NoSQL Database Technologies

Michael Madison
Mark Barnhill
Cassie Napier
Joy Godin
Georgia College
USA

ABSTRACT

As cloud computing continues to evolve, organizations are finding new ways to store the massive amounts of big data that are collected. Big data storage often require greater flexibility and scalability which can be provided by incorporating NoSQL technologies. NoSQL (Not Only SQL) is quickly becoming a popular approach to store large and unstructured data. This paper looks at the various classifications of NoSQL technologies as well as many of the notable characteristics of the technologies. The authors also describe some deficiencies of using NoSQL and give some explanation to why companies are adopting the technology. The paper concludes with suggestions for future research of NoSQL technologies and a content analysis of current articles in database management is provided in the appendix.

Keywords: big data, cloud computing, document databases, MongoDB, NoSQL, Not Only SQL, unstructured database

INTRODUCTION

Since NoSQL database technologies entered the market in 1998, these technologies have challenged the widely-accepted relational structures and database management practices supported by traditional SQL databases. Despite this trend, the perception that NoSQL seeks to eliminate the use of relational databases is a common misconception. NoSQL is best considered with the acronym "NOSQL" - Not Only SQL - which more accurately represents an approach that combines non-relational databases with the use of relational ones. This approach seeks to leverage both NoSQL and SQL technologies in order to balance the demands of performance, scalability, and schema flexibility with data integrity and consistency (Purdue, 2014; Penchikala, 2013).

The rise of "web-scale" applications creates performance, scalability, and schema flexibility demands that present significant challenges for traditional SQL / relational databases. Thus, a variety of new entrants have entered the market, each seeking to address specific challenges, including (Kent, 2012):

- The need to store large volumes of data about users, objects and products
- The need to provide near-real-time transaction processing
- The need to provide high-speed analytics on large volumes of data (big data)
- The need to support flexible data schemas for evolving business environments
- The need to scale applications horizontally with predictable hardware and licensing cost structures

  ISSN: 1543-5962-Printed Copy ISSN: 1941-6679-On-line Copy

Each NoSQL solution seeks to address one or more of the above issues, often by providing alternative approaches to long-established relational database norms. Traditional relational databases were not designed to cope with the scalability and flexibility challenges facing modern applications. Older databases were not designed to take advantage of the inexpensive storage and excess processing power available today.

The defining characteristics of NoSQL technologies, as well as the relative advantages and disadvantages of NoSQL, and the subsequent influence on adoption of NoSQL technologies are each the subject of a small, yet growing body of research. This paper provides a literature review of that research and proposes potential areas where future research may focus.

REVIEW OF LITERATURE

NoSQL database technology is a relatively recent innovation and much of the research surrounding it is of an emerging nature. For the purpose of this literature review Galileo was primarily used for identifying the articles. Articles were selected that focused directly on NoSQL database technologies (unstructured / semi-structured databases, document databases, etc.) and articles that focused on relational databases. The contrasting articles provided the researchers with the ability to compare the features that are offered with a greater degree of variability in NoSQL databases (for example, an article on SQL scalability provided a foundation for comparing newer horizontal scaling techniques used in certain NoSQL databases). Some of the search terms used included: NoSQL, relational databases, non-relational databases, semi-structured databases, unstructured databases, document databases, big data, business intelligence, data warehouse, OLAP, OLTP, index optimization, MongoDB, database consistency, eventual consistency, database scalability, and NoSQL adoption. After the articles were reviewed, several focus areas were identified across the various articles and the articles were categorized accordingly (see Appendix A in the paper).

Generally, the literature agrees that NoSQL databases possess greater flexibility and scalability than traditional SQL databases but at the expense of functions that are taken for granted in relational technologies. Consequently, much of the research surrounding NoSQL concerns the classification of NoSQL technologies, the relative advantages and disadvantages of each category, how shortcomings may be remedied (especially with regards to consistency, querying and interoperability), and the adoption of NoSQL technologies.

Classification of NoSQL Database Technologies

Research of NoSQL database technologies reveals many varieties of NoSQL databases, each tailored to serve specific functions, with many competing vendors and systems in each segment of the market. Broadly, these varieties of NoSQL technologies include key-value stores, document databases, wide-column stores and graph databases (Meijer & Bierman, 2011; Moniruzzaman & Hossain, 2013; Stonebraker, 2010).

