's source code as part of their investment.

AMAZON REDSHIFT (2014)

Amazon's flagship OLAP DBaaS.

- → Based on ParAccel's original shared-nothing architecture.
- → Switched to support disaggregated storage (S3) in 2017.
- → Added serverless deployments in 2022.

Redshift is a more traditional data warehouse compared to BigQuery/Spark where it wants to control all the data.

Overarching design goal is to remove as much administration + configuration choices from users.

AMAZON REDSHIFT RE-INVENTED
SIGMOD 2022

AWS OLAP DATABASE SYSTEMS

OG Redshift (2012)

→ Shared-nothing with managed storage that can spill to S3.

Athena (2016)

→ Serverless query engine for S3 data files based on Presto.

Redshift Spectrum (2017)

→ Extension to OG that supports querying S3 data files without having to first import them into managed storage.

REDSHIFT

Shared-Disk / Disaggregated Storage

Push-based Vectorized Query Processing

Transpilation Query Codegen (C++)

Precompiled Primitives

Compute-side Caching

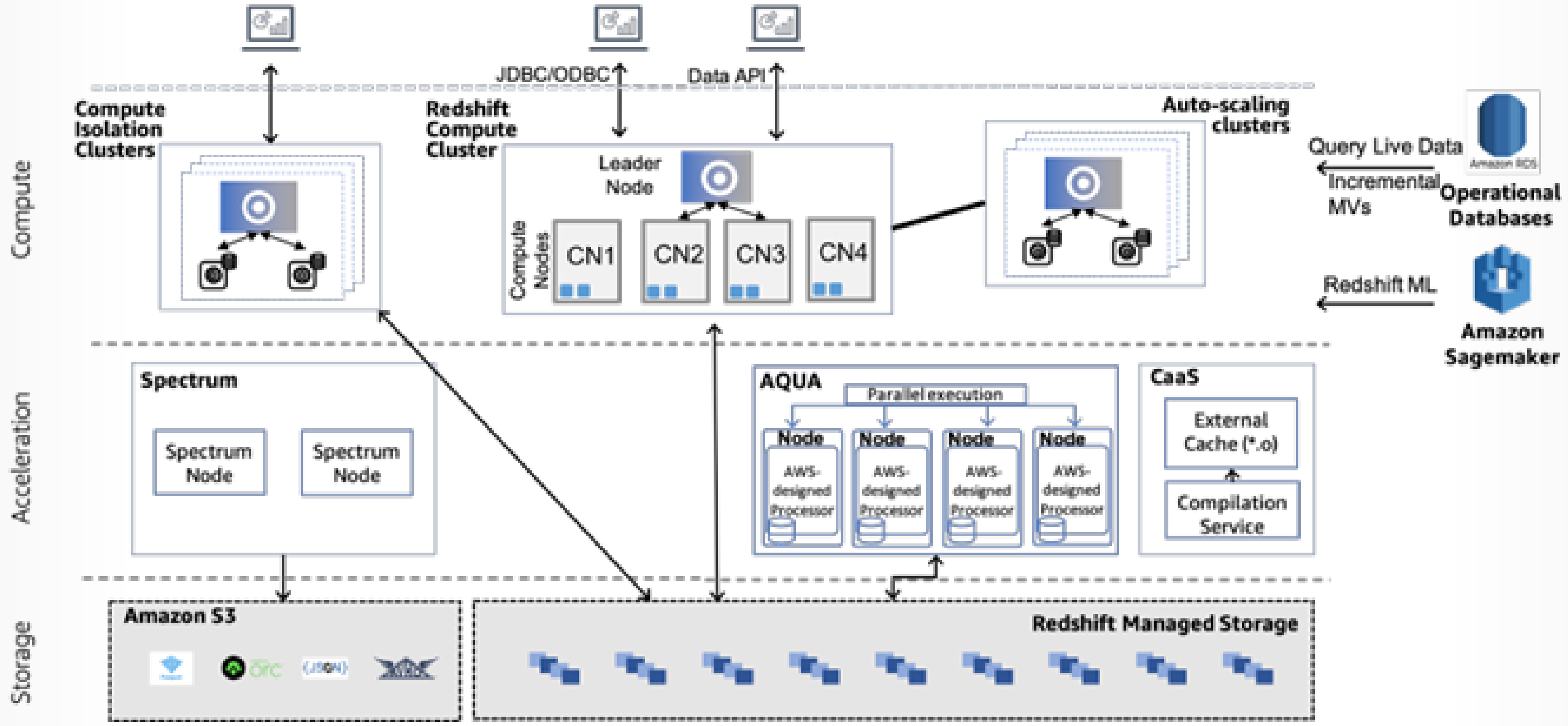
PAX Columnar Storage

Sort-Merge + Hash Joins

Hardware Acceleration (AQUA)

Stratified Query Optimizer

REDSHIFT: ARCHITECTURE



Source: Nikos Armenatzoglou, et al.

