March, 2008

DB2 9 and z/OS XML System Services Synergy Update

Akiko Hoshikawa – DB2 for z/OS Development
Judy Ruby-Brown - ATS Dallas Systems Center

http://www.ibm.com/support/techdocs

Version 3/26/2008

Table of Contents

Prologue

The white paper, "DB2 9 and z/OS XML System Services Synergy", was published in August, 2007, http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101088. It focused on initial measurements for pureXML™ technology using DB2® Version 9.1 for z/OS® (DB2 9 hereafter) as an exploiter of z/OS XML System Services.

This paper may be read instead of its predecessor if it is desired.

- *New accounting fields have been provided in the DB2 trace records.*
- *The workload has been run on a later version of the z/OS operating system and comparisons are documented.*
- *The workload has been expanded to incorporate more XML benchmarks runs, including selected queries. This is known as "mixed workload".*
- *The background and certain other information have been repeated in this paper.*

A significant enhancement in this paper has been to provide new specialty engine CPU usage fields in the DB2 accounting records.

Introduction

DB2 9 became generally available in March, 2007. DB2 9 is a hybrid data server with integrated pureXML technology to natively manage XML data with unprecedented reliability, availability, scalability and performance. DB2 9 pureXML exploits z/OS XML System Services for high-performance parsing with improved price performance utilizing IBM eServer™ zSeries® specialty engines.

DB2 9 pureXML includes the following capabilities:

- *XML data type and storage techniques for efficient management of hierarchical structures inherent in XML documents.*
- *pureXML indexes to speed search subsets of XML documents.*
- *New query language support (SQL/XML and XPath) based on industry standards and new query optimization techniques.*
- *Industry-leading support for managing, validating, and evolving XML schemas.*
- *Comprehensive administrative capabilities, including DB2 utilities and tools.*
- *Integration with popular application programming interfaces (APIs) and development environments.*
- *XML shredding, publishing, and relational view facilities for working with existing relational models.*
- *Proven enterprise-level reliability, availability, scalability, performance, security, and maturity that users expect from DB2.*

Detailed information on pureXML capabilities can be found in *DB2 Version 9.1 for z/OS XML Guide*, SC18-9858, or at http://publib.boulder.**ibm.com**/epubs/pdf/dsnxgk10.pdf.

Prior to DB2 9, an XML document could have been stored in its entirety as a CLOB (Character Large Object). Alternatively, the XML Extender could have been invoked to convert and map XML data to relational data. The XML Extender used "side tables" as rudimentary indexes into the XML data and used the z/OS XML Toolkit to manipulate XML data. DB2 9 pureXML eliminates the need for the XML Extender and can help developers to reduce application system complexity, improve development productivity, lower maintenance costs, and develop new applications to get insight into unexploited XML data.

It is recognized that XML, which is widely adopted across industries for its flexibility and portability, is nonetheless verbose and its manipulations incur more overhead than its relational counterparts. Even though z/OS XML System Services is an efficient system level XML parser on z/OS, additional general CPU cost reduction is possible. With the proper software and hardware, local DB2 calls to z/OS XML System Services are redirected to a IBM System z™ Application Assist Processor (*zAAP)* specialty engine. Included are invocations from the attachments that run in TCB mode. DB2 already has functions that can be redirected to the System z Integrated Information Processor (*zIIP)* specialty engine.

For workload already on a specialty engine, the intent is to continue to run it. For workload not running on a specialty engine, the intent is to allow the DB2 XML processing to run on a specialty engine to mitigate its cost. The following table is provided to assist the reader to understand which capabilities are eligible for redirection to a specific specialty engine:

Table 0.1. Sources of redirection for zIIP / zAAP

DB2 9 XML workload invoked from . . .	DRDA® via TCPIP	Execution Mode	zAAP eligible	zIIP eligible
CICS®	No	TCB	Yes	No
IMS™	No	TCB	Yes	No
TSO	No	TCB	Yes	No
WebSphere® for z/OS (JCC T2)	No	TCB	Yes	No
Stored Procedure/UDF/Trigger	No	TCB	Yes	No
Native SQL Stored Procedure	No	TCB	Yes	No
Call Attach (CAF)	No	TCB	Yes	No
RRS Attach	No	TCB	Yes	No
Local LOAD utility (data)	No	TCB	Yes	No
Index Build phase of LOAD	No	SRB	No	Yes
WebSphere for z/OS (JCC T4)	Yes	SRB	No	Yes
WebSphere distributed (JCC T4)	Yes	SRB	No	Yes
Distributed transactions using DB2 Connect™	Yes	SRB	No	Yes
Stored Procedure/UDF/Trigger	Yes	TCB	Yes	No
Native SQL Stored Procedure	Yes	SRB	No	Yes

In the table above, the term "SRB" means "enclave SRB." There are differences between DRDA redirection and z/OS XML System Services redirection. For XML, DB2 9 XML inserts, updates or the LOAD utility invoke z/OS XML System Services and these system invocations are the only operations eligible for zAAP. A workload arriving through DRDA can have a portion of its *entire* XML workload redirected to the *zIIP*, including select, insert, update, delete and query. A DRDA workload executes in Enclave SRB mode and none of it is zAAP-eligible.

Even though a local XML workload is 100% zAAP-eligible for its z/OS XML System Services invocation, only the z/OS XML System Services portion, not the entire execution of the XML insert/update/LOAD is redirected. DB2 actions prior to and after that call are not zAAP-eligible.

Prerequisites

The primary prerequisites for zAAP redirection with z/OS XML System Services are:

Hardware:

A zAAP processor, available on IBM System z10™ Enterprise Class (z10 EC™), IBM System z9® Enterprise Class (z9™ EC, formerly z9-109), IBM System z9 Business Class (z9 BC), IBM eServer zSeries 990 (z990) and IBM eServer zSeries 890 (z890) systems.

Software:

z/OS Version 1.9. Service to enable this capability for earlier z/OS releases is:
z/OS V1.8 OA20308
z/OS V1.7 OA20308, OA16303
DB2 9 in New Function Mode (NFM). Relevant service is listed on page 19.

Background

Since pureXML is a capability that did not exist in DB2 for z/OS before DB2 9, there are no existing workloads from which the projections of zAAP redirection can be made. There is also no practical way to relate XML Extender costs to pureXML. This paper uses the "Transaction Processing over XML (TPoX)" benchmark workload as the basis for analysis. The formal description of this open source benchmark can be found at http://tpox.sourceforge.net/. TPoX is an XML database benchmark based on a financial application scenario. It is used to evaluate the performance of XML database systems, focusing on XQuery, SQL/XML, XML storage, XML indexing, XML Schema support, XML updates, and other aspects. The TPoX benchmark uses XML documents from the financial industry standard and its workload consists of inserting, updating, and running queries against them. Unlike relational data, an update in XML data storage is a delete, followed by an insert.

