

# The evolution of Amazon Redshift (extended abstract)

Ippokratis Pandis  
Amazon Web Services  
ippo@amazon.com

## PVLDB Reference Format:

Ippokratis Pandis . The evolution of Amazon Redshift (extended abstract).  
PVLDB, 14(12): 3162-3163, 2021.  
doi:10.14778/3476311.3476391

## ABSTRACT

In 2013, Amazon Web Services revolutionized the data warehousing industry by launching Amazon Redshift [7], the first fully managed, petabyte-scale enterprise-grade cloud data warehouse. Amazon Redshift made it simple and cost-effective to efficiently analyze large volumes of data using existing business intelligence tools. This launch was a significant leap from the traditional on-premise data warehousing solutions, which were expensive, not elastic, and required significant expertise to tune and operate. Customers embraced Amazon Redshift and it became the fastest growing service in AWS. Today, tens of thousands of customers use Amazon Redshift in AWS's global infrastructure of 25 launched Regions and 81 Availability Zones (AZs), to process exabytes of data daily.

The success of Amazon Redshift inspired a lot of innovation in the analytics segment, e.g. [1, 2, 4, 10], which in turn has benefited customers. In the last few years, the use cases for Amazon Redshift have evolved and in response, Amazon Redshift continues to deliver a series of innovations that delight customers.

In this paper, we give an overview of Amazon Redshift's system architecture. Amazon Redshift is a columnar MPP data warehouse [7]. As shown in Figure 1, an Amazon Redshift compute cluster consists of a coordinator node, called the *leader node*, and multiple *compute nodes*. Data is stored on *Redshift Managed Storage*, backed by Amazon S3, and cached in compute nodes on locally-attached SSDs in compressed columnar fashion. Tables are either replicated on every compute node or partitioned into multiple buckets that are distributed among all compute nodes. AQUA is a query acceleration layer that leverages FPGAs to improve performance. CaaS is a caching microservice of optimized generated code for the various query fragments executed in the Amazon Redshift fleet.

The innovation at Amazon Redshift continues at accelerated pace. Its development is centered around four streams. First, Amazon Redshift strives to provide industry-leading data warehousing performance. Amazon Redshift's query execution blends database operators in each query fragment via code generation. It combines prefetching and **vectorized execution with code generation** to achieve maximum efficiency. This allows Amazon Redshift to scale linearly when processing from a few terabytes to petabytes

of data. Figure 2 depicts the total execution time of the Cloud Data Warehouse Benchmark Derived from TPC-DS 2.13 [6] while scaling dataset size and hardware simultaneously. Amazon Redshift's performance remains nearly flat for a given ratio of data to hardware, as data volume increases from 30TB to 1PB. This linear scaling to the petabyte scale makes it easy, predictable and cost-efficient for customers to on-board new datasets and workloads.

Second, customers needed to process more data and wanted to support an increasing number of concurrent users or independent compute clusters that are operating over the Redshift-managed data and the data in Amazon S3. We present Redshift Managed Storage, Redshift's high-performance transactional storage layer, which is disaggregated from the Redshift compute layer and allows a single database to grow to tens of petabytes. We also describe Redshift's compute scaling capabilities. In particular, we present how Redshift can scale up by elastically resizing the size of each cluster, and how Redshift can scale out and increase its throughput via multi-cluster autoscaling, called Concurrency Scaling. With Concurrency Scaling, customers can have thousands of concurrent users executing queries on the same Amazon Redshift endpoint. We also talk about data sharing, which allows users to have multiple isolated compute clusters consume the same datasets in Redshift Managed Storage. Elastic resizing, concurrency scaling and data sharing can be combined giving multiple compute scaling options to the Amazon Redshift customers.

Third, as Amazon Redshift became the most widely used cloud data warehouse, its users wanted it to be even easier to use. For that, Redshift introduced ML-based autonomies. We present how Redshift automated among others workload management, physical tuning, the refresh of materialized views (MVs), along with automated MVs-based optimization that rewrites queries to use MVs. We also present how we leverage ML to improve the operational health of the service and deal with gray failures [8].

