STRATEGIES FOR ALL YOUR DATA

Sandeepan Banerjee, Vishu Krishnamurthy, Oracle

INTRODUCTION

In order to address the latest content management, data interchange and portal initiatives, and also to prepare for the next generation of rich applications mixing the tabular and document metaphors, organizations need to include unstructured data management in their core systems architecture. Key developments in this area are the emergence of XML as an universal data model, the evolution of intelligent search, and the integration of rich media support in Oracle. This paper discusses Oracle XML DB, Oracle Text, Ultra Search, interMedia and Spatial technologies, with additional consideration of XML Query and the integration of structured and unstructured information.

ORACLE XML DB

XML has arrived as a key technology for the next stage of evolution of the Internet. In the beginning, its core characteristics of self-description and ad-hoc extensibility offered the flexibility needed for transport of messages between various applications. Lately, the next generation of XML standards -- such as XML Schema -- have enabled unification of both document modeling and data modeling.

Today, most application data and web content is stored either in a relational database or the file system or a combination of both. XML is used mostly as an artifact for transport, generated from a database or a filesystem. However, as the volume of XML being transported grows, and developers consider the costs of constant regeneration of XML documents there arises the question whether these storage methods effectively accommodate XML content.

The Oracle9*i* R2 release introduced native support for XML in the form of XML DB, which encompassed both a native XMLType storage and a new XML Repository. With 9*i*R2, we fully absorbed the W3C XML data model into the Oracle server, and provide new standard access methods for navigating and querying XML – creating a native integrated XML database within the Oracle RDBMS.

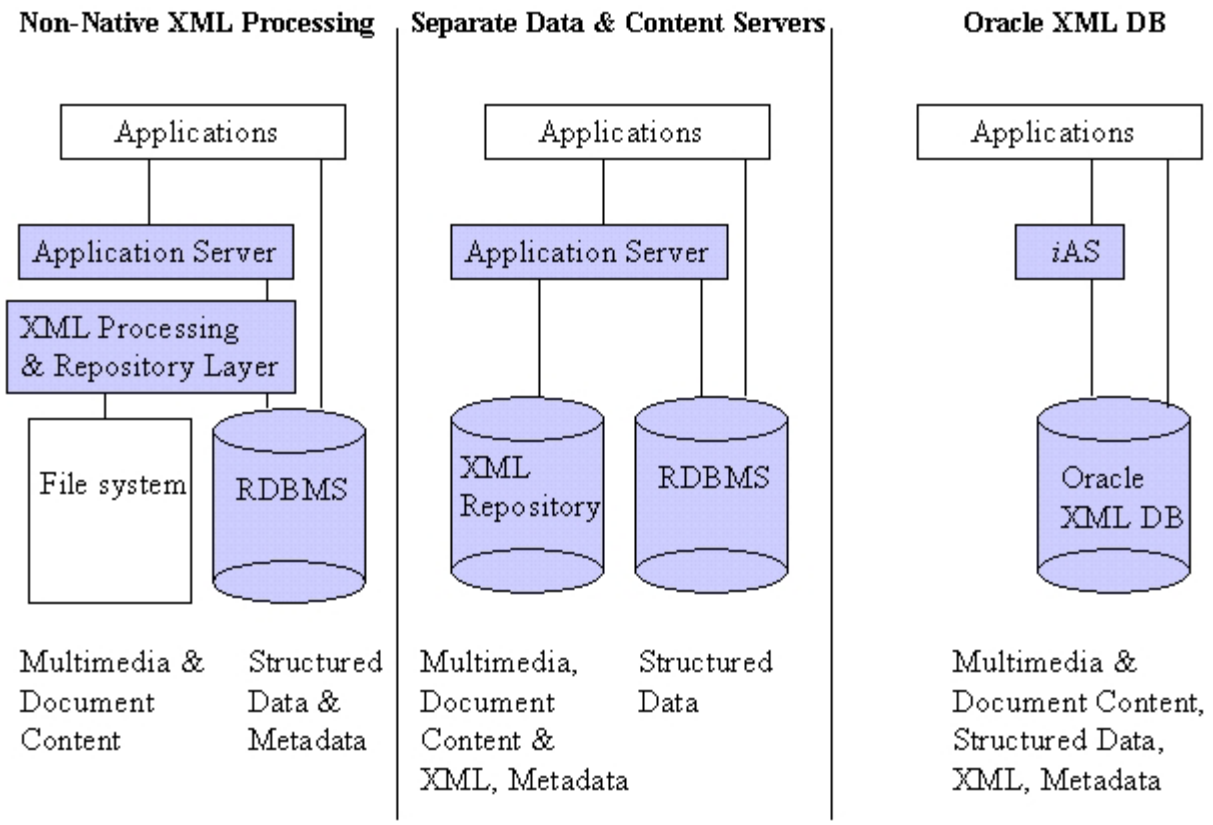

Fig1: Common XML Architectures

With Oracle 10g, we have significantly matured the XML support in the server. The major improvements are in the areas of:

Query performance Improvements, in some cases of 500% or more

XSLT optimizations

Repository Access optimizations

Direct loader support, facilitating loading large XML documents

Storage optimizations, leading to reduced disk-storage requirements for XML

I18N support for differing character sets on client and server

Transparent XML Schema Evolution

Unification of the C XML API between XDK and XML DB

KEY TECHNOLOGIES

The key Oracle XML DB technologies can be grouped into two major classes – those related to XMLType, which provide a W3C XML Schema-compliant native XML storage and retrieval capability strongly integrated with SQL, and those related to the XML Repository that provides WebDAV-oriented foldering, access control, versioning etc. for XML resources. Let us look at each of these classes of functionality in detail.

XMLTYPE

The XMLType datatype stores XML content, and can be used as the datatype of a column. XMLType includes a number of useful methods to operate on XML content. XMLTypes can be stored with 2 storage options – LOB and Object-Relational storage. The former storage model maintains accuracy to the original XML (whitespaces and all), while the latter maintains DOM (Document Object Model) fidelity. XMLType achieves DOM fidelity by maintaining information that SQL or Java objects normally don't provide for, such as:

- Ordering of child elements and attributes
- Distinguishing between elements and attributes
- Unstructured content declared in the schema (e.g. content="mixed" or <any> declarations)
- Undeclared data in instance documents, such as processing instructions, comments, and namespace declarations
- Support for basic XML datatypes not available in SQL (Boolean, QName, etc.)
- Support for XML constraints (facets) not supported directly by SQL, such as enumerated lists

Native XMLTYPE instances contain hidden columns that store this extra information that doesn't quite "fit" within the SQL object model. This information can be accessed via APIs in SQL or Java, such as ExtractNode.

Changing XMLTYPE storage from object relational to LOB (or vice versa) is also possible (via database import & export), and your application code will not have to change in response. This allows you to change XML storage when tuning your application, since each storage option has its own benefits.

PROS AND CONS OF XML STORAGE OPTIONS

LOB (WITH TEXT INDEX)	*OBJECT RELATIONAL (WITH BTREE INDEX)*
Very flexible when schemas change	Limited flexibility for schema changes (similar to the ALTER TABLE restrictions)
Maintains the original XML byte for byte, which may be important to some applications	Trailing newlines, whitespace within tags and data format for non-string datatypes is lost
Mediocre performance for DML	Excellent DML performance
	Accesibility of existing SQL features (constraints, indexes, etc.)

Some of the other key features of XMLType are:

XML Schema support: Create tables and types automatically given a W3C standard XML Schema extending the normal SQL DDL. This means you have a standard data model for all your data (structured and unstructured), and can use the database to enforce this data model.

XML Piecewise Update: Use XPath to specify individual element(s) and attributes of your document to update, without rewriting the entire document. This is more efficient, especially for large XML

documents.

XPath Search: Specify elements to query against via XPath, and then use SQL operators (conformant to the emerging ANSI SQLX) on these elements. This helps you combine the best of SQL and XML.

XML Indexes: Use XPath to specify parts of your document to create indexes for XPath searches.

XML Operators: New operators like XMLTABLE (to cast a list of nodes returned by XPath into a table), XMLELEMENT (to create XML elements on the fly), etc. to make XML queries and on-the-fly XML generation easy. ORACL EXML DB makes the SQL and XML metaphors interoperable.

XSL Transformations for XMLType: Use an XSLT to transform XML documents via a SQL operator. You can get fast response-time, database-resident XSL transformations.

Lazy XML Load: XMLType provides a virtual DOM; it only loads rows of data as they are requested, throwing away previously referenced sections of the document if memory usage grows too large. This helps you get high scalability when many concurrent users are dealing with large XML documents.

XML Views: Create XML views to create permanent aggregations of various XML document fragments or relational tables. You can create your own efficient representations of XML.

Schema Caching: ORACL EXML DB keeps structural information (like element tags, datatypes, and storage location) in a schema cache, to minimize access time and storage costs. This helps you get high performance and scalability with large documents, as well as a large number of documents.

Schema Evolution: With XML DB in Oracle 10*g*, it is possible to transparently evolve XMLSchemas, with the Oracle server silently performing the data unload/reload required.