The TPoX benchmark was run at the DB2 development laboratory in Silicon Valley to determine the amount of CPU consumed by a given amount of XML workload and the amount of CPU that can be redirected to the zAAP. While the entire TPoX benchmark has not been completed, partial results are presented in this paper in order to help developers to plan for implementation of new XML applications on System z. This version of the paper reports on the mass inserts, as well as certain query transactions. While queries do not invoke z/OS XML System Services, they are part of a normal workload which also includes inserts and updates that do use z/OS XML System Services. XML validation is not yet supported through z/OS XML System Services, but is planned to be added in a future release of z/OS (see page 17).

Using RMF and DB2 accounting traces

The examples described in this paper illustrate how the DB2 accounting trace and RMF™ can be used to verify specialty engine redirection and provide estimation for zIIPs. RMF can also provide estimation for zAAPs.

Accounting Support

DB2 9 APAR PK50575 adds support for accounting zAAP redirection in addition to that already provided for zIIP. *IBM Tivoli® OMEGAMON® XE for DB2 Performance Expert on z/OS, Version 4*, (hereafter called OMPE) incorporates the new fields with PK51045, so the DB2 fields can be properly formatted.

The fields used in the DB2 accounting trace (SMF101) records include CPU time spent on all specialty engines, not only zAAPs. The fields formerly used for zIIPs are now folded into the new fields and the user receives in those fields the combination of the values. The appendix on page 19 shows the general format of the pertinent DB2 fields.

While the accounting records record CPU time on both zIIP and zAAP engines when they exist, they cannot be used to estimate zAAP redirection. Statistics for zAAP-eligible work that runs on a general purpose CP are not available to DB2.

Using z/OS to estimate redirection

The best way to estimate specialty engine redirection is to set PROJECTCPU=YES, specified in IEAOPTxx member in SYS1.PARMLIB. With PROJECTCPU=YES, RMF identifies and populates separate fields with specialty engine eligibility values (IIPCP and AAPCP). PROJECTCPU=YES was used for all the runs described in this paper.

RMF Workload Activity Report

Setting the IEAOPTxx parmlib member option PROJECTCPU=YES directs z/OS to record the amount of work eligible for zAAP or zIIP processors. SMF Record Type 72 subtype 3 is input to the RMF post processor. The Workload Activity Report lists workloads by WLM service class. In this report the field APPL% AAPCP indicates which percentage of a processor is zAAP eligible, while the field APPL% IIPCP indicates which percentage of a processor is zIIP eligible. SMF Record Type 30 provides more detail on specific address spaces. Because the price of zIIP and zAAP specialty processors is less than that of general purpose processors, even very low values (less than 25%) for APPL% AAPCP or AAPL% IIPCP can make a new zAAP or zIIP processor worth consideration.

The “APPL %” column is the one to look at in this snippet of the Workload Activity shown:

```
REPORT BY: POLICY=POL_XML    WORKLOAD=BATCH      SERVICE CLASS=BATCHMED   RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=3
                                                 CRITICAL     =NONE
-TRANSACTIONS-  TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE----  --SERVICE TIMES--  ---APPL %---  -----STORAGE-----
AVG       6.50  ACTUAL          9.50.231  SSCHRT 665.2  IOC    884057   CPU      1448.496  CP     188.65  AVG       3751.48
MPL       6.50  EXECUTION       8.46.172  RESP     0.3  CPU    398897K  SRB         0.862  AAPCP    0.04  TOTAL    24368.79
ENDED        4  QUEUED          1.04.059  CONN     0.2  MSO         0   RCT         0.000  IIPCP    0.00  SHARED       0.00
END/S     0.01  R/S AFFIN               0  DISC     0.0  SRB    237449   IIT         0.155
#SWAPS       0  INELIGIBLE              0  Q+PEND   0.1  TOT    400019K  HST         0.000  AAP     53.16  --PAGE-IN RATES--
EXCTD        0  CONVERSION              0  IOSQ     0.0  /SEC   667455   AAP       318.578  IIP      0.05  SINGLE         0.0
AVG ENC   0.00  STD DEV                 0                              IIP         0.329                     BLOCK          0.0
REM ENC   0.00                                              ABSRPTN   103K                                 SHARED         0.0
MS ENC    0.00                                              TRX SERV  103K  PROMOTED    0.000              HSP            0.0
GOAL: EXECUTION VELOCITY 40.0%     VELOCITY MIGRATION:   I/O MGMT  73.9%     INIT MGMT 73.9%
```

The pertinent CPU percentage fields are:

1. *CP – total executing on the general purpose CPs*
2. *IIP – total spent executing on the zIIP*
3. *AAP – total spent executing on the zAAP*
4. *IIPCP – total that could have been executed on a zIIP, but there was either not enough capacity or a zIIP was not configured*
5. *AAPCP – total that could have been executed on a zAAP, but there was either not enough capacity or a zAAP was not configured*

Fields #4 and #5 are included in #1. Fields #2 and #3 are excluded from #1.

This snippet shows that the largest percentage of time on a specialty engine was spent on the zAAP (AAP=53.16). A small amount was spent on the zIIP (IIP=.05). There was insufficient zAAP capacity, so the rest executed on the general purpose CP (AAPCP=.04).

Since DB2 has combined the time eligible in both specialty engines, the RMF Workload Activity Report should be used for system-wide estimation and for specific specialty engine usage.

DB2 accounting data should be used for application based estimation for zIIPs only.

For measurement events (vs. estimation) the RMF Workload Activity Report fields labeled CP plus AAP and IIP must be totaled in order to match the OMPE accounting trace totals of CP CPU TIME + SE CPU Time.

LOAD jobs

The separate DB2 LOAD utility runs (which also call z/OS XML System Services), reported here and in the previous paper, are not part of the TPoX benchmark.

Configuration used

Measurements were run with the following configuration:

Processor	IBM System z9 Enterprise Class (z9 EC)
LPAR configuration:	4 General Purpose CPs, 1 zAAP, 1 zIIP, all dedicated
Memory:	24GB memory

Storage	IBM DS8300
Operating system	z/OS Version 1.9
DB2	DB2 9 Feb 2008 PTF level

The measurements are performed in a controlled environment where there is no other workload running in the LPAR.

Seven (7) DB2 Load utility jobs with different characteristics are executed against seven (7) XML tables. The CPU time spent on the general purpose processor and CPU time spent on the specialty engine is recorded in the accounting trace.

