

ORACLE[®]



ORACLE®



High Performance and Flexible Management of All Your XML Data with Oracle XML DB

Geoff Lee, Principal Product Manager
Oracle Corporation



Trends

- Traditional business applications mixing content with data
- Documents becoming semi-structured, driving value of database
- XML becoming the standard for representing documents
- Microsoft Office and Open Office use XML based storage formats
- Growing demand for XML-based content management (CM) systems



Document Centric Development

- Content created, edited and managed using desktop productivity software and XML editors
- Either no XML Schema or very flexible
- Database is transparent
- File / Folder metaphor for organizing content
 - Content identified and accessed using URLs
 - Support emerging XML standards XLink and XInclude



Emerging XML Vertical Standards

- Emerging vertical standards in a number of markets embracing XML
 - XBRL in Financial and Regulatory Markets
 - FpML in Financial Services
 - HL7 in Healthcare
 - GJXML in Law Enforcement / Public Safety
 - DICOM and EXIF for Digital Imaging
 - RSS for publishing / syndication of content
 - OpenGIS for spatial applications

Evolution of Oracle's XML

Support

Performance



Binary XML
Storage
& Indexing



XQuery



XML
Storage &
Repository



XML
API's

1998

2001

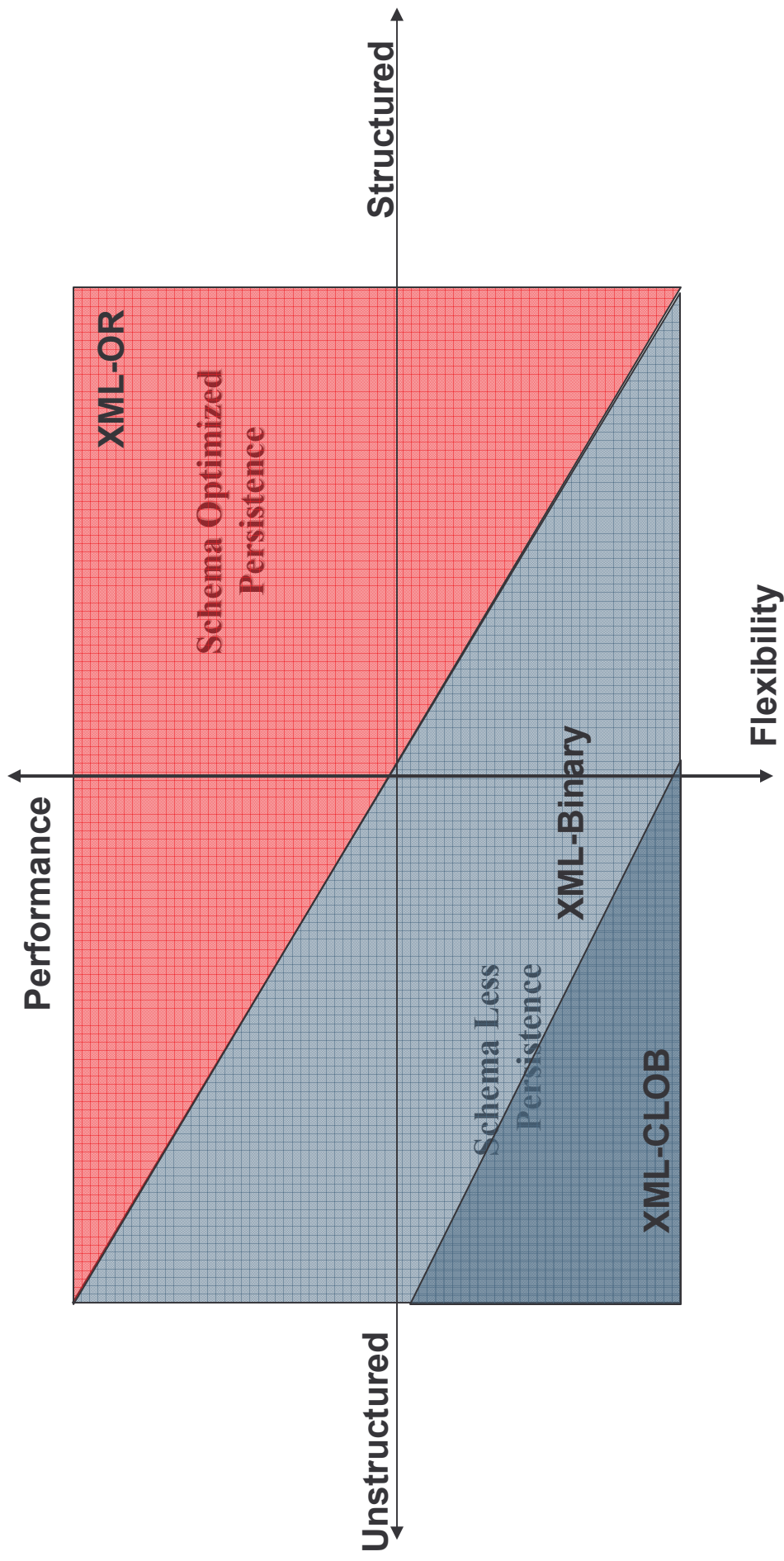
2004

2007

ORACLE



XML Storage Models





Oracle XML Support 11g

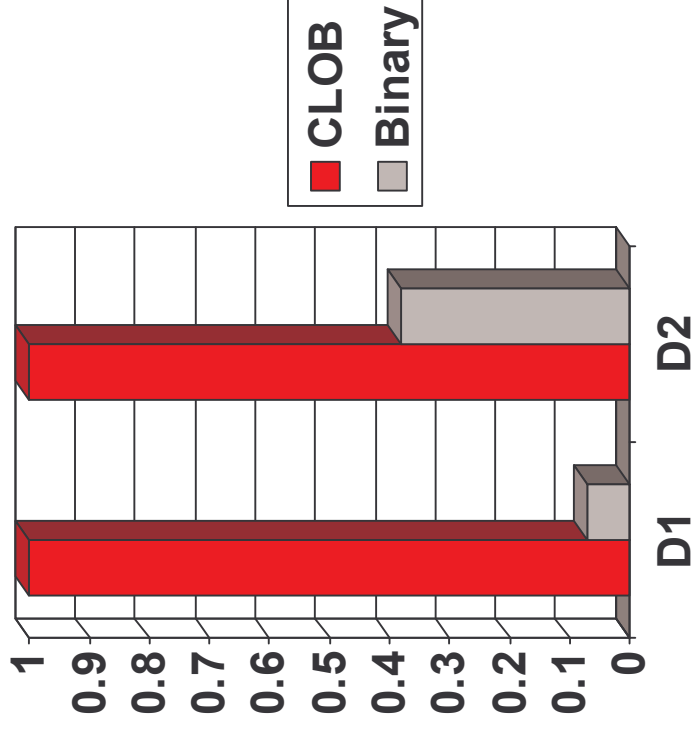
- Binary XML Storage model
 - Compact binary representation of an XML document
 - Streaming operations (rather than DOM)
 - On-disk == In-memory == On-the-wire == Client
- XML Indexing techniques
 - Universal index for XML data
 - Speeds up path and value based access
 - Usable for fragment identification and extraction

Binary XML with XMLIndex offers scalable, performant processing for Schemaless XML



