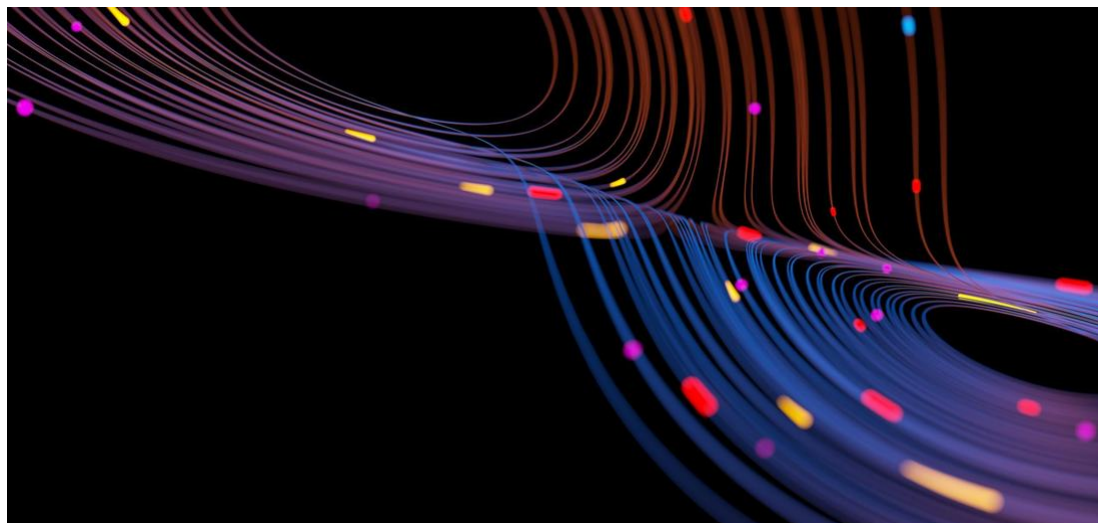


Real-time big data analytics: use cases and implementation options

Originally published at <https://www.itransition.com/blog/real-time-big-data-analytics>

Published: November 1, 2021 | By Martin Anderson Independent AI expert



Business intelligence has traditionally been an 'off-line' pursuit, where historical company data was periodically crunched into regular reports. Consequently, actionable information on trends and possible issues tended to lag behind; when critical trends finally came into focus, it was often too late for timely intervention.

By contrast, real-time big data solutions, constituting a segment of [big data services](#) at Itransition, offer 'live' views of critical corporate information flows across a number of possible applications, including sales figures, marketing reach, traffic spikes (also invaluable as an adjunct of cybersecurity), the tracking of internal metrics for staff performance, high-speed volatile markets (such as stock trading), the disposition of deployed fleets (ranging from military oversight all the way to ride-sharing services), and fraud detection, besides many other possible uses.

Offering a less 'archived' view of your business intelligence is a growing trend; Gartner [predicts](#) that by 2022 over half of major new business systems will incorporate real-time data analytics.

In this article, we'll take a look at some business use cases, potential benefits, and ways to implement real-time big data analytics

Business use cases

There are more use cases for real-time analytics than we can reasonably examine here; it's been estimated that in smart manufacturing alone, there are at least [865 applicable areas](#) for business analytics, the majority of them (such as power analysis) needing to operate and report anomalies in real time. Nonetheless, let's look at some popular examples across sectors.

Web monitoring

Digital service providers cannot wait for a historic post-mortem when data anomalies occur. For instance, providers of network infrastructures, such as Content Delivery Networks (CDNs), network providers and cybersecurity services, need immediate access to information about new downtime events in order to enable rapid response and amelioration.

