



# **OWLIM: Pragmatic OWL Reasoning within Sesame**

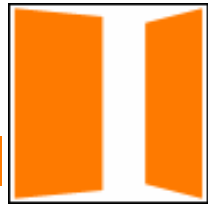
**Atanas Kiryakov**  
**Ontotext Lab.**



# Contents

openRDF.org

- Semantics & Language Support
- Implementation & Relations to Sesame
- Configuration
- ORDI: linking WSML to RDFS/OWL
- Evaluation
  - City Benchmark; LUBM; eLUBM
  - Performance Analysis
- BigOWLIM
  - Reasoning over 1 Billion of statements



# OWL Semantic Repository

openRDF.org

- OWLIM is a **scalable semantic repository** which allows
  - Management, integration, and analysis of heterogeneous data
  - combined with light-weight reasoning capabilities
- Its performance allows it to **replace RDBMS** in a wide range of applications
  - Suitable for analytical tasks and Business Intelligence (OLAP)
  - Not suitable for dynamic transaction-oriented environments (OLTP)
- OWLIM supports **full RDF(S) and limited OWL Lite**
  - It supports OWL Horst, which is more expressive than OWL DLP
  - Rule-extensions are also possible



# Contents

openRDF.org

- **Semantics & Language Support**
- Implementation & Relations to Sesame
- Configuration
- ORDI: linking WSML to RDFS/OWL
- Evaluation
  - City Benchmark; LUBM; eLUBM
  - Performance Analysis
- BigOWLIM
  - Reasoning over 1 Billion of statements



# OWL Horst

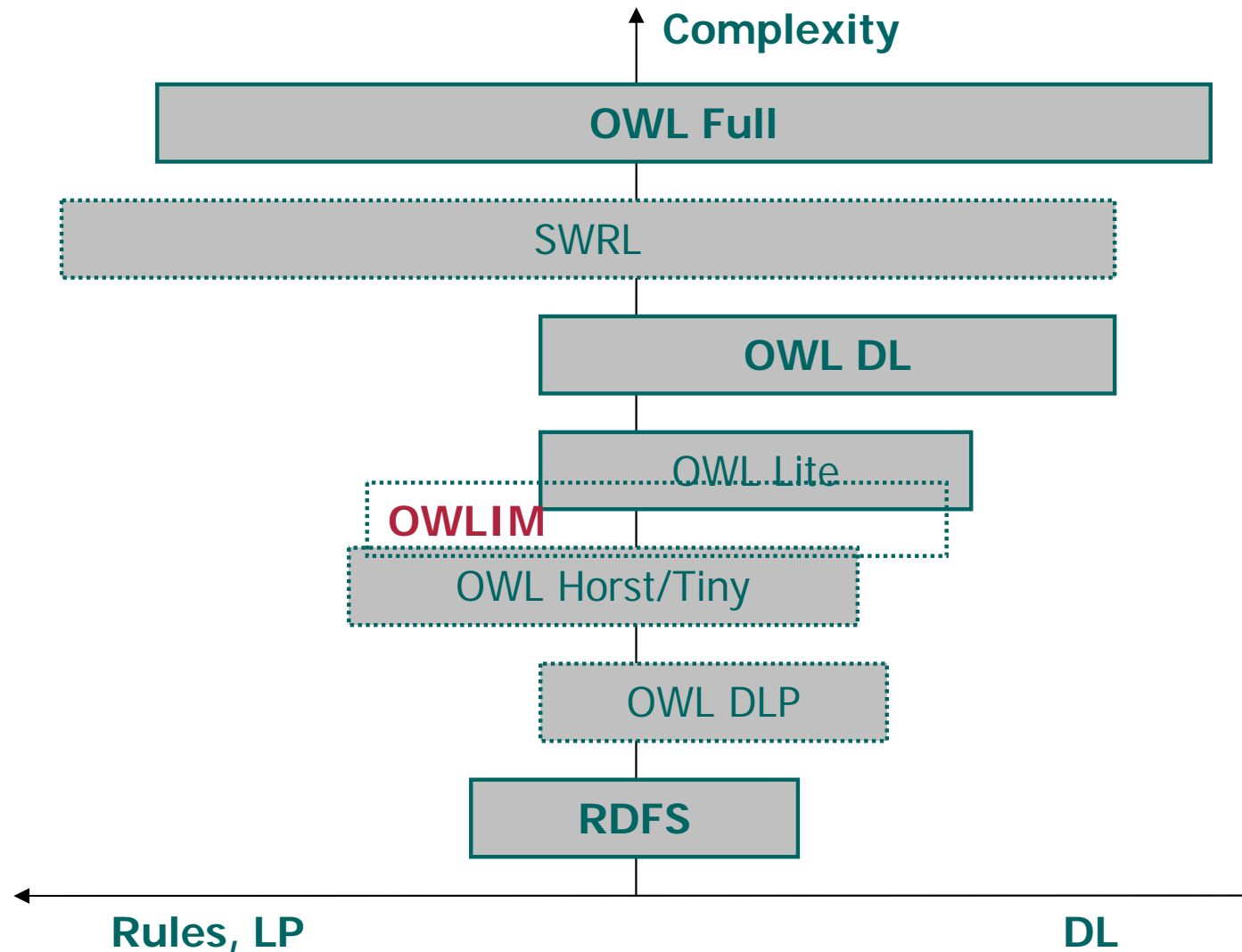
openRDF.org

- [Horst05] ter Horst, H. J. *Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity*. **ISWC'05**
- Defines **R-Entailment**: entailment over RDF graph based on rules of triple patterns
  - Rules are defined through generalized-triple patterns
    - Which may contain variable at any position, incl. predicate
  - ter Horst proves R-Entailment has better complexity\* than some Datalog-based OWL mappings and rule extensions
- Defines **pD\* entailment rules**, which:
  - Go beyond OWL DLP, dropping the DL-specific constraints
  - Are fully compatible with RDFS (including meta-modeling)
  - Extend the RDF(S) to assure better handling of the typed literals



# Naïve OWL Fragments Map

openRDF.org





# Semantics Supported by OWLIM

openRDF.org

- The reasoning support in **OWLIM** is **customizable**
- The **ruleset parameter** allows for switching between four predefined inference modes:
  - **owl-max** – the most expressive set (see the next slides);
  - **owl-horst** – a set similar to the one defined in [Horst05]:
    - It is sufficient to pass the LUBM benchmark correctly;
    - Similar to what was defined as OWL-Tiny at SWAD-Europe'03
  - **rdfs** – the standard RDF(S) semantics;
  - **empty** – as an RDF store without any inference;
- The **partRDFS parameter** allows switching on/off an optimization in the RDFS support



# OWL Support

openRDF.org

- OWLIM supports **almost all the OWL primitives**:
  - No support for: `Thing`, `Nothing`, `differentFrom`, `complementOf`
- The semantics supported is **close to OWL Lite**
  - Ignoring `(min/max)Cardinality` with values greater than 1
- Major **differences form OWL Lite**:
  - The semantics of **some primitives is only partially** supported
    - See the System Doc. and the **rules.pie file**
  - There are **no DL-specific constraints**; the following is OK:
    - Meta-classes (custom classes of classes);
    - Properties linking classes and instances need not be `owl:Annot...Pro`
- All the relevant normative **entailment tests** from “OWL Tests” pass correctly (available as JUnit tests)



# RDFS Support

openRDF.org

- The standard RDF(S) semantics, [Hayes 04], is supported
  - Except some D-entailment related to typed literals, which are omitted for performance reasons
- Optimized “partial” RDFS option is available for better performance. In this mode:
  - some “trivial” RDFS axioms are excluded;
  - `<X, rdf:type, rdf:Resource>` statements are not being inferred for all subject and predicates;
  - Rationale: most of the applications do not need such inference. With `partialRDFS=true`:
    - LUBM benchmark queries are evaluated properly
    - KIM (using the PROTON ontology) works properly
  - This optimization is irrelevant when `ruleset=empty`.



