Introduction to Apache Cassandra

WHITE PAPER

By DataStax Corporation

July 2012

Contents

Introduction

Apache Cassandra™ is a massively scalable NoSQL database. Cassandra's technical roots can be found at companies recognized for their ability to effectively manage big data – Google, Amazon, and Facebook – with Facebook open sourcing Cassandra to the Apache Foundation in 2009.

Used today by numerous modern businesses to manage their critical data infrastructure, Cassandra is known for being the solution technical professionals turn to when they need a NoSQL database that supplies high performance at massive scale, which never goes down. In particular, Cassandra addresses big data applications, which are exploding across nearly every industry.

This paper provides a brief overview and introduction to Cassandra for those wishing to understand if Cassandra is right for them and how it is uniquely positioned to address the next phase of growth in the modern database marketplace.

Built by Necessity

Traditional relational databases (RDBMSs) such as Oracle and Microsoft SQL Server have been the primary data stores for IT applications since the mid-1980s. But as the first phase of the Web got under way, out of necessity new databases (such as Oracle's MySQL) were introduced that had RDBMS roots, but different feature sets designed to handle the new data access patterns Web 1.0 produced.

Today, a new shift in data management applications has occurred – one that involves the next phase of the Web plus data-related aspects that have come to be characterized as big data in nature. Big data involves data that (1) is high velocity in nature; (2) combines structured, semi-structured, and unstructured data; (3) can include enormous volumes; and (4) typically involves complexity in data distribution and synchronization.

The massive scale, high performance, and never-go-down nature of these applications has forged a new set of technologies that have replaced the legacy RDBMS, with O'Reilly describing the situation in this way:

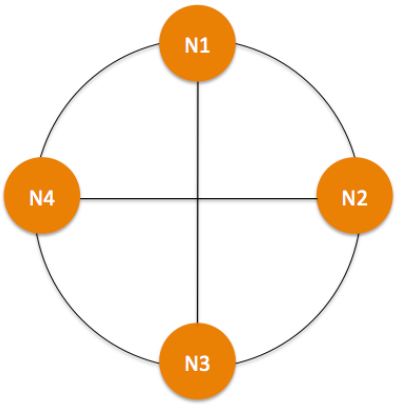
"Big data is data that exceeds the processing capacity of conventional database systems. The data is too big, moves too fast, or doesn't fit the structures of your database architectures. To gain value from this data, you must choose an alternative way to process it."[1]

[1] "What Is Big Data? An Introduction to the Big Data Landscape," by Edd Dumbill, O'Reilly Radar, January 11, 2012: http://radar.oreilly.com/2012/01/what-is-big-data.html.

Out of these new technologies, Cassandra has become one of the leading choices of IT architects and decision-makers who are building modern big data applications.

The Architecture of Cassandra

The architecture of Cassandra greatly contributes to its being able to scale, perform, and offer continuous availability. Cassandra was built from the ground up with the understanding that hardware and system failures can and do occur. This translates into Cassandra sporting a different way of managing and protecting data than a traditional RDBMS.



Rather than using a legacy master-slave or a manual and difficult-to-maintain sharded design, Cassandra has a peer-to-peer distributed architecture that is much more elegant, and easy to set up and maintain. In Cassandra, all nodes are the same; there is no concept of a master node, with all nodes communicating with each other via a gossip protocol.

Cassandra's built-for-scale architecture means that it is capable of handling petabytes of information and thousands of concurrent users/operations per second (across multiple data centers) as easily as it can manage much smaller amounts of data and user traffic. It also means that, unlike other master-slave or sharded systems, Cassandra has no single point of failure and therefore is capable of offering true continuous availability.

Distributing and Replicating Data

Cassandra provides automatic data distribution across all nodes that participate in a "ring" or database cluster. There is nothing programmatic that a developer or administrator needs to do or code to distribute data across a cluster. Data is transparently partitioned across all nodes in either a randomized or ordered fashion, with random being the default.

Cassandra also provides built-in and customizable replication, which stores redundant copies of data across nodes that participate in a Cassandra ring. This means that if any node in a cluster goes down, one or more copies of that node's data is available on other machines in the cluster.

