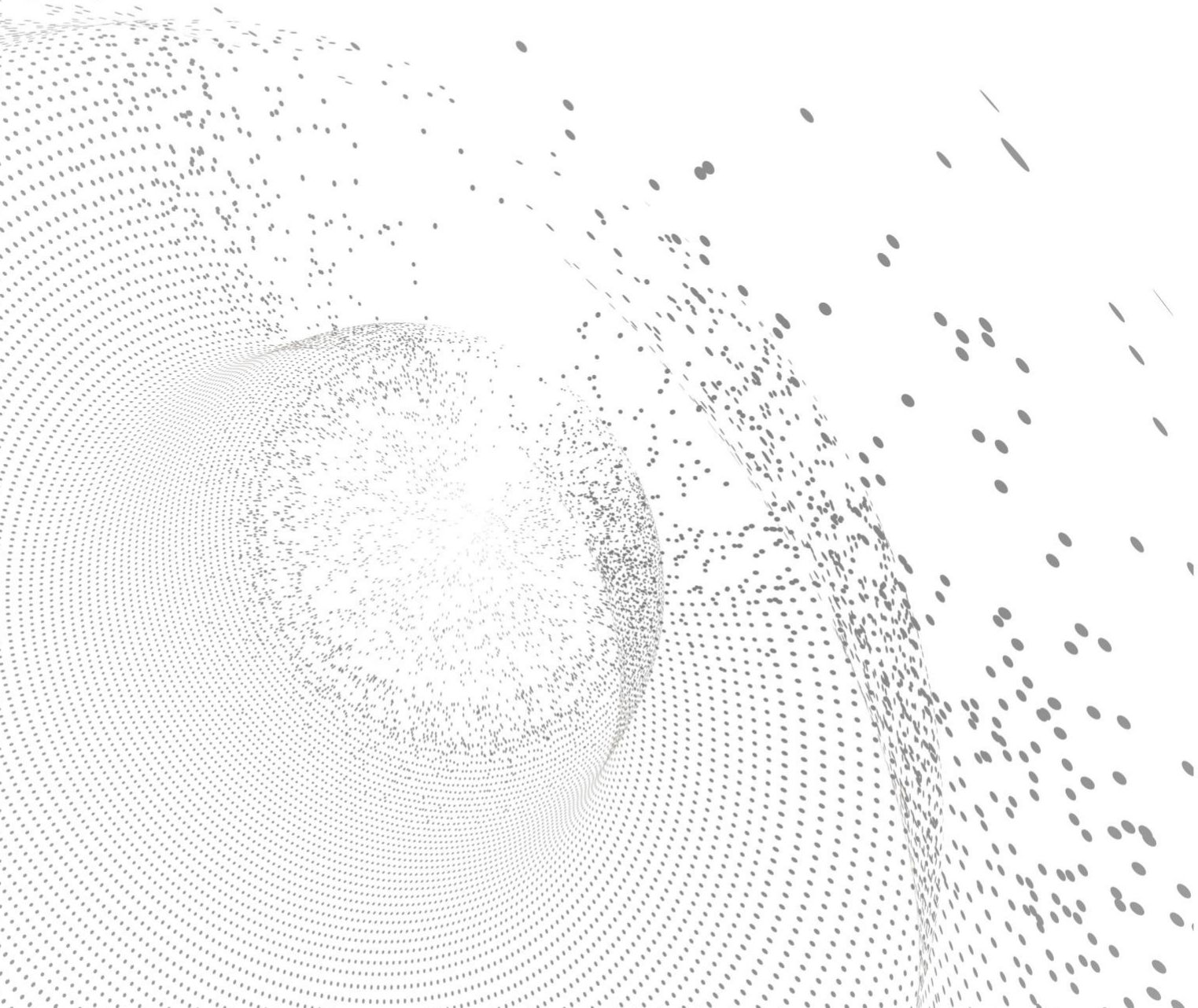
AEROSPIKE

High Performance Applications With Aerospike CacheDB

Get the performance of a cache with the persistence of a complete NoSQL database



Executive Summary

Scalability without compromise: the goal is easy to express but hard to achieve. For years, firms have relied on caching platforms to speed application access to critical data managed by corporate systems. But rapidly growing data volumes, demanding service-level agreements (SLAs), and budgetary pressures have rendered many once-serviceable solutions impractical.

