SKYTIDE

Skytide Analytical Platform for DB2 9 Data

Large Volume Business Analytics for XML Data

September, 2007

Skytide
1820 Gateway Drive
San Mateo, CA 94404
650.292.1900
www.skytide.com

SKYTIDE

Table of Contents

Co-Authored By:

Tom Tortolani
V.P. Product Management
Skytide, Inc.

Vijay Bommireddipalli
Advisory Software Engineer
IBM Silicon Valley Lab
Data Servers Solutions
IBM

Matthias Nicola
Sr. Software Engineer
IBM Silicon Valley Lab
DB2/XML Performance & Enablement
IBM

SKYTIDE

Skytide Analytical Platform for DB2 9

Large Volume Business Analytics for XML Data

Overview

The Skytide Analytic Server provides reporting and analysis capabilities for structured relational data and semi-structured data, such as XML, at the same time. This provides a complete view of your business behavior and enables better decisions and improved processes. IBM has introduced the pureXML feature in DB2 9 that allows XML data to be stored and indexed in a hierarchical format in the database. This makes XML a first-class data type in DB2, and a column of type XML can store any well-formed XML document. Once the data is stored in the database as XML, it can be queried very efficiently, as the XML is not parsed at runtime for query evaluation.

Although Skytide works equally well with relational data, the focus of this paper is on analytic models for large scale XML data, as represented by the XML database transaction processing benchmark ("TPoX"). TPoX is a financial application scenario used to evaluate the performance of XML database systems, focusing on XQuery, and SQL/XML, XML storage & indexing, XML updates, and other XML features. Its data is based on the widely accepted FIXML standard for financial information exchange. (For more information on TPoX, refer to the resources section at the end of this document.)

Based on the TPoX scenario, we performed a series of measurements of building multidimensional data cubes and running complex analytical queries. These measurements show the performance benefits of DB2 pureXML as the data source for the Skytide Business Analytics Server, as compared to XML in CLOB columns or the file system. The combination of Skytide with pureXML storage in DB2 is up to 35 times faster than processing CLOBs when building analytic models, and up to 85 times faster than processing XML files from a file system. This enables analysis of large volumes of XML data at high performance for gaining valuable business insight.

Background

The last several years have seen an explosive growth of XML data. Practically every industry has adopted XML as the de facto data model to exchange information. XML industry standards such as FIXML, FPML, ACORD, XBRL, etc. have emerged and have resulted in large volumes of transactional data in XML format. This data contains rich information about the characteristics and performance of the underlying business. Companies realize that it is vital for their success to analyze this data with sophisticated analytical tools.

For over 15 years, Business Analytics software has provided great benefit to many organizations world-wide. However, this software was designed to work well with highly structured, normalized data. Given the flexibility of the XML data model to include varying levels of descriptive information within each document, there is a lot of data available for analysis which is difficult or impossible to force into a normalized record/field format. Skytide is specifically designed to build analytic models on such variable data structures.

Abstract

This article describes the benefits and scalability of the Skytide Analytical Platform™ when used in conjunction with IBM DB2 9. Most analytics servers require data to be in a structured relational format before being able to analyze it. Skytide is a next generation business analytics server that is optimized for structured, unstructured, and extensible data. In particular, XML data can be analyzed without mapping it to a relational format. This makes Skytide a great fit for DB2 pureXML which stores XML in its inherent hierarchical format. This article describes how and why DB2 pureXML provides significant performance benefits as a data source for Skytide.

At the core of the Skytide platform is a multi-dimensional analysis engine. Multi-dimensional data, also known as a data cube, is the best conceptual data model for business users to understand, analyze, and report on large volumes of complex data. Data complexity is typically twofold. First, the format of the source data may be more complex than a standard record/field format. Secondly, most business analytics requires aggregation of data through multiple levels of dimensions and additional calculations for meaningful analysis. Deploying a Skytide data model over source data empowers end users to easily create their own data models and analytic queries. Skytide is a multi-user application and supports cube building and incremental updating while users are querying the system.

Skytide is an open platform that works with various reporting tools such as Microsoft Excel, Crystal Reports, Jasper Reports, IBM DB2 Alphablox and other reporting tools that support JDBC.

