ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 3, March 2018

Big Data Using Amazon Redshift

[1] Vinod Alone, [2] Atul Shintre, [3] Manish Gangawane, [4] Srikant Bagewadi, [5] Harshal Patil
[1] [2] [3] [4] Assistant Professor, Department of Computer Engineering, Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai, India.
[5] Institute for Technology and Management,Nerul,Navi Mumbai,India.

Abstract: - **Amazon Redshift is an Internet hosting service and data warehouse product which forms part of the larger cloud-computing platform Amazon Web Services. It is built on top of technology from the massive parallel processing (MPP) data-warehouse company ParAccel (later acquired by Actian). Redshift differs from Amazon's other hosted database offering, Amazon RDS, in its ability to handle analytics workloads on big data data sets stored by a column-oriented DBMS principle. To be able to handle large scale data sets and database migrations Amazon makes use of massive parallel processing. [1]**

Index Terms—**Big data; Amazon Services; Redshift; MPP**

I. INTRODUCTION

Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools. It allows you to run complex analytic queries against petabytes of structured data, using sophisticated query optimization, columnar storage on high-performance local disks, and massively parallel query execution. Most results come back in seconds. Amazon Redshift also includes Redshift Spectrum, allowing you to directly run SQL queries against exabytes of unstructured data in Amazon S3. No loading or transformation is required, and you can use open data formats, including Avro, CSV, Grok, ORC, Parquet, RCFile, RegexSerDe, SequenceFile, TextFile, and TSV. Redshift Spectrum automatically scales query compute capacity based on the data being retrieved, so queries against Amazon S3 run fast, regardless of data set size.[2] Amazon Redshift delivers fast query performance by using columnar storage technology to improve I/O efficiency and by parallelizing queries across multiple nodes. Data load speed scales linearly with cluster size, with integrations to other Amazon services. Redshift Spectrum enables you to run queries against exabytes of data in Amazon S3 as easily as you run queries against petabytes of data stored on local disks in Amazon Redshift, using the same SQL syntax and BI tools you use today. You can store highly structured, frequently accessed data on Redshift local disks, keep vast amounts of unstructured data in an Amazon S3 ―data lake‖, and query seamlessly across both.

II. OPTIMIZED FOR DATA WAREHOUSING

Amazon Redshift uses a variety of innovations to obtain very high query performance on datasets ranging in size from a hundred gigabytes to an exabyte or more. For petabyte-scale local data, it uses columnar storage, data compression, and zone maps to reduce the amount of I/O needed to perform queries. Amazon Redshift has a massively parallel processing(MPP) data warehouse architecture, parallelizing and distributing SQL operations to take advantage of all available resources. The underlying hardware is designed for high performance data processing, using local attached storage to maximize throughput between the CPUs and drives, and a 10GigE mesh network to maximize throughput between nodes. For exabyte-scale data in Amazon S3, Amazon Redshift generates an optimal query plan that minimizes the amount of data scanned and delegates the query execution to a pool of Redshift Spectrum instances that scales automatically, so queries run quickly regardless of data size [3].

III. PETABYTE SCALE

With a few clicks in console or a simple API call, you can easily change the number or type of nodes in your data warehouse and scale up all the way to a petabyte or more of compressed user data. Dense Storage (DS) nodes allow you to create very large data warehouses using hard disk drives (HDDs) for a very low price point. Dense Compute (DC) nodes allow you to create very high performance data warehouses using fast CPUs, large amounts of RAM and solid-state disks (SSDs). While resizing, Amazon Redshift allows you to continue to query your data warehouse in read-only mode until the new cluster is fully provisioned and ready for use.

IV. FAULT TOLERANT

Amazon Redshift has multiple features that enhance the reliability of your data warehouse cluster. All data written to a node in your cluster is automatically replicated to other nodes within the cluster and all data is continuously backed up to Amazon S3. Amazon Redshift continuously monitors the health of the cluster and automatically re-replicates data from failed drives and replaces nodes as necessary.

V. NETWORK ISOLATION

Amazon Redshift enables you to configure firewall rules to control network access to your data warehouse cluster. You can run Amazon Redshift inside Amazon VPC to isolate your data warehouse cluster in your own virtual network and connect it to your existing IT infrastructure using industry-standard encrypted IPsec VPN.

VI. AUDIT AND COMPLIANCE

Amazon Redshift integrates with AWS CloudTrail to enable you to audit all Redshift API calls. Amazon Redshift also logs all SQL operations, including connection attempts, queries and changes to your database. You can access these logs using SQL queries against system tables or choose to have them downloaded to a secure location on Amazon

VII. WORKING MODEL

A successful customer 360 view benefits from using a variety of technologies to deliver different forms of insights. These could range from real-time analysis of streaming data from wearable devices and mobile interactions to historical analysis that requires interactive, on demand queries on billions of transactions. In some cases, insights can only be inferred through AI via deep learning. Finally, the value of your customer data and insights can‘t be fully realized until it is operationalized at scale—readily accessible by fleets of applications. Companies are leveraging AWS for the breadth of services that cover these domains, to drive their data strategy. A number of AWS customers stream data from various sources into a S3 data lake through Amazon Kinesis. They use Kinesis and technologies in the Hadoop ecosystem like Spark running on Amazon EMR to enrich this data. High-value data is loaded into an Amazon Redshift data warehouse, which allows users to analyze and interact with data through a choice of client tools. Redshift Spectrum expands on this analytics platform by enabling Amazon Redshift to blend and analyze data beyond the data warehouse and across a data lake[4]. The following diagram illustrates the workflow for such a solution.