# Language Support Comparison

openRDF.org

- The owl-max semantics is **close to OWL Horst**:
  - OWL-max is generally richer than the  $pD^*$ -entailment of Horst
  - But OWLIM has no support for the inconsistency rules in  $pD^*$
  - No datatype-supporting modifications ( $D^*$ )
- OWLIM (owl-max) vs. OWL DLP
  - OWLIM supports a fragment **richer than OWL DLP**
  - OWLIM supports the **full RDFS semantics**, while DLP considers the DL-specific constraints
- OWLIM vs. OWL Lite:
  - OWLIM supports the full RDFS semantics, OWL Lite does not
  - Incomplete support for some primitives



# Contents

openRDF.org

- Semantics & Language Support
- **Implementation & Relations to Sesame**
- Configuration
- ORDI: linking WSML to RDFS/OWL
- Evaluation
  - City Benchmark; LUBM; eLUBM
  - Performance Analysis
- BigOWLIM
  - Reasoning over 1 Billion of statements

# In-memory Reasoning and Reliable Persistence

openRDF.org

- OWLIM uses **TRREE** (Triple Reasoning and Rule Entailment Engine)
  - TRREE implements **R-Entailment**
  - for **forward-chaining** and “total materialization”
    - The “inferred closure” is generated and maintained up to date
- It performs **in-memory reasoning** and query evaluation
- Combined with **reliable persistence** in RDF N-Triples
- The compromise: **relatively slow delete** operation
  - “limited” scalability on scenarios with high implicit/explicit st. ratio
  - not the case with most of the popular ontologies in OWL
- Very **fast upload, retrieval, query** evaluation for huge KB



# A Configurable SAIL for Sesame

openRDF.org

- **OWLIM is a Sesame SAIL**
  - Sesame = mature **RDF database** (v.1.2.1-1.2.4)
  - SAIL = Storage and Inference Layer
  - OWLIM benefits from Sesame's infrastructure, documentation,...
  - Support for multiple query languages, import and export formats
  - Version compatible with **Sesame 2.0** is almost ready
- **OWLIM Configuration Options:**
  - **noPersist**: the N-Triples persistency switched off
  - Configurable semantics: through the **ruleset** and **partialRDFS** parameters
  - Configurable **index size**: allows trading memory for performance
  - **stackSafe**: switches on a slower mode of the TRREE engine, which practically eliminates the possibility of stack overflow errors



# OWLIM Default Configuration

openRDF.org

- OWLIM's **default configuration** is:
  - noPersist=false (i.e. it does store the content of the repository)
  - Rule-set/semantics: owl-horst
  - partialRDFS=true
  - Index-Size: 4M entries (64MB of RAM; 16 bytes per entry)
  - stackSafe=false
- The configuration, equivalent to Sesame's most popular in-memory **RDFS** **SchemaSail**, is:
  - noPersist=true;
  - Rule-set/semantics: RDFS
  - partialRDFS=false



# Sample OWLIM Configuration

openRDF.org

```
<repository id="owlim">
  <title>OWLIM ...</title>
  <sailstack>
    <sail class="org.openrdf.sesame.sailimpl.OWLIMSchemaRepository">
      <param name="file" value="./kb/kb.nt"/>
      <param name="compressFile" value="no"/>
      <param name="dataFormat" value="ntriples" />
      <param name="new-triples-file" value="./kb/new-temp-triples.nt"/>
      <param name="ruleset" value="owl-horst" />
      <param name="partialRDFS" value="true" />
      <param name="indexSize" value="4000000" />
      <param name="stackSafe" value="false" />
      <!-- semicolon should be used as delimiter for both parameters -->
      <param name="imports" value="./ontology/owl.rdfs;..."/>
      <param name="defaultNS" value="http://www.w3.org/2002/07/owl#i..."/>
    </sail>
  </sailstack>
  <!--Access Control List can contain zero or more 'user' elements-->
  <acl worldReadable="false" worldWritable="false">
    <user login="admin" readAccess="true" writeAccess="true"/> </acl>
</repository>
```