REDSHIFT: QUERY EXECUTION

Pushed-based vectorized query processing with late materialization. Hybrid query compilation using both source-to-source transpilation (C++) and precompiled primitives.
→ SIMD via AVX2 Intrinsics

Relies on software prefetching to get data into CPU cache inside of scan loops to avoid stalls.
→ Injects prefetching instructions into the generated code.

Less aggressive adaptivity than other systems:
→ SIMD fast paths for ASCII data.
→ Dynamic Bloom filter sizing for join filters.

REDSHIFT: COMPILATION SERVICE

Separate nodes to compile query plans using GCC and aggressive caching.

→ DBMS checks whether a compiled version of each templated fragment already exists in customer's local cache.
→ If fragment does not exist in the local cache, then it checks a global cache for the **entire** fleet of Redshift customers.

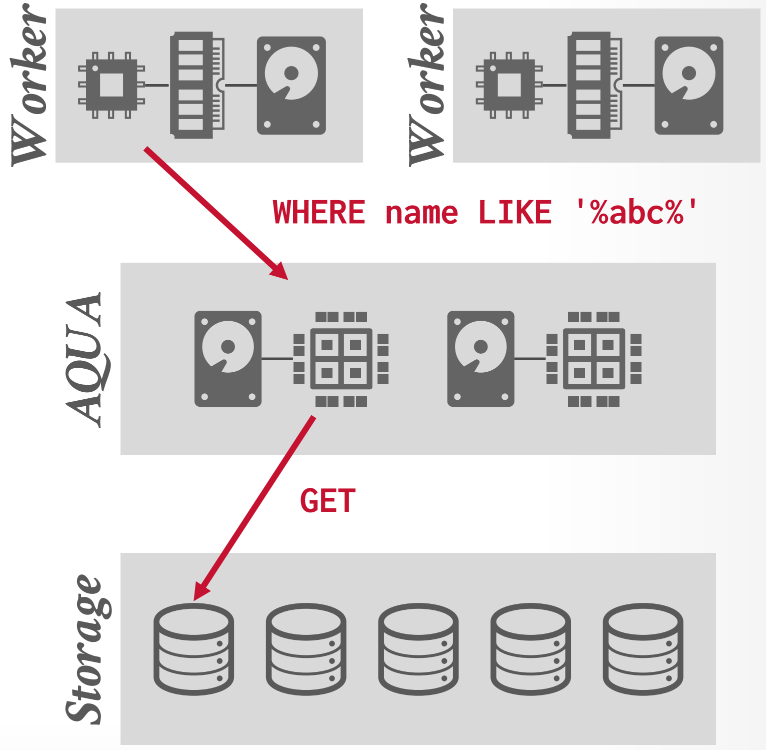
Background workers proactively recompile plans when new version of DBMS is released.

