WHITE PAPER

A high-performance feature store for real-time AI

Aerospike

Contents

Executive summary

In today's fast-paced digital economy, firms increasingly rely on artificial intelligence and machine learning (AI/ML) systems to improve everything from customer experience, sales and marketing effectiveness, and numerous business processes. As the sheer number of ML projects increases, many organizations struggle to manage the data used to create and train the data models required for prediction, recommendation, and other aspects of a successful ML program.

Identifying and managing the features used in ML models is a big challenge. The growing workloads associated with feature computations are becoming expensive; performance and accuracy are critical. This is why many firms are creating a feature store as a central real-time repository for storing, sharing, and serving features, particularly for real-time inference.

A growing number of Aerospike customers, including Sony Interactive Entertainment and Quantcast (a global AdTech), use Aerospike as the real-time data layer powering their online feature stores.

This paper explores how Aerospike helps firms build efficient, low-cost online feature stores and feature serving layers that integrate readily with popular ML tools and legacy infrastructures. Indeed, firms across finance, gaming, advertising technology (AdTech), and other industries are relying on Aerospike to power mission-critical ML applications involving fraud detection, personalization services, and more.

Background

ML applications rely heavily on "features," particularly for real-time inference and decisioning, essentially, relevant attributes about an event or phenomenon, to predict outcomes of interest, such as the likelihood that an incoming financial transaction is fraudulent or the appeal a digital ad might hold for someone requesting a web page. Features are vital input sources for training ML models and for real-time ML applications; as such, they must be carefully identified and engineered.

For online feature ingestion and retrieval, such applications require exceptionally low read/write latencies, high levels of availability, and considerable schema flexibility. For training (and retraining) ML models, high scalability, and efficient management of very large data volumes are essential. And, in both cases, maintaining a low total cost of ownership (TCO) is critical. Aerospike's unique architecture has proven itself an ideal real-time data platform for feature serving.

Feature stores can be used in various ways in ML pipelines. Figure 1 illustrates a sample ML architecture spanning ML data engineering, model development, and production stages. Indeed, feature stores support feature engineering, model training (and retraining), production services (i.e., predictions), and monitoring of model quality and drift by persisting features and serving up models as needed. From a data platform perspective, this requires efficient processing of mixed read/write workloads at multiple stages in the ML process.

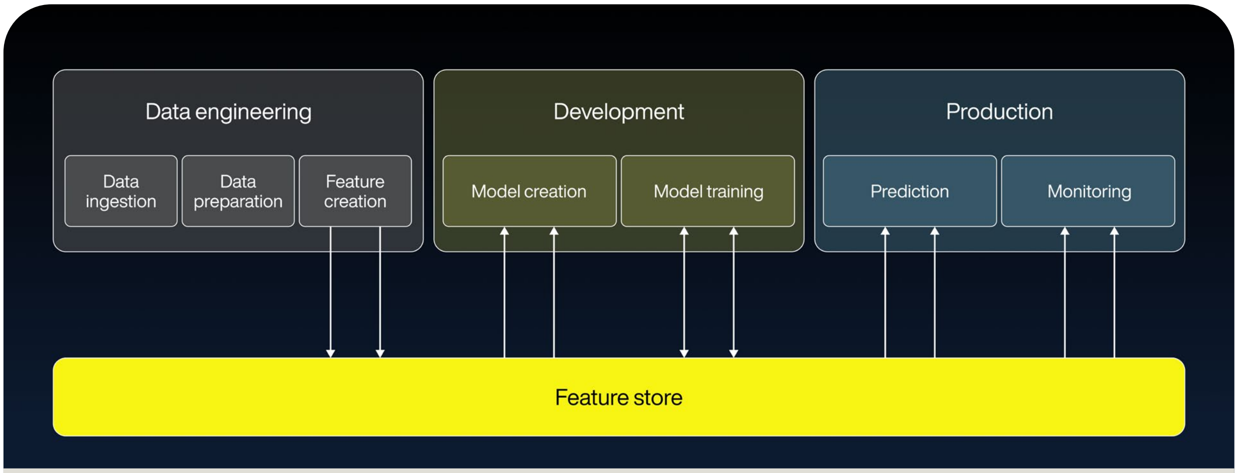

Figure 1: The role of a feature store in the ML process

Many firms create separate “offline” feature stores for training and “online” feature stores for real-time inference to support differing usage characteristics. Offline feature stores support model exploration and development; as such, they must scale readily from terabytes to petabytes and support large volumes of data, including historical data. They commonly require batch-style access, integration with Apache Spark for feature engineering, and a standard SQL interface for easy access. By contrast, online feature stores support real-time, production use of the latest feature data. As such, online feature stores demand extremely low data access latencies but manage lower data volumes than offline feature stores.

