Oracle Database 11g Semantic Technologies Overview

Zhe Wu, Ph.D.
Oracle Database Semantic Technologies

Sept. 2010
## Semantic at OOW 2010 - Sessions

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday, Sept 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:30 p.m.</td>
<td>How and Why Customers Use Oracle’s Semantic Database Technologies: A Panel</td>
<td>Moscone South Room 200</td>
</tr>
<tr>
<td>2:00 p.m.</td>
<td>Electronic Medical Records with Oracle Semantic Technologies at Cleveland Clinic</td>
<td>Moscone South Room 200</td>
</tr>
<tr>
<td>4:00 p.m.</td>
<td>How Cisco’s Enterprise Collaboration Platform Uses Oracle Semantic Technologies</td>
<td>Hotel Nikko, Golden Gate</td>
</tr>
</tbody>
</table>

## Semantic at OOW 2010 – Hands-On Labs

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuesday, Sept 21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:00 p.m.</td>
<td>A Little Semantics Goes a Long Way with Oracle Database 11g</td>
<td>Hilton SF Franciscan A/B/ C/D</td>
</tr>
</tbody>
</table>

- DEMOgrounds
  - Semantic Database Technologies - *Moscone West, W-045*
The following 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.
Agenda

• Introduction
  • Semantic technology stack

• Overview of release 11g Capabilities
  • Architecture/Query/Store/Inference/Java APIs

• Performance and scalability evaluation
Semantic Technology Stack

- **Basic Technologies**
  - **URI**
    - Uniform resource identifier
  - **RDF**
    - Resource description framework
  - **RDFS**
    - RDF Schema
  - **OWL**
    - Web ontology language

URL: http://www.w3.org/2007/03/layerCake.svg
Semantic Application Workflow

**Transform & Edit Tools**
- Entity Extraction & Transform
  - OpenCalais
  - Linguamatics
  - GATE
  - D2RQ
- Ontology Eng.
  - TopQuadrant
  - Mondeca
  - Ontoprise
  - Protege
- Categorization
  - Cyc
- Custom Scripting

**Load, Query & Inference**
- RDF/OWL Data Management
- SQL & SPARQL
- Sesame Adapter
- Jena Adapter
- Native Inferencing
- Semantic Rules
- Scalability & Security
- Semantic Indexing

**Applications & Analysis Tools**
- BI, Analytics
  - Teranode
  - Metatomix
  - MedTrust
- Graph Visualization
  - Cytoscape
- Social Network Analysis
- Metadata Registry
- Faceted Search

**Partner Tools**
- OpenCalais
- Linguamatics
- GATE
- D2RQ
- TopQuadrant
- Mondeca
- Ontoprise
- Protege
- Cyc
- Custom Scripting

**Transaction Systems**
- Unstructured Content
- RSS, email
- Other Data Formats
- Data Sources
Oracle’s Partners for Semantic Technologies

Integrated Tools and Solution Providers:

<table>
<thead>
<tr>
<th>Ontology Engineering</th>
<th>Reasoners</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>TopQuadrant</td>
<td>clarkparsia, llc</td>
<td>TERANODE</td>
</tr>
<tr>
<td>Ontoprise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>protege</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Tool Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>semantic framework jena</td>
</tr>
<tr>
<td>Sesame</td>
</tr>
<tr>
<td>cYcorp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>W3C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NLP Entity Extractors</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALAIS</td>
</tr>
<tr>
<td>GATE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SI / Consulting</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORTHROP GRUMMAN</td>
</tr>
<tr>
<td>orbis technologies, Inc</td>
</tr>
<tr>
<td>BOEING</td>
</tr>
<tr>
<td>Cognia</td>
</tr>
</tbody>
</table>
Some Oracle Database Semantics Customers

Life Sciences
- Lilly
- Pfizer
- Swiss Institute of Bioinformatics

Defense/Intelligence
- National Geospatial-Intelligence Agency
- The University of Michigan

Clinical Medicine & Research
- Health Science Center at Houston
- Cleveland Clinic