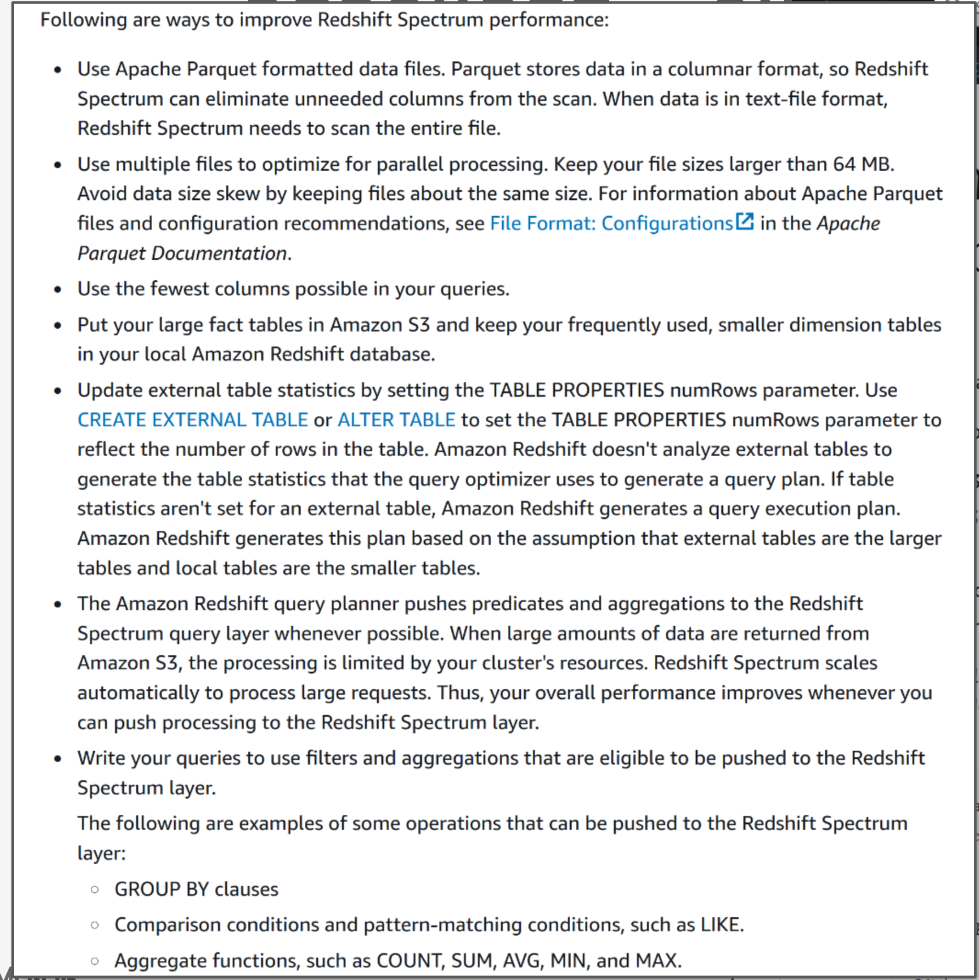
REDSHIFT: HARDWARE ACCELERATION

AWS introduced the **AQUA** (Advanced Query Accelerator) for Redshift (Spectrum?) in 2021.

Separate compute/cache nodes that use FPGAs to evaluate predicates.

