S317428: Building Really Scalable XML Applications with Oracle XML DB and Oracle Text

Michele Pompilius
Data Technology Manager
Proquest

Nipun Agarwal
Director, XML Development
Oracle
Background Information

• ProQuest Company is a privately-held global information services company
• 1500 employees
• $500 million revenue
• ProQuest partners with leading newspaper and academic journal providers in disciplines such as medicine, technology, social sciences, and humanities
• ProQuest aggregates materials and distributes digitized content to academic institutions, public libraries and schools
• ProQuest has a portfolio of 1,500 products and relationships with over 9,000 content providers
• 10 major product lines
Background Information

• **Who am I?**
  • Data Technology Manager
    • Have Worked with Oracle Products Since 1987
    • Rejoined ProQuest in June, 2007
    • Manage the Database Team
      • 3 DBAs; 3 Architects; 4 Developers
      • Part of the Global Product Development Organization
      • Support Other Areas of the Business
      • JDeveloper for Custom Internal Applications
  • New to Oracle XML DB
    • Initiated Proof of Concept in August, 2009
    • Started with 11gR2 Beta
Project Morningstar

- Enterprise-wide effort to consolidate technology across multiple business units, each with its own “silos” of content
- Two-year plan to establish new platform, integrating business units in phases
- Approximately 100 staff members involved
- Ultimate goal is a single, integrated vault comprising all ProQuest content, which can be searched from a single entry point

Technical Strategies/Challenges

- Huge volume of documents
- Very complex, internally developed XML Schema
Oracle XML DB Product/Project Specifics

• Application Architecture
  • Front-end (Customer Facing Application)
    • Web-based interface for user login
    • Never interacts directly with the content store
    • Documents are searched and served to users by FAST search engine
  • Content Store (Internal Editorial Application)
    • Complete store of documents in Oracle XML DB
    • Content Store User Interface directly interacts with content store
      • XML Search being investigated and prototyped now
• Document Manufacturing
  • Ingest rate of 10 million documents/day
Oracle XML DB Product/Project Specifics

• Data Characteristics
  • Typical document size is 10-12k
  • XML Schema has on the order of 700 nodes
    • Flexible model
    • Supports many content types: newspaper, journal, dissertations, etc

• Data Volume
  • Proof of concept: scaled to 82 million documents
  • Production: Now just shy of 800 million documents
    • Next phase (2011) will ramp document count up over 2.5 billion
  • Database is currently 7TB in size
    • XML Table and LOB segment is 5 TB
Oracle XML DB Product/Project Specifics

• Environment
  • 4 node cluster, HP DL360 G6, 2 quad core CPUs 144GB RAM
  • Running 11.2.0.1 (11gR2) on RHEL 5.3
  • Supports all online users, internal/editorial operations, and manufacturing activity
  • XML Table is Range Partitioned

• Launch Schedule
  • August 2010 – Customer Preview Successfully Launched!
  • December 2010 – General Release

• Next Steps
  • Continue to increase document manufacturing ingest rates
  • XML Index and Text Index Prototyping
Two Phases

- Phase 1
  - Live in 2010
  - Focus on
    - Ingestion speed
    - Scalability
    - Disk Storage
- Phase 2
  - Work in Progress – plan to go live in 2011
  - Focus on Query performance
  - Build XML and Text Indexes
Data Model

- Binary XMLType column in a relational table
- Partitioned by range on primary key
- Non-schema based to avoid schema evolution later
- Locally partitioned XMLIndex and Text Index
- Running on a RAC system
Ingestion Performance

- Range Partitioned Binary XML Table
- Asynchronous index for both XMLIndex and Text Index
- POC numbers
  - Target: 300 docs/sec
  - Achieved: 475 docs/sec (SQL Loader)
  - CPU utilization < 60%
Scalability

- 800 million rows
- About 50 partitions
- Concurrent load
- Parallel Query
- Ingestion rate constant
- >5 TB of XML data
Storage

- 25% compression for Binary XML
- Disk storage less than competitors
- Less I/O
- More rows in memory
- Indexes use around 3x of raw xml data
TPoX Benchmark – XML DB
Comparison of Storage Space with another DB

Oracle uses 2.4x less storage

XML Storage and Indexing

- Oracle 11gR2 Binary Storage with XTIDX
- Another DB with SB with XIDX
Phase 2 – Query Performance
Query Performance

