
RDFARM - A SYSTEM FOR STORING LARGE SETS OF RDF TRIPLES AND QUADRUPLES BY MEANS OF NATURAL LANGUAGE ADDRESSING

Krassimira Ivanova

Abstract: *In this paper we present results from experiments for storing middle-size and large sets of RDF triples and quadruples by means of Natural Language Addressing. For experiments we have realized program RDFArM aimed to store RDF triples and quadruples in multi-layer hash tables (information spaces with variable size). The main features of program RDFArM are outlined in the paper. Analysis of the experimental results and rank-based multiple comparison are discussed.*

Keywords: *Natural Language Addressing; Rank-based Multiple Comparison*

ACM Classification Keywords: *H.2 Database Management; H.2.8 Database Applications*

Introduction

The idea of Natural Language Addressing (NLA) [Ivanova et al, 2012a; 2012b; Ivanova et al, 2013a; 2013b; 2013c; 2013d; 2013e; Ivanova, 2013; Ivanova, 2014] consists in using the computer encoding of name's (concept's) letters as logical address of connected to it information stored in a multi-dimensional numbered information spaces [Markov, 1984; Markov, 2004; Markov, 2004a]. This way no indexes are needed and high speed direct access to the text elements is available. It is similar to the natural order addressing in a dictionary where no explicit index is used but the concept by itself locates the definition.

In this paper we present results from experiments for storing middle-size and large sets of RDF triples and quadruples [Klyne & Carroll, 2004] through Natural Language Addressing. For experiments we have realized program RDFArM based on NLA Access Method and corresponded NLA Archive Manager called NL-ArM [Ivanova, 2014]. RDFArM is aimed to store RDF triples and quadruples in multi-layer hash tables (information spaces with variable size). Each RDF element can be stored by appropriate path, which is set by a natural language word or phrase.

Below we will present shortly main features of program RDFArM and after that we will present several experiments with middle-size and large data sets. Analysis of the experimental results and rank-based multiple comparison conclude the paper.

RDFArM

The data of RDFArM are encoded in N-Triples or N-Quads format. The N-Quads is a format that extends N-Triples with context. Each triple in an N-Quads document can have an optional context value [N-Quads, 2013]:

<subject> <predicate> <object> <context>.

as opposed to N-Triples, where each triple has the form:

<subject> <predicate> <object>.

The main idea for storing RDF-graphs in RDFArM follows the one of *multi-layer representation* [Ivanova et al, 2012b]. In other words, the RDF-relations are assumed as layers and the RDF-subjects are assumed as paths valid for all layers. The objects as well as contexts are stored in the containers located by the path in the corresponded layers.

Screenshots from the RDFArM program are shown on Figures 1a and 1b. The main functions are RDF-Write and RDF-Read for which there are corresponded buttons.

By "**RDF-Write**" button the function for storing RDF triples or quadruples from a file can be activated. The recognition of the case (triples or quadruples) is made automatically. The lines of triples do not contain the fourth element, i.e. the context of the quadruples.

Each triple (subject, relation, and object) or quadruple (subject, relation, object, and context) occupy one record in the input file. There is no limit to the number of records in the file. After pressing the "RDF-Write" button, the system reads records sequentially from the file and after storing the triples or quadruples, it displays two informative lines in the panel near to the "RDF-Write" button (Figure 1a):

- Total time used for storing all instances from the file;
- Average time used for storing of one instance (in milliseconds).

The time used is highly dependent on the possibilities of the operational environment and the speed of the computer hardware.




By "**RDF-Read**" button the function for reading RDF triples or quadruples from the RDFArM archives can be activated (Figure 1b). RDF-Read uses as input a file with requests similar to SPARQL requests [SPARQL, 2013] and extracts from the archives the requested information. The requested elements may be given by <?>. In other words, if any of parameters are not given, i.e. <subject>, <predicate>, <object>, or <context>, as in SPARQL requests, the rest are used as constant addresses and omitted parameters scan all non empty co-ordinates for given position. This way all possible requests like (?S-?P-?O), (S-P-?O), (S-?P-O), (?S-P-O), etc., are covered (S stands for subject, P for property, O for object).

In the panel next to the RDF-Read button, two informative lines are shown (Figure 1b) (in milliseconds):

- Total time used for extracting of all quadruple instances;
- Average time used for extracting of one instance.

The time used is highly dependent on possibilities of operational environment and speed of computer hardware.

The RDFArN form has three **service** buttons:

- The first () serves as a transition to the form for manual input and output of data to/from the system archive;
- The second () is connected to the module for adjusting the environment of the system – archives, input and output information, etc.;
- The third () activates the help text (user guide) of the system.

In the same panel there is a button which enables deleting the work archives of the RDFArM (for test control in this version, they are stored on the hard disk but not in the computer memory). RDFArM is completed with compressing program and after storing the information prepares small archive for long time storage.

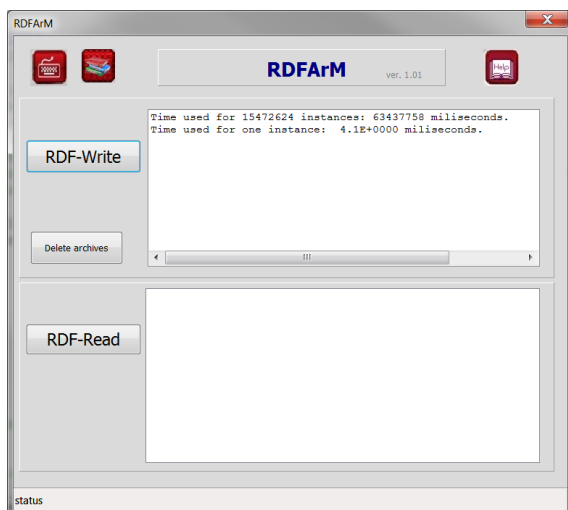


Figure 1a. Content of RDFArM RDF-Write panel with informative lines

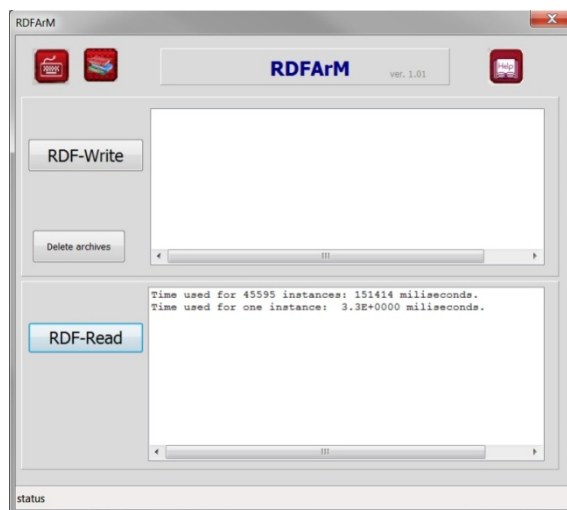


Figure 1b. Content of RDFArM RDF-Read panel with informative lines

Experiments with middle-size datasets

We have compared RDFArM with well-known RDF-stores:

- OpenLink Virtuoso Open-Source Edition 5.0.2 [Virtuoso, 2013];
- Jena SDB Beta 1 on PostgreSQL 8.2.5 and MySQL 5.0.45 [Jena, 2013];
- Sesame 2.0 [Sesame, 2012],

tested by Berlin SPARQL Bench Mark (BSBM) team and connected to it research groups [Becker, 2008; BSBMv2, 2008; BSBMv3, 2009].