Finally, as AWS offers a wide range of purpose-built services, Amazon Redshift provides seamless integration with the AWS ecosystem and novel abilities in ingesting and ELTing semistructured data (e.g., JSON) using the PartiQL extension of SQL [9]. AWS purpose-built services include the Amazon S3 object storage, transactional databases (e.g., DynamoDB [5] and Aurora [11]) and the ML services of Amazon SageMaker. We present how AWS and Redshift make it easy for their customers to use the best service for each job and seamlessly take advantage of Redshift's best of class analytics capabilities. For example, we talk about Redshift Spectrum [3] that allows Redshift to query data in open-file formats in Amazon S3. We present how Redshift facilitates both the in-place querying of data in OLTP services, using Redshift's Federated Querying, as well as the copy of data to Redshift, using Glue Elastic Views. We also present how Redshift can leverage the capabilities of Amazon SageMaker through SQL and without data movement.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing [info@vldb.org](mailto:info@vldb.org). Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.  
Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.  
doi:10.14778/3476311.3476391

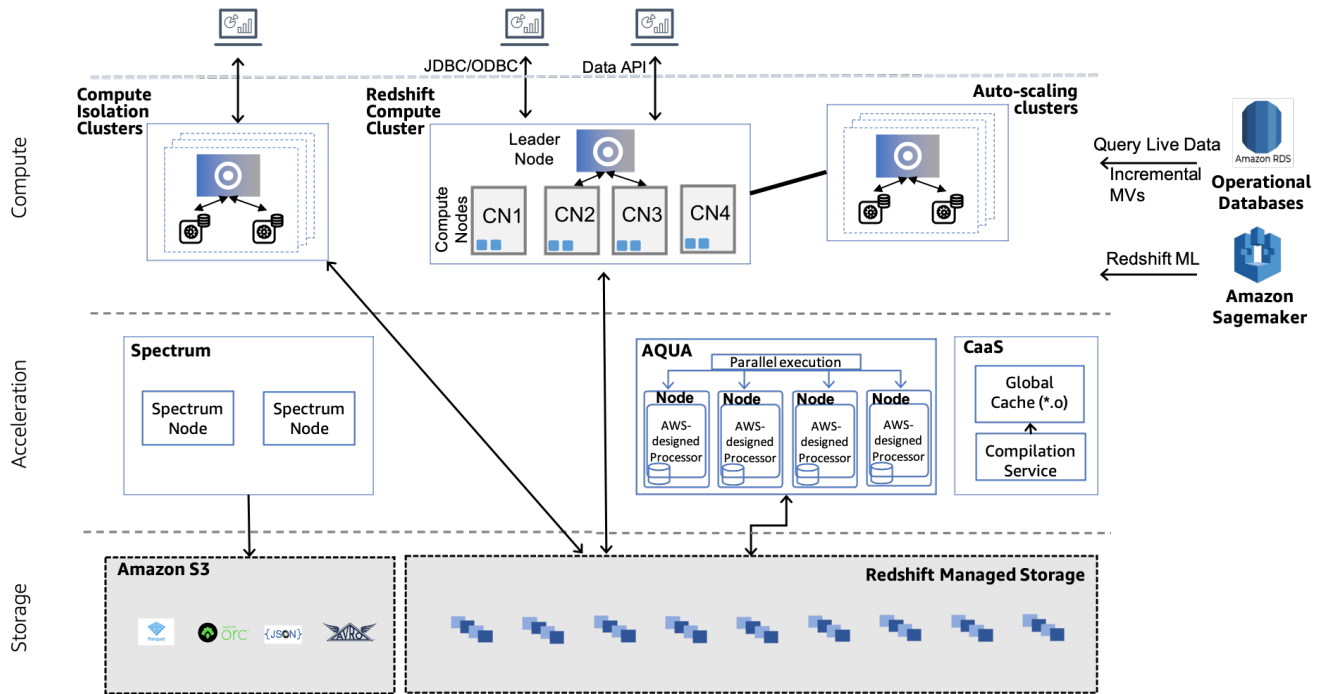


Figure 1: Amazon Redshift Architecture

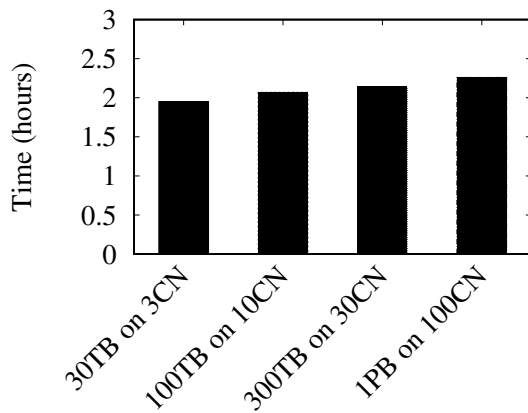


Figure 2: Scaling Cloud Data Warehouse Benchmark derived from TPC-DS from 30TB to 1PB

With a differentiating execution core, ability to scale to tens of PBs of storage and thousands of concurrent users, ML-based automations that make it easy to use, and tight integration with the wide AWS ecosystem, Amazon Redshift is a best-of-class solution for cloud data warehousing.

