White Paper

AEROSPIKE

Unleash the power of real-time data across your data pipelines

Contents

Executive overview 2

Background 2

Leveraging streaming data with Aerospike 5

- Aerospike platform overview 5
- Introduction to Aerospike's streaming technologies 6
  - Stream transport services 7
  - Stream processing services 8
  - Change notification services 9
- Aerospike usage patterns for streaming data 10
- Kafka and Pulsar connectors 11
- Event Stream Processing (ESP) connector 12
- JMS connector 14

What customers are saying 15

Summary 17

About Aerospike 18

Executive overview

For the past two decades, the proliferation of mobile devices, social media, e-commerce, and sensors has fueled the tremendous growth of streaming data. Once confined to a narrow set of industries, streaming data is now pervasive worldwide, and market researchers anticipate significant increases for the foreseeable future. According to one analysis, the global streaming analytics market will grow from $12.5 billion in 2020 to $38.6 billion in 2025.

Collecting, filtering, and analyzing streaming data is vital in today's highly demanding business landscape. Leveraging these vast amounts of data to make more accurate, timely business decisions can yield substantial tangible results. This requires an infrastructure that integrates disparate data sources and platforms in real time and makes that data available to applications without delay.

Aerospike has developed critical technologies to help firms process, persist and protect petabytes of streaming data on modest server footprints. Aerospike's highly performant, scalable interface to streaming and messaging platforms enables its real-time data platform to ingest massive data volumes captured at the edge to power operational and transactional applications. Users can combine this streaming data with other transactional and historical data stored in Aerospike. In addition, Aerospike's *change data notification* capabilities promote the sharing of new or modified data with downstream platforms.

The Aerospike Real-time Data Platform architecture features massive parallelism, support for modern hardware architectures, intelligent workload processing, strong data consistency, and self-managing features. In addition, Aerospike's support for Trino (PrestoSQL) and Spark enables SQL programmers to access streamed data stored in Aerospike for immediate query and analysis.

This paper explores how Aerospike integrates with popular streaming and messaging platforms to promote rapid consumption and processing of streamed data *(such as sensor data, clickstream records, log records, financial trading data, utility consumption data, geo-location data from mobile devices, and more)*. In addition, you'll learn how Aerospike can quickly ingest and store data streamed from multiple sources and scale from modest to massive volumes on much smaller server footprints than other platforms.

Background

Integrating streaming data (sometimes referred to as "data in motion") with other forms of enterprise data helps firms enrich the analysis of business events. This can lead to improved operational efficiency, reduced costs, and better data-driven business insights in real time.

Streaming data usually consists of relatively small data records transmitted in real-time. Typical streaming data sources include sensor and device data, log records, financial trading data, geospatial information, gaming activities, and website clickstreams. The volume, structure, and rate of data generation vary by source; in some cases, seasonal or sporadic fluctuations in data volume and velocity are common.

Once the domain of specialized systems and applications, streaming data is increasingly treated as an integral and essential form of data to be collected, processed, curated, protected, and integrated within a firm's broader IT infrastructure. Figure 1 presents the unique challenges from a data management perspective: spanning scalability, performance, data modeling, availability, and consistency.

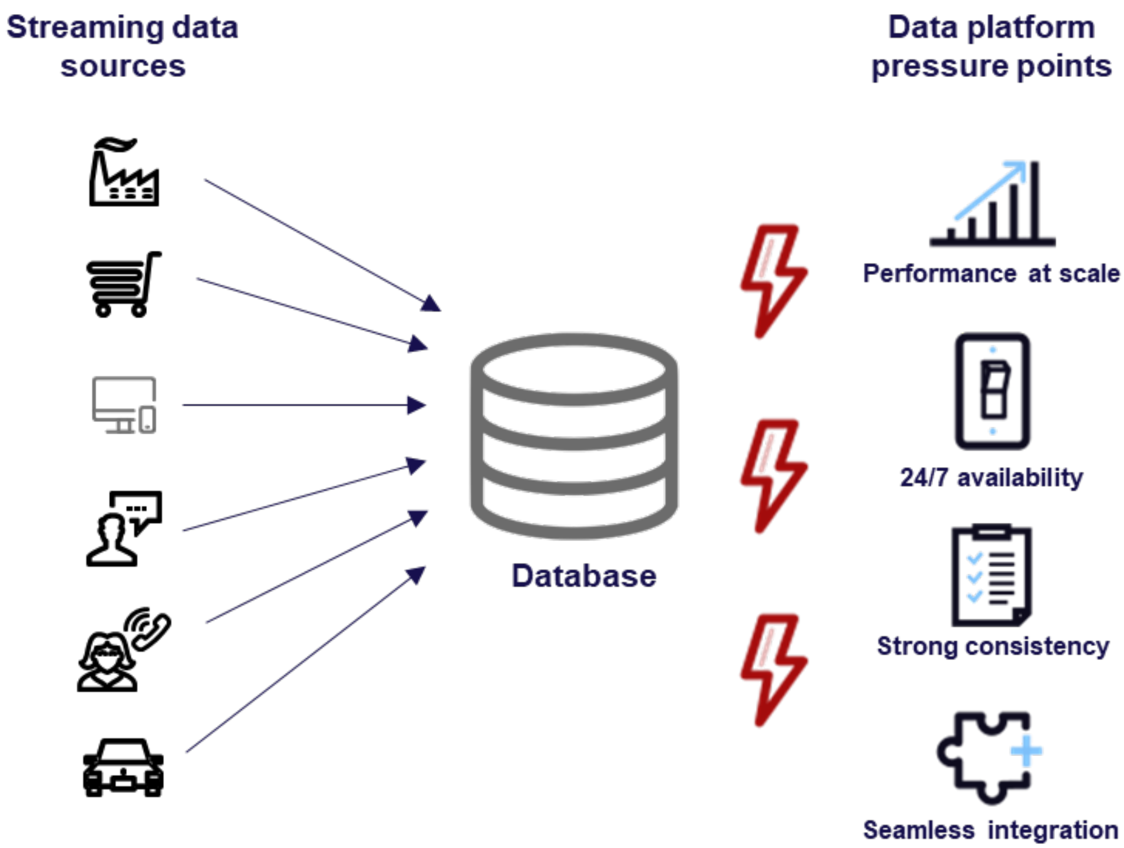

Figure 1: Streaming data can easily stress a data management platform

Let's step through several common pressure points that arise with data management platforms when firms seek to treat streaming data as a first-class citizen.