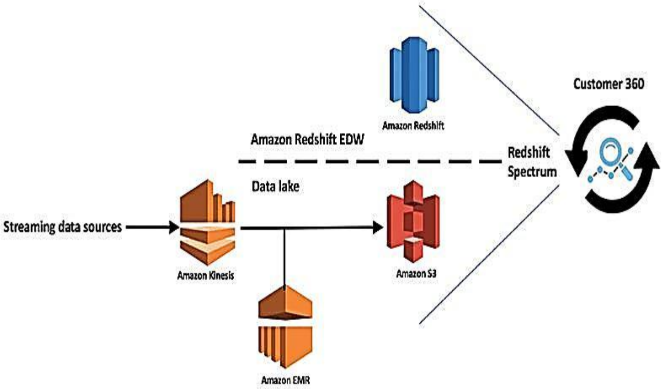

Fig: Working model of Redshift in AWS

Cluster: the cluster is the overall instance or configuration of Redshift. You cannot have just a Redshift server, only a Redshift cluster. A cluster is made up of a leader node and one or more nodes. At cluster level you decide the storage technology for your Redshift implementation – SSD or magnetic.

Leader Node: The leader node is what you and your application treat as the database. It is the front-end facade to the complexity of the Redshift cluster behind the scenes. Behind the leader node sit one or more nodes that actually do the work – but what your application sees is just one, nice, simple leader node. You can only have one leader node per Redshift cluster. It looks, to the outside world, like a Postgres database.

Nodes: Data is distributed across nodes, and an individual node is roughly analogous to a virtual machine. How data is distributed across the nodes depends on the schema design. A lot of the best practices surrounding redshift are about getting the right data into the right nodes, for optimum performance. In layman‘s terms, the more nodes you have, the more potentially powerful your Redshift cluster is. However, if you don‘t follow the best practices properly, additional nodes can yield negligible (or in fact negative) real world performance and scalability gains. Obviously, it is worth bearing in mind that the more nodes your Redshift cluster has, the more expensive it is to run. However, you can re-size the cluster at any time (as long as you don‘t reduce the cluster size below the amount of storage that you need.)

Slices: The type of node that you select governs the number of slices that each node has. A slice is roughly analogous to a processor (or core) allocated to working on the data stored on that node. The correct use of slices allows a node to make use of its multiple cores. When the cluster allocates work to a node, the node can further split this work down across its available cores/CPUs, assuming that the data is structured in a way so that it can be practically and efficiently split up.

VIII. STORAGE IN REDSHIFT

If you were implementing a data warehouse in a traditional relational database technology – MS SQL Server or MySQL, for example – then you would design a star-schema, fill it with data, and then index the fields that users want to filter, group by, and use. But because you don't know in advance what fields those are (users have an annoying habit of wanting to use the one you hadn't thought of), then you end up indexing everything. At which point, you have got two full copies of your data: one in the main tables, and one in the indices. And so, columnar data stores (like Redshift, Vertica, Netezza and Teradata), at a very simplified level, admit in advance to themselves that every column is going to need an index, and thereby do away with the main data store. They are, in effect, therefore, just an index for every column [5]. As a by-product of this indexing, all the values of the same type and those which have similar values, are organised next to each other in the indices. As such, compression in columnar data stores is far more efficient than in traditional RDBMs. This is all relevant to your use of best practices in Redshift. For instance, when you are defining a table in Redshift, you may, if you wish, choose from one of 11 different compression strategies (Redshift calls them _column encodings') for each column in your table. Picking the right one will impact your storage and as a consequence, can also impact performance. Or, if you follow other best practices, you can leave Redshift to pick the correct one for you.

IX. ADVANTAGES

a. Incredibly fast:

One of the most obvious Amazon Redshift benefits is that it is fast – incredibly fast in fact. The key to this speed lies in its ability to leverage the Massively Parallel Processing (MPP) capabilities of its data warehouse architecture. By distributing the workload across multiple nodes, this takes advantage of all available resources. Working in parallel, this optimises query performance, even when dealing with petabyte scale datasets.

b. Easily scalable

The flexibility and in particular, scalability, of Amazon Redshift makes it appealing to businesses of all shapes and sizes.

With a few simple clicks, you can easily scale the number or type of nodes in your Redshift data warehouse to suit your capacity requirements. Amazon Redshift allows you to scale from single 160GB nodes all the way up to 16TB nodes, allowing you to create a petabyte scale data warehouse, without any loss in performance.

C. Low cost

One of the most appealing Amazon Redshift benefits is, quite understandably, the price. The scalability of Amazon Redshift makes it an increasingly cost-effective alternative to traditional data warehousing practices. The on-demand pricing structure means you only pay for the resources you provision. This on-demand pricing starts at as low as $0.25 per hour for a 160GB DC1.Large node or $0.85 per hour for a larger 2TB version [6].

X. CONCLUSION

Amazon Redshift, here is the conclusion and a logical view-we can go with this fantastic data web services from Amazon which offers several demerits but it has more pros which expound that it will be beneficial for your business to make it more challenging and competitive in the world of modern technology. The usability of this service will definitely boom your company daily operations and save your time to deal with all the Computer system integration with cloud services. It will also lead to changing the era of not using software and hardware from your system where as using the system directly From the cloud. The best part of this amazing Redshift is its technological process results into saving money and time.

REFERENCES

[1] Amazon Web Services (AWS) - Cloud Computing Services https://aws.amazon.com/ Open Stack. https://www.openstack.org/
[2] Amazon Redshift Developer Guide by Amazon
[3] ―Stefan Bauer‖ Getting Started with Amazon Redshift
[4] Andreas Wittig ―Amazon Web Services in Action By
[5] Joe Baron ―AWS Certified Solutions Architect Official Study Guide‖
[6] Intragation of AWS redshift with Web App By Amazon
[7] Intragation of AWS redshift with Web App By Amazon