Why NoSQL?

White Paper

BY DATASTAX CORPORATION

October 2013

Table of Contents

Abstract

"Not Only SQL" (NoSQL) refers to progressive data management engines that meet the needs of modern business applications, needs such as scaling to previously unimagined levels and remaining always available and lightning fast. You may wonder when and why NoSQL databases should be used, and when a traditional relational database might suffice. This paper discusses the six most common reasons that NoSQL databases are deployed and highlights how Apache Cassandra™ and DataStax Enterprise fulfill those use cases.

Introduction

By all accounts, the consensus of IT professionals and industry database experts seems to be that NoSQL is here to stay. A recent study performed by a media firm on NoSQL market growth forecasts a very strong compound annual growth rate (CAGR) of 21 percent for NoSQL technology from 2013-2018[1]. Such growth and increased adoption has prompted one technology writer to go so far as to say: "NoSQL is the stuff of the Internet age."[2]

The term "NoSQL" is sometimes misused and abused by various software vendors and technology professionals. In general, NoSQL refers to progressive data management engines that go beyond legacy relational databases in satisfying the needs of today's modern business applications. A very flexible data model, horizontal scalability, distributed architectures, and the use of languages and interfaces that are "not only" SQL typically characterize NoSQL technology.

While what defines NoSQL databases has been much more clearly articulated today than just a few years ago, what still puzzles some IT professionals is when and why NoSQL databases should be used. When will a traditional RDBMS suffice for an application and when is a NoSQL database more appropriate?

This paper discusses six of the most common reasons NoSQL databases are being deployed today, and highlights how Apache Cassandra™ and DataStax Enterprise fulfill those use cases.

You Have Big Data

A recent article on SmartPlanet.com observed: "As recently as 2009 there were only a handful of big data projects and total industry revenues were under $100 million. By the end of 2012 more than 90 percent of the Fortune 500 will likely have at least some big data initiatives under way."[3]

Two unmistakable trends in IT these days are: (1) Big data is real and being used by smart businesses as a strategic weapon in how they do business, and (2) NoSQL is becoming synonymous with big data, especially with respect to how it is managed.

This last point is nearly universally agreed upon by expert IT analyst firms such as IDC, which states:

---

[1] "NoSQL Market Forecast 2013-1018," Market Research Media: http://www.marketresearchmedia.com/?p=568.
[2] "The Time for NoSQL Standards Is Now" by Andrew Oliver, June 7, 2012, InfoWorld: http://www.infoworld.com/d/data-management/the-time-nosql-standards-now-194998.
[3] "Big Data Market Set to Explode This Year, but What Is 'Big Data'?" by Joe McKendrick, SmartPlanet.com, February 21, 2012: http://www.smartplanet.com/blog/business-brains/big-data-market-set-to-explode-thisyear-but-what-is-8216big-data/22126.

"Big data technologies describe a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis."[4]

The first reason to use NoSQL is because you have big data projects to tackle. A big data project is normally typified by:

1. **Data velocity** – lots of data coming in very quickly, possibly from different locations
2. **Data variety** – storage of data that is structured, semi-structured, and unstructured
3. **Data volume** – data that involves many terabytes or petabytes in size
4. **Data complexity** – data that is stored and managed in different locales, data centers, or cloud geo-zones

How DataStax Helps Manage Big Data

DataStax is the commercial company behind Apache Cassandra – a massively scalable NoSQL database designed specifically for big data workloads. When it comes to big data scale, a recent GigaOM report confirms Cassandra is the standard for big data systems:

"Cassandra shines as the paragon of non-relational scalability, and it tends to attract users for whom scale is a pressing problem. This includes traditional web companies, financial firms, telecom firms, and others with transactional data loads larger than relational databases are able to manage."[5]

Where big data management is concerned, Cassandra has a number of advantages over relational database management systems (RDBMSs) including:

- Superior write performance for data velocity
- Strong data type support for data variety
- Linear scalability with horizontal scale-out for data volume
- Very fast response times for both reads and writes

For those wanting to use Cassandra in production for big data applications, DataStax provides DataStax Enterprise Edition, which includes an enterprise-ready version of Cassandra plus the ability to run analytics and enterprise search on Cassandra data. With DataStax Enterprise, modern businesses get a complete, production-ready big data platform that contains:

- A certified version of Cassandra that has passed DataStax's rigorous internal certification process, which includes heavy quality assurance testing and performance benchmarking
- Integrated analytics via a number of Hadoop components including MapReduce, Hive, Pig, Mahout, and Sqoop that are used to run analytics on Cassandra data
- Bundled enterprise search support using Apache Solr
- Enterprise-class security
- Automatic management services that transparently perform maintenance operations and provide intelligent help for performance management without IT staff involvement
- DataStax OpsCenter, which is a visual management and monitoring tool for Cassandra
- Expert, 24x7x365 support
- Certified software maintenance releases

[4] Extracting Value from Chaos, by John Gantz and David Reinsel, IDC, June 2011: http://idcdocserv.com/1142.
[5] "Emerging Trends in the Non-Relational Database Market," by Tim Berglund, GigaOM, September 6, 2012: http://pro.gigaom.com/2012/09/emerging-trends-in-the-non-relational-database-market/.

Big Data Performance

When it comes to scaling performance under general or big data workloads, Cassandra has a strong track record of delivering response times that do not disappoint. Verification of this came in the form of a recently published academic white paper that focused on benchmarking big data databases, which was presented at one of the Very Large Database Conferences in August 2012. The authors of the benchmark paper summarized their findings this way:

"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 a linear increasing throughput."[6]

You Need Continuous Availability

A second reason to consider a NoSQL solution is that your applications need to be continuously available. Note that this is different from just "high availability," where unplanned downtime, although not desired, is still expected. Continuous availability describes a feature of systems that can't go down.

In today's marketplace, where the competition is just a click away, downtime can be deadly to a company's bottom line and reputation. Average downtime costs vary considerably across industries, from approximately $90,000 per hour in the media sector to about $6.48 million per hour for large online brokerages. With an average outage lasting 200 minutes, or a little over three hours, the costs add up quickly.[7]

Besides up-front outages costs, the damage to a company's reputation can affect future revenue as well – just look to the memorable system outages suffered by Blackberry and airline Virgin Blue, as examples.[8]

How DataStax Helps With Continuous Availability

Cassandra is known for being a solution technical professionals turn to when they need a real-time NoSQL database that supplies high performance at massive scale, which never goes down. Cassandra was architectured from the ground up with the understanding that hardware failures can and do occur.

> *"For us, the primary motivating factors are continuous availability and multi-data center support. We also like the fact that we can trust Cassandra; when we need to write data, we [know it will] get written and be there no matter what." – RightScale*

Cassandra's distributed architecture supplies no single points of failure and built-in redundancy of both function and data. Systems can be architected to provide continuous availability in single locations, across multiple data centers, and the cloud.

This functionality extends into DataStax Enterprise, so that analytics and enterprise search operations all sport continuous availability.

---

[6] Benchmark paper presented at the Very Large Database Conference, 2012: Solving Big Data Challenges for Enterprise Application Performance Management, by Tilman Rable, et al., August 2012, p. 10: http://vldb.org/pvldb/vol5/p1724_tilmannrabl_vldb2012.pdf.
[7] Downtime, Outages, and Failures – Understanding Their True Costs, Evolven, February 2012: http://www.evolven.com/blog/downtime-outages-and-failures-understanding-their-true-costs.html.
[8] Ibid.

You Need Data Location Independence

A third motivation to use NoSQL is that you need true location independence with a database. The term "location independence" practically means the ability to read and write to a database regardless of where that I/O operation physically occurs, and to have any write functionality propagated out from that location, so that it's available to users and machines at other sites.

