Aerospike Multi-site Clustering: Globally Distributed, Strongly Consistent, Highly Resilient Transactions at Scale

Contents

Executive Summary

With strong, immediate data consistency and rack awareness capabilities, Aerospike's multi-site clustering capability enables firms to operate a single database cluster across multiple locations without risking data loss or restricting data availability. That's in marked contrast to many other database platforms, for which the very idea of operating a cluster across geographically dispersed data centers or cloud regions is too much of a stretch: costs would be too high, data inconsistencies would be too great, and resiliency would be too limited.

For businesses that require highly available inter-region transaction processing, the advantage is clear: processes that once took hours or days to complete can often be executed within seconds or minutes on an Aerospike multi-site cluster without sacrificing data correctness or reliability. With Aerospike, your applications don't need to cope with complex conflict detection and resolution scenarios. That's because Aerospike doesn't allow conflicting writes to occur -- it actively avoids them, so you don't need to worry about lost updates or other data inconsistencies. With Aerospike, new types of applications involving globally distributed transactions are now feasible and relatively straightforward to implement. Firms in the financial sector have proven that. Indeed, Aerospike provides firms in banking, financial services, telecommunications, technology, and other industries with a resilient NoSQL platform for maintaining an immutable, secure, and auditable transaction history at a low total cost of ownership (TCO).

If that sounds hard to believe, consider that Aerospike enjoys an exemplary reputation for its highly scalable, reliable operational database platform that delivers ultra-fast read/write speeds with strong data consistency at an attractive price point. For more than 10 years, firms around the globe have been using Aerospike for mission-critical applications, often cutting their server footprints up to 90% and achieving TCO savings of $1 to $10 million per application when compared to other alternatives. No other vendor is so well-positioned to deliver a comprehensive and compelling solution for supporting inter-region clusters as Aerospike.

If you're not already familiar with Aerospike, a separate white paper introduces its architecture and describes its distinguishing features. This paper will help you understand Aerospike's multi-site clustering capabilities. You'll learn how Aerospike provides strongly consistent updates (with no data loss), accepts application requests at all sites, supports immediate failover, and continues to operate without manual intervention in most failure situations. You'll also explore how an Aerospike multi-site cluster behaves under normal and failure scenarios. But first, let's review some sample use cases of this technology.

Applications and Use Cases

Economic globalization and ever-changing client demands have forced companies to compete and collaborate in ways that were once unthinkable. As a result, modern transactional applications are stressing existing IT infrastructures well beyond their design points. Applications such as trade settlements, global supply chain management, currency exchanges, parcel tracking, smart contracts, and others typically require a highly resilient, geographically distributed database platform with strong consistency and reasonable runtime performance to meet their target service level agreements (SLAs).

That's why Aerospike enhanced its platform to support strong, immediate data consistency across multiple data centers (or cloud regions) in a manner that provides fast local reads and keeps write latencies within a few hundred milliseconds. If a data center (or cloud region) becomes unavailable, failover is generally automatic and quick -- and committed writes are never lost. Aerospike's multi-site clustering capabilities complement its Cross-Data Center Replication (XDR) offering, which supports asynchronous replication of data across data centers. In an Aerospike multi-site cluster, writes are synchronous across data centers. Multi-site clusters are particularly useful for supporting many emerging transactional applications.

Two distinct financial institutions in the United States and Europe are using Aerospike multi-site clusters to transfer money between member banks within seconds. In each case, Aerospike stores the state of payment transactions and guarantees the immediate consistency of data shared across geographically distributed applications.

Banking payment transactions require a safe, multi-step process involving request validation, fraud detection, withdrawal, deposit, failure management, confirmation, and more. As you might imagine, payment transactions must be completed quickly with no loss of data, and the infrastructure must be sufficiently resilient to cope with various system failures without compromising data availability. Aerospike provides the accurate state of the transaction to all participating services so that the transfer can be completed seamlessly and promptly.

Fig. 1 illustrates the architecture that one American financial institution is using for its next-generation payments infrastructure, which relies on a messaging platform and Aerospike to enable clients to transfer funds between member banks in real time within a few seconds. Even small member banks (such as credit unions) can participate in such transfers, broadening the traditional market base. The architecture is designed to process 3000 transactional messages per second. It's worth noting that each message can generate nearly a dozen read/write database operations, each of which is processed as a separate database transaction. Supporting this architecture is a two-region Aerospike multi-site cluster that spans the eastern and western United States.

