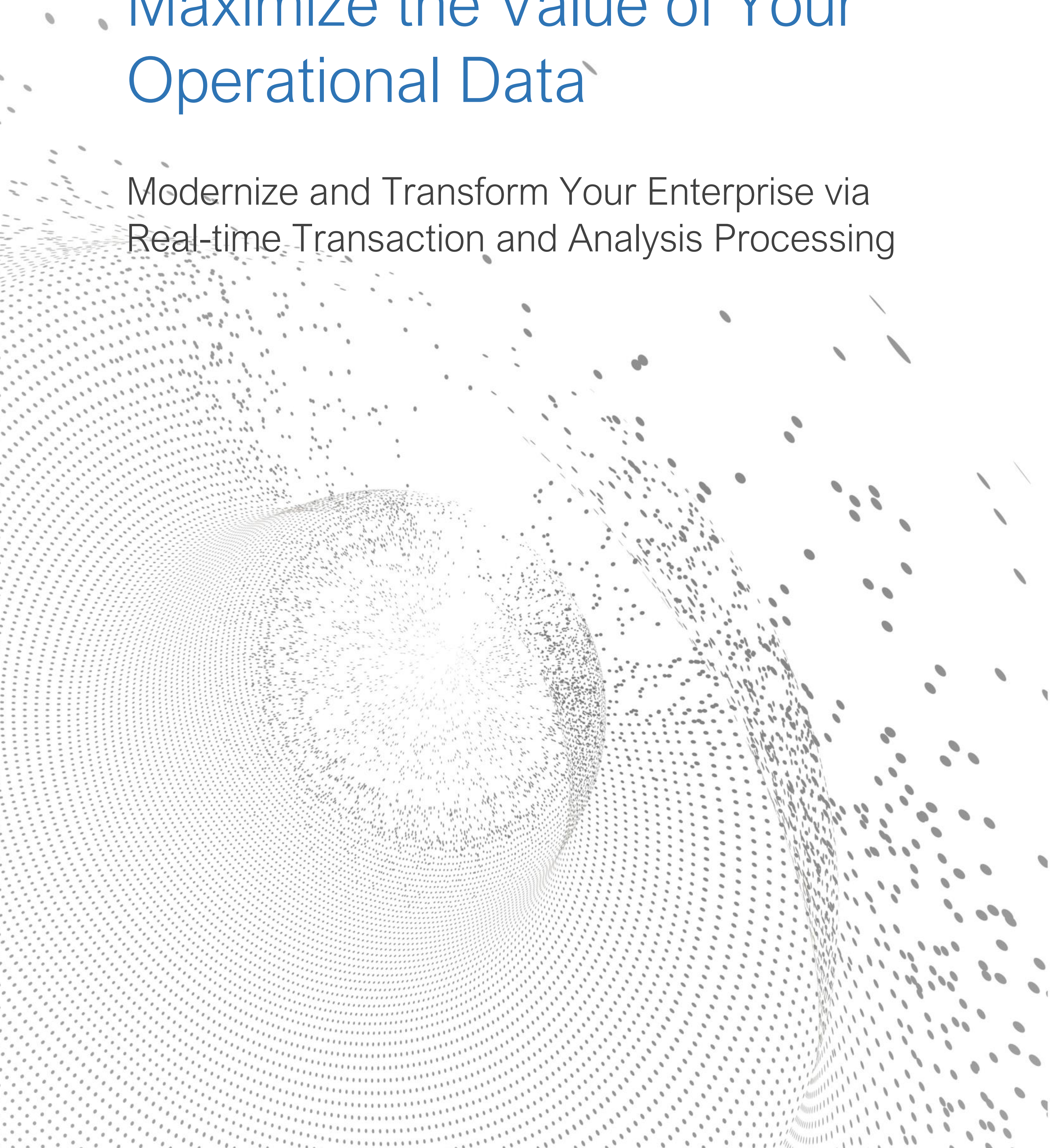
Maximize the Value of Your Operational Data

Modernize and Transform Your Enterprise via Real-time Transaction and Analysis Processing



Contents

Executive overview

It might sound too good to be true: a database system that processes large volumes of operational data in real time while delivering exceptional runtime performance, high availability, and cost efficiency while still keeping your data safe. What if early adopters in banking, telecommunications, and other industries are already harnessing such a database for achieving results that are transforming their businesses in myriad ways? What if published benchmarks demonstrate sub-millisecond response times for high throughput read/write workloads over high data volumes *with* substantial cost savings compared with traditional alternatives?

It's easy to be skeptical when confronted with such information. After all, the technology industry is hardly hype-free. Yet it wouldn't be the first time that hardware advances and software innovations paved the way for new, compelling solutions to pressing business problems that prior technologies just couldn't adequately address.

For firms struggling to manage – and make sense of – high and rapidly growing volumes of operational data in real-time, it is worth exploring how recent advances in database management technology can help achieve these goals. This paper introduces key technologies that Aerospike clients are using to modernize their data management infrastructures and realize such impressive (and seemingly impossible) results as:

- Rapid read/write speeds without extensive tuning or a separate data cache
- Substantially smaller footprints than popular alternatives, often leading to 3-year total cost of ownership (TCO) savings of $3-5 million per application
- 24x7 availability, including cross-datacenter replication
- Operational ease during scale-out and maintenance
- Interoperation with popular software offerings, including Apache Hadoop, Spark, and Kafka

Sounds unbelievable, right?

Perhaps you're wondering if there's some kind of "catch" – maybe the system requires specialized skills or leads to a complex infrastructure. Consider this: the skills required for using Aerospike's platform are relatively common, and core portions of its software are open source. Furthermore, firms often complete their first project within a few weeks and integrate the technology into their existing IT infrastructures within a matter of months.