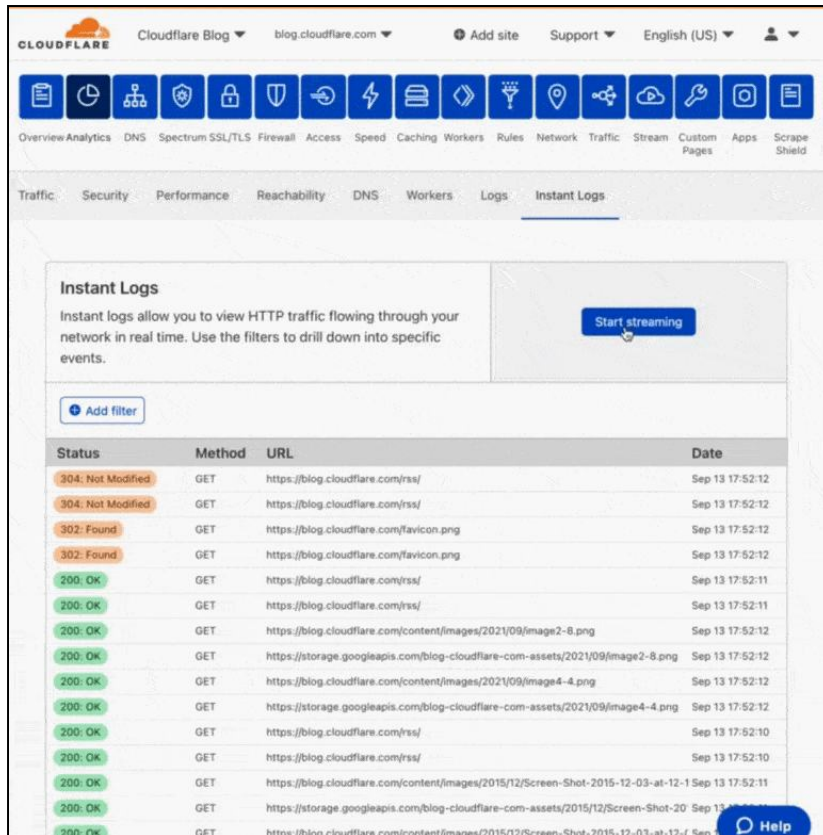
As one of the biggest operators in the CDN space, CloudFlare was among the earliest proponents of real-time big data analytics, both in terms of internal security and performance analysis tooling, and as a revenue stream in real-time dashboards for its paying customers.

In September of 2021 the company introduced 'Instant Logs' into its live-updating analytics console, offering 100% real-time reflection of data at a commercial scale to analytics subscribers.

CloudFlare Instant Logs



Since September 2021, CloudFlare analytics customers have been able to receive 'pushed' logs as real-time data in the analytics console, allowing a user to select a brand-new event and drill down into its characteristics.



Data source: blog.cloudflare.com—Data at Cloudflare just got a lot faster.
Announcing Live-updating Analytics and Instant Logs, 2021

The 'Instant Logs' functionality was in fact already built into CloudFlare's infrastructure but entailed a great deal of setting up for customers. The innovation here was to integrate the company's Logpush system into a simple navigable strand in the interface. In this way, a client can instantly investigate anomalies such as 404 errors (missing pages) or traffic spikes that can result from advertising, malware or DDoS attacks, and take relevant action before the window of opportunity closes.

Financial markets monitoring

Until the advent of real-time cybersecurity monitoring frameworks and seismic monitoring systems, keeping up with stock market prices represented perhaps the most intensely-studied and implemented of real-time data systems. The volatility of trading markets requires instant notification of stock fluctuations, both in service of automated trading algorithms powered by [stock market machine learning models](#), and to keep humans in the loop.

In terms of live analysis, rather than the simple reporting of changed values, real-time streaming financial applications offer a particular challenge to data systems developers, since the flood of data as markets open and close across the world is not consistent across a working day either in volume or origin.

Therefore, it can be necessary to carry out [risk analytics](#) based on pared-down historical patterns of market trends and behavior, and also to ensure that anomalous new events are not ignored simply because they do not fit previous event trends.

In 2005, a study titled *The 8 Requirements of Real-Time Stream Processing* led by MIT established eight core challenges affecting real-time streaming solutions for financial trading, which have changed little since:

- *Keep the data moving* — ensure continuity of analysis.
- *Provide SQL interpretability of data* — allow real-time lightweight structured query calculations on live data.
- *Handle stream imperfections* — include fault tolerance for data source outages or non-standard, unexpected errors in output.
- *High availability* — ensure minimal or zero downtime, since lack of availability is a destructive process for financial markets.
- *Stored and streamed data* — enable injection of the latest data into more historical data while allowing for its interpretation related to older data.
- *Distribution and scalability* — make systems resistant to unexpected growth of sources and/or data.
- *Instantaneous response* — be able to deliver very low-latency reporting and calculations across data that may include millions of points.