Document databases use key-value pairs tied to documents with formats including XML (eXtensible Markup Language), JSON (JavaScript Object Notation) and BSON (Binary JSON) (Moniruzzaman & Hossain, 2013). Values are stored in these semi-structured data formats and each document may contain hundreds of attributes with varying data types. Examples of document databases include CouchDB and MongoDB (Stonebraker, 2010).

Key-value stores pair alphanumeric keys with associated values in standalone hash tables. These are useful for high-speed, scalable value retrievals. Examples of key-value stores include MemcacheDB and DynamoDB (Stonebraker, 2010).

Wide-column stores are useful for distributed storage of very large quantities of data (Moniruzzaman & Hossain, 2013). These systems provide time-stamping functions that make them especially useful for managing versioned data. These systems also have applications for predictive analytics.

Finally, graph databases use structured, interconnected graphs to connect data rather than tables. Graph databases are best-suited for analyzing relationships between data rather than the data itself (Moniruzzaman & Hossain, 2013).

Notable Characteristics of NoSQL Technologies

As a variety of NoSQL categories exist, there is significant feature variability both across and within categories. This variance generally occurs with respect to each vendor’s implementation of database consistency and transaction management guarantees, query language implementation, and other strategies to improve database performance. In the midst of this variance, the common distinguishing characteristic differentiating NoSQL technologies from their SQL counterparts is the departure from traditional relational schemas.

Consistency

Certain NoSQL database technologies sacrifice ACID (atomicity, consistency, isolation and durability) transaction guarantees, which are considered the norm in relational SQL database technologies. This sacrifice is made in favor of perceived performance gains (Moniruzzaman & Hossain, 2013) and is reflected in CAP theorem. CAP theorem addresses the tension between the competing demands that NoSQL systems seek to reconcile: strong Consistency, high Availability and Partition-tolerance. NoSQL systems tend to relax requirements on consistency in order to maximize availability and partition tolerance, thus settling for eventual consistency rather than strong consistency. This relaxation of ACID requirements has led to the establishment of less-restrictive BASE standards: Basically-Available, Soft-state, Eventually-consistent.

Notably, BASE standards are viewed with doubt by some researchers and practitioners. Stonebraker (2010) writes that the performance gains provided by NoSQL are modest at best and fail to justify sacrificing ACID transactions. Stonebraker (2010) writes that the proper way to increase performance is to eliminate the overhead associated with processing ACID transactions, including logging, locking, and latching and buffer management.

Querying

Complicating the effective use of NoSQL is that NoSQL systems require specialized knowledge to install, integrate, and support. Knowledge of one NoSQL product may not transfer to another. Traditional SQL database expertise is comparatively less useful than object-oriented programming skills when implementing NoSQL databases. Meijer and Bierman (2011) predict that before the NoSQL market can mature the NoSQL market will need to develop a common approach to data modeling and a common query language in order to reduce differentiation between the NoSQL products offered by different vendors.

NoSQL query languages are emerging, but they tend to be limited to particular NoSQL technologies, such as Cassandra's CQL and CouchDB's unQL. Curé, Lamolle, and Duc (2013) propose a common query language for NoSQL databases. Curé, Lamolle, and Duc (2013) devise a method which computes ontology from the structure and instances of database sources in order to create a global ontology supporting the use of SPARQL queries. So far, the authors' work supports Cassandra and MongoDB, and their future work seeks to provide support for graph databases as well.

In addition, Ong, Papakonstantinou, and Vernoux (2014) propose a common query language for NoSQL, SQL++, with the ambitious aim of supporting NewSQL while simultaneously providing backward compatibility with traditional SQL. The goal of this effort is to create a query language that makes it possible for NoSQL systems to communicate with one another and with traditional SQL systems. The authors present FORWARD as their proof-of-concept, a virtual database query processor able to execute SQL++ queries over SQL and NoSQL databases.

Performance

Despite the common perception that NoSQL systems consistently boast higher performance than relational systems, there does appear to be evidence to challenge that belief in some cases. Studies have compared the performances of SQL and NoSQL databases and have found that the SQL systems in their studies outperformed the NoSQL systems (Floratou et al., 2012). In one test, the parallel database system (PDW) outperformed the MapReduce data warehouse Hive by a factor of nine when using the Transaction Processing Performance Council's TPC-H benchmark on a 16TB scale. At the same time, SQL-CS was found to have higher throughput and lower latency than MongoDB during tests, despite the fact that both systems were tested with the same numbers of clients.

