USE CASES

Fpr Apache Cassandra®

With its impressive capabilities and evolving technology, there is a good reason why Apache Cassandra® continues to be a popular database choice for companies of all sizes. In this white paper, discover its common use cases and how to best avoid any pitfalls.

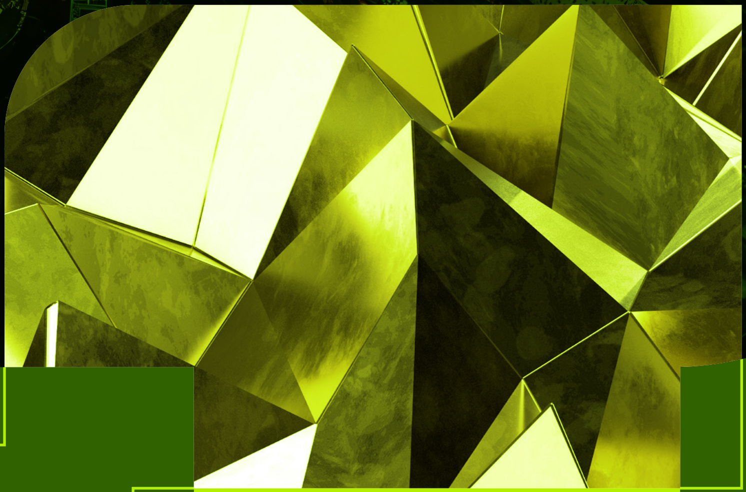

Table of contents

03 Introduction
04 Common use cases for Cassandra
06 Deciding if Cassandra is right for you
08 Why organizations choose Cassandra
09 Teaming with a technology partner
10 A final word

INTRODUCTION

Organizations today face explosive growth in data from various sources: IoT devices, user clickstreams when visiting websites, and logs from app and security management tools, just to name a few. Naturally, there is a great desire to analyze that data quickly, and over the long term to find patterns, spot anomalies, and more. These conditions are forcing many organizations to examine their database strategy as entirely in-house approaches will simply no longer do.

A look at some stats on data growth and analytics/AI adoption helps frame what has changed and now requires new thinking about a database strategy. International Data Corporation (IDC) forecasts that there will be 41.6 billion IoT devices in 2025 capable of generating 79.4 zettabytes (ZB) of data. IoT is a subset of the much larger edge computing transition that is occurring; its adoption, too, is skyrocketing, with the market forecasted to reach $168 billion by 2030 with a CAGR of 24.51%.

Similarly, large growth is expected in the demand for solutions that will help organizations make sense of such data and allow them to derive actionable insights. According to one market research firm, the global AI solution market will reach $301.2 billion by 2028, growing at 29.4% CAGR.

As such, organizations need a way to efficiently work with the large volumes of data that are constantly being created and have the capability for individuals or groups to analyze that data in multiple ways. These criteria are leading many organizations to adopt Cassandra, an open source NoSQL distributed database.

Originally developed by Facebook, but later made an open source project under the Apache Software Foundation, Cassandra is the world's leading NoSQL distributed database. Known for being a decentralized and scalable storage system, Cassandra is designed to handle incredible volumes of data across multiple commodity servers, providing high availability without a single point of failure.

Some inherent Cassandra properties make it a great choice to meet the demands that organizations require of their modern databases. For example, Cassandra has a shared nothing architecture, meaning that no single point of failure exists as all nodes are the same. Its architecture makes it highly scalable, in that read and write throughput increases linearly as new machines are added with no downtime or interruption to applications. That makes it ideal for handling heavy write workloads commonly found in applications today that must work with data from many devices.

The bottom line: Apache Cassandra helps organizations solve complex data problems including non-linear scaling issues, low writer performance, and resource constraints.

COMMON USE CASES *For Cassandra*

Organizations today use Cassandra when they need to analyze data and derive insights to reduce operating costs, quickly react to new business opportunities, improve customer engagement, detect and prevent fraud, minimize downtime and outages, and more.

Applications where Cassandra excels have a common set of characteristics. These include applications where writes exceed reads by a large margin, data can be partitioned to spread the database evenly across multiple nodes, and there is no need for joins or aggregates.

Here are some of the most popular and dynamic use cases where Apache Cassandra provides immense value:

Financial services for fraud detection and improved customer experience

Financial institutions use Cassandra to support real-time analysis of large and diverse datasets such as transaction histories, real-time transactions and events, and more to look for patterns and anomalies that are indicators of fraud. Interestingly, the same type of data analysis and investigation is being used to authenticate legitimate customers. Once a true customer is verified that knowledge can be used to simplify the log-in process and personalize services.

