Oracle Database 11g Semantic Technologies Overview

Zhe Wu, Ph.D.
Oracle Database Semantic Technologies

Sept. 2010

Semantic at OOW 2010 - Sessions

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

Semantic at OOW 2010 – Hands-On Labs

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

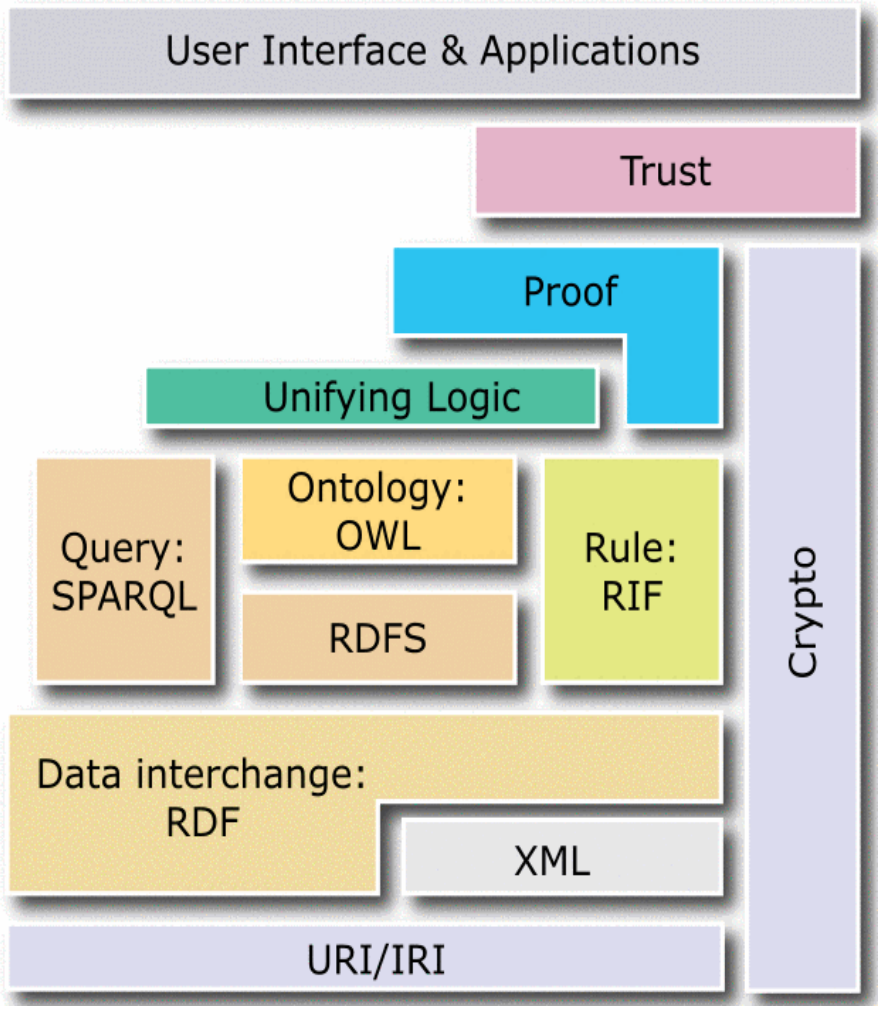
- DEMOgrounds
  - Semantic Database Technologies - *Moscone West, W-045*

