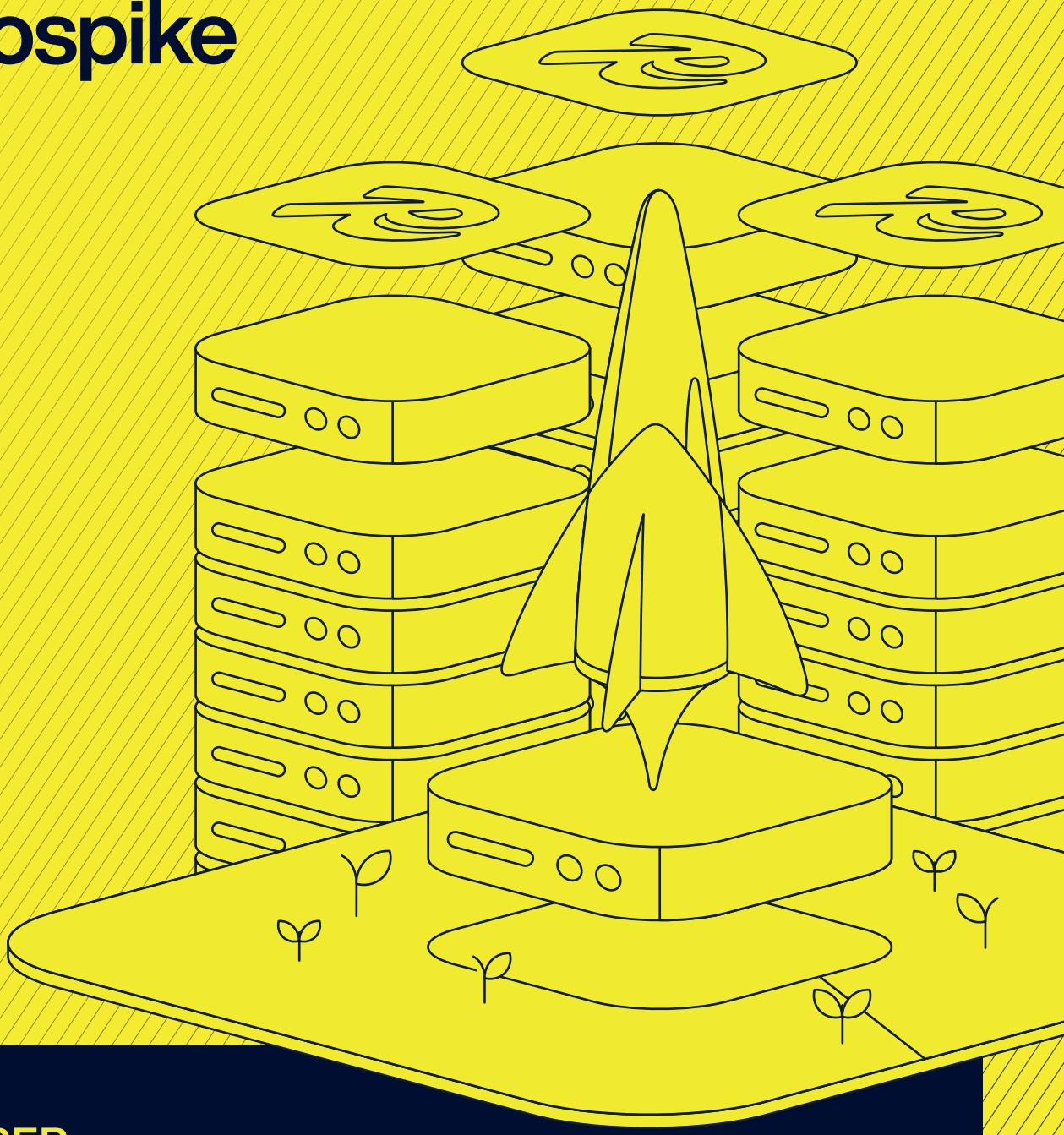
Aerospike



WHITE PAPER

Five signs you've outgrown Redis

Snapshot

Why replace Redis?