We have provided experiments with *middle-size RDF-datasets*, based on selected real datasets from DBpedia [DBpedia, 2007a; DBpedia, 2007b] and artificial datasets created by BSBM Data Generator [BSBM DG, 2013; Bizer & Schultz, 2009].

The **real middle-size RDF-datasets**, we have used, consist of DBpedia's homepages and geocoordinates datasets with minor corrections [Becker, 2008]:

- *Homepages-fixed.nt* (200,036 triples; 24 MB) Based on DBpedia's homepages.nt dated 2007-08-30 [DBpedia, 2007a]. 3 URLs that included line breaks were manually corrected (fixed for DBpedia 3.0);
- *Geocoordinates-fixed.nt* (447,517 triples; 64 MB) Based on DBpedia's geocoordinates.nt dated 2007-08-30 [DBpedia, 2007b]. Decimal data type URI was corrected (DBpedia bug #1817019; resolved).

The RDF stores have different indexing behaviors: Sesame automatically indexes after each import, while SDB and Virtuoso allow for selective index activation which cause corresponded limitations or advantages. In order to make load times comparable, the data import by [Becker, 2008] had been performed as follows:

- *Homepages-fixed.nt* had been imported with indexes enabled;
- *Geocoordinates-fixed.nt* had been imported with indexes enabled.

In the case with RDFArM no parameters are needed. The data sets were loaded directly from the source files.

The **artificial middle-size RDF-datasets** are generated by BSBM Data Generator [BSBM DG, 2013] and

published in Turtle format [BSBMv1, 2008; BSBMv2, 2008; BSBMv3, 2009]. We converted it to N-triple format using “**rdf2rdf**” program developed by Enrico Minack [Minack, 2010].

We have used four BSBM datasets – 50K, 250K, 1M, and 5M. Details about these datasets are summarized in following Table 1.

Table 1. Details about used artificial middle-size RDF-datasets

Name of RDF-dataset:	50K	250K	1M	5M
Exact Total Number of Triples:	50,116	250,030	1,000,313	5,000,453
Number of Products	91	666	2,785	9,609
Number of Producers	2	14	60	199
Number of Product Features	580	2,860	4,745	3,307
Number of Product Types	13	55	151	73
Number of Vendors	2	8	34	196
Number of Offers	1,820	13,320	55,700	192,180
Number of Reviewers	116	339	1432	12,351
Number of Reviews	2,275	6,660	27,850	240,225
Total Number of Instances	4,899	23,922	92,757	458,140
File Size Turtle (unzipped)	14 MB	22 MB	86 MB	1,4 GB

In accordance with *multi-layer representation* [Ivanova et al, 2012b] the RDF-relations are assumed as layers and the RDF-subjects are assumed as paths valid for all layers. The objects are stored in the containers located by the path in the corresponded layers. Information about quantities of Subjects, Relations, and Objects in the used middle-size RDF-datasets are presented in Table 2.

Table 2. Quantities of Subjects, Relations, and Objects in used middle-size RDF-datasets

dataset	subjects (locations)	relations (layers)	objects
BSBM 50K	4900	40	50116
homepages-fixed.nt	200036	1	200036
BSBM 250K	60884	22	250030
geocoordinates-fixed.nt	152975	6	447517
BSBM 1M	92757	40	1000313
BSBM 5M	458142	55	5000453

To make experimental results comparable we have proposed special methodic and corresponded proportionality constants [Ivanova, 2014]. This way, all results from experiments are normalized to the next computer configuration:

- Processor: Intel Core2Quad Q9450@2.66GHz, CPU Launched:2008;
- Physical Memory: 8GB DDR2 667 (4 x 2GB);
- Hard Disks: 160GB (10,000 rpm) SATA2, 750GB (7,200 rpm) SATA2;
- Operating System: Ubuntu 8.04 64-bit, Kernel Linux 2.6.24-16-generic; Java Runtime: VM 1.6.0, HotSpot(TM) 64-Bit Server VM (build 10.0-b23); Separate partitions for application data (on 7,200 rpm HDD) and data bases (on 10,000 rpm HDD).

- **Loading of BSBM 50K** - The loading time' results from our experiment and from [Bizer & Schultz, 2008] are given in Table 3 and shown on Figure 2. Virtuoso has the best time. RDFArM has same loading time as Sesame and 40% better performance than Jena.

Table 3. Benchmark results for BSBM 50K

system	loading time in seconds
Sesame	3
Jena SDB	5
Virtuoso	2
RDFArM	3

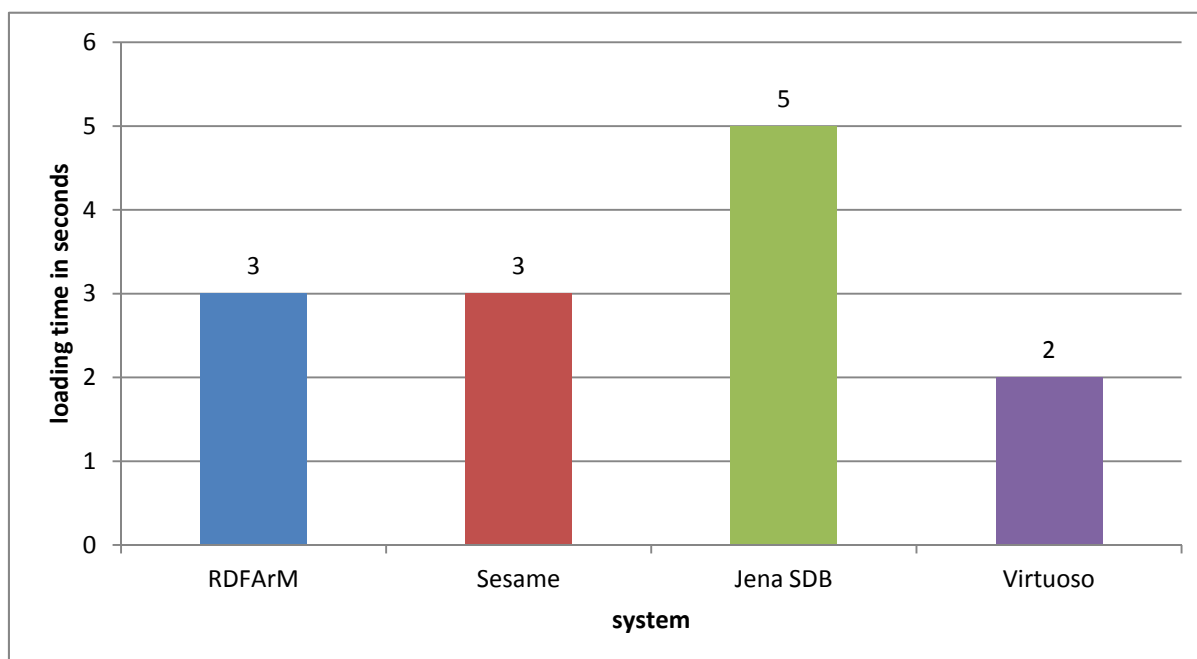


Figure 2. Benchmark results for BSBM 50K

- **Loading of homepages-fixed.nt** - The loading time' results from our experiment and from [Becker, 2008] are given in Table 4 and Figure 3. Virtuoso has the best time (about 42% better result than RDFArM). RDFArM has about 5% better time than Sesame and 36% better time than Jena (we take in account only the best result of compared system, in this case – Jena).