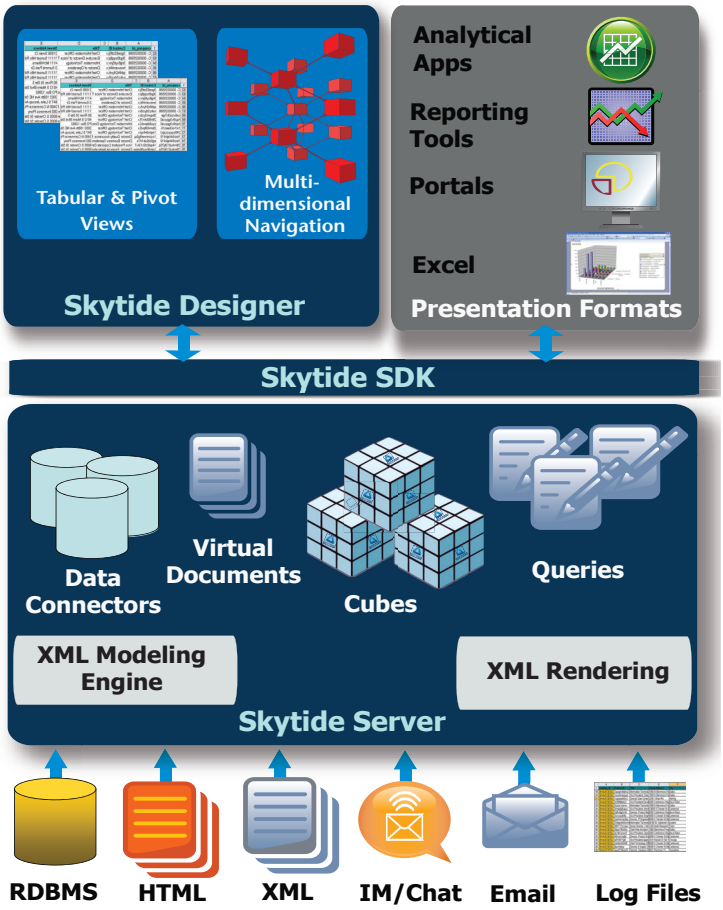

Figure 1: Skytide is a next generation XML-based analytical platform that performs analyis directly on XML data, without requiring time-consuming transformation of XML data into a data warehouse.

XML Storage Alternatives: DB2 pureXML, CLOB, File Systems

XML applications have various options for storing XML. Simple file system storage is one option. However, the need for efficient search, access control, scalability, data integrity, etc. drives many users towards databases. Before DB2 9 pureXML, CLOB (character large object) columns were a common choice for XML storage. A fundamental difference between CLOB storage and pureXML processing lies in XML parsing and its significant impact on insert and query performance.

If XML documents are inserted into CLOB columns, they are inserted as unparsed text. Without XML parsing, the structure of the XML documents is entirely ignored. This prevents the database from performing intelligent and efficient search & extract operations on the stored text objects. The only remedy is to invoke an XML parser at query execution time to "look into" the XML documents. The XML parsing causes significant CPU consumption, and often leads to low performance. Only a blind full document retrieval, which ignores the internal XML structure, can quickly read XML documents from CLOB columns.

The pureXML technology in DB2 9 parses XML documents at insert time and never at query time. The XML documents are stored and queried in a parsed format, i.e. a tree structure of nodes, which is different from the textual representation of XML. Search and extract operations can therefore be performed without XML parsing, which is a significant performance benefit

XQuery vs. Skytide

Business analytics offer a way to turn large amounts of data into meaningful information for end user analysis. However, it was not designed to be optimal for every query requirement. In some cases, data is better queried directly from the data source. A typical example is detailed address information for individual customers. While it is possible to bring such information into Skytide, this type of data is not typically required for business analytics. For example, knowing the city or state of a customer is beneficial for data analysis where data is analyzed by geographic aggregates. However, the need to know a street address of a particular a customer is typically not a business analytics problem.

The XQuery language is often not the best choice to express and execute complex analytical queries. One reason is that XQuery does not have an explicit GROUP BY construct for easy aggregation of data. Similarly, OLAP functions which have long been integrated in the SQL standard are still absent from XQuery. This makes it difficult for users to write

analytical queries in XQuery, and difficult for database systems to optimize and execute them efficiently. Another option is to use SQL/XML to combine XQuery and SQL such that XQuery reads atomic data values from the XML data, and advanced SQL functions are used to aggregate or summarize these values. However, such queries are very complex and typically do not deliver the same performance as querying an in-memory data cube.

The Skytide multi-dimensional data cube is the optimal data model for Business Analytics. Queries are designed to be easy for a non-technical user to create in various tabular and cross tabular formats, as shown below.

Test Scenario

The TPoX benchmark models a financial transaction processing application that uses XML data. In this scenario, customers have accounts and place orders to buy or sell shares of securities such as stocks, bonds, or mutual funds. This information is represented by three collections of XML documents:

- ***Customers*** – one XML document per customer that contains profile information such as name, gender, address, preferred customer status, etc. but also account information with holdings and balances.
- ***Orders*** – one XML document for each financial transaction, in this case the buy or sell of a security. An order contains information such as the account number, transaction date, security symbol purchased or sold, order amount, and more.
- ***Securities*** – an XML document for each individual market Security (stock, bond, or mutual fund) that contains information such as security symbol, security type, sector, pricing, etc.

Naturally, this data lends itself for interesting analytical evaluation, e.g. to understand customers' trading behavior based on various demographics. For comparative performance tests, we stored all of this XML data in three different ways: in DB2 pureXML, in DB2 CLOB columns, and in the file system - all on the same hardware. Then we used Skytide to build data cubes based on each of the three data sources.