AQUA was phased out and replaced with Nitro cards on compute nodes



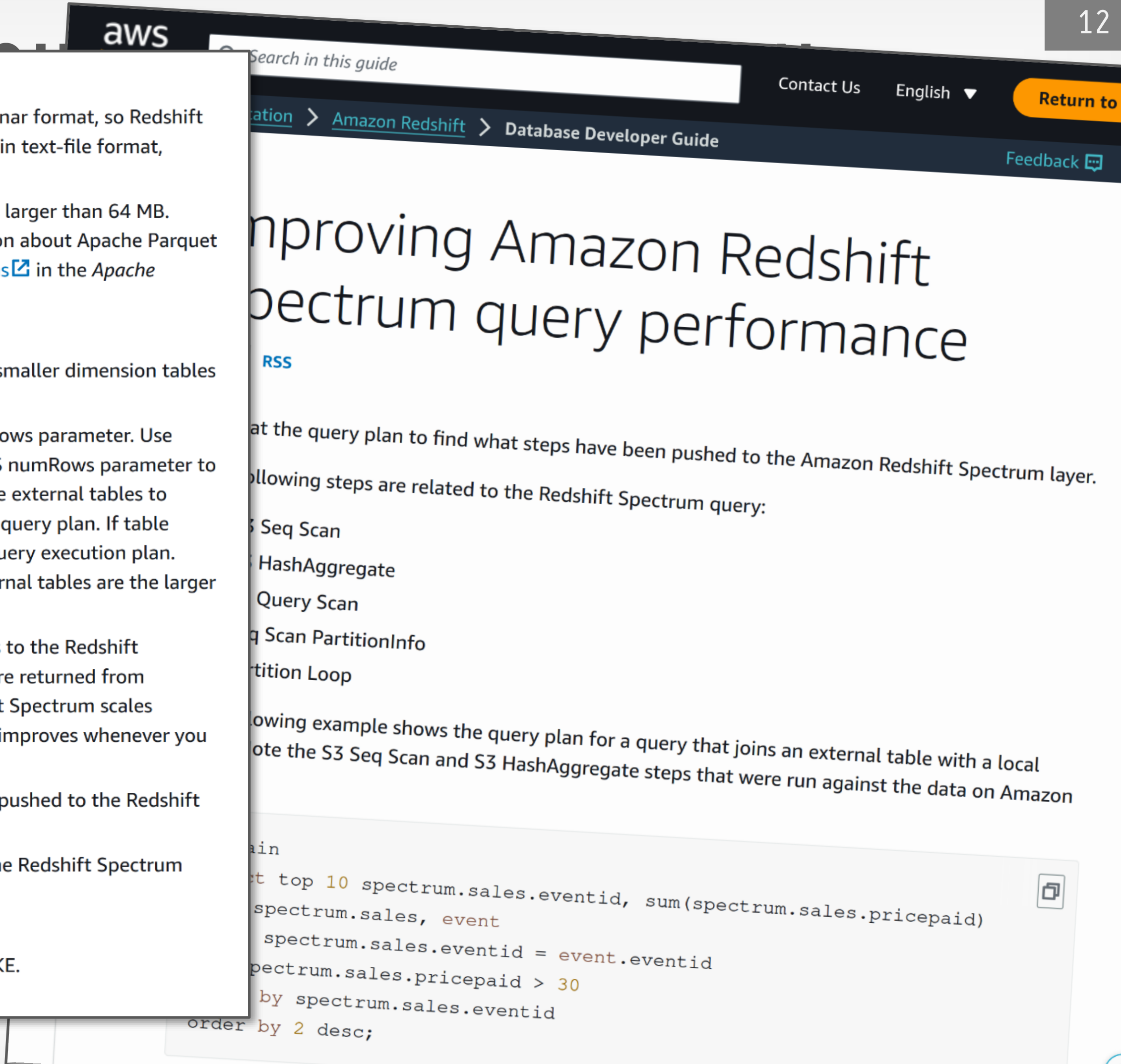


Following are ways to improve Redshift Spectrum performance:

- Use Apache Parquet formatted data files. Parquet stores data in a columnar format, so Redshift Spectrum can eliminate unneeded columns from the scan. When data is in text-file format, Redshift Spectrum needs to scan the entire file.
- Use multiple files to optimize for parallel processing. Keep your file sizes larger than 64 MB. Avoid data size skew by keeping files about the same size. For information about Apache Parquet files and configuration recommendations, see File Format: Configurations in the *Apache Parquet Documentation*.
- Use the fewest columns possible in your queries.
- Put your large fact tables in Amazon S3 and keep your frequently used, smaller dimension tables in your local Amazon Redshift database.
- Update external table statistics by setting the TABLE PROPERTIES numRows parameter. Use CREATE EXTERNAL TABLE or ALTER TABLE to set the TABLE PROPERTIES numRows parameter to reflect the number of rows in the table. Amazon Redshift doesn't analyze external tables to generate the table statistics that the query optimizer uses to generate a query plan. If table statistics aren't set for an external table, Amazon Redshift generates a query execution plan. Amazon Redshift generates this plan based on the assumption that external tables are the larger tables and local tables are the smaller tables.
- The Amazon Redshift query planner pushes predicates and aggregations to the Redshift Spectrum query layer whenever possible. When large amounts of data are returned from Amazon S3, the processing is limited by your cluster's resources. Redshift Spectrum scales automatically to process large requests. Thus, your overall performance improves whenever you can push processing to the Redshift Spectrum layer.
- Write your queries to use filters and aggregations that are eligible to be pushed to the Redshift Spectrum layer.

