Will NoSQL Databases Live Up to Their Promise?

Neal Leavitt

Organizations that collect large amounts of unstructured data are increasingly turning to nonrelational databases, now frequently called NoSQL databases.

Many organizations collect vast amounts of customer, scientific, sales, and other data for future analysis. Traditionally, most of these organizations have stored structured data in relational databases for subsequent access and analysis.

However, a growing number of developers and users have begun turning to various types of nonrelational—now frequently called NoSQL—databases.

Nonrelational databases—including hierarchical, graph, and object-oriented databases—have been around since the late 1960s. However, new types of NoSQL databases are being developed. And only now are they beginning to gain market traction.

Different NoSQL databases take different approaches. What they have in common is that they're not relational. Their primary advantage is that, unlike relational databases, they handle unstructured data such as word-processing files, e-mail, multimedia, and social media efficiently.

They are also easier to work with for the many developers not familiar with the structured query language. SQL is the programming language used for querying and updating relational databases.

Some NoSQL databases can function in a distributed setting. Users could thus scale a single database by running it across additional inexpensive machines rather than by having to run it on a single more powerful and costly machine.

Moreover, proponents say, NoSQL databases enable better performance, which is particularly important for applications with large amounts of data.

Numerous companies and organizations have developed NoSQL databases.

The approach's most influential champions are primarily Web 2.0 companies with huge, growing data and infrastructure needs such as Amazon and Google. They developed the Dynamo and Big Table NoSQL databases, respectively, which have inspired many of today's NoSQL applications.

Despite its promise, the approach must clear several technical and marketplace hurdles before achieving widespread success.

IN THE BEGINNING

The late Edgar Codd, a former IBM Fellow, is generally credited with creating the relational-database model in 1970.

A relational database is a set of tables containing data fitted into predefined categories. Each table contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns. Users can access or reassemble the data in different ways without having to reorganize the database tables.

Relational databases work best with structured data—such as a set of sales figures—which readily fits in well-organized tables. This is not the case with unstructured data, such as that found in word-processing documents and images.

Relational database limitations

The structure of data in a relational database is predefined by the layout of the tables and the fixed names and types of the columns.

Scaling. Users can scale a rela-

 Published by the IEEE Computer Society

tional database by running it on a more powerful—and expensive—computer. To scale beyond a certain point, though, it must be distributed across multiple servers.

Relational databases don't work easily in a distributed manner because joining their tables across a distributed system is difficult, said Craigslist software engineer Jeremy Zawodny.

Also, relational databases aren't designed to function with data partitioning, so distributing their functionality is a chore, said Stephen O'Grady, an analyst with market research firm RedMonk.

Complexity. With relational databases, users must convert all data into tables. When the data doesn't fit easily into a table, the database's structure can be complex, difficult, and slow to work with.

SQL. Using SQL is convenient with structured data. However, using the language with other types of information is difficult because it's designed to work with structured, relationally organized databases with fixed table information, explained Stefan Edlich, professor at the Beuth University of Applied Sciences in Berlin.

SQL can entail large amounts of complex code and doesn't work well with modern, agile development, he said.

Large feature set. Relational databases offer a big feature set and data integrity. But NoSQL proponents say database users often don't need all the features, as well as the cost and complexity they add.

INSIDE NOSQL DATABASES

Partly in response to the growing awareness of relational databases' limitations, vendors and users are increasingly turning to NoSQL databases.

One of the key moments in this shift occurred in 2007, when Amazon published a paper that introduced its Dynamo distributed NoSQL system for internal use. Amazon was one of the first major companies to store much of its important corporate data in a nonrelational database.

The technology

There are three popular types of NoSQL databases.

Key-value stores. As the name implies, a key-value store is a system that stores values indexed for retrieval by keys. These systems can hold structured or unstructured data.

Amazon's SimpleDB is a Web service that provides core database functions of information indexing and querying in the cloud. It provides a simple API for storage and access. Users pay only for the services they use.

Uppsala University's Amos II is a research prototype that can function as a standalone database or as a front end to other applications.

Research facility Zuse Institute Berlin and software developer onScale Solutions built Scalaris, a scalable, distributed database that can work with Web 2.0 services.

Column-oriented databases. Rather than store sets of information in a heavily structured table of columns and rows with uniform-sized fields for each record, as is the case with relational databases, column-oriented databases contain one extendable column of closely related data.