Data volumes and velocity. The sheer quantity of streaming data, the high rates at which it is generated, and its continuous (i.e., never-ending) nature demands ultra-fast and predictable performance at scale from any database platform tasked with ingesting, storing, and processing it. To keep costs manageable and avoid server sprawl, the platform must efficiently exploit hardware, software, and networking resources.

Scalability and elasticity. With data volumes and the number of sources of streaming data projected to increase over the next decade, scalability is mandatory for any data platform tasked with persisting streaming data. Some platforms perform well at modest levels of scale (hundreds of gigabytes to several terabytes) but struggle or fail at higher levels of scale. As a result, data latencies often rise rapidly, server footprints can sprawl to hundreds or thousands of nodes, and operational costs can skyrocket. Elasticity is also essential, as firms need to rapidly and cost-effectively scale up or down as business needs or market conditions change.

Availability. The continuous nature of streaming data demands a data platform that's always on. Reliability, automated failure recovery, and 99.99% uptime (or better) are essential. Routine operations - such as software upgrades, the addition or removal of clustered computing nodes, rebalancing of data, and so on - must occur without taking the platform offline or leading to unexpected data access latencies. To maintain reasonable operational costs, the platform should provide substantial self-managing features.

Strong Consistency. For event processing and transactional applications, the ordering of streamed data is critical. Conflicting write operations or reads of stale (dirty) data simply aren't acceptable in many business contexts. Data platforms that offer strong, immediate data consistency are important to address such requirements.

Platform integration. Maximizing the value of streaming data calls for a data platform that integrates with the most common components of your streaming data pipelines, such as Spark, Kafka, Pulsar, etc. These integrations should promote transparent, high-performance dissemination of data to target applications in the enterprise. Because different streaming data sources employ multiple data structures and formats, the underlying data platform should offer flexible data modeling options.

These data platform requirements may seem daunting, and IT organizations often struggle to find a solution. But firms that employ Aerospike have found they're able to meet unusually ambitious goals, including those that have impeded other offerings. In addition, Aerospike has proven to deliver exceptional performance at petabyte scale on remarkably small server footprints and with 99.999% (five nines) availability. A 2021 benchmark report and a separate white paper on Aerospike's resiliency explain those aspects of the platform in detail.

Leveraging streaming data with Aerospike

Delivering extremely low latency read/write operations on massive volumes of real-time data has been a hallmark of Aerospike's technology for years. Let's explore some of Aerospike's critical features that help firms leverage streaming data efficiently, cost-effectively, and seamlessly.

Aerospike platform overview

Aerospike is a distributed multi-model data platform that secures and delivers ultra-fast, predictable read/write access to billions of records, holding up to petabytes of real-time operational data. Firms worldwide support systems of engagement and systems of record 24x7, often saving $1 - $10 million per application compared with other approaches.

Aerospike delivers exceptional availability and runtime performance with dramatically smaller server footprints through deep exploitation of modern hardware, including multi-core processors with non-uniform memory access (NUMA), non-volatile memory extended (NVMe) Flash drives, persistent memory (PMem), network application device queues (ADQ), and more.

Other features that distinguish Aerospike include its ability to automatically distribute data evenly across shared-nothing clusters, dynamically rebalance workloads, intelligently route application requests to appropriate nodes for fast performance, and accommodate software upgrades and most cluster changes without downtime.

Aerospike can be deployed on-premises, in the cloud, or in hybrid and multi-cloud environments as an integral part of a real-time infrastructure that includes messaging, event stream processing, and data replication technologies to promote movement and integration of critical data throughout the enterprise.

Firms use Aerospike in production as a system of record for the enterprise, as a system of engagement at the edge, and as a global transactional system spanning multiple sites or zones. Strong, immediate data consistency, self-managing and self-healing capabilities, and flexible data replication technologies have helped drive Aerospike's increased usage worldwide. Fig. 2 illustrates several popular usage patterns.

