**Data Warehousing using AWS Redshift**

AWS Redshift is an offering of relational database management system (RDBMS) which is compatible with many RDBMS applications. It gives functionality of a traditional RDBMS, like online transaction processing (OLTP) functions but along with the USP of optimized analysis and reporting which leads to high-performance, it is based on the infamous PostgreSQL.

* **Key Components:**
* **Client Applications**

Client applications like data loading and ETL tools(extract, transform, and load), business intelligence (BI) reporting tools etc can be easily integrated with Redshift as it is based on PostgreSQL which is industry standard.

* **Connections**

PostgreSQL JDBC, ODBC drivers are used by Redshift to communicate with client applications.

* **Clusters**

It is the core component of Redshift data warehouse. It is composed of compute nodes and additional leader node to coordinate the compute nodes, and do all sorts of external communication with any client application.

* **Leader Node**

This node is responsible for creating execution plans to do database operations. It is according to this execution plan that the code is compiled, distributed among the compute nodes.

* **Compute Nodes**

These nodes are responsible for executing the compiled code and sending the intermediate calculations back. Each of these have its own CPU, storage and memory as per selected node type.

* **Node slices**

Every compute node will be partitioned into slices corresponding to core processors, memory and disk space. Using these resources, the slice completes the assigned computation. As there are multiple slices so they work in parallel to achieve concurrency. One can specify a column as distribution key according which the rows are distributed among slices so selection of distribution key plays an eminent role for balanced distribution and minimized movement.

* **Internal Network**

Amazon Redshift takes good advantage of high-bandwidth, less proximity, custom protocols to communicate which provides private, high-speed network between the compute nodes and leader node. The compute nodes always run on an isolated and separate network which is never accessed directly by the client applications.

* **Databases**

Any cluster will be containing one or more database instances. The user data is stored on compute nodes ad the SQL client interacts with leader node, which coordinates query execution on compute nodes.

* **Performance Features:**

**• Massive parallel processing**

It allowsmore fast execution of the highly complex queries which are operating on huge amounts of data. All the query processing is handled by several compute nodes, finally delivering the aggregated results.

**• Columnar type data storage**

It drastically reduces the disk I/O requirements so it is an important factor which optimizes the performance of analytical queries. This way less data has to be loaded into memory which allows Redshift to do more in-memory operations.

**• Data compression**

It reduces the storage requirements, hence reducing disk I/O, which then improves the overall performance of the query. On execution of any query, the compressed data is loaded into memory and then uncompressed during execution. Redshift applies optimal compression techniques for encoding whenever the data is loaded into table.

**• Query optimization**

Amazon Redshift has a query execution engine which incorporates an optimizer for queries. It implements several enhancements like ensuring columnar data storage, subqueries, multiple table joins and aggregation.

**• Compiled code**

It is the responsibility of the leader node to distribute optimized compiled code across nodes in the cluster. This compiled code is put in cache and subsequently shared with client applications in case of complex queries.

* **Work- Load Management**

Amazon Redshift workload management (WLM) is an important feature which helps the users in flexibly managing the priorities within workloads, hence ensuring that short queries are never stuck behind long queries. This brings down the overall waiting period.

By default, Amazon Redshift has single queue with a concurrency of five, hence enabling upto 5 queries to run at a time, along with a predefined queue, with a concurrency of 1. Upto 8 queues can be defined and each queue can have maximum concurrency of 50. The maximum total concurrency level for all user-defined queues is 50 and it can be modified using console or API.

* **Common use case scenarios:**
* Moving the data between Amazon Redshift and Amazon S3.
* Using Amazon Redshift along with Amazon DynamoDB like using copy command for loading Redshift table from Amazon DynamoDB table.
* Importing data from remote hosts over SSH by using the copy for loading data from one or multiple remote hosts, like EMR clusters, EC2 machines etc, Multiple simultaneous connections are supported by Redshift allowing parallel reading and loading of data.
* Automation and transformation of data workloads using AWS Data Pipeline by making use of built-in scheduling capabilities. For say, a periodic job to automatically copy the data from DynamoDB to Redshift.
* **Best Practices:**
* **Vacuum Operation**

Vacuum operation is used to maintain consistent query performance. In the backed, Redshift does not reclaim the freed space on deletion and updation of rows. Also, it does not sort the inserted data on the fly. This unsorted data leads to degraded query and operations’ performance. This space is reclaimed using vacuum operation which should be executed as often as possible.