Curious?

Maximize the value of your operational data – making the case

Mobile computing, IoT, microservices plus other technologies and the general flood of data are all straining existing OLTP infrastructures, generating the need for transformation and/or modernization.

Before delving into the technologies for modernization and transforming how operational data is managed, it's worth exploring a basic question: is it truly worth the effort? After all, firms have been coping with online transaction processing (OLTP) and timely analytics of operational data for decades. Why not leave well enough alone?

Simply put, customer demand, intense competition and growing data volumes are making previously serviceable solutions untenable – not to mention general issues with scaling across datacenters and geos and trying to grow revenue via better real-time customer experience. Mobile computing, social media, smart devices, and micro services are among the popular technologies forcing firms to provide highly customized products, services, and support to their clients around the clock. Combine that with global operations, shifting business alliances, and "co-opetition," and it's not difficult to see how transactional applications are evolving in ways that many existing platforms weren't designed to support. The volume and variety of such data is becoming too great – and so is the need for extremely fast, predictable data access speeds. Of course, downtime must be virtually non-existent, even during software upgrades, hardware failures, and system expansions.

Consider how these modern challenges impact traditional approaches to managing operational data. For decades, firms turned to mainframe-based solutions for this work. While such solutions remain fast, reliable, and secure, they're also expensive to use and maintain (firms get charged typically by the MIP[1]), and they are difficult to adapt to changing business needs. Furthermore, most mainframes are already operating at or near capacity, with few (if any) free cycles available for new applications.

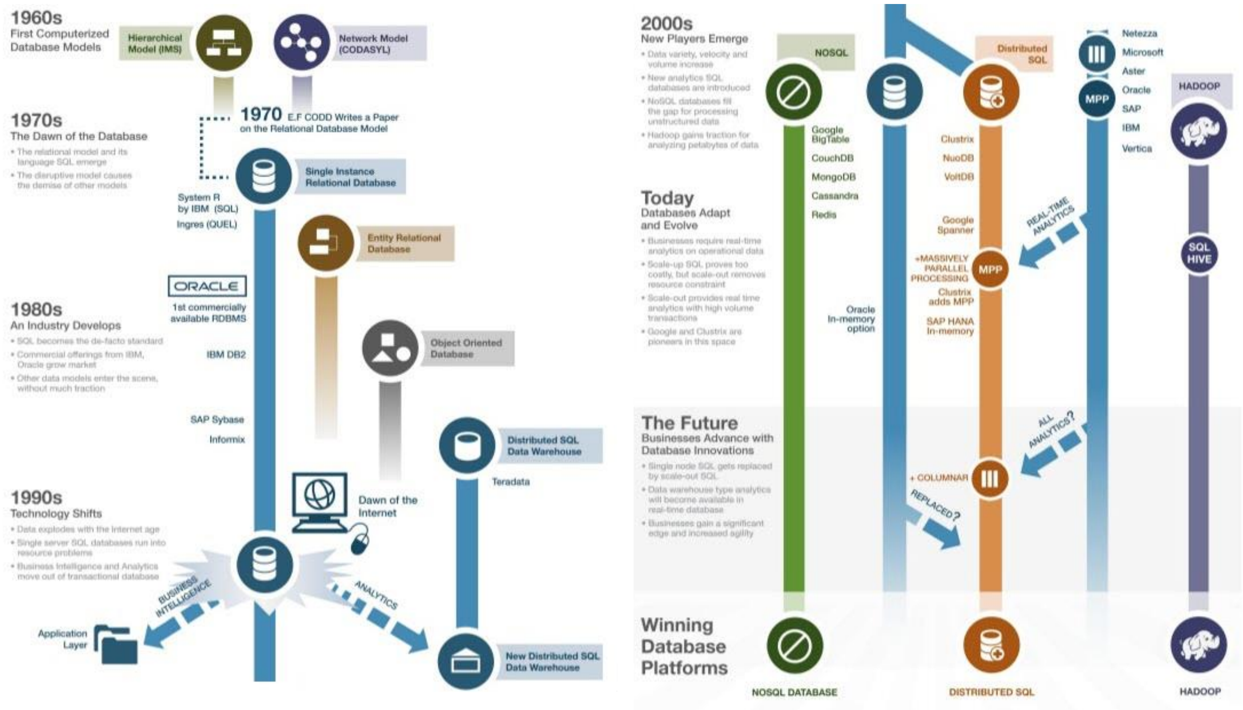

1960s
First Computerized Database Models
Hierarchical Model (IMS)
Network Model (CODASYL)
1970s
The Dawn of the Database
• The relational model and its language SQL emerge
• The disruptive model causes the demise of other models
1970 E.F CODD Writes a Paper on the Relational Database Model
Single Instance Relational Database
System R by IBM (SQL)
Ingres (QUEL)
Entity Relational Database
ORACLE
1st commercially available RDBMS
1980s
An Industry Develops
• SQL becomes the de-facto standard
• Commercial offerings from IBM, Oracle grow market
• Other data models enter the scene, without much traction
IBM DB2
Object Oriented Database
SAP Sybase
Informix
Distributed SQL Data Warehouse
Teradata
1990s
Technology Shifts
• Data explodes with the Internet age
• Single server SQL databases run into resource problems
• Business Intelligence and Analytics move out of transactional database
Dawn of the Internet
BUSINESS INTELLIGENCE
ANALYTICS
Application Layer
New Distributed SQL Data Warehouse
2000s
New Players Emerge
• Data variety, velocity and volume increase
• New analytics SQL databases are introduced
• NoSQL databases fill the gap for processing unstructured data
• Hadoop gains traction for analyzing petabytes of data
NOSQL
Google BigTable
CouchDB
MongoDB
Cassandra
Redis
Distributed SQL
Clustrix
NuoDB
VoltDB
Google Spanner
Netezza
Microsoft
Aster
Oracle
SAP
IBM
Vertica
MPP
HADOOP
Today
Databases Adapt and Evolve
• Businesses require real-time analytics on operational data
• Scale-up SQL proves too costly, but scale-out removes resource constraint
• Scale-out provides real time analytics with high volume transactions
• Google and Clustrix are pioneers in this space
REAL-TIME ANALYTICS
+MASSIVELY PARALLEL PROCESSING
MPP
Clustrix adds MPP
SAP HANA In-memory
Oracle In-memory option
SQL HIVE
The Future
Businesses Advance with Database Innovations
• Single node SQL gets replaced by scale-out SQL
• Data warehouse type analytics will become available in real-time database
• Businesses gain a significant edge and increased agility
ALL ANALYTICS?
+ COLUMNAR
REPLACED?
Winning Database Platforms
NOSQL DATABASE
DISTRIBUTED SQL
HADOOP

