



# Using DB2 pureXML and BladeCenter JS43 Server for high-performance transaction processing

*Kevin Xie, Jasmi Thekveli, Peter Shum  
IBM Software Group*

*DB2 Performance Benchmarks and Solutions Development*

*Bob Taraba  
IBM Systems and Technology Group  
Independent Software Vendor (ISV) Enablement*

*August 2009*



## Table of contents

<b>Executive summary</b> .....	<b>1</b>
<b>Introduction</b> .....	<b>2</b>
TPoX benchmark .....	2
IBM BladeCenter JS23 and JS43 .....	5
<b>TPoX results: DB2 pureXML with BladeCenter JS43</b> .....	<b>6</b>
System under test (SUT) .....	6
Transaction throughput of mixed TPoX workload.....	7
XML insert performance .....	8
Storage consumption and compression results.....	9
<b>Conclusion</b> .....	<b>10</b>
<b>Resources</b> .....	<b>11</b>
<b>About the authors</b> .....	<b>12</b>
<b>Acknowledgments</b> .....	<b>12</b>
<b>Trademarks and special notices</b> .....	<b>13</b>



## Executive summary

---

*XML has become the standard for data exchange, service-oriented architectures (SOAs), message-based transaction processing and much more. With the introduction of IBM DB2 pureXML in DB2 9, benefits include superior XML data-management capabilities, including XML storage, indexing, querying, updating and optional schema validation. This paper describes how the latest version of DB2, namely DB2 9.7, when combined with an IBM BladeCenter JS43 server, delivers outstanding XML transaction-processing results. The tests include XML document-insertion tests and multiuser-throughput tests — with various numbers of users over 1 TB of raw XML data. Traditionally, BladeCenter servers are well-known for their ease-of-use, energy efficiency, scalability and manageability, but often overlooked when selecting database servers. However, the latest BladeCenter JS23 and JS43 blades are built with the additional goal of being premier blades for 64-bit applications. The test results documented here prove that the BladeCenter JS43 blade server can be a highly scalable database server.*

*The transaction throughput and processor time were scaled well as the number of users increased. The highest XML transaction processing rate of 4107 XML transactions per second (246 420 transactions per minute, 14.7 million transactions per hour) was obtained with 300 concurrent users without any think-time. The average response time was just 0.07 seconds. XML documents were also inserted into the database at impressive speeds. For the 1 to 2 KB order XML documents, the insertion rate was 50 million documents per hour; for the 2 to 23 KB Customer Accounts XML documents, the insertion rate was 18.2 million documents per hour. The compression capabilities of DB2, as well as its XML inlining, also aided in obtaining these results by reducing the total database size by 58%. These results show that the JS43 blade server is a highly scalable database server, that DB2 pureXML is a high-performing database, and that together they make a powerful XML transaction-processing system.*

## Introduction

Because XML is the standard for data exchange, service-oriented architectures (SOAs) and message-based transaction processing, corporate databases contain large amounts of XML data. Consequently, a high-performance database is required to store, manage and query this data. With the introduction of IBM® DB2® pureXML in DB2 9, servers seamlessly handle XML and relational data. Users can define columns of type XML to store XML documents in their native form as hierarchical tree-like structures. Tables can contain XML and relational columns, which simplifies integration of XML and relational data.

Figure 1 shows that when XML documents are inserted or loaded into an XML column, they are parsed with optional XML schema validation and stored in a parsed tree format. This format allows queries and updates over the XML data to run quickly, without the need for XML parsing. You can define XML indexes on specific elements or attribute to further increase query performance. Queries and updates are written by using the standardized SQL/XML<sup>1</sup> and XQuery<sup>2</sup> languages and can access both XML and relational data in a single statement, if needed.

