Redshift

Summary:

AWS redshift is a column based column-oriented massively parallel processing data warehouse. Redshift aims to provide customer with high performant analytical queries with techniques like prefetching and vector execution to scale linearly. It outsources its compute to redshift compute cluster and storage to the Redshift managed storage backed by S3. Figure 1 beautifully depicts the Redshift architecture with various other components into the play. The data is cached at the compute nodes in ssd in a compressed column-oriented format(open source or propriety format?). Concurrency scaling handles multiple queries and data sharing is used to share the data in read only format to other redshift clusters and uses IAM policies. Redshift provides two options to connect: JDBC/ODBC connectors(java) or Data API. The Data REST API provides ease of access to the customers and can reduce the overhead of installing the connectors. Leader node receives the query made by client and is responsible for parsing and optimization. The cost-based optimizers contains the cluster topology and the cost of moment for the data and makes optimal decision accordingly. WLM manages the execution phase and the data is either taken from the cache (in compressed form from the local ssd) or from RMS. Data from other compute node are exchanged via the pipeline. It also has zone maps which has min/max values (I don't know if its similar to parquet file format or are they actually talking about parquet here?) which supports the late materialization. They have bloom filters and prefetching for further optimization. The execution model of redshift is customized to exploit the EC2 nitro hardware. The benchmarks are quite promising and I guess they refrained actually naming the cloud competitors due to legal issues. The redshift code generator generates C++ code which is compiled and the binary generated is shipped to the compute node. It does not follow the Volcano execution model and instead all the operator of the specific query are generated on the fly. It also has a SIMD vectored scan layer. Redshift also makes smart decision based on if the hash table is too large to fit into cache and generates additional code based on that. The query processing engine is responsible for optimizing object files and caches the compiled object off-cluster. Hence the object file is either found most of the time in the local cache or in external cache node.The off-clustering caching layer is provided by Advanced query accelerator(AQUA) for RMS and accelerator for scans and aggregation. It uses AWS's Nitro ASIC. RMS is build on the nitro system to get Max performance. It uses the superblock indexing system. Metadata is decoupled from data due to scaling purposes and provides cross instance restore. The Elastic restore technology has been beneficial for over 15000 times each month . It follows the MVCC for concurrency control and follows the SSN for snapshot isolation. For the maintenance task and tuning operation, it offers different form of machine learning algorithms such as XGBoast and AutoWLM to make smart business decision. It has a smart warm pool architecture to replace faulty nodes and has outlier detection algorithms for gray failures which are significantly difficult to detect. It also provides the server-less architecture to the customer so that the end user does not have to worry about the calculation of resources and can pay based on the on-demand basis. The redshift spectrum offers support for open file formats. Redshift ML and Sagemaker provides a robust machine learning pipeline for in-database prediction and inference using SQL. It also has support for federated queries. SUPER is another helpful edition to aid redshift. Hence it turns out for cloud data warehousing.

Comments:

Redshift has tight integration to the other AWS services based on the use case. While I really appreciate the number of services provided by AWS, it does create a sense of tight coupling. What if I want to replace any particular component of the architecture(like s3 or pagemaker) with external cloud provider or maybe use my own server? Like if there’s some private data or government data and it has some strict policies with regards to the data moment policies, will I be able to use AWS redshift or should I find some alternate services?

Since it is columnar storage, it is successfully able to exploit the compression schemes and reduce the number of I/O. I like how they focused on the automation expect and have tried to tackle it with ML based solution.

Since it has a leader node, what happens if the leader node fails and will that not be a single point of failure.

Redshift has many customized hardware such as Nitro for peak results, however should not that offer limited portability in case there’s a switch necessary ? Also it might be difficult for maintenance.