AEROSPIKE

Debunking the free open source myth

CASE STUDY

How Signal replaced open source Cassandra with Aerospike® for superior TCO and operations

Benefits with Aerospike

- TCO reduction of 68% over three years
- Server count reduced from 450 to 60
- Performance improved 100x at the 99th percentile
- Business processes executing in 1/10th the time – or better
- Time freed up to focus on more strategic, forward-looking projects

About Signal

Signal (www.signal.co) is the leading SaaS provider of data onboarding and real-time identity resolution. Signal's technology platform empowers people-based marketers to enhance their customer experience by providing relevance across all channels, regardless of place or time.

Challenges with Cassandra open source

Signal, being an identity resolution platform, was looking to replace its existing data store, which was becoming increasingly expensive, unreliable, and nonperforming – affecting the bottom line. One of the biggest problems they were running into was large and unpredictable latency response as well as uptime, both of which were affecting every element of their business processes. They were experiencing more frequent and more severe issues and incidents, all related to an unreliable data store.

Challenges with scale and growth

The Signal Customer Identity Platform (CIP) was built on Cassandra, an open source, no-cost license NoSQL database. However, their Cassandra footprint had grown to more than 550 servers. The company was faced with a classic case of "server sprawl" as they continued to grow. The Cassandra clusters were difficult to maintain and, worse, were proving to be high touch for their Ops teams, taking resources away from higher-value projects. In addition, performance and uptime became unpredictable, which negatively impacted the company's SLAs to its customers.

The team knew it needed to get on top of its server growth, and they were able to identify and deliver improvements to reduce the footprint from 550 to 450 nodes. But with expected 25% annual data growth (common in the industry), they knew they needed a new solution.

Goals

Signal was looking for architectural alternatives to better support their Customer Identity Platform (CIP) solution. Elements of the Signal CIP solution in need of re-architecting included:

- Continuous identification or real-time recognition, identification, and updates of customer activity both online and offline.
- Customer data foundation or customer profiles containing full brand interactions and history.
- Always active profiles or continuous real-time updates of customer profiles.
- Activation connections that activate marketing to customers via connections to marketing partners, digital advertising platforms, personalization engines, attribution analytics, and insights tools.

Signal's operational requirements

- Significantly reduce operational footprint
- Reduction of infrastructure spend
- Regional architecture of five clusters
- Support real-time cross-datacenter replication across regions
- Improve p99 SLAs to less than 10ms reads and less than 1s writes

> ***"Before Aerospike, we were spending more and more of our time on the care and feeding of Cassandra, and less and less time on the building of new product offerings. With Aerospike, we've now cleared the roadmap and we're just focused on adding new functionality to our platform for our customers."***
>
> *Jason Yanowitz*
> *EVP, Chief Technology Officer - Signal*

Why Aerospike

Compared to the other solutions that were being evaluated, the main drivers that made Aerospike so attractive to Signal was its low total cost of ownership, high performance at scale, and ease of scaling overall.

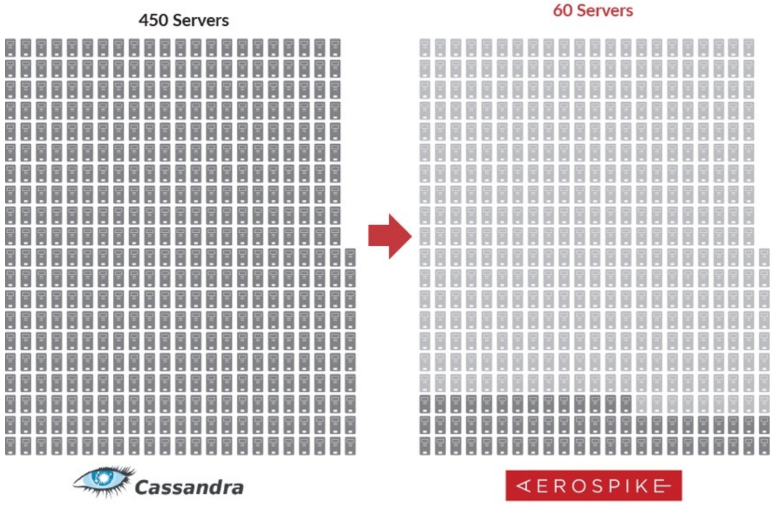
Upfront during the presales process, the Aerospike engineering team was highly engaged with Signal, demonstrating a deep understanding of the company's problems. This gave Signal a tremendous level of confidence with Aerospike.