Table 1 compares the typical data platform requirements for each type of feature store. As you’ll note, requirements overlap in some cases; for example, budgetary concerns prompt firms to demand cost-efficient options (i.e., low TCO) in both cases.

Characteristic	Offline store	Online store
Typical use case	Training/testing for model development, Batch predictions	Real-time predictions
Data volumes / scalability	Very high	Moderate to high
Data access style	Batch, query	Real-time
Data access latencies	Sub-second to seconds	Sub-millisecond to sub-second
Workload operations	Mixed read/write	Mixed read/write
Availability	Reasonably high (3 nines)	Exceptionally high (5 nines)

Data model	Flexible	Flexible
Feature sharing, security	Strong	Strong
Versioning / time travel	Yes	No
Integration	SQL, Spark, AI/ML notebooks, AWS S3, Google GCS	Kafka, AWS Kinesis, Spark, Pulsar
Total cost of ownership	Low	Low

Table 1: Requirements for data platforms supporting offline and online feature stores

Aerospike advantages as a feature store

Firms from a variety of industries have turned to Aerospike to serve and access online features in real time due to Aerospike's ultra-fast runtime performance, integration with Spark and popular streaming platforms, 99.999% (five nines) uptime, and comparatively small server footprint. Aerospike's unique approach to memory and storage management is prompting firms to deploy additional Aerospike clusters to support selective offline feature access alongside real-time serving. This is due to Aerospike's ability to scale from terabytes to petabytes without incurring high operational costs or sprawling server footprints.

Aerospike is also effective as a feature store due to its integration with SQL and Spark-based AI/ML development tooling and its overall ease of deployment on premises, in the cloud, or in hybrid configurations. Aerospike's flexible configuration options enable firms to employ its technology as the real-time foundation for online feature stores, while integrating with offline systems, simplifying ML infrastructures, and often avoiding excess hardware or cloud computing costs.

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. For more details on Aerospike's design and key features, see this solution brief.

Performance, hyper-efficiency, and cost

Aerospike is a distributed database that delivers predictable low-latency read/write access to billions of records in databases holding up to petabytes of real-time operational data. Firms around the world use Aerospike to support systems of engagement and systems of record 24x7, often saving millions of dollars per application compared with other approaches.

Aerospike delivers exceptional availability and runtime performance with dramatically smaller server footprints through efficient use of modern compute, memory, and storage technologies to deliver predictable performance at high utilization.

Support for mixed workloads

Aerospike's comparatively low TCO and ultra-fast, predictable performance for mixed workloads at scale are arguably the two most common reasons why firms select Aerospike to power online feature serving under mixed read/write workloads. Indeed, enterprises often turn to Aerospike after alternative feature store solutions failed to keep up with growing business demands. For example, Quantcast described the latencies of its legacy Redis-based feature store as "painful" and was pleased when these latencies dropped "significantly" with Aerospike, from 25 ms to 5 ms in some cases. Sony Interactive Entertainment was able to deliver millions of transactions per second while meeting strict real-time inference latency requirements with Aerospike.

Comparative benchmarks across various workloads demonstrate Aerospike's ability to outperform many popular solutions by a significant margin, often with substantial cost savings enabled by its small server footprint. For example, one petabyte-scale benchmark conducted by Aerospike, Intel, and Amazon Web Services (AWS) showed Aerospike supporting 4 - 5 million transactions per second with sub-millisecond latencies on a mere 20-node cluster, saving an estimated $11 million in operational costs over a three year period compared with a popular open source offering (Apache Cassandra).

Ease of deployment and ownership

Beyond performance, low cost, and high availability, firms find Aerospike easy to deploy and manage. The system is self-healing and self-managing under many failure scenarios, and nodes can be readily added or removed from a cluster with no downtime and minimal operator effort. Some firms also employ Aerospike's support for collection data types (CDTs), including maps and lists, when modeling feature data and metadata, which can be useful for versioning/time travel.