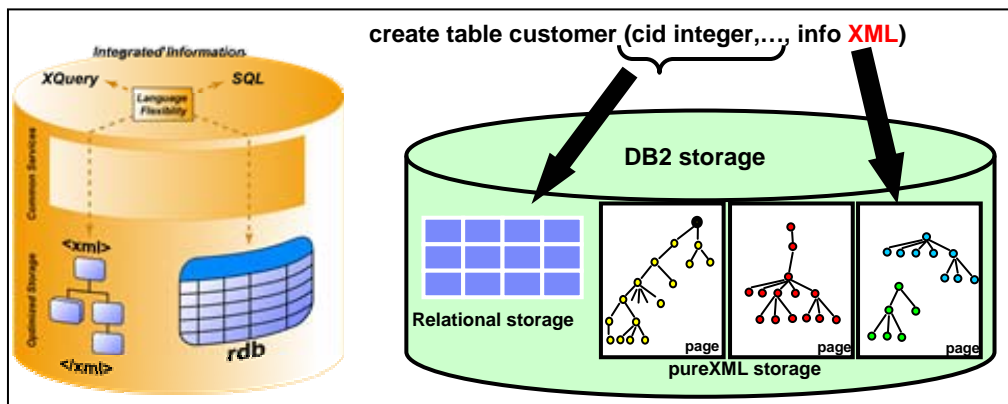


Figure 1. DB2 seamlessly handles XML and relational data

## TPoX benchmark

TPoX (Transaction Processing over XML)<sup>3</sup> is an open-source XML database benchmark that is based on a financial-application scenario, namely an online-brokerage system. This benchmark evaluates the performance of XML database systems, focusing on XQuery, SQL/XML, XML storage and indexing, XML Schema support, XML inserts, updates and deletes, logging, concurrency, and other database aspects. TPoX is designed to simulate a realistic and representative set of XML operations.

The five main logical data entities in TPoX (*customer*, *account*, *holding*, *security* and *order*) and their relationship are described in Figure 2. For each customer, an XML document contains all their personal,

<sup>1</sup> A Web site on SQLXML: [ibm.com/developerworks/xml/standards/x-sqlxmlspec.html](http://ibm.com/developerworks/xml/standards/x-sqlxmlspec.html)

<sup>2</sup> Official Web site on W3C XML Query: [www.w3.org/XML/Query](http://www.w3.org/XML/Query)

<sup>3</sup> Official TPoX Web site: <http://tpox.sourceforge.net/>.

account and holding information. Each order is represented by an XML document that complies with the FIXML 4.4 XML schema. FIXML is a complex financial industry-standard XML schema for trade-related messages. Each of the 20 833 securities is described in a separate XML document. Within the benchmark scenario, the number of security documents is fixed; however, the number of customer account (custacc) and order documents do scale as needed to test various loads.

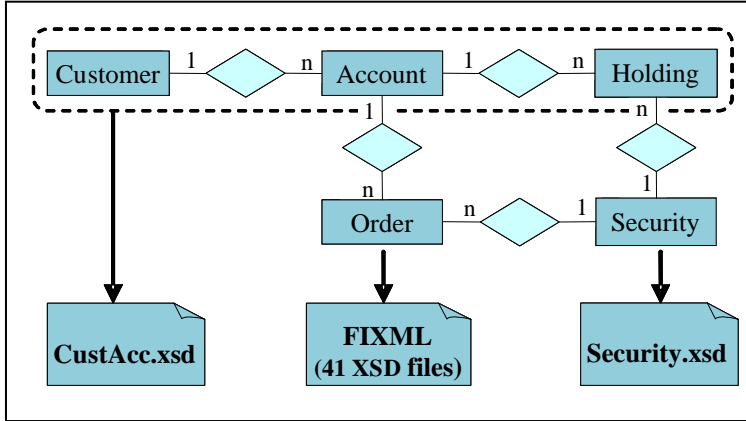


Figure 2. TPoX data entities

Table 1 shows the 17 transactions in the TPoX workload along with their relative weight in the transaction mix. About 30 percent of the workload consists of insert, update, delete (IUD) operations; the remaining 70 percent are queries. XML Schema validation is performed in transaction I2, U2 and U4. The workload driver spawns a configurable number of concurrent threads to simulate concurrent users. Each thread connects to the database and runs a stream of transactions without think-times.