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

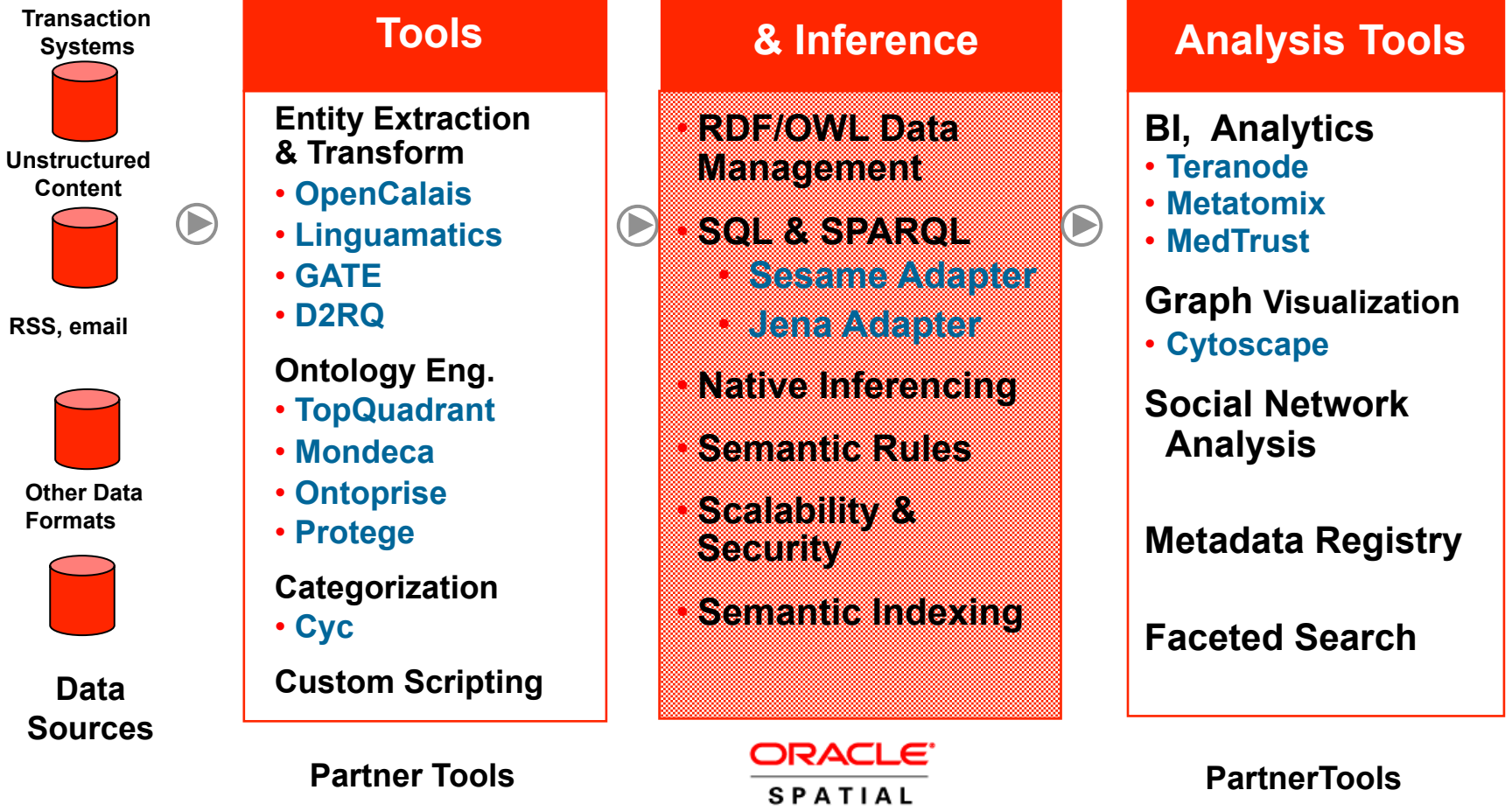
Semantic Technology Stack



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

http://www.w3.org/2007/03/layerCake.svg

Semantic Application Workflow



Oracle's Partners for Semantic Technologies

Integrated Tools and Solution Providers:

Ontology Engineering

TopQuadrant

ontoprise
know how to use Know-how

protégé

Reasoners

clark & parsia, llc

ontoprise
know how to use Know-how

Applications

TERANODE

MONDECA

metatomix

MedTrust

Query Tool Interfaces

semantic web framework
Jena

openRDF.org
Sesame

Joseki

cYcorp

Standards

W3C World Wide Web consortium

RDF

NLP Entity Extractors

CALAIS
Powered by Thomson Reuters

Linguamatics

GATE
general architecture for text engineering

SI / Consulting

NORTHROP GRUMMAN

Raytheon

orbis
TECHNOLOGIES, INC.