Table 4. Benchmark results for homepages-fixed.nt

system	loading time in seconds
Virtuoso (ogps, pogs, psog, sopg)	1327
Jena SDB MySQL Layout 2 Index	5245
Jena SDB Postgre SQL Layout 2 Index	3557
Jena SDB Postgre SQL Layout 2 Hash	9681
Sesame Native (spoc, posc)	2404
RDFArM	2272

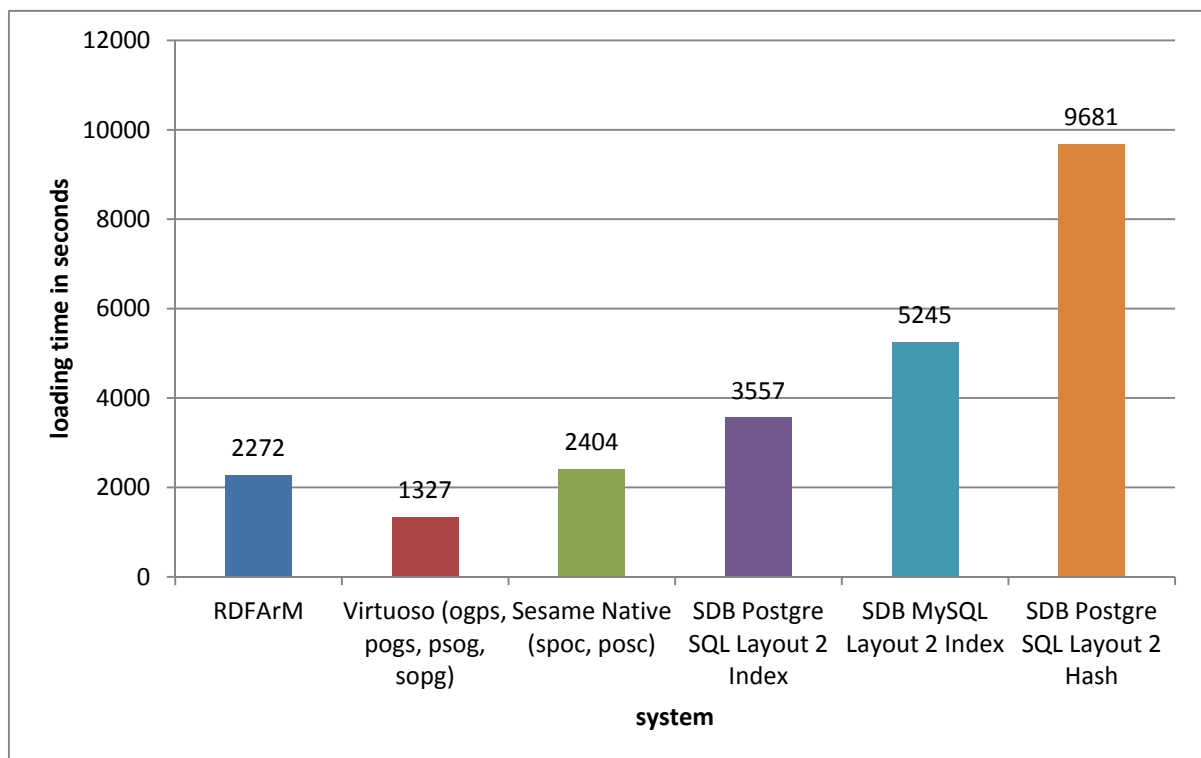


Figure 3. Benchmark results for homepages-fixed.nt

- **Loading of BSBM 250K** - The loading time' results from our experiment and from [BSBMv2, 2008] are given in Table 5 and shown on Figure 4. Virtuoso has 66% and Jena has 12% better performance than RDFArM. RDFArM has 22% better performance than Sesame.

Table 5. Benchmark results for BSBM 250K

system	loading time in seconds
Sesame	19
Jena TDB	13
Virtuoso TS	05
Virtuoso RDF views	09
Virtuoso SQL	09
RDFArM	14.79

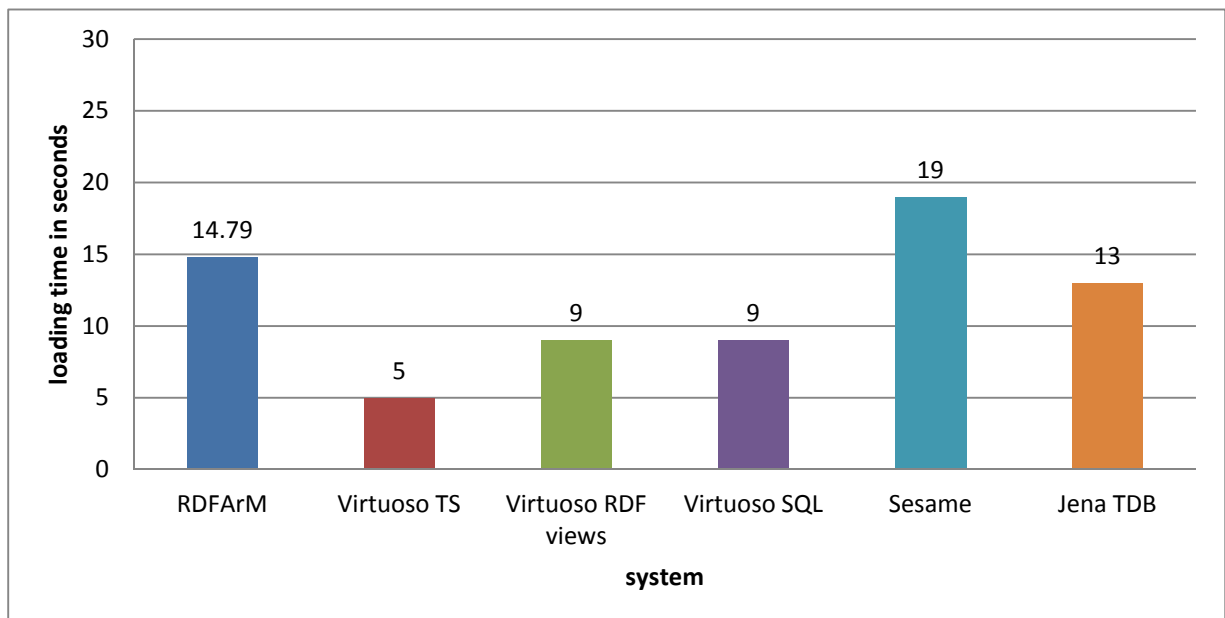


Figure 4. Benchmark results for BSBM 250K

- **Loading of geocoordinates-fixed.nt** – For this dataset we have six layers with 152975 NL-locations (containers) which contain 447517 objects, i.e. *some containers in some layers are empty*. The loading time results from our experiment and from [Becker, 2008] are given in Table 6 and Figure 5. RDFArM has the worst performance (we take the best time of Jena). Virtuoso has 64%, Sesame has 33%, and Jena has 5% better performance.

