



*Carl Olofson*

*Research Vice President, Application Development and Deployment*

## **New Technology Complements Existing Database Management Systems**

*June 2010*

---

*The database management systems (DBMS) markets are on the verge of major changes. These changes are driven by a number of factors such as increased use of clustering and virtualization in a mode of deployment known as cloud computing, a rising tide of open source DBMS usage, and new technologies and usage paradigms for databases. As the database market shifts, users will look to diversify their DBMS portfolios, applying specialized database technologies to different workloads rather than depending on one product or product line for all data management.*

The following questions were posed by NorthScale to Carl Olofson, research vice president of IDC's Application Development and Deployment program, on behalf of NorthScale's customers.

**Q.    How is the data management landscape changing? And what are the requirements for organizing/managing data to address these changes?**

A.    The World Wide Web has brought forth applications that are making very different demands for data management services than had previously existed in computing. First, we had ecommerce sites, requiring greater data structure flexibility, data volume capacity, and concurrent user support than are practical from conventional data management systems. Then, we had social networking sites and search sites, with requirements for even more flexibility and dynamism in terms of the range of data supported, the amount of data managed, and the number of users served. Add to these online gaming, Web crawlers, blogs, and various content aggregation sites, and you have requirements for data flexibility, scalability in terms of both size and users, and performance that cannot be implemented using formerly conventional methods. Because these applications are all interactive, they must deliver subsecond responses to their users, despite the underlying volume and complexity of data operations necessary to respond to a user's request.

Because such forms of data management aren't efficiently or effectively supported by relational databases, it's necessary to look at other technologies. Conventional relational database technologies all require time-consuming and expensive operations to make changes to the size or structure of the database, and most are not effective at dealing with extreme ranges of concurrent access, so they are simply not suitable for dealing with these new data management workloads.

There are a few requirements for managing data in these situations. One is that you need to be able to capture data that may be structured in a way that you didn't necessarily anticipate. Relational databases require that you define in advance what the structure of the data is going to be before you can capture the data.

Also, if you are going to collect data from online — especially semistructured sources such as XML documents and Web pages — you need to be able to capture the data and identify it first and then search through it and process it later. The types of applications we are discussing need to be able to input or load data without predefinition.

This is a big problem for a relational database because it requires that the structure and format of any data loaded into that database conform to a predefined schema. Loading data that you had not previously defined into your current relational database today requires changing its schema, which involves a series of formal steps. When you make a schema modification, you often have to convert the data, rebuild the indexes, and so forth. The ability to load that data into the database without having to go through those formal steps is often needed. If you're required to define the data first, it's a big inhibitor to being able to capture and process it.

Another issue is that it's not always possible to know the amount of storage resource that's going to be required. Essentially, it's difficult to know how big the database is going to be. When you start managing these kinds of unpredictable sets of data, traditional relational database technologies are not adequate because they assume a fixed size database. And if the database grows, you have to perform a series of administrative steps in order to expand the database; there just isn't enough flexibility with a relational DBMS (RDBMS) in managing a database of varying size. It's necessary to look for a technology that can actually expand as needed without requiring the formal processes of identifying and formatting storage, defining storage to the database, changing the database definitions to incorporate the new storage, and so on.

A third concern has to do with delivering good performance to very large numbers of concurrent users. Some relational databases are deployed in environments that enable them to process hundreds of thousands, sometimes even millions of users. But they do it by running on very large, fixed configurations of very expensive hardware. They are tuned for the "high-water mark" of usage, so if high usage occurs only from time to time, it doesn't matter, because they are inflexible; they can't adjust. The way to deliver good performance against a variable level of demand that can rise to extreme levels, and one that has been used by Web applications for years, is to deploy those applications in a clustered environment consisting of many servers with virtualized resources so the applications can expand and contract resource utilization as the demand requires. The database system needs to do the same, but an RDBMS can't do that.

**Q. How significant is the shift toward a new class of data in terms of volume, amount, and cost? How real is the need for nonschematic database technologies (or AltDB or NoSQL solutions)?**

A. In addition to the Web applications mentioned previously, we are seeing a need in many enterprises for a richer set of information to drive decisions at both a strategic level and a tactical level. Most larger businesses have fairly comprehensive data warehouses that capture data for trend analysis and visualization based on fixed data models and mathematical aggregation functions, but those activities answer the "what is happening in my business" questions. To get to the "why is it happening," we need richer information. We need to collect, sift, order, and refine information gleaned from documents, online research, emails, and ad hoc assortments of collected data from different sources. Because we don't

necessarily know what we are looking for, we need to collect huge amounts of data to sift through, and we need to deliver analytic functions in a timely manner to enrich both business analytics and operational business intelligence. These things are driving the need for managing a new class of data in a highly scalable way.

The shift toward a new class of data is actually enabled by a dramatic change in the costs associated with hardware. During the years when the relational database technologies were being developed, servers typically had one or two processors and 32-bit addressability, which limited the amount of memory to approximately the megabyte range. The memory was online and expensive. So those relational database systems based their management of the data on the way it would be laid out onto spinning disks, optimized to reduce I/O waits and disk head contention. As memory became cheaper, techniques were used to map the data between the spinning disk and memory caches or memory buffers, but the fundamentals remained the same.