We conducted these tests for two different scale factors:

- **10GB of raw XML data,** representing 600 thousand customers and 3 million orders across 20 thousand Securities.
- **100GB of raw XML data,** representing 6 million customers and 30 million orders across 20 thousand Securities.

For both scale factors we tested two types of cubes: (1) using all data of the respective scale factor, and (2) considering information related to US customers only. The second type of cube allows DB2 to use its pureXML features to efficiently perform the required filtering. In summary, we end up with the following test matrix.

	10 GB		100 GB	
	All Data	US Only	All Data	US Only
pure XML				
CLOB				
File System				

The results are described in the remainder of this paper. One additional test was to build 5 country-specific cubes in parallel.

Skytide Cube Design

Cubes are formed by defining dimensions, which are used in analytic models to organize related data (measures) into categories that are easy for the user to understand for query and reporting needs. Dimensions typically contain multiple levels of aggregation. For example, a Period dimension contains daily, monthly, and yearly levels. The actual values of a dimension are referred to as members. For example, a Customer dimension contains 100 members (Customers). Skytide cubes offer a large range of reporting and analysis possibilities. For example, Order data by country by month by security type that compares premium to non-premium customers can easily be queried and analyzed for trends and outlying activity. In fact, numerous permutations of analytic queries that compare dimensions by other dimensions are easily performed on cube data.

For comparative performance measurements, all of the tested cubes had the following dimensions and measures:

Dimensions	Levels
Customer	Customer ID, Account #, Premium customer
Geography	State, Country
Period	Day, Month, Year
Security	Symbol, Type, Industry
Orders	Order ID
Measures	# Orders, Order Amount, Account Balance, Last Trade Amount

Skytide provides a Designer UI, that lets the application designer point to various data sources, relate the data sources, and define Cubes. As illustrated in Figure 1 all the cubes were defined in the Skytide Designer UI:

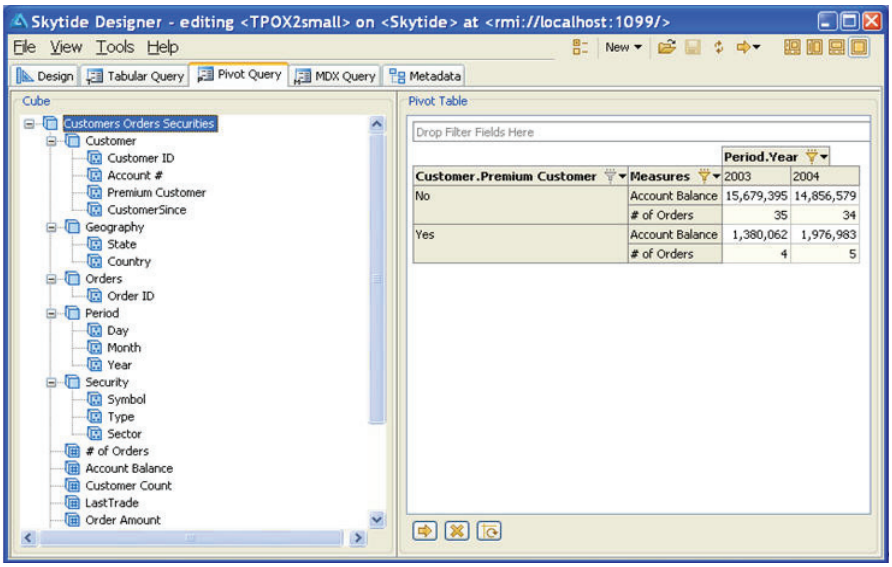

Figure 1: Skytide Designer User Interface

Once a cube is defined it is easy to form various queries by dragging dimensions from the cube into any row or column location in the right side pane of the Skytide Designer window.

Cube Design Considerations

The level of detailed data that is included in a cube is usually dependant on factors such as reporting requirements, source data latency, cube latency, the amount of historic data, etc. Our main test cubes contain world-wide daily transaction data from a two year period. Including this much detail data in a single cube is possible, but alternative cube designs are possible depending on the reporting requirements in a given case. For example, data could be partitioned into smaller specialized cubes to better suit the users' analytical needs. Also, there are reporting scenarios where detailed individual customer level data is not required but only aggregated data is of interest. Skytide does not require all detailed data to be included in a cube in order to calculate dimensional aggregates. Hence, data by individual Customer ID could be omitted from our cubes which would greatly improve the performance of cube building and certain queries.

The amount of the detail data included in our cubes is very large for a single analytical model. But, the test results show that Skytide is able to scale to these large data volumes and process data into various cubes within time frames that are well within acceptable windows for a typical business process.

