IoT Data Warehousing with Snowflake Feature Stores: A Comprehensive Cost Reduction Strategy

Presented By: Bhanudeepti Chinta
Internet of Things (IoT) 2025

The IoT Data Challenge

Unprecedented Scale

IoT ecosystems generate data at unprecedented scales. From industrial sensors monitoring manufacturing equipment to smart city infrastructure tracking traffic patterns, connected devices produce continuous streams of telemetry at millisecond intervals, reaching petabytes within months.

Diverse Workloads

IoT analytics span a diverse spectrum: operations teams need real-time dashboards, data scientists require historical data for predictive models, and business analysts want complex aggregations. Supporting these concurrent, mixed workloads traditionally required maintaining multiple data systems.

Unique Characteristics of IoT Data

High Velocity

Sensor readings arrive continuously at millisecond intervals, creating high-velocity streams demanding real-time ingestion capabilities.

Schema Variety

Device telemetry varies dramatically across manufacturers, models, and firmware versions, resulting in schema evolution challenges.

Massive Volume

Large-scale IoT deployments produce petabytes of data, requiring storage architectures that balance cost efficiency with query performance.

Bridging the Gap: Lakehouse Architecture

The Lakehouse architecture emerged to address limitations of both traditional data warehouses and first-generation data lakes. Data warehouses offer excellent query performance but prove expensive and inflexible for semi-structured IoT data. Data lakes provide cost-effective storage but lack transactional capabilities and schema enforcement.

Data Warehouses

Strong consistency, excellent performance, but expensive and inflexible

Lakehouse

Unified platform combining best of both worlds

Data Lakes

Cost-effective storage, but lack reliability and optimization

Core Lakehouse Technologies

Delta Lake

Developed by Databricks, uses transaction logs to track changes to Parquet tables, providing snapshot isolation and optimistic concurrency control.

Apache Iceberg

Created at Netflix, implements sophisticated metadata architecture supporting schema and partition evolution without rewriting data.

Apache Hudi

Focuses on incremental data processing, offering record-level update and delete capabilities crucial for slowly changing dimensions.

Lakehouse Physical Architecture

Storage Layer

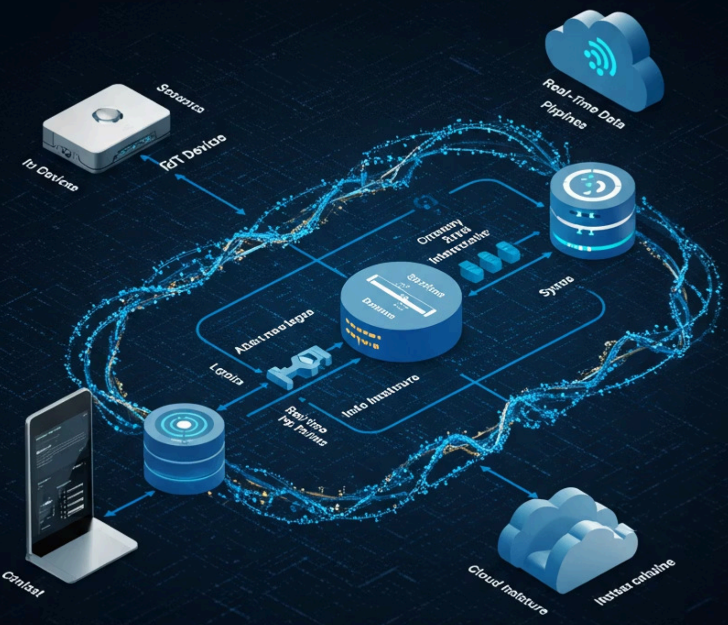
Cloud object stores like Amazon S3, Azure Data Lake Storage, or Google Cloud Storage provide virtually unlimited, cost-effective capacity.

Table Format Layer

Implements transactional semantics and metadata management through Delta Lake, Iceberg, or Hudi.

Compute Layer

Apache Spark, Presto, or Trino process queries by reading metadata and scanning data files in parallel.



Streaming Data
Archtecturre

IoT-Specific Architectural Considerations

Designing Lakehouse architectures for IoT workloads requires addressing unique challenges. Streaming ingestion forms the foundation, with tools like Apache Kafka, Amazon Kinesis, or Azure Event Hubs buffering incoming sensor data. Stream processing frameworks such as Apache Flink or Spark Structured Streaming consume these streams, performing real-time transformations before writing to Lakehouse tables.

Partitioning Strategy

Time-based partitioning by hour or day aligns with IoT data's temporal characteristics and enables efficient time-range queries.