Unlike complicated replication schemes in various RDBMSs or other NoSQL databases, replication in Cassandra is extremely easy to configure. A developer or administrator simply indicates how many data copies are desired, and Cassandra takes care of the rest. Replication options are provided that also allow for data to be automatically stored in different physical racks (thus ensuring extra safety in case of a full rack hardware failure), multiple data centers, and cloud platforms.

Multi-Data Center and Cloud Support

Cassandra is the acknowledged leading NoSQL database when it comes to replicating data across many different data centers and cloud platforms. A developer or administrator can implement a single Cassandra cluster that spans many different data centers, or involves a hybrid on-premise/cloud design.

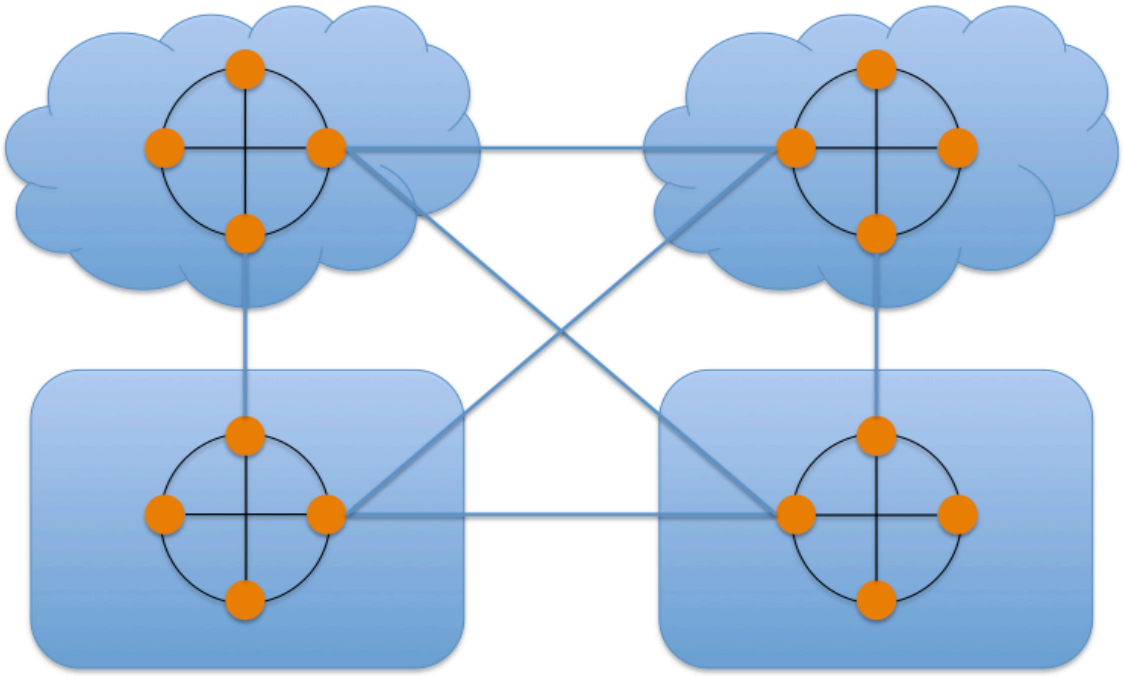


Figure 2: Cassandra supports hybrid on-premise/cloud deployments

When creating a new Cassandra database (also called a *keyspace*), a user simply indicates via a single command which data centers and/or cloud providers will hold copies of the new database; everything from that point forward is automatically handled and maintained by Cassandra.

Reading and Writing Data

Cassandra supplies a true, “location independent” architecture when it comes to reading and writing data. This means any node in a Cassandra cluster (no matter if that node is part of a single or multi-data center setup) may be read or written to, which translates into a true read/write-anywhere design.

When data is written to Cassandra, it is first written to a commit log, which ensures full data durability and safety. Data is also written to an in-memory structure called a *memtable*, which is eventually flushed to a disk structure called an *sstable* (sorted strings table).

If one or more nodes responsible for a particular set of data are down, data is simply written to another node, which temporarily holds the data. Once the node(s) come back online, they automatically bring themselves back up to date from nodes that are holding the data they maintain.

Reading data is performed in parallel across a cluster. A user requests data from any node (which becomes that user's *coordinator node*), with the user's query being assembled from one or more nodes holding the necessary data. If a particular node having the required data is down, Cassandra simply requests data from another node holding a replicated copy of that data.