Table 6. Benchmark results for geocoordinates-fixed.nt

system	loading time in seconds
Virtuoso (ogps, pogs, psog, sopg)	1235
Jena SDB MySQL Layout 2 Index	6290
Jena SDB Postgre SQL Layout 2 Index	3305
Jena SDB Postgre SQL Layout 2 Hash	9640
Sesame Native (spoc, posc)	2341
RDFArM	3469

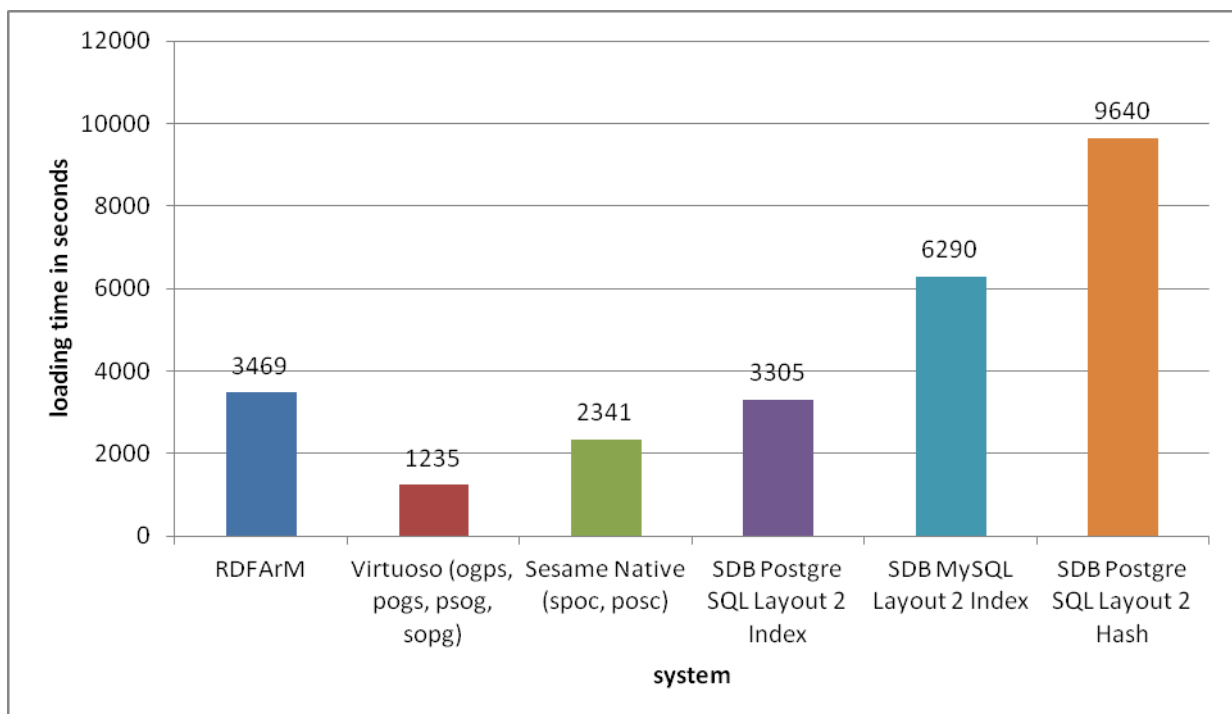


Figure 5. Benchmark results for geocoordinates-fixed.nt

- **Loading of BSBM 1M** - The loading time' results from our experiment and from [BSBMv2, 2008; BSBMv3, 2009] are given in Table 7 and shown on Figure 6. Virtuoso has 62% and Jena has 32% better performance than RDFArM. RDFArM has 67% better performance than Sesame.

Table 7. Benchmark results for BSBM 1M

system	loading time in min:sec	
	(a) [BSBMv2, 2008]	(b) [BSBMv3, 2009]
Sesame	02:59	03:33
Jena TDB	00:49	00:41
Jena SDB	02:09	-
Virtuoso TS	00:23	00:25
Virtuoso RV	00:34	00:33
Virtuoso SQL	00:34	00:33
RDFArM	01:00	01:00

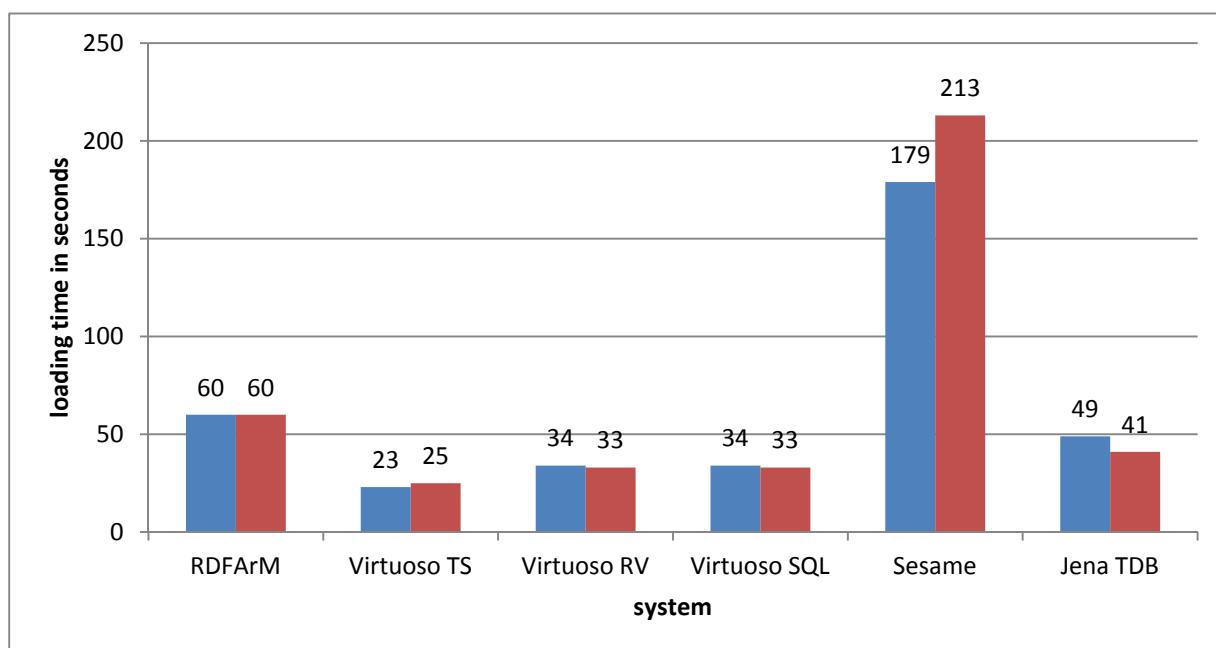


Figure 6. Benchmark results for BSBM 1M

- **Loading of BSBM 5M** - The loading time' results from our experiment and from [Bizer & Schultz, 2008] are given in Table 8 and shown on Figure 7. RDFArM has best loading time (about 85% better than Sesame, 71% than Jena, and 51% than Virtuoso).