Education
- The University of Texas

Telecomm & Networking
- Hutchinson 3G Austria

Publishing
- Thomson Reuters

Telecomm & Networking
- Cisco
# Capabilities Overview of Release 11.2

## NLP engines, Tools, Editors, Complete DL reasoners, …

## SQL/PLSQL APIs & JAVA APIs (Jena, Sesame)

### INFER
- RDF/S
- OWL/SKOS
- User defined rules

### QUERY
- Query RDF/OWL data and ontologies
- Ontology-Assisted Query of Enterprise Data

## STORE
- Incremental DML
- Batch-Load
- Bulk-Load

### Built-in Security and Versioning for semantic data
- RDF/OWL data
- Ontologies & rule bases

### Relational data
Store Semantic Data

• Native graph data store in Oracle Database
  • Implemented using relational tables/views
  • Optimized for semantic data
• Scales to very large datasets
  • No limits to amount of data that can be stored
• Stored along with other relational data
  • Leverages decades of experience
  • Can be combined with other relational data
  • Business Data
  • XML
  • Location
  • Images, Video
Infer Semantic Data

- Native inferencing in the database for
  - RDF, RDFS, and a rich subset of OWL semantics (OWLSIF, OWLPRIME, RDFS++)
  - User-defined rules
- Forward chaining.
  - New relationships/triples are inferred and stored ahead of query time
  - Removes on-the-fly reasoning and results in fast query times
- Proof generation
  - Show one deduction path
Query Semantic Data

• Choice of SQL or SPARQL
• SPARQL-like graph queries can be embedded in SQL
  • Key advantages
    • Graph queries can be integrated with enterprise relational data
    • Graph queries can be enhanced with relational operators.
      • E.g. replace, substr, concatenation, to_number, …

• Jena Adapter/Sesame Adapter for Oracle can be used, includes a full SPARQL API
Analyze Semantic Data

- Treat semantic data as a data source to business intelligence, such as OBIEE
- Logical tables/columns can be mapped to views/columns created based on semantic queries.
Java APIs: Jena Adapter

• Implements Jena’s Graph/Model/BulkUpdateHandler/… APIs

• “Proxy” like design
  • Data not cached in memory for scalability
  • SPARQL query converted into SQL and executed inside DB
    • A SPARQL with just conjunctive patterns is converted into a single SEM_MATCH query

• Allows various data loading
  • Bulk/Batch/Incremental load RDF or OWL (in N3, RDF/XML, N-TRIPLE etc.) with strict syntax verification and long literal support

• Integrates Oracle Database 11g RDF/OWL with tools including
  • TopBraid Composer
  • External complete DL reasoners (e.g. Pellet)

Release 11g RDF/OWL Usage Flow

- Create an application table
  - `create table app_table(triple sdo_rdf_triple_s);`
- Create a semantic model
  - `exec sem_apis.create_sem_model('family', 'app_table', 'triple');`
- Load data
  - Use DML, Bulk loader, or Batch loader
    - `insert into app_table (triple) values(1, sdo_rdf_triple_s('family', '<http://www.example.org/family/Matt>', '<http://www.example.org/family/fatherOf>', '<http://www.example.org/family/Cindy>'));`
  - `exec sem_apis.analyze_model('family');`
- Run inference
  - `exec sem_apis.create_entailment('family_idx', sem_models('family'), sem_rulebases('owlprime'));`
  - `exec sem_apis.analyze_rules_index('family_idx');`
- Query both original model and inferred data
  - `select p, o from table(sem_match('(<http://www.example.org/family/Matt> ?p ?o>', sem_models('family'), sem_rulebases('owlprime'), null, null));`

After inference is done, what will happen if
- **New assertions are added to the graph**
  - Inferred data becomes incomplete. Existing inferred data **will be reused** if `create_entailment` API invoked again. Faster than rebuild.
- **Existing assertions are removed from the graph**
  - Inferred data becomes invalid. Existing inferred data **will not be reused** if the `create_entailment` API is invoked again.
Release 11g RDF/OWL Usage Flow in Java

• Create an Oracle object
  • oracle = new Oracle(oracleConnection);

• Create a GraphOracleSem Object
  • graph = new GraphOracleSem(oracle, model_name, attachment);

• Load data
  • graph.add(Triple.create(...));  // for incremental triple additions

• Collect statistics
  • graph.analyze();

• Run inference
  • graph.performInference();

• Collect statistics
  • graph.analyzeInferredGraph();

• Query
  • QueryFactory.create(...);
  • queryExec = QueryExecutionFactory.create(query, model);
  • resultSet = queryExec.execSelect();

No need to create model manually!

