Semantic Technologies in Oracle Database 11g Release 2: Capabilities, Interfaces, Performance

Xavier Lopez, Ph.D., Director
Souripriya Das, Ph.D., Architect
<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Title</th>
<th>Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tues, 22-Jun-2010</td>
<td><strong>Oracle Database Semantic Technologies: Understanding How to Install, Load, Query and Inference</strong></td>
<td>Semantics for Enterprise Data</td>
</tr>
<tr>
<td>1:15 - 4:45 p.m.</td>
<td><strong>Semantic Technologies in Oracle Database 11gR2: Capabilities, Interfaces, Performance</strong></td>
<td>Semantics for Enterprise Data</td>
</tr>
<tr>
<td>Thurs, 24-Jun-2010</td>
<td><strong>An Enterprise-Class Inference Engine Inside Oracle Database 11g Release 2</strong></td>
<td>Semantic Rules (incl. RuleML, RIF, SILK, RL Profile)</td>
</tr>
</tbody>
</table>
Agenda

• **Intro** to Semantic Web and Business Use Cases
• Architecture and Storage
• Capabilities
• Interfaces
• Performance
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 DECISION. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITY DESCRIBED FOR ORACLE’S PRODUCTS REMAINS AT THE SOLE DISCRETION OF ORACLE.
Introduction to Semantic Web and Business Use Cases
Semantic Technology Stack

- **Basic Technologies**
  - **URI**
    - Uniform Resource Identifier
  - **RDF**
    - Resource Description Framework
  - **RDFS**
    - RDF Schema
  - **OWL**
    - Web ontology language
  - **SPARQL**
    - Protocol and Query Language
## Extraction, Modeling, Reasoning & Discovery Workflow

<table>
<thead>
<tr>
<th>Transform &amp; Edit Tools</th>
<th>Load, Query &amp; Inference</th>
<th>Applications &amp; Analysis Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity Extraction &amp; Transform</td>
<td>RDF/OWL Data Management</td>
<td>BI Analytics</td>
</tr>
<tr>
<td>Ontology Engineering</td>
<td>SQL &amp; SPARQL Query</td>
<td>Graph Visualization</td>
</tr>
<tr>
<td>Categorization</td>
<td>Inferencing</td>
<td>Social Network Analysis</td>
</tr>
<tr>
<td>Custom Scripting</td>
<td>Semantic Rules</td>
<td>Metadata Registry</td>
</tr>
<tr>
<td></td>
<td>Security</td>
<td>Faceted Search</td>
</tr>
<tr>
<td></td>
<td>Semantic Indexing</td>
<td>SPARQL Endpoint</td>
</tr>
<tr>
<td></td>
<td>Versioning</td>
<td></td>
</tr>
</tbody>
</table>

### Partner Tools
- Entity Extraction & Transform
- Ontology Engineering
- Categorization
- Custom Scripting

### Applications & Analysis Tools
- BI Analytics
- Graph Visualization
- Social Network Analysis
- Metadata Registry
- Faceted Search
- SPARQL Endpoint

### Partner/Oracle Tools
- Oracle Spatial
Why Organizations use Oracle RDF Database

- Oracle database 11g is the leading commercial database with native RDF/OWL data management
- Scalable & secure platform for wide-range of semantic applications
- Readily scales to ultra-large repositories (10s billions of triples)
- Choice of SQL or SPARQL query
- Leverages Oracle Partitioning. RAC supported
- Growing ecosystem of 3rd party tools partners

Key Capabilities:

**Load / Storage**
- Native RDF graph data store
- Manages billions of triples
- Fast batch, bulk and incremental load

**Query**
- SQL: SEM_Match
- SPARQL: via Jena plug-in
- Ontology assisted query of RDBMS data

**Reasoning**
- Forward chaining model
- OWLprime, OWL 2 RL, RDFS
- User defined rule base
National Intelligence: Text Mining

1. Unstructured Data (Text)
   - Blogs, open source, newsfeed
   - Signal Intelligence, message traffic
   - Analyst Reports (Content Mgmt)

2. Model
   - Entity Extraction Engine
     - Feature/term/relation Extraction, categorization (Insight, Lymba, Calais, Gate)