A practical way to partition a data set is by time or by geography. For example, detailed transaction level data can be modeled in a given cube, but only for specific geographies. A practical use case is the partitioning of data into country specific cubes. We'll describe our tests with country-specific cubes to illustrate theses performance trade-offs in cube design

Performance Results

Cube Building

In order to make the XML data available for reporting and analysis, Skytide directly loads the relevant data values during the cube build process. This process loads and links only the XML elements and attributes that are required by the Skytide cube definition. The time to build a cube over XML data in Skytide needs to be weighed against the cost of the most common alternative, i.e. the combined cost of the following steps: (1) Developing a complex mapping from XML to a relational schema, possibly incurring data loss, (2) shredding large amounts of XML data into the relational tables, and (3) defining and building a data cube over the relational data with a traditional BI product.

Skytide with DB2 9 eliminates the need for the first two steps, a significant saving of labor and processing time. Another major advantage of this combination is how efficiently this cube is built when it is built over selective data. Skytide is able to leverage DB2's XQuery and SQL/XML support to pre filter and source only qualified documents which results in significantly lower cube build times as you will see in the following sections.

- ***Cube from 10GB XML data, US only***. This cube contains data for 12,885 orders across 2,577 US customers and 20,833 securities. The processing time directly from DB2 pureXML was about 2 minutes and significantly faster than using CLOBs (39 minutes) or the File System (2hrs50min).

  The 10GB XML source contains customer and order data for all countries but only the data for US customers is required for this cube. For XML data stored in DB2 pureXML, Skytide can pass an XML predicate to DB2 so that DB2 uses its XML index and query capabilities to

pass only US data to Skytide for cube building. This is a significant performance benefit. For XML data in CLOBs or the file system, Skytide needs to ingest all documents to find the US data which is then used for the cube. (Figure 2)

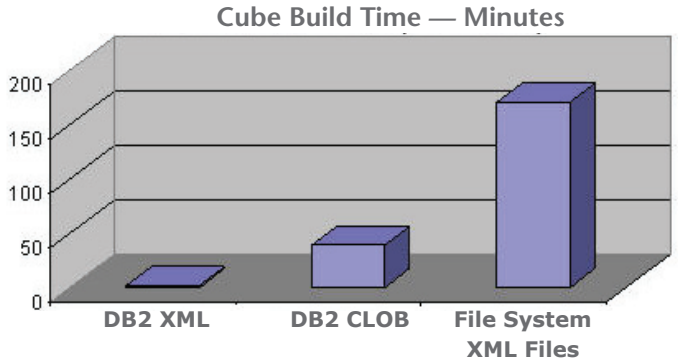

Figure 2: Cube building performance, US data from 10GB of XML

- ***Cube from 10GB XML data, all countries.*** This large cube contains data for 3 million Orders across 600,000 Customers and 20,833 Securities. The processing time directly from DB2 pureXML was 1hr26min as compared to 1hr45min from CLOBs and 2hr30min from the File System. (Figure 3)

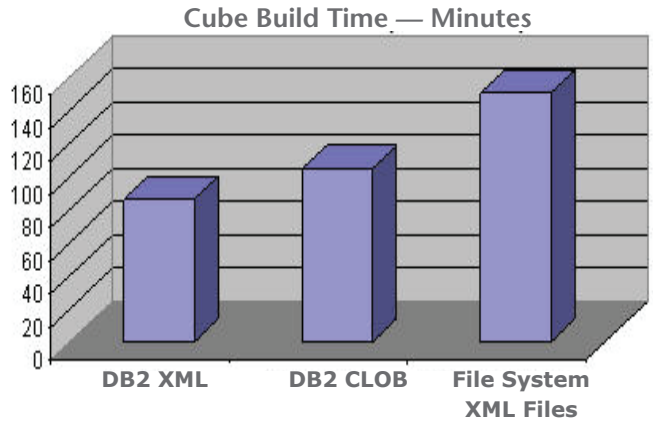

Figure 3: Cube building performance, 10GB of XML

- ***Cube from 100GB XML data, US only.*** This cube contains data for 129,191 Orders across 25,838 US Customers and 20,833 Securities. In addition to the US cube, four additional country cubes were defined and loaded in parallel. The query burden on DB2 was insignificant and the overall time to build five cubes simultaneously in Skytide was approximately the same as the processing time for a single cube. The processing time directly from DB2 pureXML was significantly faster than using CLOBs, i.e. 15 minutes vs. 9hrs20min. (Figure 4)

- ***Cube from 100GB XML data, all countries.*** This very large data set contains 30 million Orders across 6 million Customers, and 20,833 Securities. It took approximately 17 hours to build a cube directly from DB2 pureXML. Given the nature of the data, the business scenario for this cube would be that of a one-time large up-front historic data processing (2 years of daily orders) to create the cube. Then incremental batches of new transaction data get added to the cube, e.g. on a daily or hourly basis. Loading the data from CLOBs or XML files from the file system was not practical given the large amount of upfront processing time required.

Analytical Queries

One significant benefit of the data cube is the ability to easily perform ad-hoc queries and analysis in a point-and-click fashion with no query language. The dimensional model of the cube lets the user construct a virtually unlimited number of tabular and cross-tabular (pivot) query results.

