Modern data warehouses

Data warehouses – a collection of methods, techniques & tools used to support people conducting data analysis which helps with decision making processes & improves information resources. Primary purpose of a data warehouse is for BI

Types of data warehousing

On-premises data warehouse

Cloud data warehouse systems – address all issues faced by traditional data warehousing

Data warehouse architecture

There are many different types of data warehousing solutions. One of the most common types of traditional warehouse architecture is the 3-tier architecture approach:

Bottom tier – contains actual database server

Middle tier – server for online analytical processing (OLAP). This server is responsible for transforming data. It can map multi-dimensional data to relational operations or leverage a multi-dimensional OLAP model, etc.

Top tier – similar to UI layer. Contains tools common for data warehousing analytics such as reporting & mining.

Data warehousing approaches

2 of the most commonly used approaches to data warehousing were created by Bill Inmon & Ralph Kimball.

Bill Inmon’s approach

It’s also referred to as a top-down approach

Data warehouse is treated as a central repository for the data

Data marts are then created to address individual business lines

Ralph Kimball’s approach

Bottom-up approach

Data mart is the main method for data storage

Data warehouse is then created & comprises of multiple data marts with shared analytics, reporting & BI essentials. This allows for uniform analytics jobs

On-premises data warehouse components

Data analytics layer - There is a specific layer for data analytics, data mining jobs

Staging area – different sources have different formats & structures. In the staging area, data is converted into a specific structure & format

Data Marts – they assist the staging area by acting as a housing for a specific line of business & are summarized for specific queries

Cloud-based data warehouses advantages

Up-front costs

Ongoing costs – pay as you go model

Speed – substantially faster than on-premise options, due to the use of ETL which is not commonly used in on-premise options

Flexibility – designed to account for the variety of formats & structures in big data

Scalability – clouds are elastic

Earlier the focus of data warehouses was to stack data & focus on processing it for analytics. Now instead of just stacking the data, we are also focusing on storing loads of it as big data for advanced analytics.

A traditional data warehouse can be compared to a data store & a modern data warehouse can be compared to a mega distribution store. Traditional data warehouses are focused more on data analytics & modern data warehouses focus on enterprise analytics.

Data lake – a repository that takes data from multiple sources & stores them in a variety of formats

Advantages of a modern data warehouse

User benefits

Less time spent on preparing & moving data

More time is spent on innovative analytics & new business models

Organizational benefits

Analyze new data faster

Reduce overall cost of ownership

Increased productivity

Technological benefits

100% availability

Highly scalable

Resolve queries correctly in any schema

Provides real-time updates

Handles ETL processes

Supports batch & interactive workloads

Supports large no: of simultaneous users

Previously, the strategy used by organizations to save money was to transfer data from databases to file systems. Now, the trend is to move data from file systems to object storage. This is because cheap storage is of lower priority than data availability.

Modernized data warehousing tools

Apache Hbase – key-value columnar database

Apache HCatalog – metadata, table & storage management system

Hadoop MapReduce – scalable data processing tool

Apache Hive – open-source language built for MapReduce to assist with large data analytics

Oozie – MapReduce job scheduling tool

Apache Pig – language with MapReduce functionality used in parallel data processing

Apache Zookeeper – hierarchical key-value store for synchronization

A modern data warehouse must

Support cloud-based platforms & cloud-to-cloud interactions

Must support multiple intercloud interactions & interactions with on-premises systems

Should facilitate easy reconciliation of structured & unstructured data

Benefits of unified data warehouse

Great scalability & agility

Cost reduction

Faster deployment & processing speeds

Easier disaster recovery

Improved governance & security capabilities

Modern data warehouses separate the compute & storage functions to optimize infrastructure investments. This also provides the ability to query the data at any tier. This is also cost effective especially if you are using a cloud vendor as they charge different rates for storage & compute resources. Therefore the compute resources can be enabled/disabled according to usage & thereby save money.