Retail personalization and product availability

Retailers use Cassandra's extremely low latency, fast response times, and ability to handle all types of data from various sources to customize user experiences, leading to the creation of a highly personalized recommendation engine. Cassandra's peer-to-peer architecture also allows data to reside closer to a customer—enabling interactions to be highly responsive and fast. On the back end, Cassandra's capabilities can provide faster catalog refreshes and allow retailers to analyze their catalogs and inventory in real time.

Messaging

Messaging apps have become many users' preferred means of communication, leading to a surge in messaging volumes. Mobile phone and telecom providers use Cassandra's scalability and performance capabilities to provide robust messaging services. Cassandra will also likely play a key role in supporting 'conversational commerce,' the intersection of messaging apps and shopping. This emerging market needs Cassandra's responsiveness, performance, and scalability to enable interactions between users and businesses through messaging and chat apps. These include text apps like Facebook Messenger, WhatsApp, and WeChat and voice technology like Google Assistant and Amazon Alexa, which interface through voice commands.

Healthcare

One key feature that makes Cassandra especially appealing to healthcare data analytics applications is that it does not have a single point of failure. While this is certainly true for all Cassandra use cases, it is a particularly critical asset for healthcare where the ability to quickly access patients' data can result in life-or-death consequences and is an everyday reality. As Cassandra is a database built on availability and partition tolerance (with respect to the CAP theorem of databases), read and write operations do not conflict with each other, allowing uninterrupted access to critical data.

IoT and Edge

Manufacturing and other asset-heavy industries such as oil and gas, utilities, mining, and more use Cassandra for their IoT and edge applications. The reason: Cassandra can handle a large volume of high-velocity, time-series data that IoT and edge devices produce. The database also offers high availability and can ingest concurrent data from any node. Analytics based on such data are routinely used for inventory management and preventive and predictive maintenance.

DECIDING IF
Cassandra is right for you

While Apache Cassandra is a great fit for many use cases, there are certain scenarios where it may not be the best option available. Given the variety of data sources today, the different analytics approaches available, and the wide range of business objectives in using the data, the same database may not be right for every situation. Selecting any database requires matching desired features against application needs; that goes for Cassandra, too.

Cassandra's behavior can be explained by the CAP theorem. The CAP theorem states that a distributed system can provide only 2 of 3 desired properties: consistency, availability, and partition tolerance.

C

Consistency
Every client sees the same data. Every read receives the data from the most recent write.

A

Availability
Every request gets a non-error response, but the response may not contain the most recent data.

P

Partition tolerance
The system continues to operate despite one or more breaks in inter-node communication caused by a network or node failure.

Because a distributed system must be partition tolerant, the only choices are deciding between availability and consistency. If part of a cluster becomes unavailable, a system will either:

- Safeguard data consistency by canceling the request even if it decreases the availability of the system. Such systems are called CP systems.
- Provide availability even though inconsistent data may be returned. These systems are AP distributed systems.

Cassandra chooses to be an AP distributed system. What this means is that Cassandra prioritizes Availability (every request receives a non-error response) over Consistency (every read receives the most recent write). In practice, Cassandra writes multiple copies of data (usually three) to achieve high availability of different cluster nodes. This ensures data is not lost if a node goes down or becomes unavailable. However, when the data is written to multiple nodes, there will be cases when propagation takes time to traverse the network, or a host might be temporarily down or unreachable.

Thus, there exists a tradeoff between consistency and latency as well.

By default, Cassandra provides **eventual consistency.** Simply put, this means a read that follows a write is not guaranteed to return the latest data. Once the data is finished replicating, consistency is restored, at least until the next write.

Cassandra cannot guarantee that all replicas will have the same data. If the update to one node takes longer than to another, a query or read of the same data a short time (a few milliseconds) later may yield two different versions of the data.

Cassandra can be made strongly consistent if operations follow the formula **R + W > RF** where ***R = read consistency, W = write consistency, and RF = replication factor.*** Strong consistency returns up-to-date data for all prior successful writes but at the cost of slower response time and decreased availability.

However, Cassandra does offer **"tunable"** consistency. Developers can change ("tune") this default behavior at the query level to increase data consistency at the expense of availability. In an extreme case, a developer could make Cassandra behave with a similar level of consistency as a single SQL database.

Cassandra does not support isolation or atomicity. If 2 processes update the same data at the same time, Cassandra follows the principle of "last write wins". Exceptions to this behavior can be found in using the BATCH command and lightweight transactions. However, there is a performance penalty when either of these commands is used.