# Contents

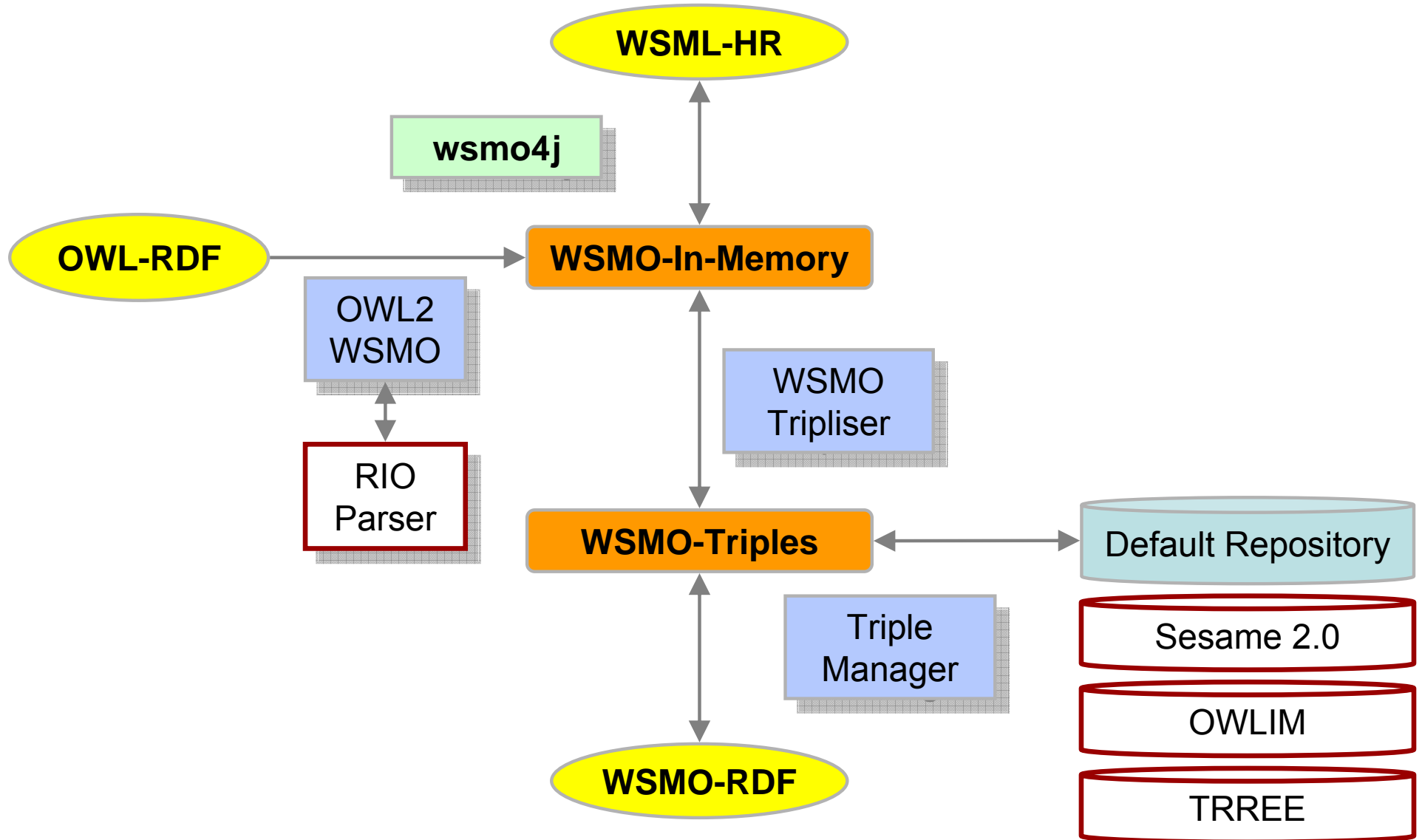
openRDF.org

- Semantics & Language Support
- Implementation & Relations to Sesame
- Configuration
- **ORDI: linking WSML to RDFS/OWL**
- Evaluation
  - City Benchmark; LUBM; eLUBM
  - Performance Analysis
- BigOWLIM
  - Reasoning over 1 Billion of statements



# OWLIM & Sesame serve WSML

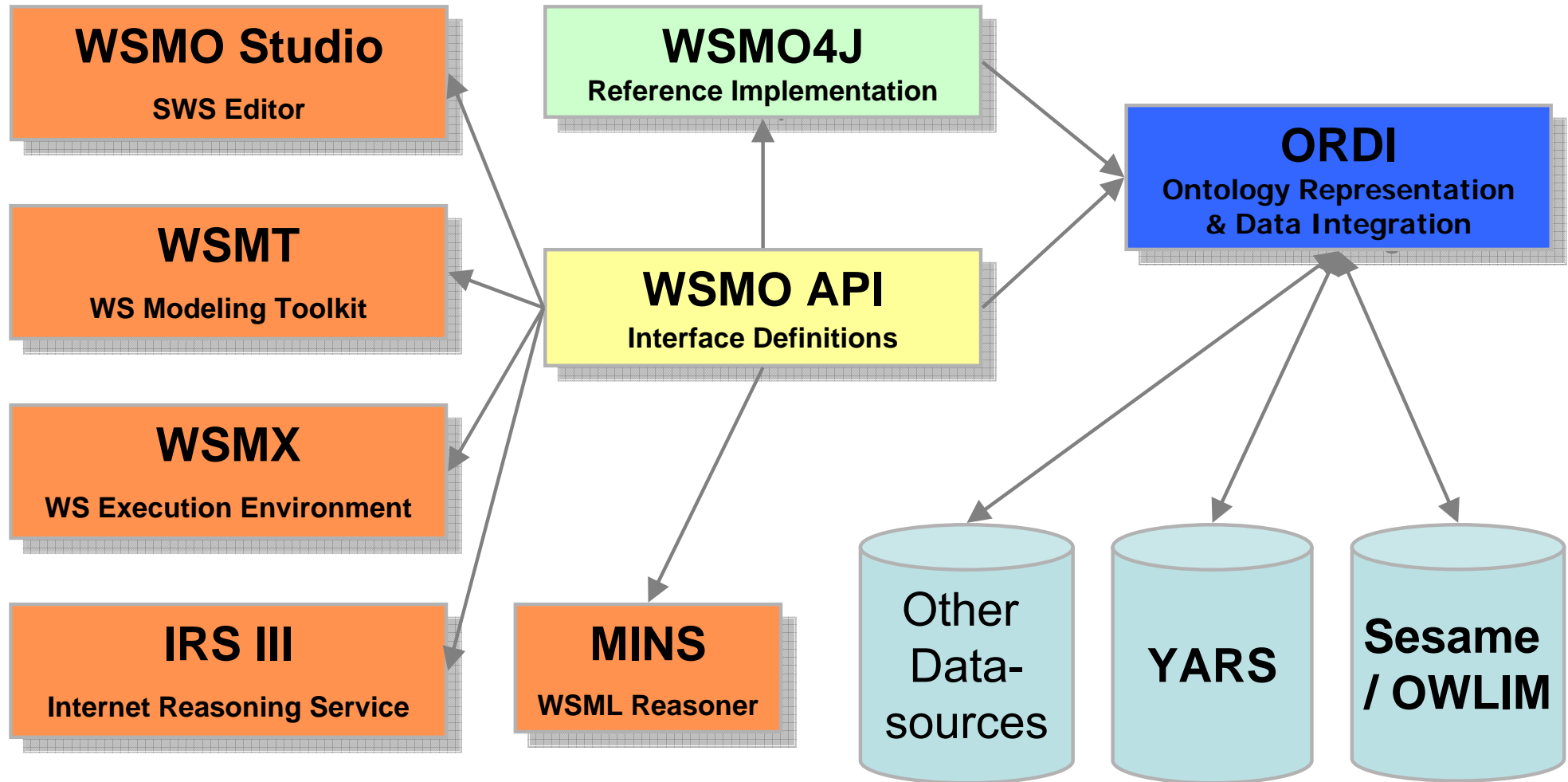
openRDF.org





# WSMO Infrastructure

openRDF.org

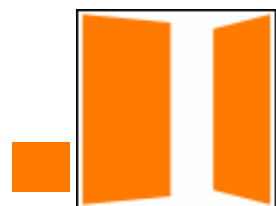




# Contents

openRDF.org

- Semantics & Language Support
- Implementation & Relations to Sesame
- Configuration
- ORDI: linking WSML to RDFS/OWL
- **Evaluation**
  - City Benchmark; LUBM; eLUBM
  - Performance Analysis
- BigOWLIM
  - Reasoning over 1 Billion of statements



# Hardware Configurations

openRDF.org

Name	Configuration	RAM (jvm -Xmx)	JDK	Comment
4cOpt12g	2 x Opteron 270 (2.0GHz, dual-core), Suse Linux v.10, 64-bit	12GB, DDR400	JDK 1.5 64-bit	A DB/application server; SATA2 drives; RAID10; ~4000 EURO
2Opt6.0g	2 x Opteron 246 (2.0GHz) Windows Server 2003 64-bit	6GB, DDR400	JDK 1.5 64-bit	A DB/application server; SATA2 drives; RAID10; ~3000 EURO
Pdc1.6g	Pentium D 920 (2.8GHz, dual-core), Win XP	1.6GB, DDR2 667	JDK 1.5	Workstation
Piv0.9g	Pentium IV 630 (3.0GHz), Win XP	900MB, DDR2 533	JDK 1.5	Office desktop
Pm0.7g	Pentium Mobile 1.6GHz, Win XP	680MB, DDR266	JDK 1.5	Notebook (Q2'03)



# City Benchmark

openRDF.org

- The repository is **pre-populated with about 0.5m expl. st.:**
  - a real ontology (PROTON) and
  - a knowledge base – the small version of KIM's World KB
- **Synthetic descriptions of cities are added incrementally**
  - each transaction adds one city described in about 10k st.
- **Interlinking to the non-synthetic part:**
  - the cities are linked to real provinces, randomly chosen from WKB
  - Ten synthetic organizations are created and “located” in each city;
  - 38 persons are created and settled in each of the organizations;
  - This way WKB is extended with realistic LDAP-like data
- **A couple of test queries** (in SeRQL) are evaluated after the addition of each 10 cities (i.e. after each new 100k st.)