XML REPOSITORY

The second key aspect about Oracle XML DB is that it provides an Internet repository for managing XML data and documents. Important items of Repository functionality include:

Access Control Lists (ACLs): Create high-performance access control lists for any XMLType object, and define your own privileges in addition to the system-defined ones.

Foldering: Enable folders to map resources (XML files) into database structures and enable directory traversal; also, use XMLTypes or views to map rows into URLs (via ALTER TABLE ENABLE FOLDERING), providing access control, modification date tracking, and other metadata management for those rows.

WebDAV and FTP Access: Access any foldered XMLType row via WebDAV and FTP (Note that XMLType can manage arbitrary binary data as well, including any file format).

SQL Repository Search: Operators like UNDER_PATH and DEPTH, allow applications to search folders, file metadata like owner and creation date, as well as file contents via SQL, and enable the SQL optimizer to choose the best execution plan.

Hierarchical Index: Oracle XML DB provides a special hierarchical index designed to speed pathname resolution and folder search. Additionally, you can automatically map hierarchical data in relational tables into folders, where the hierarchy is defined by existing relational information.

Servlet Access: Users manipulating XML data in the Oracle server can use the servlet API to process XML via Java.

KEY BENEFITS

The integration of a native XML capability within the database brings a number of benefits.

- Users today manage structured data as tables and unstructured data as files or BLOBs, and have to subject their applications to different paradigms for managing different kinds of data. Systems channel application development down either the unstructured path (making document access transparent but table access complex) or the structured one (vice versa). Oracle XML DB provides a unique ability to store and manage both structured and unstructured data, under a standard W3C XML data model. Oracle XML DB provides complete transparency and interchangeability between the XML and SQL metaphors. You can perform XML operations over table data and SQL operations over XML documents. This opens up the database for a new class of 'XML-shaped' content.
- Oracle XML DB provides valuable Repository functionality – foldering, access control, FTP and WebDAV protocol support with versioning – enabling applications to retain the file abstraction when manipulating XML data brought into Oracle.
- Users today face a performance barrier in storing and retrieving complex XML. Oracle XML DB provides superior performance and scalability for XML operations
- Oracle XML DB provides better management of unstructured XML data through piecewise updates, indexing, search, multiple views on the data, managing intra-document and inter-document relationships and so on.
- Oracle XML DB enables data and documents from disparate systems to be accessed (e.g. through gateways, external tables) and combined into a standard data model. This integrative aspect reduces the complexity of developing applications that must deal with data from different stores.

These unique features will be attractive to B2B applications, developers of Web Services, Internet applications, content-management applications, as well as data- and application integrators. In the absence of strong database support for XML, many of these developers have leaned towards file-storage or unstructured storage of XML. If you store XML data in files or CLOBs, you are not exploiting several

key capabilities of databases.

- Indexing and Search: Real applications need to do queries like "find me all of the product definitions created between March & April 2003", a query that is typically supported by a BTREE index on a date column. This type of query is why most content management vendors need to use an RDBMS, since even document metadata requires BTREE indexes. Content management vendors have had to build proprietary query APIs to handle this problem. Oracle XML DB enables efficient structured search over XML data.

- Updates & Transaction Processing: Today's commercial relational databases enable fast updates of subparts of a record, with minimal contention between users trying to update. As traditionally document-centric data becomes more structured (via XML), this requirement gains in importance. File- or CLOB- storage cannot provide the granular concurrency control that Oracle XML DB does.

- Managing Relationships: Data with any structure will typically have some type of foreign key constraint. Currently, XML data stores lack this feature, so you must implement these in application code. Oracle XML DB enables you to constrain XML data to XML schemas, as well as relational constraints, and thus achieve the control over relationships that structured data has always enjoyed.

- Multiple Views of Data: Most enterprise applications need to group data together in different ways for different modules. This is why relational views are necessary—to allow for these multiple ways to combine data. By allowing views on XML, Oracle XML DB allows you to create different logical abstractions on XML.

- Performance and Scalability: People expect data storage, retrieval, and query to be fast. Loading a file or CLOB and parsing is much slower than relational data access. Oracle XML DB can speed up XML storage and retrieval.

- Ease of Development: Databases are foremost an application-development platform, that provide standard, easy ways to manipulate, transform and modify individual data elements. While XML parsers give read access to XML data in a standard way, they don't provide an easy way to modify individual XML elements and store them. Oracle XML DB supports a number of standard ways to store and retrieve data – using XML Schema, XPath, DOM etc.

On the other hand, if the drawbacks of XML file storage are forcing you to break down XML into database tables and columns, there are several advantages of XML you are leaving on the table.