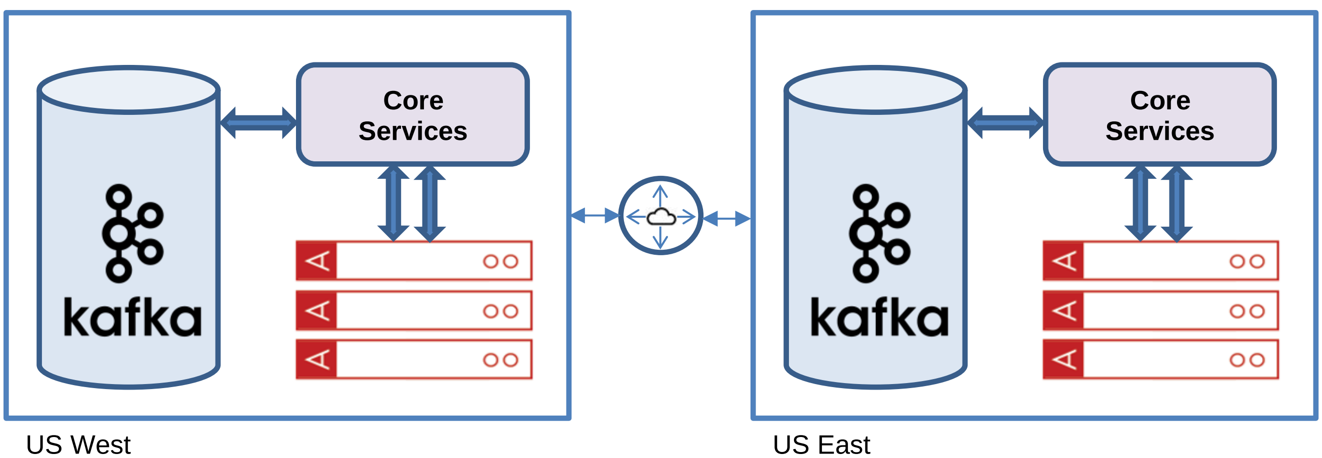

Figure 1: Two-region Aerospike Multi-site Cluster Supports Inter-bank Payments

An Aerospike multi-site cluster is deployed in a similar way in Europe to support the TARGET Instant Payment Settlement (TIPS) service, shown in Figure 2. TIPS enables individuals and firms in various European locations to transfer money between each other within seconds, regardless of the time of day. To track payment state, a major European bank deployed a single Aerospike cluster across two data centers, each with three nodes. This Aerospike infrastructure readily met the bank's target of processing 2000 transactions per second and up to 43 million transactions per day with round-the-clock availability. It also supported the bank's mandate that costs be within €0.0020 per payment. Other solutions didn't meet the bank's objectives for resiliency (100% uptime), consistency (no data loss and no dirty or stale reads), and low transaction cost. The bank is planning to add a third data center to the cluster to increase capacity and further enhance resiliency.

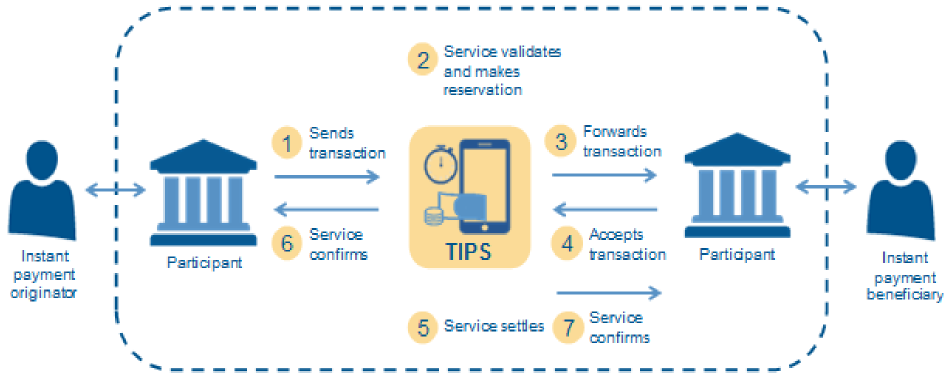

Figure 2: Target Instant Payment Settlement Service in Europe, Powered by Aerospike

Fundamental Concepts

Modern applications like those that we just discussed demand that their database infrastructures be:

- Always available (no planned or unplanned downtime)
- Always right (no data inconsistencies, such as lost data or conflicting writes)

Such requirements imply the need for a distributed database platform spanning multiple geographic locations to be resilient and available during localized disasters that could take out a data center as well as during more mundane failure situations, such as the loss of connectivity to a data center or a node hardware failure at a data center. And while firms naturally expect to spend a bit more and incur some runtime performance overhead for such an infrastructure, any deployed solution must meet reasonable budgetary and SLA targets.

So how have researchers and vendors responded to these challenges? What technologies have they offered to companies seeking a geographically dispersed database platform that's always available and always right?

Active / active databases span multiple regions (at least two) and service application requests at all locations. Thus, each location is "active." Data records are replicated across regions so that reads may be processed at any location. In some architectures, writes of a given data record are only handled at a single master location; other architectures allow such writes to occur at multiple locations.

Each approach has its challenges involving availability, consistency, and (to varying extents) performance. For example, if writes for any given record are allowed at any data center, how is the data kept synchronized and consistent? The two-phase commit protocol, which first debuted with distributed relational DBMSs in the 1990s, solved this problem by having a global transaction coordinator communicate repeatedly with each participant in a two-phase commit protocol to ensure that ultimately the transaction was committed by all or none. This protocol enforced **strong consistency** but was costly and slow. Consequently, other database vendors implemented **eventual consistency,** guaranteeing that eventually all access to a given record would return the same value, assuming no other updates were made to that record before copies converged. But this came at a cost. During normal operations – not just failure scenarios – readers might see stale data. Furthermore, if the same record was updated more than once before all copies converged, conflicts had to be resolved and some data could be lost.

In a moment, you'll explore how Aerospike addresses data availability, data consistency, and performance issues in its multi-site clusters. But before we delve into Aerospike's technology, it's worth briefly mentioning **active / passive** database architectures. Such architectures consist of one active data center that processes all read/write requests. A second, passive database is maintained at a remote data center, standing by in case the active system fails. In such architectures, the passive database is typically updated asynchronously; failover may be automatic or require manual intervention.