Schema Evolution

Lakehouse must accommodate schema changes as firmware updates modify telemetry formats without disrupting analytics.

Lifecycle Management

Hot, warm, and cold data tiers balance access frequency with storage costs while maintaining unified query interface.

The Medallion Architecture

Bronze Tables

Capture raw sensor data exactly as received, preserving complete fidelity for auditing and reprocessing.

Silver Tables

Apply cleaning, validation, and standardization transformations, creating curated datasets suitable for most analytics workloads.

Gold Tables

Materialize business-level aggregations and metrics optimized for specific consumption patterns like dashboards or reporting.

Real-Time and Batch Processing Integration

Streaming Analytics

Continuously process sensor data as it arrives, computing aggregates, detecting anomalies, and triggering alerts. Apache Flink and Spark Structured Streaming maintain running aggregations with exactly-once processing semantics.

Micro-Batch Processing

Collect small batches at second or minute intervals, balancing streaming overhead with near-real-time latency. Particularly effective for IoT workloads where aggregated patterns reveal insights.

Kappa Architecture

Single streaming pipeline serves both real-time and historical use cases. Incremental materialized views continuously update, providing current metrics while feeding historical analytics.

Machine Learning Integration

Feature Engineering

Complex temporal aggregations and transformations of raw sensor data

Model Training

Access Lakehouse data through native integrations with ML frameworks

Model Serving

Batch scoring, stream scoring, or edge deployment based on latency needs

Feedback Loops

Capture predictions and outcomes for continuous monitoring and retraining

Case Study: Manufacturing Predictive Maintenance

A global automotive manufacturer implemented a Delta Lake-based architecture ingesting telemetry from thousands of robotic assembly units across multiple factories. Sensors measuring vibration, temperature, current draw, and positional accuracy stream data through Kafka into bronze tables.

Machine learning models trained on historical failure patterns identify anomalous sensor signatures predicting equipment degradation. The system generates maintenance recommendations days or weeks before critical failures, enabling scheduled downtime during planned breaks rather than emergency production stoppages.

1

Robotic Units

Monitored across facilities

2

Sensor Types

Per assembly station

Case Study: Connected Vehicle Fleet Optimization

Routing Optimization

Real-time analysis of traffic patterns and historical congestion data for efficient trip routing.

Driver Scoring

Process acceleration, braking, and cornering metrics to identify training opportunities and safety risks.

Vehicle Health

Predict maintenance needs based on odometer readings, diagnostic codes, and part failure histories.

Demand Prediction

Models predict rider demand by location and time, enabling proactive driver positioning.

Case Study: Smart City Infrastructure

A metropolitan government deployed an Apache Iceberg-based platform consolidating data from traffic sensors, environmental monitors, public transit systems, and utility meters. The architecture accommodates varying schemas and update patterns across data sources while providing unified query access.

Traffic Optimization

Analyze sensor data identifying congestion patterns and adjust signal timing dynamically

Environmental Monitoring

Track air quality, noise levels, and weather conditions with automated threshold alerts

Open Data Initiative

Publish anonymized datasets enabling researchers to build applications addressing urban challenges

Case Study: Utility Smart Meter Analytics

Extreme Scale Operations

An electric utility monitors smart meters across millions of households, ingesting consumption readings at intervals ranging from seconds to hours. The Lakehouse handles schema heterogeneity across multiple meter generations and manufacturers while time-series compression reduces storage costs.

Advanced Analytics

Detect electricity theft through consumption anomalies, forecast peak demand for capacity planning, and identify energy efficiency program candidates. Grid operations integrate Lakehouse data with weather forecasts to optimize distribution and renewable energy integration.

Smart Meters

Monitored households

Readings/Hour

Per advanced meter

Coverage

Grid visibility achieved

The Future of IoT Data Architecture

The convergence of IoT proliferation and Lakehouse architecture maturity creates unprecedented opportunities for organizations to derive value from connected device data. Lakehouse platforms address longstanding challenges: supporting both streaming and batch workloads, flexible schema evolution, cost-effective petabyte-scale storage, and integration of operational monitoring with advanced analytics.

1. **Enhanced Streaming**

   Reduced latency enabling sophisticated real-time decision systems

2. **Time-Series Optimization**

   Warehouse-class performance on temporal queries essential for IoT

3. **Edge Integration**

   Distributed analytics closer to data sources, reducing bandwidth costs

4. **Advanced Governance**

   Fine-grained access control and privacy protection for regulated industries

Thank You!

Questions And Discussions.?