The results of the aforementioned study (Floratou et al., 2012) were found to be due to the more mature, optimized and efficient nature of SQL applications. The results suggested that additional and repeated research into the relative performances of a wider array of applications may be valuable, particularly as NoSQL technologies mature. Finally, while tests found SQL systems to have greater performance, the authors acknowledged that NoSQL systems possess an edge over SQL systems in terms of flexibility and sharding support. Sharding allows NoSQL systems to distribute data across multiple sites; this capability is built into many NoSQL technologies (Stonebraker, 2010).

Addressing NoSQL's Deficiencies

In multiple articles, Stonebraker (2010, 2012) questions NoSQL's ability to satisfy the rising need for large-scale online transaction processing (OLTP) applications, such as those demanded by online games, gambling sites and social networks. Stonebraker (2010) argues that NoSQL is not suited for OLTP due to the need for ACID guarantees to ensure data accuracy and integrity. The lack of native ACID guarantees within the NoSQL database places greater demand on the application layer and makes querying difficult. As an alternative to NoSQL, Stonebraker (2012) suggests that NewSQL systems are appropriate for responding to the needs of OLTP due to NewSQL's ability to preserve SQL ACID guarantees while continuing to offer high performance and scalability.

Another study investigated avenues to improve NoSQL's performance and address the relative immaturity of NoSQL scalability features (Thomson, et al., 2014). "Calvin" is a "transaction scheduling and data replication layer" that allows clusters of low-cost commodity machines to preserve their scalability while also maintaining transactional guarantees, even at strong consistency levels. Calvin provides a layer of ACID-compliant, transactional support on top of a NoSQL database.

A study by Ghosh (2010) encourages a multi-paradigm approach to database design. Ghosh recognizes that there are a large variety of NoSQL technologies, with varying strengths and weaknesses, and advocates using an effective combination of technologies in order to get the best of their strengths while offsetting their weaknesses. Ghosh further encourages the use of relational databases in order to support NoSQL systems while providing additional functions, including report generation and audit support.

As a means for unifying the different technologies, Ghost (2010) recommends a system of asynchronous messaging, which would achieve eventual consistency between the technologies. However, the author recognizes that there are barriers to multi-paradigm design, including the rapidly evolving nature of NoSQL technologies, which could cause compatibility issues, and difficulties that could arise in syncing NoSQL and relational databases if the relational database enforces ACID transaction standards with which NoSQL is unable to comply.

NoSQL Adoption

NoSQL systems tend to be adopted by companies after traditional SQL systems fall short of their requirements for performance, flexibility or scalability (Stonebraker, 2010, Moniruzzaman & Hossain, 2013). Social networking, big data and business intelligence applications, for instance, all pushed traditional relational databases to their limits, thus helping to spur the creation of non-relational, horizontally scalable, distributed databases. This trend is made evident by companies that have adopted NoSQL technologies, including Facebook, Twitter, Digg, Amazon, LinkedIn and Google. Nearly fifty percent of respondents to a 2012 Couchbase survey cited the inflexibility of relational systems as the reason for adopting NoSQL systems, while 35 percent cited scalability, and 29 percent cited performance (Moniruzzaman & Hossain, 2013).

Business intelligence, traditionally based on relational online analytical processing (OLAP) databases, provides a case for the move away from relational systems. Muntean and Surcel (2013) suggest a transition from the traditional, relational-based business intelligence model to a new paradigm of agile business intelligence which, like NoSQL, will be able to utilize semi-structured data in addition to structured data. In addition, this new paradigm calls for the adoption of in-memory technology (rather than disk-based technology) in order to drastically increase processing speeds (Muntean & Surcel, 2013).

NoSQL may also impact mobile device management, which is currently underpinned either by relational databases or proprietary software. NoSQL technologies could stand to make a contribution in this field due to their scalable nature and flexible schemas (Fotache & Cogean, 2013).

CONCLUSIONS

As evident in the preceding sections, the literature review identified NoSQL research focusing in the following areas:

- The definition and classification of NoSQL and its sub-types (key-value stores, document databases, wide-column stores, and graph databases)
- Characteristics of NoSQL technologies and their impact on database consistency, query-ability, and performance
- Specific performance issues or deficiencies of NoSQL database technologies

While there is a general perception that NoSQL database technologies are a panacea for improved performance, there are findings to question that assumption. What the literature generally supports is that the market is growing with a variety of new database tools, each positioned to solve specific problems. This trend provides IT organizations with greater flexibility and choice than that which existed in the past when relational databases were the sole option. It is now the opportunity and challenge to IT organizations to better understand the tools available at their disposal in order to select the best mix of technologies to support their specific business and technical environments.