# Delete Operation (City Bench.)

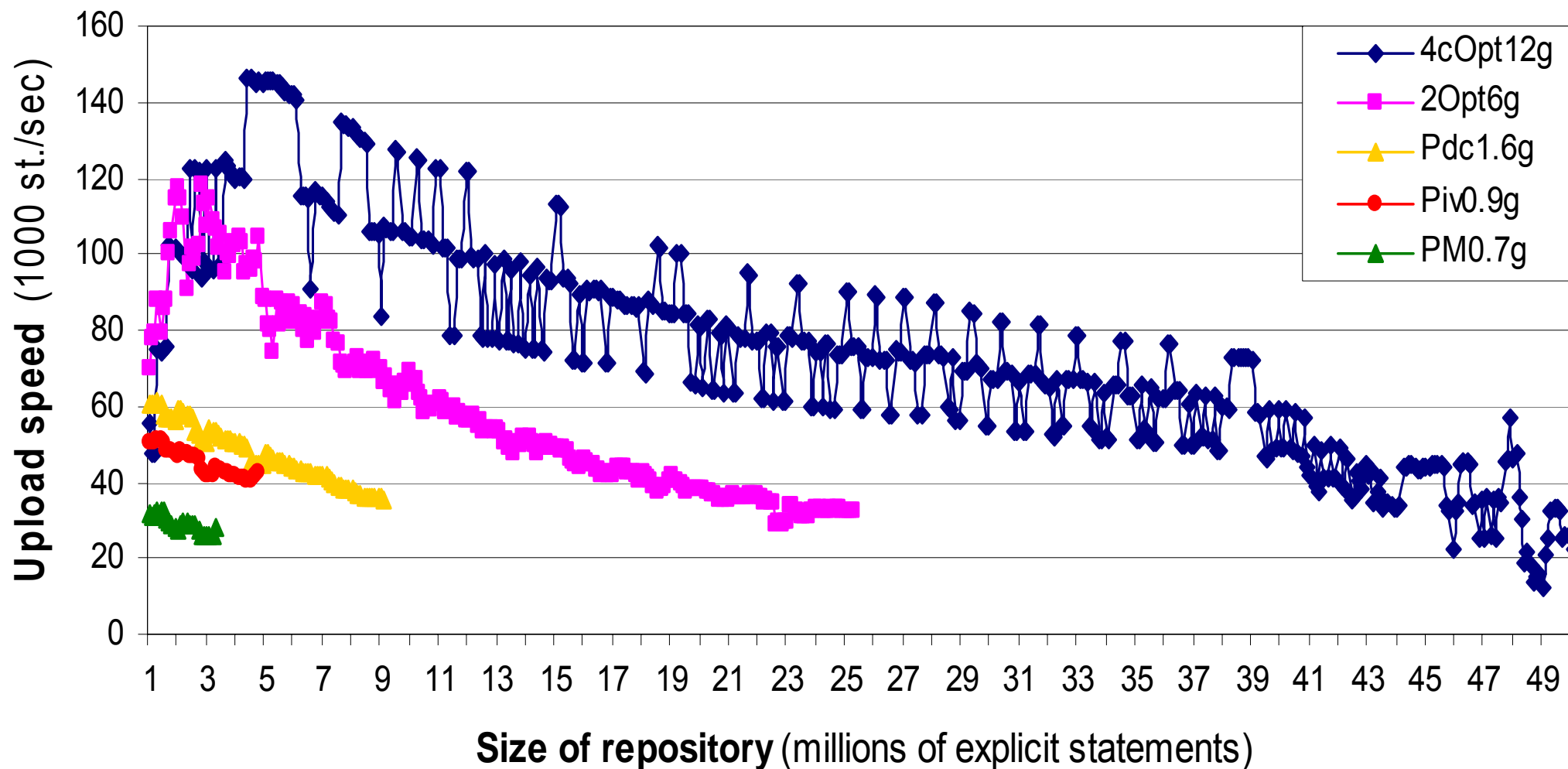
openRDF.org

- Delete is also tested on each 100 cities generated
- The time for finalization of a delete transaction on Piv0.9g varies with respect to the size of the repository as follows:
  - 19 sec. for 1.5M st.;
  - 31 sec. for 2.5M st.;
  - 43 sec. for 3.5M st.
- As it can be expected due to the straightforward invalidation of the inferred closure:
  - The delete operation is relatively slow;
  - delete time grows linearly with about 10 sec. for each new million of statements in the repository.



# Upload and Infer. (City Bench.)

openRDF.org





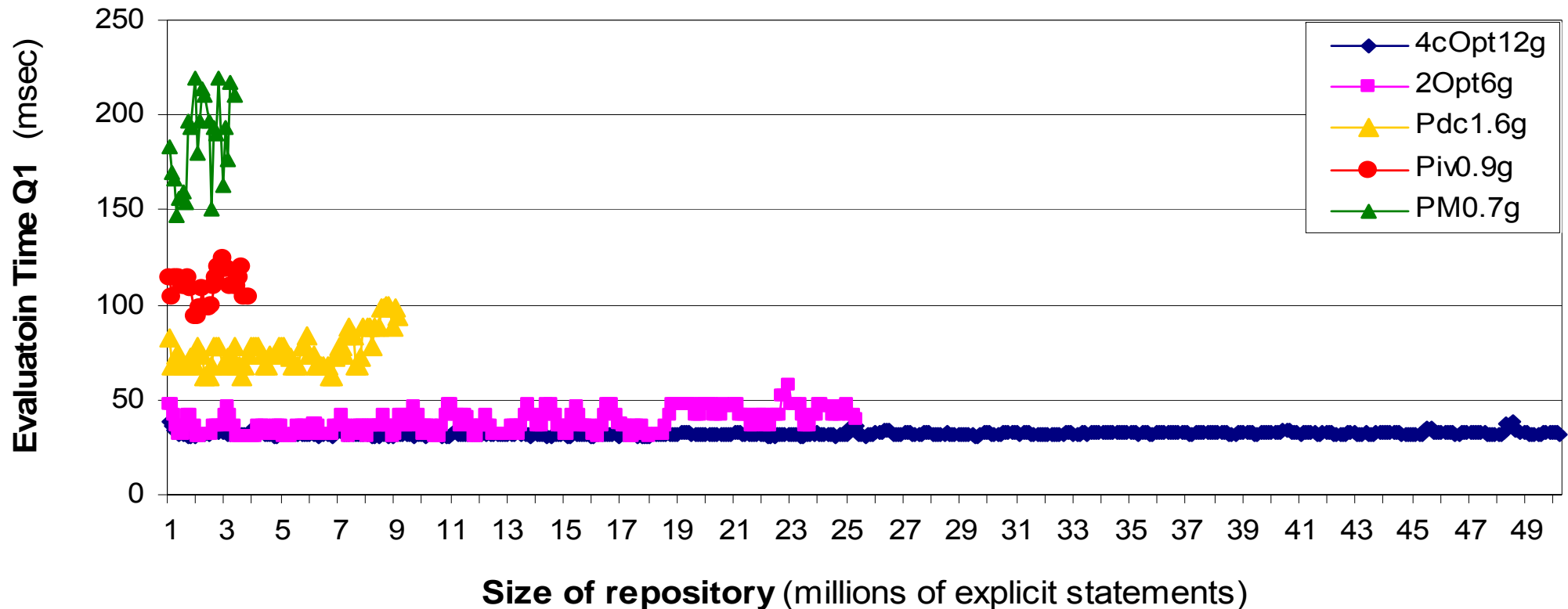
# Scale and Performance (City B.)