Amazon Redshift

Data warehouse product which is part of AWS.

Redshift – shift from oracle (logo is red)

Built on massive parallel processing technology

Can easily handle large-scale data & facilitate data migrations

Easily handle analytical workloads on big data sets

Amazon Redshift features

Decreases command execution time using parallel processing & compression

Can perform operations on billions of rows at once making it useful for data storage & analytics

Redshift architecture

Typically, a Redshift cluster consists of 2 or more cluster nodes which is coordinated through a leader node. The clients communicate to the leader node. The leader node manages internal & external communication & is responsible for the preparation of query execution plans.

Once the query execution plan is ready, the leader node then dispenses query execution code to each of the compute nodes & assigns slices of data to each node in the cluster for easy computation of results.

The leader node dispenses query execution to compute nodes only when the query involves accessing data stored on compute nodes. Otherwise, the query is executed on leader node itself.

There are 2 types of compute nodes:

Dense storage – allow us to create large data warehouses using HDDs hard-disk drives

Dense compute – allow us to create large data warehouses using SSDs solid-state drives

A typical compute node consists of node slices. Each node slice is assigned a part of the compute node’s memory & disk, where it performs query operations

Redshift uses columnar data storage – reduces no: of IO requests & minimizes data loaded into memory during querying

Data compression – reduces storage footprint & enables loading of large amounts of data

Redshift query optimizer – generates query plans that are MPP-aware & takes advantage of columnar data storage

Amazon Redshift use cases

Redshift allows users to easily build pipelines to multiple sources & perform BI operations on the incoming data which is cheaper than a traditional data warehouse.

Advantages of storing raw data in Redshift

Higher level of analytical insight

Minimal data loss & no aggregation requirements

Possibility of historic replays of data

Amazon redshift is commonly used for mission-critical workloads with time-sensitive information flaws. In these operations, the key concern is having the database running all the time.

Google BigQuery

Serverless data warehouse

Easy scalable analysis

PaaS

Queries in ANSI SQL

It contains built-in ML capabilities & provides access to Google’s Dremel technology – a scalable, interactive ad hoc query system for analysis of nested data

Google BigQuery Features

Data management – enables creating & deleting objects such as tables, views, user-defined objects etc., importing data

Data queries – queries are expressed in SQL & results are returned in JSON format with a max. reply length of 128 MB & size is unlimited when large query results are enabled

Data integration – BigQuery can be used in any apps using its REST API etc.

Access control – enables sharing of datasets with arbitrary individuals, groups, etc.

ML capabilities – ML models can easily be created & executed using SQL

Google BigQuery Architecture

Dremel – highly scalable query execution engine for petabyte-scale datasets. It uses a combination of columnar data layouts & a 3 type architecture that can process incoming query requests. It is capable of independently scaling compute nodes to meet the demands of the highest level queries.

Colossus – distributed file system with high storage capabilities & disaster recovery strategies

Jupiter network – bridge between Colossus’ data storage & Dremel which offers bi-directional large volume data movement

Google BigQuery Features

Manageability – administrator is not required to manage the service. It’s taken care of by Google. Patching, upgrades, storage management, compute allocation, etc. is managed by Google thereby offering serverless execution to users.

Highly scalable

Load data in different formats & has a built-in conversion to BigQuery

Data ingestion – supports data streaming & batch data ingestion at no extra cost

Security – supports multiple authentication models & granular permissions

Usability – traditional data warehousing patterns are closely followed

ETL – a data warehouse process which includes 3 actions

Extract – extract data from source

Transform – transform data for business use

Load – load data to target table in a data warehouse or different locations inside data warehouse

Column selection approaches

Conditions that determine which columns are to be loaded are:

Translating coded values – for example, something can be indicated using colors but to store them in the system we could use numeric values which denote each color

Value mapping

Joining different values. Eg: merging

Summing several data rows

Surrogate key selection

Transposing – changing multiple columns to multiple rows or vice versa

From staging area to data stores

ETL process transfer data from source/operational systems -> staging area which is a temporary storage area where data is transformed as necessary -> Operations management tools which process the data -> finally the data is kept in data storage

Data quality measures

Data quality is measured in terms of –

Completeness

Consistency

Accuracy

Improving data quality

Data ingestion firewall – data from a data warehouse can’t be used directly before it’s processed & cleansed. The firewall ensures data quality.

