Test Report on gStore v0.4.0

Li, Zeng[1]

February 26, 2017

[1] EECS of Peking University, zengli-bookug@pku.edu.cn

Contents

1 Preface

gStore[1] is a graph-based database management system, which keeps the structure of original RDF[2] data.

The data model is directed graph with labels, and each vertex corresponds to a subject or object.

Given a SPARQL[3] query(only select...where clause is well supported now), gStore will transfer it to a directed graph with labels first.

Then the query problem will be equivalent to a subgraph matching problem. An index called VSTree is used in gStore to speed up the matching process. For each variable in the SPARQL query, gStore acquires its candidates through VSTree, and finally a join process is performed to get the final result.

We compare the performance of gStore with apache-jena[4], openrdf-sesame[5] and virtuoso-openlinksw[6] on several RDF datasets. The items needing to be considered include the time to build database, the size of database and the time to answer each SPARQL query. In addition, we will give a special explanation if the query results of each database do not match. (we will not consider the memory and disk cost except for special cases)

[1] https://github.com/Caesar11/gStore

[2] http://www.w3school.com.cn/rdf/

[3] https://www.w3.org/TR/sparql11-query/

[4] http://jena.apache.org/

[5] http://www.rdf4j.org/

[6] http://virtuoso.openlinksw.com/

2 Environment Setup

The experiment is finished on a Linux server, whose configuration is as follows:

Server	CentOS7
IP	172.31.222.78
memory	128G
disk	4T

Table 1: environment

The versions of all database management systems used here are all open source. Latest versions are choosed:

DBMS	VERSION
gStore	0.3.0
apache-jena	3.0.1
virtuoso-openlinksw	7.2

Table 2: dbms series

We should not include the time to load database indexes(called offline time) when comparing the time to answer SPARQL queries. And we need to empty the buffer and cache of operation system when the experiment for each database management system is over.

Besides, the time to answer a query shouldn't be too long. We will kill the running program if the time consumed is larger than 30 minutes, and set the running time as 1800000ms.

The datasets used include WatDiv[7], LUBM[8], BSBM[9] and DBpedia[10]. DBpedia are the background data of wikipedia, while the others are generated by programs. SPARQL queries are generated by programs or copied from other essays.

All datasets and queries we used are listed in this document, to provide a more thorough understanding of the experiment.

Below is for the WatDiv datasets, and the corresponding queries are placed in WatDiv Queries.

Dataset	Size(B)	Triple	Predicate	Entity	Literal
watdiv10M	15,743,004,966	109,795,918	86	5,212,745	5,077,247
watdiv100M	15,743,004,966	109,795,918	86	5,212,745	5,077,247
watdiv200M	31,712,545,025	219,714,495	86	10,424,745	9,976,964
watdiv300M	47,676,280,476	329,584,783	86	15,636,745	14,748,846
watdiv500M	72,326,509,429	500,000,000	76	26,060,745	23,964,574

Table 3: WatDiv series

Below is for the LUBM datasets, and the corresponding queries are placed in LUBM Queries.

Below is for the DBpedia datasets, and the corresponding queries are placed in DBpedia Queries.

Below is for the BSBM datasets, and the corresponding queries are placed in BSBM Queries.

[7] http://dsg.uwaterloo.ca/watdiv/

[8] http://swat.cse.lehigh.edu/projects/lubm/

[9] https://sourceforge.net/projects/bsbmtools/files/bsbmtools/

[10] http://wiki.dbpedia.org/

Dataset	Size(B)	Triple	Predicate	Entity	Literal
lubm10M	1,927,738,602	10,828,077	18	1,843,219	897,867
lubm100M	19,218,529,024	106,909,064	18	17,473,142	8,930,863
lubm200M	38,596,745,736	213,874,370	18	34,874,223	17,873,739
lubm300M	57,993,036,169	320,711,327	18	52,254,606	26,804,722
lubm500M	85,171,063,439	500,000,000	18	81,342,489	41,804,418

Table 4: LUBM series

Dataset	Size(B)	Triple	Predicate	Entity	Literal
dbpedia170M	23,844,158,944	170,784,508	57,354	7,123,915	14,971,449

Table 5: DBpedia series

Dataset	Size(B)	Triple	Predicate	Entity	Literal
bsbm10M	2,738,760,016	10,538,484	40	11,566,839	1,312,881
bsbm100M	27,349,978,858	104,115,556	40	15,522,017	9,168,781
bsbm200M	54,788,814,738	208,134,846	40	31,042,129	17,225,761
bsbm300M	82,239,133,084	311,957,992	40	46,514,164	25,176,573
bsbm500M	137,445,999,355	519,594,458	40	77,457,230	41,063,231

Table 6: BSBM series

3 Experiment Result

All results are saved in load.log/, result.log/ and time.log/, and the format is TSV.