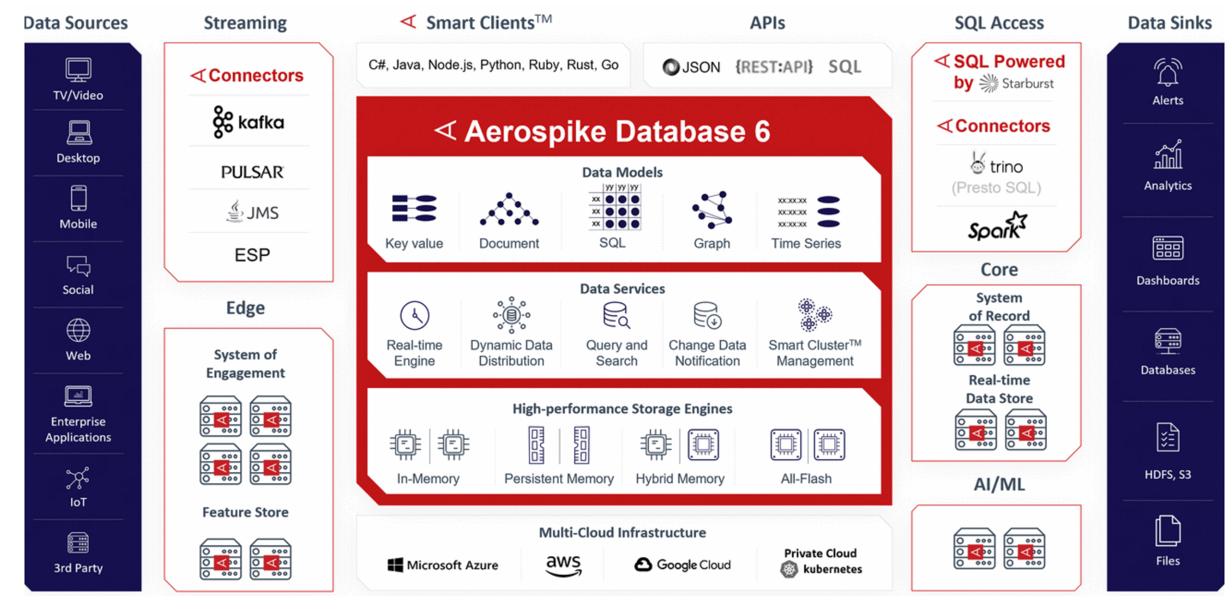

Figure 2: The Aerospike Real-time Data Platform and its usage patterns

Introduction to Aerospike's streaming technologies

To enable firms to capitalize on streaming data pipelines, Aerospike provides both *streaming transport* and *stream processing* services through connectors and Aerospike's Cross Datacenter Replication (XDR) technologies. As shown in Fig. 3, these technologies are often combined with other Aerospike capabilities to help firms integrate streaming data with SQL-based analytics and AI/ML training. The focus of this paper is on the Streaming Connectors.

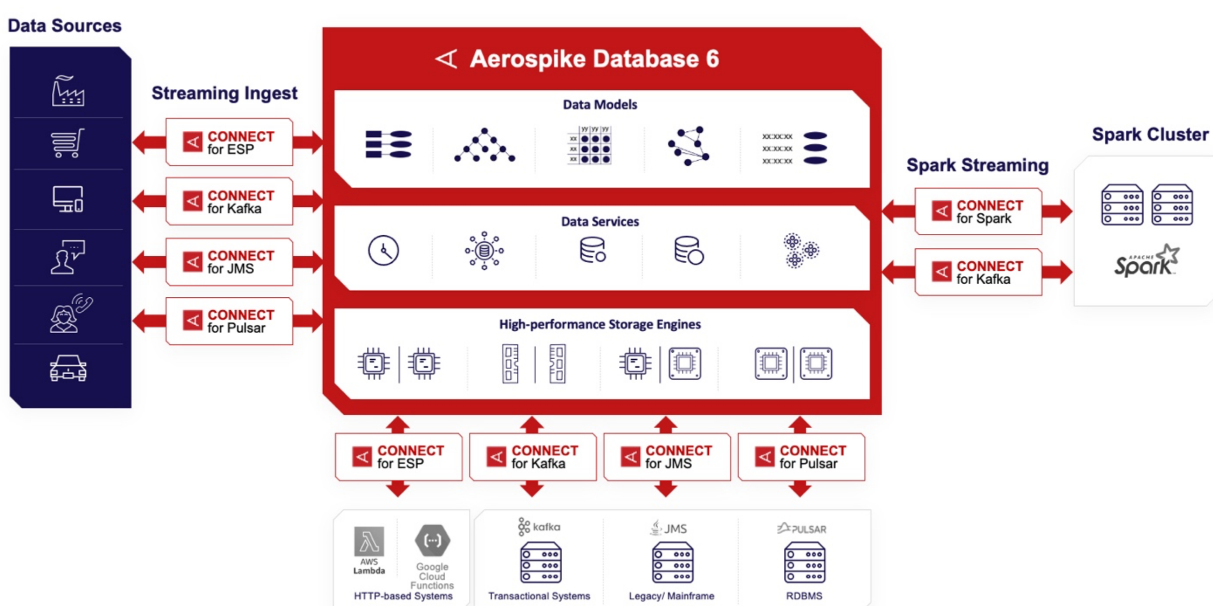

Figure 3: Overview of key Aerospike technologies for streaming data

Stream transport services

A stream transport service functions as a message bus, carrying events from a data source to a destination (or sink). Aerospike's streaming connectors allow Aerospike to serve as the source or the sink in a streaming data pipeline, providing outbound and inbound transport services for streaming data.

As shown in Fig. 4, Aerospike offers connectors to several streaming technologies such as Kafka, Pulsar and JMS in on-premises and cloud deployments. Finally, for event streaming to HTTP-based systems, Aerospike provides change notifications, commonly known as Change Data Capture (CDC), through its Event Stream Processing (ESP) connector.

A powerful feature common to these connectors is that they support outbound processing of data streamed from Aerospike. This allows users to filter and transform the data stream before being posted to the destination.

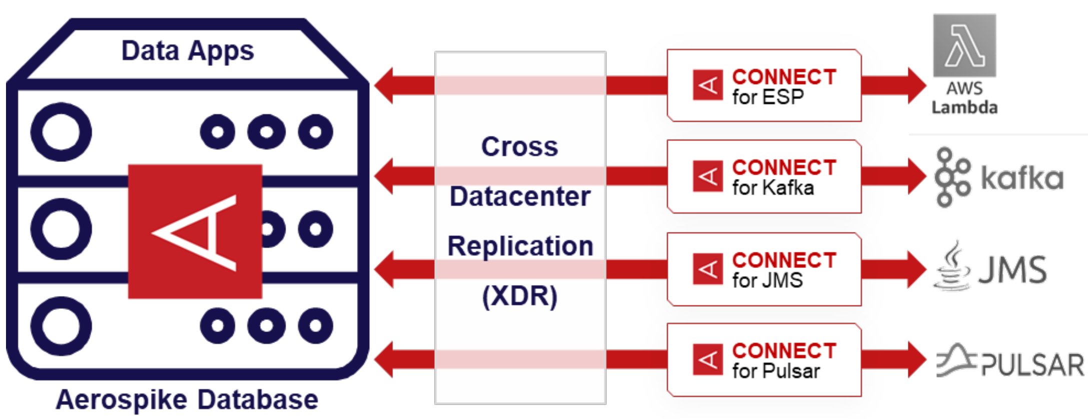

Figure 4: Outbound processing of data streamed from Aerospike

To promote scalability and elasticity, users can dynamically add more instances of the outbound connector to increase the throughput of Aerospike's change notification messages. If any connector instances become underused, they can be removed dynamically to ensure that computing resources aren't wasted.