Benefits with Aerospike

During the initial deployment with Aerospike, Signal pushed the limit up to 8 million transactions per second (TPS) and saw the p50 (middle-performance estimate) at 10 microseconds. It proved to be absolutely stunning to Signal - almost a thousand times faster than what they were seeing before.

When Signal switched completely to using Aerospike, they saw immediate improvements on a number of axes:

- **TCO reduction of 68%.** While meeting all of the operational requirements, Aerospike was able to reduce Signal's total cost of ownership (TCO) by 68% over three years, saving them millions of dollars. These savings stem from the reduction in the number of servers. Aerospike was able to slash the node count from 450 with Cassandra to just 60 with Aerospike. As a result, Signal also benefited from the reduction in time needed to support fewer servers, which in turn freed up engineering cycles.



- **Ability to focus on strategic initiatives.** According to their CTO, "Before Aerospike, Signal was spending more and more of their time on the care and feeding of Cassandra, and less and less time on the building of new product offerings. With Aerospike, Signal has now cleared our roadmap and is focusing on adding new functionality to our platform for our customers. Across the spectrum, it's been much better for Signal with Aerospike."
- **100x performance improvement**. The other notable area of improvement was performance. Signal's p99s (99th percentile of performance) plunged from 3,900 milliseconds to 23 milliseconds, more than a 100x improvement.
- **Data reliability.** Whatever data the company was putting into Aerospike was easily retrievable - which was not the case with the prior solution.
- **Gain data portability with ongoing analysis**. Due to the high performance with Aerospike, even while Signal is live and taking active traffic, they're now able to completely take their dataset and place it into a data warehouse. This ability to conduct ongoing analysis with ease while systems are online and running is a big improvement over what had been a difficult and time-consuming process with Cassandra for Signal.

> ***"Compared to the other solutions that we were evaluating, the main drivers that made Aerospike so attractive was its total cost of ownership performance and scale were all superior compared to any of the competitive offerings evaluated."***
>
> *Jason Yanowitz*
> *EVP, Chief Technology Officer - Signal*

- **Accelerate key business processes.** Signal also found Aerospike to be incredibly helpful in previously unforeseen ways. Signal's large-scale business processes have gotten much faster. Processes that used to take six days now take only 14 hours – a 10x improvement. Processes that took three hours now only require three minutes – a 60x improvement.

Total cost of ownership: Aerospike vs. Open Source Cassandra

As Signal considered updating its open source Cassandra infrastructure for its Customer Identity Platform (CIP), a thorough examination of hardware, software and management costs was undertaken to compare it to what Aerospike had to offer.

Aerospike Hybrid-Memory Architecture™

The backbone of gaining the TCO advantage

Aerospike was able to offer Signal both superior performance and total cost of operations savings over open source Cassandra due to the Aerospike Hybrid-Memory Architecture™ (HMA). Because Aerospike achieves its high performance with only indexes in memory while persisting data on SSD, it requires a much, much lower server footprint than Cassandra. As a result, Signal saw its engineers both freed up for other tasks and significantly cutting their on-call fatigue.