Seven queries were benchmarked across all four cubes. These are representative queries that would commonly be created for data analysis by end users. In most cases, query performance is a matter of seconds (see Appendix). However, depending on the size of the cube, number of dimensions in the cube, and number of dimensional aggregates that are required for a given query request, some queries take longer.

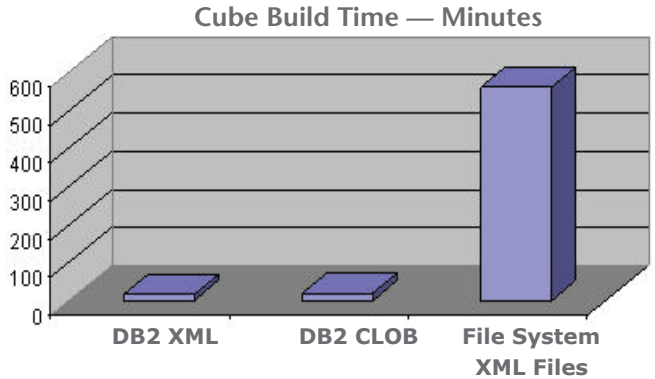

Figure 4: Cube building performance, US data from 100GB of XML

Cube Query 1 — C1 Max Stock Sector Order

This query produces aggregate order amount data for a given state (West Virginia), and stock security sector (Energy) in a tabular result set, as shown in the Skytide Designer figure below:

Geography.State	Security.Sector
West Virginia	Energy

Security.Symbol	Order Amount
CNPCX	15,538.47
AGSBX	11,085.22
PCBSX	9,966.77
BEGYX	8,844.97
TTFBX	8,590.46
STCHX	8,414.35
IMLAX	7,374.04
PABAX	7,240.69
SSCFX	6,944.61
TIIEX	6,756.53

This query was easy to build in Skytide by dragging cube dimensions onto an empty palette. In this case, order amounts are aggregated across all time periods. The query response time was approximately 1 second or less on all cubes. This particular query was designed to mimic the following candidate query from the TPoX benchmark shown below:

```
C1: max_stock_order
declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";
declare namespace s="http://tpox-benchmark.com/security";
declare namespace c="http://tpox-benchmark.com/custacc";
let $order :=
for $ss in db2-fn:xmlcolumn("SECURITY.SDOC")/s:Security[
s:SecurityInformation/s:StockInformation/s:Industry ="Energy"]
for $ord in db2-fn:xmlcolumn("ORDER.ODOC")/FIXML/Order[
Instrmt/@Sym= $ss/s:Symbol/fn:string(.)]
for $cs in db2-fn:xmlcolumn("CUSTACC.CADOC")/c:Customer[
c:Addresses/c:Address/c:State= "West Virginia"]/c:Accounts/c:Account[
@id =$ord/@Acct/fn:string(.)]
return $ord/OrdQty/@Cash
return string(max($order))
```

Skytide provides multiple advantages over this XQuery. Many users will find it easier to create such a query in Skytide's GUI rather than in XQuery notation. Secondly, the basic data filtering as well as the column sorting can easily be changed by mouse clicks on various column headings. This provides instant query results for different state and stock sectors, or different ordering, at no or negligible additional cost.

The XQuery in DB2 however would need to be re-rerun if results for a different state or stock sector are desired. This would be particularly expensive for the more complex queries that follow.

Cube Query 2 — Q7 Ascending Order Amounts

This query returns order details for a given customer in a tabular result set shown below:

Customer.Customer ID
10011

Period.Day	Orders.Order ID	Order Amount	Security.Symbol	LastTrade
2003-09-07	122291	9,839.96	PFBMX	118.0323
2004-01-07	112291	8,115.31	SRCCX	90.5875
2004-07-07	152291	7,118.03	SIGI	39.9681
2004-08-08	142291	5,115.31	AIGMX	115.3153
2004-11-07	132291	4,882.49	SEIMX	53.305

The query response time was 1 second or less across all benchmarked cubes. This particular query was designed to mimic the following TPoX benchmark query included below.

```
Q7: customer_max_order
declare default element namespace
"http://www.fixprotocol.org/FIXML-4-4";
declare namespace c="http://tpox-benchmark.com/custacc";
let $orderprice :=
for $ord in db2-fn:xmlcolumn("ORDER.ODOC")/FIXML/Order
for $cust in db2-fn:xmlcolumn("CUSTACC.CADOC")/c:Customer[
@id=1011 and
c:Accounts/c:Account/@id=$ord/@Acct/fn:string() ]
return $ord/OrdQty/@Cash
return max($orderprice)
```

Cube Query 3 — Top Stock Securities 2004

This query shows aggregate order amounts grouped by security symbol and for a given year, in descending order, shown here.

Query response times across the cubes were within 2 seconds for all but the 100GB XML Data Cube. See the Table of Query times in the appendix for the exact query times.