Aerospike streaming connectors for Kafka, Pulsar and JMS also support inbound processing of streaming data, enabling Aerospike to consume data generated elsewhere and serve as the ingest point in a data pipeline. As illustrated in Fig. 5, Aerospike serves as the sink in a streaming data pipeline in such situations.

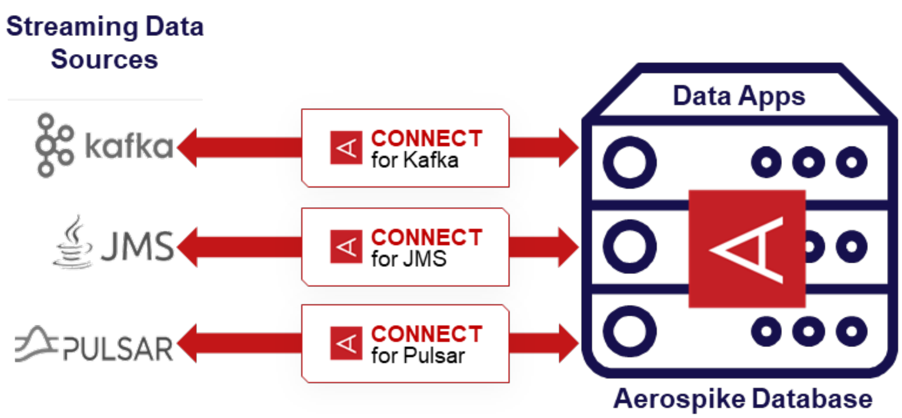

Figure 5: Inbound processing of data streamed to Aerospike

As you'll learn later in this paper, Aerospike's JMS connector promotes the integration of Aerospike with many popular open source and commercial JMS message brokers, including IBM Message Queue (IBM MQ), RabbitMQ, and others. In addition, Aerospike's ESP connector promotes integration with many popular HTTP-based systems, such as Amazon Web Services (AWS) Lambda, Elasticsearch, and others. Finally, partnerships with third-party vendors can further help customers accelerate deployments and operations of various streaming technologies with Aerospike.

Stream processing services

Stream processing involves transforming data streams in some fashion. As shown in Fig. 6, common operations include filtering data, reformatting data, and aggregating data.

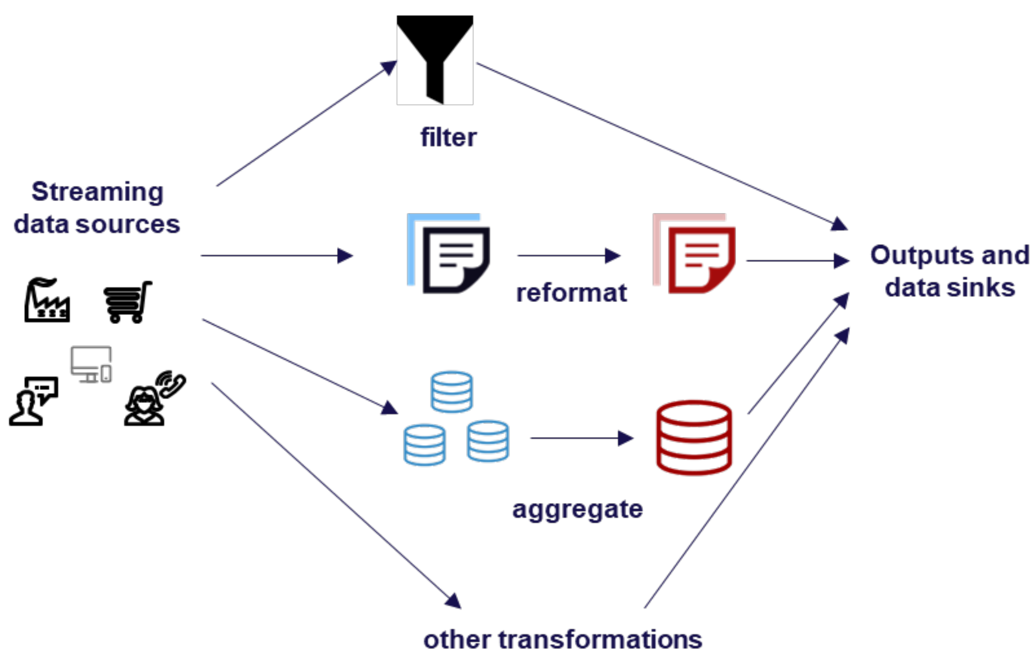

Figure 6: Common stream processing operations

Aerospike supports several approaches to stream processing:

- **Message transformers**, which are extensions to the streaming connectors discussed in the previous section.
- **Spark Structured Streaming operations**, which are supported as part of the Aerospike Spark connector.
- Filtering expressions, which are supported through Aerospike's native data replication technology, Cross Data Replication (XDR). Essentially, user-written expressions instruct XDR to apply filtering operations at the source Aerospike cluster to restrict data sent to the target destination.

Change notification services

As illustrated in Figure 4 above, the Aerospike outbound streaming connectors use Change Notifications provided by the Cross Datacenter Replication (XDR) technology. Since this is a foundational technology for the outbound connectors, it's prudent to describe the technology in a little more detail.

In a nutshell, XDR enables transparent and asynchronous replication of data between different Aerospike clusters and notifications of changed data to streaming platforms through the outbound streaming connectors. To remain within the scope of this document, we'll focus on the latter use case.

How do XDR change notifications work?

When a client writes data to an Aerospike cluster, XDR logs minimal information about the change, batches changed data, and ships the latest version of a record at a pre-configured interval (0-5000 ms) to the target destination. In this way, multiple writes to a single record in an Aerospike cluster generate only one remote message – an essential feature for minimizing overhead when dealing with "hot" data.