Table 7 shows the index size and loading time of the datasets for different systems.

	Index Size(KB)			Loading Time(ms)		
Datasets	gStore	Jena	Virtuoso	gStore	Jena	Virtuoso
dbpedia170M	25,549,812	23,151,404	18,173,919	4,516,359	28,567,000	38,580,197
bsbm10M	3,900,000	2,100,000	2,200,000	1,368,388	154,000	414,145
bsbm100M	38,000,000	20,000,000	16,000,000	1,368,388	1,699,000	4,670,565
bsbm200M	71,000,000	40,000,000	32,000,000	1,368,388	3,452,000	23,405,765
bsbm300M	243,000,000	60,000,000	33,000,000	87,702,486	5,448,000	42,047,477
bsbm500M	185,000,000	78,000,000	57,564,700	170,688,614	8,722,000	68,692,273
lubm10M	2,858,700	1,689,040	7,300,186	248,535	105,000	206,905
lubm100M	28,821,768	16,758,868	5,853,150	5,549,092	1,105,000	2,571,964
lubm200M	55,359,384	33,571,816	13,816,000	27,125,224	2,642,000	7,145,964
lubm300M	82,566,460	50,229,800	16,315,800	57,033,126	4,098,000	11,600,040
lubm500M	133,000,000	89,000,000	83,611,400	91,089,242	8,555,000	46,532,010
watdiv10M	1,438,176	1,246,276	8,275,360	228,388	171,000	107,611
watdiv100M	1,416,572	12,731,144	5,989,466	3,253,273	2,133,000	3,401,298
watdiv200M	28,899,484	25,441,824	14,516,500	10,746,940	4,350,000	8,705,439
watdiv300M	43,276,644	37,950,448	17,593,000	6,595,351	6,453,000	16,817,187
watdiv500M	64,000,000	57,000,000	26,925,300	19,638,536	9,881,000	34,325,820

Table 7: Offline Performance

The performance of different database management systems is shown in Figures 1, **??**, 5 and 6, 8 and 9.

Notice that storage buffer size is set to 8G when testing lubm500M, while 4G for other cases. The block size is set to 64K when testing lubm500M, while 4K in other cases. The query results for lubm500M are all empty, so the time is very fast and we can not tell which system is better.