Enter Aerospike, a NoSQL platform for real-time mixed workloads at scale. Whether deployed as a managed cloud service or on premises, Aerospike offers a simple, low-cost approach to caching hundreds of gigabytes to petabytes of data with predictable, ultra-fast data access speeds. Indeed, firms in finance, telecommunications, retail, and other industries depend on Aerospike to deliver sub-millisecond responses on remarkably small server footprints. Aerospike's patented Hybrid Memory Architecture™ leverages volatile and non-volatile memory in unique ways to expand cache size in a cost-efficient manner. As a next-generation caching platform, Aerospike offers critical features that other solutions don't: deep exploitation of the latest memory, storage, and networking technologies; seamless persistence; strong consistency; near 100% uptime; high-performance connectors to popular software platforms; and support for global transactions. Simply put, Aerospike caching enables firms to fulfill demanding SLAs with minimal expense and overhead.

Skeptical? Comparative benchmarks demonstrate Aerospike's remarkable speed and total cost of ownership (TCO) savings. Indeed, firms such as AppsFlyer, Charles Schwab, Signal, The Trade Desk, and others chose Aerospike for its scalability, performance, availability, and cost efficiency. This paper explores Aerospike's ability to serve as a next-generation data cache for real-time workloads at scale. You'll see how caching with Aerospike can speed application response times by keeping frequently accessed data close to the applications that need it.

The Case for Caching

Internet-scale applications must support thousands-to-millions of users, terabytes-to-petabytes of data and ultra-fast response times. That's why caching has become essential. Keeping frequently accessed files, images, session data, web content, configuration data, corporate data, and other information in cache helps applications reduce data access latencies and achieve target SLAs.

Caching platforms are also critical for coping with periods of peak usage (such as Black Friday and Cyber Monday sales in the retail industry, or a contest for gaming, for example). Caching frequently-accessed data helps prevent back-end systems from becoming overwhelmed by sudden usage spikes -- a situation that would severely degrade application responsiveness. Caching platforms often retain user session data for online shopping carts, leaderboard personalization, real-time recommendation engines, and other critical application functions. Furthermore, many firms use caching technology for quick access to user authentication tokens, API responses, configuration settings, and various types of web content.

Figure 1 illustrates two simplistic caching strategies. At right, applications interact with the cache for both reads and writes; the cache retrieves data as needed from the back-end system and replicates changed data to the back-end system in a synchronous or asynchronous manner. This approach minimizes data access logic in the application layer but can introduce write latencies and other concerns. For these reasons, many firms develop their applications to interact directly with the cache for certain operations and the back-end store for others, as depicted in the scenario at left.

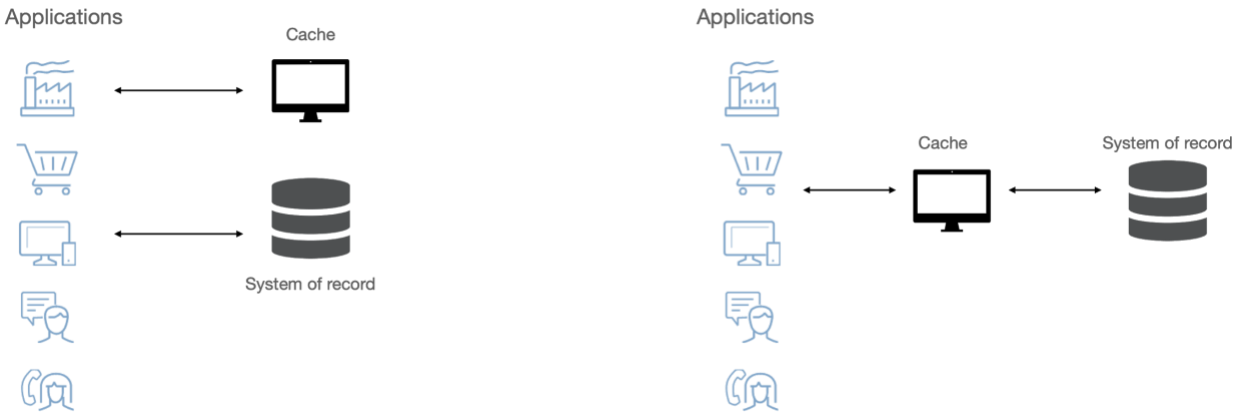


Figure 1: Application-managed caching and inline caching

As you might expect, application design patterns for caching are more nuanced than just discussed. To exploit caching, applications often employ one or more of these patterns:

