Rise of Elastic Data Warehouse and Database Services

Abhishek Tiwari

Citation: *A. Tiwari*, "Rise of Elastic Data Warehouse and Database Services", Abhishek Tiwari, 2015. doi:10.59350/vm1wv-7ty41

Published on: July 03, 2015

Currently the majority of cloud based database and data warehouse services are provisioned with fixed storage and compute resources. Resizing of resources cannot be performed without compromising availability and performance. This means service users typically end up with over-provisioned under-utilised expensive resources to accommodate possible peak demand. In the worst case, under-provisioned resources unable to handle sudden work overloads. In my opinion, this is a big limitation on the part of the cloud based database and data warehouse services, especially if you consider how in the cloud one can easily auto-scale on-demand storage and compute these days.

Resizing issues

Let's take an example here, Amazon's much advertised and highly popular data warehouse service Redshift can not be resized without putting a cluster in read-only mode. Now, depending on the amount of data and number of nodes resize can take anywhere from a couple of hours to a couple of days. Not only that,

> After Amazon Redshift puts the source cluster into read-only mode, it provisions a new cluster, the target cluster, using the information that you specify for the node type, cluster type, and number of nodes. Then, Amazon Redshift copies the data from the source cluster to the target cluster. When this is complete, all connections switch to use the target cluster. If you have any queries in progress at the time this switch happens, your connection will be lost and you must restart the query on the target cluster.

That's basically not ideal if you are after minimal production impact. Alternative approach using snapshot, restore, and resize operation is equally daunting - requires replay of the extract, transform, load (ETL) process.

If you are an existing user of Amazon Relational Database Service (RDS), when resizing database instance, having a Multi-AZ (availability zone) deployment is definitely helpful. The same can not be said for Single AZ deployment in which case database instance can not be resized without downtime. Ideally a resize operation applied on Multi-AZ RDS deployment should have low or no impact on availability and performance of the database instance. That said, in this case,

> The impact is limited to the time automatic failover takes to complete: typically one to two minutes

Understanding Elasticity

Definition of elasticity for a data warehouse and database services is quite simple - able to scale data storage and compute independently either automatically or on-demand without compromising the

availability and performance. This requires full or some level of de-coupling between storage and compute layers of a database or data warehouse service. Ideally, this de-coupling should be reflected as separate billing of storage and compute layers.

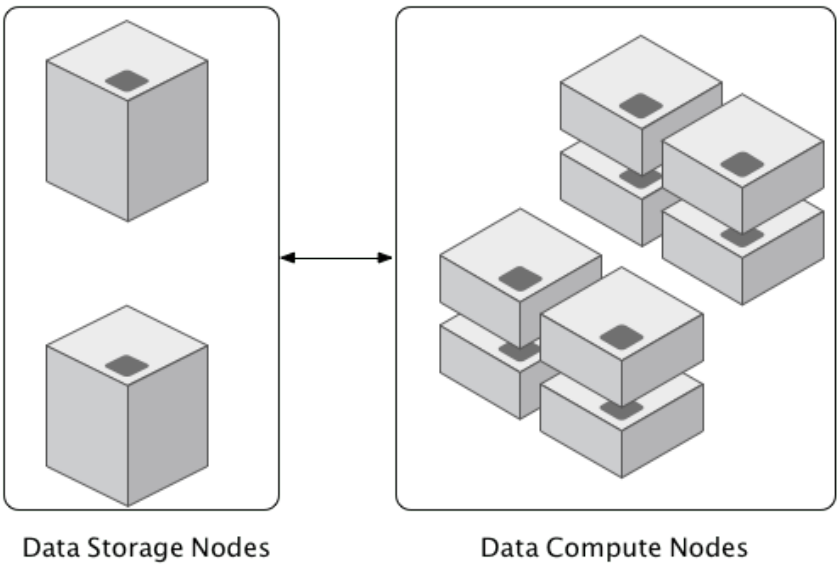

Figure 1: Elastic Database and Data Warehouse - clear separation between storage and compute layers allow to scale them independently

One key aspect of data warehouse elasticity is variety - ability to query or handle structured data (relational) as well, semi-structured/unstructured data (non-relational) is desirable if not critical.

