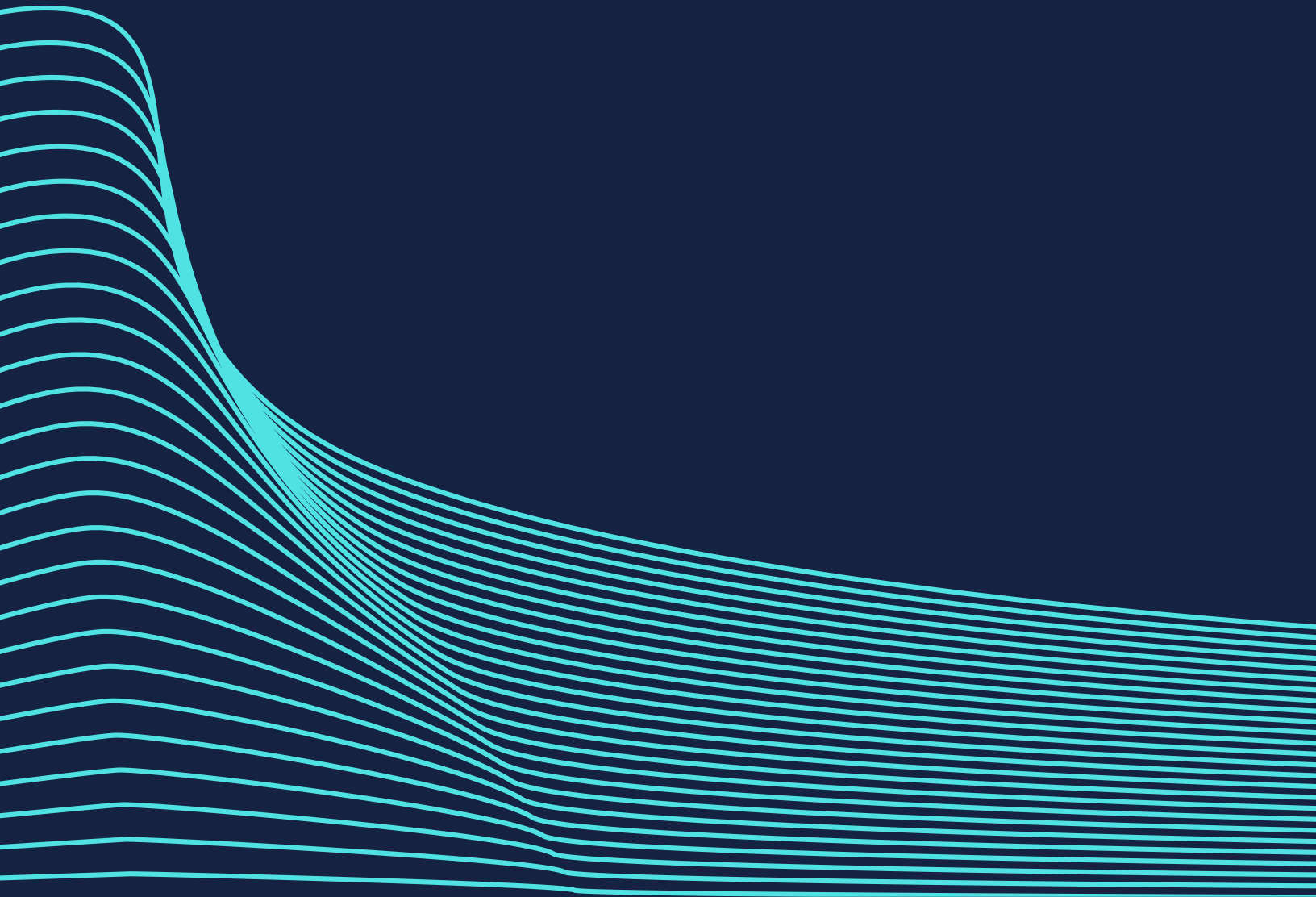


azul

***Consistent Cassandra
Performance with Azul
Platform Prime***





Azul Platform Prime allows Cassandra to deliver more throughput with fewer pauses to reduce infrastructure overheads by up to 50%.

Executive Summary

Apache Cassandra® is a NoSQL distributed data base written in Java and designed for reliability and high scalability. Data nodes are spread across lots of commodity servers and even between datacenters. Information is kept in-memory for fast read and write response times, then “flushed” to disk when memory fills.