For these tests, the latest version of TPoX, namely TPoX 2.0, was used with a scale factor of M to create one terabyte of raw XML data. The number of XML documents and their sizes is as follows:

- 500M Order XML documents; 1 to 2 KB
- 50M CustAcc XML documents; 2 to 23 KB
- 20 833 Securities XML documents; 2 to 9 KB

Transaction number	Purpose	Weight
Q1	Retrieve an order for a given order ID	10%
Q2	Retrieve a security for a given ticker symbol without root element	10%
Q3	Get a customer's personal data, construct profile document	10%
Q4	Search securities based on four predicates; return specific elements of interest	10%
Q5	Construct an account summary and statement	10%
Q6	Retrieve the price of a certain security	10%
Q7	Get a customer's most expensive order	10%
U1	Close an existing customer's account	1%
U2	Open a new account for an existing customer	1%
U3	Update the price of a security	3%
U4	Update the status of an order	3%
U5	Run a <i>buy</i> order of a given security for a given account: <ol style="list-style-type: none"> <li>1. If shares exist, increase the quantity; otherwise, add a new holding</li> <li>2. Update account balance and date values</li> <li>3. Abort if the maximum number of holdings is exceeded</li> </ol>	3%
U6	Run a <i>sell</i> order (opposite of U5)	3%
I1	Customer places a new order (insert order document)	7%
I2	Add a new customer (insert CustAcc document)	1%
D1	Cancel or archive an order (delete Order document)	7%
D2	Remove a customer (delete CustAcc document)	1%

Table 1. Business descriptions of TPoX transactions

High-performance XML transaction processing



Because of significant changes in the XML data as well as some changes in the update transactions, the results from a TPoX 2.0 benchmark cannot be compared to results obtained using earlier versions of TPoX. Table 2 summarizes some of these changes. See *TPoX 2.0 Release Notes* at <http://sourceforge.net/projects/tpox> for further details.

	TPoX 1.3 and earlier	TPoX 2.0
<b>Number of <i>CustAcc</i> and <i>Order</i> documents</b>	1:5	1:10
<b>XML document-size range</b>	1 to 20 KB	1 to 23 KB
<b>Account IDs of customers</b>	Not dense	Dense
<b>Total XML document size (e.g., 100 GB scale)</b>	Slightly less than 100 GB	Slight larger than 100 GB
<b>Average number of accounts per customer</b>	1.5	2.0

Table 2. What is new in TPoX 2.0

## IBM BladeCenter JS23 and JS43

IBM BladeCenter servers represent one of the most flexible and cost-efficient solutions for the deployment of UNIX®, IBM i and Linux® operating systems. Their innovative, open design offers an appealing alternative to sprawling racks and overheated server rooms. The IBM BladeCenter® JS23 and JS43 Express blade servers are the latest members of the IBM BladeCenter POWER6™ processor-based blade family. You can easily install them in the same chassis with other BladeCenter blade servers. This dynamic infrastructure helps deliver superior business and IT services with agility and speed in a simple-to-manage and highly efficient manner.

This new family allows processor scalability, starting with a two-processor (four cores, single-wide) blade and adds the ability to upgrade to a four-processor, eight-core blade with the addition of a second expansion blade — making it a double-wide package. The new blades support the IBM AIX, IBM i and Linux operating systems, the IBM EnergyScale technologies, IBM PowerVM™ Virtualization and AltiVec SIMD acceleration functions. You can use optional IBM Systems Director Active Energy Manager (AEM) for IBM POWER® V5.1 and browser-based IBM Integrated Virtualization Manager (IVM) software to achieve increased usage and energy efficiency by using the PowerVM and EnergyScale capabilities of the system.

Although IBM BladeCenter servers are well-known for their ease-of-use, energy efficiency, scalability and manageability, they are often overlooked when selecting database servers. The BladeCenter JS23 and JS43 blades are built with the goal of being the premier blades for 64-bit applications. The tests documented in this paper validate that the IBM BladeCenter JS43 blade server can be a highly scalable database server.