Period.Year	Security.Type
2004	Stock Fund

Security.Symbol	# of Orders	Order Amount
CRMMX	499,115	39,939,149.26
MCEFX	904,713	34,034,790.89
GVOBX	651,415	31,531,465.41
SAECX	251,412	31,531,425.09
PINOX	819,812	29,129,881.71
SPOIX	879,510	29,729,587.48
CWMCX	949,311	29,529,394.84
AQDIX	749,211	29,429,274.42
ECVSX	249,212	29,429,204.55
PVYYX	259,015	29,229,005.31
JSEBX	248,313	28,428,304.59
GEIDX	218,211	28,928,221.31
OCAAX	208,116	28,428,160.22
ODVCX	298,013	28,028,089.41
CABDX	217,811	27,927,801.13
HRITX	847,512	27,427,584.13
BTVBX	797,512	27,227,579.26
SSCCX	317,412	27,127,431.26
WSCYX	207,316	27,027,320.44

Cube Query 4 — Customer Account Activity

This query provides a cross-tabular view that contains order data by account, by order ID, and by date for a given customer ID, shown below.

Customer.Customer ID
10008

		Order Amount						
Account#	Order Id	2003-06-24	2003-12-05	2003-06-24	2003-12-05	2003-06-24	2003-12-05	2003-12-05
104930942	562288	4,069.25						
	572288		2,882.49					
	582288							5,069.25
	592288				4,069.25			
	602288					4,882.49		
104930943	682288						5,069.25	
104930944	882288			4,882.49				

This type of query, measuring multiple time period data going across the columns of a page, is a common business query used to view performance over time or for trending analysis. This type of query is typically too hard to express in a general database query language such as SQL or XQuery and usually does not perform as required without the cube as an optimized data model. Query response times across the cubes were within 2 seconds for all but the largest data cube built over 100 GB XML Data. See the Table of Query times in the appendix for the exact query times.

Cube Query 5 — Orders by Country by Month

This query provides a cross-tabular view of data aggregated across multiple dimensions. It contains order counts by country and by month:

	2003-06	2003-07	2003-08	2003-09	2003-10	2003-11	2003-12	2004-01	2004-02
Afghanistan	5	23	24	28	24	19	25	18	14
Albania	14	24	25	23	22	14	26	19	21
Algeria	9	27	26	19	27	24	17	15	16
American Samoa	10	21	27	19	16	18	21	15	24
Andorra	6	26	29	19	21	28	21	20	23
Angola	14	16	26	22	30	23	25	22	22
Anguilla	16	39	35	26	28	25	26	17	25
Antarctica	10	26	26	18	29	22	11	17	21
Antigua	13	29	27	24	35	29	34	21	20
Argentina	12	22	19	24	34	21	22	24	19
Armenia	7	32	18	21	19	25	24	27	30
Aruba	8	24	26	19	31	27	24	24	18
Australia	10	27	30	17	19	22	19	27	20
Austria	10	16	31	24	26	24	29	26	27
Azerbaijan	12	18	19	24	16	26	19	19	22
Bahamas	11	26	29	21	23	32	28	26	38
Bahrain	15	18	23	28	25	19	15	26	20

This type of query, measuring multiple time period data going across the columns of a page over other aggregated dimension(s), is a common business query used to view performance over time or for trending analysis. Data in this result form also makes it easy to create various graphs, such as lines or bars that make it easy to see outlying (good or poor) performers. Query response times across the cubes were within 2 seconds for all but the two largest cubes. See the Table of Query times in the appendix for the exact query times.

Cube Query 6 — Monthly Summary By State

This query produces a cross tabular view of order data aggregated across the dimensions state and month:

		2003-06	2003-07	2003-08	2003-09	2003-10	2003-11	2003-12
Alabama	Customer Count	356	630	654	596	642	627	
	# of Orders	370	676	714	658	707	670	
	Order Amount	461,230.58	830,621.81	900,469.16	842,303.84	883,792.62	763,951.58	968,9
Alaska	Customer Count	334	705	669	665	696	649	
	# of Orders	346	774	721	709	761	719	
	Order Amount	441,929.91	970,288.8	903,056.51	878,782.81	864,871.49	925,971.99	986,8
Arizona	Customer Count	386	683	684	640	674	636	
	# of Orders	402	731	739	698	718	697	
	Order Amount	458,769.91	909,194.93	975,707.06	935,749.76	916,371.74	892,082.69	997,3
Arkansas	Customer Count	340	678	661	656	671	646	
	# of Orders	357	746	719	700	735	708	
	Order Amount	479,837.55	960,580.1	834,143.65	843,887.58	883,841.63	872,323.24	873,5
California	Customer Count	381	667	660	639	639	659	
	# of Orders	396	720	721	700	690	710	
	Order Amount	524,569.68	914,924.89	884,768.49	878,808.15	899,470.1	902,871.74	911,6
Colorado	Customer Count	352	673	681	685	677	624	

This type of query, viewing nested groups of key metric aggregates by time period going across the columns of a page, is a common business query used to view performance over time or for trending analysis. Query response times across the cubes were from 1-5 seconds for all but the two largest cubes. See the Table of Query times in the appendix for the exact query times.