While Cassandra is not a transactional database in the same way that legacy RDBMSs offer ACID transactions, it does offer the "AID" portion of ACID, in that data written is atomic, isolated, and durable. The "C" of ACID does not apply to Cassandra, as there is no concept of referential integrity or foreign keys. In a sense, Cassandra can be said to offer support for big data transactions (OLTP).

Data Consistency

Cassandra offers *tunable* data consistency across a database cluster. This means a developer or administrator can decide exactly how strong (e.g., all nodes must respond) or eventual (e.g., just one node responds, with others being updated eventually) they want data consistency to be. This tunable data consistency is supported across single or multiple data centers, and a developer or administrator has many different consistency options from which to choose.

Moreover, consistency can be handled on a *per operation* basis, meaning a developer can decide how strong or eventual consistency should be per SELECT, INSERT, UPDATE, and DELETE operation. For example, if a developer needs a particular transaction to be available on all nodes throughout the world, they can specify that all nodes must respond before a transaction is marked complete. On the other hand, a less critical piece of data (e.g., a social media update) may only need to be propagated eventually, so in that case, the consistency requirement can be relaxed.

What About Performance?

One of Cassandra's hallmarks is its high performance, for both reads and writes, which scales linearly when new nodes are added to a cluster.

For example, Netflix, which uses Cassandra extensively in production, gave a presentation at the 2011 High Performance Transaction System workshop that demonstrated both the ease of use and linear performance capabilities of using Cassandra in the cloud. The following is an excerpt from a Netflix blog post summarizing the presentation:

"The automated tooling that Netflix has developed lets us quickly deploy large scale Cassandra clusters, in this case a few clicks on a web page and about an hour to go from nothing to a very large Cassandra cluster consisting of 288 medium sized instances, with 96 instances in each of

three EC2 availability zones in the US-East region. Using an additional 60 instances as clients running the stress program, we ran a workload of 1.1 million client writes per second. Data was automatically replicated across all three zones making a total of 3.3 million writes per second across the cluster."[2]

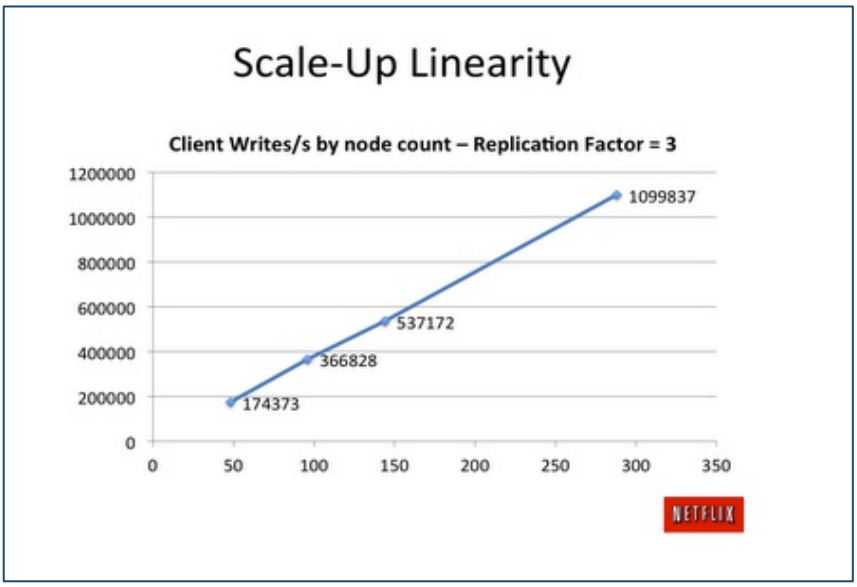


Figure 3: Performance results from Netflix's cloud benchmark of Cassandra

Cassandra tends to outpace its NoSQL competitors in performance for many use cases. One illustration of this is found in an academic benchmark paper presented at the 2012 Very Large Database Conference in Istanbul. A team of performance engineers benchmarked Cassandra along with a number of other NoSQL and SQL databases (e.g., HBase, Redis, Voldemort, MySQL cluster) in a variety of tests, with their core finding being:

"In terms of scalability, there is a clear winner throughout our experiments. Cassandra achieves the highest throughput for the maximum number of nodes in all experiments with [linearly] increasing throughput from 1 to 12 nodes."[3]

[2] "Benchmarking Cassandra Scalability on AWS – Over a million writes per second," by Adrian Cockcroft and Denis Sheahan, *The Netflix "Tech" Blog*, November 2, 2011, http://techblog.netflix.com/2011/11/benchmarking-cassandra-scalability-on.html.