Such functionality is easy to articulate, but difficult to architect for most traditional databases. Master/slave and manually sharded architectures can sometimes meet the need for location independent read operations, but writing data everywhere is a different matter.

The reasons for needing location independence are many and include servicing customers in many different geographies and needing to keep data local at those sites for fast access.

How DataStax Supports Location Independence

Cassandra provides out-of-the-box support for location independence through its built-in replication and data consistency model that supports multiple data centers and cloud geo-zones.

Moreover, systems can be architected to support a hybrid on-premise and cloud model. Data can be literally read and written to anywhere.

"I can create a Cassandra cluster in any region of the world in 10 minutes. When marketing decides we want to move into a certain part of the world, we're ready."
– Netflix

Cassandra's tunable data consistency feature also puts the developer in charge of how and when data is propagated out to other locations. Either very strong or eventual consistency can be chosen on a per-operation basis.

This means, for example, that a developer can mandate that one write operation won't be marked complete unless all machines at all locations respond, while another write operation can be labeled complete when only one machine in one location responds (with the data change eventually being propagated to all other participating nodes).

Finally, these capabilities are extended in DataStax Enterprise to include Hadoop and Solr activities in addition to Cassandra.

You Need Modern Transactional Capabilities

A fourth reason to use NoSQL databases is that you have applications that need modern transactional capabilities.

The concept of transactions appears to be changing in the Internet age. Industry expert Dan McCreary says, "Ninety-five percent (95%) of database-driven systems today don't need ACID transactions."[9]

At first blush, this assertion sounds extreme, as transactional integrity is a characteristic of most every data system with information requirements that demand accuracy and safety. However, what McCreary and others refer to is not the jeopardizing of data, but rather the new way modern applications ensure transactional consistency across widely distributed systems.

The "C" in ACID refers to data consistency in RDBMSs that are enforced via foreign keys/referential integrity constraints. This type of consistency is not utilized in progressive data management systems such as NoSQL databases because there are no join operations and such.

Instead, the "C" that concerns NoSQL databases is found in the CAP theorem, which signifies the immediate or eventual consistency of data across all nodes that participate in a distributed database. The data is still safe and meets the AID portion of the RDBMS ACID definition, but its consistency is maintained differently given the nature and architecture of the system.

How DataStax Supports Transactions

Cassandra supports AID transactions and also has a very flexible model for handling data consistency across a distributed database. As mentioned in the previous section, Cassandra's tunable data consistency provides developers with the ultimate agility in determining how strong or eventual they wish transactional consistency to be on a per-operation basis.

"Cassandra stands at the front of the NoSQL pack when it comes to supporting real-time, big data applications."
– Wikibon

You Need a More Flexible Data Model

One of the major reasons IT professionals move to a NoSQL database from a legacy RDBMS is the more flexible data model that's found in most NoSQL offerings. This is a fifth reason why you might use a NoSQL datastore: While the relational model works well for a number of use cases, a NoSQL data model can support many of those use cases and others that don't fit well into an RDBMS.

Moreover, a NoSQL datastore is able to accept all types of data – structured, semi-structured, and unstructured – much more easily than a relational database. For applications that have a mixture of datatypes, a NoSQL database is a good option.

Lastly, performance factors come into play with an RDBMS' data model, especially where "wide rows" are involved and update actions are many. However, a NoSQL data model such as Google's Bigtable easily handles both situations and delivers very fast performance for both read and write operations.

[9] "The CIO's Guide to NoSQL," Dataversity Webinar, June 12, 2012: http://www.dataversity.net/webinar-thecios-guide-to-nosql-2/.

How DataStax Provides Data Model Flexibility

Apache Cassandra uses the Google Bigtable data model, which means it sports a familiar row/column paradigm that developers will be used to. However, the data model in Cassandra is much more agile than a standard RDBMS and capable of handling many different use cases. As just one example, one row of data may have five columns and another row may have 5,000 columns – and all are query-able in the same object.