Active / active and active / passive architectures are both useful, as they target different business requirements. Aerospike's XDR offering can support active / active or active / passive configurations, but its replication process is always asynchronous. The focus of this paper is Aerospike's multi-site clustering capabilities, which support active / active configurations with synchronous data replication.

The Aerospike Approach

If you're not already familiar with Aerospike, it's a distributed NoSQL system that provides extremely fast – and predictable – read/write access to operational data sets that span billions of records in databases holding up to petabytes of data. Its patented Hybrid Memory Architecture™ delivers exceptional performance using a much smaller server footprint than competing solutions. Hallmarks of Aerospike's design include efficient use of dynamic random-access memory (DRAM), persistent memory (PMEM), and non-volatile memory (solid state disks or SSDs), sophisticated (and automatic) data distribution techniques, a "smart client" layer, and more.

Until recently, an Aerospike cluster needed to reside in one data center or in two data centers that were within 10 miles of each other. Recognizing that the demands of modern transactional applications were pressing firms to deploy shared databases across distant data centers and cloud regions, Aerospike built rack awareness into its engine. This technology, coupled with Aerospike's support for strong, immediate data consistency, allows a single Aerospike cluster to be deployed across multiple geographies with high resiliency, automated failovers, and no loss of data. Furthermore, because Aerospike's fundamental architecture is highly efficient, a multi-site cluster carries a lower TCO than other active / active alternatives.

So, what does an Aerospike multi-site cluster look like, and how does it work? Let's turn to Fig. 3, which illustrates a sample Aerospike cluster spanning three data centers, each with three nodes. Applications perceive this geographically distributed environment as a single system and read/write requests are handled seamlessly. For optimal performance, reads are processed locally, while writes are routed to remote locations, if needed. We'll cover the specifics shortly. But what's important is that Aerospike handles each read/write request as efficiently as possible while preserving strong data consistency. Applications are shielded from the mechanics and don't have to take any extraordinary measures to resolve potentially conflicting updates, cope with stale data, etc.

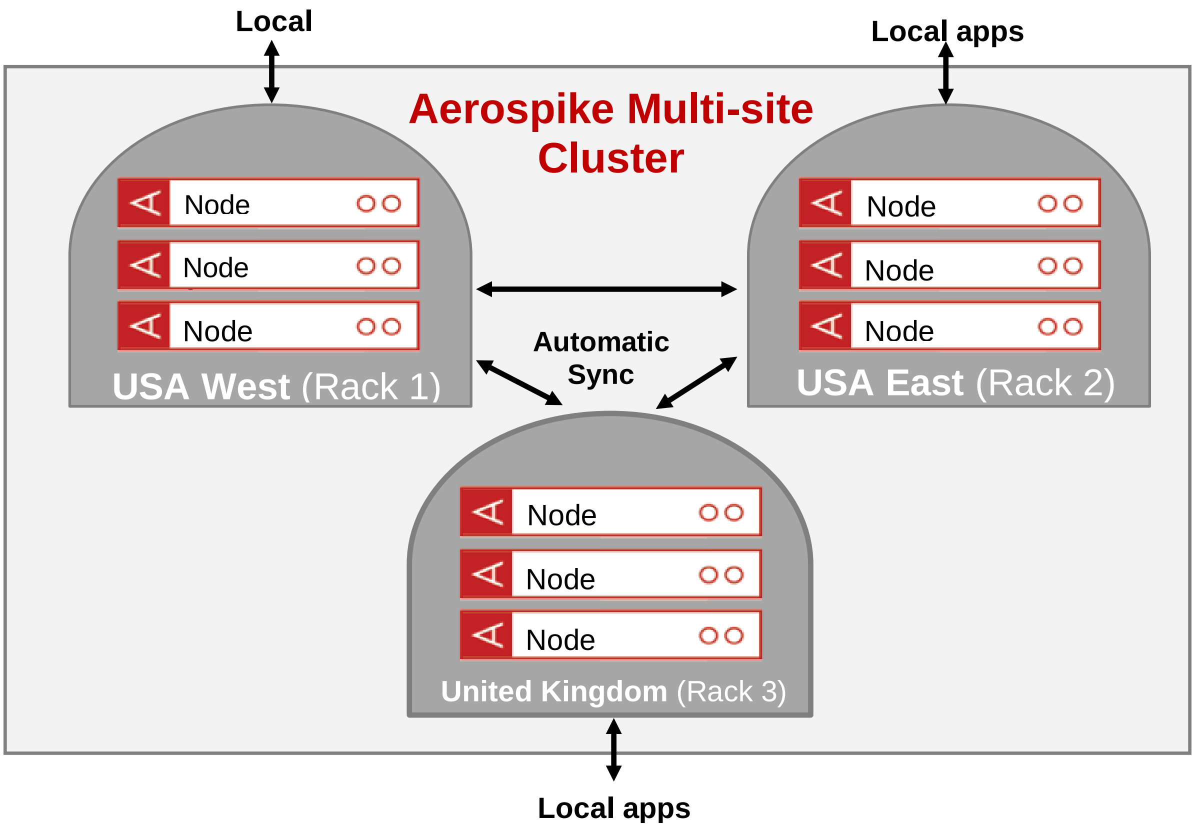

Figure 3: Sample Aerospike Cluster Deployed Across Three Data Centers

It's worth noting that an Aerospike multi-site cluster can be configured with as few as two data centers maintaining as few as two copies of user data (replication factor 2). However, a substantially higher degree of availability and failover automation is achieved with at least three data centers maintaining three copies of user data (replication factor 3).

