Topic: Modern Data Engineering Pipeline Architecture
Category: Data Infrastructure
Date: 2024-03-08
Author: Robert Kim, Data Engineering Lead

CONTENT:

DATA ENGINEERING FUNDAMENTALS:
Data Engineering involves designing, building, and maintaining systems for collecting, storing, processing, and analyzing data at scale. It bridges the gap between raw data and actionable insights.

DATA PIPELINE ARCHITECTURE PATTERNS:

1. BATCH PROCESSING PIPELINES:
   - Scheduled data processing (daily/hourly)
   - Technologies: Apache Spark, Apache Beam, Hadoop
   - Use Cases: Historical analysis, reporting, ETL jobs
   - Architecture: Lambda Architecture components

2. STREAM PROCESSING PIPELINES:
   - Real-time data processing
   - Technologies: Apache Kafka, Apache Flink, Apache Storm
   - Use Cases: Fraud detection, monitoring, real-time analytics
   - Architecture: Kappa Architecture

3. HYBRID PIPELINES:
   - Combination of batch and streaming
   - Technologies: Delta Lake, Apache Iceberg, Apache Hudi
   - Use Cases: Modern data lakes, real-time with historical context

MODERN DATA STACK COMPONENTS:

1. DATA INGESTION LAYER:
   - Tools: Fivetran, Airbyte, Stitch, Singer
   - Custom: Apache Kafka Connect, Debezium
   - Patterns: Change Data Capture (CDC), Event Streaming

2. DATA STORAGE LAYER:
   - Data Lakes: AWS S3, Azure Data Lake, Google Cloud Storage
   - Data Warehouses: Snowflake, BigQuery, Redshift, Databricks
   - Databases: PostgreSQL, MySQL, MongoDB, Cassandra

3. DATA PROCESSING LAYER:
   - Distributed Computing: Apache Spark, Dask, Ray
   - Workflow Orchestration: Apache Airflow, Prefect, Dagster
   - Stream Processing: Apache Flink, Kafka Streams

4. DATA TRANSFORMATION LAYER:
   - ELT vs ETL Paradigms
   - Tools: dbt (data build tool), SQL-based transformations
   - Data Quality: Great Expectations, Soda Core, Monte Carlo

5. DATA SERVING LAYER:

   - Analytical Databases: ClickHouse, Druid
   - APIs: GraphQL, REST APIs
   - Caching: Redis, Memcached

DATA MODELING APPROACHES:

1. DIMENSIONAL MODELING:
   - Star Schema: Fact tables + dimension tables
   - Snowflake Schema: Normalized dimensions
   - Galaxy Schema: Multiple fact tables

2. DATA VAULT 2.0:
   - Hub, Link, Satellite tables
   - Agile data warehouse modeling
   - Auditability and historization

3. ONE BIG TABLE (OBT):
   - Denormalized wide tables
   - Optimized for read performance
   - Common in BigQuery/Redshift

DATA QUALITY FRAMEWORK:

1. VALIDATION RULES:
   - Completeness: Required fields populated
   - Accuracy: Data matches real-world values
   - Consistency: Uniform format across sources
   - Timeliness: Data freshness requirements
   - Uniqueness: No duplicate records

2. MONITORING:
   - Automated data quality checks
   - Alerting on data quality issues
   - Data lineage tracking

3. GOVERNANCE:
   - Data catalog: Alation, Amundsen, DataHub
   - Metadata management
   - Access control and compliance

BIG DATA TECHNOLOGIES:

1. HADOOP ECOSYSTEM:
   - HDFS: Distributed file system
   - MapReduce: Processing framework
   - YARN: Resource management
   - Hive: SQL interface
   - HBase: NoSQL database

2. APACHE SPARK:
   - In-memory processing
   - APIs: RDD, DataFrames, Datasets
   - Spark SQL, MLlib, GraphX, Structured Streaming

   - Performance optimizations: Partitioning, Caching, Broadcast Variables

CLOUD DATA PLATFORMS:

1. AWS DATA STACK:
   - S3 for storage
   - Glue for ETL
   - Redshift for data warehousing
   - Athena for serverless querying
   - EMR for big data processing

2. AZURE DATA STACK:
   - Azure Data Lake Storage
   - Azure Databricks
   - Azure Synapse Analytics
   - Azure Data Factory

3. GCP DATA STACK:
   - Cloud Storage
   - BigQuery
   - Dataflow (Apache Beam)
   - Dataproc (Spark/Hadoop)

DATA OBSERVABILITY:
- Data Lineage: Tracking data flow through pipeline
- Data Profiling: Understanding data characteristics
- Data Drift Detection: Monitoring for statistical changes
- Performance Monitoring: Pipeline latency, throughput

MODERN DATA ARCHITECTURE PATTERNS:

1. DATA MESH:
   - Domain-oriented decentralized architecture
   - Data as a product
   - Self-serve data infrastructure
   - Federated computational governance

2. DATA FABRIC:
   - Unified architecture across environments
   - Automated data integration
   - Active metadata utilization
   - Semantic knowledge graphs

3. LAKEHOUSE ARCHITECTURE:
   - Combines data lake and data warehouse
   - ACID transactions on data lakes
   - Schema enforcement and governance
   - BI and ML directly on data

PERFORMANCE OPTIMIZATION TECHNIQUES:

1. DATA PARTITIONING:
   - Horizontal partitioning (sharding)

   - Vertical partitioning
   - Time-based partitioning

2. INDEXING STRATEGIES:
   - B-tree indexes
   - Bitmap indexes
   - Inverted indexes
   - Spatial indexes

3. COMPRESSION:
   - Columnar compression (Parquet, ORC)
   - Dictionary encoding
   - Run-length encoding

4. CACHING STRATEGIES:
   - Materialized views
   - Query result caching
   - In-memory databases

CAREER DEVELOPMENT:

- Essential skills: SQL, Python, Spark, Cloud platforms
- Certifications: AWS Certified Data Analytics, Google Professional Data Engineer
- Tools mastery: dbt, Airflow, Kafka, Snowflake
- Soft skills: Communication, project management, stakeholder management

This document provides comprehensive coverage of modern data engineering concepts suitable for designing and implementing production-grade data pipelines.