- Structure Independence: The open content model of XML cannot be captured easily in the pure tables-and-columns world. XML Schema allows global element declarations (not just scoped to a container), so that you can find a particular data item regardless of where in the XML document it moves to as your application evolves.

- Storage Independence: When you use relational design, you client programs need to know

where your data is stored, and in what format, what table, and what the relationships are between those tables. XML Schema allows you to write applications without that knowledge, and allow the DBA to map structured data to physical table and column storage.

- Ease of Presentation: XML is understood as a native format by browsers, desktop applications like Microsoft Office XP, as well as various Internet applications. Relational data isn't generally accessible directly from applications, but requires programming. Oracle XML DB allows you to store data as XML and pump it out as XML, requiring zero programming for the contents of your database to be displayed.

- Ease of Interchange: XML is the language business is using to talk to business. If you are forced to store XML into an arbitrary table structure, you are living with some sort of proprietary translation. Whenever you translate a language, information is lost, so interchange suffers. By natively understanding XML and providing DOM fidelity in the storage/retrieval process, Oracle XML DB enables clean interchange.

ORACLE TEXT & ULTRA SEARCH

Oracle offers a complete technology stack for content search, organization, and presentation. There are two aspects of this solution. One is a comprehensive information retrieval API called Oracle Text that allows developers to build any kind of search application. The second aspect is an out-of-the-box solution application for enterprise intranet search.

ORACLE TEXT

Oracle Text uses standard SQL to index, search, and analyze text and documents stored in the Oracle database, in files, and on the web. Oracle Text can perform linguistic analysis on documents; search text using a variety of strategies including keyword searching, context queries, Boolean operations, pattern matching, mixed thematic queries, HTML/XML section searching, etc. Oracle Text can render search results in various formats including unformatted text, HTML with term highlighting, and original document format. Oracle Text supports multiple languages and uses advanced relevance-ranking technology to improve search quality. Oracle Text also offers advanced features like classification, clustering, and support for information visualization metaphors.

ORACLE ULTRA SEARCH

Ultra Search can be used to search across Collaboration Suite Components, corporate web servers, databases, email servers, fileservers and Oracle10iAS Portal instances. Ultra Search is based on Oracle10g Text technology and is an out-of-the box solution that requires no SQL coding. It uses a crawler to index documents; the documents stay in their own repositories, and the crawled information is used to build an index that stays within your firewall in an Oracle database.

Let's look at the major search aspects these products can provide for All Your Data.

QUALITY

In the area of quality there are four techniques that we can use to improve search quality results.

- Spelling correction
- Link awareness
- Duplicate elimination
- KWIC

The spell-checker component is pretty straightforward. The user types a query and before issuing a search the system spell checks the entire phrase. The component has large dictionary that is also extensible.

The link awareness technique is very useful for reflecting specific characteristics of documents such as links, anchor text, and title information. There is traditional static link based analysis and query hitlist link analysis. Oracle uses a combination of both strategies that suits the intranet search topology.

Duplicate elimination is a very common problem in intranets. There are several copies of the same document or web page in different places. The idea is to remove URLs with duplicate and near-duplicate content.

The keyword in context (KWIC) feature has become an easy way to have an idea of what the document or web page is all about without clicking on the link. Before that it was common to the see the first eight characters of the page but sometimes the information was misleading. Figure 2 shows a screenshot of the KWIC component.

Figure 2. KWIC in action

PERFORMANCE

As part of the Oracle10*g* database, Oracle Text transparently integrates with and benefits from a number of key enterprise features such as

- Data partitioning (for higher throughput and availability)
- Real application clustering (for the highest server scalability)
- Query optimization (to ensure the best response time, not only for pure text queries, but also 'mixed' queries that combine text search with structure database search)

These aspects of integration are also greatly beneficial to system and database administrators, who do not have to undergo a paradigm shift to learn to manage and organization's text assets.

Common and Rare Queries

Typically we can say that 80% of the queries are common queries and 20% are less frequently or rare queries. Make sense then to have two separate indexes. In the first one we will index only the URL and the title of the web page. The idea here is to exploit the structure of the web page. For the second index we use a more traditional approach that is indexing the full content of the web page.

There are number of advantages with this approach. The overall system performance is improved. The first index is smaller and it will return most of the queries. The second index is bigger and it will return rare queries.

Query Relaxation

Query relaxation enables your application to execute the most restrictive version of a query first, progressively relaxing the query until the required number of hits is obtained. For example we search first "JDeveloper download" and then the query is relaxed to JDeveloper NEAR download to obtain more hits.

Query relaxation is most effective when the application needs the top N hits to a query. Using this technique is more efficient than re-executing a query.