openRDF.org

- OWLIM can manage millions of stat. on desktop hardware:
  - Few millions of explicit statements (Mst.) can be handled on a notebook model Q2'2003 with 1GB of RAM;
- Given 12GB of RAM on a cheap server (4cOpt12g) it handles 50Mst:
  - As expected, the 64-bit Java VM requires a bit more memory
    - this explains why 2Opt6g scales only 2.5 times more than Pdc1.6g.
- Upload speed (incl. inference and storage) 20-150 Kst/sec
  - slows down in a “slow” linear dependency to the size of the repos.
  - 4cOpt12g runs **above 100 Kst/sec** for repositories <10 Mst;
  - it speeds down to **20 Kst/sec for repository with 50 Mst**

# Query Answering (City B. Q1)

openRDF.org

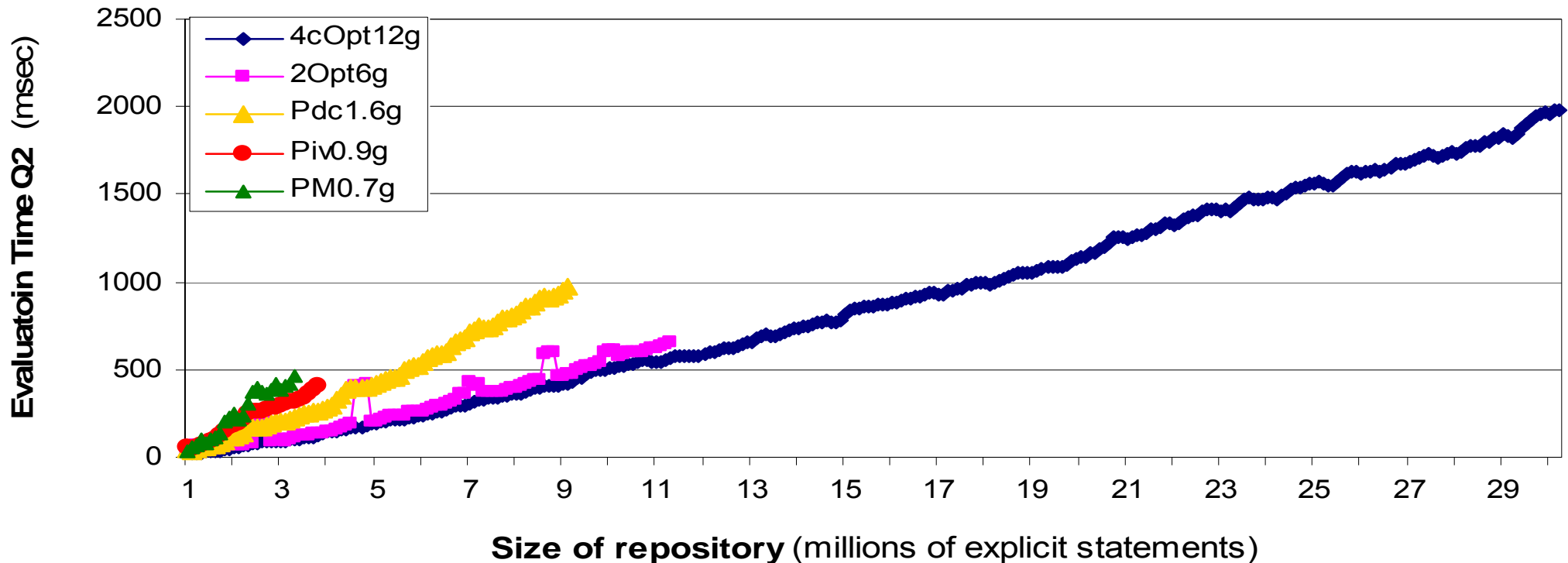


- Q1: Pattern of **11 statement-joins**
- **Fixed small result-set** – retrieval time close to 0
- The query **evaluation time is constant**



# Query Answering (City B. Q2)

openRDF.org



- Q2: Pattern of **12 statement-joins** and **LIKE “\*xyz\*”** literal constraint
- **Large result set** which grows linearly with the repository
- The query evaluation and retrieval time also grows linearly



# The LUBM Benchmark

openRDF.org

- The Lehigh University Benchmark (LUBM) is an outstanding OWL repository benchmark
  - <http://swat.cse.lehigh.edu/projects/lubm/>
- Synthetically generated datasets
  - On top of a fixed OWL ontology of “university organization”
  - The complexity is lower than **OWL Horst**; beyond OWL DLP
- There are 14 queries, checking inferences and query evaluation speed
- The biggest set available is LUBM(50,0) – 6.8 Mst.
  - About 600 MB. in about 1000 RDF/XML files



# OWLIM under LUBM Benchmark

openRDF.org

- **OWLIM loads LUBM(50,0) in 6 min. on a desktop** (machine Piv0.9g):
  - The only other system known to load it (and answer the queries afterwards) does so in 12 hours! [Guo05]
  - All the queries are answered correctly
- **OWLIM can load LUBM(300,0) on server** (machine 2Opt6g) in < 50 min.
  - About 40M statements, 3GB input files, 6GB in N-Triples persistency