Job	Number of Rows	Size of XML documents (bytes)	Number of User XML Indexes	CPU time in general CP(sec)	CPU time in zAAP	Redirection percentage
LOAD1	300,000	4K-20K	4	191	40	17%
LOAD2	300,000	4K-20K	2	152	31	17%
LOAD3	300,000	4K-20K	1	93	38	29%
LOAD4	2,000	2.5M	2	329	82	20%
LOAD5	2,000	2.5M	2	330	82	20%
LOAD6	200	25M	1	254	64	20%
LOAD7	200	25M	1	114	64	36%
Average				209	57	21%

Note: The XML index in LOAD6 is more expensive to process than that of LOAD7.

In general, the larger the document size or the fewer the number of indexes, the greater the CPU percentage is spent in z/OS XML System Services. For those conditions, there is a higher percentage of redirection to zAAP processor. In addition, document complexity and index path definition also affect the redirection rate. The more indexes that are defined, the lower the percentage of redirection, as there is more index processing within DB2 that is not related to z/OS XML System Services.

Accounting Trace

The OMPE accounting trace snippet represents all the LOAD jobs described in the preceding section. The fields in turquoise are newly added for specialty engines (SE) in OMPE APAR PK51045.

```
AVERAGE        APPL(CL.1)  DB2 (CL.2)
-------------  ----------  ----------
ELAPSED TIME   11:36.6837  11:27.1818
 NONNESTED     11:36.6837  11:27.1818
 STORED PROC     0.000000    0.000000
 UDF             0.000000    0.000000
 TRIGGER         0.000000    0.000000
CP CPU TIME    3:29.67361  3:23.08585
 AGENT         3:29.53441  3:23.08558
  NONNESTED    3:29.53441  3:23.08558
  STORED PRC     0.000000    0.000000
  UDF            0.000000    0.000000
  TRIGGER        0.000000    0.000000
 PAR.TASKS       0.139205    0.00027
  SECP CPU       0.000079         N/A
SE CPU TIME     57.497246   57.497246
 NONNESTED      57.497246   57.497246
 STORED PROC     0.000000    0.000000
 UDF             0.000000    0.000000
 TRIGGER         0.000000    0.000000
 PAR.TASKS       0.000000    0.000000
SUSPEND TIME     0.000000  5:43.35838
 AGENT                N/A  5:43.35838
 PAR.TASKS            N/A    0.000000
 STORED PROC     0.000000         N/A
 UDF             0.000000         N/A
NOT ACCOUNT.          N/A  1:23.24058
DB2 ENT/EXIT          N/A     8536.00
EN/EX-STPROC          N/A        0.00
EN/EX-UDF             N/A        0.00
DCAPT.DESCR.          N/A         N/A
LOG EXTRACT.          N/A         N/A
```

The fields

- *SECP CPU can reflect one of two things:*
  1. *It is zIIP (not zAAP) eligible work that ran on a general purpose CP because a zIIP was not configured*
  2. *The capacity of zIIP processors had been reached and the "overflow" could not be accommodated.*
- *Statistics for zAAP-eligible work that runs on a general purpose CP are not available to DB2.*
- *SE CPU TIME represents the CPU time consumed on all specialty engines combined.*

For this run the SE CPU TIME (57.49) included a small amount of index build activity for the LOAD jobs. That went to the zIIP, as it ran in an enclave SRB. The remainder of the time was zAAP. It appears the zIIP capacity was exceeded with a very small amount overflowing into SECP (.000079), but this did not occur. An amount this small can be attributed to "noise". It only takes a single slice of time where the processor was held by another user for this value to be incremented.

Note: SECP CPU contents cannot be determined from the Accounting trace alone, but through the RMF Workload Activity report, described next.

The percentage of redirection is calculated as the SE CPU time (57.49 sec) divided by the sum of CP CPU time and SE CPU Time for 21.5%= (57.49/(209.67 + 57.49)*100.

RMF Workload Activity Report

The LOAD jobs (on page 6) were executed under the service class (BATCHMED). Under the "APPL %" column, the total CPU usage was 188.65% (CP) and 53.16% (AAP) was redirected to zAAP. The ratio of redirection is 22% (= 53.16/(188.65+53.16)*100). This corresponds closely with the OMPE accounting trace on page 7. This calculation assumes the specialty engines run at the same speed as the general CPs. If the specialty engines are faster than the general CPs, the formula above must be normalized to arrive at the redirection percentage.

Note: For actual measurement, redirection percentages can be off by up to 3% due to the different measurement intervals and event activities between DB2 and RMF. Thus, the two times are considered to be equivalent.

```
REPORT BY: POLICY=POL_XML    WORKLOAD=BATCH      SERVICE CLASS=BATCHMED   RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=3
                                                 CRITICAL     =NONE
-TRANSACTIONS-  TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE----  --SERVICE TIMES--  ---APPL %---  -----STORAGE-----
AVG       6.50  ACTUAL          9.50.231  SSCHRT 665.2  IOC    884057   CPU      1448.496  CP    188.65  AVG       3751.48
MPL       6.50  EXECUTION       8.46.172  RESP     0.3  CPU    398897K  SRB         0.862  AAPCP   0.04  TOTAL    24368.79
ENDED        4  QUEUED          1.04.059  CONN     0.2  MSO         0   RCT         0.000  IIPCP   0.00  SHARED       0.00
END/S     0.01  R/S AFFIN               0  DISC     0.0  SRB    237449   IIT         0.155
#SWAPS       0  INELIGIBLE              0  Q+PEND   0.1  TOT    400019K  HST         0.000  AAP    53.16  --PAGE-IN RATES--
EXCTD        0  CONVERSION              0  IOSQ     0.0  /SEC   667455   AAP       318.578  IIP     0.05  SINGLE         0.0
AVG ENC   0.00  STD DEV                 0                               IIP         0.329                 BLOCK          0.0
REM ENC   0.00                                           ABSRPTN   103K                                  SHARED         0.0
MS ENC    0.00                                           TRX SERV  103K  PROMOTED    0.000               HSP            0.0
GOAL: EXECUTION VELOCITY 40.0%     VELOCITY MIGRATION:   I/O MGMT  73.9%     INIT MGMT 73.9%
```

RMF shows that a small percentage of time was consumed on the zIIP (IIP=.05). That was due to the Index Build part of LOAD. The capacity of the zAAP was exceeded as the AAPCP field is populated (.04). The OMPE Accounting trace can only show a zIIP overflow. RMF shows each one that overflowed.