[3] *Solving Big Data Challenges for Enterprise Application Performance Management*, by Tilman Rable, et al., August 2012, p. 10.

As a more specific one-on-one comparison, compared to HBase (which is many times likened to Cassandra as a real-time, NoSQL database), the performance team found Cassandra to have:

- 10x more read throughput
- 8x faster read latency (up to 100x faster[4])
- 8x more write throughput
- 10x slower write latency (with the default configuration; that is, no write durability for HBase)
- 8x faster scan latency
- 4x more scan throughput

The bottom line of this comparison: Cassandra, when put to the test, offers high performance at extreme scale, which cannot be said of any other NoSQL database.

Developing Cassandra Applications

The primary difference developers will find when developing applications against Cassandra vs. RDBMSs is the data model. Cassandra uses a Google Bigtable model, which provides more flexibility than a relational design and can more easily store structured, semi-structured, and unstructured data.

Because NoSQL databases like Cassandra do not support operations like SQL joins, data tends to be highly denormalized. While such a thing (wide rows) is normally a problem for an RDBMS, Cassandra provides exceptional performance for objects with many thousands of columns.

The primary container of data is a *keyspace*, which is like a database in an RDBMS. Inside a keyspace are one or more *column families*, which are like relational tables, but they are more fluid and dynamic in structure. Column families have one to many thousands of columns, with both primary and secondary indexes on columns being supported.

In Cassandra, objects are created, data is inserted and manipulated, and information queried via *CQL* – the Cassandra Query Language, which looks nearly identical to SQL. Developers coming from the relational world will be right at home with CQL and will use standard commands (e.g., INSERT, SELECT) to interact with objects and data stored in Cassandra.

Cassandra drivers and client libraries for all popular development languages can be found and freely downloaded from the DataStax website.

[4] Ibid, Figure 10.

Cassandra Use Cases

Many modern businesses and organizations are using Cassandra for critical applications today. Below are just a few examples:

Figure 4: A sample of companies and organizations using Cassandra in production

Some of the application use cases that Cassandra excels in include:

- Real-time, big data workloads
- Time series data management
- High-velocity device data consumption and analysis
- Media streaming management (e.g., music, movies)
- Social media (i.e., unstructured data) 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)
- Most write-intensive systems

Managing and Monitoring Cassandra

Much of Cassandra is self-managing, however there are various administration and monitoring tasks that are carried out with the database just as with other NoSQL and relational systems. One of the simplest ways to perform these operations is by using DataStax OpsCenter.

DataStax OpsCenter is a visual management and monitoring solution for Cassandra and other big data technologies such as Apache Hadoop™ and Solr™. Being web-based, OpsCenter allows a developer or administrator to manage and monitor all aspects of Cassandra easily from any desktop, laptop, or tablet without installing any client software.

