QuCelerate

Andrew Miller

Breaking Down Data Warehouses: What to Know Before You Choose

What is a data warehouse?

Dictionary.com: a large, centralized collection of digital data gathered from various units within an organization

Wikipedia: is a system used for reporting and data analysis and is considered a core component of business intelligence

OLTP vs OLAP access patterns

OLTP (online transaction processing)

```sql
select * from github.issues where id = '23'

insert into github.issues (id, status, author, date)
values ('123', 'pending', 'andrew', '2022-03-11')
```

OLAP (online analytical processing)

```sql
select author, count(*) from github.issues where status
= 'open' group by author

copy github.issues from 's3://issues/2022/january.csv'
```

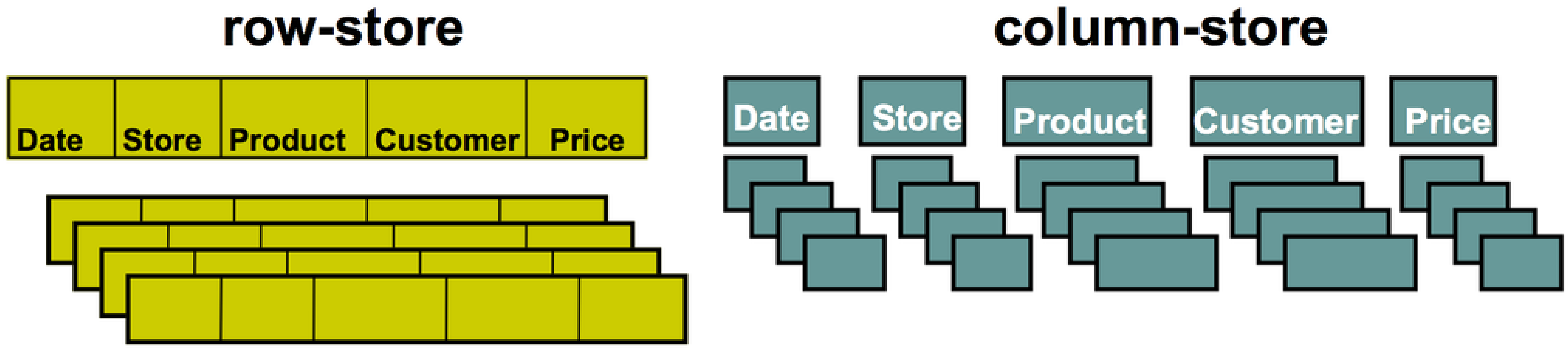
HTAP: Hybrid transaction analytical processing

- Can handle both OLTP and OLAP
- Examples of HTAP database management systems
  - Apache Cassandra
  - Apache Hbase
  - SAP HANA
  - Citus (Postgres extension)
  - SingleStore
  - TiDB

Row vs columnar database

- In row-oriented databases, data is stored on disk one row at a time.
- Optimized for OLTP (transactional, operational databases)

- In column-oriented databases, data is stored on disk one column at a time.
- Optimized for OLAP (analytical databases)



Data volume estimates (row-oriented)

- Let’s assume
  - 100 studies a year
  - 3,000 subject events in a study
  - Retaining data for 5 years
  - Each subject event is roughly 700 B (7 fields, 100 bytes each)
  - AWS gp3 SSD baseline access rate is 125 MiBps (1000 MiBps for an extra fee)
  - No index on at least one column in the query (in reality you might)
- Then
  - Subject events dataset is about 1,050 MB
  - Reading data from disk takes 1,050 MB / 131 MB/s = 8 s
  - Plus time to scan through the data in memory (depends on CPU, etc)

Data volume estimates (column-oriented)

- Let’s assume
  - Query uses two columns (group by and where, for example)
  - Data is not compressed (in reality it will be)
  - Data is not sorted, distributed, or partitioned (in reality it will be)
- Then
  - Two columns we care about take up 300 MB (2/7 of the entire table)
  - Reading data from disk takes 300 MB / 131 MB/s = 2.3 s
  - Plus time to scan through the data in memory (depends on CPU, etc)

Access patterns

- Ability to view KPI's in a grid planned vs actual data (including the difference)
- Ability to filter the data in the grid by geography level
- Assume ETL or materialized view brings latest events to a separate OBT (one big table) schema out of all historic events
  - `select count(*) from study_subject_lastest_events where status = 'enrolled' and cohort = 'cohort A'`
- Looks a lot like OLAP access pattern

Why do we need a data warehouse?

- Centralize and integrate data from multiple sources
- Support online analytical queries
- Support AI/ML workloads
- Provide a low-latency user experience
- Improve data quality

What are the options?

PostgresQL	Redshift	Snowflake
Row-oriented	Columnar	Columnar
MPP only possible with Citus extension	Built-in MPP (massively parallel processing)	Built-in MPP
Scaling is manual and takes SDE and SRE effort	Scaling is manual and takes SRE effort	Automated scaling
Storage scales independently from compute	Storage does not scale independently from compute (unless using RA3 nodes)	Storage scales independently from compute
Manual index and partition key setup	Automatic Table Optimizations	Faster raw query performance
Manual index and partition configuration management (requires SDE and SRE involvement)	More operational overhead than Snowflake (setting up sort and distribution keys, auto vacuum, etc)	Maintenance and troubleshooting are more automated
Widely used and many extensions are available	Better integration with AWS ecosystem	Better multi-cloud (AWS, Azure, GCP support) and multi-region experience
Only database snapshots can be shared	Account-to-account (instant) data sharing (less popular than Snowflake)	Account-to-account (instant) data sharing (more useful since Snowflake is more popular)
DDM is only available with Anonymizer extension	Built-in (DDM) dynamic data masking only in preview	Built-in dynamic data masking (in Enterprise tier)

Costs

- Each service uses different types of nodes
- Node specs do not overlap between services

Environment	PostgresQL	Redshift	Snowflake
Non-prod cluster size	1 node	1 node	1 node (X-Small, Standard Edition)
Non-prod node specs	db.m6g.2xlarge (8 vCPU, 32 GB RAM)	ra3.xplus (4 vCPU, 32 GB RAM)	c5d.2xlarge (8 vCPU, 16 GB RAM)
Non-production	**$1,252/month**	**$843/month**	**$144/month**
Prod cluster size	1 node (no Citus on RDS)	1 nodes	1 node (Medium, Business Critical Edition)
Prod node specs	db.m6g.2xlarge (8 vCPU, 32 GB RAM)	ra3.xplus (4 vCPU, 32 GB RAM)	c5d.2xlarge (8 core, 16 GB)
Production	**$1,252/month**	**$843/month**	**$472/month**

Gotchas/ Things to watch out for