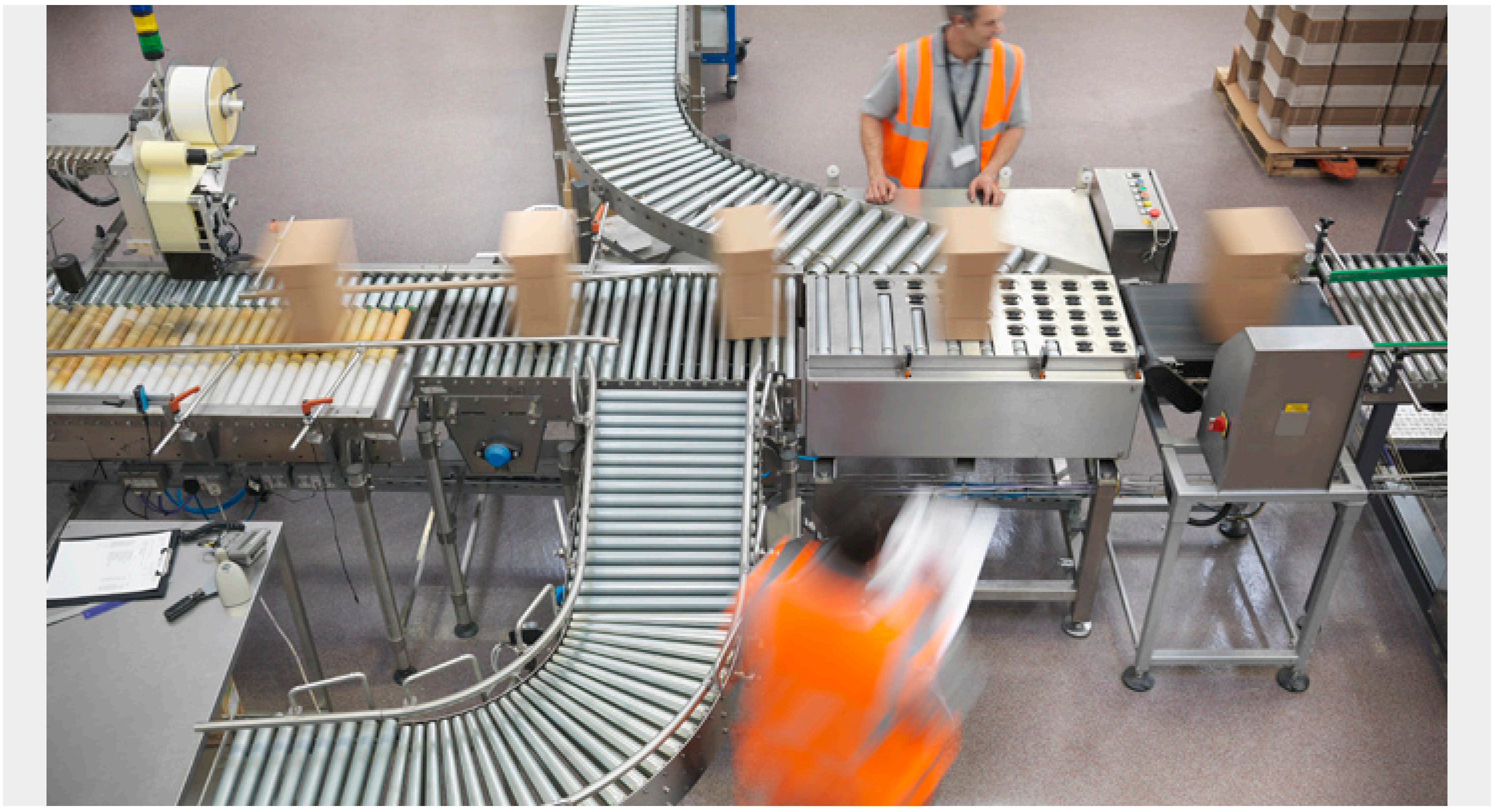
REDEFINING BATCH PROCESSING: WHY WORKFLOW ORCHESTRATION IS AS MODERN AS STREAM PROCESSING



The most common misconceptions about technology often stem from outdated content. When you search for information on batch processing, you'll find articles describing systems from decades ago—punch cards, after-hours processing, and end-of-day job runs. This framing has created a perception problem: batch processing is seen as legacy technology being replaced by modern stream processing.