Ease of use

Users want to have a simple and easy to user search interface very similar to existing Internet search engines. Our approach is to hide all the complexity of the search engine under the covers and expose a typical web search interface and let the user discover the power of the engine.

Ultra Search is an out-of-the-box search solution that provides search across multiple repositories – Oracle databases, IMPA email servers, websites, files on disks and many more. It uses a crawler to index documents; the documents stay in their own repositories, and the crawled information is used to build an index that stays within your firewall in an Oracle10*g* database.

The interface has to modes: basic search and advanced search. In the basic search a simple input box is presented. The search results are presented sorted by relevance. The advanced search mode offers more control over the collection. Figure 3 shows Ultra Search in the context of the Oracle Collaboration Suite.

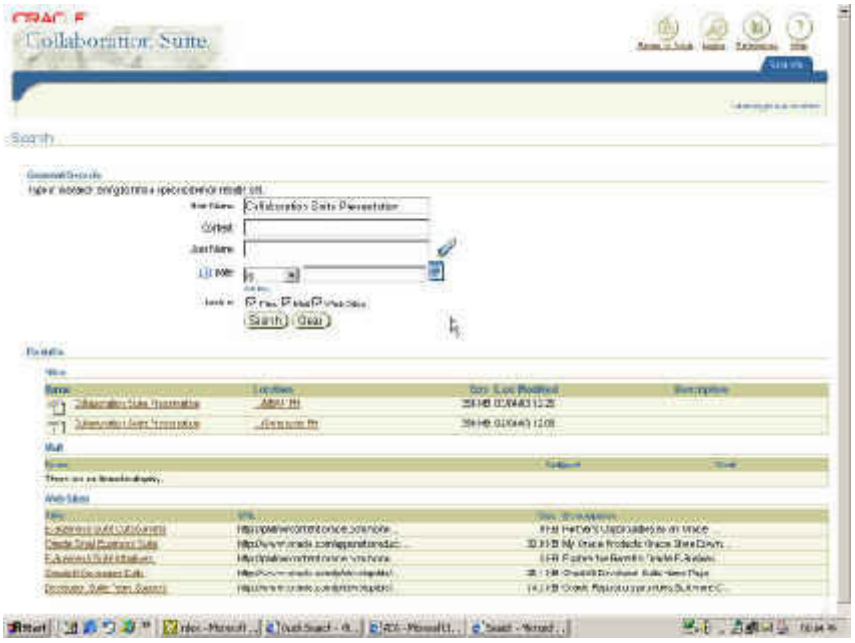

Figure 3. Ultra Search results from different data sources.

*inter*MEDIA & SPATIAL TECHNOLOGIES

Oracle *inter*Media provides an array of services to develop and deploy traditional, Web, and wireless applications that include rich media.

Multimedia content can be managed directly in Oracle 10*g* under complete database control. Alternatively, Oracle 10g can store and index meta-information together with external references that enable efficient access to media content stored outside the database.

Oracle *inter*Media is a standard feature of the database. Its media services support JDeveloper and Oracle10iAS Portal.

The multimedia services of *inter*Media allow you to:

- Parse, index, and store rich content using new or existing database schema's
- Develop content rich Web applications
- Deploy rich content on the Web
- Use standard database features to create scalable, manageable, media content repositories

Oracle 10*g* provides location-based services that support a wide range of applications -- from automated mapping/facilities management and geographic information systems (GIS), to wireless location services and location-enabled e-business. The Oracle location platform includes the Oracle 10*g* database,

Application Server, and E -Business Suite.

Oracle Spatial and Oracle Locator make location a native type within the Oracle 10*g* database. Oracle Locator provides spatial object type storage, indexing, and operations, to support a variety of location-based services (LBS) and 3rd party GIS solutions. Oracle Spatial provides advanced spatial features to support high-end GIS and LBS solutions. MapViewer is an Oracle 10*i*AS Java component used for map rendering and viewing geospatial data managed by Oracle Spatial or Locator.

Oracle Spatial and Oracle Locator have been adopted as the preferred location platform by leading GIS and LBS vendors. Oracle Spatial and Oracle Locator have also been deployed by telecommunications, utilities, and e-government organizations worldwide.

CONCLUSION

Oracle 10*g* provides a complete solution for managing All Your Data. The complete platform helps application developers create the next generation of rich applications mixing the tabular and document metaphors, organizations need to include unstructured data management in their core systems architecture. Key developments in this area are the emergence of XML as an universal data model, the evolution of intelligent search, and the integration of rich media support in Oracle, as well as the ability to find documents based on their textual, content metadata, or attributes. Taken together, these features make the Oracle database the single point for all data management.