Whitepaper

CHECKLIST

Top 6 considerations for a real-time analytics platform

Real-time analytics: The cornerstone for powerful decision-making

All data is generated in real-time, be it, geo-location, weather reports, data from social media, and other event streams. Analyzing incoming data in real-time helps businesses take more effective decisions and accelerate response to critical events.

However, most enterprises aren't fully exploiting real-time streaming data for business intelligence. Many are unable to swiftly combine multiple streaming and static sources of data to realize 'in the moment', powerful insights.

There is a pressing need for a platform that helps and visualize high velocity data from multiple disparate live data sources, and unify these insights with data from batch sources. This white paper focuses on how a next-gen analytics platform can help identify simple and complex patterns to detect strategic opportunities and act in real-time.

> "End users can harness increasingly sophisticated analytic capabilities through packaged real-time analytics embedded into data analytics tools and applications without prohibitive processing wait times or the need for developers to intervene.
>
> https://www.gartner.com/smarterwithgartner/six-best-practices-for-real-time-analytics

Gartner®

Key attributes of a next-gen real-time analytics platform

1. Unified platform for big data processing

A real-time analytics platform must enable continuous analysis of data in motion across all stages of data processing:

Data ingestion

Data ingestion Ability to natively connect with multiple, disparate data sources and storage systems. These should include message queues (like Apache Kafka, RabbitMQ, Amazon Kinesis, and Azure Event Hubs), indexing stores (like Elasticsearch and Solr), cloud data stores (like Amazon S3), distributed file systems, NoSQL databases (HBase and Cassandra) and relational databases (Oracle and MySQL).

Change Data Capture

Ability to capture changes made in a database, and replicate those changes to a destination such as a data warehouse in real-time or near real-time.

Data transformation

Ability to perform in-memory data processing and apply transformations to filter, cleanse, blend, and enrich data at scale.

Analytics

Ability to take analytical actions such as complex event processing, aggregation, grouping, and correlation. Users should be able to run statistical and predictive models as well as geospatial analytics.

Data visualization

Built-in or custom dashboards to visualize real-time insights.

2. No-code, visual UI

A no-code/low-code platform can dramatically accelerate the time taken to build and operationalize reliable data pipelines for real-time analytics.

Easy-to-use, visual elements like a drag-and-drop canvas and built-in ETL and ML operators support rapid pipeline development, deployment and management. They help both tech and non-tech users work 10x faster compared to hand-coding.

However, though a drag-and-drop interface considerably accelerates time to insight, the demand for custom applications has never been higher. A real-time analytics platform must minimize hand-coding yet enable easy integration of hand-written custom logic into data pipelines for complex use cases.

3. Real-time, near real-time, and batch processing

A modern real-time analytics platform must offer both batch and streaming workflow orchestration. This can help users build predictive/ machine learning models in batch mode periodically and transfer them to the real-time data "circuit" to speed up decision-making with real-time scoring.

Using underlying high-performant engines like Apache Spark can solve the complexities of real-time, near-real-time, and batch processing. It includes functions such as MapReduce as well as models like recursive computing, graph mode calculations, etc.– all at 50-100X faster processing speed. Spark uses an asynchronous integration mode in near real-time called 'structured streaming' and collects streaming data and events together for processing in 'micro batches.' It expresses streaming computations identical to batch computation on static data, by running it incrementally and continuously.

'Real-time' can mean different time frames for various business use cases. Not all events require action in a time window of milliseconds to seconds. For instance, an e-commerce recommendation engine requires acting in milliseconds to offer an immediate personalized digital experience to the customer. However, this time window changes to minutes in the context of refreshing retail sales analytics dashboards. This 'near real-time' processing scenario involves combining data from multiple sources to detect patterns.

> Many enterprises leverage the 'build once and deploy as both batch and streaming jobs' feature of Spark's structured streaming APIs to run identical code in both modes with minimal changes.

The following figure showcases a Lambda architecture that enables both batch and stream processing methods. This approach attempts to balance latency, throughput, and fault-tolerance using batch processing, while simultaneously using real-time stream processing to provide views of online data; enabling 360-degree visibility.