Table 8. Benchmark results for BSBM 5M

system	loading time in seconds
Sesame	1988
Jena SDB	1053
Virtuoso	609
RDFArM	301

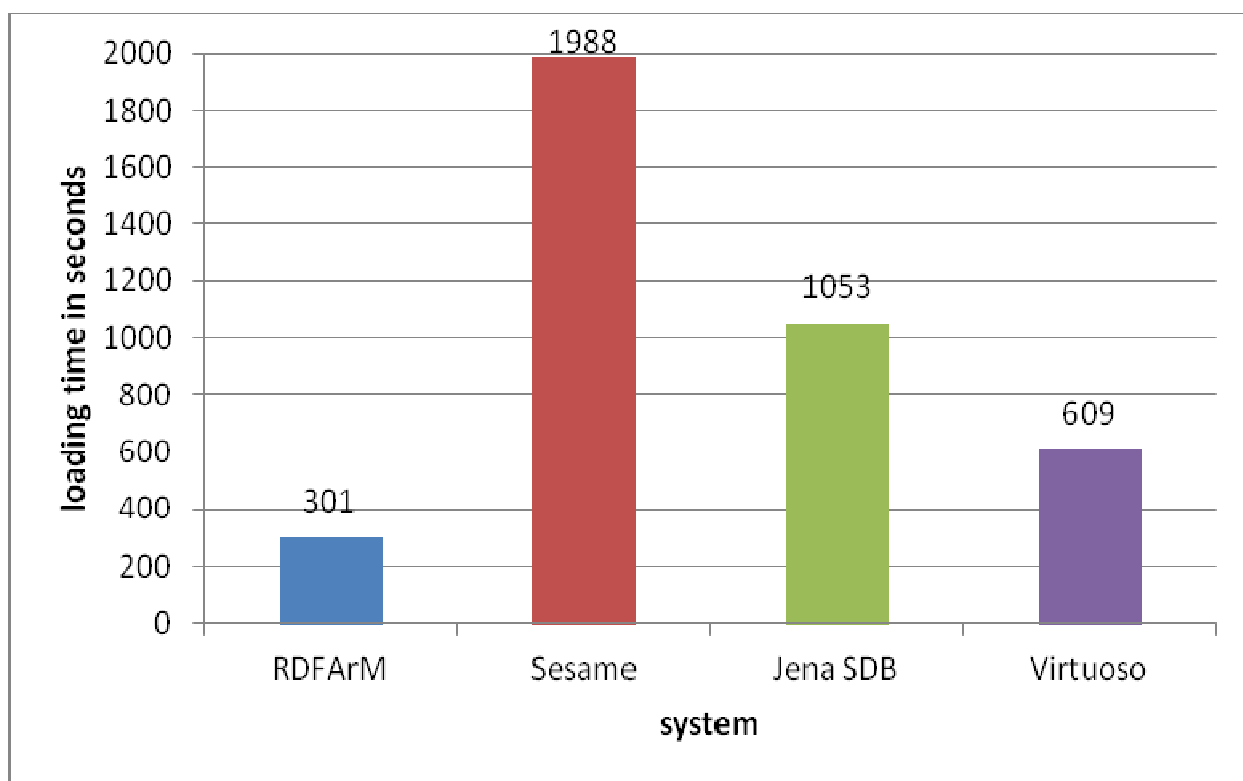


Figure 7. Benchmark results for BSBM 5M

Experiments with large datasets

We have provided experiments with **real large datasets** which were taken from DBpedia's homepages [DBpedia, 2007c] and Billion Triple Challenge (BTC) 2012 [BTC, 2012].

The real dataset is DBpedia's *infoboxes-fixed.nt* (15,472,624 triples; 2.1 GB) based on DBpedia's *infoboxes.nt* dated 2007-08-30 [DBpedia, 2007c]. 166 triples from the original set were excluded because they contained excessively large URIs (> 500 characters) that caused importing problems with Virtuoso (DBpedia bug #1871653). RDFArM has no such limitation. *infoboxes-fixed.nt* was imported with indexes initially disabled in SDB and Virtuoso. Indexes were then activated and the time required for index creation time was factored into the import time. In the case with RDFArM no parameters are needed. The datasets were loaded directly from the source file.

The RDF Stores, tested by [Becker, 2008], are:

- OpenLink Virtuoso Open-Source Edition 5.0.2 [Virtuoso, 2013];
- Jena SDB Beta 1 on PostgreSQL 8.2.5 and MySQL 5.0.45 [Jena, 2013];
- Sesame 2.0 beta 6 [Sesame, 2012].

The RDF stores feature different indexing behaviors: Sesame automatically indexes after each import, while SDB and Virtuoso allow for selective index activation.

Artificial large datasets were taken from Berlin SPARQL Bench Mark (BSBM) [Bizer & Schultz, 2009; BSBMv3, 2009; BSBMv5, 2009; BSBMv6, 2011]. Details about the benchmark artificial datasets are summarized in the following Table 9:

Table 9. Details about artificial large RDF-datasets

Number of Triples	25M	100M
Exact Total Number of Triples	25000244	100000112
Number of Products	70812	284826
Number of Producers	1422	5618
Number of Product Features	23833	47884
Number of Product Types	731	2011
Number of Vendors	722	2854
Number of Offers	1416240	5696520
Number of Reviewers	36249	146054
Number of Reviews	708120	2848260
Total Number of Instances	2258129	9034027
File Size Turtle (unzipped)	2.1 GB	8.5 GB

Information about quantities of Subjects, Relations, and Objects in the used large RDF-datasets are presented in Table 10.

Table 10. Number of Subjects, Relations, and Objects in used large RDF-datasets

dataset	subjects (locations)	relations (layers)	objects
infoboxes-fixed.nt	1354298	56338	15472624
BSBM 25M	2258132	112	25000244
BSBM 100M	9034046	341	100000112

- **Loading of infoboxes-fixed.nt** - For this dataset we have 56338 layers with 1354298 NL-locations (containers) which contain 15472624 objects, i.e. *some containers in some layers are empty*. The loading time' results from our experiment and from [Becker, 2008] are given in Table 11 and Figure 8. RDFArM has the worst loading time. Virtuoso is 95%, Sesame is 84%, and Jena is 48% better than RDFArM (we take in account only the best results of compared systems).

Table 11. Benchmark results for infoboxes-fixed.nt

system	loading time in seconds
Virtuoso (ogps, pogs, psog, sopg)	7017
Jena SDB MySQL Layout 2 Index	70851
Jena SDB Postgre SQL Layout 2 Index	73199
Jena SDB Postgre SQL Layout 2 Hash	734285
Sesame Native (spoc, posc)	21896
RDFArM	136412

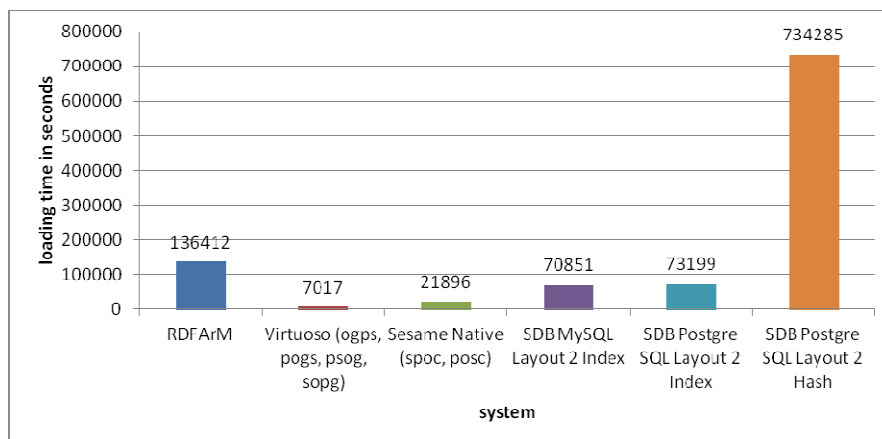


Figure 8. Benchmark results for infoboxes-fixed.nt

- **Loading of BSBM 25M** - The loading time' results from our experiment and from [Bizer & Schultz, 2009; BSBMv3, 2009] are given in Table 12 and Figure 9. Jena (with 30%) and Virtuoso (with 29%) are better than RDFArM. RDFArM has 97% better performance than Sesame.