With the Aerospike HMA, performance is also more predictable: Aerospike always reads the data in the same, highly repeatable manner. (Aerospike has nearly a dozen patents optimizing how it is able to read data from disk as fast as or faster than Cassandra

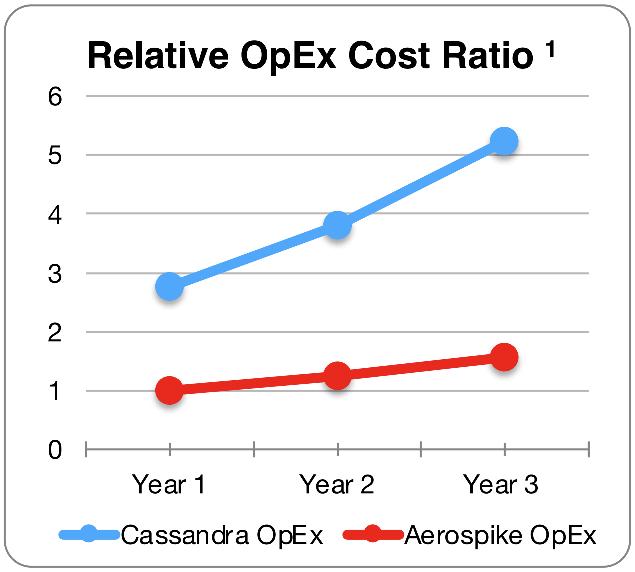


Figure 1: Relative OpEx Cost Ratio 1
Based on 1 year up-front AWS pricing for each of three years (Signal's preference).

[1] Relative OpEx Cost Ratio is defined as the ratio of actual hard OpEx $ Spent (AWS instance costs plus Dev Ops costs) comparing Aerospike spend vs. the required Cassandra spend for the same exact use case.
(See Table 1 for further explanations.)

Idoes from DRAM). The Aerospike Hybrid-Memory Architecture also has node awareness features, where each node knows what data all the other nodes contain. As a result, there is no time wasted searching for data. In addition, the performance for Signal was enhanced by placing data locally with copies at each of five geographically dispersed datacenters, each kept in synch via the Aerospike Cross-Datacenter Replication (XDR) feature.

In addition, Signal was forecasting a significant 25% data growth year-over-year for the considered three-year period. Aerospike's total cost of ownership advantage in year one only increases in years two and three.

In Figure 1 (above) and Table 1 (below), we can see clearly that the OpEx for Cassandra starts much higher than that of Aerospike for the first year and is at a steeper rate of increase for the second and third years.

	Year 1	Year 2	Year 3	Total
Cassandra Relative OpEx Cost Ratio[1]	2.8	3.8	5.2	**11.8**
Aerospike Relative OpEx Cost Ratio[1]	1.0	1.3	1.6	**3.8**
OpEx saved with Aerospike	**64%**	**67%**	**70%**	**68%**

Table 1 – Relative OpEx Cost Ratio Savings *- 1 year up-front AWS pricing for each of three years (Signal's preference).*

[1] Relative OpEx Cost Ratio is defined as the ratio of actual hard OpEx $ Spent (AWS instance costs plus Dev Ops costs) comparing Aerospike spend vs the required Cassandra spend for the same exact use case.
Example: In Year 1, for every $1 spent on Aerospike OpEx, it will require $2.8 spent on Cassandra OpEx for the same exact use case. In Year 3, it will require a spend of $1.6 on Aerospike OpEx vs a $5.2 on Cassandra OpEx. This means the cost basis of Aerospike will grow from $1 to $1.6 over 3 years - a total of 60%. However, the cost of the Cassandra infrastructure will grow from $2.8 to $5.2 for a total of an 185% increase over the same 3 years! In aggregate, Aerospike will reduce the Cassandra OPEX costs by 68% over the 3 years.

TCO comes down to server footprint and support headcount

The Open Source myth debunked for Signal

The allure of open source is the zero-license cost, fueling the myth that it must be less expensive overall. However, large resultant server footprints (a.k.a. "server sprawl"), as well as the costs to support it (not to mention open source performance issues and care and feeding needed at scale), all contribute to it being cost ineffective. (Note: costs of power and cooling are not factored in this TCO analysis but would, in general, favor lower server counts.)