Optimized for real-time read/write workloads using its native APIs, Aerospike also offers efficient SQL-based access through its Presto (Trino) connector. Other connectors support Spark, Kafka, JMS, and additional platforms. Such integration promotes sharing of ML features across different tools and applications, with Aerospike's core engine protecting sensitive data through strong authentication and access control capabilities.

Aerospike and the ML process

Figure 2 illustrates a sample ML process spanning feature engineering, model creation, and real-time inference using Aerospike as a feature store.

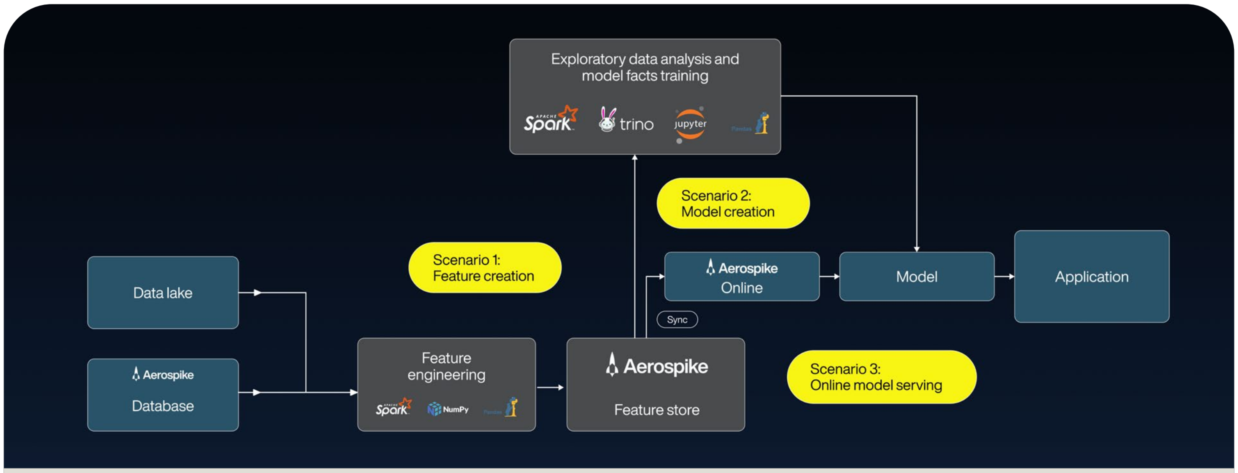
Exploratory data analysis and model facts training
Spark
trino
jupyter
Scenario 2: Model creation
Scenario 1: Feature creation
Data lake
Aerospike Online
Model
Application
Sync
Aerospike Database
Feature engineering
Spark
NumPy
Aerospike Feature store
Scenario 3: Online model serving

Figure 2: Sample scenario with Aerospike as a feature store

Let's walk through three common stages in an ML lifecycle.

Scenario 1 - Feature creation

Data from an Aerospike database and other data lake sources (at left in Figure 2) is loaded into Spark or other distributed compute frameworks for analysis and feature engineering. Aerospike Connect for Spark provides an easy means to extract and process the raw data from the source Aerospike database for pre-processing by data engineers, who clean, transform, and reduce the data to create features using Jupyter notebooks or similar tools. Once features have been created, they're loaded into a separate Aerospike cluster serving as the online feature store.

Scenario 2 - Exploratory data analysis and model creation

Data scientists create ML models using the features sourced from the feature store. Data scientists explore and analyze these features to identify those best suited for the specific models (ML algorithms) appropriate for their use cases. During this process, Spark, notebooks, pandas, and other common ML development tools can be leveraged. The model is finalized after training and testing. Features used in the model and corresponding metadata are stored in the Aerospike feature store in the format needed for the ML framework.

Scenario 3 - Model serving

Models are deployed in production and can be called by applications to make predictions based on incoming data. Additionally, a database can be used to store other application data and host the online feature store. An Aerospike-powered feature store gives companies the option to support both model serving options:

- **Online model serving and prediction**

  This is real-time where the prediction occurs immediately. The application retrieves features from the feature store and passes them to the model for prediction.

- **Offline model serving**

  Primarily used for batch inference, there's no need to serve the features from the store for the model in real-time. Thus, latency requirements are more relaxed. However, scalability and availability requirements are often stringent.