Nothing could be further from the truth. Today's batch processing—more accurately called workflow orchestration—powers the most critical business services and the most cutting-edge AI workloads in production. When OpenAI recently committed $300 billion to Oracle for compute infrastructure, that investment serves batch processing for model training. The question isn't whether batch is relevant; it's why outdated terminology has obscured its modern role.

In this article, we'll reframe how batch and stream processing relate to each other, examine why the distinction matters, and provide guidance on choosing the right approach for your workflows.

The Perception Problem: What Changed and What Didn't

Classic definitions of batch processing describe after-hours work: jobs running overnight when computers were idle, processing accumulated transactions from the business day. This made sense when businesses closed at 5 PM.

But businesses no longer stop working. They operate 24/7. The kinds of processing that have to be done in batch now happen around the clock. The after-hours framing collapsed, yet the terminology stuck.

What hasn't changed is the fundamental purpose of batch processing: coordinating multi-step workflows over bounded datasets where correctness, completeness, and throughput matter more than millisecond latency. The technology has evolved dramatically—from simple job schedulers to sophisticated workflow orchestration platforms—but the "batch" label carries outdated baggage.

The Real Distinction: Why You Process, Not When

The key distinction between batch and stream processing isn't timing. It's business purpose.

Stream processing handles scenarios where data is a signal: something has occurred, and action must be taken immediately. Examples include IoT sensor alerts, real-time fraud flags, and live personalization. The characteristic pattern is a single event requiring immediate response.

Batch processing (workflow orchestration) handles scenarios where actions depend on a series of events correlated over time. This includes follow-up workflows, multi-step coordination, data completeness requirements, and compute-intensive operations like model training or financial reconciliation.

Both run continuously. Both are modern. They serve different purposes.