Table 2 (below) shows the elements of debunking the open source myth. Note the significant differential in the number of servers forecasted, for example, in year three of 853 for Cassandra versus 94 for Aerospike (see Appendix for sizing calculations). While Aerospike utilizes more powerful servers (with correspondingly higher per-server costs – not shown), the overall resultant infrastructure cost is dwarfed by the magnitude of servers needed for Cassandra.

	Cassandra[1]			Aerospike[2]		
	Year 1	Year2	Year 3	Year 1	Year 2	Year 3
Cluster Size	450	619	853	12	15	19
Total Servers	450	619	853	60	75	94

Table 2: Cluster Size Comparison

With lower hardware server counts comes lower maintenance and support, all contributing to a 68% OpEx savings totaling several million dollars over three years for Signal. Once the number of nodes and the instance types are determined, it becomes relatively straightforward to calculate the total operational cost and infrastructure cost for each solution. Plus, the larger number of nodes for Cassandra will cost more in DBA overhead1. Furthermore, Aerospike was able to replicate all data to each cluster within each of the five datacenters via its Cross-Datacenter Replication (XDR) feature – a requirement of the project. As a result, data is more proximate and responsive to the needs of Signal's business. Even with this factored in, Aerospike was able to save Signal millions of dollars in total operational expenditures and again debunk the open source myth that free licenses result in lower overall costs.

Table 2 Notes:

1. Cassandra servers: m4.2xlarge
2. Aerospike servers: i3.8xlarge

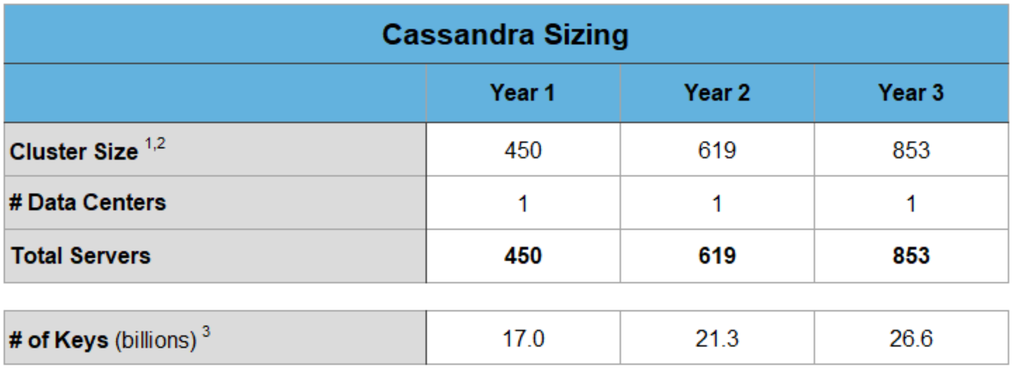
[1] Source: https://www.c-sharpcorner.com/article/it-services-cost-cheat-sheet-part-one/

Appendix

Sizing details and calculations

Cassandra sizing

With their CTO's latest round of cost reductions, Signal, for their CIP, was running 450 AWS M4.2xlarge nodes, running open source Cassandra for their 17 billion keys. With 25% data growth, their server counts projected out to 619 for year two and 853 in year three. (See Table 3 below.)



Cassandra Sizing			
	Year 1	**Year 2**	**Year 3**
Cluster Size [1,2]	450	619	853
# Data Centers	1	1	1
Total Servers	**450**	**619**	**853**
# of Keys (billions) [3]	17.0	21.3	26.6

Table 3: Cassandra Sizing calculations

With input from the Signal CTO and his team's operational experience with Cassandra, they noticed a "penalty" of an extra 10% inefficiency when scaling nodes on top of the 25% data growth (represented by number of keys).

