

Challenges in Transitioning from Traditional Data Warehousing to Modern Data Architectures

Aneehika Nellutla

Optum Insight
Indian Land, SC, USA
aneehika.n[at]gmail.com

Abstract: *The transition from traditional data warehouses, such as SQL Server, to modern data architectures like the Medallion architecture, has introduced a paradigm shift in data management and processing. This paper explores the challenges faced during this transition, particularly focusing on differences in scalability, schema design, data partitioning, and query optimization. It also examines the implications of adopting distributed systems and new technologies, highlighting the practices that need adaptation. By addressing these challenges, organizations can fully realize the potential of modern data platforms.*

Keywords: Modern Data Warehouse, Traditional Data Warehouse, Medallion Architecture, MPP Systems, Scalability, Data Partitioning, Query Optimization, Schema-on-Read, Real-Time Processing, Distributed Systems

1. Introduction

Traditional data warehouses have been the backbone of enterprise data storage and processing for decades. They rely on monolithic architectures, rigid schema enforcement, and centralized query optimization. Modern data architectures, such as the Medallion architecture, leverage distributed, massively parallel processing (MPP) systems to meet the demands of big data, real-time analytics, and scalability.

This paper discusses the key challenges in migrating from traditional data warehouses to modern platforms, focusing on technical, architectural, and operational aspects. Figures are included to illustrate key comparisons and trends.

2. Understanding Medallion Architecture

The Medallion architecture, popularized by Databricks, is a framework for organizing data within a data lake to ensure scalability, maintainability, and reliability. It divides data into three distinct layers—Bronze, Silver, and Gold—to manage data at different stages of its lifecycle effectively.

Bronze Layer

- **Purpose:** This layer contains raw, unprocessed data ingested from various sources, including IoT devices, logs, and transactional databases.
- **Characteristics:**
 - Data is stored in its original format (e.g., JSON, CSV, Parquet).
 - Minimal transformations are applied.
 - Acts as the immutable source of truth for ingested data.

Silver Layer

- **Purpose:** This layer focuses on data cleansing, deduplication, and normalization. It serves as the intermediate processed data layer.
- **Characteristics:**
 - Data is structured and enriched to provide consistency.
 - Data validation checks and error handling are implemented.

- Optimized for consumption by downstream analytics or machine learning systems.

Gold Layer

- **Purpose:** The Gold layer contains aggregated, business-ready data used for reporting and analytics.
- **Characteristics:**
 - Highly curated datasets optimized for specific use cases.
 - Often includes metrics, KPIs, and dimensional modeling.
 - Data is typically consumed by dashboards, business intelligence tools, and advanced analytics platforms.

Advantages of Medallion Architecture

- **Scalability:** By processing data in stages, this architecture accommodates large-scale data efficiently.
- **Modularity:** Each layer is decoupled, enabling independent scaling and maintenance.
- **Governance:** Clear boundaries between layers enhance data governance and lineage tracking.
- **Real-Time Capabilities:** Streaming data can be integrated into the Bronze layer and progressively refined for near real-time analytics.

3. Key Challenges

Scalability and Resource Management

- **Traditional Approach:** SQL Server and similar databases use vertically scaled architectures with limited hardware [1].
- **Modern Architecture:** MPP systems scale horizontally, introducing challenges in data distribution, node balancing, and resource allocation [2].

Identity Columns and Key Management

- **Traditional Practice:** Identity columns provide sequential IDs, simplifying indexing and referencing [3].

Volume 13 Issue 12, December 2024

Fully Refereed | Open Access | Double Blind Peer Reviewed Journal

www.ijsr.net

- **Modern Challenge:** Distributed environments prefer surrogate keys or UUIDs to prevent bottlenecks and data skew [4].

Data Partitioning

- **Traditional Approach:** Often neglected due to smaller datasets.
- **Modern Requirement:** Effective partitioning strategies are critical to optimize distributed queries and minimize data shuffling [5].

Schema Design: Schema-on-Write vs. Schema-on-Read

- **Traditional Approach:** Schema-on-write ensures data integrity but limits flexibility.
- **Modern Transition:** Schema-on-read enables raw data ingestion, increasing flexibility but requiring stronger governance [6].

Query Optimization

- **Traditional Practice:** Indexes and execution plans drive query performance [7].
- **Modern Requirement:** Optimization involves minimizing inter-node communication and leveraging caching mechanisms [8].