XML Insert comparison of z/OS Versions 1.8 to 1.9

The TPoX workload is a simplification of a real world security brokerage application. It has three different types of XML documents: Order, Security and CustAcc. CustAcc includes a customer with all his/her accounts. The information about holdings is included in the account data. The objective of this workload is to demonstrate the CPU usage of z/OS XML System Services in insert operations. In the previous paper, measurements were taken for the following two scenarios described in the TPoX benchmark:

- *INSERT1: A customer places a new order to buy a stock*
  - *Result: Insert a new Order document in the collection of order documents*
- *INSERT2: A new customer signs up for an online brokerage account*
  - *Insert a new CustAcc document in the collection of CustAcc documents*

z/OS Version 1.9 introduced performance improvements in z/OS over that of z/OS V1.8. While this is excellent for DB2 XML performance in general, the improvement can affect the amount of redirection obtained. The following measurement demonstrates the effect of z/OS 1.9 on z/OS XML System Services redirection for a locally run application with heavy insert processing. This measurement reran only "INSERT2".

As the original benchmark in the previous paper used internal tools to measure z/OS XML System Services CPU utilization to determine the redirection percentage due to the release level of z/OS code, this repeat of the benchmark employs the same internal tools to measure. Please refer to page 22 in the Appendix for earlier measurements.

Configuration used

Measurements were run with the following configuration, which was the same hardware for both cases:

Processor	IBM System z9 Enterprise Class (z9 EC)
LPAR configuration:	3 dedicated General Purpose CPs (no zIIP no zAAP)
Memory:	24GB memory
Storage	IBM DS8300

The measurements are performed in a controlled environment where there is no other workload running in the LPAR. Case-A represents the measurements obtained in June, 2007. Case-B denotes the current environment.

Case–A
Operating system z/OS Version 1.8
DB2 DB2 9 June, 2007 PTF level
Case –B
Operating system z/OS Version 1.9
DB2 DB2 9 Feb. 2008 PTF level

Results

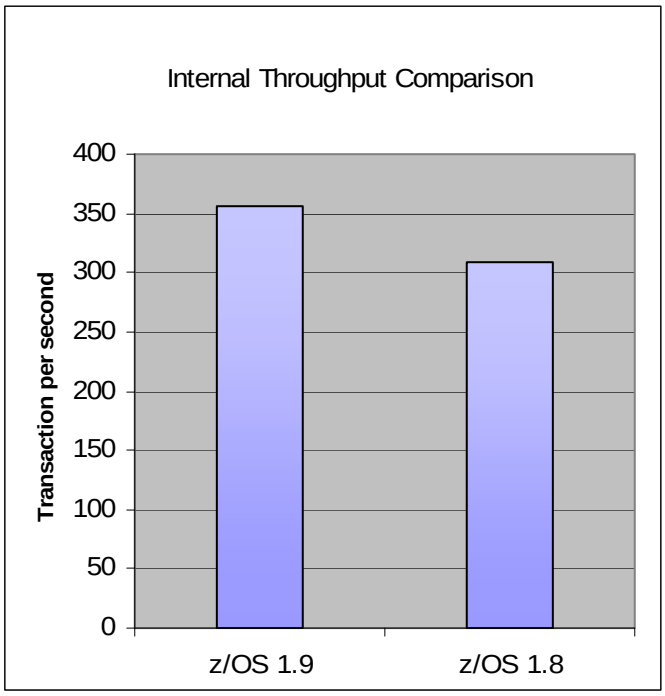
Six million documents were inserted against the TPoX CustAcc table with 20 clients (INSERT2). The description is found at http://tpox.sourceforge.net/. A commit is issued every 10 inserts.

As the chart below shows, there is observed an average CPU reduction of 4% from z/OS 1.8 to z/OS 1.9 due to the z/OS XML System Services improvement. The estimated portion of zAAP redirection has dropped from 32% to 18%.

TPoX Mass insert2	**z/OS 1.8 (Case-A)**	**z/OS 1.9 (Case-B)**
LPAR CPU utilization	76.45%	63.71%
Number of concurrent thread	20 Threads	20
XML insert per second (average)	2363	2269
DB2 class1 average elapsed time (ms, per commit)	77.565	88.501
DB2 class1 average CPU time (ms, per commit)	7.909	7.586
Number of transaction per second (10 insert)	236.3	226.9
Internal Throughput rate	309.09	356.15
XML System Service CPU usage in LPAR	**32.30%**	**18.23%**

Since there was a commit after each 10 inserts, the throughput rate in the chart is lower than that in the table.

No RMF or OMPE Accounting trace reports are provided, since those capabilities did not exist at the time of the original measurement.



Mixed XML transaction workload

The following measurements were done to determine the amount of redirection in this mixed workload. A mixture of query and insert processing yields a more accurate representation of the amount of redirection that can be obtained over an entire workload.

Configuration used:

Processor	IBM System z9 Enterprise Class (z9 EC)
LPAR configuration:	3 General Purpose CPs, dedicated (no zAAP or zIIP)
Memory:	24GB memory
Storage	IBM DS8300
Operating system	z/OS Version 1.9
DB2	DB2 9 Feb. 2008 PTF level

The measurements are performed in a controlled environment where there is no other workload running in the LPAR.

Mixed transactions and their weights

The following configuration and transactions were used from TPoX benchmark workload. They are described in http://tpox.sourceforge.net/ under "Benchmark Specification." The number in parentheses under the "Transaction Name" column is the query number in the benchmark specification document.

Transaction name	Type of transaction	Transaction Weight
Get _order (1)	Query	12
Get_security (2)	Query	12
Customer_profile (3)	Query	12
Account_summary (5)	Query	12
Get_security_price (6)	Query	12
Insert_custacc	Insert	20
Insert_order	Insert	20

Refer to the appendix on page 24 for the SQL used in the Query transactions.

Results

Transaction	Average Transaction Response time
Get _order	0.03 second
Get_security	0.03 second
Customer_profile	0.03 second
Account_summary	0.03 second
Get_security_price	0.02 second
Insert_custacc	0.03 second
Insert_order	0.02 second
Transaction per second	1207 tps
CPU utilization	58.3%
Internal Throughput Rate	2068 tps
z/OS XML System Services CPU consumption	3.5%

The system was able to drive 1200 transactions per second (or internal throughput rate of 2068) using 35 threads. z/OS XML System Services is invoked only through the two insert transactions (Insert_custacc and Insert_order) out of seven transactions. Each transaction issues a commit for every insert statement.

z/OS XML System Services usage for insert transaction (in this case one insert per transaction) is about 8%, based on an IBM internal tool. Had the workload been run locally (not done in the interests of time), the RMF Workload Activity Report would have provided in field AAPCP the percentage of redirection. Since inserts are 40% of the transaction mix, 40% of 8% yields 3.5% of potential redirection, which is reasonable in this case.