You need scalability and elasticity

You need enterprise-grade reliability, stability, and consistency

You need a more precise, efficient, and durable cache

You need a flexible real-time database with dependable storage and persistence options

You need predictable, low TCO without hindering future growth

Introduction

Many firms find Redis easy to use when their data volumes and workloads are modest, but that changes quickly as their needs grow. High total cost of ownership (TCO), poor performance at scale, and operational complexity can lead to budget overruns, service level agreement (SLA) violations, and delayed application rollouts.

What are 5 signs that your organization may have outgrown Redis?

1

You need scalability and elasticity

- Have you struggled to maintain high performance over growing data sets with ROF, Speedb, or RocksDB?
- Are seasonal workload peaks causing service slowdowns and interruptions?
- Do you need a predictable way to grow or shrink your cluster as your workloads change?

2

You need enterprise-grade reliability, stability, and consistency

- Are you struggling with Redis system reliability?
- Are you concerned about data loss or reading stale data? Is eventual consistency insufficient for your applications?

3

You need a more precise, efficient, and durable cache

- Is your Redis cache consuming too many computing resources?
- Do you have too many caches deployed? Could you benefit from consolidating caching instances on a centrally managed cluster?
- Are you concerned about performance degradation from cache misses? Do you wish your cache eviction strategy was more precise?
- Do you wish your in-memory data and LRU information (reads and writes) could survive restarts and crashes?

4

You need a flexible, real-time database with dependable storage and persistence options

- Do you wish Redis persistence was fast, reliable, and straightforward?
- Are Redis snapshots impractical as your data volumes grow?
- Are Redis append-only files (AOF) slowing runtime performance?

5

You need predictable, low TCO without hindering future growth

- Are growing business or workload demands stressing your Redis cluster?
- Are you concerned about Redis hardware, tuning, and operational expenses?

1. You need scalability and elasticity

Having a scalable, elastic, real-time database is increasingly critical as data volumes grow and application demands evolve. Redis struggles on both counts, largely because it was initially designed as a single-instance, single-threaded system for in-memory caching. While recent releases and optional offerings provide some relief, Redis users still find themselves required to add many more nodes with reliance on expensive DRAM to scale Redis.

DRAM is expensive, and managing increasingly large clusters isn't easy. Furthermore, single-threading negates the benefits of multi-core processors and limits the ability to scale up. Scaling out with Redis (i.e., adding more nodes) presents its own challenges. Although some automation of resharding is provided, the process still requires multiple steps from a system operator.

ROF doesn't solve Redis' scalability problems because it keeps metadata and indexes in memory, caches "hot" data for performance, and relies on memory-hungry RocksDB processes behind the scenes. Speedb (a RocksDB-compatible engine) offers some performance improvements and was recently acquired by Redis. However, Speedb is still 2x slower than Redis in memory (rather than the 5x slower speed of RocksDB). For many firms, that's still too slow, and full integration of Speedb with Redis won't be available until Redis 8.0.

Aerospike manages large data volumes and mixed workloads with a fraction of the computing resources that Redis requires. Aerospike's Hybrid Memory Architecture™ (HMA) uses flash storage in parallel to perform reads at sub-millisecond latencies at very high throughput (100K to 1M+ TPS), even under heavy write loads. This enables enormous vertical scaleup at a 5x lower cost than pure DRAM.

Redis configuration requirements inhibit elasticity, too. A cluster can only be scaled out by a multiple of the current number of shards, and users can't remove shards from a cluster. So, quickly scaling up before peak periods or removing them after the fact can be painful and expensive. Not so with Aerospike. Dynamic cluster management, automatic data redistribution, a Smart Client layer, and cost-efficient use of volatile and non-volatile memory (DRAM and SSDs) contribute to Aerospike's exceptional scalability and elasticity.

2. You need enterprise-grade reliability and data consistency

Mission-critical applications require a highly available, reliable database that serves up the most current data and prevents data loss. In this case, Redis falls short.