As mentioned earlier, Aerospike is particularly well-suited for the online inference scenario which requires a low-latency/highly performant data platform. An Aerospike online feature store meets this requirement by supporting sub-millisecond reads for inference. As seen from the process, Aerospike can handle all steps of the ML process in conjunction with various ML tools.

Aerospike reference architecture

Figure 3 below depicts a high-level reference architecture for an Aerospike powered deployment of a resilient, high performance feature store.

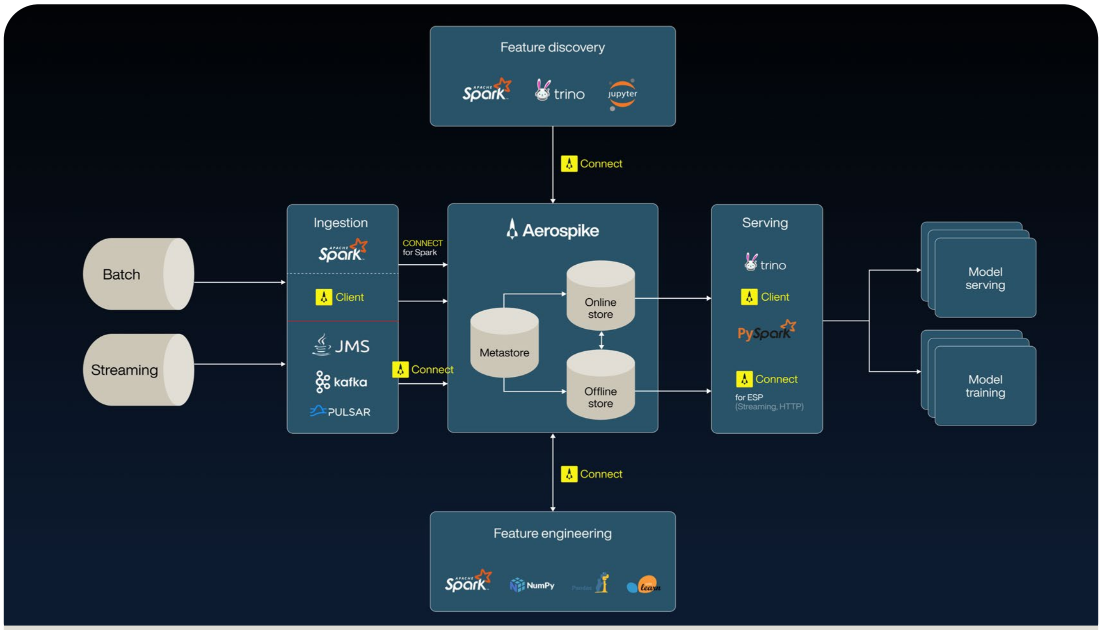

Figure 3. High-level reference architecture

Integration with AI/ML tools

Aerospike integrates with most mainstream AI/ML tools, allowing customers to utilize their tools of choice. It is designed to ingest large amounts of data in real-time for parallel processing while connecting to compute platforms, notebooks, and ML packages.

Fast data ingestion

Aerospike provides several options for high-speed ingestion of raw data originating from various sources. These options include Cross Datacenter Replication (XDR) for raw data sourced from another Aerospike cluster and Aerospike connectors for Kafka, JMS, Pulsar, and other platforms for data residing in data lakes. Each enables firms to create low-latency streaming pipelines to Aerospike, leveraging parallelism and other features to improve efficiency and reduce operational costs.

Low latency storage architecture

Most real-time AI/ML infrastructures require low latency and high throughput feature stores for model serving in their ML pipelines. Aerospike is a shared-nothing, multi-threaded data platform that exploits modern hardware and network technologies to drive reliably fast performance, often at sub-millisecond speeds across petabytes of data. Aerospike provides tiered storage options for optimal densities with high performance, as shown in Figure 4.