Finally, Aerospike automatically maintains information about what comprises a healthy cluster and where data is stored; this enables Aerospike to detect -- and quickly overcome -- various types of failures, ranging from the loss of a single node at a data center to the loss of an entire data center.

With that background, we'll explore two core technologies that enable firms to deploy Aerospike across multiple regions. Then we'll explore various operational scenarios that illustrate how Aerospike works in such a configuration.

Core Technologies

Rack awareness and strong, immediate data consistency are critical capabilities that allow Aerospike clusters to be deployed across distant data centers or cloud regions. We'll review each in turn.

Rack Awareness

In a multi-site cluster, Aerospike's Rack Aware (RA) feature enables replicas of data records (grouped in data partitions) to be stored on different hardware failure groups (i.e., different racks). Through data replication factor settings, administrators can configure each rack to store a full copy of all data. Doing so maximizes data availability and local read performance. For example, in a three-region cluster that contains one rack each, a replication factor of 3 instructs Aerospike to maintain copies of all data in each rack. To prevent hot spots, Aerospike evenly distributes data among all nodes within each rack. As you'll soon learn, only one node in one rack of the cluster maintains a master copy of a given data partition at any time; other racks have nodes that store replicas of this partition. Aerospike automatically synchronizes the master copy with the replicas on different racks/nodes.

You may be wondering how Aerospike keeps track of where various master and replica data resides. This information, as well as a full list of the racks and nodes that comprise a healthy cluster, is kept in a roster that Aerospike automatically maintains. The roster is stored on every node of every rack of the Aerospike cluster. As you'll soon see, this roster plays an important role for processing write operations as well as managing various failure scenarios.

Fig. 4 provides a more detailed view of the sample cluster that we introduced earlier. Each data center has one rack with three nodes. Each node has a copy of the roster that Aerospike uses to track the state of a healthy cluster and the distribution of data, including where roster-master and replica data partitions reside. In this example, the roster-master copy of one data partition (shown in yellow) is on Node 3 of Rack 2 (USA East); replicas exist on Node 1 of Rack 1 and Node 2 of Rack 3.

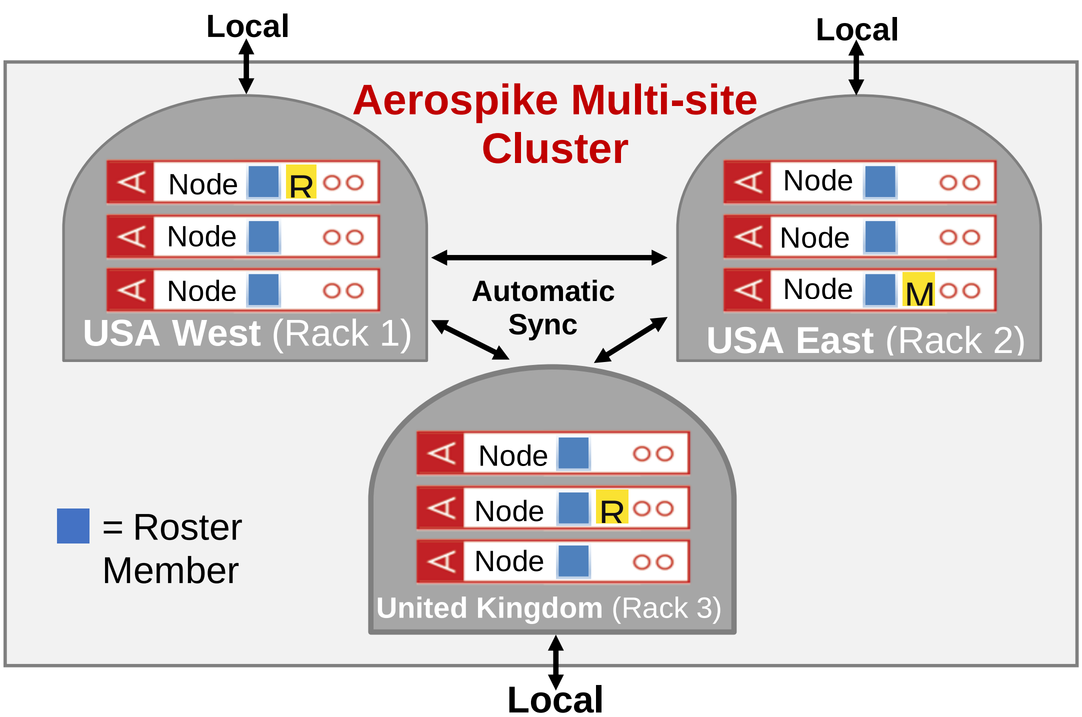

Figure 4: Aerospike Distributes Master & Replica Copies of Data Across Multiple Racks

Strong, Immediate Data Consistency

The application scenarios we discussed earlier demand a multi-region database platform that delivers absolute correctness of data. In particular, reads should only see the latest committed value (not stale or dirty/uncommitted data), and committed writes should never be lost. Aerospike's strong consistency (SC) mode fulfills these business demands.

A separate white paper describes Aerospike's SC approach in a single-region cluster, and this same approach applies to multi-site clusters. For that reason, we'll briefly summarize Aerospike's SC behavior here.