Cube Query 7 — Premium Customer Summary

This query provides a cross-tab view of data aggregated across multiple dimensions. It contains order data by year, by security type, and premium customer status:

This type of query, showing many summarized dimensional values with multiple groupings of columns and rows, is a common business query used to analyze performance in a simple but compact view. Query response times across the cubes were within 4 seconds for all but the largest cube built over 100 GB of XML data. See the Table of Query times in the appendix for the exact query times.

		No	No	Yes	Yes
		# of Orders	Order Amount	# of Orders	Order Amount
2003	Bond Fund	35705	45,005,801.61	6314	7,796,424.88
	Mixed Fund	7875	9,783,567.56	1451	1,772,291.56
	Stock	45855	58,343,525.05	8024	9,890,357.57
	Stock Fund	49207	61,383,005.96	8807	10,808,691.02
2004	Bond Fund	29727	36,860,022.45	5375	6,908,953.56
	Mixed Fund	6813	8,322,391.57	1245	1,744,218.16
	Stock	38146	47,285,947.42	6863	8,516,782.52
	Stock Fund	41243	51,444,485.31	7411	9,229,501.46

Tom Tortolani is VP of Product Management at Skytide. For 20 years, he has led Product Management at various successful enterprise software start-ups. As an early employee at Hyperion Solutions/ Arbor Software, Tom was a key contributor to the creation of the BI and OLAP industry, managing over 20 product releases of the Essbase product family. In earlier years, Tom was a Financial Analyst at the Target Corporation with a focus on forecasting, planning, and business analysis. Tom is the primary inventor on US patent 6,317,750 for advanced multi-dimensional navigational techniques. He holds a BA degree in Economics from Bowdoin College.

Vijay Bommireddipalli is an advisory software engineer in the Data Servers Solutions team at IBM's Silicon Valley Lab in San Jose, CA, where he assists customers and partners in developing solutions on DB2. Prior to joining this team, Vijay was a developer on the Warehouse Manager development team. He joined IBM in July 2000, after finishing his Masters degree in Electrical and Computer Engineering at University of Massachusetts - Dartmouth.

Matthias Nicola is a senior software engineer for DB2 pureXML at IBM's Silicon Valley Lab. Matthias works with the DB2 XML development teams as well as with customers and business partners who are using XML, assisting them in the design, implementation, and optimization of XML solutions. Prior to joining IBM, Matthias worked on data warehousing performance for Informix Software, and in research and industry projects on distributed and replicated databases. He received his doctorate in computer science in 1999 from the Technical University of Aachen, Germany.

Skytide, Inc.

1820 Gateway Drive,
Suite 300,
San Mateo, CA 94404

Phone:
1.650.292.1900

Fax:
1.650.312.1400

E-mail:
info@skytide.com

Internet:
www.skytide.com

Conclusion

Skytide and DB2 are designed for efficient management of XML data. DB2 pureXML stores and accesses XML data efficiently. Skytide offers a full set of business analytics functionality directly on XML, eliminating the need to normalize and map XML data to a traditional relational format for the purpose of business analytics. Shredding and mapping is typically a very complex task that is also difficult to maintain over time. The combination of Skytide and DB2 pureXML removes this pain by directly operating over XML data. Moreover, Skytide's capability to utilize DB2's XML querying capabilities to selectively pull only relevant data, leads to significantly lower cube build times and a highly optimized analytics solution.

The measurement results described in this article show clearly that there is a substantial performance benefit for building multi-dimensional data cubes over XML data if DB2 9 pureXML and Skytide are used together.

References

- The TPoX Benchmark, http://tpox.sourceforge.net .
- DB2 9 XML performance characteristics: http://www.ibm.com/developerworks/db2/library/techarticle/dm-0606schiefer/index.html
- A performance comparison of DB2 9 pureXML and CLOB or shredded XML storage: http://www.ibm.com/developerworks/db2/library/techarticle/dm-0612nicola/index.html
- Check the DB2 XML wiki for papers, presentations and demonstrations related to DB2 9 and pureXML. Authors recommend the "DB2 Demo program on developerWorks" as a starting point.
- Check "Using DB2 XML and Java" (developerWorks, Oct 2006) for a tutorial on developing in Java with DB2 XML.
- Learn more about Skytide at: www.Skytide.com

About Skytide

Skytide is a leading provider of next-generation analytical solutions that deliver an unprecedented view into what is driving business performance. Skytide's award-winning technology uses XML as a common layer to dramatically reduce system complexity while offering advanced functionality that cannot be achieved by traditional BI technology. Application areas for Skytide technology include network services, financial services, contact centers, and other areas of business that generate significant volumes of mission-critical unstructured and semi-structured data. Skytide partners include IBM, Sun Microsystems and Inxight. Based in San Mateo, Calif., Skytide is a privately held company funded by Granite Ventures and El Dorado Ventures. For more information about Skytide, please visit www.skytide.com or call 650-292-1900.