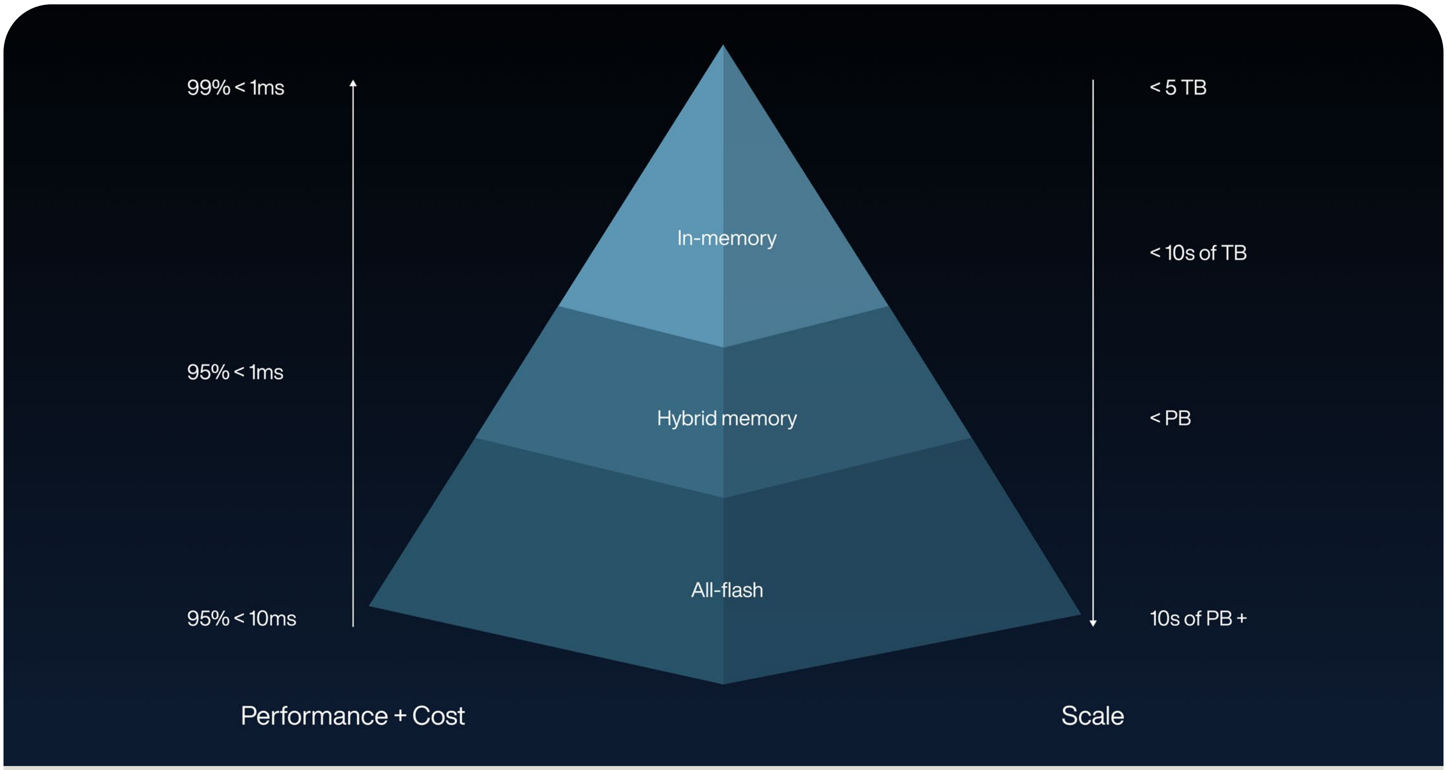

Figure 4: Aerospike's storage options provide high performance at scale and low TCO

- **In-memory**

  This configuration is best suited for the smallest datasets with the highest performance requirements. It provides the lowest latencies but may be ideal only for smaller feature store implementations due to its cost structure.

- **Hybrid memory**

  With hybrid memory configurations, Aerospike stores indexes in DRAM while persisting and accessing data in Flash, virtually as fast as if it were in DRAM. Typically, this is the best option for data ranging from a few terabytes up to all but the largest real-time data sets. This option is ideal for feature stores of different sizes, as it maintains performance at a larger scale.

- **All flash**

  When the number of objects starts to crest into the billions, even storing indexes in DRAM becomes untenable. Thus, Aerospike created an “all flash” option that places both indexes and data on flash, yet can still access 95% of the data in under 10 milliseconds, even at multiple petabytes. Like the hybrid memory option, this is suitable for various feature store implementations, though it has some limitations for online model serving.

Aerospike can power the most resource-intensive online feature stores with sub-millisecond reads for inference. As mentioned earlier, Aerospike can also support offline model serving scenarios, where the latency requirements are relaxed.