Furthermore, XDR supports considerable granularity regarding what data changes are captured. For example, firms can configure XDR to capture changes involving one or more namespaces (databases), one or more sets (collections of records), as well as all or a subset of bins (fields) within records. In addition, users can write filtering expressions to restrict or manipulate the content of change notifications. For streaming data pipelines, filtering expressions can be particularly useful for compliance as well as traditional data processing concerns, such as reformatting or cleansing data.

XDR's granularity options, its treatment of hot data, and its minimal log record format help reduce network overhead and promote faster, more targeted dissemination of important streaming data to the target destination.

Aerospike usage patterns for streaming data

As described earlier, Aerospike can act as a data sink or data source in streaming data pipelines, effectively supporting both inbound (sink) and outbound (source) processing of streaming data as needed for various applications. Examples of these applications are pictured in Fig. 7 below.

Figure 7: Sample applications that benefit from streaming data and Aerospike

A single Aerospike cluster can support both inbound and outbound processing as well as multiple types of connectors. First, we'll briefly discuss some inbound and outbound usage scenarios.

Edge-based Aerospike clusters are often deployed to persist and process inbound streaming data. In such scenarios, ingesting large quantities of data at rapid speeds is essential. AI/ML applications, fraud detection, real-time bidding, and personalization services are examples where inbound processing of streaming data is critical.

Aerospike's ultra-fast performance at scale, high availability, and seamless integration with popular streaming platforms through connectors make it well suited to manage streaming data and operational data generated at the edge. This includes time-series data generated by sensors, geolocation data generated by mobile devices, financial trading data, clickstream data, and more. Data collected at the edge can be filtered, aggregated, and shared with downstream applications through Aerospike's change notification services and outbound streaming connectors. Such outbound processing can feed a centralized system of record (SOR) with data in near real-time to improve business processes, tailor customer services, and promote cost savings. It also helps data scientists retrain and refresh AI/ML models with current data.

For more details, see [Aerospike documentation](#).

Kafka and Pulsar connectors

Aerospike offers connectors for Kafka and Pulsar that each support inbound and outbound processing of streaming data. Aerospike Connect for Kafka and Aerospike Connect for Pulsar share a common design, so we'll discuss both connectors here. Kafka is a popular open source distributed event streaming platform, and Pulsar is a popular open source cloud-native event streaming and messaging platform.

Aerospike connectors for Kafka and Pulsar enable each of those platforms to serve as the source in streaming data pipelines. Fig. 8 illustrates an inbound processing scenario in which Aerospike connectors read messages from Kafka and Pulsar and initiate appropriate insert, update, and delete operations in the target Aerospike cluster.

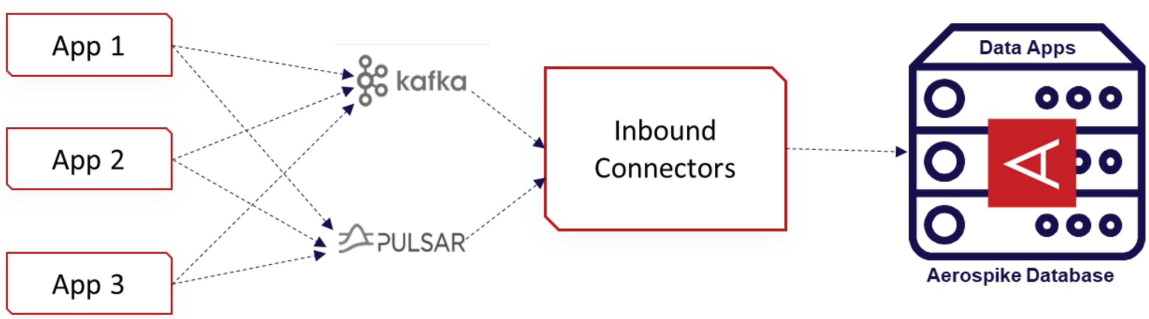

Figure 8: Aerospike as a sink for Pulsar and Kafka

Users can transform incoming Kafka and Pulsar messages through custom Java code before writing the data to Aerospike. This message transformation support offered by Aerospike Connect for Kafka and Aerospike Connect for Pulsar is useful for filtering out records that contain personal information for compliance, masking fields that contain sensitive data, performing complex operations on maps or lists, and so on.

Using the outbound connectors, Aerospike can also stream data from Aerospike to Kafka and Pulsar using XDR change notification services and the respective Aerospike connector, as shown in Fig. 9. This outbound processing scenario calls for the Kafka or Pulsar connector to subscribe to Aerospike database changes, including insertion of new records, updates to existing records, and deletion of records. When such operations occur, the connector transforms them from Aerospike's XDR custom wire format into one of several common data formats such as Avro, Kafka Avro, Flat JSON, JSON, MessagePack and posts the messages to Kafka or Pulsar topics.

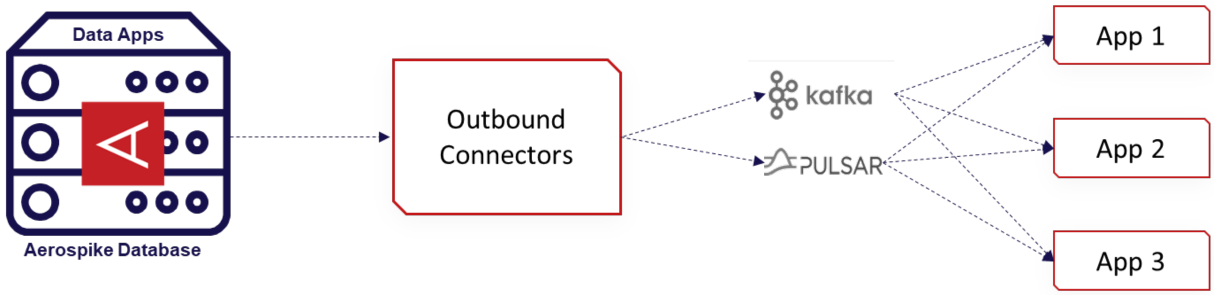

Figure 9: Aerospike as a source for Pulsar

The Kafka and Pulsar connectors can each route messages to one or multiple topics and supports both static and dynamic message routing, depending on the nature of the message. For example, the value of a bin (field) within an Aerospike record can determine how a message is routed.