FUTURE RESEARCH

While the body of research concerning NoSQL database technologies continues to grow, specific opportunities exist for improving both academic and practical understanding of NoSQL technologies, including:

- Comparative analyses of NoSQL technologies, by NoSQL classification and across specific vendors
- Additional performance and scalability analyses comparing NoSQL databases with SQL databases
- Cost / pricing comparisons for NoSQL vendors (and/or compared with SQL vendors)
- Proposals for establishing consistent best practices in NoSQL data / document models
- Study of security management capabilities, issues, and opportunities in NoSQL technologies

- Case studies adoption of NoSQL databases in specific use cases and business domains

Additional studies in each of these areas could provide IT organizations with the ability to make a more informed choice in their selection of database technologies.

In the course of the literature review, numerous research sources were reviewed concerning database management, of which only a fraction of the research dealt with NoSQL specifically (that fraction is the specific focus of this paper). Future expansion of this paper may more completely address the findings of that more comprehensive review, though our raw content analysis of sixty-two articles is included in Appendix A. That content analysis sought to identify the key points of focus of each article; the analysis resulted in ten categories for classifying the primary focus area(s) of each article.

REFERENCES

Curé, O., Lamolle, M., & Duc, C. L. (2013). Ontology based data integration over document and column family oriented NOSQL.

Floratou, A., Teletia, N., Dewitt, D. J., Patel, J. M., & Zhang, D. (2012). Can the elephants handle the NoSQL onslaught? *Journal Proceedings of the VLDB Endowment, 5*(*12*), 1712-1723,

Fotache, M., & Cogean, D. (2013). NoSQL and SQL databases for mobile applications: Case study: MongoDB versus PostgreSQL. *Informatica Economica*, *17*(2), 41-58. doi:10.12948/issn14531305/17.2.2013.04

Ghosh, D. (2010). Multiparadigm data storage for enterprise applications. *IEEE Software*, *27*(5), 57. doi:10.1109/MS.2010.87

Kent, A. (2012). NoSQL adoption is on the rise. Retrieved from JAXenter: http://jaxenter.com/nosql-adoption-is-on-the-rise-40811.html

Meijer, E., & Bierman, G. (2011). A Co-relational model of data for large shared data banks. *Communications of the ACM*,*54*(4), 49. doi:10.1145/1924421.1924436

Moniruzzaman, A. M., & Hossain, S. A. (2013). NoSQL database: new era of databases for big data analytics – classification, characteristics and comparison.

Muntean, M., & Surcel, T. (2013). Agile BI - the future of BI. *Informatica Economica*, *17*(3), 114-124. doi:10.12948/issn14531305/17.3.2013.10

Stonebraker, M. (2012). New opportunities for new SQL. *Communications of the ACM*, *55*(11), 10-11. doi:10.1145/2366316.2366319

Ong, K. W., Papakonstantinou, Y., & Vernoux, R. (2014). The SQL++ semi-structured data model and query language: a capabilities survey of SQL-on-Hadoop, NoSQL and NewSQL databases, *arXiv preprint arXiv:1405.3631*.

Penchikala, S. (2013, July 23). NoSQL database adoption trends. Retrieved from Infoq.com: http://www.infoq.com/research/nosql-databases

Perdue, T. (2014, October 14). An overview of NoSQL. Retrieved from newtech.about.com: http://newtech.about.com/od/databasemanagement/a/Nosql.htm

Stonebraker, M. (2010). SQL databases v. NoSQL databases. *Communications of the ACM, 53*(4), 10-11. doi:10.1145/1721654.1721659

Thomas, A., Diamond, T., Shu-Chun, W., Kun, R., Philip, S., & Abadi, D. J. (2014). Fast distributed transactions and strongly consistent replication for OLTP database systems. *ACM Transactions on Database Systems, 39*(2), 11:1. doi:10.1145/2556685

Appendix A - Content Analysis of Database Management Research Articles

Not all of the articles listed below emphasize NoSQL database technologies. The table below provides the raw data for a content analysis of peer-reviewed database management articles that were identified through a series of searches using the Galileo research database. A "1" indicates that the article's focus emphasizes that topical category, where a "0" indicates that the article's focus does not include that topic. Future expansion of this paper may more fully address the findings of this raw data (and would include complete source citations for each article; only NoSQL-focused articles directly cited in this paper are included in this paper's "References" section).

