

Avoiding the challenges and pitfalls of Cassandra implementation

Abstract

Big data is a different problem that requires a different data storage and management solution.

Apache Cassandra has emerged as a winning solution. However the right deployment strategies and best practices can mean the difference between on-time deployment of applications that scale massively, are always available, and perform blazingly fast, and those that bring your applications to a crawl.

Overview

Big Data is a different problem that requires a different data storage and management solution. It's CR data that is too big, moves too fast, and doesn't fit standard relational data structure. Apache Cassandra has emerged as a winning solution for handling Big Data.

However, the right deployment strategies and best practices can mean the difference between on-time deployment of applications that scale massively, are always available, and perform blazingly fast, and those that bring your applications to a crawl.

This paper is for you if you are part of a team within your company charged with addressing Big Data challenges. It is designed to help you understand some of the most common pitfalls and mistakes that those new to Big Data make.

The paper is divided into five major parts:

- **Part 1** discusses the new challenges of 21st century computing—that of dealing with Big Data, and how Big Data is different from any former data management challenges faced by technologists. Companies don't only have to deal with this new challenge, they will have to deal with it in the face of significant shortage of resources skilled in dealing with Big Data.
- **Part 2** briefly discusses how and why Cassandra was born, and what makes Cassandra the preferred choice for handling big data challenges over other alternatives such as key value, extensible record, and scalable relational data stores. This section also briefly discusses a benchmark test conducted by the University of Toronto team of data scientists and their findings showing that Cassandra was the clear winner for writing intensive applications.
- **Part 3** discusses four common pitfalls and mistakes that technologists make when implementing Cassandra—mistakes made because engineers approach Big Data from a relational database perspective.
- **Part 4** discusses the recommended Best Practices and strategies for deploying, configuring, monitoring, and maintaining Cassandra—getting it up and running in production in a matter of weeks rather than months, at less than 20% of your typical costs, while avoiding very real risks of wrongly deploying Cassandra that could bring your applications to a crawl.
- **Part 5** provides a case study that illustrates the power of Cassandra when implemented correctly.

Part 1: Computing in 21st Century - Big Data

A 2011 McKinsey study entitled, “Big Data: The next frontier for innovation, competition, and productivity” provides some important insights into the sheer scale of Big Data and how fast it is growing:

- ▶ As of 2017, there are 4.43 billion mobile phones in use – by 2020, this number is expected to rise to 4.78 billion
- ▶ Over 30 billion pieces of content are shared each month
- ▶ This amount of data continues to double every three years, while IT spending is projected to grow at 5% per year
- ▶ Fifteen out of seventeen sectors in the US have more data per company than the entire US Library of Congress
- ▶ A 60% potential increase in operating margin for retailers *IF* they get the Big Data challenge right
- ▶ Over \$600 billion potential annual increase in consumer spending from using location data globally
- ▶ A shortage of least 140,000 to 180,000 skilled deep data analysts and a need for over 1.5million data managers to handle and take full advantage of this Big Data revolution.



The report summarizes the Big Data phenomenon as follows:

Data have become a torrent flowing into every area of the global economy

1. Companies churn out a burgeoning volume of transactional data, capturing trillions of bytes of information about their customers, suppliers, and operations. Millions of networked sensors are being embedded in the physical world in devices such as mobile phones, smart energy meters, automobiles, and industrial machines that sense, create, and communicate data in the age of the Internet of Things.
2. Indeed, as companies and organizations go about their business and interact with individuals, they are generating a tremendous amount of digital “exhaust data,” i.e., data that are created as a by-product of other activities. Social media sites, smartphones, and other consumer devices including PCs and laptops have allowed billions of individuals around the world to contribute to the amount of big data available.

And the growing volume of multimedia content has played a major role in the exponential growth in the amount of big data. Each second of high-definition video, for example, generates more than 2,000 times as many bytes as required to store a single page of text. In a digitized world, consumers going about their day—communicating, browsing, buying, sharing, searching, create their own enormous trails of data.