# OWLIM Performance Analysis

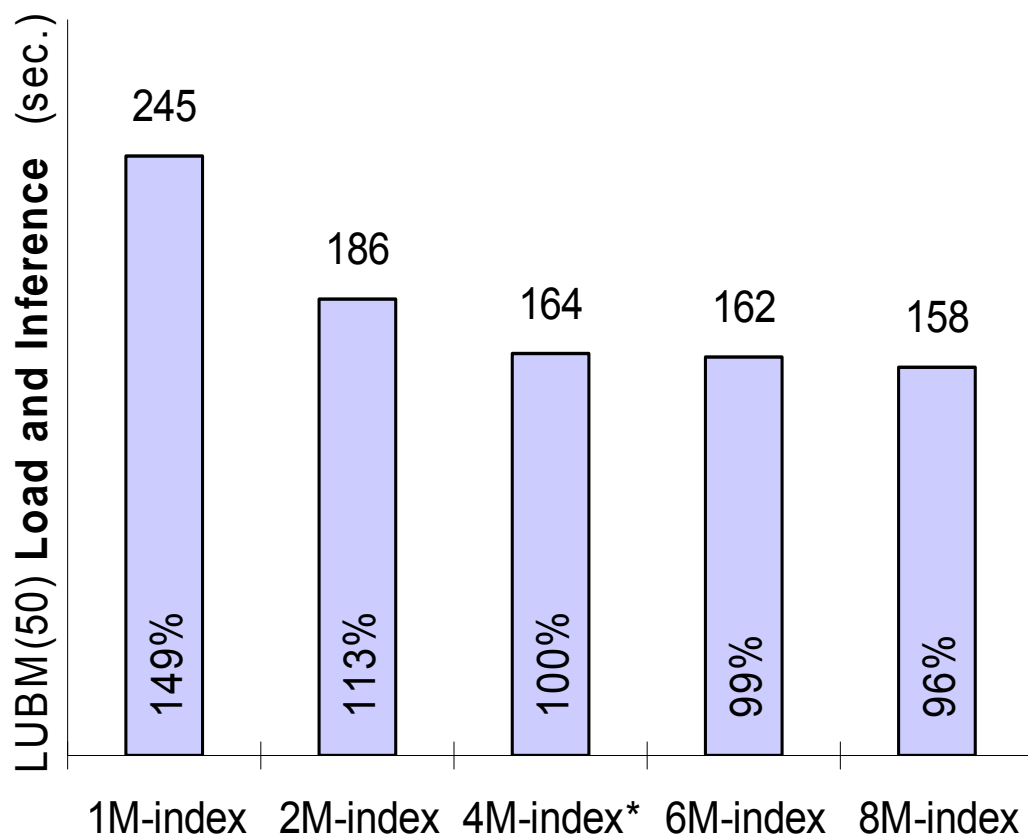
openRDF.org

- Several tests of different configurations and parameters
- Loading LUBM(50,0), which includes:
  - **Parsing** of the input RDF/XML files;
  - **Inference** – the inferred closure is calculated through forward-chaining and total materialization.
    - Irrelevant for ruleset=empty; then OWLIM acts as a plain RDF store
  - **Persistence** of all the data (unless switched off)
- The **basic configuration** (100% relative score):
  - Machine 4cOpt4g: 2xOpteron 270; 12GB of RAM;
  - 64-bit JDK 1.5.0;
  - 64-bit Suse Linux, ver. 10;
  - OWLIM with its default settings: noPersist=false; ruleset=owl-horst;partialRDFS=true; Index-Size: 4M entries; stackSafe=false



# Optimal Index Size Analysis

openRDF.org

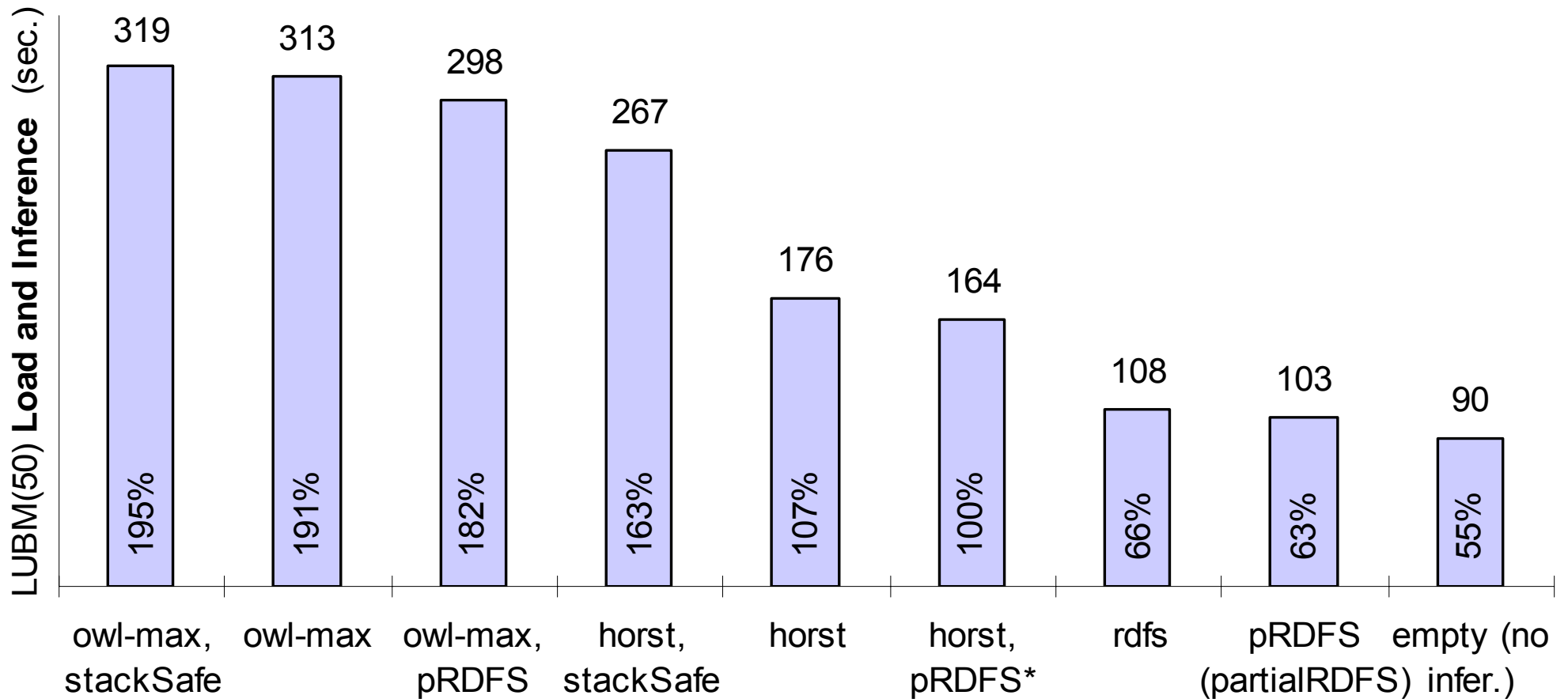


- As expected, larger index sizes lead to better performance.
- Critical for the performance on LUBM(50,0) is the border line between 1 and 2 millions of index entries.
- Index sizes larger than the default setting (4 million entries, 64MB of RAM) seem to deliver very little improvement.