3. Structured Data
   - organization
   - founder
   - person
   - "Oracle's founder Larry Ellison wins 2010 America's Cup Race …"

4. Mining & Discovery
   - SPARQL/SQL
   - XML/OWL/N3
   - Triple Structure: Subj – Pred - Obj
   - 10's of billions of triples

Oracle's founder Larry Ellison wins 2010 America's Cup Race …

Explore

Browsing, Presentation, Reporting, Visualization, Query Tools (e.g. i2, Centrifuge, Visual Analytics)
Data Integration Platform in Health Informatics

Enterprise Information Consumers (EICs)

Access

Run-Time Metadata

Integration Server (Semantic Knowledge base)

Deploy

Design-Time Metadata

Access

Model Virtual

Relate

Model Physical

LIS  CIS  HTB  HIS
Oracle Database 11g
Ontology repository

Oracle Text 11g
For text based search

Oracle RDF 11g
For graph based search

Semantic Portal

Vocabulary Management

Intelligent Topic Manager (ITM)

Manage
Edit
Maintain
Search
Control
Import
Export
Audit

Ontology model (OWL)
Terminology – Taxonomy
(RDF SKOS)
Knowledge representation
Catalog - Yellow pages (RDF)

ITM Features
Web User Interfaces
Multilingual
Connectors to text mining
Collaborative maintenance
Import / export
Scalability
API & Web Services
Java – J2EE – LDAP

Oracle Database 11g
Ontology repository
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>protégé</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Tool Interfaces</th>
<th>Standards</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jena</td>
<td>W3C</td>
<td>Teranode</td>
</tr>
<tr>
<td>openRDF.org</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sesame</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joseki</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cYcorp</td>
<td>RDF</td>
<td>MedTrust</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NLP Entity Extractors</th>
<th>SI / Consulting</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALAIS</td>
<td>Northrop Gruman</td>
</tr>
<tr>
<td>Linguamatics</td>
<td>Raytheon</td>
</tr>
<tr>
<td>Gate</td>
<td>Orbis Technologies, Inc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General Architecture for Text Engineering</th>
<th>Boeing</th>
<th>McDonald Bradley</th>
<th>Cognia</th>
<th>Accenture</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Some Oracle Database Semantics Customers

Life Sciences
- Lilly
- Pfizer
- Swiss Institute of Bioinformatics

Defense/Intelligence
- GeoSpatiaIntelligence National Geospatial-Intelligence Agency

Education
- The University of Michigan

Clinical Medicine & Research
- The University of Texas Health Science Center at Houston
- Cleveland Clinic

Telecomm & Networking
- Hutchinson 3G Austria
- Cisco

Publishing
- Thomson Reuters
Semantic Technologies in Oracle 11g Release 2
Agenda

- Intro to Semantic Web and Business Use Cases
- Architecture and Storage
- Capabilities
  - Fundamentals
    - Load / Query / Inference
  - Enterprise
    - Semantic indexing of docs
    - fine-grained Security (at triple level)
    - Versioning
- Interfaces
  - SQL-based
  - Java-based: Jena Adapter and Sesame Adapter
  - SPARQL Endpoints
- Performance
Architectural Overview

SPARQL Endpoints: Joseki / Sesame

3rd Party Tools: Topbraid Composer

Reasoners: Pellet

NLP Info. Extractor: Calais, GATE

Java API support

SPARQL: Jena / Sesame

Java Programs

SQL Interface: SQLplus, PL/SQL, SQLdev.

Tools (cytoscope)

3rd Party Callouts

Programming Interface

Java Programs

JDBC

SQL Interface

SPARQL Endpoints

Joseki / Sesame

3rd Party Tools

Topbraid Composer

Reasoners: Pellet

NLP Info. Extractor: Calais, GATE

Core functionality

LOAD

Bulk-Load

Incr. DML

INFER

OWL subsets

RDF/S

User-def.

