StreamSets

WHITE PAPER

12 Best Practices For Modern Data Integration

Introduction

Before big data and streaming data, data movement was simple. Data moved in a linear way from static, structured databases to data warehouses or between databases and applications. Once your pipeline was built, it operated consistently because data moved like trains on a track.

Modern data has upended the traditional approach to data movement. Big data processing operations are more like a busy city traffic grid than a railroad line. Your data movement system has to be prepared to:

- Accommodate streaming, batch, or micro-batch processing
- Support structured, semi-structured and unstructured sources
- Enable streaming into multiple applications and evolving destinations
- Handle schema and semantics changes without affecting downstream analysis
- Respond to change from sources and applications controlled by different departments and even external entities

To move data at the speed of business and unlock the flexibility of modern data architectures, modern data integration systems must be architected to handle change with the ability to continually monitor and manage performance. These 12 best practices will help you develop a mindset and data integration practice to support continuous deployment and continuous operations for modern data integration.

1. Eliminate Hand Coding As Much As Possible

It has been commonplace to write custom code to ingest data from sources into your data stores. While sometimes this is necessary due to use-case requirements, this practice doesn't scale given the dynamic nature of big data and cloud. Custom code creates brittle data pipelines where minor changes to the data schema can cause the pipeline to drop data or fail altogether. Also, since instrumentation has to be explicitly designed in and often isn't, dataflows can become black boxes offering no visibility to pipeline health. Lastly, low-level coding leads to tighter coupling between components, making it difficult to upgrade your infrastructure and stifling organizational agility.

Today, there are modern data integration systems available that create code-free plug-and-play connectivity between data source types, intermediate processing systems (such as Kafka and other message queues), and your data platforms and object stores. The benefits you get from such systems are flexibility instead of brittleness, visibility instead of opacity, and the ability to upgrade data processing components independently. If you're concerned about customization or extensibility, these tools usually augment their built-in connectors with support for powerful expression languages or the ability to plug in custom code in your choice of programming languages and complex SQL queries.

2. Be Intent Driven (Minimize Schema Specification)

While it is a standard requirement in the traditional data world, developing full schema specification upfront of big data and cloud deployments leads to wasted engineering time and resources. Consuming applications often make use of only a few key fields for analysis, and, in addition, data sources often have less control over schemas that change over time and require ongoing maintenance.

Rather than relying on full schema specification, data integration systems should be intent-driven, whereby you specify conditions for, and transformations on, only those fields that matter to downstream analysis. This minimalist approach reduces the work and time required to develop and implement pipelines. It also makes dataflows more reliable, robust, and easy to diagnose.

3. Plan for Both: Streaming and Batch

Despite all of the hub-bub about streaming analytics, enterprise data is still a batch- oriented world based on applications and source databases developed over the past 30 years. So while you are building for cybersecurity, IoT and other modern applications that capitalize on streaming data, you must account for the fact that this data often needs to be joined with or analyzed against historical and other batch sources such as master or transactional data.

Rather than provisioning for a streaming-only framework, practical use cases demand that you incorporate streaming into the batch-driven data architecture while maintaining or improving performance and reliability of the overall data operations.

4. Sanitize Raw Data Upon Ingest

The original mantra for early Hadoop users was that you should store only immutable raw data in your data store.This approach spawned the “data swamp” metaphor from Gartner and others. Having data team members prune the data for each consumption activity is a common approach but an inefficient use of resources. Moreover, storing raw inputs invariably leads you to have personal data and otherwise sensitive information in your data platform, which increases your security and compliance risk.

With modern data integration systems you can and should sanitize your data upon ingest. Basic sanitization includes simple “row in, row out” transformations that normalize or standardize data formats. More advanced sanitization includes ranking, sorting, rolling averages and other time-series computations--the results of which can be leveraged broadly by data scientists and business analysts.

Sanitizing data as close to the data source as possible makes data scientists much more productive. They can focus on use-case specific “data wrangling” rather than reinventing generic transformations that should be centralized and automated.