In the 1980s and 1990s, this was actually a very clever way to enable the systems to manage very large amounts of data with reasonably good efficiency. However, now the economics have changed. Memory is cheap, and we have 64-bit addressability, which means that on a typical server, the practical limit of available main memory for database management has grown to about a terabyte of data.

It's now possible to manage most of your data in memory. In addition, we have techniques for clustering servers together so they can work together, spreading the database across all their main memory to create almost limitless capacity. As a result, you can create a very flexible environment for sharing data across servers, yielding very large databases that you can modify and expand dynamically. This has created a sea change in terms of what can be done with database technology. A wide variety of ways that you can manage data in main memory are now practical due to inexpensive processors with multiple cores. This is revolutionary because it opens up whole new approaches to data management.

New nonschematic database technology has emerged to take advantage of these advances. The technology allows you to collect data and analyze it or to collect data in an ad hoc manner and organize it programmatically. In the past, this wasn't practical because it would've been too expensive in terms of memory, processing power, I/O activity, and so forth.

**Q. Does the RDBMS go away?**

A. Absolutely not. Relational database management systems serve specific purposes, which will continue to exist. Companies still have to do accounting or track sales. They still have to maintain their chart of accounts and inventories. Companies have to do all those traditional things that computer applications have been doing for decades and will continue to be doing for decades.

Relational database systems are exquisitely optimized to perform those fixed repeatable transaction processes so that you can do them with maximum efficiency and throughput. The new technologies that we're talking about here are expanding the range of data management. The new technologies are not replacing the relational database technology — they are augmenting what you already have with new capabilities in order to deal with new kinds of data and new kinds of workloads. In effect, the new technologies do things that the relational database technologies can't handle.

**Q. Are we talking about evolution or revolution? Will these new DBMSs sweep away the schematic DBMSs, or can (and should) they coexist somehow?**

A. There will still be a need for transactional databases and data warehouses. What will happen is that we'll have new databases that solve new problems centered on collecting data and crunching it for different purposes, such as doing more effective searching, managing social media type applications, blending multimedia content together with some of the data that you're keeping in the relational database. Nonschematic databases can greatly expand the functionality of operational applications by adding support for sizes and types of data that can't be supported today and by doing so in a highly scalable manner. So relational databases and nonschematic databases will coexist. Some practical examples are immediately apparent. For instance, nonschematic databases could be used to load and analyze data as a preliminary step to building or modifying a data warehouse that is housed on a relational database.

The two technologies will complement each other. It's entirely possible that at some point in time they might be joined together technically in some way. However, it's not a case of nonrelational databases eclipsing relational databases. Instead the process will be evolutionary.

**Q. What are the best target applications for these scale-out data technologies?**

A. The best target applications fall into several broad categories. One category is the area of applications that involve collecting data of many different data types in a fairly dynamic way such as social media applications or some of the emerging Web-based applications around information search and discovery.

There are also those applications such as media content that augment traditional applications with enriched presentations and involve the crunching of unstructured data. Ad hoc applications are another category. Basically these are things such as cloud-based applications that are developed by individuals or groups on cloud-based application development platforms from time to time and require a bunch of data to be pulled into a space where it is worked on. Developing these applications is a project of limited duration where the structure and organization of the data is sufficiently simple and well understood so that it's not necessary to have to go through formal steps such as defining a schema, creating a database, loading the database, and so on.

There is a new generation of cloud-based applications that involve creating services that require a large pool of shared data to operate. These applications are very dynamic and require a changeable and dynamic data space in which to operate. There are caching technologies that enable such data sharing, but they are not persistent; if there is an outage, for example, the data gets lost. So what's needed is technology that can act as a persistent data store for that kind of data so that it is reliably available when applications need it. New database technologies can work in this way.

Another category of applications has to do with collecting data that exists in the enterprise and pulling it into one place in order to better understand it. Typically, the data is sifted through, defined, and then potentially used in the nonschematic database as a complement to the relational database. The nonschematic database can help identify and define more correctly and completely what the schema is so that you can then apply the schema to your relational database.

## ABOUT THIS ANALYST

*Carl Olofson performs research and analysis for IDC's Information Management and Data Integration Software service within the Application Development and Deployment research group. Mr. Olofson's research involves following sales and technical developments in the information and data management (IDM) markets and database management systems (DBMS) markets; data movement and replication software; data management software; metadata management software; and vendors of related tools and software systems.*

---

## ABOUT THIS PUBLICATION

This publication was produced by IDC Go-to-Market Services. The opinion, analysis, and research results presented herein are drawn from more detailed research and analysis independently conducted and published by IDC, unless specific vendor sponsorship is noted. IDC Go-to-Market Services makes IDC content available in a wide range of formats for distribution by various companies. A license to distribute IDC content does not imply endorsement of or opinion about the licensee.

## COPYRIGHT AND RESTRICTIONS

Any IDC information or reference to IDC that is to be used in advertising, press releases, or promotional materials requires prior written approval from IDC. For permission requests, contact the GMS information line at 508-988-7610 or [gms@idc.com](mailto:gms@idc.com). Translation and/or localization of this document requires an additional license from IDC.

For more information on IDC, visit [www.idc.com](http://www.idc.com). For more information on IDC GMS, visit [www.idc.com/gms](http://www.idc.com/gms).

Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 [www.idc.com](http://www.idc.com)