- Look-aside caching (sometimes called cache-aside). Applications attempt to read data from the cache. If requested data isn't present (i.e., if a "cache miss" occurs), the application layer retrieves the data from the DBMS and caches it for future use.
- Read-through caching. Applications direct read requests to the cache. If the requested data is present, it's returned immediately. If not, the caching layer retrieves the data from the back-end DBMS and returns it to the application.
- Write-through caching. As with read-through caching, applications interact only with the caching layer, which synchronously updates the back-end DBMS to reflect write operations at the cost of higher overall write latencies.
- Write-behind or write-back caching. Similar to write-through caching, this approach relies on the cache to transparently update the back-end DBMS. However, updates are asynchronous. This minimizes write latencies but introduces brief periods when data in the cache and the back-end DBMS may be inconsistent.
- Write-around caching. This approach relies on the application layer to write changed data directly to the back-end DBMS; some designs call for the application to invalidate the cache entry upon successful completion of the write. When combined with a look-aside design, the application layer manually refreshes the cache with DBMS data. When combined with a read-through design, the caching layer periodically refreshes itself with DBMS data.

Caching Challenges

While caching is a compelling strategy for many application scenarios, first-generation solutions often fall short of meeting enterprise demands. Availability, persistence, data consistency, and operational issues commonly emerge, particularly as data volumes and workloads increase. Indeed, today's global digital economy places incredible demands on data management infrastructures, such as:

- Predictable, ultra-fast data access speeds, often at the sub-millisecond level.
- Petabyte-level scalability without massive server footprints nor cloud computing costs.

- Self-managing environment with 24x7 availability.
- Flexible data modeling backed by comprehensive key-value operations and query capabilities.
- Easy enterprise integration through message queuing and other technologies.
- Option to exploit persistence, strong data consistency, and data replication features.

That's why many firms have turned to Aerospike to power their real-time, production applications. Aerospike can efficiently cache modest data volumes (e.g., hundreds of gigabytes) and scale with ease to manage petabytes of data. This eliminates the need to change your infrastructure and retrain your staff as your workloads and data volumes grow. Indeed, many firms have replaced "legacy" caching solutions as well as contemporary NoSQL caching solutions with Aerospike.

For example, one firm moved to Aerospike after experiencing "weird" latency spikes, increased average response times, long and disruptive failovers, and other problems with an alternate system. Aerospike's cost-efficient architecture addressed these issues and cut costs by 85%. Another firm replaced its old infrastructure with Aerospike, Apache Kafka, and Apache Spark so it could "flawlessly" manage a three-fold data increase and improve performance. Yet another firm used Aerospike to cache data managed by its overloaded mainframe DBMS and support new applications that would otherwise have been impractical to implement.

How? Let's take a closer look at Aerospike.

Next-Generation Caching with Aerospike CacheDB

Aerospike CacheDB enables users to enjoy remarkable runtime speeds and low TCO because it's different from other caching solutions. Indeed, it's more than just a fast, reliable cache -- it's a highly scalable distributed NoSQL system optimized for 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. Such optimizations, coupled with Aerospike's multi-threaded architecture and other features, provide firms with distinct advantages for both on premises and cloud deployments. For example, Aerospike automatically distributes data evenly across its shared-nothing clusters, dynamically rebalances workloads, and accommodates software upgrades and most cluster changes without downtime. In the Amazon Web Services (AWS) cloud, Aerospike exploits caching on ephemeral devices, backing up data on Elastic Block Store (EBS) volumes.