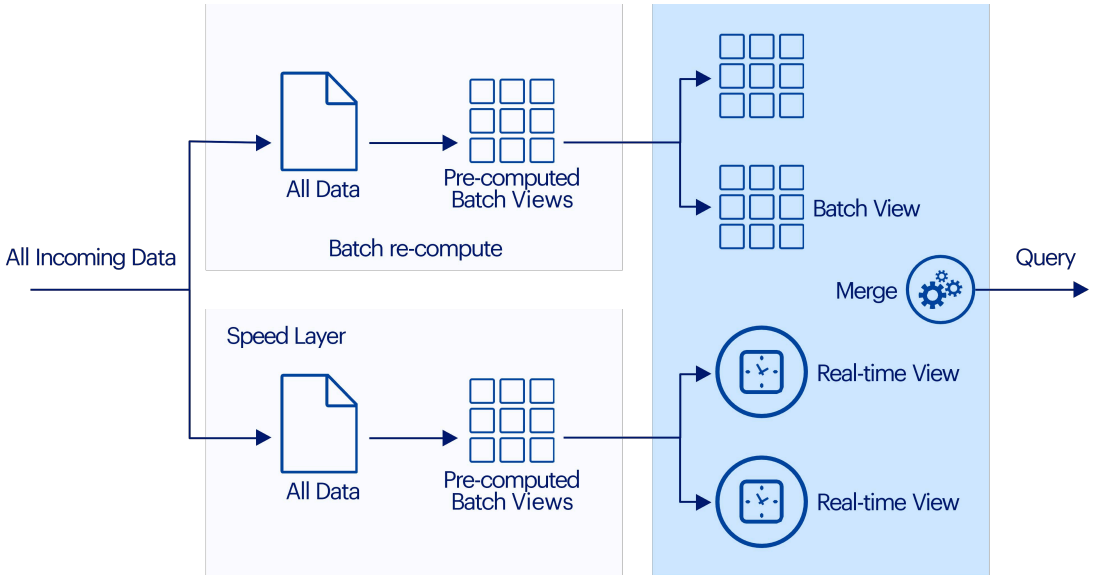

Figure 1: Big and fast data combined in Lambda architecture

4. Advanced analytics and machine learning capabilities

With the ever-growing velocity, volume, and variety of incoming data, businesses are looking for a unifiedplatform that offers everything needed to enable advanced analytics. Real-timeanalytics solutions must enable natural language processing, anomaly detection, predictive modelling, CDC, ETL/ELT, machine learning, and more. A single platform with batch and streaming work flow orchestration helps data teams build machine learning models on static data and train and score machine learning jobs on real-time data or any combination of training and scoring modes.

To enable modularity and reuse of analytical and data processing pipelines, users should be able to string together different sub-pipelines to create larger or more complex streaming pipelines. The eventual goal is to rapidly launch applications to detect crucial business events and react in real-timeto complex event streams or feed real-time dash boards for use cases such as anomaly detection, churn prediction, hyper-personalization, cybersecurity, and fraud and risk detection to name a few.

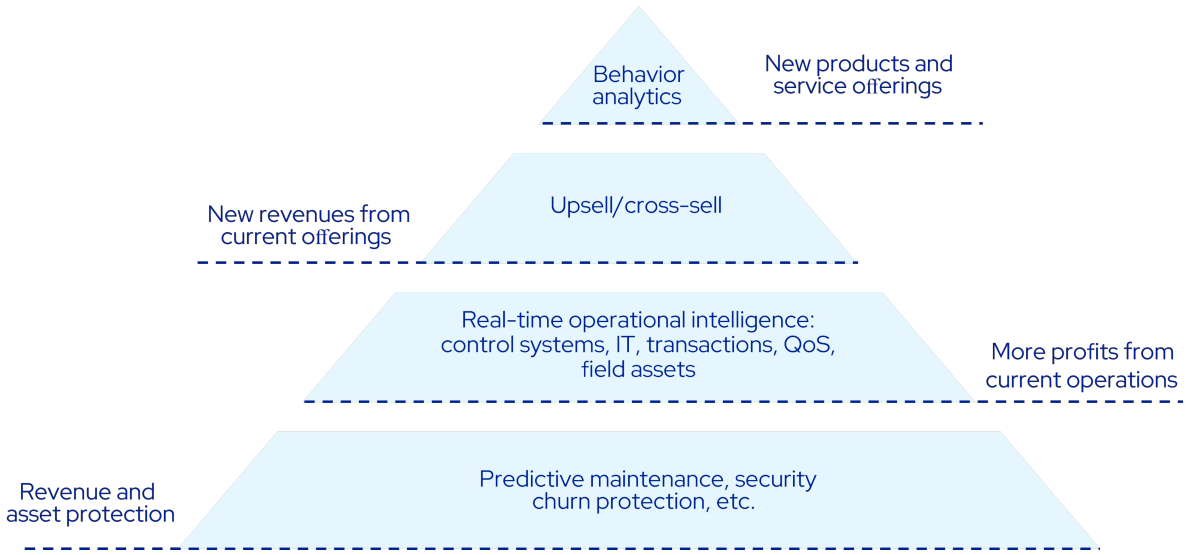

Figure 2: Business outcomes of applying machine learning models on streaming data in real-time

5. Application lifecycle management

The time-to-market advantage of no-code development vs. hand-coding is diminished if the plat form does not facilitate as seamless way to move real-time applications along the life cycle .No-code development plat forms must there fore also provide an Integrated Development Environment(IDE)that supports the entire application delivery lifecycle–design, build, test, deploy, and manage.

Besides visual development tools, the platforms must include one-click deployment options, application governance tools such as data inspection (iterative debugging while building by viewing results with test data injection), data lineage (run-time audit of the complete journey of a data record), and smart alerting and monitoring capabilities.

6. Open, cloud-agnostic technology

The modern data technology landscape is rapidly evolving, especially with significant contributions and active community support from the open source world. A real-time analytics platform must offer abstraction over this mix of complex technologies, enabling users to focus on the underlying business logic.

An open, interoperable, cloud-agnostic platform helps avoid vendor lock-ins and enables future readiness. Today, there are many open source stream processing engines available, including Apache Storm, Apache Spark, and Apache Flink. These are designed for different business requirements and limiting a user to any single streaming engine, open source or proprietary can constrain analytics use cases.

A multi-engine streaming platform which enables connecting different pipelines with sub-system integration across compute engines can provide a unified solution for diverse business needs.

The platform must also offer flexible portability to deploy and manage your applications across public, private, and virtual private clouds, or on-premises.

Scan and
start free 14-day trial

Scan to
schedule a demo