Redis users often turn to Redis Sentinel or Redis Cluster to improve availability. The former monitors cluster status, alerting users if a primary node fails and assists with failover. However, Sentinel suffers from scalability issues: it's not a clustering solution, all writes go to the master, and sharding isn't supported. Although Redis Cluster is a clustering solution, it doesn't have strong high availability features or support strong data consistency.

Aerospike is designed as a distributed database and requires no further tech stack components to provide clustering. Users typically configure Aerospike to maintain two copies of the data (one primary, one replica) for high availability. The database is largely self-healing, with many network and node failures automatically detected and resolved without data loss, significant performance degradation, or operator intervention. This promotes 24x7 operations, helps firms achieve target SLAs, and reduces operational complexity.

While data consistency requirements vary among applications, having a database that can easily enforce strong consistency while maintaining high performance gives firms a distinct edge, enabling them to use one database to satisfy a wider range of use cases and business needs. Redis supports eventual consistency, which can result in stale reads and even data loss under network partitions. The WAIT command is most closely associated with consistency, yet Redis documentation acknowledges that WAIT does not make Redis a strongly consistent store.

Aerospike supports strong, immediate data consistency for record-level transactions. It guarantees that write transactions will be applied in a specific sequential order and will never be lost. For each read transaction, Aerospike users can choose full linearizability or session consistency. Aerospike also passed the Jespen test, which Redis has yet to do.

3. You need a precise, efficient, and durable cache

While caching helps firms offload work from legacy systems and boost application performance, its effectiveness depends on delivering exceptional data access speeds with few latency spikes and making efficient use of hardware resources to avoid server sprawl.

Redis works well as a cache because of its in-memory performance, but users often complain of excessive DRAM consumption and inordinate growth in cluster size as data volumes increase. Built to work on commodity servers, Redis features a predominantly single-threaded design, so it cannot effectively support today's modern multi-core processors. Furthermore, Redis uses imprecise sampling methods for its least recently used (LRU) data eviction strategy.

Aerospike takes a different approach. It is easily configured as a high-speed in-memory cache with multiple persistence options (e.g., NVMe SSDs or lower-cost, NVMEe-compatible networked block storage, including cloud-based block storage and traditional Network-Attached Storage).

Aerospike can also be deployed as an ultra-fast, real-time database with persistence. Indeed, users can configure Aerospike to store all data and indexes in DRAM (e.g., as a cache), all data and indexes on SSDs (Flash), or a combination of the two (data on SSDs and indexes in DRAM). This flexibility enables firms to use a single data platform for a wide variety of use cases and needs, simplifying infrastructure and thereby reducing costs.

Aerospike provides enterprise-like durability for in-memory systems like caches. It allows in-memory data to survive a restart or crash and persists LRU information (reads and writes) across restarts. These features are virtually unheard of for caches — they're typically associated with databases, making Aerospike a great option for your cache.

Aerospike is massively parallel and exploits modern hardware to maximize runtime performance and cost efficiency. Such innovations include cache-line optimized indexes, lockless data structures, and data partitioning that reduces contention between service threads running efficiently in parallel on multi-core CPUs. SSDs are treated as raw block devices to avoid overhead from standard storage drivers and file systems. Finally, Aerospike Database 7.1 added precise LRU eviction, providing an efficient eviction strategy with multiple options for persisting cached data while still providing very low latencies.

Aerospike's policy-driven control of cached data, efficient use of computing resources, and is low operational costs enable firms to deploy the platform as a shared in-memory cache service. Instead of maintaining dozens or hundreds of separate Redis caches, firms can consolidate these caches into a single Aerospike cluster.

4. You need a flexible, high-performance, distributed database

Growing data volumes and demanding real-time workloads require a high-performance distributed database capable of supporting a variety of applications. However, several design aspects of Redis inhibit its ability to fulfill such business needs.