Important for performance!
Enterprise Security for Semantic Data

- RDF data security for defense and intelligence, and the commercial regulatory environment
  - Intercept and rewrite the user query to restrict the result set using additional predicates and return only “need to know” data

- Access control policies on semantic data
  - Uses Virtual Private Database feature of Oracle Database
  - Applies constraints to classes and properties
  - Restricts access to parts of the RDF graph based on the application/user context

- Data classification labels for semantic data
  - Uses Oracle Label Security option of Oracle Database
  - Assigns sensitivity labels to users and RDF data.
  - Restricts access to users having compatible access labels.
Semantic Indexing for Documents

- Links people – places – things – events to documents stored in Oracle Database though a semantic index

- Extends the power of Oracle Database to include semantic search in cross-domain queries.

- Key Components
  - Programmable API to plug-in 3rd party entity extractors
    - E.g. OpenCalais from Thomson Reuters
  - SEM_CONTAINS Operator
  - SEM_CONTAINS_SELECT Ancillary Operator
  - SemContext Index type
Semantic Indexing and Query Flow

- Extracting RDF from documents

<table>
<thead>
<tr>
<th>DocId</th>
<th>Article</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>1</td>
</tr>
<tr>
<td>r2</td>
<td>2</td>
</tr>
<tr>
<td>..</td>
<td></td>
</tr>
</tbody>
</table>

**Newsfeed table**

**Triples table**

<table>
<thead>
<tr>
<th>NG</th>
<th>Subject</th>
<th>Property</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>p:Marcus</td>
<td>rdf:type</td>
<td>rc:Person</td>
</tr>
<tr>
<td>r1</td>
<td>p:Marcus</td>
<td>pred:hasName</td>
<td>&quot;Marcus&quot;^^xsd:string</td>
</tr>
<tr>
<td>r1</td>
<td>p:Marcus</td>
<td>pred:hasAge</td>
<td>&quot;38&quot;^^xsd:integer</td>
</tr>
<tr>
<td></td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>r2</td>
<td>c:AcmeCorp</td>
<td>rdf:type</td>
<td>rc:Organization</td>
</tr>
</tbody>
</table>

- Semantic query through SEM_CONTAINS

```sql
SELECT docId, SEM_CONTAINS_SELECT(1) binding FROM Newsfeed
WHERE SEM_CONTAINS (article,
    '{ ?org pred:categoryName c:BusinessFinance .
      ?org pred:score ?score .
      FILTER (?score > 0.5)}', 1 ) = 1
```
Change Mgmt./Versioning for Semantic Data

- Manage public and private versions of semantic data in database workspaces *(Workspace Manager)*

- An RDF Model is version-enabled by version-enabling its application table.

- Application table data modified within a workspace is private to the workspace until it is merged.

- SEM_MATCH queries on version-enabled models are version aware and only return relevant data.
  - New versions created only for changed data

- Versioning is provisioned for inference
Performance and Scalability Evaluation
Setup for Performance (1)

- Use a balanced hardware system for database
  - A single, huge physical disk for everything is **not** recommended.
  - Multiple hard disks tied together through ASM is a good practice
- Make sure throughput of hardware components **match** up

<table>
<thead>
<tr>
<th>Component</th>
<th>Hardware spec</th>
<th>Sustained throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU core</td>
<td>-</td>
<td>100 - 200 MB/s</td>
</tr>
<tr>
<td>1/2 Gbit HBA</td>
<td>1/2 Gbit/s</td>
<td>100/200 MB/s</td>
</tr>
<tr>
<td>16 port switch</td>
<td>8 * 2 Gbit/s</td>
<td>1,200 MB/s</td>
</tr>
<tr>
<td>Fiber channel</td>
<td>2 Gbit/s</td>
<td>200 MB/s</td>
</tr>
<tr>
<td>Disk controller</td>
<td>2 Gbit/s</td>
<td>200 MB/s</td>
</tr>
<tr>
<td>GigE NIC (interconnect)</td>
<td>2 Gbit/s</td>
<td>80 MB/s*</td>
</tr>
<tr>
<td>Disk (spindle)</td>
<td></td>
<td>30 - 50 MB/s</td>
</tr>
<tr>
<td>MEM</td>
<td></td>
<td>2k-7k MB/s</td>
</tr>
</tbody>
</table>