Table 12. Benchmark results for BSBM 25M

system	loading time in seconds
Sesame	44225
Jena TDB	1013
Jena SDB	14678
Virtuoso TS	2364
Virtuoso RV	1035
Virtuoso SQL	1035
RDFArM	1453

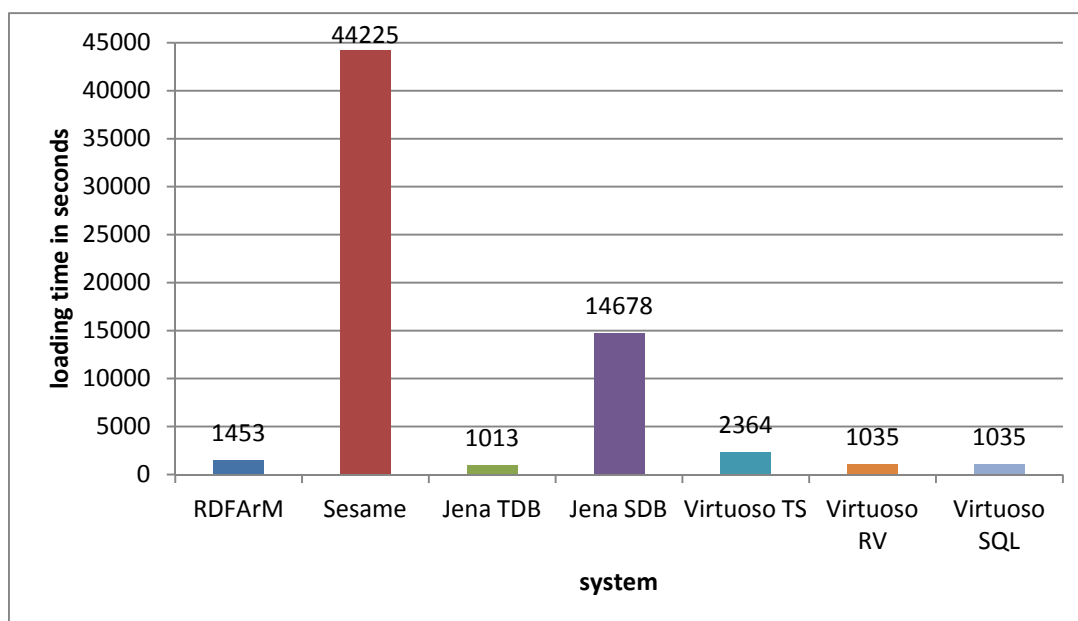


Figure 9. Benchmark results for BSBM 25M

- **Loading of BSBM 100M** - In this case we have 341 layers with 9034046 NL-locations (containers) which contain 100000112 objects, i.e. *some containers in some layers contain more than one object*. The loading time' results from our experiment and [Bizer & Schultz, 2009; BSBMv3, 2009] are given in Table 13 and Figure 10. Virtuoso is 35% better than RDFArM, and Jena is 4% better than RDFArM. RDFArM is 98% better than Sesame.

Table 13. Benchmark results for BSBM 100M

system	loading time in seconds
Sesame	282455
Jena TDB	5654
Jena SDB	139988
Virtuoso TS	28607
Virtuoso RV	3833
Virtuoso SQL	3833
RDFArM	5901

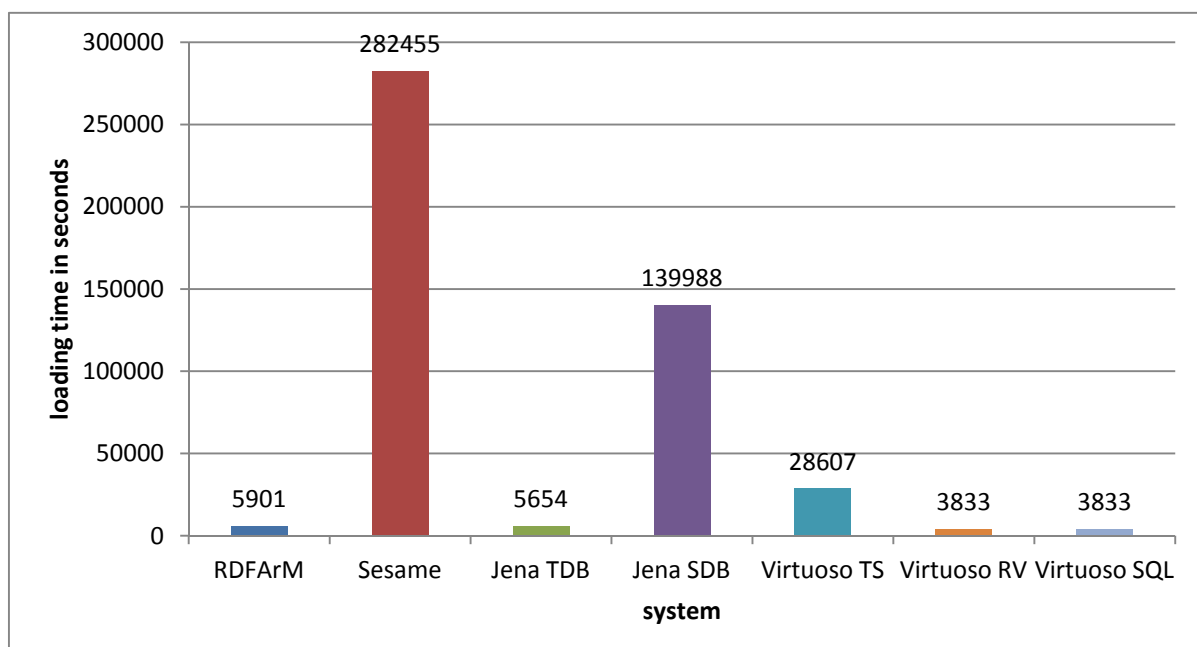


Figure 10. Benchmark results for BSBM 100M

Analysis of experiments with semi-structured datasets

In this work, the applicability of NL-addressing for middle-size and large semi-structured RDF-datasets was concerned. We have provided experiments based on selected datasets from DBpedia's homepages and Berlin SPARQL Bench Mark (BSBM) to make comparison with published benchmarks of known RDF triple stores.

We have used the Friedman test to detect statistically significant differences between the systems [Friedman, 1940]. The Friedman test is a non-parametric test, based on the ranking of the systems on each dataset. It is equivalent of the repeated-measures ANOVA [Fisher, 1973]. We have used Average Ranks ranking method, which is a simple ranking method, inspired by Friedman's statistic [Neave & Worthington, 1992]. For each dataset the systems are ordered according to the time measures and are assigned ranks accordingly. The best system receives rank 1, the second – 2, etc. If two or more systems have equal value, they receive equal rank which is mean of the virtual positions that had to receive such number of systems if they were ordered consecutively each by other.

Let n is the number of observed datasets; k is the number of systems.

Let i_j be the rank of system j on dataset i . The average rank for each system is calculated as

$$R_j = \frac{1}{n} \sum_{i=1}^k r_j^i.$$

The null-hypothesis states that if all the systems are equivalent than their ranks R_j should be equal. When null-hypothesis is rejected, we can proceed with the Nemenyi test [Nemenyi, 1963] which is used when all systems are compared to each other. The performance of two systems is significantly different if the corresponding average ranks differ by at least the critical difference

$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}}$$

where critical values q_{α} are based on the Studentized range statistic divided by $\sqrt{2}$. Some of the values of q_{α} are given in Table 14 [Demsar, 2006].

Table 14. Critical values for the two-tailed Nemenyi test

	number of systems								
	2	3	4	5	6	7	8	9	10
$q_{0.05}$	1.960	2.343	2.569	2.728	2.850	2.949	3.031	3.102	3.164
$q_{0.10}$	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920

The results of the Nemenyi test are shown by means of critical difference diagrams.

Benchmark values from our experiments and corresponded published experimental data from BSBM team are given in Table 15. Published results do not cover all table, i.e. we have no values for some cells. To solve this problem we have taken in account only the best result for given system on concrete datasets (Table 16). Sesame had no average values for tests 10a and 10b. Because of this we did not use these tests in our comparison. They were useful to see the need of further refinement of RDFArM for big data.