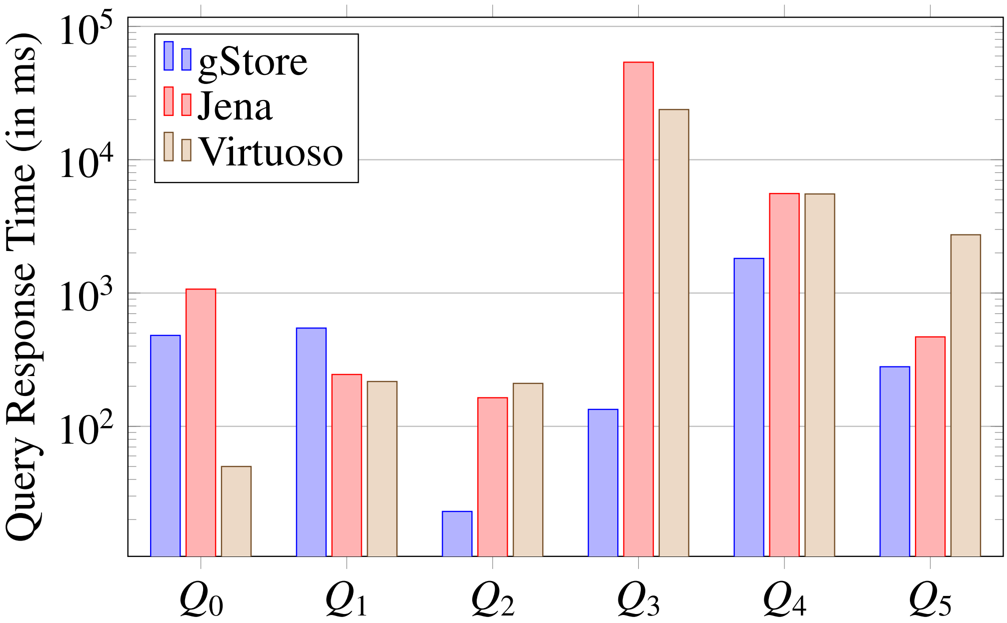
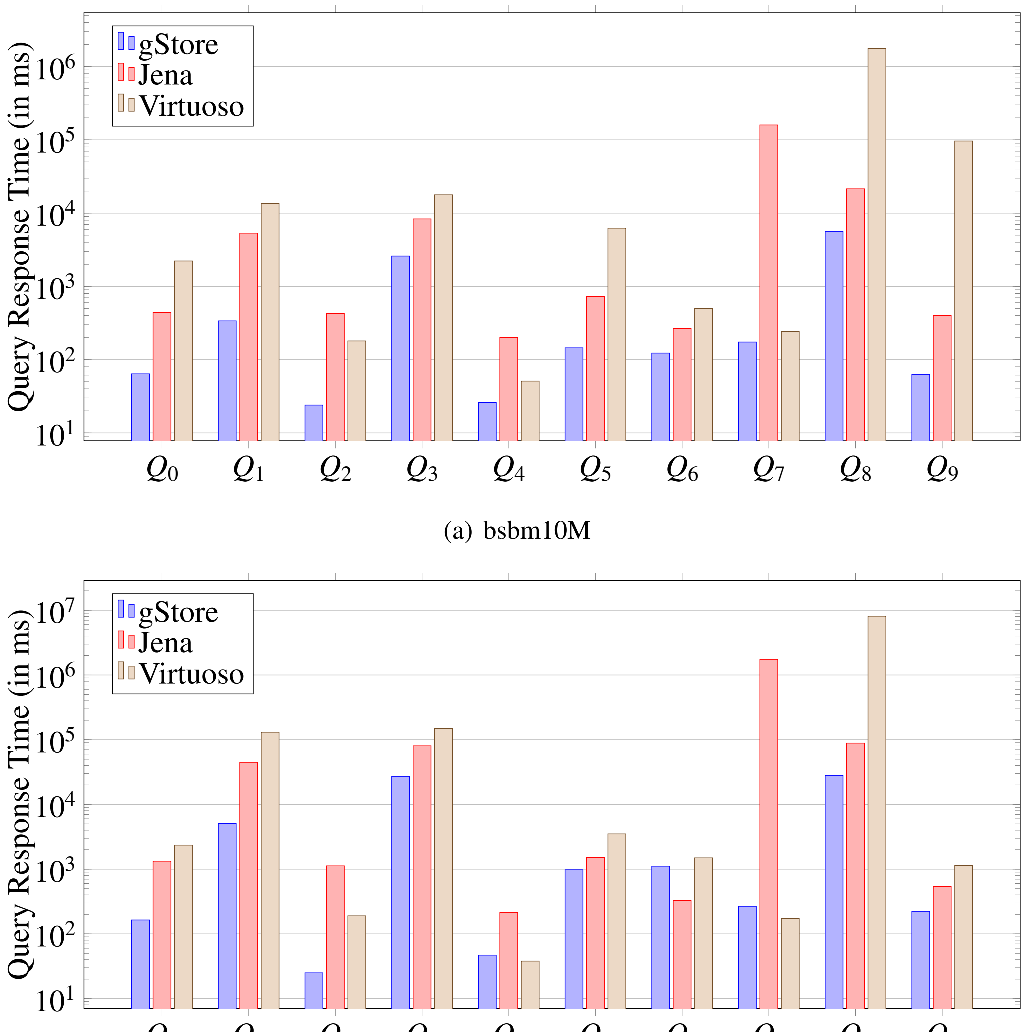