Figure 1: Database Evolution
(source – Robin Purohit http://insights.wired.com/profiles/blogs/the-future-of-the-database-infographic)

UNIX-based DBMSs designed for parallel processing emerged years ago as another option. Often less costly and easier for programmers to work with than classic mainframe platforms, such systems need considerable tuning as data volumes and transaction rates grow. SQL-based DBMSs also require database schemas to be defined up front; subsequent changes can be expensive and break existing applications. NoSQL DBMSs typically offer schema flexibility but lack strong data consistency, adequate performance for operational workloads, and high availability.

These challenges – and others – have led many firms to deploy a data caching layer in front of their parallel UNIX or mainframe-based DBMS. This 2-layer approach attempts to meet aggressive performance and scalability mandates that their operational database system couldn't satisfy with tuning and hardware upgrades. Yet a caching layer introduces its own challenges: increased costs (from server sprawl,

[1] http://www.veristorm.com/blog/reducing-cost-mips
http://modernsystems.com/tweak-your-mips-getting-smart-about-mainframe-cost-reduction/

administrative overhead, and heavy use of expensive dynamic random access memory (DRAM), increased complexity, latency issues due to cache "misses" or invalidations, and more.

The essential question to ask is: what would you do if you had a database that is as fast as your cache today and avoided all the issues inherent with cache-based systems? As you'll soon see, Aerospike doesn't need a caching layer to achieve very low latencies for reading or writing operational data at scale. It does that by itself, driving down ownership costs with a smaller server footprint and simplified operations. That might sound impossible, but it's not. It's real. Production applications deployed by Aerospike customers already prove that. The possibilities are quite tantalizing and can spur any enterprise on to improving the top line. Essentially, Aerospike brings the technology used by Google, Facebook, Netflix, or Amazon to within reach of the rest of the world so they can also have their own state-of-the-art game-changing tech that the biggest internet companies have today.

The technology in brief

A unique approach to managing volatile and non-volatile memory in clustered computing environments enables Aerospike to keep ownership costs low while delivering exceptional performance for read/write operational workloads.

Aerospike provides a distributed, highly scalable database management system for demanding read/write workloads involving operational data. It was designed to deliver extremely fast – and predictable – response times for accessing data sets that span billions of records in databases of 10s – 100s TB. Other design features address fault tolerance and near 100% uptime even during upgrades and maintenance.

How? By capitalizing on proven architectural approaches – such as distributed computing and parallelism – and developing new technologies to meet business demands that hadn't even surfaced when older systems were originally built. Indeed, Aerospike's patented Hybrid Memory Architecture™ (HMA) drastically reduces traditional I/O and network communication compared with other approaches; it also uses CPU resources considerably more efficiently. The cumulative impact of these features (and others) enables Aerospike to deliver remarkable speed at scale.

So, what *is* Aerospike? Very briefly, it's a key-value database system that runs on a cluster of computing nodes using a shared-nothing architecture, as shown in the Fig. 2. For maximum flexibility, enterprises can deploy Aerospike on premises or in cloud environments. Intelligent client and server layers address the three common performance challenges we just discussed: I/O, communications, and CPU usage.

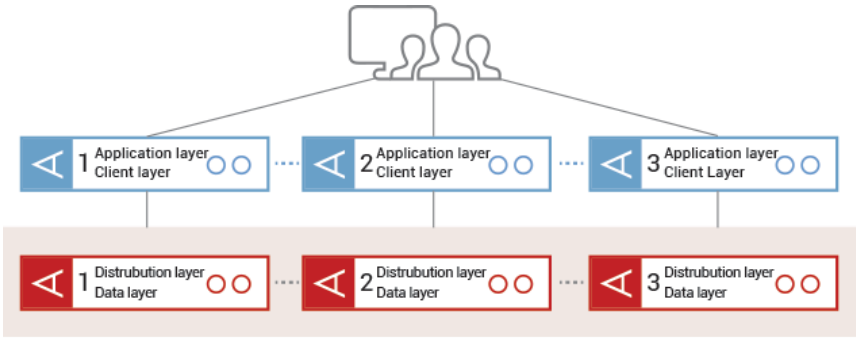


Figure 2: Overview of Aerospike hybrid memory architecture (HMA), including client and server layers. Application layer software is provided by users or third parties.

To minimize I/O, Aerospike keeps indexes in volatile memory (DRAM), thereby eliminating disk reads for index lookups. Furthermore, Aerospike persists user data to fast, non-volatile flash memory (solid state disks or SSDs) in a distinctive way. It doesn't treat SSDs like any other disk; instead, it exploits their unique characteristics. For example, Aerospike treats SSDs as raw devices and writes data in large blocks into a highly efficient, custom file format which also avoids wear-leveling issues common with other providers. Exploiting DRAM and SSDs in this way provides strong runtime performance and scalability at an order of magnitude lower overall cost than other alternatives, both SQL and NoSQL.