- Mixed queries containing `XML EXISTS` and `CONTAINS`.
- `CONTAINS` may use INPATH, HASPATH.
- One predicate uses index, the other evaluated as a post filter.
- Cost of predicates determines index usage.
- Queries use parallel processing to utilize available CPU.
- Contains clause optimized to push down most processing, including `count`, to text index.
- Result Set Interface with parallel table function.
Proquest Sample Queries

```sql
select p.doc
from PROQUEST_DATA p
where
  xmlexists('/RECORD/ObjectInfo/Copyright/CopyrightData' passing p.doc) and
  xmlexists('/RECORD/ObjectInfo/RecInfo/ObjectRevisions/ObjectRevision[UpdatedDate="20090614150554"]' passing p.doc)
order by goid
/

select /*+ FIRST_ROWS(50) no_index(p pd_text_index) rparse*/ p.doc
from PROQUEST_DATA p
where
  xmlexists('/RECORD/ParentInfo/Parent[GroupingID="23468"]' passing p.doc) and
  contains(p.doc, 'new') > 0
order by goid
/```
XML Index

- Primary use case in conjunction with Binary XML
- Accelerates path, predicate and structural attribute searches
- Path based index: 11gR1
- Structured index: 11gR2
XMLIndex (Path Based)

- Accelerates path and predicate searches
- Organizes paths and values in single path table
- Supports searching and fragment extraction
- Path sub-setting for indexing specific paths
- Asynchronous mode for deferred maintenance
- Ideal when XPath to be queried not known in advance
- Also called Unstructured XMLIndex
## XMLIndex (Path Based) Layout

<table>
<thead>
<tr>
<th>RID</th>
<th>PATHID</th>
<th>ORDER KEY</th>
<th>LOCATOR</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>/Document</td>
<td>1</td>
<td>Locator to get binary content</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>/Document/Title</td>
<td>1.1</td>
<td>Locator to get binary content</td>
<td>Indexing XML Techniques</td>
</tr>
<tr>
<td>10</td>
<td>/Document/Affiliation</td>
<td>1.2</td>
<td>Locator to get binary content</td>
<td>Oracle</td>
</tr>
<tr>
<td>10</td>
<td>/Document/pubDate</td>
<td>1.3</td>
<td>Locator to get binary content</td>
<td>2007-04-10</td>
</tr>
<tr>
<td>20</td>
<td>/Document</td>
<td>1</td>
<td>Locator to get binary content</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>/Document/Title</td>
<td>1.1</td>
<td>Locator to get binary content</td>
<td>Object relational storage</td>
</tr>
</tbody>
</table>
XMLIndex (Structured)

- Project out commonly searched structured attributes

- Pivot each item as a column in the table
  - All xpath matching is avoided at run time

- Secondary Indexes can be created on Structured Index
  - Relational indexes on projected scalar attributes
  - Text Index on projected text attributes
  - Domain specific Index on domain attributes, e.g. image

- Physical rewrite using XQuery/XPath expression matching
XML data

<Document>
<title>Indexing XML Techniques</title>
<affiliation>Oracle</affiliation>
<pubdate>2007-04-10</pubdate>
....
</Document>

<Document>
<title>Object relational storage</title>
<affiliation>Oracle</affiliation>
<pubdate>2003-03-15</pubdate>
...
</Document>

<table>
<thead>
<tr>
<th>Row ID</th>
<th>Title</th>
<th>Affil</th>
<th>Pubdate</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Indexing XML Techniques</td>
<td>Oracle</td>
<td>2007-04-10</td>
</tr>
<tr>
<td>20</td>
<td>Object relational storage</td>
<td>Oracle</td>
<td>2003-03-15</td>
</tr>
</tbody>
</table>
Oracle Text and XML

- Oracle Text is the full text search engine in Oracle Database
- Free with all versions of the database
- The power of a standalone search engine plus full integration with the Oracle stack
- Can perform fast free-text search within XML text
  
  \[
  <title>Crouching Tiger, Hidden Dragon</title>
  \]
  … contains(movieInfo, ‘tiger within title’) …

- Result Set Interface (new in 11.2.0.2) allows you to
  - Specify Query request and hitlist requirements in XML
  - Fetch Hitlist as XML
Indexes for Query Performance

- XMLIndex
  - Path Subsetted
  - Asynchronous maintenance
  - Structured XML Index

- Text Index
  - AUTO LEXER
  - Path Section Group
  - Interval Sync
  - Asynchronous maintenance
Querying XML Content in XML DB

SQL/XML

XQuery

XMLType Abstraction

DB XQuery

Procedural XQuery

XQuery Rewrite

Pushdown

XVM

Functional Evaluation

DOM Tree Model

XMLIndex

Pushdown

Relational Access Methods

Streaming XPath Evaluation

Object-Relational

Relational Storage

Binary XML

Secure Files
Binary XML - Comparison with another DB

1/3rd the size 3x faster

<table>
<thead>
<tr>
<th>Storage needed for TPoX data</th>
<th>Mean TPoX Query Response functional eval</th>
<th>Storage needed for XMark data</th>
<th>Mean XMark Query Response functional eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>601MB</td>
<td>1821 msec</td>
<td>67MB</td>
<td>451 msec</td>
</tr>
<tr>
<td>189MB</td>
<td>508 msec</td>
<td>10MB</td>
<td>161 msec</td>
</tr>
</tbody>
</table>

Oracle...
TPoX Benchmark

Comparison of Oracle XML DB with another DB (with Indexes)
Conclusion

• Proquest live on 11gR2 with XML DB
  • Focus on Ingestion speed and scalability
  • Binary XML Storage
  • Range Partitioning
  • 1TB of data

• Prototype underway for Content Store
  • Focus on Query performance
  • XML Index
  • Text Index