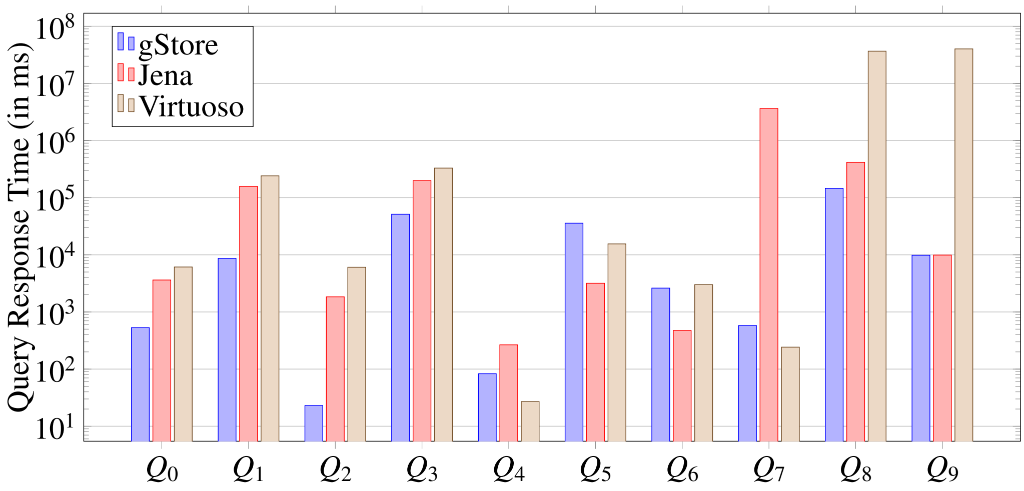
Figure 1: Query Performance over dbpedia170M