Figure 2 illustrates a typical way in which Aerospike can be deployed to speed application access to data managed by transactional systems, RDBMSs, and various NoSQL platforms. In this scenario, applications interact directly with the Aerospike cache to read or write frequently accessed data; these same applications direct requests for less frequently accessed data to existing enterprise systems. Aerospike connectors for Apache Kafka and other technologies enable firms to transfer data between Aerospike and enterprise systems if desired, as the dotted line at right indicates.

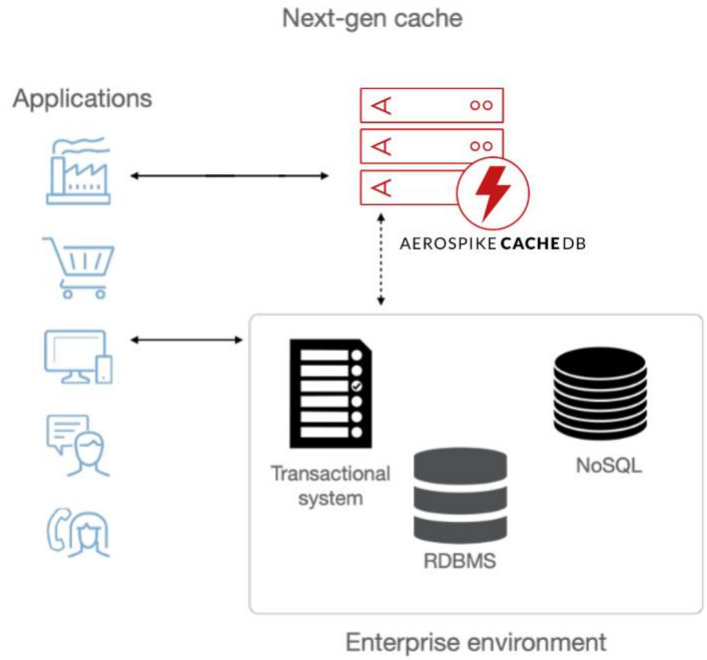


Figure 2: Sample Aerospike CacheDB deployment

Written in C by a development team with deep expertise in networking, storage and databases, Aerospike delivers unprecedented resource utilization that translates into remarkably low server footprints. Indeed, one firm that migrated to Aerospike reduced its server count from 450 to 60 nodes and expects to save $8 million in TCO over three years.

Unlike many other caching solutions, Aerospike doesn't need to rely extensively on costly DRAM to achieve high throughput rates and rapid response times. Although a DRAM-only configuration is fully supported, Aerospike's Hybrid Memory Architecture™ delivers exceptional speed at scale by exploiting Flash and PMem technologies. Such configurations often yield substantially lower server footprints -- and substantially lower operational costs. Furthermore, Aerospike's smart client software layer understands how data is partitioned across the cluster, so it automatically routes data access requests to the appropriate node, avoiding extra network "hops" that can cause slow -- and unpredictable -- runtime performance.

Usage Patterns

To accommodate a wide range of demanding real-time applications, Aerospike supports several caching strategies. For example, firms seeking to speed access to critical data in existing SQL or NoSQL DBMSs often deploy Aerospike as a front-end cache to such platforms. By adopting look-aside and write-around coding patterns in their applications, firms greatly minimize data access latencies without taxing their back-end systems or undertaking costly data and application migration efforts. Cache expiration policies define how data is expunged from the cache. Aerospike supports Least Recently Used (LRU) and maximum age with time to live (TTL). For an LRU scenario, Aeropsike updates the TTL on each read to ensure that recently requested items remain in the cache. Furthermore, Aerospike's built-in persistence and strong data consistency give firms the option of writing directly to the cache without risking data loss. Aerospike's Kafka connector, as well as various messaging queuing technologies, can streamline data integration between the Aerospike cache and legacy systems.

Firms embarking on digital transformation efforts often use Aerospike as a high-speed cache for applications at the edge and a separate Aerospike cluster as a system of record at the core. Aerospike's Cross Data Center Replication (XDR) technology transparently propagates data in an asynchronous, near real-time fashion. As a result, cache misses are often avoided. Writes are automatically handled in an efficient, transparent manner. For example, multiple cache writes for one record may generate only one write to the Aerospike system of record – an important feature for frequently updated data.