How Batch and Stream Work Together

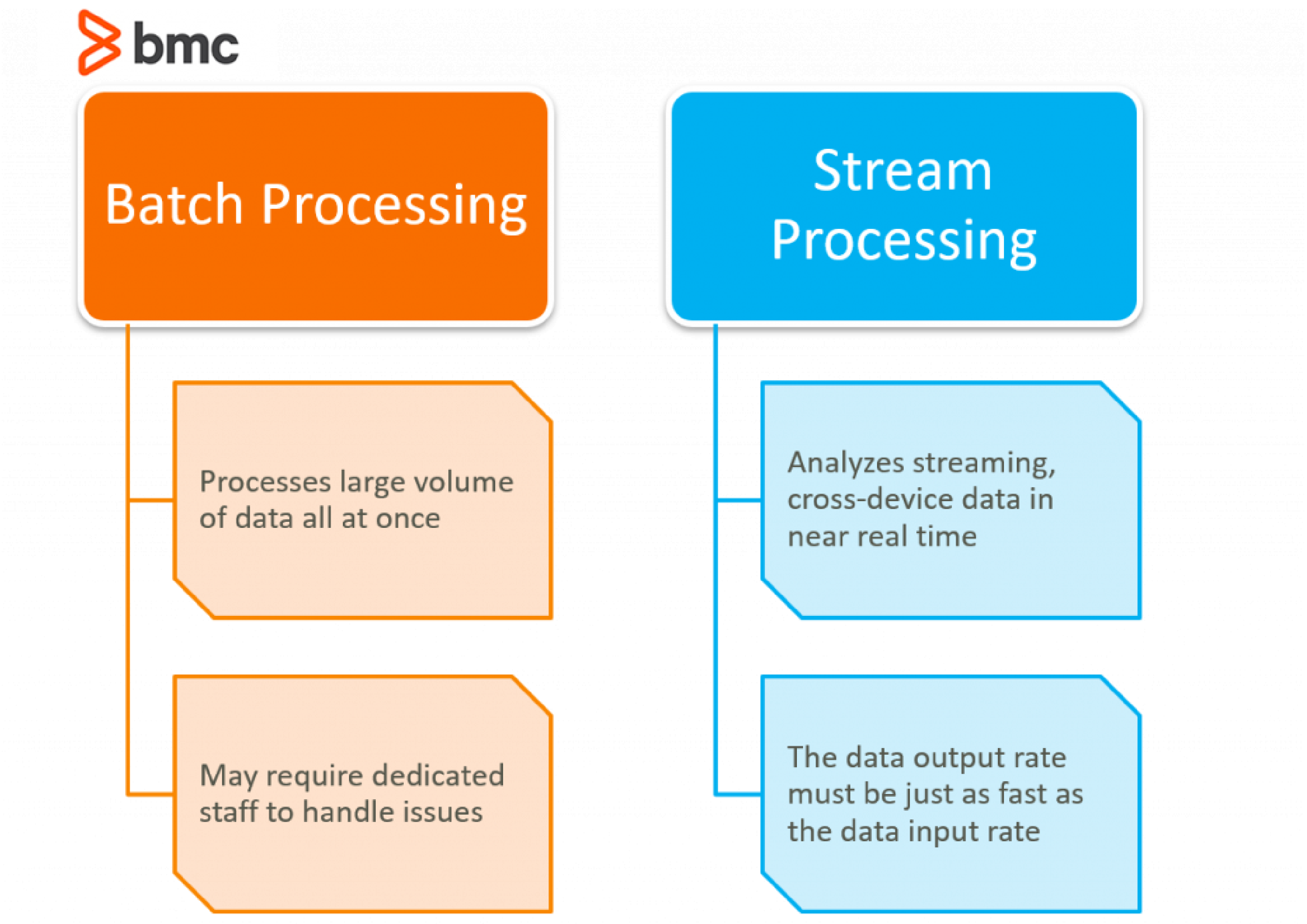
Consider fraud detection—often cited as the quintessential stream processing use case. Stream processing detects suspicious activity in real time, generating an immediate alert. But what happens next?

The system must:

- Notify stakeholders across departments
- Block affected payment cards
- Halt pending payouts or transactions
- Open investigation cases with complete context
- Reconcile all related transactions
- Update customer communications and audit logs

These multi-step workflows happen in batch. Stream detected the signal; batch orchestrates the response.

This pattern appears everywhere: stream processing identifies events requiring immediate attention, then triggers batch workflows to handle coordinated, multi-step follow-up. They're complementary, not competing.


bmc
Batch Processing
Stream Processing
Processes large volume of data all at once
May require dedicated staff to handle issues
Analyzes streaming, cross-device data in near real time
The data output rate must be just as fast as the data input rate

Benefits and drawbacks of common data processing types

When Stream, When Batch, When Both

Organizations should choose based on business requirements, not technology trends.

Use stream processing when:

- A single event (or very short window) requires immediate action
- Latency SLAs are measured in milliseconds or seconds
- You can justify the higher complexity and resource intensity

Use batch (workflow orchestration) when:

- Actions depend on correlating multiple events over time
- You need complete, validated datasets for correctness and compliance
- Workloads are compute-intensive and throughput-oriented
- You want simpler operations and better cost efficiency for large volumes

Use both when:

- A real-time signal triggers a multi-step follow-up process
- You need fast detection and reliable, auditable completion of downstream work

Stream processing is more complex and more resource-intensive than batch. If your business process relies on collecting data—even small collections—or correlating events over time, batch is often the better choice.

(Learn more about real time vs. batch processing vs. stream processing)

Batch Powers the Most Modern Workloads

The strongest proof that batch isn't legacy comes from cutting-edge technology: large language models and generative AI.