Aerospike uses its internal roster and the "heartbeats" of nodes grouped in racks to assess the cluster's current state. This information enables Aerospike to determine what operations are valid during various failure scenarios, such as a network failure that causes some portions of the system to be unable to communicate with others. (This is sometimes called a "split brain" scenario.)

To preserve appropriate ordering of events across the multi-site cluster, Aerospike employs a custom Lamport clock that combines timestamps with additional information, including a counter that increments when certain events occur. With Aerospike, each read or write operation is considered a separate transaction. Aerospike processes all writes for a given record sequentially so that writes won't be re-ordered or skipped. Independent tests of the Jepsen workload revealed no errors when operating with Aerospike's recommended configuration settings.

As you'll see shortly, Aerospike automatically takes corrective action to recover from various failure scenarios, including those in which the roster-master copy of a given data partition becomes unavailable. Whenever possible, Aerospike transparently transfers primary responsibility for managing write operations for that partition to another available data center containing a valid replica of the data so that application requests can still be processed. Of course, Aerospike does this in such a way that conflicting writes won't occur and committed writes won't be lost when the cluster is restored to a fully healthy state.

Operational Scenarios

It's arguably easiest to understand Aerospike's multi-site clustering capabilities by walking through some sample scenarios. We'll first cover read/write operations under normal conditions when a cluster is healthy. Then we'll explore what happens when different types of failures occur, such as the loss of a data center or the loss of network communications between one or more data centers.

It's worth noting that, unlike some platforms, Aerospike supports rolling upgrades with no disruption of service. For example, an administrator at a given data center can take a node offline for software upgrades and bring it back into the cluster when ready with no loss of data availability or consistency. This is a basic capability of Aerospike in both single- and multi-site configurations. Similarly, different data centers within a multi-site cluster can have nodes running different software versions during a rolling upgrade, avoiding the need to take the cluster offline to synchronize software levels on all nodes across the cluster.

Healthy Cluster

All Aerospike clusters feature a "smart client" software layer that maintains information in memory about how data is distributed in the cluster. This information maps data partitions to the racks/nodes managing them. Aerospike automatically and transparently routes an application's request to read a given data record to the appropriate rack/node in its local data center. Returning to the sample multi-site cluster shown in Fig. 4, an application in USA East seeking to read a record contained in the yellow data partition will access the master copy of the data in Rack 2 Node 3 of its local data center with only one network "hop." Another application in the UK seeking to access the same data record will also enjoy "one-hop" access, as Aerospike will retrieve the data from replica 2 stored in its local data center on Rack 3 Node 2. By intelligently -- and transparently -- processing read requests in this manner, Aerospike can deliver the same sub-millisecond read latencies to applications using its multi-site clusters as it does to applications using a single-region cluster.

Write transactions are handled somewhat differently. Like reads, writes can be initiated by any authorized application regardless of its location. But to ensure strong, immediate data consistency across the cluster, Aerospike routes each write to the rack/node that contains the current master of the data. The master node ensures that the write activity is reflected in its own copy as well as in all replica copies before the operation is committed.

Let's step through an example. Returning again to Fig. 4, a write request in USA East for the data shown in yellow will be routed to Rack 2 Node 3 in the local data center because it contains the master of the target record. A write request from USA West or the UK for this same record will be routed to Rack 2 Node 3 in USA East for the same reason -- that's where the master resides. As you might expect, this routing of write operations -- and the need to synchronize the effects of writes across all replica locations -- introduces some communication overhead. Simply put, writes won't be as fast as reads, but most firms using Aerospike's multi-site clusters are experiencing write latencies of a few hundred milliseconds or less, which is well within their target SLAs.

Failure Situations

Resiliency in the face of failure is a critical requirement of any multi-region operational database. Let's face it: natural disasters, power outages, hardware failures, and network failures can render one or more components of a multi-region cluster inaccessible. Fig. 5 illustrates examples involving a data center failure, failure of a single node within a rack of a data center, and a network failure between data centers.

