Real-time data pipelines: pairing message queues and databases

Orenstein, Gary . InfoWorld.com ; San Mateo (Mar 29, 2018).

ABSTRACT (ENGLISH)

Real-time data sources Modern applications drive real-time data in areas such as: * Finance, for up-to-the-second portfolio analytics * Media, for keeping track of streaming analytics * Retail, to develop customer 360 perspectives * Energy, for predictive IoT analytics In addition, data may come from existing operational databases inside a company and combine with multiple sources. [...]a traditional data pipeline might be Application data >Database >ETL >Data Warehouse >BI Dashboard Moving to a real-time pipeline we no longer have the luxury of a traditional ETL step, nor do we have the pain that goes hand in hand with managing batch ETL processes. Kafka >(transform in DB) Database >BI Dashboard Delivering the right application data infrastructure New applications demand the ability to handle the fast loading of live data, analytics that cover real-time and historical data, and the ability to support a large concurrent user base.

FULL TEXT

In a connected world, real-time data pipelines power applications and insights, providing the digital infrastructure for active data. This helps data-driven companies understand how their customer base is behaving in the moment, especially when it involves live operations. With an ongoing pursuit of new digital growth opportunities, data leaders need to stay on top of these leading architectures.

A stereotypical real-time data pipeline might look as follows:

Real-Time Data Source >Message Queue >Database >Application

Data sources and applications can be unique to specific industries. Choices for message queues and databases apply horizontally and are the primary focus of this article.

Real-time data sources

Modern applications drive real-time data in areas such as:

* Finance, for up-to-the-second portfolio analytics
* Media, for keeping track of streaming analytics
* Retail, to develop customer 360 perspectives
* Energy, for predictive IoT analytics

In addition, data may come from existing operational databases inside a company and combine with multiple sources.

Picking the right message queue

Many choices exist for message queues across both cloud services and software offerings. Message queue considerations include the following.

Deployment choices

For cloud offerings such as AWS Kinesis, the only option is to deploy on the cloud provider's infrastructure. For software such as Apache Kafka, options are available to deploy anywhere, including your own datacenter. A supported enterprise option for Kafka is available from Confluent. There are other message queue choices as well, such as RabbitMQ and ZeroMQ.

Instrumenting a large manufacturing facility with a messaging backbone for predictive analytics would point to an

on-premises focus. A company-wide contract with a major cloud provider, and push button integration with that cloud ecosystem, could lead to another choice.

The right semantics for your application

Message queues support a range of semantics with “at least once” and “exactly once” being popular choices.

“At least once” means messages are sent once, often more than once; messages are guaranteed to arrive, and there will be duplicates on the other end.

“Exactly once” means message are sent once, sometimes more than once; the message is guaranteed to arrive, and duplicates are eliminated on entry to the receiving end. The “exactly once” refers to the destination being duplicate-free.

Each of these modes provide benefits and drawbacks based on your application requirements. Apache Kafka supports both semantics; whereas AWS Kinesis focuses on only “at least once” semantics. AWS has another message queue, SQS, that supports “exactly once” semantics, but it is less focused on the real-time, high-volume pipelines that gravitate to Kinesis.

Choosing the database

For databases, there are too many choices to list. But the following criteria help ensure you pick the best option for real-time data pipelines.

Native high-speed ingest

Pick a database that supports high-speed ingest and scales out as a distributed system. Some databases provide native ingest from Kafka or Kinesis. Databases should support the ability to ingest data in different data types, including JSON.

Reduce data duplicates with an operational datastore

Operational databases that support transactions, including commands such as INSERT, UPDATE and DELETE, allow for simple data deduplication. This pairs well with a message queue that may not natively support exactly once semantics, such as AWS Kinesis.

Real-time and historical analytics with SQL

Databases supporting SQL provide immediate sophisticated analytics to live and historical data. Business units can also see detailed reporting across a variety of dimensions. This occurs without having to move the data between systems or rely on another software layer for analytics. For example, landing your data in a NoSQL database often means another tool is required to access that data via SQL.

Architecting without the luxury or pain of ETL

ETL stands for extract, transform, and load, the quintessential glue for moving data between systems. However, a perennial maxim for data architects is “no data movement” because data movement introduces latency and complexity to any pipeline. Real-time pipelines, when constructed with care, help data engineers reduce latency, contain data proliferation and limit sprawl. Furthermore, these pipelines can process rich transformation functions continuously instead of in batch mode.

For example, a traditional data pipeline might be

Application data >Database >**ETL** >Data Warehouse >BI Dashboard

Moving to a real-time pipeline we no longer have the luxury of a traditional ETL step, nor do we have the pain that goes hand in hand with managing batch ETL processes.

One option is to make use of popular transformation engines like Apache Spark. Spark has the speed and scale to handle continuous processes in place of traditional batch ETL. Spark in the pipeline offers this real-time transformation ability. Coupled with a database that handles real-time and historical analytics, the pipeline, including transformations, becomes quite simple.

Application Data >Spark >Database >BI Dashboard

Of course, this pipeline could use a message queue like Kafka as well:

Application Data >Kafka >Spark >Database >BI Dashboard

Finally, some databases allow for transformation upon ingest, such as running a function in the database itself.

That can simplify the pipeline further:
Kafka >(transform in DB) Database >BI Dashboard

Delivering the right application data infrastructure

New applications demand the ability to handle the fast loading of live data, analytics that cover real-time and historical data, and the ability to support a large concurrent user base.

By picking the right combination of message queue and database, architects will put themselves in the best position to handle these application requirements.

This article is published as part of the IDG Contributor Network. Want to Join?

Credit: Gary Orenstein

DETAILS

Subject:	Software; Architects; Data bases; Infrastructure; Queues; Semantics; Pipelines
Publication title:	InfoWorld.com; San Mateo
Publication year:	2018
Publication date:	Mar 29, 2018
Section:	opinion
Publisher:	Infoworld Media Group
Place of publication:	San Mateo
Country of publication:	United States, San Mateo
Publication subject:	Computers--Microcomputers, Computers--Computer Industry
Source type:	Trade Journals
Language of publication:	English
Document type:	Opinions
ProQuest document ID:	2019689655
Document URL:	https://search.proquest.com/docview/2019689655?accountid=10472
Copyright:	Copyright Infoworld Media Group Mar 29, 2018
Last updated:	2018-03-30
Database:	Business Premium Collection

LINKS

Check SFX for Availability