Every AI model in production today relies heavily on batch processing to train models, evaluate performance, and gather the data required for regular updates. The most modern applications—the ones dominating headlines and investment—run on batch infrastructure.

When OpenAI committed $300 billion to Oracle for compute resources, that infrastructure serves batch processing workflows. LLMs and generative AI are as modern as technology gets, yet they depend entirely on batch processing to function.

This extends beyond AI. Critical business services rely on batch processing:

- Internal services: payroll, employee onboarding, benefits administration
- Customer-facing applications: invoicing, billing, statements, notifications
- Data operations: ETL pipelines, data quality checks, reporting, analytics
- Machine learning: model training, evaluation, feature engineering, backfills

The focus isn't whether organizations need batch processing—they do. The focus is how to make batch more efficient, more automated, and capable of processing ever-growing data volumes.

Modern Operating Models: Platform Engineering and Self-Service

Organizations are evolving how they structure teams around workflow orchestration.

The emerging model separates platform engineering from workflow usage. Platform engineering teams focus on the technology itself—ensuring orchestration platforms are installed, maintained, updated, and available. Business teams (product, data, application) own their workflows and can self-serve without waiting for infrastructure tickets.

This separation allows infrastructure experts to focus on reliability while empowering domain teams to iterate quickly on business processes. Whether called self-service or something similar, the model works because it matches ownership with expertise.

Balancing Automation with Human Judgment

The goal of automation is minimizing human intervention while maintaining quality. But even in the most modern automation use cases, artificial intelligence hasn't replaced human intelligence.

There's a concept called human-in-the-loop where AI might generate several options, but humans make final decisions. This appears in content moderation, medical diagnosis support, financial approvals—anywhere risk or judgment matters.

The optimization principle is clear: ensure human intervention is utilized and leveraged when needed, but avoid relying on it when it's not. This requires identifying high-risk steps and designing checkpoints that catch issues without becoming bottlenecks.

AI assistance is evolving to make humans more effective at managing workflows: giving quicker access to institutional knowledge, suggesting root causes during incidents, and surfacing relevant runbooks. The goal isn't replacing expertise but augmenting it.

The Evolution from Simple Jobs to Workflow Orchestration

Modern batch processing is event-driven as often as it's schedule-driven. Workflows can be triggered by:

- Messages arriving in queues
- Files landing in cloud storage
- API calls from other services
- External system events
- Time-based schedules

The sophistication lies in dependency management: one step can't begin until others complete, workflows branch based on conditions, and compensating actions handle partial failures. This is far beyond simple job scheduling.

Observability is essential: per-step metrics, structured logs, data lineage, and SLA monitoring with precise alerting. Teams need visibility into what's running, what failed, and why—without manually checking dashboards every hour.

Practical Questions to Ask

If you're evaluating whether to invest in workflow orchestration or how to improve existing batch systems, consider:

On multi-step coordination:

- How do you ensure jobs complete in the correct order?
- Do you have workflows waiting to start that depend on others finishing?
- How do you track dependencies across different systems and cloud environments?

On reliability and monitoring:

- How do you know when workflows fail or take longer than expected?
- Do you manually check progress, or rely on automated alerts?
- Can you replay or reprocess workflows when issues occur?

On operations:

- Who maintains your orchestration infrastructure versus who owns business workflows?
- Could you benefit from separating platform concerns from workflow ownership?
- How quickly can teams deploy new workflows or modify existing ones?

On data and compliance:

- Do you have data quality gates before downstream processing?
- Can you audit who ran what, when, and with what data?

- Do workflows meet regulatory requirements for financial or healthcare data?

Key Takeaway

Batch processing is as modern and essential as stream processing. It's the backbone of critical business operations and the engine behind today's largest AI workloads.

The terminology may be outdated, but the technology isn't. Organizations that understand batch and stream as complementary tools—choosing based on business purpose rather than perception—will be best positioned to deliver reliable services at scale.

When the most advanced AI platform in the world commits $300 billion to batch processing infrastructure, it's time to retire the "legacy" label for good.