Because Cassandra is built in Java, it is subject to the limitations and performance hiccups associated with traditional Java Virtual Machines (JVMs). Common issues include response time outliers, frequent Java garbage collections, out of memory errors and system hangs. Companies deploying Cassandra spend extensive time tuning and tweaking the JVM to avoid these issues, then need to re-tune each time load (or the application accessing Cassandra) changes.

The Azul Platform Prime runtime allows Cassandra to run smoothly, free of Java-related performance glitches and hangs. Platform Prime is an innovative JVM that removes garbage collection as an issue, delivering consistent response times all the time, regardless of load or object allocation rates. Azul Prime also provides runtime optimizations that improve code performance and faster warm up.

The upshot is that through more efficient Cassandra performance on each node means smaller, more efficient Cassandra clusters, on bare-metal and in the cloud, **delivering millions of dollars in cost savings** in some cases.

A Simplified Overview of Apache Cassandra

Cassandra is a NoSQL distributed database. It provides a reliable highly scalable data store and has no single point of failure. Every Cassandra node is a peer to the others. Deployments can consist of any number of nodes across many commodity servers, across datacenters, and in clouds. New nodes can be added easily on-the-fly.

Cassandra is row oriented. Each row has a unique key and can have different columns than the rows around it. New data is written to a log, then to a Memtable in-memory. Once the Memtable is full the contents are flushed to disk. Each row is kept on multiple nodes, so if a node is no longer reachable the data is not lost.

Cassandra is a good choice for enormous amounts of data where high write and read throughput is required. Users include Netflix for movie streaming, Walmart for updating catalogs and managing analytics, and Uber for managing driver availability.

“We had this 15-minute cycle where the server would just stop responding for 15 full minutes and would do the garbage collection, which is just the way Java works. We tried different approaches to performance-tune it: we tried to reduce our heap size, to increase our heap size, to change our cache strategies.”

Ariel Pizetsky, VP of IT, Taboola

Introducing Azul Platform Prime

Azul Platform Prime includes an innovative JVM that is fully compatible with the Java SE specification. Azul Platform Prime is optimized for Linux server deployments and designed for enterprise applications and workloads that require any combination of large

memory, high transaction rates, low latency, consistent response times or high sustained throughput to deliver improved response times all the time and infrastructure cost savings.

Now Java apps like Cassandra can scale to high levels of performance with unprecedented response time consistency and meet the most demanding SLAs - without application changes or even recompilation. Azul Prime allows Cassandra to run smoothly while handling far more data and delivering more throughput than traditional JVMs. Azul Prime is proven to open up new business capabilities and opportunities not practical with other Java software platforms.

How Azul Platform Prime Improves Cassandra Performance.

Because Cassandra is written in Java, certain elements are very sensitive to garbage collection issues. If you're not familiar with garbage collection, the basic idea is that when memory allocated to the JVM fills up, garbage collection is triggered to free up space for more objects (www.azul.com/products/components/pgc/). A "minor collection" usually takes little time. However, in at traditional JVM, the system will eventually need to "stop-the-world" or pause the operation of the Cassandra node to perform a "full collection" that compacts the memory heap to free space.

"Millions of dollars in savings."

"With Platform Prime, we were able to reduce our front-end server footprint by more than 30%, which is hundreds of servers, and our database server footprint by about 50%. This translates into millions of dollars in savings, both hardware we freed up and hosting costs avoided."

Ariel Pizetsky, VP of IT, Taboola

Azul Platform Prime's innovative garbage collection algorithm solves many common Cassandra performance issues:

Insensitive to Memtable thresholds and sizes.

Memtables store updates in memory until they can be

pushed to disk. Data kept in a Memtable usually ends up in the "old generation" in the Java heap. Lots of other information is stored in the heap as well as the Memtable, so the Memtable size should be set low enough (the default is 1/3 of the overall heap size) so that it doesn't squeeze out row caches and other important data. Companies often spend a lot of time tuning Memtable sizes to avoid full garbage collection. Platform Prime allows you to avoid tuning and get your deployment launched faster.

Platform Prime also provides flexible memory allocation to the JVM, so if the heap is filling fast, more memory is made available to avoid an out of memory error or a slowdown in processing.



Reduce out of memory errors.

The default JVM setting in Cassandra triggers a major collection when the heap is around 92 - 93% full. Because this threshold is so high, sometimes the heap fills before garbage collection can be triggered, causing an out of memory error.