BIG DATA IS NOT JUST BIG. As Edd Dumbill, principal analyst for O’Reilly Radar, and program chair for the O’Reilly Strata Conference and the O’Reilly Open Source Convention, describes it:

 *Big data is data that exceeds the processing capacity of conventional database systems. The data is too big, moves too fast, or doesn’t fit the structures of your database architectures. To gain value from this data, you must choose an alternative way to process it.* 

Different opportunities. Different scale. Different challenges. Different solutions required.

Part 2: Dealing with Big Data

Big Data requires a new and different architecture, designed to solve a new and different data storage challenge - one that requires high scalability, near-100% availability, and high performance under a number of data operations including high read and write - which is where Cassandra comes in.

Apache Cassandra is a distributed storage system for managing very large amounts of structured data spread out across many commodity servers, while providing highly available service with no single point of failure (*Avinash Lakshman, Prashant Malik: Facebook*). In their paper entitled “*Cassandra - A Decentralized Structured Storage System*”, Lakshman and Malik discuss the challenges that Facebook faced and how Cassandra was born in 2008 to address these challenge.

Cassandra scales exceptionally well - it can handle petabytes of information with thousands of concurrent users as well as much smaller amount of both data and user traffic. Just as importantly, its peer-to-peer architecture removes the “single-point-of-failure” problem of the past data distribution architectures, making it a near 100% available data storage system.

A number of benchmark performance studies show that Cassandra provides the best tradeoff in read-write performance and latency when compared with both legacy SQL and NoSQL data stores. Two of these studies are briefly discussed below.

BENCHMARK STUDY 1: NOSQL VS. SQL DATA STORES

In a benchmark study entitled, “Solving Big Data Challenges for Enterprise Application Performance Management”, a University of Toronto team of big data scientists benchmarked six open-source data stores and evaluated these systems with data and workloads that can be found in application performance monitoring, as well as, on-line advertisement, power monitoring, and many other use cases. The team chose from three classifications of data stores - key value (Project Voldemort and Redis); extensible record stores (HBase and Cassandra); and scalable relational stores (MySQL Cluster and VoltDB). The team benchmarked these six different open-source data stores in order to get an overview of the performance impact of different storage architectures and design decisions. Their goal was not only to get a pure performance comparison but also a broad overview of available solutions. The team used the following workloads for the benchmarking test:

- Workload **R**: This workload consisted of 95% read and only 5% write operations.
- Workload **RW**: This workload consisted of 50% write operations, which is typical of most online applications
- Workload **W**: This workload consisted of 99% write operations.
- Workload **RS**: this workload consisted of 47% read and scan operations and 6% write operations, with the remainder consisting of read operations.

In terms of scalability, the team concluded that the clear winner throughout the experiments was Cassandra. Cassandra achieved the highest throughput for the maximum number of nodes in all experiments with a linear increasing throughput from 1 to 12 nodes. The team’s conclusion was that Cassandra’s performance is best for high insertion rates.

BENCHMARK STUDY 2: COMPARISON OF FOUR NOSQL DATABASES

Another benchmarking study, published in April 2015 by EndPoint, compared four NoSQL databases: Apache Cassandra, Couchbase, HBase, and MongoDB.

For mixed operational and analytical workloads, Cassandra clearly outperformed its NoSQL counterparts—performing six times faster than HBase, and 195 times faster than MongoDB.

Common use cases

Consistent with the above benchmark study, Cassandra has been found to be the clear choice for the following use cases where write-intensive operations are the norm:

- Real-time, transactional big data workloads
- Time series data management
- High-velocity device data consumption and analysis
- Media streaming management (e.g., music, movies)
- Social media input and analysis
- Online web retail (e.g., shopping carts, user transactions)
- Real-time data analytics
- Online gaming (e.g., real-time messaging)
- Software as a Service (SaaS) applications that utilize web services
- Online portals (e.g., healthcare provider/patient interactions)

Why Siteminder loves Cassandra

Siteminder's rapid success meant looking for a database technology that would be able to handle the continued massive write-data growth while ensuring greater flexibility, performance, scalability and reliability. That's when they found Cassandra.



