ROBERT FRANCES GROUP
Business Advisors to IT Executives

Nov 15, 2013

The Evolution of NoSQL – Part 1

RFG POV: In the ever-evolving world of enterprise IT, choice is generally considered a good thing – albeit having too many choices can create confusion and uncertainty. For those application owners, database administrators and IT directors who pine for the good old days when one could count the number of enterprise-class databases (DBs) on one or two hands, the relational-database-solves-all-our-data-management-requirements days are long gone. NoSQL databases in all their varieties are not going away and IT executives will need to understand the alternatives and select a minimum set that best meets corporate needs.

Thanks to the explosion of Big Data throughout every industry sector and requirements for real-time, predictive and other forms of now indispensable transactions and analytics to drive revenue and business outcomes, today there are more than 50 DBs in a variety of categories that address different aspects of the Big Data conundrum. Welcome to the new normal world of NoSQL – or, Not only Structured Query Language – a term used to designate databases which differ from classic relational databases in some way.

In August, more than 20 NoSQL solution providers and 100-plus experts gathered at the San Jose Convention Center for 2013's version of NoSQL Now! Exhibitors and speakers included familiar names such as Oracle along with a score of venture-backed NoSQL solution providers eager to disseminate their message and demonstrate that the time has come for enterprises of every ilk to adopt innovative database solutions to tackle Big Data challenges. More than a dozen sponsors were interviewed at the event and profiled in this research note.

The Evolution of NoSQL

In the beginning, there was SQL (structured query language). Developed by IBM computer scientists in the 1970s as a special-purpose programming language, SQL was designed to manage data held within a relational database management system (RDBMS). Originally based on relational algebra and tuple relational calculus, SQL consists of a data definition language and a data manipulation language. Subsequently, SQL has become the most widely used database language largely due to the popularity of IBM, Microsoft and Oracle RDBMSs.

NoSQL DBs started to emerge and become enterprise-relevant in the wake of the open-source movement of the late 1990s. Aided by the movement toward Internet-enabled online transaction processing (OLTP), distributed processing leveraging the cloud and the inherent limitations of relational DBs, including lack of horizontal scale, flexibility, availability, findability and high cost, use of NoSQL databases has mushroomed.

Amazon's instantiation of DynamoDB is considered by many as the first large-scale, or web-scale, production NoSQL database. To quote author Joe Brockmeier, who now works for Red Hat, "Amazon's Dynamo paper is the paper that launched a thousand

NoSQL databases." Brockmeier suggests that the "paper inspired, at least in part, Apache Cassandra, Voldemort, Riak and other projects."

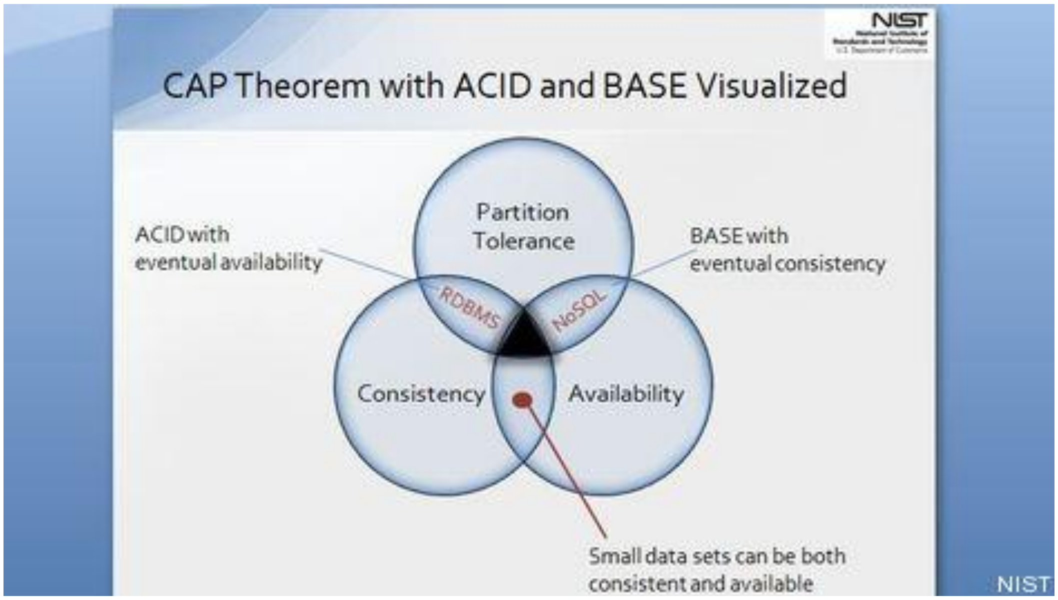
According to Amazon CTO Werner Vogels, who co-authored the paper *entitled Dynamo: Amazon's Highly Available Key-value Store*, "DynamoDB is based on the principles of Dynamo, a progenitor of NoSQL, and brings the power of the cloud to the NoSQL database world. It offers customers high availability, reliability, and incremental scalability, with no limits on dataset size or request throughput for a given table." DynamoDB is the primary DB behind the wildly successful Amazon Web Services business and its shopping cart service that handles over 3 million "checkouts" a day during the peak shopping season.