# Rule-set and Inference Mode

openRDF.org





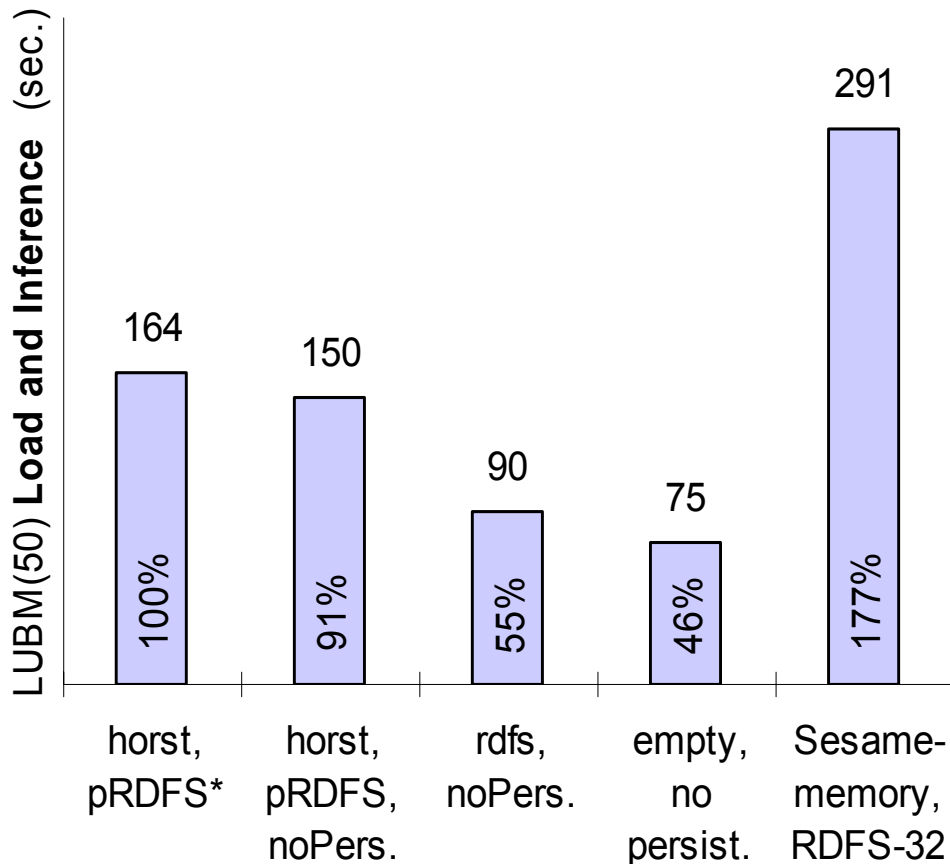
# Rule-set and Inference Mode (II)

openRDF.org

- The **partRDFS optimization provides 6-7% speedup**
- The **inference takes a bit less than half of the processing time**
  - with the default configuration: owl-horst with partRDFS
  - considering that the plain RDF version (empty) requires only 55% of the time to load the dataset;
- **owl-max is two times slower**
  - than the default setup owl-horst
  - In both cases partialRDFS switched on
- The **stack-safe mode slows down by 50%**
  - the version owl-horst with partialRDFS switched off
  - but has almost no impact on the owl-max inference

# The Impact of the Persistence

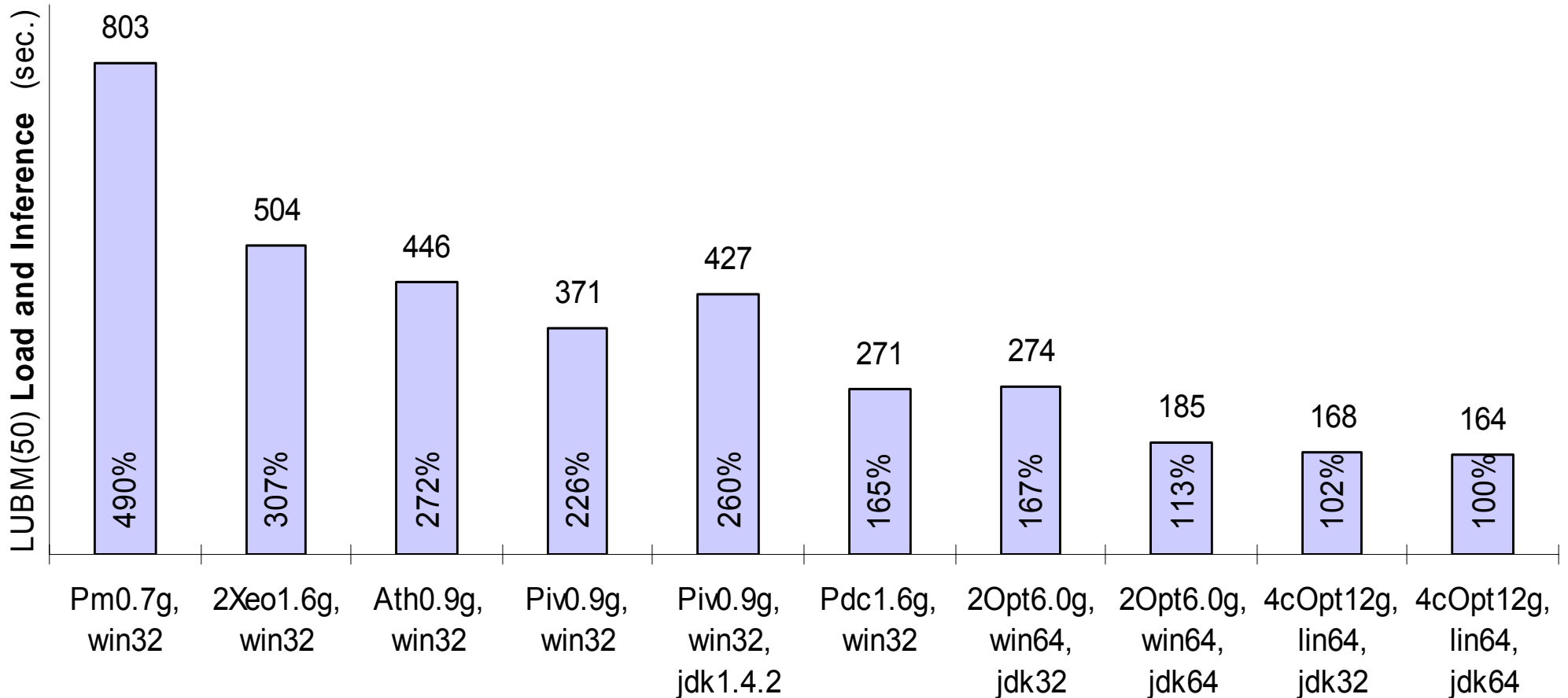
openRDF.org



- The **parsing** and building of in-mem repr. takes about **45%** of the load time (“empty, noPers.”)
- The **persistence takes 8-10%** on top of the time for loading (“horst, pRDFS, noPers.”);
  - Or flat 15 sec., 20% on top of the time for parsing;
- “Sesame-memory” is slower than the “rdfs, noPers.” setup of OWLIM.
  - It performs faster on 32-bit JDK 1.5 (291 s.); the time on 64-bit JDK was even higher (329 s.).

# Different Hardware, OS, JDK

openRDF.org



Refer to OWLIM's system documentation for analysis and comments.