Data Consistency Models

- **Traditional Approach:** ACID compliance ensures strong consistency [9].
- **Modern Challenge:** Eventual consistency models, common in distributed systems, demand reconciliation and conflict resolution mechanisms [10].

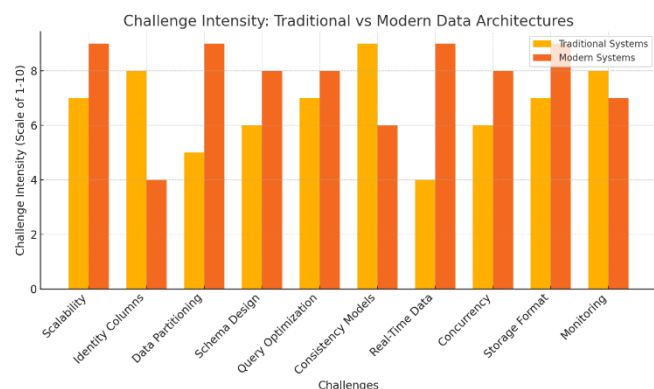


Figure 1: Challenge Intensity: Traditional vs Modern Data Architectures

This figure compares the intensity of challenges faced by traditional and modern data architectures.

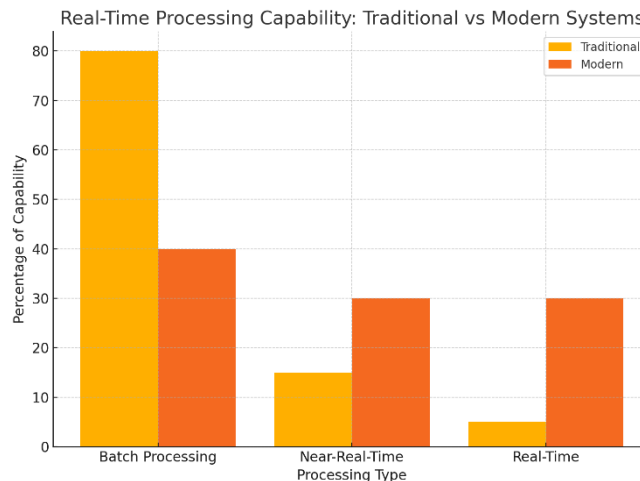


Figure 2: Real-Time Processing Capability: Traditional vs Modern Systems

This figure illustrates the distribution of batch, near-real-time, and real-time processing capabilities.

4. Implications

The transition to modern architectures offers scalability, performance, and flexibility but requires substantial changes in practices. Organizations must invest in upskilling teams, adopting new tools, and rethinking traditional processes to avoid pitfalls during this shift.

Operational Implications

- Emphasizing team training on distributed systems and open storage formats.
- Revisiting data governance policies for schema-on-read.

Technical Implications

- Implementing robust partitioning and data distribution strategies.
- Ensuring optimized workload management for high concurrency environments.

5. Conclusion

The shift from traditional data warehouses to modern architectures is inevitable as data volume and velocity grow. By understanding the challenges outlined in this paper, organizations can better prepare for and navigate the complexities of this transition. The adoption of modern practices such as distributed key management, effective partitioning, and schema governance is essential to unlock the full potential of MPP systems like the Medallion architecture.

References

[1] Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley.
 [2] Databricks. (2023). *Medallion Architecture for Data Lakehouse*. Retrieved from databricks.com.

- [3] Snowflake Inc. (2023). Best Practices for Distributed Data Warehousing.
- [4] Delta Lake Documentation. (2023). Delta Lake for Modern Data Warehousing.
- [5] Stonebraker, M., & Cattell, R. (2011). Ten Rules for Scalable Performance in 'Big Data'.
- [6] Microsoft. (2022). SQL Server Partitioning Strategies.
- [7] Apache Hive. (2023). Optimizing Query Execution Plans.
- [8] Presto Documentation. (2023). Caching in Distributed Query Engines.
- [9] Amazon Web Services. (2023). ACID Compliance in Modern Architectures.
- [10] Google BigQuery Documentation. (2023). Consistency and Data Reconciliation.

Author Profile



Aneehika earned her B.Tech. degree in Electronics and Communication Engineering from JNTU Hyderabad, India, and her M.S. degree in Electrical Engineering from Southern Illinois University, Edwardsville. Throughout her career, she has worked in the healthcare industry, developing a deep understanding of healthcare systems and data engineering.