5. Address Data Drift to Ensure Consumption-Ready Data

An insidious challenge of data management is dealing with data drift: the unpredictable, unavoidable and continuous mutation of data characteristics caused by the operations, maintenance and modernization of source systems. It can be categorized into 3 forms:

- Structural drift (changes to schema)
- Semantic drift (changes to meaning)
- Infrastructure drift (changes to data processing software, including virtualization, datacenter and cloud migration)

Data drift erodes data fidelity, reliability of your data operations and ultimately the productivity of your data scientists and engineers. It increases your costs, delays your time-to-analysis and leads to poor decision making based on polluted or incomplete data.

If your end goal is to democratize data access by having as much data as possible available to as many users as possible, then you should look for data integration tools and systems that can detect and react to changes in schema and keep the datastores in sync, or at the very least generate alerts and notifications when changes occur.

6. Architect for Cloud and Hybrid Cloud

Cloud managed services give users access to powerful tools that mask complexity and alleviate the burden of managing infrastructure and scaling. But data integration design patterns look fundamentally different when architected in the cloud. Companies often land raw data into object stores without knowing the end analytical intent. The analytics value chain needs to be reimagined as recreating architectures exactly as they are on-premises is cost prohibitive. Legacy tools for data integration often lack the level of customization and interoperability to take full advantage of cloud services. Modern data integration systems operate without the reliance on these differences, allowing users to design pipelines and port them to the environment that makes the best sense for the business. They can optimize for cost and efficiency in the cloud and still accomplish the goals of the data transformation. This can help reduce redundancy and provide insight into future cloud modernization.

7. Instrument Everything

You can never have enough visibility in a complex data system. End-to-end instrumentation of your data movement gives you a window into performance as you contend with the challenge of evolving sources and systems. This instrumentation is needed for time-series analysis of a single dataflow to tease out changes over time. Even more critical, it can help you correlate data across pipelines to gain insights in real-time.

Organizations should endeavor to capture details of every aspect of the overall dataflow architecture while minimizing overhead or tight coupling between systems. A well- instrumented approach will asynchronously communicate the measured values to external management systems, and allow you to drill down from coarse metrics used for monitoring to the fine-grained measurements ideal for diagnosis, root cause analysis and remediation of issues.

8. Don't Just Count Packages, Inspect Contents

To manage and solve for data drift you must understand the actual content as it flows through your infrastructure. Otherwise you leave yourself at risk to unannounced changes in data format or meaning. A major change in data values might indicate a true change in the real world that is interesting to the business, or might indicate undetected data drift that is polluting your downstream analysis.

An additional benefit of data introspection is that it allows you to identify personal or otherwise sensitive data transiting your infrastructure. Many industries and geographies have strict requirements around storage of personal data, such as the EU's 2018 "right to be forgotten" GDPR requirements. Continually monitoring incoming data for patterns helps companies comply, by providing real-time detection and tracking of any personal data they are collecting and storing.

9. Implement a DataOps Approach to Data Movement

The DevOps sensibility of an agile workflow with tight linkages between those who design a system and those who run it is well-suited to data operations. Data pipelines will need to be adjusted frequently in a world where there is a continual evolution of data sources, consumption use cases, and data-processing systems.

Traditional data integration systems date back to when the waterfall development methodology was king, and tools from that era tend to focus almost exclusively on the design-time problem. This is also true of the early big data ingest developer frameworks such as Apache Sqoop and Apache Flume. Fortunately, there are now modern dataflow tools that provide an integrated development environment (IDE) for continual use through the evolving dataflow lifecycle.

10. Decouple Data Pipelines from Your Infrastructure

Unlike monolithic solutions built for traditional data architectures, big data and cloud infrastructure requires coordination across best-of-breed, open source, and managed service components for specialized functions such as ingest, message queues, storage, search, analytics, and machine learning. These components evolve at their own pace, and need to be upgraded based on business needs. Thus, the large and expensive lockstep upgrades you're used to in the traditional world are being supplanted by an ongoing series of one-by-one changes to componentry.