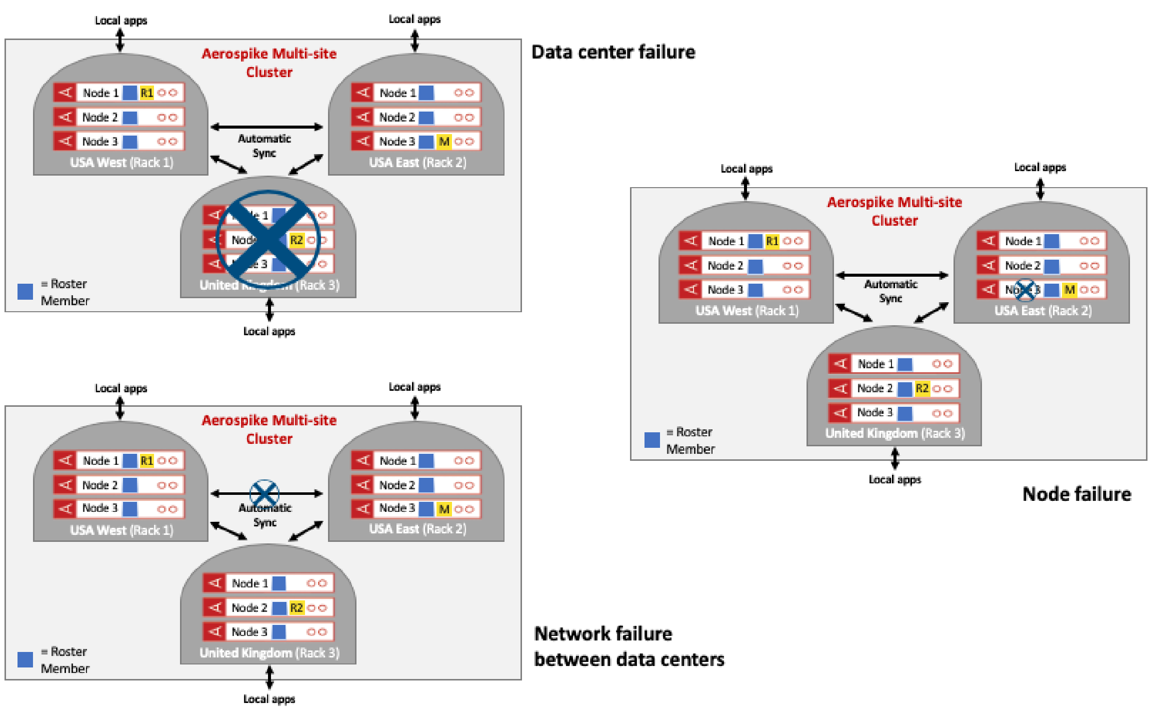

Figure 5: Three Sample Failure Scenarios

As mentioned earlier, Aerospike's internal roster and heartbeat mechanism enable it to detect when some portion of a multi-region cluster has failed or becomes inaccessible. Aerospike reacts immediately, usually forming a new sub-cluster within seconds to handle application requests. In many cases, this sub-cluster can seamlessly process all read/write operations even though a portion of the full cluster is unavailable. However, depending on the severity of the failure, there are cases when operations are restricted. Aerospike maintains the highest level of data availability possible without compromising data correctness.

General failover rules

Let's first review the general rules Aerospike follows to recover from failures and form new sub-clusters to process application requests. If the roster-master is unavailable, Aerospike designates a new master from the available replicas and creates new replicas so that each data partition has the required number of replicas as specified by the system's replication factor setting.

In a multi-site cluster, the new master will typically be on another rack. Furthermore, creation of new replicas is typically accomplished by copying data from another rack. This occurs in the background and does not impact availability. With high network bandwidth connectivity and pipelined data transfers, a replica can be created quickly.

When the cluster is split into multiple sub-clusters, only one sub-cluster can accept requests for a given partition to ensure strong consistency of data. Aerospike follows these rules to determine what the operational sub-cluster can do:

1. If a sub-cluster has both the master and all replicas for a partition, then the partition is available for both reads and writes in that sub-cluster.
2. If a sub-cluster has a strict majority of nodes and has either the master or a replica for the partition, the partition is available for both reads and writes in that sub-cluster.
3. If a sub-cluster has exactly half of the nodes and has the master, the partition is available for both reads and writes in that sub-cluster.

As you might imagine, in any failure scenario, it's important to have enough capacity among the remaining cluster nodes to hold an additional copy of the data. For example, a 3-rack cluster with a replication factor of 3 will still operate with 3 copies of the data even if the number of racks is reduced to 2 due to a failure of some sort. Aerospike will automatically -- and transparently -- undertake the necessary work to create and populate a new copy when needed and remove the temporary copy when the cluster becomes fully healthy again. This approach is the same for multi-site and single-site installations of Aerospike that employ strong consistency.

With that backdrop, let's return to the three sample failure scenarios shown in Fig. 4 and assess what happens from an application point of view. While other types of failure scenarios certainly are possible, stepping through these three scenarios should help you better understand Aerospike's approach to coping with failures.

Data center failure

Let's start with the first scenario shown in which the UK data center fails, shown in more detail in Fig. 6. As the figure depicts, the 3-region cluster was configured with a replication factor of 3.

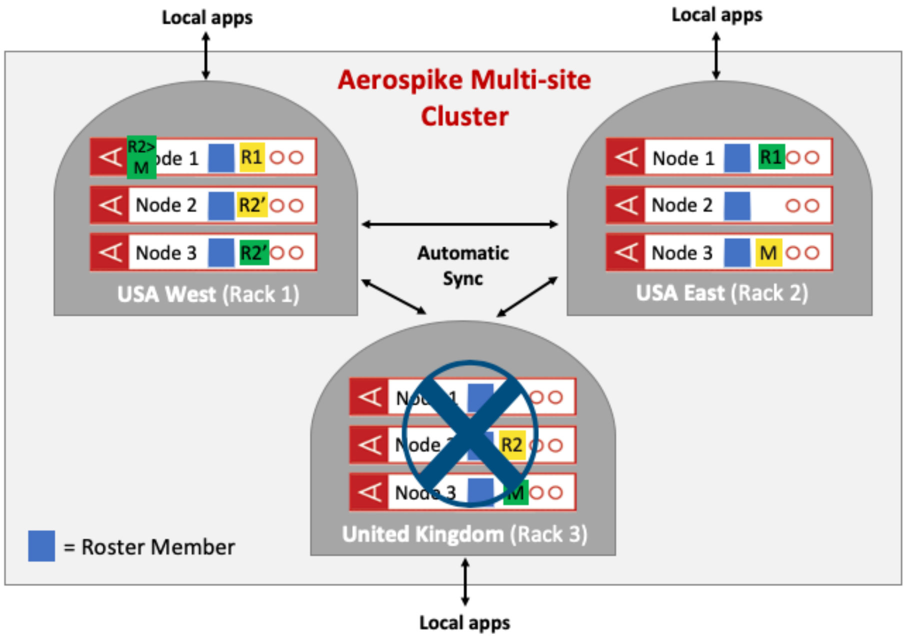

Figure 6: UK data center fails. Remaining centers provide full data availability.