But I/O savings is only one facet of Aerospike's high performance. Minimizing network overhead is another. Aerospike's data distribution techniques automatically co-locate index and user data on the same node, reducing the need for inter-node communication to satisfy a data access request. In addition, all nodes are equal in an Aerospike cluster, so client requests aren't funneled through a "master" node for dispatch. Moreover, Aerospike's client layer understands how data is distributed on the server so that it can automatically and transparently route an application's request to the appropriate node(s).

For CPU efficiency, Aerospike's multi-threaded design exploits the non-uniform memory access (NUMA) architecture of modern multi-core processors. Written in C, Aerospike avoids inefficiencies associated with Java runtimes, such as garbage collection. Furthermore, Aerospike custom-tuned certain C libraries (e.g., jemalloc) for memory management to avoid fragmentation. Finally, Aerospike's cooperative scheduling approach balances CPU use across various types of requests ranging from simple record retrievals to more time-consuming batch queries or scans.

Aerospike gets clusters up and running automatically with its dynamic partitioning scheme that auto-balances the data. It does this with a robust hashing technique – all of which avoids hot spots. Aerospike also synchronously replicates data across multiple nodes to ensure high availability during hardware or network failures. When a node is added to or removed from the cluster, Aerospike automatically redistributes the data without incurring any downtime or administrator involvement. Such characteristics contribute to Aerospike's design objectives of providing fast – and predictable – runtime performance, high scalability, and high availability.

Applications and use cases

Firms in banking, telecommunications, advertising, and other industries have transformed their operational data infrastructures to satisfy aggressive real-time SLAs and improve cost efficiency – at the same time.

Enterprises often turn to Aerospike to handle large volumes of transactions from customer-facing applications and to support new applications requirements that frequently arise with "systems of engagement" – i.e., systems that foster collaboration and interaction with users. Social media, mobile devices, cloud computing, real-time chats, streaming data, and other technologies are enabling individuals to engage with enterprises in new ways, driving IT organizations to develop systems that support such engagement.

Aerospike's transactional capabilities have prompted some firms to use its technology as a system of record as well. This is a more classic data management application in which the underlying system provides a single authoritative record of critical, high-fidelity business data, such as account information or sales records.

Applications that benefit from Aerospike typically share some or all of these characteristics:

- Service-level agreements (SLAs) that require sub-millisecond database response times.
- High throughput for mixed workloads (e.g., 3 – 5 million operations per second).
- Support for managing billions of business records in databases of 10s – 100s TB.
- High availability and fault tolerance for mission-critical applications.

- High scalability for handling unpredictable increases in data volumes and transactions.
- Adaptable infrastructure for managing varying types of data with minimal effort.
- Low total cost of ownership (TCO).

Sample scenario

Firms often reduce their operational database server footprint by 10:1, saving millions in ownership costs.

According to a 2017 survey conducted by Forrester Research, firms that deployed Aerospike HMA often reduced their database server footprint by a factor ranging from 4 to 20. That's quite striking, so let's walk through a simple example.

Consider a firm that needs to manage 10TB of unique operational data consisting of 10 billion objects with an average size of 1K per object. Two popular options are: (1) a caching layer with an operational data store or (2) Aerospike. To compare these options, assume the firm will use servers with roughly 200GB free DRAM and 4TB in SSD storage per node. As Table 1 shows, the first option would require 150 nodes just for the caching layer; more nodes would be needed for the ODS and data storage. By contrast, Aerospike would recommend only 15 nodes to achieve comparable runtime performance *and* store the data.

	Cache + Operational Data Store (ODS)			Aerospike		
	Total data	# nodes	Notes	Total data	# nodes	Notes
DRAM	30TB *10TB x 3 replicas*	150	200GB * 5 nodes = 1TB without replication, 3TB with replication	1.2TB *indexes only*	6	1 billion objects @ 1K apiece, each with 64-bit index key
Storage	Depends on ODS	Depends on ODS		< 58TB	15	28.7TB for replicas and overhead. Target 50% disk usage for 57.4TB. 57.4TB / 4TB per node = 15 nodes
Total nodes	>150			15 (higher of DRAM, storage)		

Table 1: Capacity planning comparison for sample client scenario. Goal: Real-time access and management of 10TB operational data (1 billion objects with average size of 1K)

How can Aerospike do this? Its unique use of volatile and non-volatile memory, data distribution techniques, client-side optimizations, and other features enable Aerospike to do more with less. You'll learn more about Aerospike's distinguishing technologies shortly. But for now, let's turn to another important topic: how firms are using Aerospike.

Client profiles

Explore how Aerospike clients are streamlining their operational database infrastructures to achieve tangible – and compelling – business benefits.

To give you a better idea of how firms are using Aerospike today, we'll walk through customer use cases in digital payments, financial services, telecommunications, and advertising technology. These represent just a few application domains in which Aerospike's technology can prove essential to database modernization efforts. For more details on these and other customer case studies, visit Aerospike's website.

Fraud detection for digital payments. A global payments provider is minimizing its annual fraud losses by improving its applying fraud detection algorithm SLAs by a factor of 30x with Aerospike. The firm moved from a 2-layer architecture consisting of Oracle RAC and 360 Terracotta servers to a 20-node Aerospike cluster. Now it runs fraud detection rules against 99.95% of its transactions within its target SLA of 750 milliseconds. Fig. 3 illustrates the firm's fraud detection infrastructure.