Article	Comparative / Overview	Storage & Infrastructure	Scalability	Performance	Consistency	Data Modeling	Security	Integration	Analytical Applications	Business Issues (including Adoption)
Scalable SQL	0	0	1	1	1	0	0	0	0	0
Eventually Consistent: Not What You Were Expecting?	0	1	0	1	1	0	0	0	0	0
Modeling temporal dimensions of semistructured data	0	0	0	0	0	0	0	0	0	0
Unstructured Data	0	0	0	0	0	1	0	0	0	1
Determinants of the Use of Relational and Nonrelational Information Sources	0	0	0	0	0	0	0	0	0	0
HARNESSING BIG DATA VOLUMES	0	0	0	0	0	0	1	0	1	1
In-memory analytics - strategies for real-time CRM	0	1	0	1	0	0	0	0	1	1
NoSQL Database: New Era of Databases for Big data Analytics - Classificatio...	1	0	0	0	0	0	0	0	1	1
Smart Data Web Services	0	0	0	1	0	0	0	1	0	0
Puzzling out Big Data	0	0	1	1	0	0	0	1	1	1
BIG DATA PROCESSING: BIG CHALLENGES AND OPPORTUNITIES	0	1	1	1	0	0	0	0	1	0
Agile BI - The Future of BI	0	1	0	0	0	1	0	0	1	1

ISSN: 1543-5962-Printed Copy ISSN: 1941-6679-On-line Copy

A Comparison of Data Warehousing Methodologies	1	0	0	0	0	1	0	1	1	1
Optimizing Index Deployment Order for Evolving OLAP (Extended Version)	0	0	0	1	0	0	0	0	0	0
Data Warehousing and OLAP	0	0	0	0	0	0	0	0	0	0
Application of the Olap Cubes Technology in the Data Mining	1	0	0	0	0	1	0	0	0	0
Compacting Transactional Data in Hybrid OLTP	0	0	0	1	0	0	0	0	0	0
Multidimensional (OLAP) Analysis for Designing Dynamic Learning Strategy	0	0	0	0	0	1	0	0	1	0
Forecasting with In-Memory Technology	0	0	0	0	0	0	0	0	1	0
Towards the Next Generation of Data Warehouse Personalization System: A Sur...	0	0	0	0	0	0	0	0	1	0
A MODEL FOR OBJECT RELATIONAL OLAP	0	0	0	0	0	1	0	0	0	0
OVERVIEW OF ORACLE OLAP AND USING SQL FOR MANIPULATE MULTIDIMENSION AL DATA	1	0	0	0	0	0	0	0	0	0
AN OVERVIEW OF RELATIONAL DATABASES AND OLAP TECHNOLOGY	1	0	0	0	0	0	0	0	0	0
Emerging technologies to speed information access	1	0	0	1	0	0	0	0	0	0
Capacity and Performance Analysis of Distributed Enterprise Systems	0	0	0	0	0	0	0	0	0	0

Hybrid Analytic Flows - the Case for Optimization	0	1	0	1	0	0	0	1	0	0
Nearest surrounder searching in mobile computing environments	0	0	0	1	0	0	0	0	1	0
Reducing the size of databases for multirelational classification: a subgrade	0	0	0	1	0	1	0	0	0	0
SQL Databases v. NoSQL Databases	1	0	0	0	0	0	0	0	0	0
ANALYZING THE PERFORMANCE OF DATA LOADING AND PROCESSING IN RELATIONAL DATA...	0	0	0	0	0	0	0	0	0	0
The Pathologies of Big Data	1	0	1	1	0	0	0	0	0	0
Continuous Online Index Tuning in Moving Object Databases	0	0	0	1	0	0	0	0	0	0
A workload-driven approach to database query processing in the cloud	0	0	0	1	0	0	0	0	0	0
Experiment- Driven Quality Assessment and Optimization of Cloud-Based Distr...	0	0	1	0	0	0	0	0	0	0
Using a relational database for scalable XML search	0	0	0	1	0	1	0	0	0	0
OPTIMIZATION MODEL FOR WEB APPLICATIONS DATABASES	0	0	0	1	0	0	0	0	0	0
Solving Big Data Challenges for Enterprise Application Performance Management	1	0	1	1	0	0	0	0	0	0
Load Balancing Content-Based	0	0	0	1	0	0	0	0	0	0