## TPOX results: DB2 pureXML with BladeCenter JS43

This section describes the impressive performance results that can be obtained when combining the IBM BladeCenter JS43 and DB2 9.7. The TPOX benchmark was used to measure XML transaction-processing throughput and XML insertion rates. Details of the system under test (SUT) are also explained.

### System under test (SUT)

The tests documented in this paper were done on the IBM BladeCenter JS43 using IBM DB2 9.7 (see Figure 3). The platform was 64-bit AIX 6.1 TL02. The client machine was an IBM BladeCenter JS12 Express blade server with a dual-core processor, 3.8 GHz and 16 GB memory.

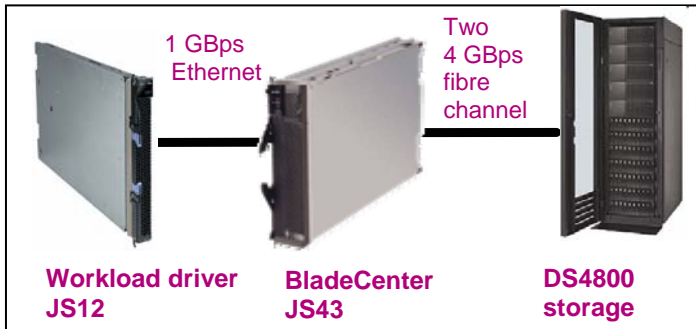


Figure 3. BladeCenter details

See Table 3 for JS43 hardware details.

Platform name	IBM JS43 Blade Server on BladeCenter H
<b>Processors</b>	Four POWER6 sockets Two cores per socket 32 MB L2 (4 MB per core) 128 MB L3 (32 MB per socket) Clock frequency: 4.2 GHz
<b>Memory</b>	128 GB
<b>Storage</b>	Two internal drives and an IBM System Storage® DS4800 storage system with the following: <ul style="list-style-type: none"> <li>- 110 disks (for the DB2 database)</li> <li>- 14 disks (RAID 10 for logging)</li> <li>- 96 disks (RAID 5 for data)</li> </ul>
<b>HBA interface</b>	One Emulex PCI Express 4 GB Dual Port Fibre Channel adapter
<b>Network Interface</b>	LP-HEA logical host Ethernet port (integrated on the blade)

Table 3. SUT details



The DB2 database was created with the DB2 automatic-storage feature and a page size of 16 KB, using six logical volumes plus a separate volume for the log. The system was set up as a production-like environment with RAID 10 and RAID 5 for database logging and data storage, respectively. The database schema consisted of three tables with one XML column in every table for each of the three TPoX XML document types (*order custacc* and *security*). XML inlining and compression were used to reduce the storage footprint. The following DDL was used to create the tables:

```
create table security (sdoc xml inline length 16288) in SECURITY_ALL compress yes
create table custacc (cadoc xml inline length 16288) in CUSTACC_DATA index in CUSTACC_INDEX compress yes
create table order (doc xml inline length 16288) in ORDER_DATA index in ORDER_INDEX compress yes
```

Five tablespaces (three tablespaces for the three tables, one tablespace for custacc indexes and one for order indexes) were created. These tablespaces were configured with NO FILE SYSTEM CACHING and AUTOMATIC STORAGE. A separate temp tablespace was defined as SMS file-system caching. Each tablespace had its own buffer pool. Minimal tuning was done.

### Transaction throughput of mixed TPoX workload

The TPoX mixed workload was run, using different numbers of users for one hour after a 30-minute ramp-up. This test was done on a database that was created from 1TB of raw XML data (550 million XML documents). As seen in Figure 3, the throughput and processor usage scales up well as the number of users increase. The highest throughput of 4107 XML transactions per second (246 420 transactions per minute, 14.7 million transactions per hour) was obtained using 300 users. The average response time with the 300 users was just 0.07 seconds. It was also noted that slightly higher throughput, around 3%, was obtained when compared to running DB2 9.5 Fixpack 3 on the same setup.