The ranks of the systems for the ten tests are presented below in Table 17.

Table 15. Benchmark values for middle size datasets

system	TEST											
	1	2	3	4	5a	5b	6	7	8	9	10a	10b
RDFArM	3	2272	14.79	3469	60	60	301	136412	1453	5901	15742	31484
Sesame Native (spoc, posc)	3	2404	19	2341	179	213	1988	21896	44225	282455		
Virtuoso (ogps, pogs, psog, sopg)	2	1327		1235			609	7017			6566	14378
Virtuoso TS			05		23	25			2364	28607		
Virtuoso RDF views			09									
Virtuoso SQL			09		34	33			1035	3833		
Virtuoso RV					34	33			1035	3833		
Jena SDB	5		13		129		1053		14678	139988		
Jena TDB					49	41			1013	5654	4488	9913
Jena SDB MySQL Layout 2 Index		5245		6290				70851				
Jena SDB Postgre SQL Layout 2 Hash		3557		3305				73199				
Jena SDB Postgre SQL Layout 2 Index		9681		9640				734285				

Table 16. Chosed benchmark values for middle size datasets

system	TEST											
	1	2	3	4	5a	5b	6	7	8	9	10a	10b
RDFArM	3	2272	14.79	3469	60	60	301	136412	1453	5901	15742	31484
Sesame	3	2404	19	2341	179	213	1988	21896	44225	282455		
Virtuoso	2	1327	05	1235	23	25	609	7017	1035	3833	6566	14378
Jena	5	3557	13	3305	49	41	1053	70851	1013	5654	4488	9913

Table 17. Ranking of tested systems

system	ranks for the tests										average rank
	1	2	3	4	5a	5b	6	7	8	9	
RDFArM	2.5	2	3	4	3	3	1	4	3	3	2.85
Sesame	2.5	3	4	2	4	4	4	2	4	4	3.35
Virtuoso	1	1	1	1	1	1	2	1	2	1	1.2
Jena	4	4	2	3	2	2	3	3	1	2	2.6

All average ranks are different. The null-hypothesis is rejected and we can proceed with the Nemenyi test. Following [Demsar, 2006], we may compute the critical difference by formula:

$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6N}}$$

where q_{α} we take as $q_{0.10} = 2.291$ (from Table 12 [Demsar, 2006; Table 5a]); k will be the number of systems compared, i.e. $k=4$; N will be the number of datasets used in benchmarks, i.e. $N=10$. This way we have:

$$CD_{0.10} = 2.291 * \sqrt{\frac{4 * 5}{6 * 10}} = 2.291 * \sqrt{\frac{20}{60}} = 2.291 * 0.577 = 1.322$$

We will use for critical difference $CD_{0.10}$ the value 1.322.

At the end, average ranks of the systems and distance to average rank of the first one are shown in Table 18.

Table 18. Average ranks of systems and distance to average rank of the first one

place	system	average rank	Distance between average rank of the system and average rank of the first one
1	Virtuoso	1.2	0
2	Jena	2.6	1.4
3	RDFArM	2.85	1.65
4	Sesame	3.35	2.15

The visualization of Nemenyi test results for tested systems is shown on Figure 11.

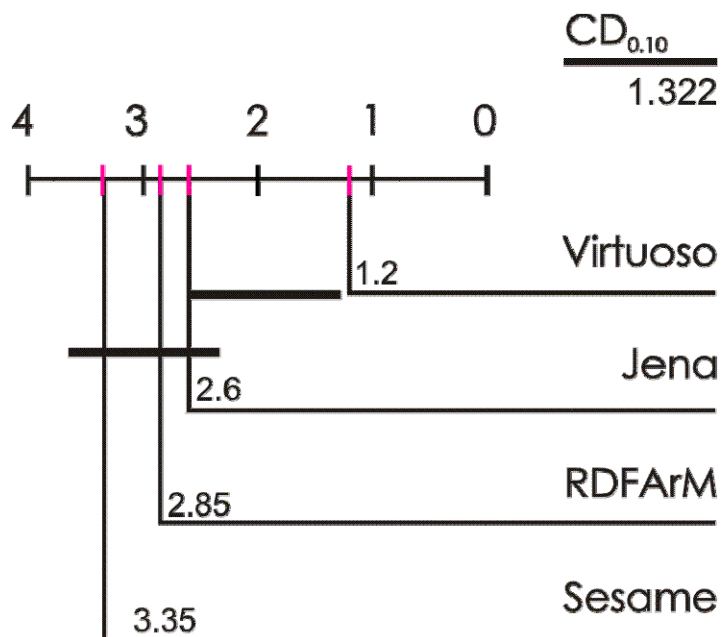


Figure 11. Visualization of Nemenyi test results

Analyzing these experiments we may conclude that RDFArM is at critical distances to Jena and Sesame. RDFArM is nearer to Jena than to Sesame. RDFArM, Jena, and Sesame are significantly different from Virtuoso.

Some recommendations to RDFArM may be given. RDF triple datasets has different characteristics depending of their origination. This causes the need to adapt NL-ArM storage engine to specifics of concrete datasets. For instance, important parameters are length of strings and quantity of repeating values of subject, relation, and object.

Conclusion

We have presented results from series of experiments which were needed to estimate the storing time of NL-addressing for middle-size and very large RDF-datasets. To make different configurations comparable, special proportionality constants for hardware and software were used.

Experiments were provided with both real and artificial datasets. Experimental results were systematized in corresponded tables. For easy reading visualization by histograms was given.

The goal of experiments for NL-storing of middle-size and large RDF-datasets were to estimate possible further development of RDFArM. What gain and loss using NL-Addressing for RDF storing?

The loss is additional memory for storing internal hash structures. But the same if no great losses we will have if we will build balanced search trees or other kind in external indexing. It is difficult to compare with other systems because such information practically is not published.

The benefit is in two main achievements:

- High speed for storing and accessing the information;
- The possibility to update and access the information immediately after storing *without recompilation* the database and rebuilding the indexes. This is very important because half or analyzed systems do not support updates.

The main conclusion is optimistic because RDFArM is at critical distances to Jena and Sesame, RDFArM is nearer to Jena than to Sesame, and, at the end, RDFArM, Jena, and Sesame are significantly different from Virtuoso.

Bibliography

- [Becker, 2008] Christian Becker, "RDF Store Benchmarks with Dbpedia", Freie Universität Berlin, 2008, <http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/> (accessed: 05.04.2013)
- [Bizer & Schultz, 2008] Christian Bizer, Andreas Schultz: Benchmarking the Performance of Storage Systems that expose SPARQL Endpoints; In: Proc. of the 4th International Workshop on Scalable Semantic Web knowledge Base Systems (SSWS2008), <http://www4.wiwiss.fu-berlin.de/bizer/pub/BizerSchulz-BerlinSPARQLBenchmark.pdf> (accessed: 31.07.2013)
- [Bizer & Schultz, 2009] Christian Bizer, Andreas Schultz, "The Berlin SPARQL Benchmark", In: International Journal on Semantic Web & Information Systems, Vol. 5, Issue 2, Pages 1-24, 2009, <http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/Bizer-Schultz-Berlin-SPARQL-Benchmark-IJSWIS.pdf>; see also <http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/> (accessed: 31.07.2013)
- [BSBM DG, 2013] Data Generator and Test Driver, In: Berlin SPARQL Benchmark (BSBM) - Benchmark Rules, <http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/spec/BenchmarkRules/index.html#datagenerator> (accessed: 31.07.2013)
- [BSBMv1, 2008] Berlin SPARQL Benchmark Results, V1, 2008, <http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/V1/results/index.html> (accessed: 31.07.2013)
- [BSBMv2, 2008] Berlin SPARQL Benchmark Results, V2 2008, <http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V2/index.html> (accessed: 31.07.2013)
- [BSBMv3, 2009] Berlin SPARQL Benchmark Results, V3, 2009, <http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V3/index.html> (accessed: 31.07.2013)