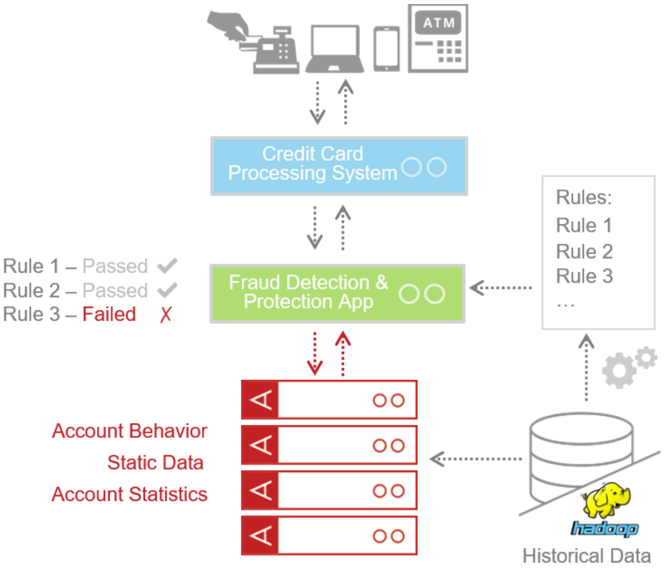


Figure 3: Aerospike manages critical account information for a global digital payments provider, so the firm can apply extensive fraud detection rules in real time to incoming transactions.

Intra-day trading for financial services. A global investment bank with $3 trillion in client assets uses Aerospike to offload work from its mainframe solution for financial trades. Initially, it used a caching layer over its mainframe DBMS to improve response time. When the firm determined that the cache would need to grow from 150 to 1000 nodes to meet future business needs, it turned to Aerospike. Today, a 12-node Aerospike cluster serves as the system of record for intra-day trades, replacing the cache and offloading some work previously done on the mainframe. The mainframe continues to serve existing applications, and the firm regularly transfers data between the two systems, so each can fulfill application-specific needs. With its modernized infrastructure (shown in Fig. 4), the firm now enjoys a five-fold increase in processing speeds. Database access times dropped to sub-milliseconds even though the database size increased from 4 to 14TB. Furthermore, Aerospike enabled the firm to accomplish this with 90% fewer servers deployed, saving an estimated $10,000 per trading day.

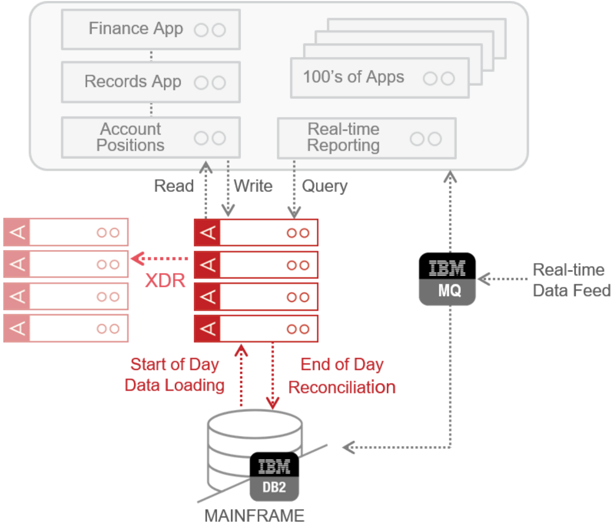


Figure 4: Aerospike manages intra-day trading data for a major financial services firm, saving an estimated $10,000 per trading day

Real-time billing for telecommunications. Consumer demand for mobile data services has prompted telecommunications providers to develop sophisticated, personalized data plans. Real-time tracking of data usage at a granular level – such as the source of a data packet – is critical for generating accurate bills, as some carriers offer unlimited data for specific streaming services (such as Spotify) but not others (such as YouTube). Several firms now rely on Aerospike to keep pace with these billing challenges. Typically, they deploy Aerospike at the edge of their networks to monitor traffic, record chargeable events, and update user device settings to reflect authorized changes to data plans. The systems replaced by Aerospike in these use cases include Oracle TimesTen and Coherence as well as internal home-grown in-memory systems in some cases. Such applications require Aerospike to manage several terabytes of data and process up to 200,000 transactions per second. Going forward, Aerospike, in its Strong Consistency version, can be reliably used for directory lookup of devices in a mobile network.

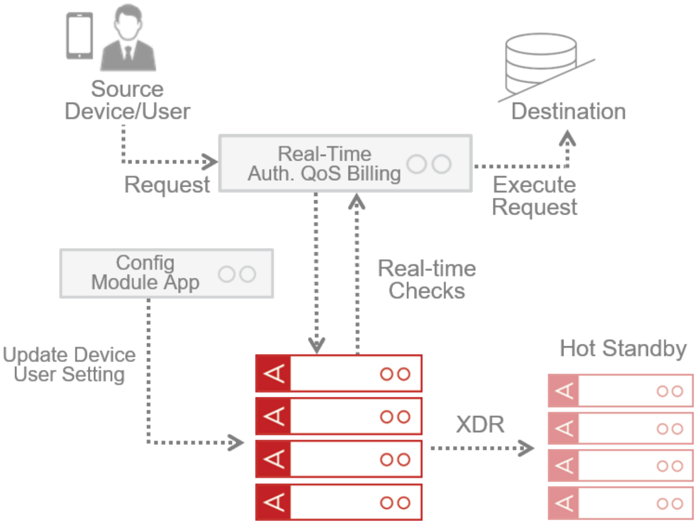


Figure 5: Aerospike helps Telcos manage real-time billing for generating granular bill accuracy with 200,000 TPS

Real-time bidding for online advertisements. Advertising technology (AdTech) firms were among Aerospike's early adopters and have driven Aerospike's dominance as the preferred database infrastructure for real-time bidding applications in that industry. AppNexus, InMobi, and others rely on Aerospike to power online auctions for advertising space that occurs in real time just before a web page is displayed. SLAs in this field are extremely demanding: 50 milliseconds for real-time bidding and 100 milliseconds for rendering the ad. For each real-time bid, multiple database reads and writes occur to execute a single transaction, all within these super-tight SLA windows – a challenging situation that is compounded by the magnitude of the database store of billions of records and up to 100 TB of data.