BOEING

McDONALD BRADLEY

Cognia

accenture

Some Oracle Database Semantics Customers

Life Sciences

Lilly

Pfizer

SIB Swiss Institute of Bioinformatics

Defense/ Intelligence

National Geospatial-Intelligence Agency

Education

The University of Michigan 1817

Clinical Medicine & Research

The University of Texas Health Science Center at Houston

Cleveland Clinic

Telecomm & Networking

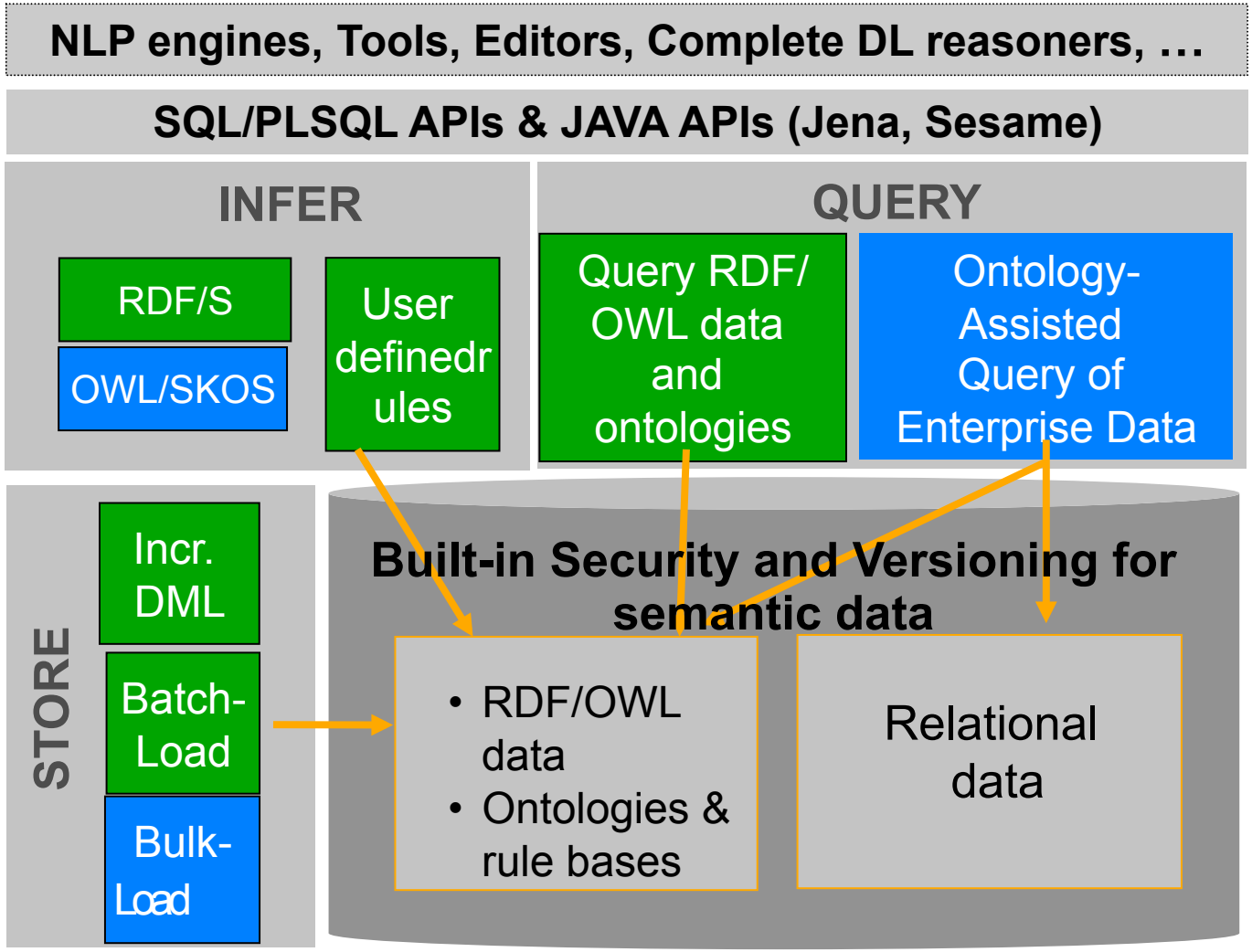
Hutchinson 3G Austria

Cisco

Publishing

Westlaw Thomson Reuters