Facebook created the high-performance Cassandra to help power its website.

The Apache Software Foundation developed Hbase, a distributed, open source database that emulates Google's Big Table.

Document-based stores. These databases store and organize data as collections of documents, rather than as structured tables with uniform-sized fields for each record. With these databases, users can add any number of fields of any length to a document.

The Apache Software Foundation hosts CouchDB as an open source, scalable database written in Erlang and accessible from any browser.

10gen commercially supports and sponsors the development of MongoDB, an open source document database built for scalability and ease of use.

Basho Technologies' Riak is a distributed, scalable, decentralized, open source database suitable for Web-based applications.

NoSQL databases are starting to gain market traction.

Open source

Most NoSQL databases are open source, reflecting developments in the overall software market.

Disruptive software trends such as NoSQL databases frequently do better in an open source environment, which lets users perform technical evaluations at low cost, said Basho chief technology officer Justin Sheehy.

NOSQL PROS AND CONS

NoSQL databases have numerous advantages and disadvantages.

Advantages

NoSQL databases generally process data faster than relational databases.

Relational databases are usually used by businesses and often for transactions that require great precision. They thus generally subject all data to the same set of ACID (atomicity, consistency, isolation, durability) restraints, said Uppsala University professor Tore Risch.

Atomicity means an update is performed completely or not at all,

and consistency means no part of a transaction will be allowed to break a database's rules, he explained. Isolation means each application runs transactions independently of other applications operating concurrently, and durability means that completed transactions will persist, he added.

Having to perform these restraints on every piece of data makes relational databases slower, Risch noted.

Developers usually don't have their NoSQL databases support ACID, in order to increase performance, he said, but this can cause problems when used for applications that require great precision.

NoSQL databases are also often faster because their data models are simpler, noted Kyle Banker, a software engineer at 10gen. "There's a bit of a trade-off between speed and model complexity, he said, but it's frequently a tradeoff worth making,"

Because they don't have all the technical requirements that relational databases have, proponents say, most major NoSQL systems are flexible enough to better enable developers to use the applications in ways that meet their needs.

Concerns and doubts

NoSQL databases face several challenges.

Overhead and complexity. Because NoSQL databases don't work with SQL, they require manual query programming, which can be fast for simple tasks but time-consuming for others.

In addition, complex query programming for the databases can be difficult, Risch noted.

Reliability. Relational databases natively support ACID, while NoSQL databases don't. NoSQL databases thus don't natively offer the degree of reliability that ACID provides. If users want NoSQL databases to apply ACID restraints to a data set, they must perform additional programming.

Consistency. Because NoSQL databases don't natively support ACID transactions, they also could compromise consistency, unless manual support is provided. Not providing consistency enables better performance and scalability but is a problem for certain types of applications and transactions, such as those involved in banking, Risch said.

Unfamiliarity with the technology. Most organizations are unfamiliar with NoSQL databases and thus may not feel knowledgeable enough to choose one or even to determine that the approach might be better for their purposes, Beuth University's Edlich said.

Limited ecostructure. Unlike commercial relational databases, many open source NoSQL applications don't yet come with customer support or management tools.

During the next five years, according to RedMonk's O'Grady, NoSQL proponents will focus on developing better application compatibility and management tools.

According to Dave Rosenberg, founder of open source infrastructure provider MuleSource and adviser to several technology companies, NoSQL databases will be used largely for working with unstructured data in ways that require scalability.

NoSQL adoption will be small-scale and only in some niches because relational databases are more mature and represent huge investments by vendors and users, said Anant Jhingran, IBM's chief technology officer for information management, analytics, and optimization.

During the next one or two years, O'Grady predicted, users will adopt NoSQL databases primarily for specialized projects, such as those that are distributed, that involve large amounts of data, or that must scale. After that, he said, broader adoption could occur.

NoSQL databases won't replace relational databases, he stated, but instead will become a better option for certain types of projects.

"People will learn to look at their data and select many databases for many needs," said Edlich.

Added Basho's Sheehy, "There will be a growing realization that the relational databases in use today are often good tools but that other tools have their place as well."

__Neal Leavitt__ is president of Leavitt Communications (www.leavcom.com).

Editor: Lee Garber, *Computer*, l.garber@computer.org

Selected CS articles and columns are available for free at http://ComputingNow.computer.org.