Aerospike will automatically form a new sub-cluster spanning the two operational data centers (USA West and USA East). This sub-cluster will seamlessly service all application read/write requests, including those from UK-based applications (assuming they have connectivity to the surviving sub-cluster). Simply put, the data remains available and -- equally important -- it's guaranteed to remain consistent during the failure and after the recovery of the UK center.

How's that possible? Consider access to the data shown in yellow. The newly-formed sub-cluster contains the roster-master of the data and a majority of the racks/nodes that comprise the fully healthy cluster. Reads from applications in USA West and USA East will be handled locally, while read requests from applications in the UK will be automatically (and transparently) re-routed to one of the two available data centers, which each have a copy of the data. Writes initiated by any application will be routed to USA East, as it contains the roster-master. Consequently, USA East will be responsible for synchronizing the write across all available copies. When the UK data center undergoes recovery processing, it will ensure its replicas are current with all remote masters.

To maintain a replication factor of 3 for this new sub-cluster, Aerospike will create a new R2 replica in the background on a surviving node to temporarily replace the original R2 replica in the UK data center. In Fig. 6, R2' is shown in Rack 1 Node 2 of USA West. As you might expect, when the UK data center fully recovers, R2' will automatically be removed.

You might be wondering how Aerospike deals with roster-master data managed by the failed data center -- a situation depicted by the data in green in Fig. 6. Again, the sub-cluster USA West and USA East will service all read/write requests. Aerospike will select one of its replicas to be promoted to the master copy. Fig. 6 shows this occurring with replica 2, which is stored in Rack 1 Node 1 of USA West. Reads will be handled as described in the previous paragraph -- they will occur locally for applications connecting to USA West and USA East, while UK applications will read from one of these data centers. Writes from any application will be directed to USA West, which contains the current master of the data. Furthermore, since the original R2 was promoted to master, Aerospike will create another replica (R2') to preserve the replication factor in the surviving cluster. In this example, Rack 1 Node 3 in USA West was selected to host R2' for data shown in green.

When the UK data center undergoes recovery processing, it will ensure its replicas are current with all remote masters and the green R2' will automatically be removed. The original designations for each copy of data -- roster-master or replica -- will be re-established to promote data availability and load balancing across the full cluster.

It's worth considering what might happen if the UK data center wasn't offline but simply suffered from a network outage that rendered it inaccessible by USA West or USA East. Aerospike will handle this situation in the same way as just described. Why? The UK data center, though operational, will have only a minority of nodes for the cluster, while USA West and USA East will have a majority of nodes. As result, Aerospike will automatically and transparently direct all read/write requests to USA West and USA East, as described earlier. The UK data center will not process any requests until network communications are restored and the full cluster is healthy again.

Node failure

Next, let's turn to a less catastrophic failure scenario in which one node in one data center fails, as shown in Fig. 7. Aerospike handles this situation in much the same manner for single-site clusters and multi-site clusters. Basically, it forms a new sub-cluster without the failing node and allows all read/write processing to continue. Replicas on functioning nodes will be promoted to masters as needed to take over from the failing node. Assuming a replication factor of 3 in this scenario, Aerospike will create another replica in the background, prioritizing Nodes 1 and 2 of Rack 2 as the target nodes for the new replica so that Rack 2 will retain a full copy of the data.

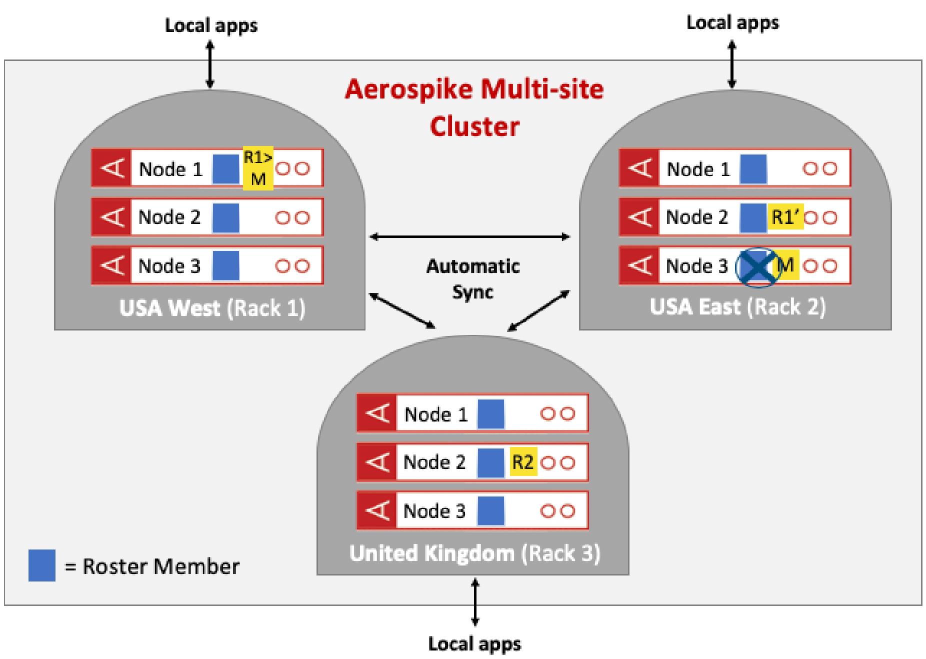

Figure 7: Node failure at USA East. Remaining nodes provide full data availability.