QUERY (SQL-based SPARQL)

Query RDF/OWL data and ontologies

Ontology-assisted Query of Enterprise Data

Oracle DB

RDF/OWL data and ontologies

Rulebases: OWL, RDF/S, user-defined

Inferred RDF/OWL data

Semantic Indexes

Enterprise (Relational) data

Security: fine-grained

Versioning: Workspaces

Oracle DB

RDF/OWL data and ontologies

Rulebases: OWL, RDF/S, user-defined

Inferred RDF/OWL data

Semantic Indexes

Enterprise (Relational) data

Security: fine-grained

Versioning: Workspaces
Core Entities in Oracle Database Semantic Store

- **Sem. Network**: Dictionary and data tables. **New entities**: models, rule bases, and rules indexes (entailments). OWL and RDFS rule bases are preloaded.

- **SDO_RDF_TRIPLE_S**: A new object type for RDF.

- **Application Table**: Contains col of object type sdo_rdf_triple_s to allow loading and accessing RDF triples, and storing ancillary values.

- **Model**: A model holds an RDF graph (set of S-P-O triples) and is associated with an sdo_rdf_triple_s column in an application table.

- **Rulebase**: A rulebase is a set of rules used for inferencing.

- **Entailments**: An entailment stores triples derived via inferencing.
Load / Query / Inference

- **Load**
  - Bulk load
  - Incremental load

- **Query**
  - SPARQL (direct or SQL-based)
  - Simple data access

- **Inference**
  - Using OWL 2 RL, RDFS, etc.
  - Using user-defined rules

Oracle Database

- Rulebases & Vocabularies
  - OWL subset
  - RDF / RDFS
  - Rule base \( m \)

Application Tables

- Models
  - \( M_1 \)
  - \( M_2 \)
  - \( M_n \)

Semantic Network (MDSYS)

- Entailments
  - \( X_1 \)
  - \( X_2 \)
  - \( X_p \)

- Triples

- Oracle Database
Interfaces

- SQL-based (SQL and PL/SQL)

- Java-based
  - Jena (using Jena Adapter from Oracle)
  - Sesame (using Sesame Adapter from Oracle)

- SPARQL Endpoints
  - Joseki
  - OpenRDF Workbench
SPARQL Query Architecture

HTTP

Java

Jena API
Jena Adapter

Sesame API
Sesame Adapter

Standard SPARQL Endpoint

SQL

SEM_MATCH
SEM_MATCH: Adding SPARQL to SQL

SPARQL

PREFIX foaf: <http://...>
SELECT ?n1 ?n2
FROM <http://g1>
WHERE
{
?p foaf:name ?n1
  OPTIONAL {?p foaf:knows ?f .
    ?f foaf:name ?n2 }
FILTER (REGEX(?n1, "^A"))
}

SQL

SELECT n1, n2
FROM TABLE(SEM_MATCH(
  '{?p foaf:name ?n1
    OPTIONAL {?p foaf:knows ?f .
      ?f foaf:name ?n2 }
    FILTER (REGEX(?n1, "^A")) }',
  SEM_MODELS('g1'),...,
  SEM_ALIASES(  
    SEM_ALIAS('foaf', 'http://...'))), ...)
**SEM_MATCH: Adding SPARQL to SQL**

**Rewritable SQL Table Function**

```sql
SELECT n1, n2
FROM
(SELECT v1.value AS n1, v2.value AS n2
FROM VALUES v1, VALUES v2
TRIPLES t1, TRIPLES t2, ...
WHERE t1.obj_id = v1.value_id
AND t1.pred_id = 1234
AND ...
)
```

**Get 1 declarative SQL query**
- Query optimizer sees 1 query
- Get all the performance of Oracle SQL Engine
  - compression, indexes, parallelism, etc.
GeoSpatial Querying in an RDBMS

SQL-Based Approach

Oracle 11g

Object-Relational Data (Spatial Objects)

Triple Store (Semantic Data)

Geo-Semantic SQL-Based Querying: SEM_MATCH + SDO_RELATE
Ontology-assisted Query using SQL Operators

SELECT p_id, diagnosis FROM Patients
WHERE SEM_RELATED (diagnosis,
    'rdfs:subClassOf',
    'Upper_Extremity_Fracture',
    sem_models('Medical_ontology'),
    sem_rulebases('RDFS') ...
    123) = 1
AND SEM_DISTANCE(123) <= 2;
Inference