To keep your data operation up to date in this new data world, you should use a data integration system that acts as a middleware layer, and keeps each system in the data chain loosely coupled from its neighbors. This enables you to modernize a la carte without having to re-implement foundational pieces of infrastructure.

11. Engineer for Complex Design Patterns

Not only have data pipelines become complex, but they now span a range of deployment alternatives. Industry surveys confirm that enterprises are expecting to deploy data across multiple clouds while still retaining on-premises data operations. Since each deployment option has its own advantages, it is a mistake to expect a single approach to work now and forever. Realistically, business requirements will dictate an enterprise architecture that combines many of them.

Regardless of where you are in your journey, it is best to assume a world where you have data stored in many different environments. Building an architecture based on complete "workload portability." means you can move data to the point of analysis based on the best price and performance characteristics for the job, and do so with minimal friction. Also, you should assume that the constellation that describes your multi-cloud will change over time as cloud offerings and your business needs evolve.

12. Create a Center of Excellence for DataOps

The movement of data is evolving from a train line to one that resembles a traffic grid. You can no longer get by with a fire-and-forget approach to building data ingestion pipelines. In such a world, you must formalize the management (people, processes, and systems) of the overall operation to ensure that it functions reliably and meets internal SLAs on a continual basis. This means adding tools that provide real- time visibility into the state of traffic flows with the ability to receive alerts and notifications, so you can act on issues that may violate contracts around data delivery, completeness, and integrity.

Otherwise you'd be trying to navigate a busy city traffic grid with ever-changing conditions using a static map, with the risk that the data feeding your critical business processes and applications arrives late, incomplete or not at all.

Build a Modern Data Integration Practice

The StreamSets DataOps Platform helps enterprises in building a modern data integration platform, combining high-performance execution engines and visual, full-lifecycle tools for designing, operating, managing, and optimizing data pipelines across your enterprise. With end-to-end instrumentation and visibility, the platform provides real-time operational insight across all your dataflow pipelines, no matter where they are, on-premises or in the cloud.

Key features of the platform include:

Smart pipelines to conquer data drift–Inspect data while it is in motion, and detect and resolve unexpected changes on the fly.

A living data map to conquer data sprawl–Displays all data movement on a single canvas. Its ability to auto-update brings continuous integration and continuous deployment (CI/CD) methods to data pipelines.

Data SLAs to meet data urgency–Set and enforce rules around dataflow performance to ensure that business rules for quality and timeliness are met.

Developer Productivity: Build Batch & Streaming Pipelines
Drag-and-drop UI for connecting sources to destinations with transformations. Build, preview, debug and schedule dataflow pipelines from a unified interface. Leverage pre-built origins, destinations and transformations.

Execute at Enterprise Scale
Deploy in data centers, on-prem, or on Amazon Web Services (AWS), Microsoft Azure or Google Cloud Platform and scale via YARN, Mesos or Kubernetes.

Operational Efficiency: Monitor Runtime Performance
Set alerts and notifications to track throughput, latency, error rates, and enforce data SLAs. Visualize interconnected pipelines on topology maps with the ability to drill down to individual jobs.

Protect Sensitive Data
Set policies to detect PII in-stream via pattern matching and obfuscate, hash or quarantine data for GDPR and HIPAA compliance.

Architectural Agility
Upgrade data systems with minimal downtime. Ability to detect data drift and auto-sync changed fields to downstream platforms.

To learn more about StreamSets, please visit www.streamsets.com.

ABOUT STREAMSETS

StreamSets built the industry's first multi-cloud DataOps platform for modern data integration, helping enterprises to continuously flow big, streaming and traditional data to their data science and data analytics applications. The platform uniquely handles data drift, those frequent and unexpected changes to upstream data that break pipelines and damage data integrity. The StreamSets DataOps Platform allows for execution of any-to-any pipelines, ETL processing and machine learning with a cloud-native operations portal for the continuous automation and monitoring of complex multi-pipeline topologies.

TRY NOW

Get up and running with StreamSets in minutes. Visit us at:

www.streamsets.com