SiteMinder



Working with InstaClustr was key to our successful migration from legacy relational database platform to Apache Cassandra in AWS. They have the operational experience and expertise to manage our Apache Cassandra database for our production clusters that underpin our core platform.



Part 3: Cassandra Implementation

Common Pitfalls & Mistakes

The greatest challenge that most developers face when it comes to Big Data is the radical shift in thinking required in how to design and manage high velocity, always-available data stores. Most developers come schooled in relational database modeling, and continually attempt to apply their relational modeling skills to Cassandra deployments.

Data modeling for Cassandra is no exception. It is architected differently to address very different challenges than the ones that relational (SQL) data models were designed to solve. Furthermore, the need to deal with large number of servers only increases the complexities.

Below, we have listed some of the most common pitfalls and mistakes that new Cassandra developers must avoid in order to get the best performance out of their Cassandra deployment.

PITFALL 1: TUNING BEFORE UNDERSTANDING

A common mistake that most new Cassandra users commit is to dive into changing the default settings. Unfortunately, premature tuning leads to a slow cluster and a bad experience. For instance, a new user will often be tempted to increase the amount memory allocated to Cassandra. This typically results in increased latency, and a confused user scratching their head.

Adjusting and tuning Cassandra requires a solid understanding of the data model, usage pattern, previous trends and how each setting impacts the cluster.

PITFALL 2: MAKING CASSANDRA DO THINGS IT SHOULDN'T

Another common mistake that new Cassandra developers coming from a relational background commit is to try to apply rules about relational modeling to Cassandra. Cassandra excels at handling lots of writes and reads. However, applications that require heavy updates or deletes can cause performance headaches. While Cassandra does offer a number of strategies for dealing with heavy update/delete use cases, you must know what you are doing.

Some of the most common mistakes are:

- **Minimizing writes:** While writes are not free, they are very cheap in Cassandra. Data modeling for minimizing writes is a waste of time.
- **Minimizing data duplication:** Cassandra was designed to handle de-normalized and duplicated data. In fact the power of Cassandra in dealing with large volumes of fast writes is based on de-normalization and duplication of data. Trying to minimize these only reduces the power of Cassandra.
- **Modeling data around relations or objects:** In Cassandra deployment, the right way is to model the data around the queries that will be executed, rather than around the data relations.

Rather, the goal is to do things that are different from the goals of traditional relational database optimization. Two important goals are to spread data evenly throughout the cluster and to minimize the number of partitions that will be read. These are different and new optimization skills that must be acquired in order to get the most out of your Cassandra deployment.

PITFALL 3: LEAVING CASSANDRA UNATTENDED

Once Cassandra has been deployed and your data is correctly modeled, you will need to monitor your deployment to ensure Cassandra continues to perform as desired. While Cassandra is generally bulletproof when it comes to outages and network partitions, it is still critical to keep an eye on latency, disk usage, and throughput among other key performance indicators.

Continuous 24x7 monitoring is critical to optimal Cassandra deployment. In today's constantly online world, changes happen—both internally, and on the outside. Internally, applications are developed in agile fashion, with new features constantly getting pushed to production in very frequent intervals. On the outside, users change their usage patterns, promoted by both external changes, as well as your changes to your own application. Surges in write operations, as well as the type of data inserted can change as events occur on the outside that could be short lived, but still have significant impact on the availability of your data.

What to monitor and how to monitor are critical skills required for proper and optimal deployment of Cassandra. Another critical aspect of monitoring and maintaining is applying patches to Cassandra in a production environment—a task that can be confusing and frustrating to new Cassandra users.