As a result of the Amazon DynamoDB and other enterprise-class NoSQL database proof points, it is not uncommon for an enterprise IT organization to support multiple NoSQL DBs alongside legacy RDBMSs. Indeed, there are single applications that often deploy two or more NoSQL solutions, e.g., pairing a document-oriented DB with a graph DB for an analytics solution. Perhaps the primary reason for the proliferation of NoSQL DBs is the realization that one database design cannot possibly meet all the requirements of most modern-day enterprises – regardless of the company size or the industry.

The CAP Theorem

In 2000, Berkeley, CA, researcher Eric Brewer published his now foundational CAP Theorem (consistency, availability and partition tolerance) which states that it is impossible for a distributed computer system to simultaneously provide all three CAP guarantees. In May 2012, Brewer clarified some of his positions on the oft-used "two out of three" concept.

- Consistency (all nodes see the same data at the same time)
- Availability (a guarantee that every request receives a response about whether it was successful or failed)
- Partition Tolerance (the system continues to operate despite arbitrary message loss or failure of part of the system).


NIST
CAP Theorem with ACID and BASE Visualized
Partition Tolerance
ACID with eventual availability
BASE with eventual consistency
RDBMS
NoSQL
Consistency
Availability
Small data sets can be both consistent and available
NIST

According to Peter Mell, *a senior computer scientist for the National Institute of Standards and Technology, "*In the database world, they can give you perfect consistency, but that limits your availability or scalability. It's interesting, you are actually allowed to relax the consistency just a little bit, not a lot, to achieve greater scalability. Well, the Big Data vendors took this to a whole new extreme. They just went to the other side of the Venn diagram, and they said we are going to offer amazing availability or scalability, knowing that the data is going to be consistent eventually, usually. That was great for many things."

ACID vs. BASE

In most organizations, upwards of 80% of Big Data is in the form of "unstructured" text or content, including documents, emails, images, instant messages, video and voice clips. RDBMSs were designed to manage "structured" data in manageable fields, rows and columns such as dates, social security numbers, addresses and transaction amounts. ACID (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantees database transactions are processed reliably and is a necessity for financial transactions and other applications where precision is a requirement.

Conversely, most NoSQL DBs tout their schema-less capability, which ostensibly allows for the ingestion of unstructured data without conforming to a traditional RDBMS data format or structure. This works especially well for documents and metadata associated with a variety of unstructured data types as managing text-based objects is not considered a transaction in the traditional sense. BASE (basically available, soft state, eventually consistent) implies the DB will, at some point, classify and index the content to improve the findability of data or information contained in the text or the object.

Increasingly, a number of database cognoscenti believe NoSQL solutions will or have overcome the "ACID test" as availability is said to trump consistency – especially in the vast majority of online transaction use cases. Even Eric Brewer argued recently that bank transactions are BASE not ACID because availability = $.

Conclusion

No one type of NoSQL database neither satisfies all business requirements, nor is it expected that one will any time soon. While the market is still immature and the options are myriad, IT executives cannot wait before selecting the right NoSQL platforms. This research report and the next one will cover the evolution of NoSQL. Three more will follow that cover 21 NoSQL innovators worth exploring.

RFG POV: The NoSQL wave of database technology is immature and expanding and a myriad of options exist to confound IT executives and slow down decision-making. IT executives and data architects should understand the variety of options and then map them to current and future business and technical requirements for each application type where a NoSQL database might apply. As pointed out in the report, no one solution may meet all the requirements so IT executives should be prepared to adopt and standardize on a minimum set of multiple database solutions.

Additional relevant research is available. Interested readers should contact Client Services to arrange further discussion or interview with Mr. Gary MacFadden, Principal Research Analyst.