- Inference done using forward chaining
  - Triples inferred and stored ahead of query time
  - Removes on-the-fly reasoning and results in fast query times
- Various native rulebases provided
  - E.g., RDFS, OWL 2 RL, SNOMED (EL+), SKOS
- Validation of inferred data
- User-defined rules
- Proof generation
  - Shows one deduction path
OWL Subsets Supported

- **OWL subsets for different applications**
  - RDFS++
    - RDFS plus owl:sameAs and owl:InverseFunctionalProperty
  - OWLSIF (OWL with IF semantics)
    - Based on Dr. Horst’s pD* vocabulary¹
  - OWLPrime
    - Includes RDFS++, OWLSIF with additional rules
    - Jointly determined with domain experts, customers and partners
  - OWL 2 RL
    - W3C Standard
    - Adds rules about keys, property chains, unions and intersections to OWLPrime
  - SNOMED

¹ Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary
11g Release 2 Inference Features

• Richer semantics support
  • OWL 2 RL, SKOS, SNOMED (subset of OWL 2 EL)

• Performance enhancements
  • Large scale owl:sameAs handling
    • Compact materialization of owl:sameAs closure
  • Parallel inference
    • Leverage native Oracle parallel query and parallel DML
  • Incremental inference
    • Efficient updates of inferred graph through additions
  • Compact Data Structures
Semantic Indexing of Documents

CREATE INDEX ArticleIndex ON NewsFeed (Article) INDEXTYPE IS SemContext PARAMETERS (‘gate_nlp’)

Newsfeed table

<table>
<thead>
<tr>
<th>docId</th>
<th>Article</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Indiana authorities filed felony charges and a court issued an arrest warrant for a financial manager who apparently tried to fake his death by crashing his airplane in a Florida swamp. Marcus Schrenker, 38 ...</td>
<td>CNN</td>
</tr>
<tr>
<td>2</td>
<td>Major dealers and investors ...</td>
<td>NW</td>
</tr>
</tbody>
</table>

SELECT docId FROM Newsfeed WHERE SEM_CONTAINS (Article, ‘{?x rdf:type rc:Person . ?x :hasAge ?age . FILTER(?age >= 35)}’) = 1 AND Source = ‘CNN’

Analytical Queries On Graph Data
Enterprise Security for Semantic Data: VPD (Virtual Private Database)

[VPD] Access control policies on semantic data

**Policy ➔ user can access value of projects s/he leads**

**Match pattern ➔**
\[
\{ \texttt{?x :hasValue ?v} \}
\]

**Apply pattern ➔**
\[
\{ \texttt{?x :hasLead \texttt{“sys\_context(…)”}} \}
\]

**Query: Get the list of projects and their values**

```
SELECT  ?proj  ?val
FROM    ProjectsGraph
WHERE   { ?proj :hasValue ?val .
           ?proj :hasLead \texttt{“sys\_context(…)”} }
```
Enterprise Security for Semantic Data: OLS (Oracle Label Security)

[OLS] Data classification labels for semantic data

**SELECT operation:** Labels for triples are used to restrict access to the triples.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
<th>Row Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>projectHLS</td>
<td>Organization</td>
<td>N.America</td>
<td>SE:HLS:US</td>
</tr>
<tr>
<td>projectHLS</td>
<td>ContractValue</td>
<td>1000000</td>
<td>SE:HLS,FIN:US</td>
</tr>
</tbody>
</table>

**INSERT operation:** Labels for resources to control their use as component of a triple during insertion. (Also, determines min. access label for inserted triple.)

<table>
<thead>
<tr>
<th>Position &amp; Resource</th>
<th>Row Label</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subject:</strong> projectHLS</td>
<td>SE:HLS:US</td>
</tr>
<tr>
<td><strong>Predicate:</strong> hasContractValue</td>
<td>TS:FIN:</td>
</tr>
</tbody>
</table>

**Label Security Modes**
- **Triple-level:** Easier to use. Any one can insert, bulk-load, do inference. (Inserted/inferred triples get user’s label. Query/Infer on visible triples.)
- **Resource-level:** Secures each resource independently.
Change Mgmt./Versioning for Sem. Data