Feature engineering and sharing

Using Aerospike Connect for Spark (or other popular tools), data engineers clean, transform, and reduce the data to create features during data pre-processing. Once features have been created, they're loaded into the Aerospike feature store. Features stored in Aerospike can be easily shared (accessed), and any team involved in AI/ML activities can be granted access. Complex features can be modeled by utilizing Aerospike's CDTs, and the feature metadata (including provenance and versions) can be stored in Aerospike using its general data model. The next step in the AI/ML process is model creation and training.

Accelerated model training with Aerospike and Spark

Models must be created, trained, and validated before they can be served for inference. Aerospike can help significantly accelerate model training pipelines. For example, an Aerospike-powered feature store enables data to be loaded rapidly with massive parallelism into Spark DataFrames using Aerospike Connect for Spark, which helps to reduce the model training time significantly. Aerospike Connect for Spark achieves exceptional performance through features such as:

- **Predicate pushdown:** This enables data to be processed where it resides, minimizing data transfers and network overhead.
- **Partition mapping:** Aerospike's ability to map 4,096 Aerospike partitions to 32K Spark partitions (which is configurable) promotes massively parallel reads.
- **Throughput tuning:** Administrators can configure various parameters related to batch processing for reads and writes, worker pool thread size, and more to improve throughput and performance for specific workloads, as described in Aerospike documentation.

The benefits of these features are reduced load and Spark job execution times, resulting in reduced training time, thus increasing the frequency of retraining. This ultimately leads to more accurate predictions.

Online model serving

Online model serving poses stringent performance requirements on the ML infrastructure. Indeed, an online feature store must support ultra-fast read latencies, which Aerospike is known to deliver. Indeed, sub-millisecond reads are common in Aerospike deployments, even at scale. This is critical, as increased use of ML technologies leads to a significant increase in the number of features and models that must be supported. While some data platforms can deliver submillisecond reads at small scales, none has proven to match Aerospike's runtime performance at scale on a relatively small server footprint. Aerospike's unparalleled stability can help maintain an ML application's availability and performance, and any advanced functions that are needed can be implemented through Aerospike's rich ecosystem of clients.

In some architectures, Aerospike can also be deployed closer to the edge, improving the ML infrastructure's performance further. There are advantages to having Aerospike at the edge: it can provide backpressure, provide data for compliance initiatives, and filter out unnecessary data.

Aerospike solution components

For the Aerospike reference architecture, note that in addition to the Aerospike data platform itself, there are a number of platform technologies that comprise an Aerospike solution for feature stores, as shown in Table 2.

Data ingest	Feature engineering and model creation	In production
Aerospike Connect for Kafka	Aerospike Connect for Spark Aerospike Connect for Presto	SQL access: Aerospike Connect for Spark Aerospike Connect for Presto
Aerospike Connect for JMS	Inbound message transformer Outbound message transformer	Aerospike Client APIs for Java, C/C#, Go, Python, etc.
Aerospike Connect for Pulsar	XDR filtering	HTTP: Aerospike Connect for Event Stream Processing (ESP)

Table 2: Aerospike feature store solution components beyond the base data platform

Customer examples

Let's explore two examples of how firms are using Aerospike to power their feature stores.

Sony Interactive Entertainment

Sony, makers of the popular PlayStation gaming platform, uses Aerospike as its runtime feature store to make personalized recommendations for PlayStation users based on their past behavior and other factors. Before selecting Aerospike, Sony evaluated several data platforms to determine which was best suited to meeting their requirements, which included:

- 100+ million users
- 5TB+ of data
- 100+ features per user
- Sub-10ms data retrieval
- Low TCO

The firm recognized that such requirements would place considerable demands on the underlying data platform for their feature store, so they compared several popular NoSQL offerings before settling on Aerospike.

> “We had to make snap decisions at runtime at a scale of millions of requests a second and our latency requirements were under 10 ms. And we wanted our total personalization solution to have a low total cost of ownership. The candidates (for our feature store solution) were Cassandra, Aerospike, and Couchbase. Given the . . . scale and low latency requirements that we had, we felt that Aerospike’s design . . .was really good for the use case that we have. Total cost of ownership is pretty low because a small cluster can handle several terabytes of data.”
>
> **Suresh Bathini**
>
> Vice president of software engineering, PlayStation, Sony Interactive Entertainment

With Aerospike powering its runtime feature store for personalization services, Sony’s overall ML architecture supports both model training and prediction services, as shown in Figure 5.