In last 6 months, several new database and data warehouse services have been launched with elasticity as a major theme. Although many of these services are still in early or preview stage, but on the surface these offerings are really promising.

Azure SQL Data Warehouse

Recently launched Azure SQL Data Warehouse (DW) is a fully managed, elastic, and petabyte-scale columnar data-warehouse service. Both Amazon Redshift and Azure SQL DW use massive parallel processing (MPP) architecture to deliver breakthrough performance. But unlike Amazon Redshift, Azure SQL DW architecture allows data and compute to scale independently without any downtime. In addition, Azure SQL DW enables one to dynamically grow and shrink resources taking advantage of best-in-class price and performance.

Most importantly, with Azure SQL DW, storage and compute are billed separately. Storage rates are based on standard blob rates. Compute is priced based on DWU (Data Warehouse Unit) blocks - a new metric to measure the compute capacity when using Azure SQL DW. Performance in Azure SQL DW scales linearly, and changing from one compute block to another (for instance 100 DWUs to 1000 DWUs) happens in seconds without disruption.

Azure SQL DW uses SQL Server's Transact-SQL and PolyBase technologies to query across both relational data (data warehouse) and non-relational data (Azure blob storage).

Snowflake Data Warehouse

The Snowflake Data Warehouse is designed and built for the cloud, and its architecture physically separates and logically integrates compute and storage layers. For storage Snowflake is leveraging highly-available and elastic Amazon S3 for centralise storage of data at a low cost. The snowflake's architecture enables it to independently scale up and down on the fly exactly when it is needed without disrupting storage and compute availability. With Snowflake dedicated but separate compute capacity can be provisioned to optimise the read and write throughput.

Snowflake also combines semi-structured data with relational data using SQL queries.

> Load semi-structured data without being forced to define a fixed schema, and then query that data using SQL in combination with your structured data with the benefit of the full optimisations available from a relational database.

Amazon Aurora

Amazon Aurora is next-generation Amazon RDS offering. It is a drop-in substitute for MySQL - fully compatible with MySQL 5.6. On a high-level Aurora storage and compute layers are de-coupled with a major focus on storage elasticity and durability.

Aurora data are stored in a cluster volume, which is a single, virtual volume that utilises solid state disk (SSD) drives. Aurora cluster volumes automatically grow in increments of 10 GB and your database

size can be increased up to a maximum of 64 TB. Needless to say, you never over-provision the storage capacity with Aurora. Storage auto-scales without any disruption or performance degradation. Aurora automatically backups incremental and continuous chunks in Amazon S3.

Scaling the compute powering an Aurora database deployment either up or down requires adjusting compute resources (vCPU and RAM). Compute scaling operations typically complete in a few minutes again without any downtime. To handle high read-throughput you can add up to 15 Aurora read replicas per database. Because Aurora read replicas are using same underlying storage as the source instance, there are no write overheads for replica nodes.

Apache Drill and Qubole

Although Apache Drill is not a database or data warehouse service, it's been quite suitable for a self-managed elastic data warehouse architecture. Apache Drill offers schema-free SQL query engine for a variety of data sources including Amazon S3. One key aspect of Drill based data warehouse architecture is separate and on-demand scaling of storage (in this case Amazon S3) and compute (1000s of Amazon EC2 servers). Compute or query layer can be automatically grown and shrunk using Amazon EC2 autoscaling rules.

Similarly, Qubole which offers big data as service in the cloud using an elastic model can independently scale the storage layer (primarily Amazon S3) and compute layer (fleet of Amazon EC2 servers). Qubole also offers variety of query processing engines such as Pig, Hive and Presto as Service. A key selling point for Qubole is it's ability to detect change in utilisation of compute resources which means it can grow and shrink the size of compute cluster to avoid waste of resources.