Some numbers are from Data Warehousing with 11g and RAC presentation
Setup for Performance (2)

- Database parameters
  - SGA, PGA, filesystemio_options, db_cache_size, ...

- Linux OS Kernel parameters
  - shmmmax, shmall, aio-max-nr, sem, ...

- For Java clients using JDBC (Jena Adaptor)
  - Network MTU, Oracle SQL*Net parameters including SDU, TDU, SEND_BUF_SIZE, RECV_BUF_SIZE,
  - Linux Kernel parameters: net.core.rmem_max, wmem_max, net.ipv4.tcp_rmem, tcp_wmem, ...

- No single size fits all. Need to benchmark and tune!

## Bulk Loader Performance on Desktop PC: 11.2 Latest

<table>
<thead>
<tr>
<th>Ontology size</th>
<th>Time</th>
<th>Space (in GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bulk-load API(^2) Time (incl. Parse)</td>
<td>Sql*loader time (^3)</td>
</tr>
<tr>
<td>LUBM50 6.9 million</td>
<td>2.6min</td>
<td>0.4min</td>
</tr>
<tr>
<td>LUBM1000 138.3 million</td>
<td>1hr 10min</td>
<td>8 min</td>
</tr>
<tr>
<td>LUBM8000 1,106 million</td>
<td>9hr 15min</td>
<td>1hr 5min</td>
</tr>
</tbody>
</table>

- Used Core 2 Duo PC (3GHz), 8GB RAM, ASM, 3 SATA Disks (7200rpm), 64 bit Linux. Planned for an upcoming patchset.
- Empty network is assumed

\(^1\) This is an internal version of latest Oracle RDBMS 11.2
\(^2\) Uses flags=>' parse parallel=4 parallel_create_index ' plus a new as-yet-unnamed option for value processing
\(^3\) Uses parallel=true option and 8 to 10 gzipped N-Triple files as data files and a no-parse control file. \(^4\) Application table has table compression enabled.
\(^5\) Staging table has table compression enabled.
### Query Performance on Desktop PC

<table>
<thead>
<tr>
<th>Ontology LUBM50 6.8 million &amp; 5.4 million inferred</th>
<th>LUBM Benchmark Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query</strong></td>
<td><strong>Q1</strong></td>
</tr>
<tr>
<td># answers</td>
<td>4</td>
</tr>
<tr>
<td>Complete?</td>
<td>Y</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>0.05</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td><strong>Q8</strong></td>
</tr>
<tr>
<td># answers</td>
<td>7790</td>
</tr>
<tr>
<td>Complete?</td>
<td>Y</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>1.07</td>
</tr>
</tbody>
</table>

- Setup: Intel Q6600 quad-core, 3 7200RPM SATA disks, 8GB DDR2 PC6400 RAM, No RAID. 64-bit Linux 2.6.18. Average of 3 warm runs
• OWLPrime (11.1.0.7) inference performance scales really well with hardware. It is *not* a parallel inference engine though.
### 11.2.0.1 Inference Performance on Desktop PC