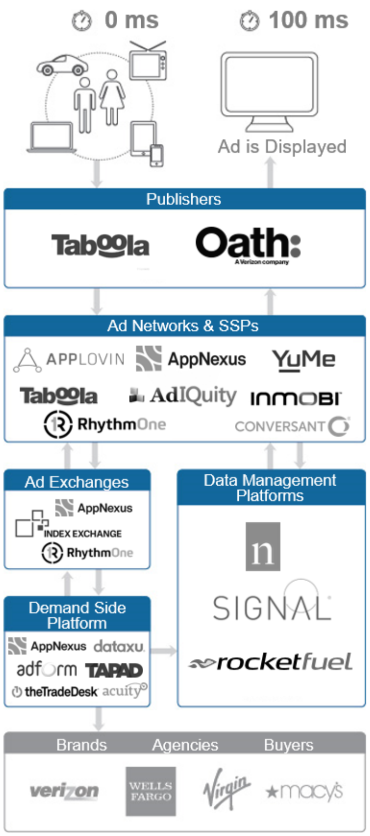


Figure 6: Real-Time Bidding Ecosystem for online advertisements

Key features and technologies

Modernizing and transforming operational data requires architectural advances geared for speed, scalability, availability, and cost efficiency.

To help you understand how Aerospike delivers speed, scalability, cost efficiency, and other capabilities to enterprise clients, this section provides a deeper dive into the Hybrid Memory Architecture™ (HMA) and supporting features.

As mentioned earlier, Aerospike is a shared-nothing database system that operates on a cluster of commodity server nodes. Each node consists of CPU, DRAM, and disks (typically SSDs), and nodes that communicate with one another via TCP/IP. All nodes are considered identical, with no "master" node that could become a bottleneck or single point of failure.

Server-side components of Aerospike manage and distribute index and user data, enforce record-level consistency, interact with storage devices, and perform other critical functions. The client layer supports popular programming and scripting languages as well as manages communication between applications and the server, using its knowledge of the active cluster configuration to identify the best server node(s) to target for the client's request.

Simple, flexible data model

It's a schema-free, key-value data store.

Aerospike is a NoSQL platform with data modeled in records containing keys and related values. Records in the same set can have different named fields (or "bins"), and a single bin can contain different types of data. Null or missing values aren't stored; if a record lacks a particular field, no bin is maintained, providing efficient storage for sparsely populated data sets. Finally, the structure of a set doesn't need to be defined up front, making it easy to accommodate new application requirements.

Since most people are familiar with relational DBMSs, let's use them as a reference point. A "set" in Aerospike is analogous to a table in that both manage a collection of related data. A "record" in Aerospike consists of a unique key (like a table's primary key) and one or more named "bins" (columns) that store values. These values are strongly typed, but the bins themselves aren't; this provides considerable flexibility. For example, one record in the Customer set could contain an integer country code in the Location bin, while another Customer record could use a string for the Location. By contrast, relational DBMSs require a column's data type to be declared when the table is created, and all rows must store data of the same type in that column. In addition, Aerospike provides certain complex types that relational DBMSs may not (such as list, map, and geospatial types). Like many relational DBMSs, Aerospike supports multiple "namespaces" (databases) simultaneously.

Hybrid memory subsystem for speed at scale

Aerospike exploits volatile and non-volatile memory in a distinctive way, providing rapid access to index and user data.

Aerospike supports two memory subsystem configurations, classic in-memory where both data and index are stored in DRAM (backed in a persistent rotational disk or SSD) and Hybrid Memory, unique to Aerospike, where the indexes are stored in DRAM and data is directly stored on SSDs (PCIe, NVMe, Optane, etc.)

For the Hybrid Memory subsystem, Aerospike maintains primary and secondary indexes in DRAM for ultra-fast access, while user data is typically stored in flash memory on SSDs. As shown in Fig. 7, Aerospike treats SSDs as low-level block devices, using a proprietary log-structured file system that supports parallelization across all devices on a chassis. In addition, Aerospike buffers write operations, persisting data to SSDs in large block writes to achieve optimal performance.

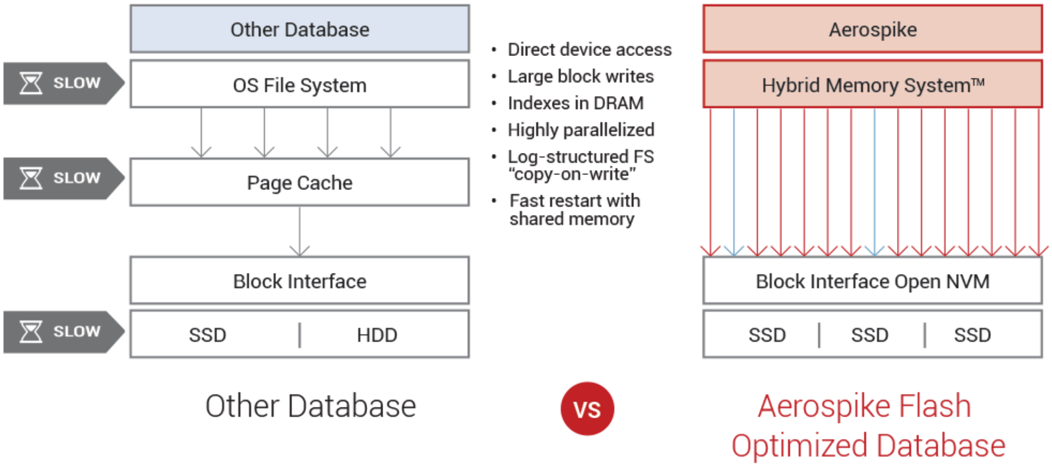


Figure 7: Aerospike exploits flash memory (SSDs) in ways other systems don't

Aerospike indexes use 64-bit keys and efficient structures to conserve DRAM. These indexes are kept in a shared memory space separate from other server processes to avoid the need to rebuild the index whenever an Aerospike process must be restarted on a node. Finally, Aerospike co-locates index and user data on each node to minimize cross-node communication.