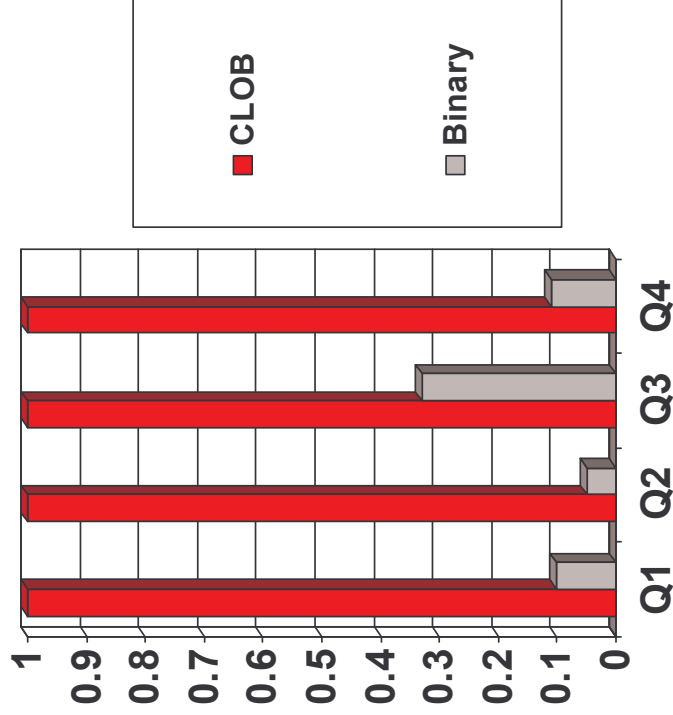
Sample compression

- D1 –mostly structured
- D2 – semi-structured
- These are based on actual customer datasets and comprise a mix of XML documents ranging from a few KB to many MB.