"Cassandra's NoSQL data model allows us to insert and query data much more naturally than what we had previously. The analysts who routinely use this data were impressed with the flexibility and speed at which the queries came back." – CSC/NASA

The dynamic data model in Cassandra also handles all types of data easily, including unstructured data, and provides indexes for both primary and secondary key columns.

You Need a Better Architecture

Although this has been covered, in part, in previous sections of this white paper, a sixth reason why you would use a NoSQL database is because you need a more suitable architecture for a particular application. Some, but not all, NoSQL solutions provide modern architectures that can tackle the type of applications that require high degrees of scale, data distribution, and availability.

Legacy master/slave, manually sharded, and shared storage architectures all have pitfalls – from write bottlenecks (master/slave) to high maintenance overhead (manually sharded) to various points of failure (all of the above).

How DataStax Provides a Better Architecture

Rather than using a legacy master/slave or a manual and difficult-to-maintain sharded design, Cassandra has a masterless distributed "ring" architecture that is much more elegant, easy to set up, and maintain.

"Cassandra was just a better design all around – more truly horizontally scalable and with less management overhead – and there's no single point of failure. I looked at Cassandra's architecture and thought, 'Yeah, that's how you do it.'" – Backupify

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 and with each capable of being read and written to.

Cassandra's built-for-scale architecture means it is capable of handling terabytes of information and thousands of concurrent users/operations per second across one to many 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 manually sharded systems, Cassandra has no single point of failure and is therefore capable of offering true continuous availability.

NoSQL Use Cases

Many modern businesses and organizations are using Cassandra for critical applications today. Here are just some examples:

Figure 1: A sample of companies and organizations using Cassandra in production

DataStax supports a wide variety of use cases in DataStax Enterprise. Because DataStax Enterprise provides real-time data management with Cassandra, as well as analytics and enterprise search on Cassandra data all in a single database cluster, it's capable of handling the following application scenarios:

Real-time/Online:

- Time series data
- Device/sensor/data "exhaust" systems
- Distributed applications
- Media streaming
- Online web retail (e.g., transactional, shopping carts)
- Real-time data analytics
- Social media capture and analysis
- Web clickstream analysis
- Write-intensive transactional systems

Analytics:

- Buyer behavior analytics
- Compliance/regulatory analysis
- Customer recommendation output
- Fraud detection
- Risk analysis
- Sales program campaign analysis
- Supply chain analytics
- Batch web clickstream analysis

Enterprise search:

- General web search
- Web retail-faceted (categorization) search
- Search/hit prioritization and highlighting
- Application log search and analysis
- Document (e.g., PDF, MS Word) search and analysis
- Geospatial search
- Real estate location and property search
- Social media matchups

Conclusion

Why might you choose to use a NoSQL solution like Apache Cassandra and DataStax Enterprise? There are myriad reasons why IT professionals might select a NoSQL datastore for an application, but the most common reasons are:

1. You have big data.
2. You need continuous availability for an application.
3. You need location independence for a system.
4. You need modern transaction support.
5. You need a more flexible data model.
6. You need a better architecture.

To find out more about Apache 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. Note that DataStax Enterprise Edition is completely free to use in development environments, while production deployments require a software subscription to be purchased.

About DataStax

DataStax powers the big data applications that transform business for more than 300 customers, including startups and 20 of the Fortune 100. DataStax delivers a massively scalable, flexible and continuously available big data platform built on Apache Cassandra™. DataStax integrates enterprise-ready Cassandra, Apache Hadoop™ for analytics and Apache Solr™ for search across multi-data centers and in the cloud.

Companies such as Adobe, Healthcare Anytime, eBay and Netflix rely on DataStax to transform their businesses. Based in San Mateo, Calif., DataStax is backed by industry-leading investors: Lightspeed Venture Partners, Crosslink Capital and Meritech Capital Partners. For more information, visit DataStax or follow us @DataStax.