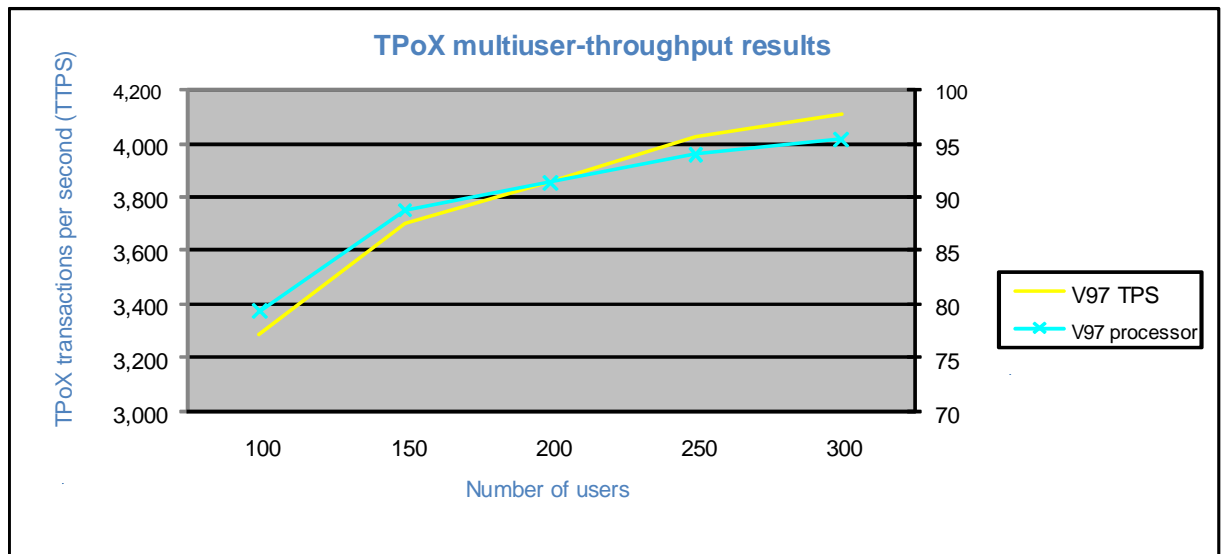


Figure 4. Scaling of transactions with the number of users

## XML insert performance

In addition to high transaction rates with the TPoX mixed workload, impressive insertion rates with the insert-only TPoX workload were also obtained. In the insert-only workload transactions, I1 (insert order XML documents) or I2 (insert customer order documents) are run. XML Schema validation was not performed during the insert. Results for these two insert transactions are in Table 4.

Transaction	Insert rate	Document size
I1 (insert order documents )	- 13 945 documents per second - 50 million documents per hour - 75 GB per hour	1 to 2 KB
I2 (insert custacc documents)	- 5119 documents per second - 18.2 million documents per hour - 120 GB per hour	2 to 23 KB

Table 4. Insert Rates for Order and Customer XML documents

The same tests were performed on DB2 9.5 Fixpack 3, revealing that DB2 9.7 has about a 9% higher insertion rate than DB2 9.5 Fixpack 3 (see Figure 5).