PITFALL 4: NEGLECTING SECURITY

Security can sometimes be an afterthought in production deployments especially when time frames are tight. Configuring security in Cassandra can have real world performance and availability impacts. Security is about protecting your company's most critical assets. For many companies that do business online, the most critical and most difficult to protect is typically their data and informational assets.

The challenge with protecting information is that it is very difficult to know when it is stolen since, unlike most assets (including money), it can still be there and be stolen at the same time. This makes it more difficult to know a breach has occurred, as it may seem everything is working normally.

Furthermore, depending on your industry, there are legal or regulatory requirements with which you have to comply. However, while this level may be enough to pass the “have taken reasonable measures” test, it may not be enough to protect all of your assets.

To truly protect your informational assets, you must have the ability to immediately detect intrusions as they occur, prevent the intrusion from succeeding (stealing or compromising your data); and to collect sufficient forensic information on the intrusion to enable prevention of future attacks. Cassandra comes with sophisticated security features, the understanding of which is critical to proper protection of your Cassandra deployment.

Part 4: The Right Strategy for Cassandra Deployment

At this point, you have probably come to the conclusion that Cassandra is the right solution for your needs. What remains to be decided is what your deployment strategy will be:

Do you deploy Cassandra yourself, and therefore commit to building world-class expertise in Cassandra deployment and management, or do you outsource it?

This is the classic “buy or build” decision companies have been making regarding practically every business process:

- Do we hire our own recruiters, or do we rely on outside recruiters?
- Do we manage our mission critical email servers, or do we outsource email hosting?
- Do we hire our own CPA to do our taxes, or do we outsource tax preparation to a reputable CPA firm?

In all these cases, the decision process is exactly the same. Unless there is a compelling reason for “building” something, the right decision is always to “buy” ***what is not core to your business.***

Outsource the critical. Focus on the strategic.

There is no question that Cassandra deployment is a critical business process—after all, your mission critical applications will run on it. The question is:

1. Can you build the necessary expertise and infrastructure to meet this critical requirement in time, before your competitors do; or
2. Is the right strategy for you to outsource this responsibility to a **Cassandra Managed Services Provider** that has already made the necessary investments to build world-class expertise and infrastructure, so that you can focus on building world class expertise in your applications?

This is a serious decision to be made, one that can have very serious consequences.

To help you analyze this decision and make the right options, we bring to your attention three key points to consider when deciding whether it makes sense to build your own Cassandra competency center, or to outsource Cassandra management to an expert Cassandra Managed Services Provider.

For most companies that have made the decision to run their applications on Cassandra, the cost, timeline, and overall business risk levels of deploying their own Cassandra infrastructure are just too high to take on, especially when there are better options available.

Key Point 1: Financial Considerations

Unless your company is willing to invest, at a minimum, \$400,000 to \$500,000 a year on building the necessary expertise, infrastructure, and processes, you should outsource this very critical operation to those that have already made such an investment.

The chart on the below shows that, for the first year, a company can expect to spend at least \$400,000 to \$450,000 to deploy its own Cassandra infrastructure, and nearly as much for each year after.

At present, the demand for Cassandra expertise far outweighs the supply. Recruiting fees are high, and so is the cost of keeping these experts within your company.

Each year, you can expect the salaries for your core Cassandra team to grow by 10% or more just to keep them in your company as recruiters call on them with higher and higher financial rewards if they leave your company.

Year 1	Base	Burdened
Cassandra expert	200,000	236,000
Support 1	90,000	106,200
Recruiting cost	58,000	58,000
Equipment	30,000	30,000
Other	5,000	5,000
TOTAL FIRST YEAR		435,200
Year 2	Base	Burdened
Cassandra expert	220,000	259,600
Support 1	99,000	116,820
Equipment	15,000	15,000
Other	5,000	5,000
TOTAL SECOND YEAR		396,420

Compare this with the cost of outsourcing to a Managed Services Provider that charges an average of \$60,000 to \$70,000 per year to expertly deploy, manage, and monitor your Cassandra database - for less than one-fifth the cost.