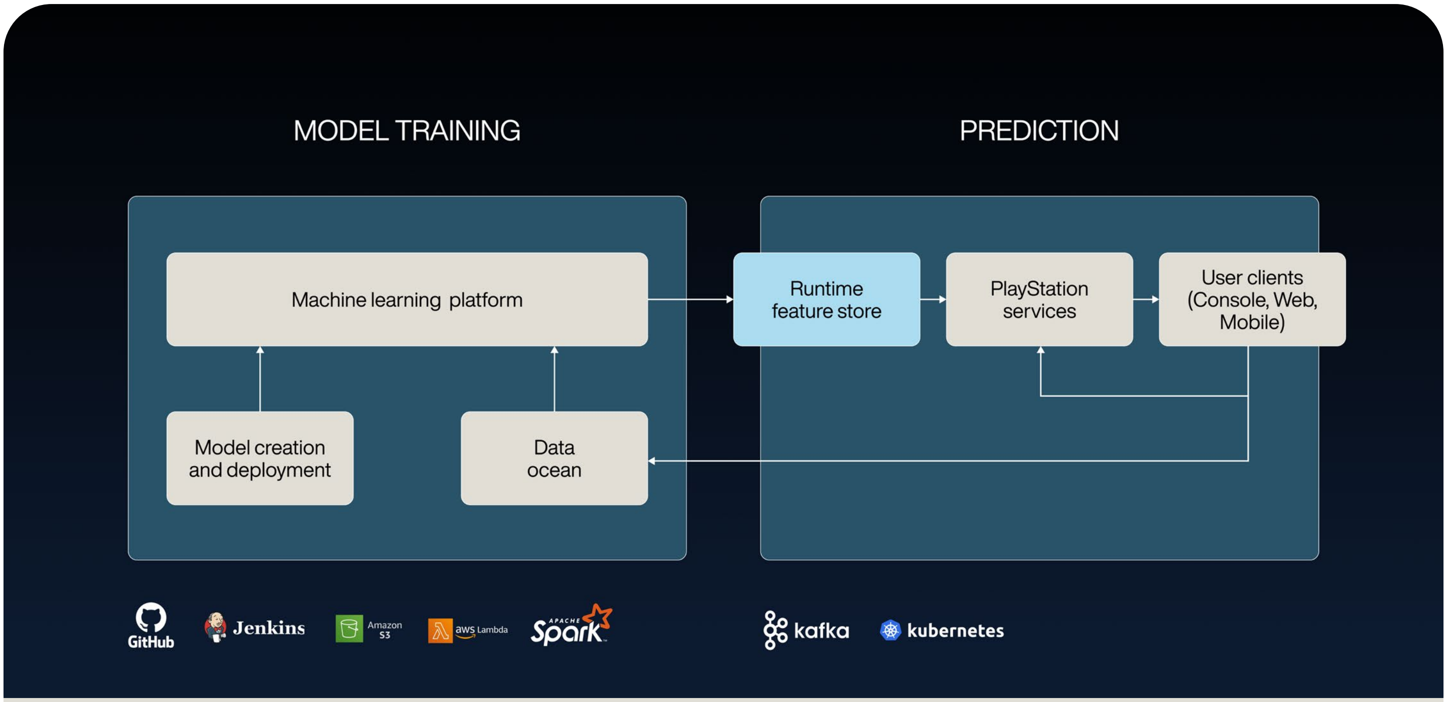

Figure 5: Sony’s architecture for supporting PlayStation personalization services

Quantcast

Quantcast, a global AdTech firm, turned to Aerospike after recognizing that its legacy feature store for personalized advertising was struggling to fulfill the needs of its growing business. Problems with the legacy platform included poor flexibility, scaling and operational issues, unreliable and inconsistent data, maintenance difficulties, and more.