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

- An RDF Model is version-enabled by version-enabling the corresponding application table.
  - exec DBMS_WM.enableVersioning (table_name => 'contracts_rdf_data');
- RDF 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
Utility APIs

- **SEM_APIS.merge_models**
  - Can be used to clone model as well.
  - e.g. exec sem_apis.merge_models('model1’,’model2’);

- **SEM_APIS.alter_model**
  - e.g. sem_apis.alter_model(‘m1’, ’MOVE’, ’TBS_SLOWER’);

- **SEM_APIS.swap_names**
  - e.g. exec
    sem_apis.swap_names(’production_model’,’prototype_model’);

- **SEM_APIS.rename_model/rename_entailment**

- **SEM_APIS.remove_duplicates**
  - e.g. exec sem_apis.remove_duplicates(’graph_model’);
Agenda

• Intro to Semantic Web and Business Use Cases
• Architecture and Storage
• Capabilities
  • Fundamentals
    • Load / Query / Inference
  • Enterprise
    • Semantic indexing of docs
    • fine-grained Security (at triple level)
    • Versioning
• Interfaces
  • SQL-based
  • Java-based: Jena Adapter and Sesame Adapter
  • SPARQL Endpoints
• Performance
Query Performance tuning: Network Indexes

- Some B-tree indexes (including one for enforcing uniqueness constraint) are created automatically at sem. network creation time.
- The set of usable indexes can be manipulated using
  - `sem_apis.add_sem_index`
  - `sem_apis.drop_sem_index`
  - `sem_apis.alter_index_on_model`
  - `sem_apis.alter_index_on_entailment`
- **Tip**: when creating a network index, use the `model_id` column as part (usually, the last col) of the index key
Best Practice: Virtual Model

• Similar to simple database Views
• One may define a *Virtual Model* consisting of multiple sem. Models, and optionally the corresponding entailment.
• The relevant subprograms are
  – `sem_apis.create_virtual_model`
  – `sem_apis.drop_virtual_model`
• Benefits
  – Allows ease and flexibility in specifying target models plus (optionally) the corresponding entailment
  – Facilitates access control
  – Query performance (esp. for queries against multi-model, or single/multi-model+entailment RDF data source)
Bulk-Load performance tuning

- Various options for “flags” parameter
  - Parallel=<n>
  - Parallel_Create_Index
  - Join hints for sub-steps (e.g., MBV_JOIN_HINT=USE_HASH)
- ASM
- temp tablespace should be on a different disk than data and indexes
- For detailed event tracing, use RDF$ET_TAB table
Inference performance tuning

- SameAs optimization
- Incremental inference
- Parallel inference
Tuning Tips for Best Inference Performance

• Analyze models before running inference
  • execute immediate sem apis.analyze_model(...);

• Need a **balanced** hardware setup to use parallel inference
  • E.g., a server with multi-core/multi-cpu processors and ample I/O throughput
  • Use Oracle Automatic Storage Management (ASM) to manage the disks

• Use RAW8=T option for compact data structures
  • Smaller data structures imply less I/O

• Dynamic incremental inference
  • Selectively applies semi-naïve rule evaluation while generating the entailment
  • Off by default, could be turned on with DYN_INC_INF=T option
Query Hint in SEM_MATCH

SELECT e, loc FROM
TABLE(SEM_MATCH( ‘{?e rdf:type :SoftwareEngineer .
                ?e :worksFor ?company .
                ?company :headQuarterLoc ?loc}’,
              SEM_Models(‘emp’, ‘gmap’), SEM_Rulebases(‘OWLPRIME’),
              SEM_ALIASES(SEM_ALIAS(″, 'http://example.org/’)),NULL,NULL,
‘ HINT0 = { LEADING(t0 t1 t2) USE_NL(t1) ‘}));

- HINTS in SEM_MATCH query
  - Similar to SQL hints
    - Join order: LEADING(…), ORDERED
    - Join type: USE_HASH(…), USE_NL(…), …
    - Access path: INDEX(…), …
  - Aliases are a bit different
    - Variable names: ?e, ?company, ?loc, …
    - Pattern ordinal based: t0, t1, t2, …
Query Performance Tuning: special hints