Finally, Aerospike supports **distributed caching**, a capability rarely found in first-generation solutions. Because Aerospike's multi-site cluster capability allows a single Aerospike cache to span multiple cloud regions or data centers, Aerospike can service applications in different geographies that need ultra-fast access to globally consistent data. Failovers are immediate and free of operator intervention in most cases.

Client Scenarios

Firms in finance, telecommunications, retail, advertising, and other industries use Aerospike for mission-critical applications involving recommendation engines, fraud detection, payment processing, machine learning, and more. Such applications demand predictable, ultra-fast data access at scale from a platform that's always available.

Quite often, organizations turn to Aerospike to speed access to data managed by legacy SQL or NoSQL systems, to replace existing cache solutions that have become too unwieldy and costly to operate, or to fulfill SLAs that other systems can't. Aerospike's relatively small server footprint, ultra-low and predictable data access latencies, and ease of operations are common motivating factors. Unlike other alternatives, Aerospike scales easily from hundreds of gigabytes to petabytes, enabling you to readily expand your caching layer as your business grows. Let's explore a few customer scenarios.

Sony Interactive Entertainment chose Aerospike for its PlayStation Network personalization. With 38 million users, Sony needed to process millions of requests a second; this required data access latencies of less than 10 milliseconds. By offloading their back-end Oracle database, Aerospike flash-optimized in-memory processing enabled Sony to meet its price/performance SLAs.

A large fantasy sports platform, Dream11, allows up to 2.7 million concurrent users to play cricket, hockey, and football. To improve response times and handle peak traffic spikes, Dream 11 moved from Amazon ElastiCache to Aerospike for improved performance, cost and low latency leaderboards and contest management. In this design pattern, Aerospike is deployed as a cache and AWS RDS acts as the backend database.

Aerospike is also frequently used to power user profiles, personalization and sessions stores. Online retailer Wayfair sought a replacement for its "outdated" caching platform after encountering performance problems, operational complexity, and other issues with another NoSQL system. They turned to Aerospike for customer scoring and segmentation, online events tracking, onsite advertising, recommendation engines, and other efforts. By 2019, Wayfair had deployed a 7-node Aerospike cluster with 180TB disk space to manage 6 billion master objects. Aerospike serviced an average of 100,000 reads and 20,000 writes per second around the clock, typically with sub-millisecond latencies. By offloading the user profile and limiting writes to their backend database, Aerospike increased their shopping cart size by 30%. As one data architect put it, "Other groups [in our firm] hate us since nothing goes wrong, no downtime."

Getting Started

Aerospike CacheDB: it's a caching solution like no other. Predictable performance at scale. Remarkably low TCO. Ease of integration and deployment. Scalability from hundreds of gigabytes to petabytes. And a collection of built-in, sophisticated features you won't find elsewhere: seamless persistence, strong and immediate data consistency, multi-site clustering, flexible data modeling options, and near 100% uptime.

With Aerospike, you don't have to compromise. Why not contact Aerospike to explore how you can upgrade your data architecture to meet the aggressive demands of today's always-on global economy?

AEROSPIKE

About Aerospike

Aerospike is the global leader in next-generation, real-time NoSQL data solutions for any scale. Aerospike enterprises overcome seemingly impossible data bottlenecks to compete and win with a fraction of the infrastructure complexity and cost of legacy NoSQL databases. Aerospike's patented Hybrid Memory Architecture™ delivers an unbreakable competitive advantage by unlocking the full potential of modern hardware, delivering previously unimaginable value from vast amounts of data at the edge, to the core and in the cloud. Aerospike empowers customers to instantly fight fraud; dramatically increase shopping cart size; deploy global digital payment networks; and deliver instant, one-to-one personalization for millions of customers. Aerospike customers include Airtel, Banca d'Italia, Experian, Nielsen, PayPal, Snap, Verizon Media and Wayfair. The company is headquartered in Mountain View, Calif., with additional locations in London; Bengaluru, India; and Tel Aviv, Israel.

2525 E. Charleston Road, Suite 201
Mountain View, CA 94043
Tel: 408 462 AERO (2376)
www.aerospike.com