One area of cost and reliability difference between the two architectures was that Aerospike's spanning of datacenters is much more robust. With Cassandra's ring architecture, it is more costly and complicated. For Aerospike, given the small cluster size and the Aerospike Cross-Datacenter Replication (XDR) feature, having a synchronized copy of the data for each of the five datacenters was seen as a great way to put data closer to the need for better performance. Plus, "It's awesome," said Signal's CTO Jason Yanowitz, "Signal could continue to run even if all datacenters running Aerospike save one went down. In other words, a single Aerospike cluster would still be able to soak up all the load." (Note: while this did increase the Aerospike number of servers five-fold, it still had 7x fewer server nodes than the Cassandra solution.)

> ***"It's awesome... Signal could continue to run even if all datacenters running Aerospike save one went down. In other words, a single Aerospike cluster would still be able to soak up all the load."***
>
> *Jason Yanowitz*
> *EVP, Chief Technology Officer - Signal*

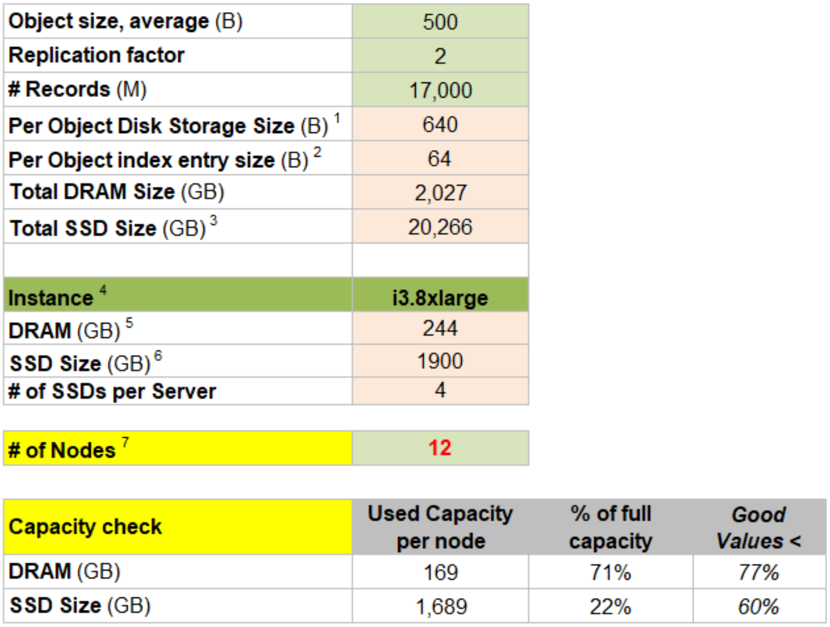
Table 3 Notes:

1. Cluster Size based on growth in # of keys, plus 10%, observed "penalty" from Signal CTO
2. Amazon EC2 m4.2xlarge
3. Year 2, Year 3 have 25% YoY growth factor included

Aerospike Sizing – Year 1

Dramatically fewer servers than Cassandra with data on SSD and only indexes in-memory

Sizing the initial cluster required examining the data footprint for persistent memory on SSD and indexes on DRAM. Based on experience, there is a class of server Aerospike tends to work with and a recommended percent of capacity (i.e. "good values"). As a result, Aerospike would require only 12 servers for one cluster. (See Table 4 below.)



Object size, average (B)	500
Replication factor	2
# Records (M)	17,000
Per Object Disk Storage Size (B) [1]	640
Per Object index entry size (B) [2]	64
Total DRAM Size (GB)	2,027
Total SSD Size (GB) [3]	20,266
Instance [4]	**i3.8xlarge**
DRAM (GB) [5]	244
SSD Size (GB) [6]	1900
# of SSDs per Server	4

# of Nodes [7]	**12**

Capacity check	**Used Capacity per node**	**% of full capacity**	***Good Values <***
DRAM (GB)	169	71%	*77%*
SSD Size (GB)	1,689	22%	*60%*

Table 4: Aerospike Sizing

17B objects, 64 b index entry size, 640 b object size (effective; 500 b actual); i3.8xlarge instances; 12 nodes required with data on SSD and indexes in DRAM.