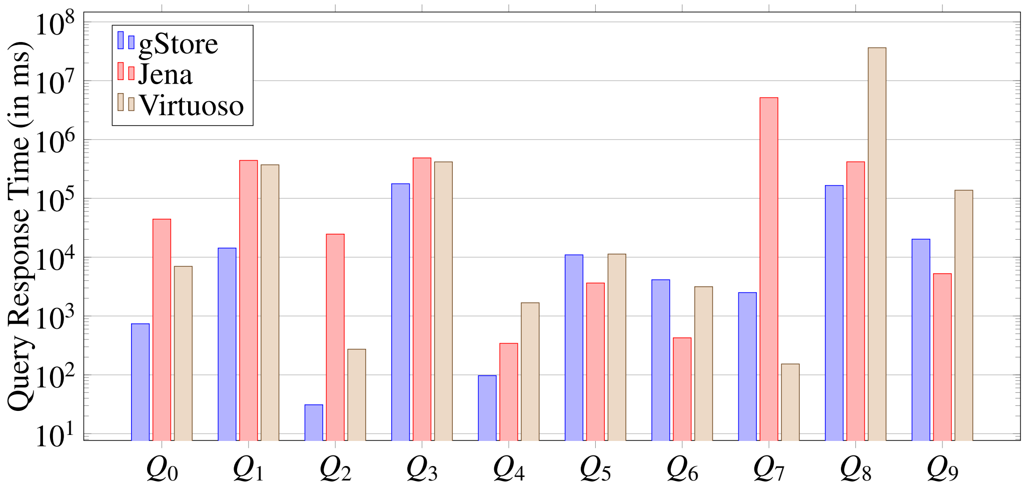
(a) bsbm10M

(b) bsbm100M

Figure 2: Query Performance over BSBM 10M and 100M

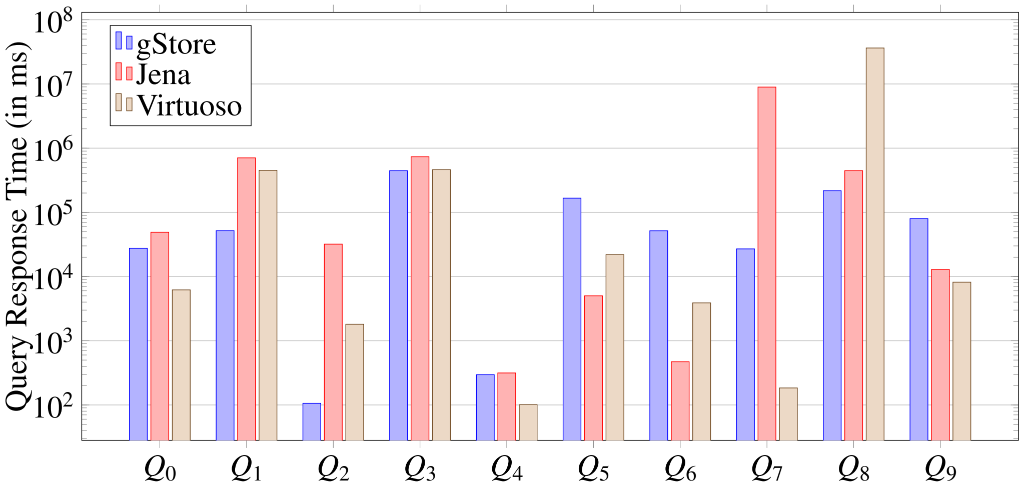


(a) bsbm200M



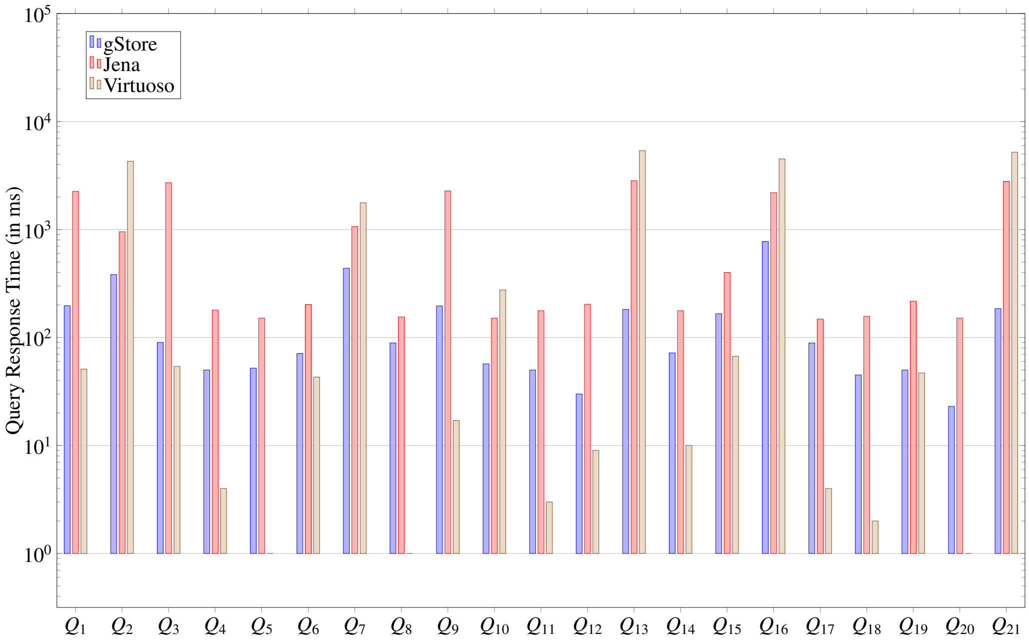
(b) bsbm300M