Key Point 2: Timeline Considerations

As has been indicated in the above section, there are very few Cassandra experts who have done enough Cassandra implementations to gain sufficient expertise in bringing your application into production quickly. Typical timelines for in-house deployments run between four (4) to six (6) months to get Cassandra up and running optimally.

Compare that with a Cassandra Managed Services Provider that can get you up and running expertly in a matter of **2-3 weeks** - again this is an **order of magnitude difference!**

Unlike the cost consideration - which is a mostly operational in nature - this has strategic implications. In today's ever-changing customer needs, the ability to develop rapidly or to quickly release new features that can keep pace with growing customer needs must be matched with an ability to monitor 24x7 and rapidly make necessary changes to your Cassandra deployment. Otherwise, your very own agile development competencies could bring your production performance to a crawl.

Cassandra Managed Service Providers assume this to be the case. They constantly monitor throughput, latency, and disk usage, and make necessary adjustments on a 24x7 basis.

Key Point 3: Overall Business Considerations

A third, and perhaps the most important, consideration is the risk of getting it all wrong. As pointed out in some detail in Part 3, this is a very real concern. Cassandra is a different type of data store, and it takes some time to get a real feel for how it works and the proper way to implement and manage it. A team can spend months getting it up and running, only to find out it is not performing as expected, or have difficulty keeping up with changes in customer usage that affect the performance of Cassandra. Furthermore, if one or more of your Cassandra team members leaves for any reason, you run a very serious risk of having little or no in-house capabilities to deal with any failures that may occur. And, as we have already indicated, the current demand for Cassandra experts far exceeds the supply, making it nearly impossible to find quality Cassandra resources quickly.

While this third consideration is the hardest to quantify, it is perhaps the most important consideration as it is difficult to determine the impact on your business, until something bad happens.

Summary comparison chart

The chart below compares the cost, timeline, and overall risk considerations between deploying in-house, and outsourcing to an expert Cassandra Managed Services Provider.

Issue	In-house Deployment	Cassandra Managed Service	Remarks
Cost	\$400,000 to \$500,000 minimum per year	\$60,000 to \$70,000 per year for most deployments	An 80% cost savings
Timeline	Minimum of 4-6 months to be on production environment	Typically 2-3 weeks to be on production environment	Order of magnitude difference
Overall risk	Risk of failure if one or more staff leaves	Minimal to no risk as 80% or more of technical staff are Cassandra experts	Not easy to quantify risk but significant to unacceptable for most mission critical applications

Part 5: Case Study

Peloton scales data volumes reliably with InstaClustr



Overview

Peloton was founded in 2012 to create a new concept in fitness. The founding team loved cycling but had a hard time finding a workout that consistently fit with their busy schedules, and at-home workouts never felt quite as good as a class. They set out on a mission to create a world-class indoor cycling studio experience that would rival the in-class experience – all from the comfort of home.

Highlights

- Fast growing startup has raised \$325 million
- Has tripled revenue and subscriber base in 12 months
- 100,000 paying subscribers
- Data-driven software provides personalized experience
- Data growth is exponential
- Was having issues with Apache Cassandra in production
- Moved to InstaClustr to scale reliably with zero downtime

Working with InstaClustr has been great. They fixed our production issues with Apache Cassandra very quickly, and have since taken on responsibility for running our clusters in production. This means we can focus our efforts on building better products for our growing subscriber base, while feeling safe in the knowledge that our Cassandra clusters are always running at peak performance .

Kengo Hashimoto
Senior Software Engineer at Peloton Cycle

Challenge

Peloton Cycle has a challenge that any startup would love to have. They are growing so fast from both a revenue and customer perspective, that it can be difficult to keep up with the growing data requirements. In order to provide the very best experience to their customers that they can, data is incredibly important to the company.