The z/OS XML System Services consumption and potential for redirection can vary significantly depending on the transaction distribution in the workload as well as the type of documents accessed.

DB2 DRDA zIIP estimation for the mixed XML workload

This measurement illustrates the estimation capabilities of the OMPE Accounting and RMF Workload Activity Reports when a zIIP specialty engine has not yet been installed.

An XML workload can take advantage of DB2 DRDA zIIP redirection when it enters DB2 for z/OS via DRDA (see table on page 3). This scenario is the same as for any other DRDA zIIP redirection. When the mixed workload described on page 13 was executed through DRDA, it showed the following zIIP projections both in Accounting and in the RMF Workload Activity Reports.

Accounting Trace

The OMPE accounting trace snippet represents the entire mixed XML workload: The fields in turquoise are those on which to focus:

AVERAGE	APPL(CL.1)	DB2 (CL.2)
ELAPSED TIME	0.020269	0.005248
NONNESTED	0.020269	0.005248
STORED PROC	0.000000	0.000000
UDF	0.000000	0.000000
TRIGGER	0.000000	0.000000
CP CPU TIME	0.001165	0.000994
AGENT	0.001165	0.000994
NONNESTED	0.001165	0.000994
STORED PRC	0.000000	0.000000
UDF	0.000000	0.000000
TRIGGER	0.000000	0.000000
PAR.TASKS	0.000000	0.000000
SECP CPU	0.000587	N/A
SE CPU TIME	0.000000	0.000000
NONNESTED	0.000000	0.000000
STORED PROC	0.000000	0.000000
UDF	0.000000	0.000000
TRIGGER	0.000000	0.000000
PAR.TASKS	0.000000	0.000000

The fields

- *SECP CPU is the zIIP eligible work that ran on a general purpose CP (.000587).*
- *SE CPU TIME represents the CPU time consumed on the specialty engine (in this case none.)*

According to the table on page 3, the workload can only be redirected to a zIIP since the distributed workload runs in an enclave SRB.

The SECP CPU time is included in the CP CPU Time total, so the percentage that could have been redirected to the zIIP is:

(SECP CPU/CP CPU)*100) or 50.3% = ((.000587/.001165)*100)

RMF Workload Activity Report

```
-TRANSACTIONS-  TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE----  --SERVICE TIMES--  ---APPL %---  -----STORAGE-----
 AVG      28.08  ACTUAL                22  SSCHRT 547.0  IOC          0  CPU        432.139  CP    138.85  AVG           0.00
 MPL      28.08  EXECUTION             22  RESP     0.7  CPU     119006K  SRB          0.000  AAPCP   0.00  TOTAL         0.00
 ENDED   367849  QUEUED                 0  CONN     0.1  MSO          0  RCT          0.000  IIPCP  69.88  SHARED        0.00
 END/S  1181.92  R/S AFFIN              0  DISC     0.5  SRB          0  IIT          0.000
 #SWAPS       0  INELIGIBLE             0  Q+PEND   0.1  TOT     119006K  HST          0.000  AAP     0.00  --PAGE-IN RATES--
 EXCTD        0  CONVERSION             0  IOSQ     0.0  /SEC    382373   AAP          0.000  IIP     0.00  SINGLE        0.0
 AVG ENC  28.08  STD DEV               16                                IIP          0.000                BLOCK         0.0
 REM ENC   0.00                                           ABSRPTN    14K                              SHARED        0.0
 MS ENC    0.00                                           TRX SERV   14K  PROMOTED     0.000          HSP           0.0
```

Under the "APPL %" column, the total CPU usage was 138.85% (CP) and 69.88% for IIPCP. For this workload a zIIP was not in place, so this activity ran on the general CP. The ratio that redirection would have occurred was 50.3% = ((69.88/138.85)*100) because IIPCP time is included in the total CP time.

As can be determined from the above projection, DB2 DRDA zIIP redirection can be 50%. 50% estimation is derived from RMF IIPCP/CPU or OMPE accounting trace class1 SECP CPU/ CP CPU. Both RMF and OMPE Accounting trace show the same redirection percentages.

Conclusions

With new accounting support, users can measure the redirection across all existing specialty engines. The accounting fields in DB2 reflect consumption of both zIIP and zAAP specialty engines at the application level. They can estimate the zIIP redirection before one is actually installed. The RMF Workload Activity Report in conjunction with PROJECTCPU should be used for estimation of specific specialty engines. As the measurements demonstrate, the redirection rate is also affected by z/OS release level and workload mixture in addition to XML document characteristics.

As the measurements were run under a strictly controlled laboratory environment and with minimum application logic, the results should not be viewed as an estimate of usage for the average customer workload. These measurements should not be used for sizing purposes as it is expected that a customer workload will show different results. Nevertheless, it shows the approximate redirection possible for this workload at this DB2 service level.

General guidelines for evaluation of specialty engines usage on XML workload are:

- *Up to 100% zAAP redirection can be utilized when z/OS XML System Services is invoked through TCB mode. z/OS XML System Services is invoked from inserts and updates (without validation), XML LOAD utility, and XMLPARSE function.*
- *The larger and more complex the document is, the more CPU is consumed by z/OS XML System Services. When the size and number of nodes of the documents increase, the redirection percentage could increase accordingly, up to about 60%, according to laboratory measurements. Beyond that, there is no additional redirection because the corresponding DB2 processing to construct the internal representation of the XML document increases at the same time.*
- *As more XML indexes are defined in XML tables, the percentage of redirection will be lower, as there is then more index processing within DB2 that is not related to z/OS XML System Services.*
- *Consider zAAP redirection*
  - *For workloads with heavy LOAD utility invocation with XML data, as zAAP provides significant CPU redirection in XML Load utility. The redirection rate is greater as the size of the document increases when few indexes exist.*
  - *When the size of document is larger than 100K and insert/update is a significant portion of the workload that executes in TCB mode.*
- *For a distributed XML workload with high read rate, DB2 DRDA zIIP likely will provide the best potential saving compared to local execution.*
- *XML Query does not run in parallelism and cannot be eligible for complex query zIIP offload, but still takes advantage of zIIP redirection when it is executed from DRDA.*
- *z/OS XML System Services performance has improved in z/OS Version 1.9. As a result, the CPU consumed in z/OS XML System Services is reduced, and the redirection percentage is lower than that of prior z/OS releases.*
- *When encoding conversion is necessary, the percentage of redirection will be lower. DB2 always uses UTF-8 to process XML data. If the source XML that is inserted/updated or loaded is not in UTF-8, z/OS conversion services will be invoked to convert it to UTF-8.*

These guidelines are for general purpose only. Individual cases should be evaluated to project accurate estimation.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. No assurance is given that an individual user will achieve throughput or performance improvements equivalent to the ratios that may be set forth in this paper.