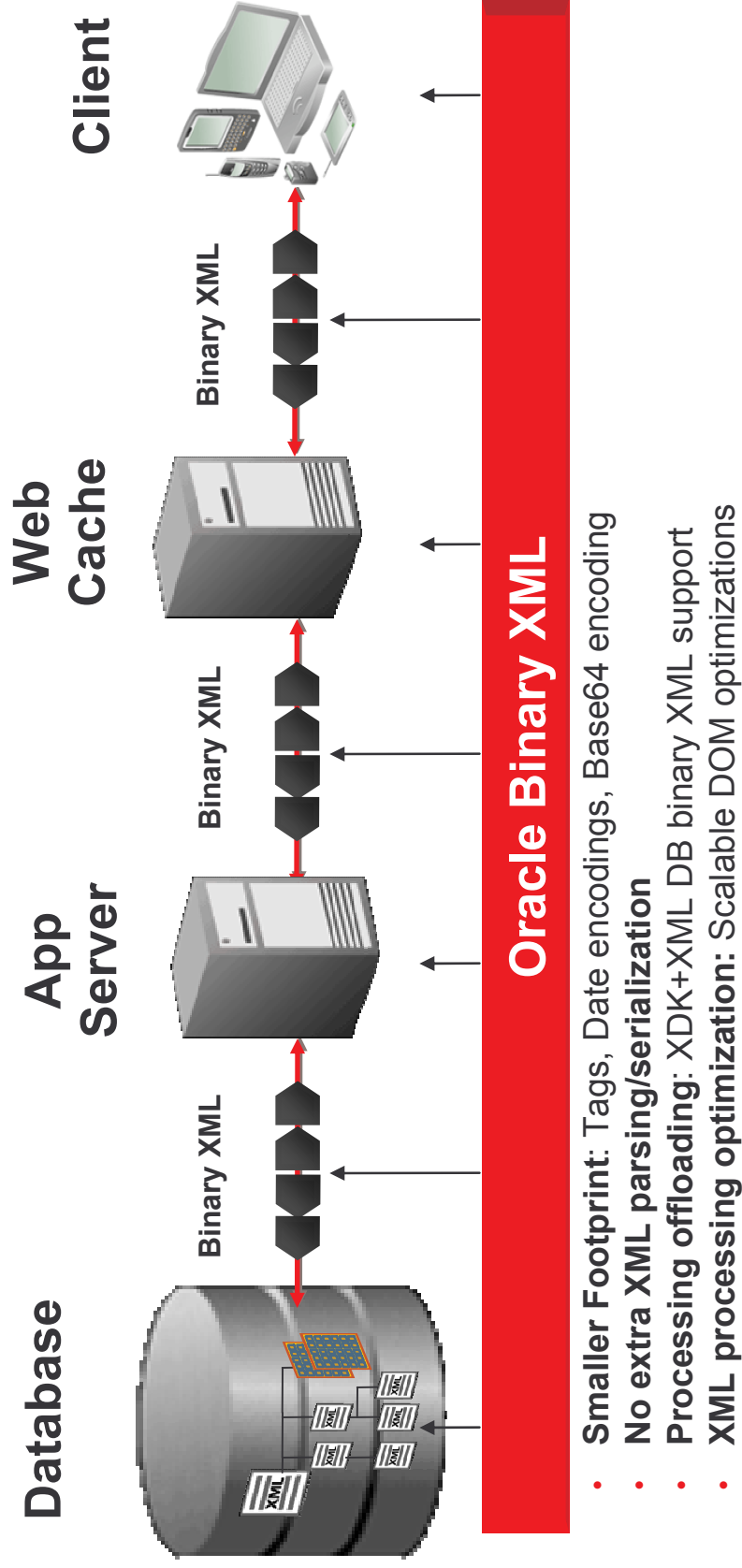


Query Performance

- Q1 – Q4 are representative queries for a semi-structured application.
- No Indexes present.
- Numbers are normalized based on CLOB elapsed time.
- Only for illustrating performance gains obtained by eliminating parsing and conversion overheads.



Binary XML





Oracle Binary XML Format

- Compact Schema-aware XML Format
 - Addresses most of the issues with XML 1.x serialization
 - Oracle proprietary format (at least for now)
- Intended for use in all tiers of Oracle stack
 - Oracle XML DB
 - Oracle iAS / XDK Java
- Exploits XML Schema information if available
 - Also supports non-schema-based encoding
- Preserves Infoset or Data Model fidelity – Not bytes



Format Basics

- Sequence of opcodes (serialized version of SAX events)
- Opcode has fixed number of operands
- Tag names are tokenized
 - Central repository (or)
 - Inlined definitions
- Uses native data-types in the presence of XML schema



Streaming Capabilities

- Format designed for streaming processing (rather than DOM based)
- Streaming XPath evaluation
 - Multiple XPathS evaluated in a single pass
- Streaming Validation
 - XML Schema Validation does not require DOM
- Streaming Piece-wise updates
 - Most common update scenarios handled in streaming manner

Streaming instead of DOM = Scalable Processing



When should you use Binary?

- If the XML Schema is not known a-priori
- If a schema is available but results in too many tables and joins in an Object relational mapping
- Document centric use-cases with flexible structures

Schema Flexibility required → Binary Storage



When should you use Binary ?

- Unstructured (schema-less), semi-structured use-cases originally satisfied by CLOB storage
- Advantages of Binary XML:
 - Avoids bottleneck of expensive XML parsing
 - Good performance even without indices
 - Data-type aware storage removes expensive conversions.
 - Compression of XML content to avoid text bloat



Binary XML Storage