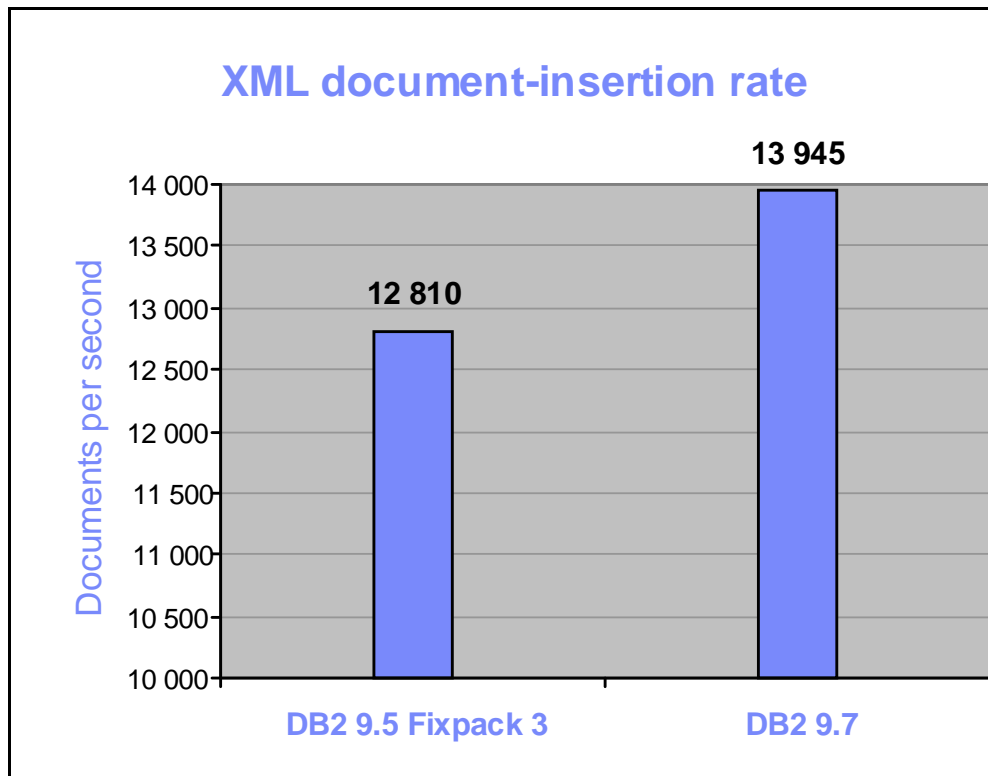


Figure 5. Insertion rates for DB2 9.5 Fixpack 3 as compared to DB2 9.7



## Storage consumption and compression results

A key new feature that DB2 9.5 introduces is data-row compression, which allows you to compress rows in the DB2 table by using a static dictionary-based compression algorithm. Compressing data at the row level makes it possible to repeat patterns that span multiple column values within a row that is to be replaced with shorter symbol strings. Applying compression to tables saves disk-storage space and disk I/O transactions. Additionally, more data can be cached in the buffer pool, thereby leading to increased bufferpool hit ratios. However, there is an associated cost in the form of extra processor cycles that are needed to compress and decompress data. The reduced I/O and higher bufferpool hit ratio typically outweighs the drawback of the extra processor consumption by a significant margin. The notable exception is when a system is already nearly 100-percent processor bound.

Row compression was applied to all the tables along with XML Inlining. XML Inlining allows small XML documents to be stored physically along with the other data in the table — rather than in the separate XDA area. When the 1 TB of raw XML data is loaded into DB2, there was a resultant 1.4 TB database (uncompressed), including the XML data and indexes. Applying data compression and inlining reduced the entire database size, including XML data and indexes, by 58 percent — yielding a 604 GB database. Individual compression ratios of the *order* and *custacc* data are shown in Table 5.

	Uncompressed	Compressed	Compression ratio
<b>Custacc XML data</b>	348.06 GB	134.81 GB	61.3 percent
<b>Order XML data</b>	978.03 GB	400.10 GB	59.1 percent

Table 5. Compression ratio of customer and order documents



## Conclusion

---

Because of the popularity of XML, companies now have large amounts of XML data in their system and they require high-performing databases to store, manage and query this data. With the introduction of pureXML in DB2 9, users have a database that can seamlessly handle both XML and relational data. This paper explored how the latest IBM BladeCenter JS43 blade server can be combined with DB2 pureXML to act a powerful transactional-processing data server. The Transaction Processing over XML (TPoX) benchmark was used to evaluate the performance of DB2 9.7 on the JS43 server. The TPoX application benchmark simulates an online brokerage system and is designed to exemplify a realistic and representative set of XML operations.

The results show that the JS43 blade server is in fact a highly scalable database server, that DB2 pureXML is a high-performing database and that together, they make a powerful XML transaction-processing system.