Future Direction

The preview announcement on February 27, 2008, for z/OS Version 1.10 added additional XML support, part of which follows (italics added for emphasis):

In z/OS V1.10, IBM intends to provide these functions in z/OS XML System Services:

- Additional zIIP exploitation. z/OS XML System Services plans include additional zIIP exploitation, specifically enabling all z/OS *XML parsing in enclave SRB mode to be eligible for zIIP. For example, with respect to DB2, z/OS XML processing may be partially directed to zIIPs when utilized as part of a distributed request (like DB2 DRDA) today. This enhancement can help further benefit DB2 pureXML workloads by optionally directing all z/OS XML System Services parsing that is executed in enclave SRBs to the zIIP.* This function is planned to be available on z/OS V1.8 and V1.9 with PTF for APAR OA23828. Delivery of this function satisfies the statements of direction in Hardware Announcement 107-190, dated April 18, 2007, and Software Announcement 207-175 dated August 7, 2007.
- A validating parser. Validation support is designed to allow a program to determine whether an XML document meets the requirements expressed in an XML Schema Definition (XSD). z/OS XML System Services plans to add validating parsing. z/OS XML System Services validating parsing workload is to be eligible for zIIP and zAAP as well. Delivery of this function satisfies the statements of direction in Hardware Announcement 107-190, dated April 18, 2007, and Software Announcement 207-175, dated August 7, 2007.

The full text of the announcement (208-042) can be found at:
http://www-01.**ibm.com**/common/ssi/cgi-bin/ssialias?subtype=ca&infotype=an&appname=iSource&supplier=897&letternum=ENUS208-042

Special Notice

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While IBM may have reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environments do so at their own risk

Any performance data contained in this document were determined in various controlled laboratory environments and are for reference purposes only. Customers should not adapt these performance numbers to their own environments as system performance standards. The results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Appendix

Relevant DB2 9 APARs used in the measurements

PK47594	XML Load performance improvement
PK50575	zAAP accounting
PK51571, PK51572, PK51573:	XMLTABLE support
PK57158	XML index access path improvement

DB2 Accounting fields for specialty engines

The main fields discussed in this paper:

- *SECP CPU reports the zIIP-eligible work that ran on a general purpose CP.*
  – Either the amount of zIIP eligible that ran on a general purpose CP because there was no zIIP configured
  – Or the amount of zIIP workload that exceeded the capacity of the existing specialty processors
  Its DB2 trace record field name is QWACZIP_ELIGIBLE,
- *SE CPU TIME reports the specialty engine (both zAAP and zIIP) CPU time consumed on all specialty engines. Its DB2 trace record field name is QWACCLS1_ZIIP.*

Statistics for zAAP-eligible work that runs on a general purpose CP are not available to DB2.