A streaming framework requires adequate bandwidth, low latency, and appropriate interpretive and analytical architectures that can corral these elements into a truly responsive dashboard or API. This is far from the exclusive preserve of major tech SaaS providers, but rather can be enacted by smaller teams with open-source software and appropriate bandwidth provision.

Ways to implement real-time big data analytics

Here we'll look at some of the core requirements for real-time big data streaming and compare 'off-the-shelf' hyperscale SaaS offerings to the way that open-source software can enable highly effective in-house data streaming implementations.

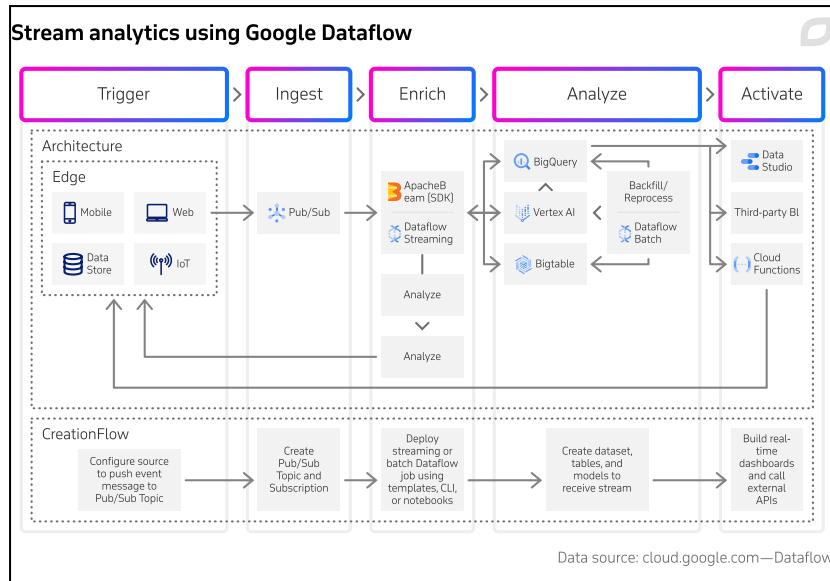
Real-time stream processing

The majority of data in a real-time big data analytics framework is historical. How far back the data goes depends on the configuration that you make available to your end users and the extent to which your back-end architecture is capable of handling high-volume data (or is capable of extracting broad trends from older data in order to lighten the processing load) while keeping a dashboard responsive.

The development of a real-time data reporting system depends on being able to 'inject' up-to-the-minute events into this tranche of older data. This is known as real-time stream processing (RTSP) and, unsurprisingly, the largest tech giants are in the vanguard of the technology.

FAANG streaming offerings

Google Cloud offers Dataflow, the core technologies of which have been powering the search giant's Google Analytics framework for 16 years. A typical pipeline application might involve as many as a dozen other frameworks and filtering mechanisms (see image below), as well as load-balancing systems to ensure that the data is adequately current.



Though Google itself has commented that RTSP must be capable of dealing with 'hundreds of millions of events per hour', this should be considered in light of the company's hyperscale customer base; only a very poorly-optimized typical corporate network would usually have to bear such a brunt.

Amazon sells a similar service with Amazon Kinesis, which offers metrics extraction and real-time KPI generation, among many other features, most of which are also offered (or can be engineered) in competing products.

Microsoft Azure has a wide array of RTSP solutions, including Azure Stream Analytics, HDInsight with Spark Streaming, and several other mix-and-match architectures accommodating a variety of programming languages, including C#/F#, Java, Python, R and Scala. Therefore, a company setting out to create an analytics platform from scratch has the luxury of choosing a performant programming language, in which the company has existing expertise, with a service that caters to the language.

Regarding cost: as we have noted many times on this blog in comparing FAANG SaaS services, the capabilities and features of hyperscale cloud service providers are difficult to directly compare, since they are not equally distributed or equally subdivided into layers of utility.