## Resources

---

These Web sites provide useful references to supplement the information contained in this document:

- IBM Publications Center  
[www.elink.ibm.com/public/applications/publications/cgi-bin/pbi.cgi?CTY=US](http://www.elink.ibm.com/public/applications/publications/cgi-bin/pbi.cgi?CTY=US)
- IBM Redbooks®  
[ibm.com/redbooks](http://ibm.com/redbooks)
  - *IBM BladeCenter JS23 and JS43 Implementation Guide*  
[ibm.com/redbooks/redpieces/pdfs/sg247740.pdf](http://ibm.com/redbooks/redpieces/pdfs/sg247740.pdf)
- IBM BladeCenter JS23 and JS43 Express  
[ibm.com/systems/in/bladecenter/hardware/servers/js2343](http://ibm.com/systems/in/bladecenter/hardware/servers/js2343)
- IBM systems on PartnerWorld®  
[ibm.com/partnerworld/systems](http://ibm.com/partnerworld/systems)
- For more information about IBM Power Systems servers  
[ibm.com/systems/power](http://ibm.com/systems/power)
- For more information about AIX  
[ibm.com/servers/aix](http://ibm.com/servers/aix)
- For more information about IBM storage  
[ibm.com/servers/storage](http://ibm.com/servers/storage)
- DB2 9.7 Information Center  
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.wn.doc/doc/r0051514.html>
- Best Practices -- Managing XML data  
[http://download.boulder.ibm.com/ibmdl/pub/software/dw/dm/db2/bestpractices/DB2BP\\_XML\\_0508I.pdf](http://download.boulder.ibm.com/ibmdl/pub/software/dw/dm/db2/bestpractices/DB2BP_XML_0508I.pdf)
- Essential Reading for DB2 pureXML  
[ibm.com/developerworks/wikis/display/db2xml/Technical+Papers+and+Articles](http://ibm.com/developerworks/wikis/display/db2xml/Technical+Papers+and+Articles)
- DB2 Best Practices  
[ibm.com/developerworks/data/bestpractices](http://ibm.com/developerworks/data/bestpractices)
- Standardized SQL/XML language  
[ibm.com/developerworks/xml/standards/x-sqlxmlspec.html](http://ibm.com/developerworks/xml/standards/x-sqlxmlspec.html)
- TPoX Web site  
<http://tpox.sourceforge.net>
- Additional TPoX benchmark results  
<http://tpox.sourceforge.net/tpoxresults.htm>
- W3C XML Query (XQuery language)  
[www.w3.org/XML/Query](http://www.w3.org/XML/Query)

*High-performance XML transaction processing*



## About the authors

---

**Kevin Xie** is a DB2 Performance Analyst in the DB2 Performance Benchmarks and Solutions Development team at the IBM Toronto Laboratory. Kevin holds a Masters of Science degree in database optimization, and currently focuses on DB2 XML performance benchmarking and analysis.

**Jasmi Thekveli** is a DB2 Performance Analyst in the DB2 Performance Benchmarks and Solutions Development team. Jasmi joined IBM in 2002 after completing her Masters of Science degree in computer science at Queen's University, Canada. Jasmi is currently the XML team lead in the DB2 Performance Quality Assurance area.

**Peter Shum** is a Senior Technical Manager in the DB2 Performance Benchmarks and Solutions Development team at the IBM Toronto Laboratory. Peter has been with IBM for 28 years in the development, architecture and performance analysis of the DB2 data server. The current focus is on the benchmarking and performance improvement of DB2 when working with middleware such as SAP and IBM WebSphere®, and with technology such as XML and resiliency.

**Bob Taraba** is a Senior Technical Solutions Manager in the IBM Systems and Technology ISV Global support organization. Bob has been with IBM for 25 years as a Systems Engineer, Systems Technical Sales Specialist, ISV Solutions Manager in IBM PartnerWorld® and IBM Software Group technical enablement. His currently focuses on working with key IBM Software Group products and leveraging their value on IBM Systems platforms.

## Acknowledgments

---

The following individuals deserve special recognition for their involvement in creating this paper:

**Peter Jordan, Matthias Nicola, Thanh Do, Anupama Padmanabhan, Harsha Anand and Berni Schiefer**



## Trademarks and special notices

---

© Copyright IBM Corporation 2009. All rights Reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml).

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.