-
- [BSBMv5, 2009] BSBM Results (V5) for Virtuoso, Jena TDB, BigOWLIM, 2009, <http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V5/index.html> (accessed: 31.07.2013)
- [BSBMv6, 2011] Berlin SPARQL Benchmark Results, V6, 2011, <http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V6/index.html> (accessed: 31.07.2013)
- [BTC, 2012] Billion Triple Challenge 2012 Dataset <http://km.aifb.kit.edu/projects/btc-2012/> (accessed: 16.03.2013)
- [DBpedia, 2007a] DBpedia dataset "homepages.nt" dated 2007-08-30, <http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/homepages-fixed.nt.gz> (accessed: 31.07.2013)
- [DBpedia, 2007b] DBpedia dataset "geocoordinates.nt" dated 2007-08-30, <http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/geocoordinates-fixed.nt.gz> (accessed: 31.07.2013)
- [DBpedia, 2007c] DBpedia dataset "infoboxes.nt" dated 2007-08-30, <http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/infoboxes-fixed.nt.gz> (accessed: 31.07.2013)
- [Demsar, 2006] Demsar J., "Statistical comparisons of classifiers over multiple data sets" J. Mach. Learn. Res., 7, 2006, pp. 1-30
- [Fisher, 1973] R. A. Fisher, "Statistical methods and scientific inference (3rd edition)", Hafner Press, New York, 1973, ISBN 978-002-844740-7
- [Friedman, 1940] Friedman, M.: "A comparison of alternative tests of significance for the problem of m rankings", Annals of Mathematical Statistics, Vol. 11, 1940, pp.86-92.
- [Ivanova et al, 2012a] Krassimira Ivanova, Vitalii Velychko, Krassimir Markov. "About NL-addressing" (К вопросу о естественно-языковой адрессации) In: V. Velychko et al (ed.), Problems of Computer in Intellectualization. ITHEA@ 2012, Kiev, Ukraine - Sofia, Bulgaria, ISBN: 978-954-16-0061 0 (printed), ISBN: 978-954-16-0062-7 (online), pp. 77-83 (in Russian).
- [Ivanova et al, 2012b] Krassimira Ivanova, Vitalii Velychko, Krassimir Markov. "Storing RDF Graphs using NL-addressing", In: G. Setlak, M. Alexandrov, K. Markov (ed.), Artificial Intelligence Methods and Techniques for Business and Engineering Applications. ITHEA@ 2012, Rzeszow, Poland; Sofia, Bulgaria, ISBN: 978-954-16-0057-3 (printed), ISBN: 978-954-16-0058-0 (online), pp. 84 – 98.
- [Ivanova et al, 2013a] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "Introduction to the Natural Language Addressing", International Journal "Information Technologies & Knowledge" Vol.7, Number 2, 2013, ISSN 1313-0455 (printed), 1313-048X (online), pp. 139–146.
- [Ivanova et al, 2013b] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "Introduction to Storing Graphs by NL-Addressing", International Journal "Information Theories and Applications", Vol. 20, Number 3, 2013, ISSN 1310-0513 (printed), 1313-0463 (online), pp. 263 – 284.
- [Ivanova et al, 2013c] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "Storing Dictionaries and Thesauruses Using NL-Addressing", International Journal "Information Models and Analyses" Vol.2, Number 3, 2013, ISSN 1314-6416 (printed), 1314-6432(online), pp. 239 - 251.
- [Ivanova et al, 2013d] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "The Natural Language Addressing Approach", International Scientific Conference "Modern Informatics: Problems, Achievements, and Prospects of Development", devoted to the 90th anniversary of academician V. M. Glushkov. Kiev, Ukraine, 2013, ISBN 978-966-02-6928-6, pp. 214 - 215.
- [Ivanova et al, 2013e] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "Storing Ontologies by NL-Addressing", IVth All-Russian Conference "Knowledge-Ontology-Theory" (KONT-13), Novosibirsk, Russia, 2013, ISSN 0568 661X, pp. 175 - 184.

- [Ivanova, 2013] Krassimira Ivanova, "Informational and Information models", In Proceedings of 3rd International conference "Knowledge Management and Competitive Intelligence" in the frame of 17th International Forum of Young Scientists "Radio Electronics and Youth in the XXI Century", Kharkov National University of Radio Electronics (KNURE), Kharkov, Ukraine, Vol.9, 2013, pp 6-7.
- [Ivanova, 2014] Krasimira Ivanova, "Storing Data using Natural Language Addressing", PhD Thesis, Hasselt University, Belgium, 2014
- [Jena, 2013] Apache Jena, http://jena.apache.org/about_jena/about.html (accessed: 23.03.2013)
- [Klyne & Carroll, 2004] Graham Klyne and Jeremy J. Carroll, Editors, Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>. Latest version available at <http://www.w3.org/TR/rdf-concepts/> (accessed: 21.02.2013).
- [Markov, 1984] Krassimir Markov, "A Multi-domain Access Method", Proceedings of the International Conference on Computer Based Scientific Research, PLOVDIV, 1984, pp. 558 - 563.
- [Markov, 2004] Krassimir Markov, "Multi-domain information model", Int. J. Information Theories and Applications, 11/4, 2004, pp. 303 - 308
- [Markov, 2004a] Krassimir Markov, "Co-ordinate based physical organization for computer representation of information spaces", (Координатно базирана физическа организация за компютърно представяне на информационни пространства) Proceedings of the Second International Conference "Information Research, Applications and Education" i.TECH 2004, Varna, Bulgaria, Sofia, FOI-COMMERCE – 2004, стр. 163 - 172 (in Bulgarian).
- [Minack, 2010] Enrico Minack, "RDF2RDF converter", <http://www.l3s.de/~minack/rdf2rdf/> 2010, (accessed: 31.07.2013).
- [Neave & Worthington, 1992] Neave, H., Worthington, P., "Distribution Free Tests", Routledge, 1992.
- [Nemenyi, 1963] Peter Nemenyi, "Distribution-free multiple comparisons Unpublished", PhD thesis; Princeton University Princeton, NJ, 1963
- [N-Quads, 2013] N-Quads: Extending N-Triples with Context <http://sw.deri.org/2008/07/n-quads/> (accessed: 16.03.2013).
- [Sesame, 2012] Sesame, OpenRDF, <http://www.openrdf.org/index.jsp>
<http://www.openrdf.org/doc/sesame2/2.3.2/users/userguide.html#chapter-sesame2-whats-new> (accessed: 01.12.2012)
- [SPARQL, 2013] SPARQL Query Language for RDF "W3C Recommendation", 2008, <http://www.w3.org/TR/rdf-sparql-query/> (accessed: 23.03.2013).
- [Virtuoso, 2013] OpenLink Virtuoso Universal Server: Documentation <http://docs.openlinksw.com/pdf/virtdocs.pdf>, <http://virtuoso.openlinksw.com/> (accessed: 23.03.2013)

Authors' Information



Ivanova Krassimira – University of National and World Economy, Sofia, Bulgaria;
e-mail: krasy78@mail.bg

Major Fields of Scientific Research: Software Engineering, Business Informatics, Data Mining, Multidimensional multi-layer data structures in self-structured systems