<table>
<thead>
<tr>
<th>Inference Type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parallel Inference (LUBM8000)</strong></td>
<td>1.06 billion triples + 860M inferred</td>
</tr>
<tr>
<td>Time to finish inference: 12 hrs.</td>
<td></td>
</tr>
<tr>
<td>3.3x faster compared to serial inference in release 11.1</td>
<td></td>
</tr>
<tr>
<td><strong>Parallel Inference (LUBM25000)</strong></td>
<td>3.3 billion triples + 2.7 billion inferred</td>
</tr>
<tr>
<td>Time to finish inference: 40 hrs.</td>
<td></td>
</tr>
<tr>
<td>30% faster than nearest competitor</td>
<td></td>
</tr>
<tr>
<td>1/5 cost of other hardware configurations</td>
<td></td>
</tr>
<tr>
<td><strong>Incremental Inference (LUBM8000)</strong></td>
<td>1.06 billion triples + 860M inferred</td>
</tr>
<tr>
<td>Time to update inference: less than 30 seconds after adding 100 triples.</td>
<td></td>
</tr>
<tr>
<td>At least 15x to 50x faster than a complete inference done with release 11.1</td>
<td></td>
</tr>
<tr>
<td><strong>Large scale owl:sameAs Inference</strong></td>
<td>(UniProt 1 Million sample)</td>
</tr>
<tr>
<td>60% less disk space required</td>
<td></td>
</tr>
<tr>
<td>10x faster inference compared to release 11.1</td>
<td></td>
</tr>
</tbody>
</table>

- **Setup:** Intel Q6600 quad-core, 3 7200RPM SATA disks, 8GB DDR2 PC6400 RAM, No RAID.
- 64-bit Linux 2.6.18. **Assembly cost:** less than USD 1,000
Load Performance on Server

- **LUBM1000 (138M triples)**
  - 8.3 minutes to load data into staging table
  - 78.8 minutes to load data from staging table (DOP=8)

- **LUBM8000 (1B+)**
  - 25 minutes to load data into staging table
  - 10hr 36 minutes to load data from staging table (DOP=8)

Setup: Dual quad-core, Sun Storage F5100 Flash Array, 32 GB RAM
Inference Performance on Server

• Inference performance for LUBM1000 (138M)
  • 24.6 minutes to infer 108M+ new triples (DOP=8)

• Inference performance for LUBM8000 (1B+)
  • 226 minutes to infer 860M+ new triples (DOP=8)

• Setup: Dual quad-core, Sun Storage F5100 Flash Array, 32 GB RAM
Query Performance on Server

- Parallel query execution

Setup: Server class machine with 16 cores, NAND based flash storage, 32GB RAM, Linux 64 bit, Average of 3 warm runs
Load Performance on Exadata V2

• LUBM 25K benchmark ontology
  (3.3 Billion triples)
  - (Note: These are preliminary numbers and will be updated.)
  - 105 minutes to load the data into staging table
  - 730 minutes for the bulk-load API, but with values pre-loaded

• Setup: Sun Oracle Data Machine and Exadata Storage Server (8 node cluster, Full Rack)

Inference Performance on Exadata V2

• LUBM 25K benchmark ontology (3.3 Billion triples)
  – OWLPrime inference with new inference components took 247 minutes (4 hours 7 minutes)
  – More than 2.7 billion new triples inferred
  – DOP = 32

• Preliminary result on LUBM 100K benchmark ontology (13 Billion+ triples)
  – One round of OWLPrime inference (limited to OWL Horst semantics) finished in 1.97 hours
  – 5 billion+ new triples inferred
  – DOP = 32

• Setup: Full Rack Sun Oracle Data Machine and Exadata Storage Server (8 node cluster)
## Query Performance on Exadata V2

<table>
<thead>
<tr>
<th>Ontology LUBM25K</th>
<th>3.3 billion &amp; 2.7 billion inferred</th>
<th>LUBM Benchmark Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query</strong></td>
<td>Q1</td>
<td>Q2</td>
</tr>
<tr>
<td># answers</td>
<td>4</td>
<td>2528</td>
</tr>
<tr>
<td>Complete?</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>0.01</td>
<td>20.65</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>Q8</td>
<td>Q9</td>
</tr>
<tr>
<td># answers</td>
<td>7790</td>
<td>6.8M</td>
</tr>
<tr>
<td>Complete?</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>0.48</td>
<td>203.06</td>
</tr>
</tbody>
</table>

- **Setup:** Full Rack Sun Oracle Data Machine and Exadata Storage Server (8 node cluster)
- **Auto DOP is used.** Total # of answers 465,849,803 in less than 5 minutes
For More Information

http://search.oracle.com

semantic technologies
ORACLE IS THE INFORMATION COMPANY