Cassandra’s way of dealing with data may allow partially successful transactions, contain duplications, contradictions and more. If an application requires strong ACID compliance, then Cassandra may not be the best choice. Other areas where Cassandra is not a good fit include applications where there are many updates and deletes, there is a need for a flexible schema, or an app that has unpredictable data usage or query patterns.

WHY ORGANIZATIONS CHOOSE *Cassandra*

That said, Cassandra is known for certain features that make it a good match for many modern data-intensive applications. Some of the most important features include:

Elastic scalability

A Cassandra database deployment can easily be scaled up or down. Its scalability is due to its nodal architecture. Adding or removing nodes can adjust the database system to align with the needs of an organization. One particularly important point about elastic scalability is that it can be done without disrupting the system as a whole.

Reliability

Cassandra replicates data to multiple nodes, which can be geographically dispersed. That provides a level of fault tolerance. A node failure or a site becoming unavailable due to a natural disaster or network disruption does not bring the system down. Data is highly available.

Performance

Cassandra is great for applications that have heavy write workloads and those that require high throughput. Its ability to write quickly is due to its data-handling process. The first step is a write to the commit log, followed by a write to the “Memtable” or memory. After writing to Memtable, a node acknowledges the successful writing of data. Because the Memtable is in the database memory, writing is much faster than writing to a disk. These factors contribute to the speed of Cassandra writes. Memtables are flushed as SSTables to disk once certain thresholds are reached. Durability is maintained by the fact that the write is distributed across multiple nodes, and the commitlog is flushed to disk every 30 seconds or so depending on configuration.

TEAMING WITH *A technology partner*

Like many other open source offerings, organizations always have the option of taking a do-it-yourself approach to using Cassandra. But that requires expertise and financial means that many organizations may not have—and that's exactly why they are turning to technology partners for help.

Building and investing in the necessary expertise, infrastructure, and processes for Cassandra (or any open source technology, for that matter) can easily require an upfront investment of over $450,000—not to mention ongoing operating expenditures, which can cost just as much per year. While these initial outlays are enormous, so is the time required to set up your Cassandra database: typically 4 to 6 months at the very minimum.

If the ultimate goal of using Cassandra is to make fast and efficient use of the vast amounts of data available today, the time spent putting Cassandra into place and optimizing its use would be far better spent on the development of business applications.

As a managed service provider with more than 220 million node hours of management experience, Instaclustr makes it incredibly easy to set up your own Cassandra clusters in weeks and at a fraction of the cost of going it alone.

With zero downtime migrations, the Instaclustr Managed Platform for Apache Cassandra is built with flexible hosting options in mind: on-prem or in your cloud provider of choice. Instaclustr customizes and optimizes the configuration to meet your organization's requirements.

Other features and benefits of Instaclustr for Apache Cassandra include 24x7 support, 100% availability SLA and 99.99% latency SLA, SOC 2 Certification, enterprise-grade security and more. Additionally, Instaclustr's managed platform is designed specifically to work with other common services and apps that complement Cassandra's capabilities, including a Cassandra Connector for Kafka® Connect and a managed service for Apache Spark™.

By removing the biggest obstacles and pain points that trip up so many organizations when pursuing an open source solution, Instaclustr allows you to focus your resources and expertise on what you do best: building and scaling applications.

A FINAL *Word*

There's no getting around it: the right database technology is absolutely critical for business success. Cassandra is frequently the database of choice for many organizations due to its scalability, reliability, and performance. But to unlock its full potential, organizations now require extensive operating expertise, deep pockets, & massive time commitments—precious resources that most simply cannot afford.

That's why managed service providers like Instaclustr are becoming critical components for organizations and their data infrastructure. By only offering 100% open source solutions, Instaclustr will never lock-in your data with proprietary code like so many "open core" managed service providers—giving you unparalleled freedom and flexibility to build and scale applications.

No two organizations are ever the same, and their Cassandra data solutions shouldn't be either. From migrations to managed services, expert support, consulting, and more, NetApp Instaclustr will get you up and running with Cassandra no matter where you are in the project lifecycle.

Discover how PubNub breaks through infrastructure barriers with managed Apache Cassandra services.
Read the Case Study

NetApp® Instaclustr specializes in open source technologies for enterprises. Our managed platform streamlines data infrastructure management, backed by experts who ensure ongoing performance, scalability, and optimization. This enables companies to focus on building cutting edge applications at lower costs.

NetApp Instaclustr

info@instaclustr.com | www.instaclustr.com