Event Stream Processing (ESP) connector

Aerospike Connect for Event Stream Processing (ESP) supports outbound processing of streaming data from Aerospike. Specifically, the ESP connector converts XDR change notifications into HTTP requests for downstream consumption. This enables serverless event processing when deployed with other popular offerings, such as AWS Lambda.

Fig. 10 illustrates a sample usage scenario in which the ESP connector is notified when insert, update, and delete operations occur at an Aerospike cluster. The connector converts these XDR-based notifications to HTTP1 or HTTP2 requests and forwards them to a Layer 7 load balancer, which routes them to target destinations (applications).

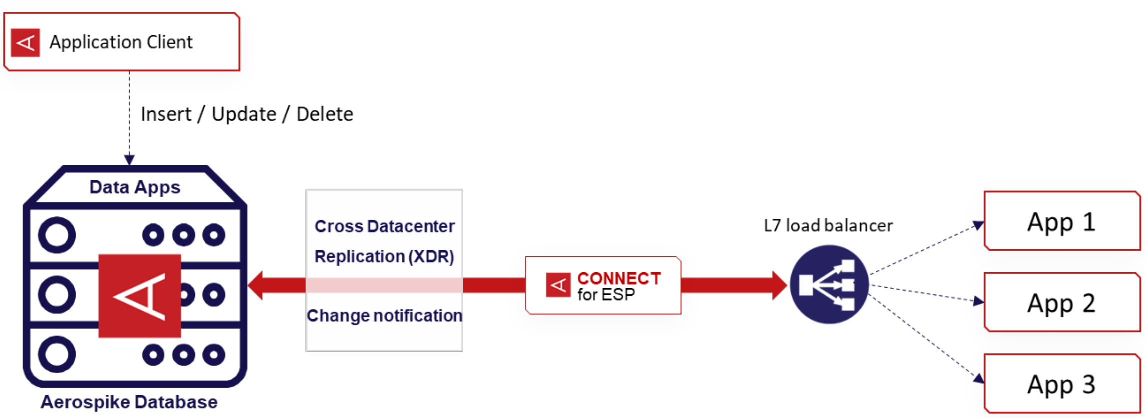

Figure 10: Aerospike as a source for event stream processing (HTTP-based sinks)

Using a load balancer is optional; however, it provides a convenient means to increase the throughput of data streamed from Aerospike, distributing the workload more effectively than possible with only TCP-based load balancing. *(XDR change notifications are TCP-based.)* Fig. 11

illustrates a configuration in which a load balancer (shown at center) helps distribute the workload streaming from the ESP connector (shown at left) to multiple downstream Kafka connectors.

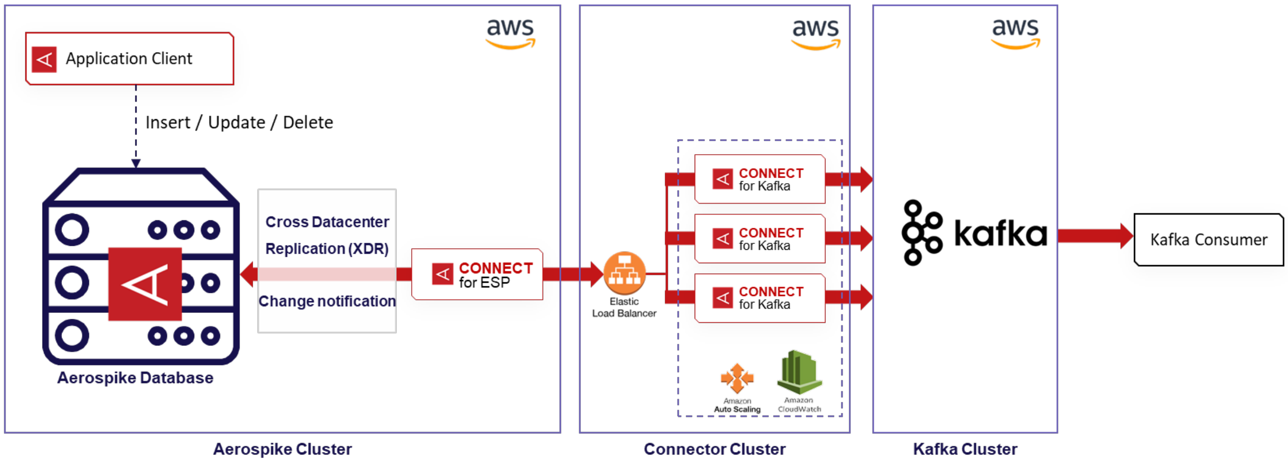

Figure 11: Sample load balancing scenario with ESP and Kafka connectors

The ESP connector serializes Aerospike change notifications into any of a number of supported formats, as discussed earlier.

If desired, users can create XDR filtering expressions to manipulate the data before delivery to target destinations. Any HTTP-based system can serve as a destination (sink) in the pipeline, including ElasticSearch, Splunk, and others. Furthermore, a single ESP connector cluster can support multiple different target destinations. A few popular use cases include:

- Load balance Aerospike streaming connectors for Kafka, JMS, or Pulsar to build scalable streaming pipelines.
- Export Aerospike messages to Elasticsearch using its Document REST API, allowing applications to exploit Elasticsearch's fast search capability while maintaining a 1:1 relationship with the objects stored in the Aerospike Database.
- Archive Aerospike data to S3 (using AWS Lambda).
- Stream Aerospike data to AWS SageMaker (using AWS Lambda).
- Encrypt Aerospike data for compliance (using AWS KMS with Lambda) before writing it into a data lake.

For more details on the ESP connector, see the [product documentation.]()

JMS connector

Aerospike Connect for JMS supports both inbound and outbound processing of streaming data so that a JMS message broker can serve as the source or sink in streaming data pipelines. The ubiquity of JMS and Aerospike's support of this messaging standard promotes integration between Aerospike and popular open source and commercial offerings, such as RabbitMQ, ActiveMQ Artemis, IBM Message Queue (IBM MQ), and Solace.