Redis is a single instance, in-memory database. Clustering was added after the fact to make Redis act like a distributed database. It creates separate child processes for Redis Cluster proxy, persistence, replication, and data consistency. Each of these processes competes for CPU, memory, cache, I/O, and network resources, adding overhead and latency to every workload. Aerospike is a multi-threaded, highly performant distributed database with these capabilities integral to its design.

While Redis Enterprise can group multiple Redis shards into a shared-nothing cluster, its design has certain drawbacks. Incoming client requests are first serviced by a proxy, which determines which server has the desired data and routes the request. This adds latency to each transaction and inhibits workload scaling.

Aerospike was designed as a clustered system with all nodes aware of one another. Its Smart Client™ layer knows how data is distributed in the cluster and directs traffic to the appropriate node(s). Without the overhead of a proxy, client requests can often be satisfied with a single network hop. Automatic client load balancing improves both performance and correctness. Such capabilities ensure the lowest possible latencies.

Data storage is another area in which Redis and Aerospike diverge. Redis stores data as key-value pairs holding a single data type, with no notion of records or bins, so any data operation must be performed on individual key-value pairs. Firms can upgrade to Redis Enterprise for additional data model support (including document, spatial, search, time series, and vector), but this costs more and consumes more resources.

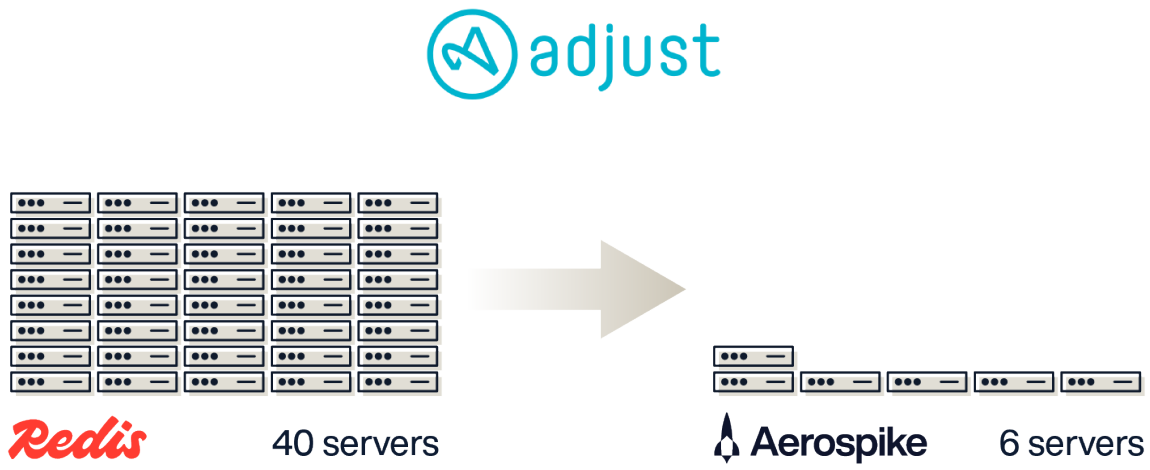
Aerospike enables a single record to contain a key and multiple named fields ("bins"). These bins can hold different types of scalar or complex data so developers can process multiple items (e.g., first name, last name, account balance) with a single transaction. Redis would require three separate transactions to process this sample data. Such overhead hinders performance and scalability in most real-world scenarios. Aerospike's collection data types (CDTs) are the high-performance foundation for supporting document and graph data (note: Redis sunset its graph solution).

Redis publishes benchmarks that use full DRAM instances and only one copy of user data. As many firms have discovered, persistence in Redis, via snapshots and append-only files, can drastically reduce performance and even lead to data loss. Aerospike, by contrast, was built with persistence in mind. In the industry, users will freely share results of their own benchmarks, something Redis users find difficult due to licensing restrictions. Aerospike users often enjoy sub-millisecond data access speeds even with hundreds of billions of records in databases of tens of terabytes (TBs) to several petabytes (PBs).