At this point, you might be wondering how Aerospike distributes data within a cluster, as hot spot avoidance is a common problem facing any distributed database environment. Aerospike uses a sophisticated hashing and partitioning scheme. In brief, Aerospike hashes each primary key value into a 160-byte digest. This digest space is further divided into 4096 non-overlapping partitions, which are automatically distributed across the cluster. Aerospike's client layer understands this partitioning scheme and leverages this knowledge to further speed runtime performance, as we'll discuss next. Changes to the cluster's configuration – whether planned or unexpected – automatically trigger Aerospike to dynamically rebalance the data, as a subsequent section will discuss.

For high availability and fault tolerance, Aerospike keeps two copies of user data on different nodes in the cluster. Data replication occurs synchronously with immediate consistency within a cluster, while replication across clusters (if configured) occurs asynchronously.

Smart client for "one-hop" access

An intelligent client layer minimizes costly network "hops" needed to access data.

Aerospike's client layer is an open source library that supports popular languages (including C, Java, PHP, Python, and others) and maintains awareness of the cluster's current state. Its API supports creating, reading, updating, and deleting (CRUD) data as well as performing other functions, such as querying data. Indeed, Aerospike's query API uses SQL-like `SELECT` statements, and programmers can even create their own user-defined functions. Developers who have worked with other key-value database systems are likely to find many aspects of Aerospike's API to be quite familiar and, therefore, easy to use.

However, Aerospike's client layer includes certain optimizations that aren't common in other systems – optimizations that help drive Aerospike's runtime performance advantages. Its client layer maintains information in memory about how data is distributed in the cluster – specifically, a mapping of data partitions to nodes. Consequently, Aerospike can route an application request directly to an appropriate node, minimizing costly network "hops." Indeed, retrieving a data record typically involves only one "hop": the Aerospike client layer requests the record of interest from the node where it resides and the server returns it. Requests that span the cluster – such as scans or queries – are sent to multiple nodes for parallel processing. To keep the map of partitions and nodes up-to-date, the client layer periodically checks its version against the latest version on the server nodes, requesting a refresh when needed.

Consider how this approach differs from others. Some use a proxy service to process and route each client request. Others rely heavily on inter-node communications, with the node originally receiving the request often re-routing it elsewhere. In either case, latency increases, and runtime performance suffers. Aerospike's smart client approach avoids such problems, providing faster runtime performance.

Data consistency and availability

Immediate record-level consistency and high availability are guiding principles.

While some alternate solutions support only delayed or eventual data consistency, Aerospike was designed with ACID (atomicity, consistency, isolation, and durability) properties of traditional DBMSs in mind. As such, the system supports these key properties at a record level, providing firms with fast data access in a familiar context.

In Aerospike, each record operation is applied atomically – i.e., as a separate transaction. This differs from relational DBMSs, which enable operations on multiple records to comprise a single transaction. Although more restrictive, Aerospike's implementation affords many runtime performance benefits not possible with multi-record transactions. Aerospike supports durability by synchronously replicating data to multiple nodes. Furthermore, it guarantees that programmers reading data after a write will see the correct (i.e., most recently committed) version of the data. Latches and short-term record locks provide appropriate isolation from operations pending from other clients.

Like any distributed database system, Aerospike needed to address data consistency and availability issues that can arise with network failures. If some nodes become unreachable, should it favor full data consistency (ensure applications always read the most recent write else return an error) or full availability (ensure applications always see a non-error response)? Most commercial offerings favor one or the other. But Aerospike enables firms to tailor the behavior of each namespace to their needs. For deployments that require certain read/write speeds with no down time, Aerospike can favor availability; indeed, this is the default mode. For deployments that require one authoritative source of data, Aerospike can enforce strong consistency. By giving enterprises a choice, Aerospike enables firms to deploy the same high-performance database management infrastructure for both types of applications. In Aerospike, Strong Consistency is possible while still preserving virtually all of the performance in AP mode in certain configurations. The combination of Strong Consistency and High Performance is what makes Aerospike unique among all data bases that exist to-date.

You may be wondering how Aerospike copes with network outages and other changes in cluster state. While specific self-healing actions vary depending on the system's configuration and the type of failure, it's worth exploring a few common scenarios with the default configuration in mind.

To ensure overall cluster reliability, each node periodically broadcasts a "heartbeat" message, essentially indicating its readiness for work. Transaction requests between nodes also count as heartbeats. Aerospike uses heartbeats to determine the cluster's state, such as the failure or addition of a node. When the system detects a change in state, it automatically starts a consensus voting process to arrive at a consistent, current view of the cluster across nodes and subsequently trigger data rebalancing. In this way, the system scales easily when new nodes are brought online. In addition, data remains highly available and access times remain relatively predictable even when a node fails.

Aerospike also supports rolling upgrades with no disruption of service. In particular, a single node can be taken offline for software upgrades and brought back into the cluster when ready with no loss of data availability or consistency, thanks to Aerospike's built-in data replication technology. Furthermore, a single cluster can operate with nodes running different software versions during a rolling upgrade, avoiding the need to take the cluster offline to synchronize software levels on all nodes. For rapid scale-out, multiple nodes can be added to the cluster at the same time.