Fig. 12 illustrates an inbound processing scenario in which the connector reads messages from a JMS message broker and initiates appropriate insert, update, and delete operations in the target Aerospike cluster.

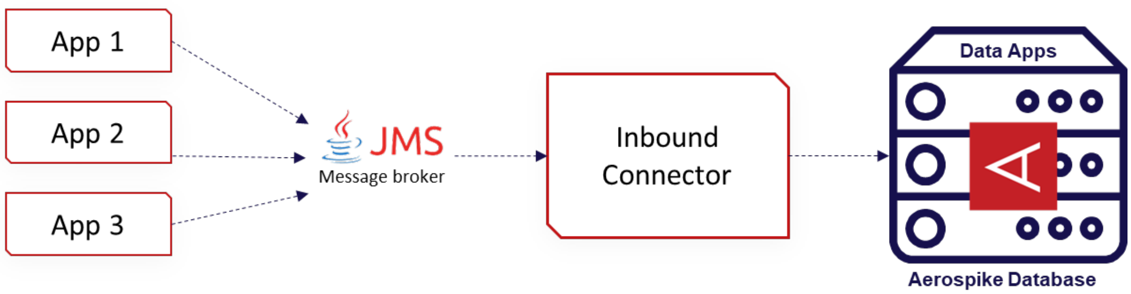

Figure 12: Aerospike as a sink for a JMS message broker

The inbound JMS connector can stream data from one or more JMS queues/topics originating from a single broker, persisting the data to an Aerospike cluster. If desired, users can transform incoming JMS messages with custom Java code before writing the data to Aerospike. The connector's message transformation support is useful for filtering out records that contain personal information for compliance, masking fields that contain sensitive data, performing complex operations on maps or lists, and so on.

In addition, Aerospike can stream data to a JMS message broker using XDR change notification services and the JMS connector, as shown in Fig. 14. This outbound processing scenario calls for the JMS connector to subscribe to Aerospike database changes, including insertion of new records, updates to existing records, and deletion of records. When such operations occur, the connector transforms them from Aerospike's XDR custom wire format into one of several common data formats discussed earlier (JSON, Avro, etc.), and sends the messages to the JMS message broker.

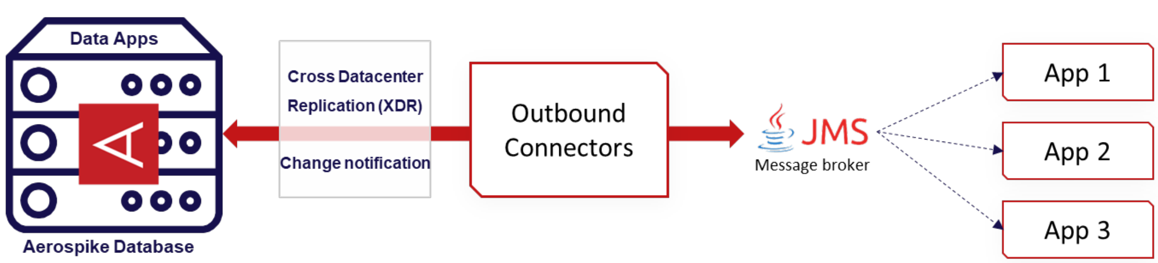

Figure 14: Aerospike as a source for a JMS message broker

Fig. 15 illustrates one JMS use case in which data is offloaded from a mainframe system into an Aerospike cluster through the use of IBM MQ and Aerospike Connect for JMS (configured for inbound processing). Offloading work in this manner can help firms deploy new applications quickly and inexpensively and relieve pressure on overburdened mainframe systems.

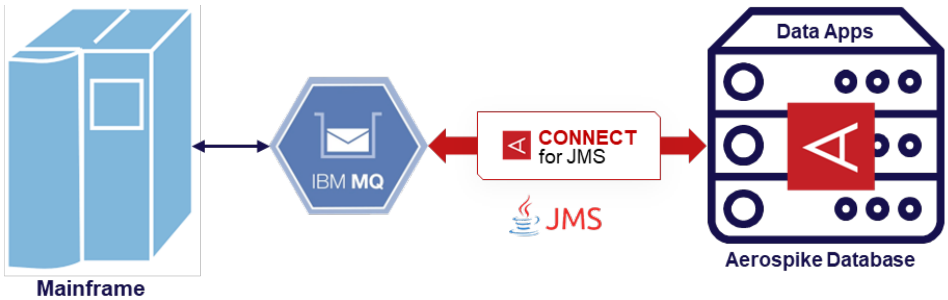

Figure 15: Using JMS connector to offload mainframe data into Aerospike

What customers are saying

Firms in many industries rely on Aerospike to power their mission-critical applications. Deployed at the edge or the core, Aerospike frequently enables firms to achieve aggressive business goals after other solutions have proved ineffective or too costly to maintain. Production customers often use Aerospike to persist, process, and protect high volumes of streaming data relevant to multiple in-house applications; messaging and replication technologies promote integration of such data within the enterprise, powering AI/ML efforts and near-real time data analytics. Let's explore what some of Aerospike's customers are saying about their experiences.

DBS Bank developed a global banking API using Aerospike, Kafka, and remote procedure calls (RPC) to power a wide variety of banking applications. Speed, scalability, and support for complex data structures were some of the factors that led the firm to select Aerospike as its real-time data platform after evaluating several alternatives, including Redis, MongoDB, and FoundationDB.

"The best solution that we found was Aerospike because it supported everything that we wanted. . . . We decided to adopt Aerospike as the database of choice for all services [for our global banking API]."
Matteo Pelai, SVP at DBS Bank

Adform, a leading advertising technology (AdTech) firm, uses Aerospike at the edge to collect and store ad events and, at the core, to manage its advertising campaign data, which includes information critical to billing. Kafka messaging provides communications between the edge and core systems.