Additionally, the metrics for billing (and even units of usage) are not always directly comparable either, and Microsoft Azure's Stream Analytics pricing page illustrates a truth applicable to all FAANG RTSP providers — costs will depend on the country in which the framework is operating as well as its crossover into other jurisdictions (a likely scenario for global operators).

Open-source stream processing

The FAANG corporates offer something close to on-demand functionality for a company's RTSP needs, but, as with other sectors of cloud infrastructure, their chief attraction remains their enormous network and processing capacity. No matter which way you navigate the pricing calculators of Google, Azure and Amazon (among many others), there's inevitably a price premium attached to this rentier model.

If your company can anticipate capacity and has well-defined targets for growth, it's possible to bypass the major cloud providers and develop in-house real-time streaming systems using open-source repositories — in most cases, the same repositories that the FAANG offerings use.

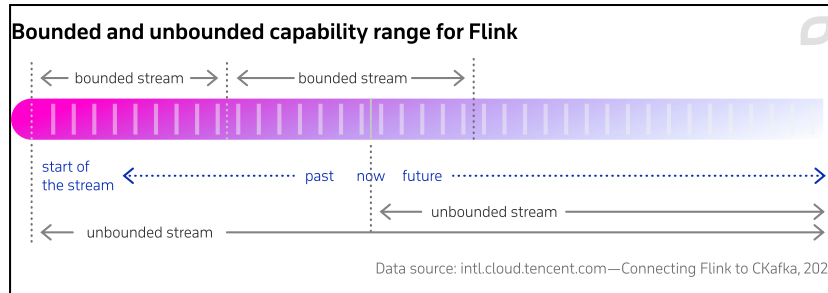
Apache

Apache open-source server offerings are dominant in rolling your own real-time streaming, with several prominent contenders for local custom build systems.

With origins as an academic project in Germany, **Apache Flink** is now a top-level Apache project, offering RTSP at blistering update speeds and traversing vast amounts of historical data in real time at extremely low latencies. It's also one of the most resilient and scalable FOSS real-time streaming solutions, and can be set to only process brand new data, making its information flows even more responsive.

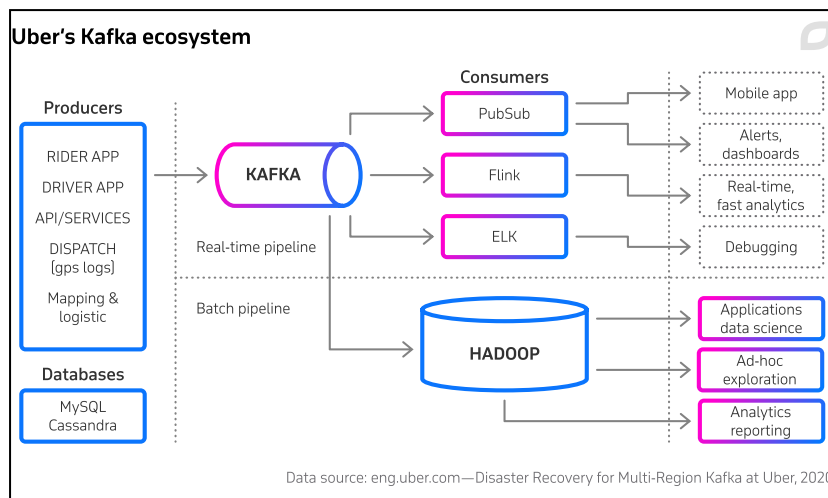
Flink can handle bounded and unbounded data, which means that it can either complete mission-oriented projects or create a perpetually updating live-streamed data environment. It's a good adjunct to Kafka (see below) and a variety of other frameworks, including open-source Hadoop, and

can be implemented in a range of languages, including SQL, Scala, Java, and Python.