- New storage option for XMLType tables, columns
- Binary data stored in underlying BLOB column

**CREATE TABLE purchaseOrder OF XMLTYPE
STORE AS BINARY XML;**

- Can store schema-based and schemaless XML documents in the same table
- Existing applications built against XMLType work transparently



XML Schema Support

- XML Schemas can be registered for use in binary XML encoding
- No SQLType mappings to manage
- Native data-type encoding used for numbers, dates and other XML schema types
- Enhanced compression by leveraging schema
- Backwards compatible schema evolution



Efficient DML

- Optimized support for piece-wise changes to XML documents

```
updateXML( '/purchaseOrder/Reference/text()', 'XXXX' )
```

- Can be directly applied on disk avoiding expensive DOM construction
- Can take advantage of the new SecureFile storage to modify only the changed portion



Mid-Tier Support

- XDK Java components for Binary XML
 - Also C components available in OCI client library
- Text <-> Binary XML <-> DOM
 - Support for SAX and Pull models on binary XML
 - Scalable DOM based on binary XML
- Optimized for accessing XML DB
 - Tokens synchronized with XML DB
 - Generate binary XML to offload DB processing during insert



Schema Evolution

- Allow any backwards-compatible schema evolution
 - Old instances can always be decoded using new schemas
- Examples of allowed schema evolution
 - Adding optional element or attribute
 - Adding new values to enumeration
 - Increasing maxOccurs, maxLength facets
 - Changing type to “larger” type: short->int, float->double, any simple type to string



Indexing Binary XML

- Universal index for XML data
- Efficient access for identification and extraction of XPath based fragments.
 - SQLX and XQuery Data models
- Provides good query and DML performance.
 - Schema-less and Schema-based data
 - Ad-hoc query support
- High scalability in terms of size and number of XML documents.
 - Index creation and maintenance in parallel