# The eLUBM Benchmark

openRDF.org

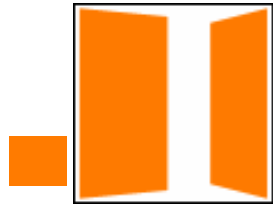
- eLUBM, [Ma et al, 2006] is a further development of the LUBM, <http://www.alphaworks.ibm.com/tech/semanticstk>
  - To be presented: Tuesday, June 13th, 14:00
- **eLUBM uses the LUBM** evaluation framework, but provides alternative ontology, KB and the queries which allow for:
  - More comprehensive coverage of the OWL Lite and DL;
  - Additional connections across the datasets for the universities;
- Two distinct datasets for **OWL Lite and OWL DL**, including collections for 1, 5 and 10 universities: Lite-1, Lite-5, Lite-10, DL-1, DL-5, and DL-10.
- **Two sets of queries** are provided – one for each of the datasets – covering various aspects of the semantics



# OWLIM under eLUBM

openRDF.org

- **[Ma et al, 2006]** reports evaluation of DLDB-OWL, OWLIM (a previous version) and MINERVA:
  - OWLIM is the fastest to load Lite-1, but fails on the other datasets
  - **The problem with OWLIM v.2.8.2 was stack overflow**
  - It was caused by large “equivalence” clusters formed by **hasSameHomeTownWith** relation, which is defined to be symmetric and transitive; “clusters” of fellow citizens
  - It is **solved after v.2.8.3** with the `stackSafe` parameter
- To pass eLUBM OWLIM has to be configured as follows:
  - **`stackSafe=true; ruleset=owl-max`**
- Given 512MB of RAM on a desktop machine (Piv0.9g) OWLIM shows the following results ...



# OWLIM under eLUBM (II)

openRDF.org

	Lite-1		Lite-5		DL-1		DL-5	
<b>Loading and inference</b>	24 sec.		123 sec.		45 sec.		192 sec.	
<b>Query Evaluation</b>	Time (ms)	Res. #	Time (ms)	Res. #	Time (ms)	Res. #	Time (ms)	Res. #
<b>Query 4</b>	593	397	3,099	397	598	414	3,199	414
<b>Query 11</b>	950	1,409	19,721	6,140	1,310	1,499	28,104	6,230
<b>Query 12</b>	118	69	545	69	126	37	578	37
<b>Query 14</b>					62	6,696	242	6,696

- Results for queries evaluated in less than 100 ms, for 1 university, and under 200 ms., for 5 universities, are not shown



# Contents

openRDF.org

- Semantics & Language Support
- Implementation & Relations to Sesame
- Configuration
- ORDI: linking WSML to RDFS/OWL
- Evaluation
  - City Benchmark; LUBM; eLUBM
  - Performance Analysis
- **BigOWLIM**
  - Reasoning over 1 Billion of statements



# BigOWLIM

openRDF.org

- Fully functional pre-release of BigOWLIM is available:
  - Check <http://www.ontotext.com/owlim/big/>
- BigOWLIM is an even more scalable not-in-memory version, using the corresponding ver. of TRREE engine
  - The “standard” OWLIM version, which uses in-memory reasoning and query evaluation is referred as SwiftOWLIM
- BigOWLIM does not need to maintain all the contents of the repository in the main memory in order to operate
- BigOWLIM stores the contents of the repository (including the “inferred closure”) in binary files; not in N-Triples
  - This allows instant startup and initialization of large repositories, because it does not need to parse, re-load and re-infer all the knowledge from scratch



# BigOWLIM vs. SwiftOWLIM

openRDF.org

- BigOWLIM uses **sorted indices**
  - While the indices of SwiftOWLIM are essentially hash-tables
  - In addition to this BigOWLIM maintains data statistics, to allow ...
- Database-like **query optimizations**
  - Re-ordering of the constraints in the query has no impact on the execution time
  - Combined with other optimizations, this feature delivers dramatic improvements to the evaluation time of “heavy” queries
- Special handling of **equivalence classes** (owl:sameAs)
  - Large equivalent classes does not cause excessive generation of inferred statements



# BigOWLIM's Memory Usage

openRDF.org

- The caching of BigOWLIM is highly configurable
- An XLS “calculator” is provided; follow sample configs

		CFG1, LUBM(50,0)		CFG2, LUBM(8000,0)	
Parameter	Factor	Value	Memory Size (mb)	Value	Memory Size (mb)
key.index.size	16	500,000	8	25,000,000	381
cache-size	29	500,000	14	60,000,000	1,659
entity-index-size	4	1,000,000	4	120,000,000	458
page-cache	16,000	1,000	15	10,000	153
<i>total cache</i>			<i>41</i>		<i>2,651</i>
entity dictionary	29	1,384,243	38	221,478,842	6,125
literals	29	461,414	13	73,826,281	2,042
<i>dict+literals</i>		<i>1,845,657</i>	<i>51</i>	<i>295,305,122</i>	<i>8,167</i>
<i>trree index ref</i>	<i>32</i>	<i>2,000,000</i>	<i>61</i>	<i>2,000,000</i>	<i>61</i>
Total (MB)			153		10,818



# BigOWLIM: 100 MSt on a Desktop

openRDF.org

- **LUBM(1000,0) on desktop machine with 2GB of RAM**
  - Hardware: Piv0.9g (Pentium 4, 3.0GHz, #630)
  - 32-bit JDK 1.5 given -Xmx1600
  - Loading, inference, and storage takes 11h 20 min.
  - LUBM(1000,0) contains above **130 Mst**
  - 10GB RDF/XML files; the pure parsing time is about 4h
- **LUBM(50,0) processed with only 192MB of RAM**
  - Piv0.9g; 32-bit JDK 1.5 given -Xmx1600
  - Loading, inference, and storage takes **26 min**
    - It is only 4 times slower than the in-memory version
  - Average upload speed around 4,000 st./sec.



# Reasoning over 1 Billion OWL St.

openRDF.org

- BigOWLIM successfully passed **LUBM(8000,0)**
  - Hardware: 4cOpt12g (2 x Opteron 270, 16GB of RAM, RAID 10)
  - OS: Suse 10.0 Linux, x86\_64, Kernel 2.6.13-15-smp
  - 64-bit JDK 1.5 given -Xmx12000
  - Loading, inference, and storage took **69 hours** and 51 min
  - LUBM(8000,0) contains **1.06 Billions** of explicit statements
    - The “inferred closure” contains about 786M statements
    - Managing over **1.85 Billions of statements in total**
  - 92GB RDF/XML files; 95 GB binary storage files
  - Average Speed: **4 538 statements/sec.**

# Reasoning over 1 Billion St. (II)

openRDF.org

Query no	Evaluation time (msec.)	Results Count
query1	1 829	4
query2	966 495	2 528
query3	15 688	6
query4	77 685	34
query5	20 663	719
query6	3 033 415	83 557 706
query7	65 401	67
query8	25 379	7 790
query9	4 736 695	2 178 420
query10	26 212	4
query11	1 186	224
query12	2 490	15
query13	321 145	37 118
query14	2 091 983	63 400 587



# Thank You!

openRDF.org

<http://www.ontotext.com/owlim>

Based on the limited evaluation results, publicly  
available:

**OWLIM is the fastest and most scalable  
OWL semantic repository in the world!**