Platform Prime eliminates this issue by continuously and concurrently garbage collecting (that is, without stopping the node) and by providing 'elastic memory', so more memory is automatically made available to the JVM in real time.

Solve memory fragmentation.

Attempts to tune away long garbage collection pauses on Cassandra nodes can lead to insufficient available space for new objects. As you tune away full collection to avoid response time issues, node memory can become too fragmented - the spaces between existing objects may be too small to accommodate the next

stored object, causing performance degradation over time and eventual out of memory errors. A full garbage collection with heap compaction, which in a traditional JVM stops the processing of the node, is required to free up this space.

Azul Platform Prime includes the **only JVM** that can continuously and concurrently compact. It constantly reclaims memory fragments to make larger spaces available for new objects and never falls back to “stop-the-world” to compact the heap.



Prevent long GC on a node from bringing down the cluster.

In Cassandra every row is stored on multiple nodes and every write is sent to all nodes that contain the row. If one node fails, it's ok; all read and write requests can still be handled. However, every node has a coordinator queue that will wait for all requests to finish - even if this is after the response has been given to the client. What this means is that a very slow node (for example, one experiencing a full GC) can cause the queue to fill up and make that node unable to respond to any requests. This can lead to the entire cluster becoming unresponsive.

Platform Prime eliminates GC pauses on the nodes, preventing GC-caused cluster failures.

Prevent GC thrashing. If the JVM heap size is set too small or if residency in the heap is over 50%, Cassandra can spend most of its time doing garbage collection. Of course, the time needed to do garbage collection is directly proportional to the size of the heap. If you're tuning to try to make full garbage collection pause time slow, you can end up causing the system to garbage collect all the time.

Platform Prime allows you to use a large heap without garbage collection pauses. It removes the trade-off between holding lots of information in memory and keeping response times low. Consistent Cassandra performance is guaranteed with Azul Platform Prime.

"We use Apache Cassandra, which recommends nodes with only a terabyte of data; but we're pushing nodes with 10 TBs of data densities and still achieving the performance we need."

Ariel Pizetsky, VP of IT, Taboola

Support row caches. The row cache in Cassandra allows information that has been moved to disk to be cached in memory for faster read response times. The memory consumption of a row cache is high, and with a traditional JVM an efficient row cache will trigger lots of full garbage collections that pause operation of the node. Platform Prime is designed for systems that benefit from large amounts of in-memory data. Platform Prime allows the row cache to use lots of memory without any garbage collection penalties.

Mitigating compaction pauses due to tombstones. When data is deleted from disk, Cassandra creates a 'tombstone' to indicate its new status. The column continues to exist for a short period of time before compaction permanently deletes it. Cassandra compaction of these files can trigger a full Java garbage collection.

Platform Prime never falls back to a “stop-the-world” full garbage collection. It constantly compacts the heap while the application is running and does not cause a stall due to the removal of tombstones in Cassandra.

Cassandra Performance with Platform Prime - Online Advertising

Online advertising platforms are very sensitive to response time issues. Ads are placed in real time bidding. When an ad request is received, the system has just a few milliseconds to respond with a targeted ad or another ad service may get the placement.

One online advertising company using Cassandra contacted Azul about improving response times and throughput on their cluster. In testing:

	Other JVM	Platform Prime
Throughput: requests/sec	1,000	1,300
Request response time	Baseline	24 - 30% faster

Platform Prime also removed response time outliers and allowed the company to make better use of available server CPU and memory.

Platform Prime, Cassandra and Feedzai

Feedzai is a well-established, innovative data science and software company that uses Cassandra to power real-time, machine-based learning to help payment providers, banks and retailers prevent fraud.

"Azul powers the largest banks in the world and with peak load demands of up to 50,000 transactions per second, Azul Platform Prime will help ensure that we can deliver the best that artificially intelligent machines can offer."

Nuno Sebastiao, Chief Executive Officer, Feedzai

Find out more



To learn more about Azul Platform Prime or to request a free trial copy to try with your Cassandra deployment, contact Azul today.

Learn more about Azul Platform Prime:

<http://www.azul.com/products/prime>

Request a free trial copy of Azul Prime here:

<https://www.azul.com/products/prime/trial-download/>

Contact Azul

385 Moffett Park Drive, Suite 115
 Sunnyvale, CA 94089 USA
 ☎ +1.650.230.6500
www.azul.com

© 2021 Azul Systems, Inc 4-21v1