Security for your data

Access management controls and transport encryption protect sensitive data.

Aerospike supports many of the common security features of traditional database systems. For example, administrators can confer various permissions to individual users, groups of users, or user roles. Permissions are granted at a set or namespace level, depending on the tasks involved. Users authenticate to Aerospike using internally managed passwords, and a future release is targeting support for external authentication methods such as Kerberos and LDAP. If desired, Aerospike can maintain an audit trail of security violations, successful data transactions, successful authentications, and other events.

Transport-level (i.e., network) encryption leverages industry standard techniques, including bidirectional certification authentication, TLS (Transport Level Security) 1.2, and AES (Advanced Encryption Standard) 256. Encryption for data at rest (i.e., stored data) can be achieved with a high performance encrypted block device or a "shadow device" cloud configuration. In general, these approaches have minimal (if any) impact on performance.

Cross-data center replication

Asynchronous replication across data centers provides disaster recovery.

For continuous operations during a natural disaster or other crisis that compromises an entire cluster, Aerospike offers cross-datacenter replication (XDR), which transparently and asynchronously replicates namespaces or specific sets to one or more remote clusters. Administrators can specify various replication strategies, including active/active, active/passive, chain, star, and multi-hop.

Replication is log-based. Clients write to their "local" cluster, and XDR logs minimal information about the change (not the full record). For efficiency, XDR retrieves changed data records from storage in batches and ships only the latest version of a given record. This allows multiple local writes for one record to generate only one remote write – an important feature for "hot" or frequently updated data. Additionally, XDR maintains a pool of open connections for each remote cluster, allowing replication requests to be pipelined in a round-robin fashion.

Integration with popular third-party software

Ready-made connectors, a publish/subscribe messaging system, and partner offerings help firms integrate Aerospike into their existing IT infrastructures.

Curious about how Aerospike works with other software offerings? Aerospike and the community have developed various connectors and plug-ins, and many include sample application code to help programmers get off to a quick start. As of this writing, connectors or plug-ins are available for Apache Hadoop, Apache Spark, Apache Kafka, the Spring framework, ASP.NET, and more.

For firms that need to trigger external actions based on changes to user data in Aerospike, a publish/subscribe messaging system serves as a handy mechanism. Programmers can write data from Aerospike to Kafka message feeds. Subscribing applications can read this data and act as needed, such as writing information to a relational data warehouse, for example.

In addition, Aerospike has cultivated a partner ecosystem that offers hardware, software, cloud services, systems integration, and other support for Aerospike clients. Finally, Aerospike provides consulting services to help firms integrate its technology into their IT infrastructures. This includes moving data between other sources (such as RDBMSs and Hadoop) and Aerospike.

Summary

Transforming IT infrastructures by leveraging operational data in real-time can yield compelling business results. Indeed, firms that have teamed with Aerospike cite remarkable performance, substantial cost savings, and increased business agility among the key benefits they're enjoying for their mission-critical applications. Not surprisingly, a 2017 market survey by Forrester Research cited very high levels of satisfaction.

While such achievements might sound too good to be true, production use has proven otherwise. Earlier sections of this paper took you through several client examples as well as presented an overview of Aerospike's HMA to help you understand how it achieves such results. Innovative use of volatile and non-volatile memory, exploitation of clustered computing environments, sophisticated data distribution techniques, and a “smart client” software layer are just some of the technologies that make this possible.

Yet, for all its strengths, Aerospike isn’t the solution to every data management problem. It’s not designed to replace existing data warehouses, OLAP (online analytical processing), or data lake infrastructures that are commonly supported by relational DBMSs or Hadoop. Instead, it’s best used to support applications that require strong, immediate record-level consistency for high throughput transactional applications and very low data latencies.

While exploring any new technology involves some effort and risk, innovation often drives new opportunities. And, as this paper discussed, Aerospike clients often save millions of dollars by shrinking the server footprint required for their critical applications. Maybe you still find all this hard to believe. Or maybe you’re not convinced that your business can enjoy the same type of results. That’s understandable. To find out if Aerospike can really impact your business and deliver tangible results, why not take the next step? Contact us for a personalized total cost of ownership (TCO) estimate.

Interested in delving deeper into Aerospike’s technology? Similarly, contact Aerospike to arrange a technical briefing or discuss potential pilot projects. If you’d prefer to explore the technology independently, you’ll find plenty of materials available on the Aerospike web site, including free on-demand webinars, benchmarks, and product documentation. You can even get your hands dirty by downloading the free Community Edition or signing up for the Aerospike Quick Start program.

So why wait? With other enterprises already blazing the trail – and reaping substantial rewards – doesn’t it make sense to see if Aerospike might benefit you?

About Aerospike

Aerospike is the world's leading enterprise-grade, internet scale, key-value store database whose patented Hybrid Memory Architecture™ enables digital transformation by powering real-time, mission critical applications and analysis. Only Aerospike delivers strong consistency, predictable high performance and low TCO with linear scalability. Serving the financial services, banking, telecommunications, technology, retail/ecommerce, adtech/martech and gaming industries, Aerospike has proven customer deployments with zero downtime for seven years running.   Recognized by industry analysts as a visionary and leader, Aerospike customers include Nielsen, Williams Sonoma, Kayak, Neustar, Bharti Airtel, ThreatMetrix, InMobi, Applovin and AppNexus. Aerospike is based in Mountain View, CA, and is backed by New Enterprise Associates, Alsop Louie Partners, Eastward Capital Partners, CNTP and Silicon Valley Bank.

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

20180425