Capabilities Overview of Release 11.2



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

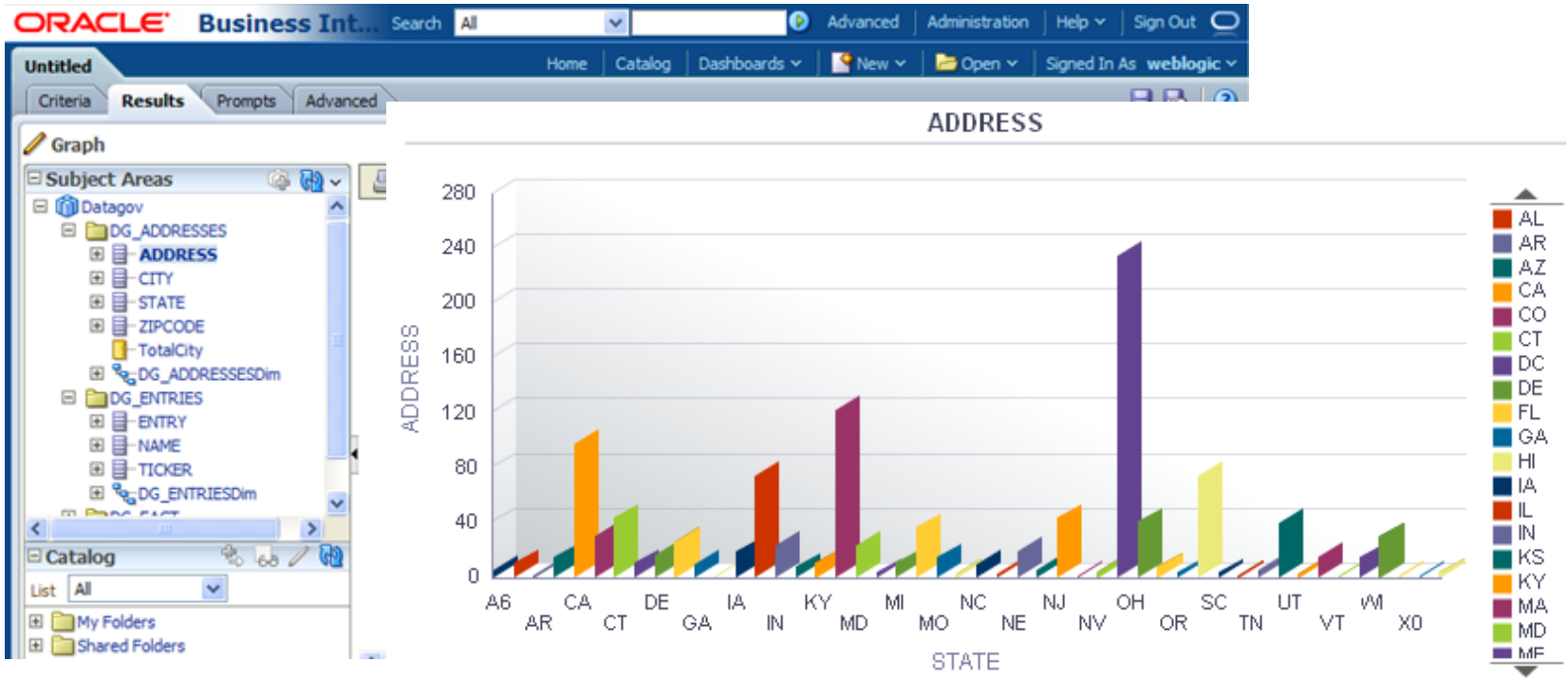
- Native inferencing in the database for
  - RDF, RDFS, and a rich subset of OWL semantics (OWLSIF, OWLPRIME, RDFS++)
  - User-defined rules
- Forward chaining.
  - New relationships/triples are inferred and stored ahead of query time
  - Removes on-the-fly reasoning and results in fast query times
-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