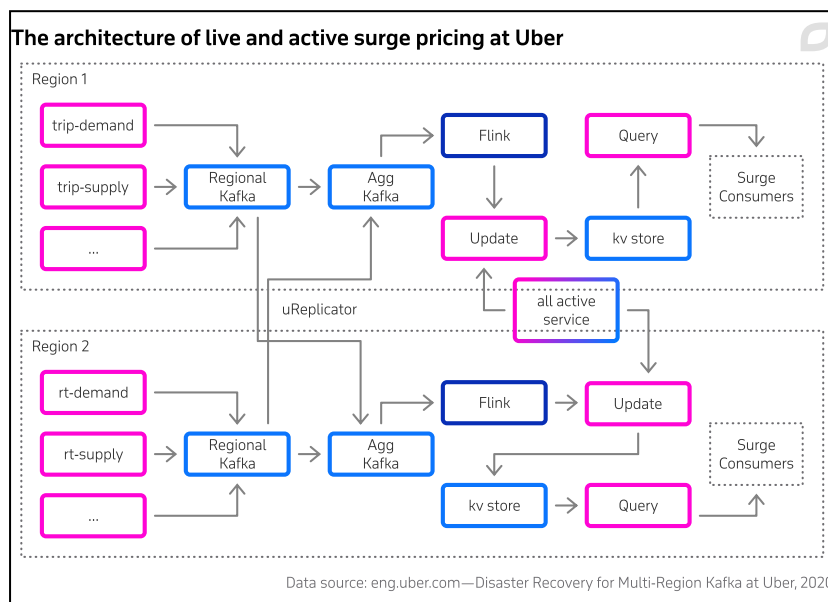
Flink constitutes part of some of the most ambitious and wide-reaching global data frameworks and is used by eBay, Huawei, Pinterest, Uber (also see below), AWS, Tencent, Capital One, and Alibaba, among many others.

Written in Scala and Java, **Apache Kafka** was open-sourced from a LinkedIn project in 2011, and has since developed a wide user base, even including some of the biggest tech service providers. Uber describes its extensive use of Kafka as the 'cornerstone' to its technology stack, with the framework responsible for passing and receiving millions of messages daily between drivers and clients.



Additionally, Kafka powers Uber's internal streaming analytics systems, handles data ingestion to the global ride-share giant's Hadoop database, and streams changelogs to key players further down the monitoring and maintenance hierarchy at Uber.

Perhaps most crucially, this impressive open source solution also powers Uber's famously flexible price system, allowing for cheaper rides at times of lower demand and for drivers to benefit from 'surge' pricing when demand outstrips supply.



Kafka is implemented at Uber with region clusters that feed on to aggregate clusters capable of delivering a global overview of current activity across the world in areas where Uber operates. Redundancy is handled by a custom fork of Kafka's MirrorMaker package.

Kafka is also used by LinkedIn to track operational metrics and report on *seven trillion* messages passed each day, while Twitter pairs Kafka with Apache Storm to create a stream-processing framework. Kafka is also a great solution for [IoT use cases](#), with recent versions offering disaster recovery and reduced dependency on Java.

Other offerings from Apache's powerful stable of RTSP offerings include **Apache Storm**, a real-time computation system that can operate at a million tuples per second per node, and which effectively offers MapReduce operations on a live basis, running on top of Hadoop YARN; and **Apache Samza**, a distributed stream processing framework that's derived from Kafka and YARN, and which offers rapid throughput for a number of high-profile companies, including CDN provider Optimizely, Redfin real estate brokerage, Slack and TripAdvisor, among others.

Do it yourself?

There has only been scope in this article to overview the top FOSS live streaming solutions, but each come with community resources and a range of extending software and projects, and each can be customized to a company's needs (as many of the above-mentioned companies have done) without the appellant expense associated with high-volume commercial cloud-based framework solutions.

While avoiding FAANG provisioning for architectural infrastructure for live data streaming, it should be considered that you often can't bypass the giants in terms of network architecture, particularly for dominant players such as AWS, whose architectural solutions will streamline this process in a way that's more of a challenge to replicate at an independent level. In-house projects will need greater upfront development resources and a willingness to follow what can be a steep learning curve in terms of load balancing and latency development.

Nonetheless, custom solutions provided by data analytics consulting companies represent the difference between owning and renting your business intelligence architecture — with all the implications for ultimate profitability that's associated with dedicated local solutions.