Per Table 4, for 17 billion records, Aerospike had a per-object average disk size of 640 bytes, a per-object key DRAM size of 64 bytes, and a replication factor (RF) of two. Multiplying these out results in 20TB of Total SSD.

Similarly, per Table 4, multiplying out the number of records, replication factor, and per-object index entry size resulted in a total DRAM size required of just more than 2TB.

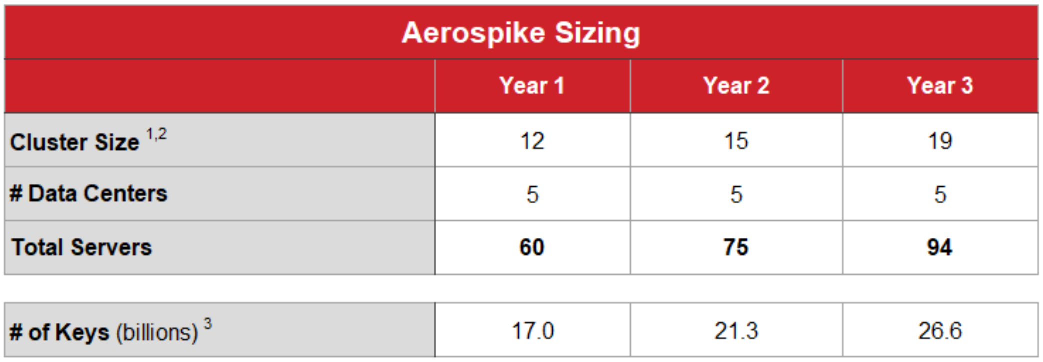
Given the index and data size, the next step in sizing for Aerospike is selecting an Amazon instance. Aerospike has guidelines of DRAM and SSD utilization to help assess the number of nodes. Among Amazon instances in the i3 family, i3.8xlarge yielded a low cost for the Signal configuration. AWS i3.8xlarge instances have four 1.9TB SSDs and 244 GB DRAM. For this number of objects, object size, and AWS instance type, it turned out DRAM was the limiting factor. In other words, there was more "headroom" for the percent utilization on SSDs than for DRAM. The result: 12 nodes with Aerospike were required, which is in the range of common deployment cluster size for Aerospike customers.

Table 4 Notes:

1. Includes storage overheads
2. Standard for Aerospike
3. Total amount of data to be stored on SSD across the cluster (distributed)
4. Selected as a powerful server with good storage (common for Aerospike use cases)
5. 244 is raw DRAM storage. Effective DRAM for Aerospike is 238 GB or 97.5%
6. Flash/SSD storage space assumes for data only & not indexes (indexes will be stored in RAM).
7. Given instance characteristics, lowest number that meets "Good Values < " criteria

Aerospike Sizing – Years 2 and 3 for Scale-out

Signal indicated their data growth rate would be 25% per year for the second and third years. Starting with 17 billion keys, this works out to be 21.3 billion for year two and 26.6 billion keys for year three. This translated into Aerospike requiring 75 servers in the second year and 94 in the third year. (See Table 5 and notes below for calculations.)



Aerospike Sizing			
	Year 1	Year 2	Year 3
Cluster Size [1,2]	12	15	19
# Data Centers	5	5	5
Total Servers	**60**	**75**	**94**
# of Keys (billions) [3]	17.0	21.3	26.6

Table 5: Aerospike Sizing calculations

Table 5 Notes:

1. Cluster Size based on growth in # of keys
2. AWS i3.8xlarge instance
3. Year 2, Year 3 have 25% YoY growth factor included

Aerospike unleashes the power of real-time data to meet the demands of The Right Now Economy. Global innovators and builders choose the Aerospike real-time, multi-model, NoSQL operational database and data platform for its predictable sub-millisecond performance at unlimited scale with dramatically reduced infrastructure costs.