Publish/Subscribe Systems										
Distributed Data Management Using MapReduce	1	0	1	1	0	0	0	0	0	0
Fast Distributed Transactions and Strongly Consistent Replication for OLTP ...	1	0	1	1	1	0	0	0	0	0
On the Scalability of Multidimensional Databases	0	0	0	1	0	0	0	0	0	0
Scalability of Data Binding in ASP.NET Web Applications	0	0	0	1	0	0	0	0	0	0
New Opportunities for New SQL	1	0	0	0	0	0	0	0	0	0
20 Obstacles to Scalability	0	0	1	1	0	0	0	0	0	0
10 Rules for Scalable Performance in 'Simple Operation' Datastores	0	0	1	1	0	0	0	0	0	0
An Empirical Assessment of Performance and Scalability of Decentralized Dis...	0	1	0	1	0	0	0	0	0	0
A Normal Form for XML Documents	0	0	0	0	0	0	0	0	0	0
Incremental Entity Resolution from Linked Documents	0	0	0	0	0	1	0	0	0	0
The Impact of XML Databases Normalization on Design and Usability of Intern...	0	0	0	0	0	1	0	0	0	0
Unstructured Social Networks Data for Business Context Analysis	0	0	0	0	0	1	0	0	1	0
Multiparadigm Data Storage for Enterprise Applications	0	0	0	0	0	1	0	1	0	0
Big Data Drives Cloud Adoption in Enterprise	0	0	0	0	0	0	0	0	0	1

A Co-Relational Model of Data for Large Shared Data Banks	1	0	0	0	0	0	0	0	0	0
Big data learning resources integration and processing in cloud environment...	0	0	0	0	0	0	0	0	1	0
SQL Databases v. NoSQL Databases	1	0	0	0	0	0	0	0	0	0
NoSQL Databases	1	0	1	1	0	0	0	0	0	0
Can the Elephants Handle the NoSQL Onslaught?	1	0	1	1	0	0	0	0	0	0
The SQL Semi-structured Data Model and Query Language: A Capabilities Sur...	1	0	0	0	0	0	0	0	0	0
Ontology Based Data Integration Over Document and Column Family Oriented NO...	0	0	0	0	0	1	0	0	0	0
A novel clustered MongoDB-based storage system for unstructured data with h...	0	1	1	1	0	0	0	0	0	0
NoSQL and SQL Databases for Mobile Applications. Case Study: MongoDB versus...	1	1	0	0	0	0	0	0	0	0

About the Authors

Volume 24, Number 1, 2015

NoSQL Database Technologies

Mike Madison is a Senior Product Manager for SAI Global and co-founder of Neetli, LLC. He has designed and implemented enterprise software applications for Fortune 500 customers spanning a variety of industries including financial services, insurance, telecommunications, and healthcare. Mike holds a BBA in MIS and a MBA from Georgia College and will complete Georgia College's MMIS program in Summer 2015. Mike's interests include cloud computing, data analytics and visualization, database technologies, enterprise software architecture, and SaaS product management.

Mark Barnhill is a Georgia College graduate student and will graduate in May 2015 with a Masters in Management Information Systems.

Cassie Napier is a Program Coordinator for Georgia College Continuing Education. Cassie is currently a Masters of Management Information Systems student at Georgia College.

Dr. Joy Godin is a Management Information Systems faculty member in the J. Whitney Bunting College of Business at Georgia College & State University in Milledgeville, Georgia. She teaches both undergraduate and graduate students various courses including Database Management, Programming I and II, Information Technology for Managers and Principles of Information Systems. Dr. Godin's research interests include virtual teams, collaboration technologies, technology acceptance, online learning, project-based learning, database technologies, and cloud computing.

An Innovative Clustering Approach to Market Segmentation for Improved Price Prediction

Donghui Shi is a Professor in the Department of Computer Engineering, School of Electronics and Information Engineering, Anhui Jianzhu University, China. He received his Ph.D. in Computer Science from the University of Science and Technology of China. He is the Prometeo Researcher in Universidad Técnica Particular de Loja, Ecuador. He was a visiting scholar at the Department of Computer Information Systems, the College of Business, University of Louisville. His research interests are in machine learning and data mining, especially in adaptive neuro-fuzzy inference systems, outlier mining, and text mining. His recent work applies data mining algorithms to solve practical problems in real estate, biomedicine, power load forecasting, and aviation safety. His other research interests are management information systems, hospital information systems, and geographic information systems.

Jian Guan is an Associate Professor in the Department of Computer Information Systems of the College of Business, University of Louisville. He earned his Ph.D. degree from the University of Louisville. His research interests include accounting information systems, data mining, and