http://www.oracle.com/technology/tech/semantic_technologies/documentation/jenaadaptor2_readme.pdf

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>'));
- Collect statistics using exec sem_apis.analyze_model('family');
- Run inference
  - exec sem_apis.create_entailment('family_idx',sem_models('family'), sem_rulebases('owlprime'));
- Collect statistics using 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.

Important for performance!

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

	DocId	Article
r1	1	Indiana authorities filed felony charges and a court issued an arrest warrant for a financial manager who apparently tried to fake his death …
r2	2	Major dealers and investors in over-the-counter derivatives agreed to report all credit ..
	..	

Newsfeed *table*

RDF/XML for each document

NG	Subject	Property	Object
r1	p:Marcus	rdf:type	rc:Person
r1	p:Marcus	pred:hasName	“Marcus”^^xsd:string
r1	p:Marcus	pred:hasAge	“38”^^xsd:integer
..	..	..	..
r2	c:AcmeCorp	rdf:type	rc:Organization

Triples *table*

- Semantic query through SEM_CONTAINS

```
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

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

Some numbers are from Data Warehousing with 11g and RAC presentation

Setup for Performance (2)

- Database parameters[1]
  - SGA, PGA, filesystemio_options, db_cache_size, …
- Linux OS Kernel parameters
  - shmmax, 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!

1 http://www.oracle.com/technology/tech/semantic_technologies/pdf/semantic_infer_bestprac_wp.pdf

Bulk Loader Performance on Desktop PC: 11.2 Latest [1]

Ontology size	Time		Space (in GB)				
	bulk-load API[-2] Time (incl. Parse)	Sql*loader time [3]	RDF Model: Data Indexes	RDF Values: Data Indexes	Total: Data Index	App Table: Data [4]	Staging Table: Data [5]
LUBM50 6.9 million	2.6min	0.4min	0.15 0.48	0.13 0.17	0.28 0.65	0.16	0.32
LUBM1000 138.3 million	1hr 10min	8 min	3.07 9.74	2.55 3.49	5.62 13.23	3.14	6.36
LUBM8000 1,106 million	9hr 15min	1hr 5min	24.56 78.71	20.74 27.65	45.30 106.36	*22.10*	51.30

- **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**

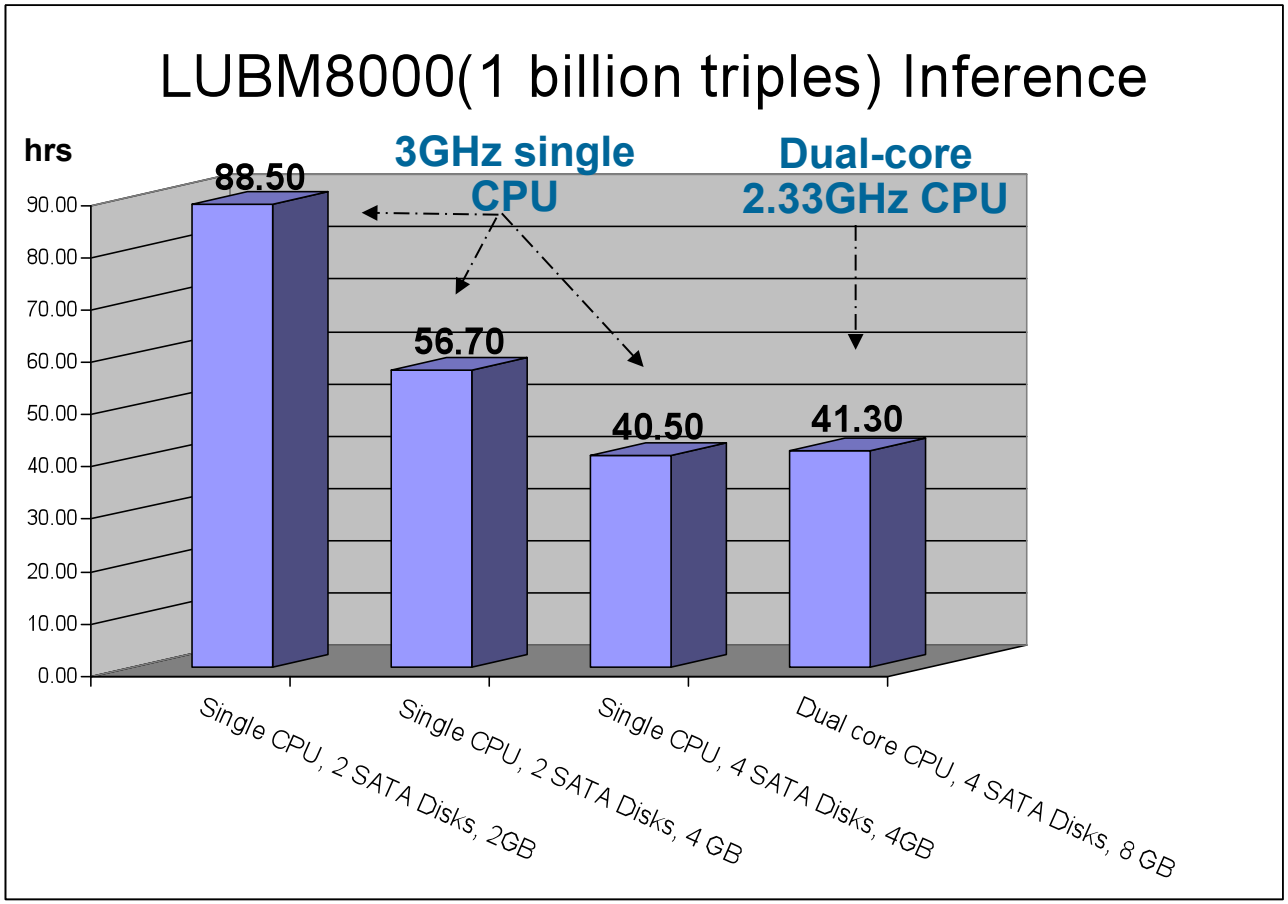
[1] This is an internal version of latest Oracle RDBMS 11.2