Data profiling – ensure the data quality is always up to standard

Hadoop data storage advantages

Data is stored across multiple serves

Missing data is replicated from other servers

Analysis software is separate from raw data storage

Relatively low cost of data storage

Hadoop data storage disadvantages

May cause data quality issues due to processing

Data needs to be cleansed & mined thoroughly before performing analyses

Specialists are required to file & report data extraction using Hadoop

Data warehousing techniques

Data management

Data management is responsible for data quality

Increasing data complexity requires effective data management strategies

Data management provides a unified view on data ingested from multiple sources

Identical data definitions are required across organizational boundaries

Data management enables data sources consolidation into a master reference database

Service oriented architecture

Implement the organization resources towards a services approach

It’s a design philosophy providing a service instead of a product.

Any IT resource can be reached using a service device

Web services use specific standards & protocols when they are implemented as SOA solutions

Data ingestion & analytics can be done in 3 different ways:

Batch processing

Real-time processing

Streaming analytics

Batch processing

Netflix users generate lots of data everyday. The data scientists & engineers have to collect this data & implement data analytics models to find customer behavior.

First, they write an ETL job & pipeline the data

The processed data is stored in AWS S3

Iceberg is used in Netflix for data tabulation

Netflix also utilizes a microservice architecture that emphasizes on the separation of concerns

The above jobs require a fast lookup which can’t be channeled through data warehouse as it requires a lot of time. Therefore, the key attributes are stored on a global low latency & reliable key-value store.

Bulldozer – self-serve data platform that moves data efficiently from data warehouse tables to key-value stores in batches.

Scalable & efficient no-code solution

Provides functionality to auto-generate data schema

Can be executed at the desired frequency – once or many times a day

Protobuf – filing method used by Bulldozer

For representing warehouse table schema into key-value schema

Serializing & de-serializing key-value data when performing read-write operations to KvDal (Key value data abstraction layer). This keeps key value storage engine abstracted so users don’t have to work with it.

Bulldozer requirements

Data integrity – for 1 bulldozer job moving 1 version of data it should write the full dataset or nothing

Seamless data consumption – once a bulldozer job finishes moving a new version of data, the consumer should be able to start reading the new data seamlessly

Data fallback – fallback to previous version if data is corrupted

Real-time analytics

IPL cricket analytics platform uses Amazon Redshift to store all historic data of the league aggregated from multiple databases & uses this data for ML & predictive modelling.

Key cricket metrics used are:

Match performance moving averages

Score forecasting

Fitness & performance insights for pairs of players

Individual player contributions to wins & losses

Using real-time data analytics during the game, the team can even improve their winning chances by coming up with new tactics guided by data.

Streaming analytics

Uber uses big data to gain insights of in-the moment traffic conditions to predict estimated delivery time of UberEats.

Uber needed:

Easy to comprehend data analytics platform

Scalable & efficient for real-time analytics

Robust enough to continuously support a large number of critical jobs

They made AthenaX

Hybrid modern data warehouses

Retrieving data from a data warehouse poses latency issues. Therefore, data marts are created to store all domain specific data from a data warehouse locally. This reduces server downtime & speeds up data analytics processes.

Batch processing is required when day-to-day company data needs to be analyzed & presented in a report.

Real-time analytics using data marts – data marts are used for intermediate real-time processing of data. Data marts make it possible for some domain-specific data to be loaded into a data model & analyzed without utilizing a data warehouse.

Streaming analytics need to analyze data as it is generated which is useful for making instantaneous decisions like stock market trading.