The firm sought a new low latency data platform that would be reliable, promote greater feature experimentation, and serve as a prototype for other bidding data stores. Finally, the new system needed to meet the following targets:

- 10 billion records
- 8TB of data
- Sub 2ms lookup latency
- 1 million lookups/second
- 200K updates/second

> “At the end of our proof of concept (with Aerospike) . . . we were really happy with the way things ended up. Our p99 and p95 latencies were pretty painful in our legacy (Redis) system. They dropped significantly . . . (Aerospike) ended up being super reliable. It was easy to set up. It was well documented. Aerospike support was #1 in my book. Best support (that) I think I ever received. Aerospike met all of our expectations.”
>
> **Kristi Tsukida**
> Sr. Software Engineer, Quantcast

Figure 6 illustrates Quantcast’s ML architecture for personalized advertising; Aerospike serves as the online feature store.

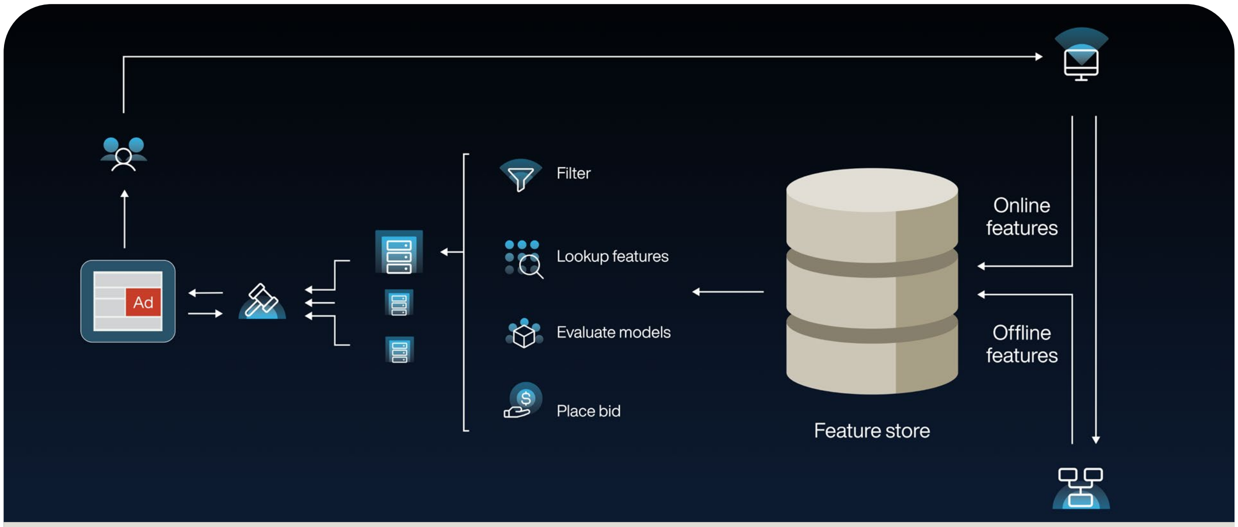

Figure 6: Quantcast’s architecture for supporting personalized advertising

Summary

Firms around the globe are turning to Aerospike to power real-time inference and decisioning workloads, often using Aerospike clusters as the core underpinning of their feature stores. Aerospike's ultra-fast performance at scale, exceptional availability, integration with popular tooling and streaming platforms, and small server footprints are the driving forces behind such decisions.

Earlier in this paper, we explored key aspects of Aerospike's technology designed to deliver superior data management services for feature store deployments, including:

- Flexible memory and storage management architecture suitable for online and offline feature store support.
- Self-healing and self-managing features for high availability.
- Spark, Kafka, JMS, Presto/Trino, and other connectors for easy integration with popular AI/ML tools, streaming platforms, messaging systems, and legacy data infrastructures.
- Massive parallelism and deep exploitation of advanced hardware and network technologies for ultra-fast, predictable runtime performance at scale.

To learn more about Aerospike and its support for low-latency, AI-driven applications at scale, contact sales@aerospike.com or visit the Aerospike website.

About Aerospike

Aerospike is the **real-time database** for **mission-critical use cases and workloads**, including transactional applications, **machine learning, generative, and agentic AI**. Aerospike powers millions of transactions per second with millisecond latency, at a fraction of the cost of other databases.

Global leaders like Adobe, Airtel, PayPal, Sony, and The Trade Desk rely on Aerospike for customer 360, fraud detection, real-time bidding, and other use cases. Aerospike is trusted by over 300 enterprises worldwide and powers apps used by more than 2.5 billion people across the globe.

Try Aerospike for free: aerospike.com/try-now

2440 W. El Camino Real, Suite 700, Mountain View, CA 94040 | (408) 462-2376 | aerospike.com