Reads will be processed locally. Writes will be redirected to the new master, which will synchronize the operation across the sub-cluster. Once the failed node is brought back online, Aerospike will redistribute and rebalance data as needed in the background, making sure that the recovered node contains current data and reclaims its roster-master and replica data partitions.

Network failure between data centers
Finally, let's consider a situation in which one data center is unable to communicate with another, perhaps due to a networking failure. Fig. 8 illustrates a case in which there is a network failure between USA West and USA East.

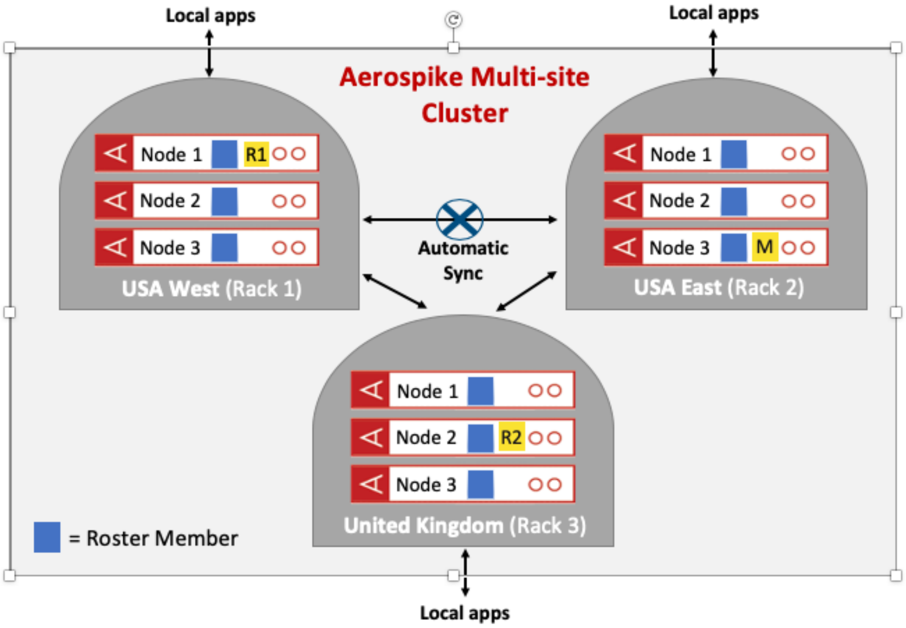

Figure 8: Network failure between USA West and USA East

This situation presents a "split brain" scenario in which the USA West and UK data centers can form one sub-cluster and the USA East and UK data centers can form another sub-cluster. Note that each sub-cluster contains a majority of nodes as defined in the roster. So what will Aerospike do? Using a deterministic algorithm, Aerospike will select one sub-cluster to remain active and render the other inactive. The active sub-cluster will process all read/write requests; the other sub-cluster will not process requests until recovery processing has occurred and the cluster is fully healthy again.

In this example, if Aerospike selects the USA East-UK sub-cluster to be active, reads for the yellow data will occur at USA East or the UK while writes for the yellow data will be routed to USA East, which contains the master. If Aerospike selects the USA West-UK sub-cluster to be active, one of the replicas of the yellow data will be promoted to master and a new replica will be created in one of the data centers to preserve the required replication factor of 3. Reads will occur at either USA West or the UK, while writes will be routed to the rack/node containing the newly appointed master.

Summary

The "always on" demands of a global economy, coupled with evolving requirements of modern transactional applications, are forcing firms to pursue new database infrastructures that can span multiple locations, deliver 24x7 availability, and maintain strong data consistency. Aerospike delivers a compelling, cost-effective solution. Simply put, Aerospike enables firms to deploy a single cluster across multiple geographies with high resiliency, automated failovers, and no loss of data. Early adopters in banking and other industries are already deploying sophisticated, mission-critical operational applications that rely on this technology.

If that sounds hard to believe, consider that firms in financial, technology, telecommunications, retail, manufacturing, and other industries have deployed single-region Aerospike clusters for more than a decade. Thanks to Aerospike's resource efficiency and high-performance design, Aerospike clients can cut their server footprints up to 90% and realize TCO savings of $1 to $10 million per application when compared to other alternatives.

No other vendor is so well-positioned to deliver a comprehensive and compelling solution for supporting inter-region clusters as Aerospike. Why not explore how you might benefit from a highly resilient and fully consistent multi-region database platform? Contact Aerospike to arrange a technical briefing or discuss potential pilot projects.

Resources

Exploring data consistency in Aerospike Enterprise Edition, Aerospike white paper, 2018.

Jespon test report for Aerospike 3.x, Kyle Kingsbury, Jepsen.io, March 2018.

Maximize the value of your operational data, Aerospike white paper, 2018.

What is TARGET Instant Payment Settlement (TIPS)?, European Central Bank web site.

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, 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.

© 2020 Aerospike, Inc. All rights reserved. Aerospike and the Aerospike logo are trademarks or registered trademarks of Aerospike. All other names and trademarks are for identification purposes and are the property of their respective owners.