"So why did we use Aerospike? AdTech makes money every time we win a bid request. If we can't bid, we can't make money. If we're down for an hour, that's not quite a million Euros that we lose. So we need something that has low latency, huge capacity, high throughput, and high availability. We used to use something else at Adform [Cassandra] -- a cluster with 150 nodes. There were always some nodes that

were broken. We went from 150 to 8 nodes with Aerospike and then eventually 12 nodes. It's easy to use, it's value for money, and it works as advertised."
Peter Milne, Head of Technology Architecture, Adform

Cybereason provides clients with a multi-layer protection stack to defend against malware. They rely on Kafka and Aerospike to power their business.

"We are reading the data from Kafka using microservices and then we write it to Aerospike. Kafka helps us to handle more data, a massive amount of data, and then write it in the optimal way into Aerospike. We are running Aerospike on both GCP and Amazon cloud. . . . Aerospike is so easy to work with. Aerospike does everything automatically. . . . We are using Aerospike as the single point of truth and Elasticsearch is the search engine, and . . . (with Aerospike) we managed to boost the Elasticsearch performance by five times. . . ."
Dotan Gutmacher, Big Data, DevOps & infrastructure team leader, Cybereason

Dream11, India's largest fantasy sports platform, uses Aerospike on AWS to cache data for real-time leaderboards and contest management, supporting 2.7 million concurrent users. Dream11 moved from Elasticache Redis to Aerospike for improved availability, cost, latency and elasticity. Aerospike is deployed with Spark and Kafka to manage geographical traffic spikes across multiple clusters.

"Aerospike has been the most important aspect of the new architecture of (Dream11's) Team Service 2.0. . . . Aerospike is an awesome database . . . (and) we also have their managed service to take care of it for us."
Hammond Pereira, product engineering, Dream11

When **Hewlett Packard Enterprise (HPE)** needed to plan for emerging "exascale" (extreme scale) requirements -- e.g., ingesting more than 85 PB of data per day, storing 2.6 EB (exabytes) of raw data monthly for 5 - 10 years -- the chief architect of its AI-drive big data effort led an effort to benchmark several offerings, including Aerospike, Redis, RocksDB, and Cassandra. As a result, HPE selected Aerospike.

"Aerospike is second to none for ingesting and persisting millions of events per second . . . . (It) allows me to do near-instantaneous machine learning on the data as it lands."
Theresa Melvin, chief architect of AI-Driven Big Data Solutions, HPE

Yahoo (formerly Verizon Media) embarked on a global modernization effort of its data management infrastructure and ultimately selected Aerospike to replace its current system-of-record (managing multiple petabytes of data) and its user data profile store tracking data from its 2 billion customers. The system of record supports a multi-tenant application for all key-value stores across the company, while the profile database serves as the key-value store for users across all Verizon Media domains.

Applications targeted for the new Aerospike infrastructure require low data access latencies, high scalability (at petabyte scale), ease of operations, strong data consistency, global active-active deployments spanning multiple data centers, agility, and a low server footprint. Verizon Media expects to deploy Aerospike in several worldwide data centers using cross-data center replication (XDR) and Apache Pulsar as needed. By replacing multiple open-source platforms and some homegrown technology with Aerospike, Verizon Media expects to simplify its infrastructure, support emerging business requirements, prepare for future growth, and keep within its target budget.

"We tested out many, many key-value stores. Aerospike was the best fit in terms of performance and closest to our feature set. We use Pulsar as the standard messaging bus in our company, and we wanted support for Pulsar for notifications. . . . "
Joe Francis, Director of Storage and Messaging, Verizon Media

VDx.tv, a global AdTech firm focused on video platforms, uses Aerospike on GCP and Kafka to power its real-time bidding services. The firm relies on Aerospike for user and household profile information, advertising campaign data, and critical reporting functions.

"We are using Aerospike as a primary store and caching layer for all critical information for whenever we need high throughput and low latency. . . . Cluster management is easy and doesn't involve operator overhead. The cluster automatically rebalances with minimal impact . . . . Aerospike scales with ease."
Manoj Pal, principal engineer, VDx.tv

Summary

Aerospike is helping firms around the globe process, persist, and integrate streaming data at scale into their IT infrastructures to improve business processes, enhance customer service, drive new revenue opportunities, and lower costs. Through seamless integration with popular streaming and messaging technologies, Aerospike can rapidly ingest massive volumes of streaming data from a variety of sources, enable applications to access and combine that data with other operational and transactional data in near real-time, and share changed data with downstream applications and legacy systems. Furthermore, Aerospike typically delivers these capabilities at a fraction of the cost of alternate solutions, requiring smaller server footprints and less operational overhead even as data volumes scale from terabytes to petabytes.

Earlier in this paper, we explored Aerospike connectors for Apache Pulsar, Apache Kafka, Java Message Service (JMS), and Event Stream Processing (ESP) technologies. Together with other Aerospike technologies, these connectors deliver ultra-fast performance at scale for streaming data and provide critical connectivity between the big data ecosystem and existing IT infrastructures. In addition, Aerospike's built-in change notification capabilities enable firms to stream changes reliably to targets around the globe, and its exceptional availability supports the always-on nature of streaming data and mission-critical business demands.

To learn more about Aerospike and its support for streaming data pipelines at scale, contact sales@aerospike.com or visit Aerospike's website.

About Aerospike

Aerospike unleashes the power of real-time data to meet the demands of The Right Now Economy. Global innovators and builders choose the Aerospike real-time, multi-model, NoSQL data platform for its predictable sub-millisecond performance at unlimited scale with dramatically reduced infrastructure costs. With support for strong consistency and globally distributed, multi-cloud environments, Aerospike is an essential part of the modern data stack for Adobe, Airtel, Criteo, DBS Bank, Experian, PayPal, Snap, Sony Interactive Entertainment, The Trade Desk, and Wayfair. A global company, Aerospike is headquartered in Mountain View, California, with offices in London, Bangalore, and Tel Aviv.

For more information, please visit https://www.aerospike.com.