  The following are examples of some operations that can be pushed to the Redshift Spectrum layer:

  - GROUP BY clauses
  - Comparison conditions and pattern-matching conditions, such as LIKE.
  - Aggregate functions, such as COUNT, SUM, AVG, MIN, and MAX.



nproving Amazon Redshift pectrum query performance

at the query plan to find what steps have been pushed to the Amazon Redshift Spectrum layer.

llowing steps are related to the Redshift Spectrum query:

- Seq Scan
- HashAggregate
- Query Scan
- q Scan PartitionInfo
- rtition Loop

owing example shows the query plan for a query that joins an external table with a local ote the S3 Seq Scan and S3 HashAggregate steps that were run against the data on Amazon

```
ain
t top 10 spectrum.sales.eventid, sum(spectrum.sales.pricepaid)
spectrum.sales, event
 spectrum.sales.eventid = event.eventid
pectrum.sales.pricepaid > 30
 by spectrum.sales.eventid
order by 2 desc;
```

REDSHIFT: STORAGE

Redshift Managed Store (RMS) are separate storage nodes storing data in Amazon's proprietary file format on direct-attached SSDs.

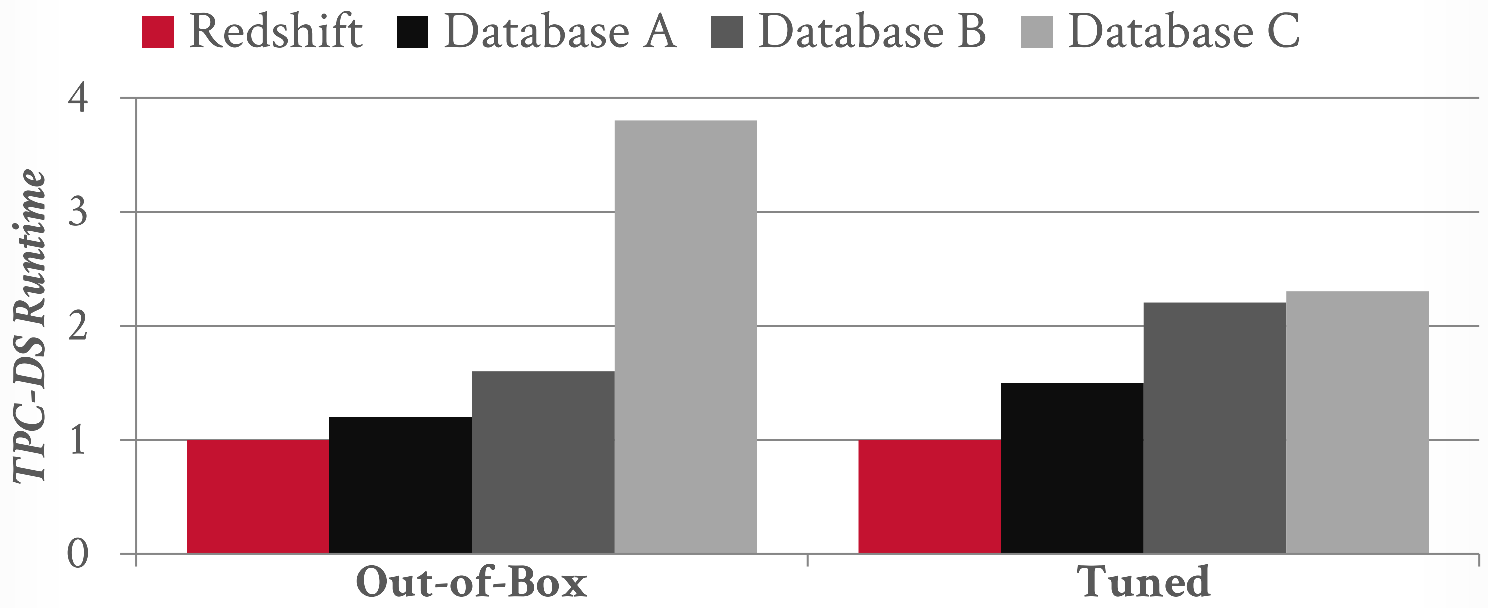
- → Compute nodes also maintain local SSD cache(?)
- → Can spill data to S3 if the local SSD gets too large.

When creating a table schema, users specify what encoding scheme to use per column

- → Dictionary, Delta, RLE, AZ64

BENCHMARK

Performance Relative to Redshift
TPC-DS (3TB)



AMAZON REDSHIFT RE-INVENTED
SIGMOD 2022

PARTING THOUGHTS

Redshift has evolved organically over the last decade based on the usage telemetry Amazon collects.

Amazon makes billions from Redshift each year.

ParAccel was acquired by Actian in 2013.
Rebranded as Actian Matrix in 2014.
But then they killed it in 2016.

NEXT CLASS

FINAL PROJECT PRESENTATIONS