## REFERENCES

- [1] J. Aguilar-Saborit, R. Ramakrishnan, K. Srinivasan, K. Bocksrocker, I. Alagianis, M. Sankara, M. Shafiei, J. Blakeley, G. Dasarathy, S. Dash, L. Davidovic, M. Damjanic, S. Djunic, N. Djurkic, C. Feddersen, C. Galindo-Legaria, A. Halverson, M. Kovacevic, N. Kicovic, G. Lukic, D. Maksimovic, A. Manic, N. Markovic, B. Mihic, U. Milic, M. Milojevic, T. Nayak, M. Potocnik, M. Radic, B. Radivojevic, S. Rangarajan, M. Ruzic, M. Simic, M. Sosic, I. Stanko, M. Stikic, S. Stanojkov, V. Stefanovic, M. Sukovic, A. Tomic, D. Tomic, S. Toscano, D. Trifunovic, V. Vasic, T. Verona, A. Vujic, N. Vujic, M. Vukovic, and M. Zivanovic. POLARIS: The distributed SQL engine in Azure Synapse. *PVLDB*, 13(12), 2020.
- [2] M. Armbrust, T. Das, L. Sun, B. Yavuz, S. Zhu, M. Murthy, J. Torres, H. van Hovell, A. Ionescu, A. Luszczyk, M. Switakowski, M. Szafranski, X. Li, T. Ueshin, M. Mokhtar, P. Boncz, A. Ghodsi, S. Paranjpye, P. Senster, R. Xin, and M. Zaharia. Delta Lake: High-performance ACID table storage over cloud object stores. *PVLDB*, 13(12), 2020.
- [3] M. Cai, M. Grund, A. Gupta, F. Nagel, I. Pandis, Y. Papakonstantinou, and M. Petropoulos. Integrated querying of SQL database data and S3 data in Amazon Redshift. *IEEE Data Eng. Bull.*, 41(2), 2018.
- [4] B. Dageville, T. Cruanes, M. Zukowski, V. Antonov, A. Avanes, J. Bock, J. Claybaugh, D. Engovatov, M. Hentschel, J. Huang, A. W. Lee, A. Motivala, A. Q. Munir, S. Pelley, P. Povinec, G. Rahn, S. Triantafyllis, and P. Unterbrunner. The Snowflake Elastic Data Warehouse. In *SIGMOD*, 2016.
- [5] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's highly available key-value store. In *SOSP*, 2007.
- [6] S. Gromoll et al. Cloud data warehouse benchmark derived from TPC-DS 2.13. Technical report, Amazon, 2019. <https://github.com/aws-labs/amazon-redshift-utils/tree/master/src/CloudDataWarehousebenchmark/Cloud-DWB-Derived-from-TPCDS>.
- [7] A. Gupta, D. Agarwal, D. Tan, J. Kulesza, R. Pathak, S. Stefani, and V. Srinivasan. Amazon Redshift and the case for simpler data warehouses. In *SIGMOD*, 2015.
- [8] P. Huang, C. Guo, L. Zhou, J. R. Lorch, Y. Dang, M. Chintalapati, and R. Yao. Gray failure: The Achilles' Heel of cloud-scale systems. In *HotOS*, 2017.
- [9] Y. Papakonstantinou, A. Goo, B. Ruppert, J. Wilsdon, and P. Varakur. Announcing PartiQL: One query language for all your data. Technical report, Amazon, 2019. <https://aws.amazon.com/blogs/opensource/announcing-partiql-one-query-language-for-all-your-data/>.
- [10] K. Sato. An inside look at Google BigQuery. Technical report, Google. <https://cloud.google.com/files/BigQueryTechnicalWP.pdf>.
- [11] A. Verbitski, A. Gupta, D. Saha, M. Brahmadesam, K. Gupta, R. Mittal, S. Krishnamurthy, S. Maurice, T. Kharatishvili, and X. Bao. Amazon Aurora: Design considerations for high throughput cloud-native relational databases. In *SIGMOD*, 2017.