SKYTIDE

Appendix

Hardware was supplied by the IBM Innovation Center in San Mateo, CA. Results were recorded using the following machine configurations:

Machine	OS	CPUs	RAM
Intel	SUSE Linux	2 x 2.5 Ghz	8 GB
Power Series	SUSE Linux	4 x 1.9 Ghz	16 GB
Power Series	AIX	16 x 1.9 Ghz	64 GB

Note that the Skytide Server currently takes advantage of 2 CPUs for certain processes. There is no performance benefit to additional CPUs.

Table of Cube Build Times

Cube	Data Source	Cube Build Time	Cube Size	JVM Size	Machine - OS
10GB XML - US Only - 12,885 Orders - 2,577 US Customers - 20,833 Securities	DB2 XML	2 min.	6.8 MB	32 Bit - 2.6 GB RAM	Power Series - SUSE
	DB2 XML	2 min	6.8 MB	64 Bit - 32 GB RAM	Power Series - AIX
	DB2 XML	2.5 min	6.8 MB	32 Bit - 2.6 GB RAM	Intel - SUSE
	DB2 CLOB	39 min.	135 MB	64 Bit - 14 GB RAM	Power Series - SUSE
	File System XML Files	2 hr. 50 min	135 MB	65 Bit - 14 GB RAM	Power Series - SUSE
10GB XML - 3,000,000 Orders - 600,000 Customers - 20,833 Securities	DB2 XML	1 hr. 17 min.	975 MB	32 Bit - 2.6 GB RAM	Intel - SUSE
	DB2 XML	1 hr. 26 min	975 MB	64 Bit - 14 GB RAM	Power Series - SUSE
	DB2 XML	1 hr. 33 min	975 MB	32 Bit - 2.6 GB RAM	Power Series - SUSE
	DB2 CLOB	1 hr. 45 min	975 MB	64 Bit - 14 GB RAM	Power Series - SUSE
	File System XML Files	2 hr. 30 min	975 MB	64 Bit - 14 GB RAM	Power Series - SUSE
100GB XML -US Only - 129,191 Orders - 25,838 US Customers - 20,833 Securities	DB2 XML	15 min.	46 MB	64 Bit - 14 GB RAM	Power Series - SUSE
	DB2 XML	17 min.	46 MB	64 Bit - 32 GB RAM	Power Series - AIX
	5 Country Parallel DB2 XML	18 min.	46 MB	64 Bit - 14 GB RAM	Power Series - SUSE
	5 Country Parallel DB2 XML	20 min.	46 MB	64 Bit - 32 GB RAM	Intel - SUSE
	CLOB S	9 hr. 20 min.,	2.2 GB	64 Bit - 14 GB RAM	Power Series - SUSE
100GB XML - 30,000,000 Orders - 6,000,000 Customers - 20,833 Securities	DB2 XML	17 hr. 1 min	9.6 GB	64 Bit - 32 GB RAM	Power Series - AIX

Table of Query Times

Cube	Cube Size	Query	Query Time
10GB XML - US Only - 12,885 Orders - 2,577 US Customers - 20,833 Securities	6.8 MB	Cube Query 1	1 sec.
		Cube Query 2	1 sec.
		Cube Query 3	1 sec.
		Cube Query 4	1 sec.
		Cube Query 5	1 sec.
		Cube Query 6	1 sec.
		Cube Query 7	1 sec.
10GB XML - 3,000,000 Orders - 600,000 Customers - 20,833 Securities	975 MB	Cube Query 1	1 sec.
		Cube Query 2	1 sec.
		Cube Query 3	2 sec.
		Cube Query 4	2 sec.
		Cube Query 5	34 sec.
		Cube Query 6	83 sec.
		Cube Query 7	4 sec.
100GB XML -US Only - 129,191 Orders - 25,838 US Customers - 20,833 Securities	46 MB	Cube Query 1	1 sec.
		Cube Query 2	1 sec.
		Cube Query 3	1 sec.
		Cube Query 4	1 sec.
		Cube Query 5	1 sec.
		Cube Query 6	5 sec.
		Cube Query 7	1 sec.
100GB XML ** - 30,000,000 Orders - 6,000,000 Customers - 20,833 Securities	9.6 GB	Cube Query 1	1 sec.
		Cube Query 2	1 sec.
		Cube Query 3	25 sec.
		Cube Query 4	55 sec.
		Cube Query 5	704 sec.
		Cube Query 6	1629 sec.
		Cube Query 7	145 sec.

** Note: While the largest cubes in our tests, show the scalability of the Skytide analytics server, a summary data level cube would be more realistic and appropriate to optimize summary level reporting and query performance. For the cube built on top of 100GB raw XML data containing 30 million orders across 6 million customers, each query generates tens of millions of aggregated values. Changing the cube design to use a cross tab summary with multiple levels of aggregate data, could reduce the query response times down to a matter of seconds.