- Use ALLOW_DUP=T in the case of multi-model queries
Performance on Desktop PC
## Bulk Loader Performance on Desktop PC
(Core2Duo, 8GB memory, ASM with 3 disks)

<table>
<thead>
<tr>
<th>Ontology size</th>
<th>Time on 11.2 Latest (internal) [1] (Core 2 Duo, 8GB, ASM)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Parallel=true</td>
</tr>
<tr>
<td>LUBM50 6.9 million</td>
<td>0.4 min</td>
</tr>
<tr>
<td>LUBM500 69 million</td>
<td>4 min</td>
</tr>
<tr>
<td>LUBM1000 138 million</td>
<td>8 min</td>
</tr>
<tr>
<td>LUBM8000 1,106 million</td>
<td>65 min</td>
</tr>
</tbody>
</table>

---

1. Assumes empty network to start with.  
2. Uses flags=>' parse parallel=4 parallel_create_index ' plus a shadow 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.
## Query Performance

### Ontology LUBM50

<table>
<thead>
<tr>
<th># answers</th>
<th>Time (sec)</th>
<th>Complete?</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.05</td>
<td>Y</td>
</tr>
<tr>
<td>130</td>
<td>0.75</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>0.20</td>
<td>Y</td>
</tr>
<tr>
<td>34</td>
<td>0.5</td>
<td>Y</td>
</tr>
<tr>
<td>719</td>
<td>0.22</td>
<td>Y</td>
</tr>
<tr>
<td>519842</td>
<td>1.86</td>
<td>Y</td>
</tr>
<tr>
<td>67</td>
<td>1.71</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># answers</th>
<th>Time (sec)</th>
<th>Complete?</th>
</tr>
</thead>
<tbody>
<tr>
<td>7790</td>
<td>1.07</td>
<td>Y</td>
</tr>
<tr>
<td>13639</td>
<td>1.65</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>Y</td>
</tr>
<tr>
<td>224</td>
<td>0.02</td>
<td>Y</td>
</tr>
<tr>
<td>15</td>
<td>0.03</td>
<td>Y</td>
</tr>
<tr>
<td>228</td>
<td>0.01</td>
<td>Y</td>
</tr>
<tr>
<td>393730</td>
<td>1.47</td>
<td>Y</td>
</tr>
</tbody>
</table>

### LUBM Benchmark Queries

<table>
<thead>
<tr>
<th>Query</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>Q9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q11</td>
<td></td>
<td></td>
<td></td>
<td>130</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
<td></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. Average of 3 warm runs
# 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>Query</td>
<td>Q1</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>Query</td>
<td>Q8</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
## 11.2 Inference Performance

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

**LUBM1000 Query Performance**

- Setup: Server class machine with 16 cores, NAND based flash storage, 32GB RAM, Linux 64 bit, Average of 3 warm runs
Performance on Sun Oracle Database Machine (Exadata V2)
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 3.3 billion &amp; 2.7 billion inferred</th>
<th>LUBM Benchmark Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>Q1</td>
</tr>
<tr>
<td># answers</td>
<td>4</td>
</tr>
<tr>
<td>Complete?</td>
<td>Y</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>0.01</td>
</tr>
<tr>
<td>Query</td>
<td>Q8</td>
</tr>
<tr>
<td># answers</td>
<td>7790</td>
</tr>
<tr>
<td>Complete?</td>
<td>Y</td>
</tr>
<tr>
<td>Time (sec)</td>
<td>0.48</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
Oracle database is the leading commercial database with native support for W3C standards compliant RDF/OWL data store with comprehensive capabilities for reasoning and discovery.

**Reasoning and Discovery** supporting standard ontologies persistent, native & 3rd party inference, and user-defined rules.

**Scalability** to evolve schemas dynamically and grow to 10’s billions of triples, incremental & parallel inference.

**Data Integration** to link structured & unstructured content, loosely couple business silos.

**Security** to protect data on a “need to know” basis.

**Integrated querying & manageability** SPARQL & SQL for RDF/OWL, relational, XML, text, location, & multimedia data.
For More Information

search.oracle.com

Semantic Technologies

or

oracle.com
Oracle is the information company