XMLIndex Structure

- Logically organized into a *path index*, *order index* and one or more *value indexes*
 - *Path index* - Useful for identifying fragments based on simple (navigational) path expressions
 - *Order index* - Useful for determining parent-child, ancestor-descendant, and sibling relationships
 - *Value index* - Allows type aware lookups based on value equality or range
- Ad-hoc XML queries are decomposed into a composite operation that performs path, order and value constrained lookups



Oracle XMLIndex Features

- Asynchronous index maintenance
- Path subsetting
- XQuery support
- Support for indexing CLOBs embedded within O-R storage
- XML-DB repository
- Partitioned index
- Parallel index creation, maintenance and query



Oracle XMLIndex Path Subsetting

- Specify
 - nodes that will be used in common queries or
 - nodes that will rarely be used
- Can change the specified paths later
- Better DDL, DML performance
- Reduces size of primary and secondary indexes. Less storage overhead
- Transparent to queries



Binary XML Query Support

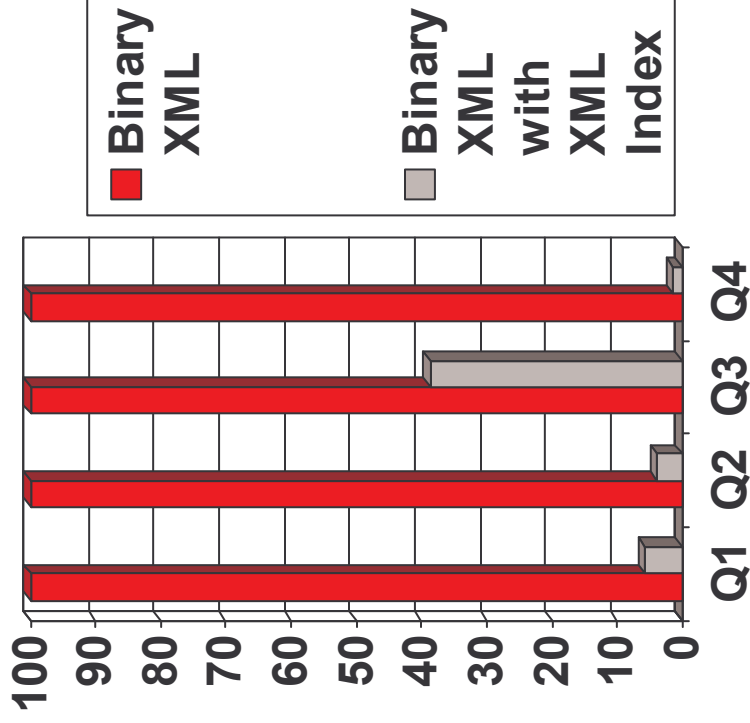
- SQL/XML and XQuery operators
- XMLIndex
 - Indexes paths, hierarchy and values
 - Speeds up fragment identification across multiple XML documents
- Streaming XPath Evaluation
 - Streaming evaluation of multiple XPath's over an XML document

Overall Query Strategy: Combination of XMLIndex and Streaming XPath evaluation



Query Performance with XML Index

- Q1 – Q4 are representative queries for a semi-structured application
- Numbers are normalized based on CLOB elapsed time





Advantages of Binary XML

- Improved storage/retrieval
 - Efficient conversion to and from XML text
 - Smaller footprint results in less disk IO
 - Reduced CPU cost for loading XML info-set into memory
- Leveraging XML Schema
 - With or without a schema (even in the same column)
 - Avoids data conversions to/from string
 - Backward compatible schema evolutions



Advantages of Binary XML

- Improved Query Performance
 - In conjunction with XMLIndex
 - Streaming single-pass evaluation of many XPath's
- Enhanced Mid-Tier Support
 - Oracle Mid-Tier understands the same binary format.
 - Better performance by offloading XML processing.
 - Thin JDBC support




Summary

- New storage model for XMLType – Binary XML
 - Integrated with Oracle Mid-Tier XDK components
 - Improved query processing and compressed disk footprint.
- New indexing for XMLType – XMLIndex
 - Complete indexing solution for schemaless XML
 - Scalable handling of large datasets
- Use cases
 - Leverage mid-tier support
 - Requiring schema flexibility
 - XML based content management systems



Q&A



The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE IS THE **INFORMATION** COMPANY