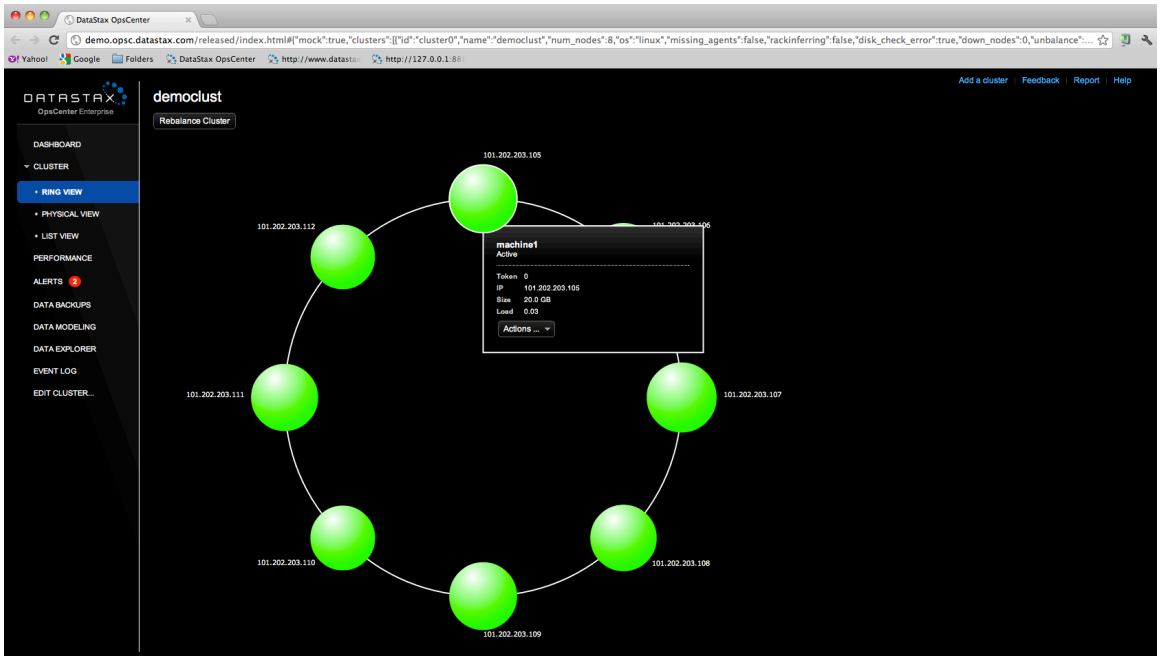


Figure 4: Managing an 8-node Cassandra cluster with DataStax OpsCenter

DataStax OpsCenter comes in both a free/community edition and an enterprise edition.

Deploying Cassandra in the Enterprise

Modern businesses know there is a difference in using open source for non-production projects and deploying it in critical production systems throughout the enterprise. Cassandra is no different in this regard.

From an open source perspective, Cassandra is a top open source project for the Apache foundation and enjoys strong community support and developer involvement. New community releases and patches are produced very quickly, with the understanding that community builds

are not put through any real quality assurance process, and often contain a mixture of enhancements plus bug fixes.

Smart enterprises that want to use Cassandra in production know better than to blindly select one of the many Cassandra community builds, and hope all goes well. Instead, they turn to DataStax for ensuring their success with Cassandra.

DataStax is the commercial company behind Cassandra, and employs the Apache chair of the Cassandra project as well as most of the committers. For enterprises wanting to use Cassandra in production, DataStax supplies DataStax Enterprise Edition, which includes:

- A certified version of Cassandra that has passed DataStax's rigorous internal certification process; this includes heavy quality assurance testing, performance benchmarking, and more
- An integrated Apache Hadoop distribution for analytic operations that includes MapReduce, Hive, Pig, Mahout, and Sqoop support
- Bundled enterprise search support with Apache Solr
- An enterprise version of DataStax OpsCenter
- Expert, 24x7x365 production support
- Certified maintenance releases

DataStax Enterprise Edition is completely free to use in development environments. Production deployments require that a software subscription be purchased.

DataStax also provides DataStax Community Edition for open source enthusiasts who want the fastest and easiest way to try out the latest Apache Cassandra release in non-production environments.

Conclusion

There's no question that many modern applications have outgrown legacy relational databases. To handle big data workloads, these systems require a massively scalable NoSQL database.

While there are a number of NoSQL database providers in the market, only Cassandra is able to offer the linear scale performance and key enterprise-class features that meet the expectations and requirements of big data systems.

To find out more about Cassandra and DataStax, and to obtain downloads of Cassandra and DataStax Enterprise software, please visit www.datastax.com or send an email to info@datastax.com.

About DataStax

DataStax, the commercial leader in Apache Cassandra™, offers products and services that make it easy for customers to build, deploy and operate big data applications. Over 190 customers use DataStax today, including leaders such as Netflix, Cisco, Rackspace, and Constant Contact, with industries served including web, financial services, telecommunications, logistics and government.

DataStax is backed by industry-leading investors, including Lightspeed Venture and Crosslink, and is based in Burlingame, CA, with offices also in Austin, TX. For more information, visit www.datastax.com.