[2] Uses flags=>' parse parallel=4 parallel_create_index ' plus a new as-yet-unnamed option for value processing
[3] Uses parallel=true option and 8 to 10 gzipped N-Triple files as data files and a no-parse control file. [4] Application table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

Query Performance on Desktop PC

Ontology LUBM50 6.8 million & 5.4 million inferred		LUBM Benchmark Queries						
OWLPrime & new inference components	Query	Q1	Q2	Q3	Q4	Q5	Q6	Q7
	# answers	4	130	6	34	719	519842	67
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	0.05	0.75	0.20	0.5	0.22	1.86	1.71
	Query	Q8	Q9	Q10	Q11	Q12	Q13	Q14
	# answers	7790	13639	4	224	15	228	393730
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	1.07	1.65	0.01	0.02	0.03	0.01	1.47

- 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

11.1.0.7 Inference Performance on Desktop PC



- OWLPrime (11.1.0.7) inference performance scales really well with hardware. It is *not* a parallel inference engine though.

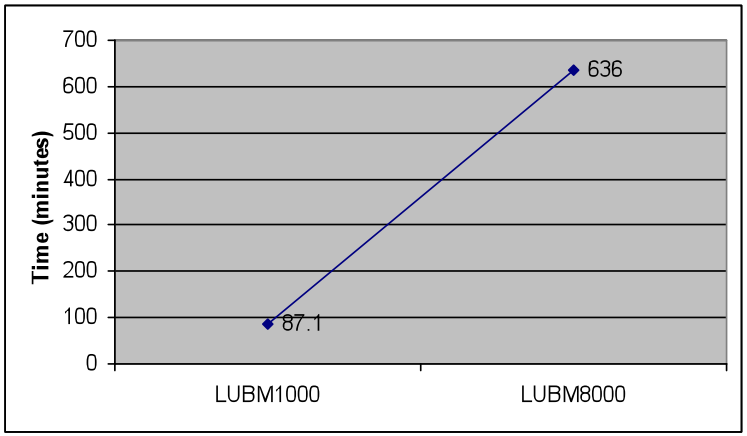
11.2.0.1 Inference Performance on Desktop PC

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