This record snippet is found in DSN910.SDSNMACS (DSNDQWAC), the accounting mapping macro. It is provided with DB2 APAR PK50575.

```
QWACBJST DS     XL8      /*Beginning CPU time for all environments   */
*                        /*(CICS, IMS, TSO, etc.). This CPU time     */
*                        /*does not include CPU consumed on an IBM   */
*                        /*specialty engine.                         */
...
QWACEJST DS     XL8      /*Ending CPU time for all environments      */
*                        /*(CICS, IMS, TSO, etc.). This CPU time     */
*                        /*does not include CPU consumed on an IBM   */
*                        /*specialty engine.                         */
...
QWACAJST DS     CL8   /* Accumulated CPU time consumed while        */
*                     /* executing in DB2.  This CPU time does not  */
*                        /* include CPU consumed on an IBM specialty   */
*                        /* engine. In addition, this time does not    */
*                        /* include CPU consumed while processing SQL  */
*                        /* statements in a stored procedure.          */
...
QWACSPCP DS      CL8      /* The accumulated CPU time used to        */
*                         /* satisfy stored procedure requests       */
*                         /* processed in a DB2 stored procedure/    */
*                         /* WLM address space.  SQLP times are      */
*                         /* included in this time IF the SQLP       */
*                         /* was called on a nested task and was     */
*                         /* not invoked by the main application     */
```

```
*                       /* execution unit. This time does not      */
*                       /* include CPU consumed on an IBM specialty*/
*                       /* engine.                                 */
QWACA302 EQU    *
QWACSPTT DS     CL8     /* The accumulated CPU time consumed       */
*                       /* in DB2 processing SQL statements        */
*                       /* issued by stored procedure(s) processed */
*                       /* in a DB2 stored procedure/WLM           */
*                       /* address space.  This time also          */
*                       /* includes in DB2 time needed to          */
*                       /* connect and disconnect the SP task for  */
*                       /* non-SQLP stored procedures.             */
*                       /* SQLP times are included in this time IF */
*                       /* the SQLP was called on a nested task and*/
*                       /* was not invoked by the main application */
*                       /* execution unit. This time is a subset   */
*                       /* of QWACSPCP and does not include CPU    */
*                       /* CPU consumed on an IBM specialty        */
*                       /* engine.                                 */
...
QWACUDCP DS     CL8     /* The accumulated CPU time used to        */
*                       /* satisfy UDF requests                    */
*                       /* processed in a DB2 stored procedure/    */
*                       /* WLM address space. This time does not   */
*                       /* include CPU consumed on an IBM specialty*/
*                       /* engine.                                 */
QWACA303 EQU    *
QWACUDTT DS     CL8     /* The accumulated CPU time consumed       */
*                       /* in DB2 processing SQL statements        */
*                       /* issued by UDF(s) processed              */
*                       /* in a DB2 stored procedure/WLM           */
*                       /* address space.  This time also          */
*                       /* includes in DB2 time needed to          */
*                       /* connect and disconnect the udf task.    */
*                       /* This time does not include CPU consumed */
*                       /* on an IBM specialty engine.             */
...
QWACTRTT DS     CL8     /* The accumulated CPU time consumed while */
*                       /* executing triggers on the main          */
*                       /* application execution unit.  This time  */
*                       /* does not include CPU consumed on an IBM */
*                       /* specialty engine.                       */
...
QWACTRTE DS     CL8     /* The accumulated CPU time consumed while */
*                       /* executing triggers on a nested task.    */
*                       /* This time does not include CPU consumed */
*                       /* on an IBM specialty engine.             */
...
QWACZIIP EQU *
QWACCLS1_zIIP     DS  CL8 /* Accumulated CPU time consumed while    */
*                         /* executing on an IBM specialty         */
```

```
*                          /* engine in all environments.           */
QWACCLS2_zIIP     DS  CL8 /* Accumulated CPU time consumed while    */
*                          /* executing in DB2 on an IBM specialty   */
*                          /* engine.                                */
QWACTRTT_zIIP     DS  CL8 /* Accumulated CPU time consumed executing*/
*                          /* triggers on the main application       */
*                          /* execution unit on an IBM specialty     */
*                          /* engine.                                */
QWACZIIP_ELIGIBLE DS  CL8 /* Accumulated CPU executed on a standard */
*                          /* CP for zIIP-eligible work.
QWACSPNF_zIIP     DS  CL8 /* Accumulated CPU time consumed executing*/
*                          /* stored procedure requests on the main  */
*                          /* application execution unit on an IBM   */
*                          /* specialty engine. Since these SPs run  */
*                          /* entirely within DB2, this time         */
*                          /* represents class 1 and class 2 time.   */
QWACUDFNF_zIIP    DS  CL8 /*   ** RESERVED FOR FUTURE         **    */
QWACNF EQU *
QWACSPNF_ELAP     DS  CL8 /* Accumulated elapsed time consumed      */
*                          /* executing stored procedure requests on */
*                          /* the main application execution unit.   */
*                          /* Since these SPs run entirely           */
*                          /* within DB2, this time represents class */
*                          /* 1 and class 2 time.                    */
QWACSPNF_CP       DS  CL8 /* Accumulated CPU time consumed executing*/
*                          /* stored procedure requests on the main  */
*                          /* application execution unit. This time  */
*                          /* does not include CPU consumed on an    */
*                          /* IBM specialty engine.  Since these SPs */
*                          /* run entirely within DB2, this time     */
*                          /* represents class */
*                          /* 1 and class 2 time.
QWACUDFNF_ELAP    DS  CL8 /*   ** RESERVED FOR FUTURE         **    */
QWACUDFNF_CP      DS  CL8 /*   ** RESERVED FOR FUTURE         **    */
QWACSP_CLS1se     DS  CL8 /* The accumulated CPU time used to       */
*                          /* satisfy stored procedure requests      */
*                          /* processed in a DB2 stored procedure/   */
*                          /* WLM address space while executing on   */
*                          /* an IBM specialty engine.  SQLP times   */
*                          /* are included in this time IF the SQLP  */
*                          /* was called on a nested task and was    */
*                          /* not invoked by the main application    */
*                          /* execution unit.                        */
QWACSP_CLS2se     DS  CL8 /* The accumulated CPU time consumed      */
*                          /* in DB2 processing SQL statements       */
*                          /* issued by stored procedure(s) processed*/
*                          /* in a DB2 stored procedure/WLM          */
*                          /* address space while executing on an    */
*                          /* IBM specialty engine. SQLP times are   */
*                          /* included in this time IF the           */
*                          /* SQLP was called on a nested task and   */
*                          /* was not invoked by the main application*/
```

```
*                          /* execution unit. This time is a subset  */
*                          /* of QWACSP_CLS1se                       */
QWACUDF_CLS1se    DS CL8   /* The accumulated CPU time used to        */
*                          /* satisfy UDF requests                   */
*                          /* processed in a DB2 stored procedure/   */
*                          /* WLM address space while executing on   */
*                          /* an IBM specialty engine.               */
QWACUDF_CLS2se    DS CL8   /* The accumulated CPU time consumed      */
*                          /* in DB2 processing SQL statements       */
*                          /* issued by UDFs processed               */
*                          /* in a DB2 stored procedure/WLM          */
*                          /* address space while executing on an    */
*                          /* IBM specialty engine. This time is a   */
*                          /* subset of QWACUDF_CLS1se               */
QWACTRTE_se       DS CL8   /* The accumulated CPU time consumed on   */
*                          /* an IBM specialty engine while          */
*                          /* executing triggers on a nested task.   */
QWACEND  DS     0C
```

The formulas for Class 1 and Class 2 times are:

- *Class 1 CP = (qwacejst - qwacbjst) + qwacspcp + qwacudcp + qwactrte*
- *Class 1 specialty engine = qwaccls1_ziip + qwacsp_cls1se + qwacudf_cls1se + qwactrte_se*
- *Class 2 CP = qwacajst + qwacsptt + qwacudtt + qwactrte*
- *Class 2 specialty engine = qwaccls2_ziip + qwacsp_cls2se + qwacudf_cls2se + qwactrte_se*

The member DSN910.SDSNSAMP (DSNWMSGS), which describes the trace records, will be updated following the closure of this service.

Insert workload background and measurement results

Orders are represented using FIXML 4.4. FIXML is an industry standard XML schema for trade-related messages such as buy or sell orders (see http://www.fixprotocol.org/). Order documents have many attributes and a high ratio of nodes to data. Typical document sizes are 1 to 2 KB for Order and 4 to 20 KB for combined Customer/Account documents. There are hundreds of optional attributes and elements with typically only a small subset present in any given document instance to simulate real world FIXML usage.

For INSERT1, 20 concurrent users insert 6 million Order documents into an XML column in the ORDER table. A commit is issued after every 10 documents are inserted. There are two XML indexes defined on the ORDER table.

For INSERT2, 20 concurrent users insert 6 million CustAcc documents into an XML column defined in the CustAcc table. A commit is issued after every 10 documents insert. There are two XML indexes defined on the CustAcc table.

It should be noted that TPoX, as defined, requires a commit after every transaction with a mixed workload. Since the original measurement did not use queries, this was not done. For the mixed workload where queries were run with the inserts, a commit was taken every insert and there was more overhead due to commit in this instance.

The DDL used for the both inserts can be found in a downloadable file https://sourceforge.net/project/showfiles.php?group_id=185925. Following the download the DDL is located under directory TPoX/DB2/ddl.

While the inserts could have been run locally, in the interest of time they were run using distributed threads. As there was no actual zIIP or zAAP on the System z9 that was used, the SQL/XML code was the same regardless of whether it originated locally or remotely. The internal tool measured the potential redirection capability.