Peloton tracks and stores telemetric data from every ride that their customers take on their bikes, and due to the exceptional customer growth data volumes have been exploding. They track everything from how fast someone is pedaling, what resistance setting they are using, what their heart rate is

like, and much more. Peloton then uses this data to display information back to their users with charts and graphs, including the top 10 all-time rides, their ratings, and the top 10 routines from all users at any given time.

Apache Cassandra is architected as a massively scalable, distributed database, and will have no issues with the exponential data growth that Peloton has been seeing, and is forecasting over the next three years. Being a small team Peloton does not have a lot of Cassandra skills in house. The engineering team wants to spend their valuable time building an increasingly better product, so doesn't want to wrestle with having to learn how to run open source Cassandra in production at massive scale.

They were starting to have some serious issues with performance in their production systems. The team had opted for fewer dense nodes, instead of a larger number of smaller nodes. One day there was a reconciliation issue between Chef and AWS which led to two nodes being destroyed, which in turn caused a painful and lengthy recovery process. The remaining nodes experienced read pressures, and fell behind.

They needed a partner that could diagnose and fix the issues they were experiencing right then, and then also take over as a managed service to free up important resources.

Solution

Peloton contacted Instaclustr, that provides a host of managed services for best of breed open source technologies, including Apache Cassandra, to see if their consultants could figure out the root cause of the performance issues that they had been experiencing.

Instaclustr diagnosed and fixed the production issues that same day. They found the problems related to running very dense nodes in production, and once the environment was right-sized from a node standpoint, Apache Cassandra resumed writing and reading the packet data in real-time.

Apache Cassandra is an amazing distributed database that is capable of handling incredible volumes and scale with ease. While relatively easy to develop against once the right data model is in place, it takes a considerable amount of experience and expertise to run the database in production.

For ongoing operations Peloton had two choices; they could hire for Cassandra expertise in-house, and have a dedicated team for operations, or look to Instaclustr's managed service. Having a dedicated team was deemed to be too costly, and would detract from core competencies.

Instead, the team decided to engage with Instaclustr to manage their production environment for them with tight SLAs in place. They already were using a managed service for another technology from a different vendor so it was a familiar model.

The Results

Since engaging with Instaclustr Peloton has been able to focus on their core competencies, while Apache Cassandra has experienced tremendous growth with zero downtime. Their data volumes are growing at a monthly rate of 8% and they expect to double their node count by next year, with over

500TB of data planned. Along with this data growth they expect their monthly active users to grow quickly to 350,000, and within two years they forecast that they will have over a million monthly active users.

They are currently running in two data centers in AWS, and as they grow their footprint globally they plan to add more data centers, which Cassandra can handle with ease. Additionally, Peloton is excited that Instacluster is agnostic and can run in any cloud environment, so it gives them freedom and flexibility to move to or adopt a different public cloud vendor in the future if required.

About Instacluster

Instacluster is the open source as-a-service company, delivering reliability at scale with Apache Cassandra, Apache Spark, Elasticsearch and other related technologies.

Our expertise leverages over 15 million node hours and one petabyte under management, allowing us to run the world's most powerful NoSQL distributed database effortlessly.

Our open source data management solutions power mission critical, highly available applications for our customers and help them achieve scalability, reliability and performance in their database.

Our Services

Instacluster provides range of consulting and support services to help our customers develop and deploy solutions that use Apache Cassandra and related technologies.

Managed Solutions	Our managed solutions enable our customers to run seamlessly in the cloud with confidence that our team is monitoring and managing their clusters.
Consulting Services	Our team of experts have extensive experience with a wide range of use cases and deployed architectures.
Enterprise Support	Our dedicated 24x7 support team of experts have hundreds of thousands of hours of operational engineering and support experience.

Build, run and scale your app with confidence

Like what you see?

If you're looking at building a proof of concept for your application, or looking for production grade nodes, contact a member of our [Sales Team](#) to discuss your specific needs.