5. You need to lower your TCO without hindering future growth

Soaring data volumes and competitive pressures are forcing firms to deliver new applications faster and process tens of TBs to PBs of data in real time. Such demands can stress Redis clusters, prompting users to deploy more nodes, memory, and manpower, which drives up TCO.

Dynamic cluster management, automatic data redistribution, a Smart Client layer, and cost-efficient use of volatile and non-volatile memory (DRAM and SSDs) contribute to Aerospike's exceptional scalability and cost efficiency. Aerospike operates so efficiently that it often reduces TCO by 5x or more compared with Redis and other first-generation caching solutions. Aerospike requires substantially fewer servers due to our ability to store data on Flash (while accessing it at DRAM speeds) and employing a replication factor of just two.



By saving 85% in infrastructure, Adjust was also able to lower its maintenance costs.

Even if your data volumes are modest today, what will happen as your needs grow? Scaling Redis requires substantial memory and leads to large clusters, meaning more complexity and frequent node failures.

Do you use Redis as an in-memory cache to speed your access to data managed by a SQL or NoSQL store? If so, you're probably working hard to manage and synchronize both environments so your applications don't access stale data or experience slow performance due to cache misses. Covering the operational costs of scaling two systems can be daunting, particularly as your data volumes and workloads grow. Aerospike manages extremely large data sets on comparatively small server footprints, yielding a reliable, simpler environment.

Get ready for the future

A multi-model database for caching data or real-time transaction processing, Aerospike delivers ultra-fast performance for high-throughput read/write workloads, exceptional scalability, enterprise-grade reliability, and data consistency – all with a low TCO. Firms in finance, AdTech, technology, retail, and other industries rely on Aerospike to support their critical applications.

Compared to Redis, Aerospike offers smart client software, a wider range of deployment and storage options, and built-in support for persistence. Its multi-threaded design, HMA, and self-managing features contribute to the ease and efficiency with which it can scale from 100s of GBs to petabytes.

When configured as an in-memory system, Aerospike operates as a fast, efficient, and durable cache to offload work from other data stores. It enables firms to consolidate multiple caches into a single Aerospike cluster on a fraction of the hardware.

If you're struggling to achieve what you want with Redis, or have experienced any of the 5 signs just discussed, why not explore what Aerospike can do for you? Contact Aerospike to estimate TCO savings for your workload, or try Aerospike and see how you can benefit, too.

Adjust

adjust

Fraud prevention

The decision was made to transition from Redis to Aerospike to enhance server efficiency and bolster data accuracy for fraud prevention. This shift was prompted by scalability and performance challenges encountered with Redis during the expansion phase.

Lowered infrastructure costs by reducing its server count from 40 to 6

Improved latency, failover capabilities, and data accuracy to improve campaign effectiveness

Wix

WIX

User personalization service

Wix's key objectives were enhancing personalization features, reducing costs, and minimizing latency. To adequately manage its vast amounts of data, Wix initially employed a combination of Redis and HBase. However, scalability challenges emerged, rendering Redis inadequate for real-time data processing.

Wix saved 45% in costs

Reduced latency from 18ms to just 2 – 3ms

Gained improved personalization and bolstered the responsiveness of the user experience

TomTom

Navigation and mapping

TomTom set out to slash operational expenses and shrink its environmental impact in a strategic shift driven by the need for enhanced efficiency and sustainability

Were able to handle a 1 TB use case with significant hardware reductions, saving 93% in costs

Prevented outages, saving €100,000 per avoided outage

Reduced its carbon footprint by 86%

About Aerospike

Aerospike is the real-time database built for infinite scale, speed, and savings. Our customers are ready for what's next with the lowest latency and the highest throughput data platform. Cloud and AI-forward, we empower leading organizations like Adobe, Airtel, Criteo, DBS Bank, Experian, PayPal, Snap, and Sony Interactive Entertainment. Headquartered in Mountain View, California, our offices include London, Bangalore, and Tel Aviv.

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

2440 W. El Camino Real, Suite 100, Mountain View, CA 94040 | (408) 462-2376 | aerospike.com