Statistics	INSERT1 into ORDER Table	INSERT2 into CustAcc Table
LPAR CPU utilization	70%	76.45%
Number of concurrent thread	20 Threads	20 Threads
Average XML document size (input size)	1.49KB	5.46KB
Number of XML user indexes	2	2
Number of XML insert per second (average)	3688 insert	2363 insert
Total time	31987 seconds	50155 seconds
DB2 class1 average elapsed time (per commit)	52.658 (ms)	77.565 (ms)
DB2 class1 average CPU time (per commit)	4.814(ms)	7.909 (ms)
DB2 class2 average elapsed time (per commit)	24.096(ms)	60.478 (ms)
DB2 class2 average CPU time (per commit)	4.361(ms)	7.452 (ms)
DB2 total CPU time including MSTR, DBM1, and IRLM (per commit)	5.092(ms)	8.265 (ms)
z/OS XML System Services CPU usage in LPAR	**13.80%**	**32.37%**

Observations

For the TPoX insert workload, 13.8% to 32.4% of total CPU usage was observed as z/OS XML System Services usage when CPU utilization of the LPAR was 70% to 76%. z/OS XML System Services CPU usage varies depending on the size and complexity of the documents. The CustAcc documents are larger, more complex, and contain a greater number of nodes compared to those of the ORDER documents.

TPoX queries used in mixed workload benchmark

The SQL used for the TPoX queries for this workload:

get_order

```
-------------------------------------------------------------------------
-- Retrieve an order with the specific ID.
-------------------------------------------------------------------------
SELECT XMLQUERY
(
'declare namespace o="http://www.fixprotocol.org/FIXML-4-4";
$odoc/o:FIXML/o:Order
'
PASSING odoc AS "odoc"
)
FROM order3
WHERE XMLEXISTS
(
'declare namespace o="http://www.fixprotocol.org/FIXML-4-4";
$odoc/o:FIXML/o:Order[@ID=$id]
'
PASSING odoc AS "odoc", cast (? as varchar(10)) as "id"
)
;
```

get_security

```
--get_security
-------------------------------------------------------------------------
--  Return a security having the specified Symbol.
-------------------------------------------------------------------------
SELECT XMLQUERY
(
'declare default element namespace "http://tpox-benchmark.com/security";
$sdoc/Security
'
PASSING sdoc AS "sdoc"
)
FROM security3
WHERE XMLEXISTS
('
declare default element namespace "http://tpox-benchmark.com/security";
$sdoc/Security[Symbol=$sym]
'
PASSING sdoc AS "sdoc", cast(? as varchar(10)) as "sym"
)
;
```

Customer Profile

```
--customer_profile
-------------------------------------------------------------------------
--  Return a customer profile for the specified customer.
-------------------------------------------------------------------------
SELECT XMLELEMENT (name "Customer_Profile",
  XMLNAMESPACES (DEFAULT 'http://tpox-benchmark.com/custacc'),
          "@id", "Name", "DateOfBirth", "Gender",
          "CountryOfResidence", "Languages",
          "Addresses", "EmailAddresses"
     ) AS CustomerProfile
 FROM custacc3 C,
      XMLTABLE(
      XMLNAMESPACES (DEFAULT 'http://tpox-benchmark.com/custacc'),
               '$cadoc/Customer'
```

```
                PASSING C.cadoc AS "cadoc"
                Columns "@id" XML,
                        "Name" XML,
                        "DateOfBirth" XML,
                        "Gender" XML,
                        "CountryOfResidence" XML,
                        "Languages" XML,
                        "Addresses" XML,
                        "EmailAddresses" XML) X
 WHERE XMLEXISTS
 (
 'declare default element namespace
 "http://tpox-benchmark.com/custacc";
 $cadoc/Customer[@id=$id]'
 PASSING cadoc AS "cadoc", cast (? as double) as "id"
 )
;
```

Account Summary

```
--account_summary
-- As the first step, we extract all the accounts together.

SELECT
    XMLELEMENT(
        Name "Customer",
        XMLNAMESPACES
              (DEFAULT 'http://tpox-benchmark.com/custacc'),
        id,
        Name,
        XMLELEMENT(Name "Customer_Securities",
          (SELECT XMLAGG(A.C1 )
           FROM
              (SELECT

   XMLAGG(

     XMLELEMENT(Name "Account",
        XMLATTRIBUTES(Y.Balance as "BALANCE"),
        Account_id,
        XMLELEMENT(Name "Securities",
           SecurityName
         )
       )
     )
   FROM
      XMLTABLE(
        XMLNAMESPACES(DEFAULT
          'http://tpox-benchmark.com/custacc'),
                     '$accts/Account'
                PASSING X.Accounts AS "accts"
                COLUMNS
                  Balance VARCHAR(30) PATH
                      'Balance/OnlineActualBal',
                  Account_id XML PATH '@id',
                  SecurityName XML PATH 'Holdings/Position/Name'
              ) Y
            )  AS A(C1)
         )
       )
     )
```

```
   FROM custacc3 C,
      XMLTABLE(
        XMLNAMESPACES(DEFAULT 'http://tpox-benchmark.com/custacc'),
          '$cadoc/Customer'
          Passing C.cadoc as "cadoc"
          COLUMNS
             id  XML PATH '@id',
             Name XML PATH 'Name',
             Accounts XML PATH 'Accounts'
       ) X
   WHERE XMLEXISTS
   (
   'declare default element namespace "http://tpox-benchmark.com/custacc";
   $cadoc/Customer[@id=$id]'
   PASSING cadoc AS "cadoc", cast (? as integer) as "id"
   )
   ;
```

get_security price

```
--get_security_price
-------------------------------------------------------------------------
-- Print an open price of a particular security in a user-friendly text
-- message.
-------------------------------------------------------------------------
SELECT
    XMLELEMENT(
         NAME "print",
         XMLNAMESPACES (DEFAULT 'http://tpox-benchmark.com/security'),
         'The open price of the security',
         XMLQUERY('declare namespace s="http://tpox-benchmark.com/security";
                   $sdoc/s:Security/s:Name/text()'
                   PASSING sdoc AS "sdoc"
         ),
         ' is ',
         XMLQUERY('declare namespace s="http://tpox-benchmark.com/security";
                  $sdoc/s:Security/s:Price/s:PriceToday/s:Open/text()'
                  PASSING sdoc AS "sdoc"
         ),
         ' dollars'
    )
FROM security3
WHERE XMLEXISTS
(
'declare namespace s="http://tpox-benchmark.com/security";
$sdoc/s:Security[s:Symbol=$sym]
'
PASSING sdoc AS "sdoc", cast (? as varchar(10)) as "sym"
)
;
```

IBM Corporation
New Orchard Rd.
Armonk, NY 10504
U.S.A

Produced in the United States of America
03/08

ZSW03034-USEN-00