- 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)

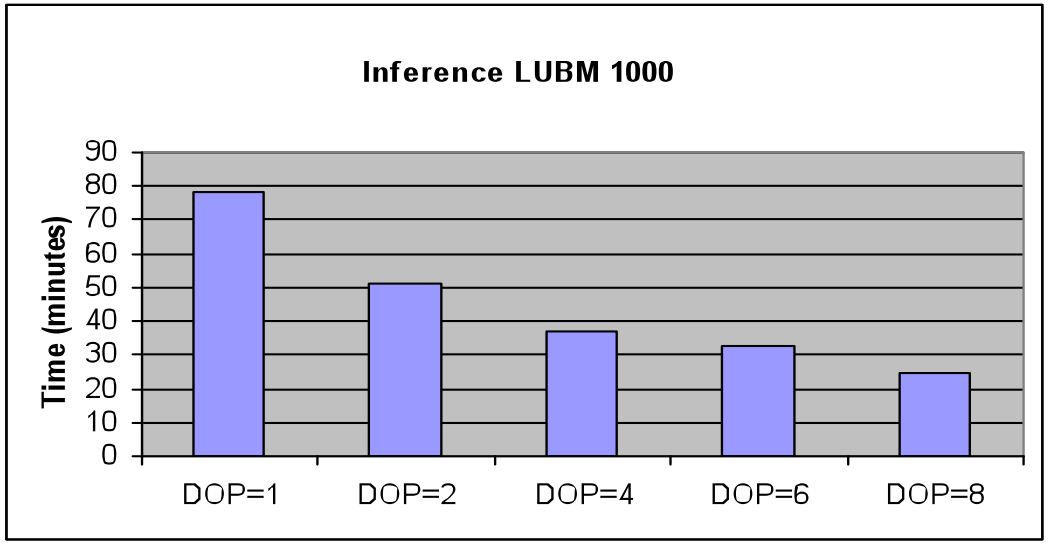


- **LUBM8000 (1B+)**
  - 25 minutes to load data into staging table
  - 10hr 36 minutes to load data from staging table (DOP=8)

- 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)

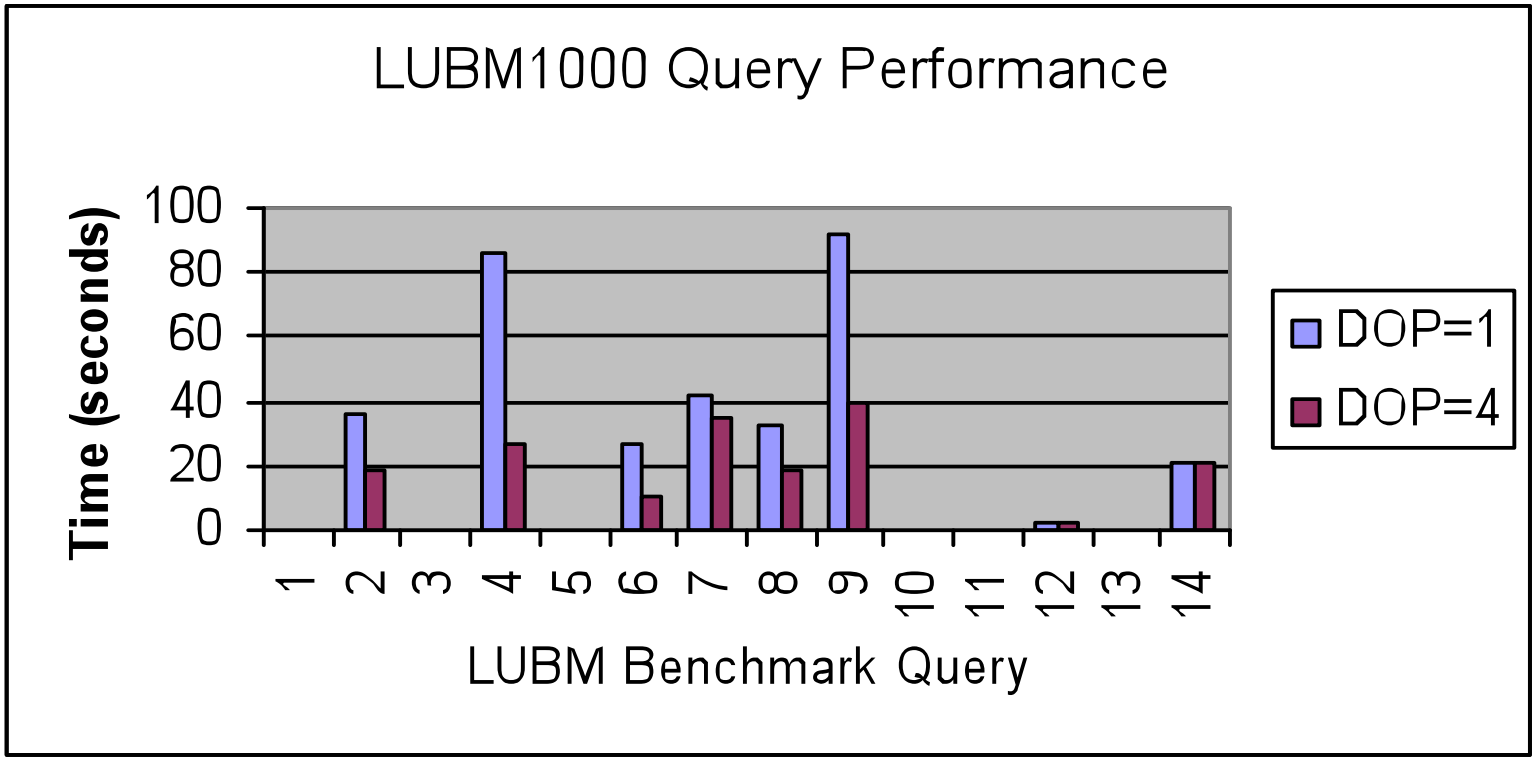


- **Inference performance for LUBM8000 (1B+)**
  - 226 minutes to infer 860M+ new triples (DOP=8)

- 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)**

http://www.oracle.com/technology/products/bi/db/exadata/pdf/exadata-technical-whitepaper.pdf

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

Ontology LUBM25K 3.3 billion & 2.7 billion inferred		LUBM Benchmark Queries						
OWLPrime & new inference components	Query	Q1	Q2	Q3	Q4	Q5	Q6	Q7
	# answers	4	2528	6	34	719	260M	67
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	0.01	20.65	0.01	0.01	0.02	23.07	4.99
	Query	Q8	Q9	Q10	Q11	Q12	Q13	Q14
	# answers	7790	6.8M	4	224	15	0.11M	197M
	Complete?	Y	Y	Y	Y	Y	Y	Y
	Time (sec)	0.48	203.06	0.01	0.02	0.02	2.40	19.45

- 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®


ORACLE IS THE INFORMATION COMPANY