* **Analyze Operation**

Analyze operation is for high performance, it minimizes the data scanned. Actually, AWS Redshift uses stats of tables which get outdated on insertion of new data. Analyze operation updates these stats.

* **Backup Snapshot**

Amazon Redshift replicates the data within cluster on loading and it is continuously backed up to S3. It maintains at least three copies of data- first, the original one , second, the replica and third, a backup in S3. Only the changed data is backed-up so only small amount of storage is used. On deletion of cluster, a final snapshot is created which is useful to restore of the data which is deleted.

* **Cluster Resizing**

When we modify the cluster, the requested changes are applied immediately, the cluster becomes unavailable for a period of about four to eight minutes. But with the feature of concurrency scaling, virtually unlimited concurrent queries and users. On enabling concurrency scaling in Redshift, it automatically keeps adding capacity to cluster when needed and the cluster remains fully available without any downtime.

* **AQUA (Advanced Query Accelerator)**

AQUA is new distributed, hardware-accelerated cache feature which enables Redshift to perform up to 10x faster as compared to other cloud data warehouses. It does a substantial part of data processing in-place, that is on the cache memory itself that is data intensive operations are performed close to storage layer. It also supports encryption, authentication and isolation to keep data at rest and in transit securely.

* **Maintenance Window**

Periodic maintenance is performed to apply fixes, upgrades and add new features to the cluster. These maintenance windows are schedulable and can be modified, either programmatically or by using the console.

* **Performance Monitoring**

Metrics like utilization of storage and compute, read/write traffic operations to Amazon Redshift data warehouse cluster are traceable in Amazon EventBridge. Also, additional, user-defined metrics can be added. Amazon Redshift provides information about queries and performance of cluster enabling users to keep a tab on high consumption or resource utilization and execution statistics.

* **Network Security**
* **Cluster security groups**

For granting inbound access, Amazon Redshift gives a cluster security group named default, it is automatically created on launch of cluster.

* **VPC**

For restricting access to cluster, there is an option of launching it in an Amazon Virtual Private Cloud. For connecting this cluster from outside the VPC, there is option to use elastic IP address. This also helps to upgrade configurations without changing IP address again and again, this is helpful for recovery after failure.

* **Data Security**
  + **Data encryption** **at load time**

Server-side or client-side encryption is used while uploading data files to S3. On loading from server-side, S3 bucket will handle decryption operation transparently. And on loading it from client-side, copy command of Redshift is used for decryption.

* + **Data in transit**

For data protection in transit, Redshift automatically makes use of hardware-accelerated SSL for communication with Amazon S3 and DynamoDB while performing copy, backup, restore and unload operations.

### **Selection of distribution styles in tables**

### Redshift automatically assigns an optimal distribution style on the basis of data size in table. Due to this we get improved query performance and utilization of storage.

### It initially assigns ALL distribution, then change it to EVEN distribution on size increases. This change happens in background, in just and never affects the performance of current queries which are in execution.

### **Selection of sort key in tables**

* Storage of data according to sort key, helps query optimizer in determining optimal plan for query execution.
* Redshift skips reading whole data blocks for one particular column by tracking minimum and maximum stored and skipping the ones which are out of that range.
* **Loading data**
* Splits the dataset into multiple sections for uniform distribution across cluster.
* **Use a Manifest File**

It helps in specifying S3 locations by using s3 prefixes.

* **Compression of data files**

Depending on the priority is to reduce the time spent by copy commands use **LZO compression** if time spent by copy command has to be reduced, else if priority is on reduction in s3 size then make use of **BZ2 compression**.

* One should load data in the order of sort key and use IAM role for the same

In conclusion, selecting AWS Redshift for your data warehousing needs is the latest and an upmarket way to go about big data solutions in cloud environment.