Figure 3: Query Performance over BSBM 200M and 300M

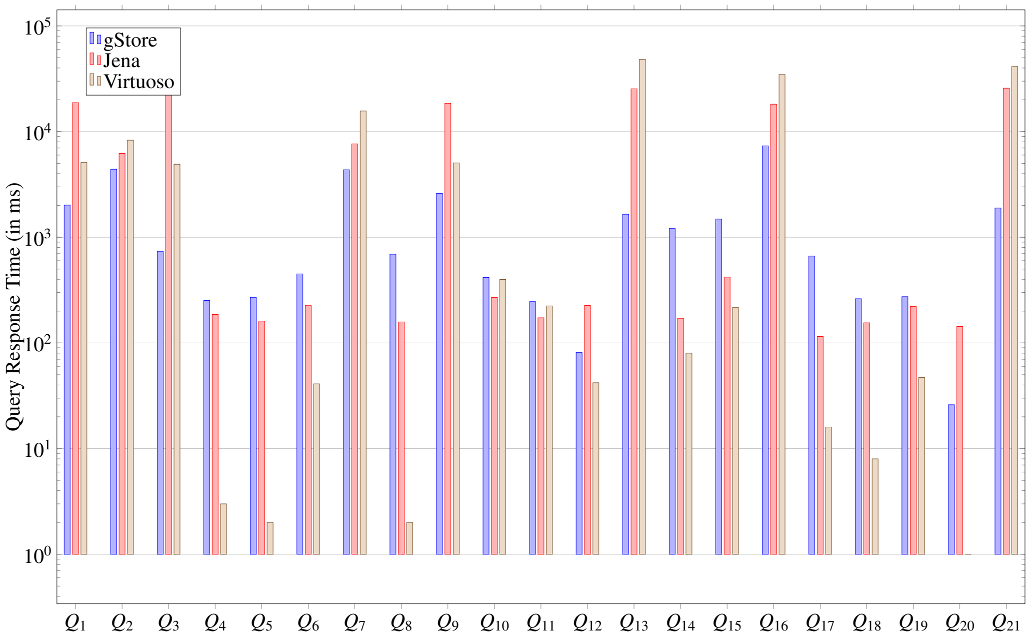


(a) bsbm500M

Figure 4: Query Performance over BSBM 500M

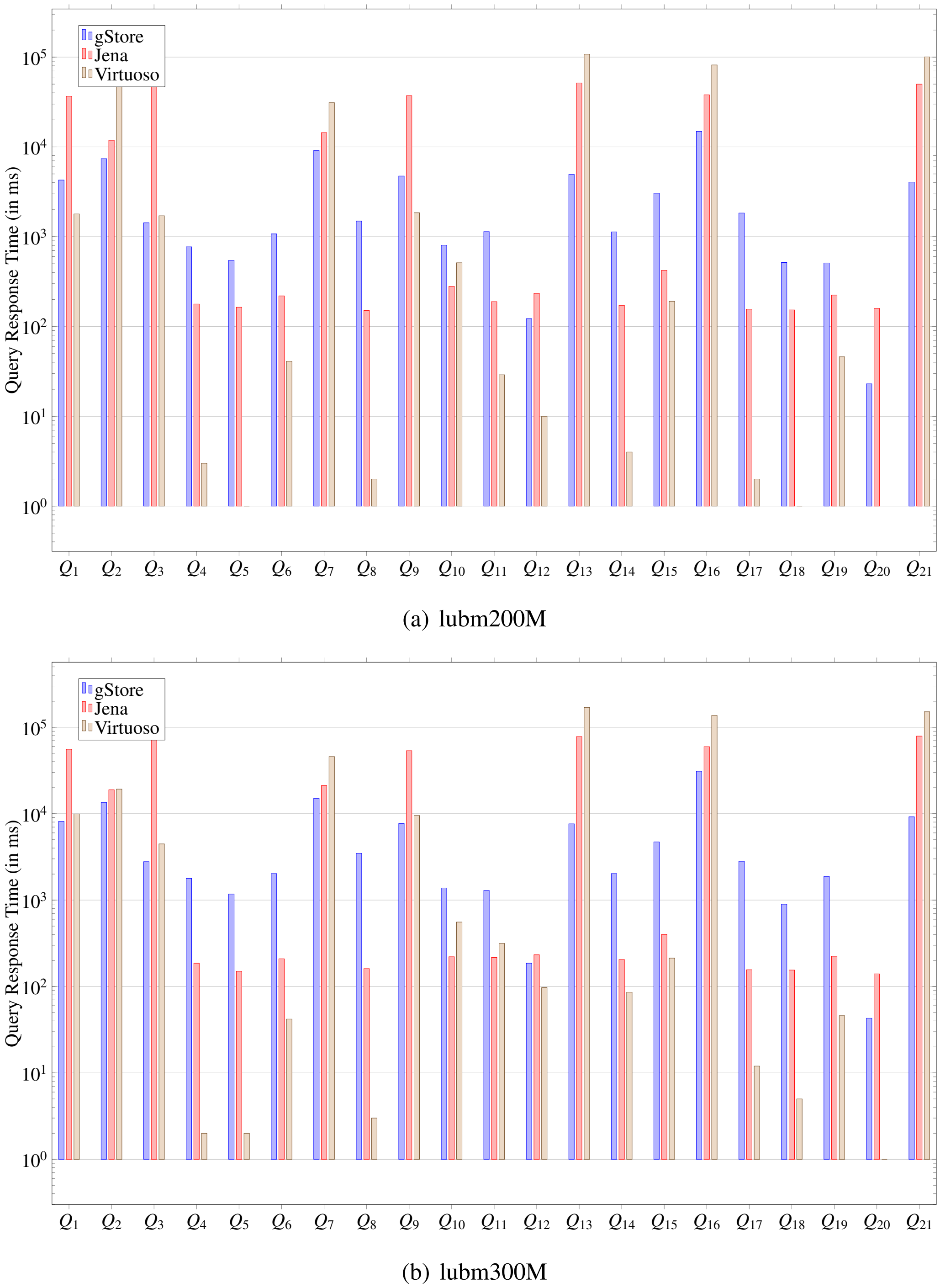


(a) lubm10M



(b) lubm100M

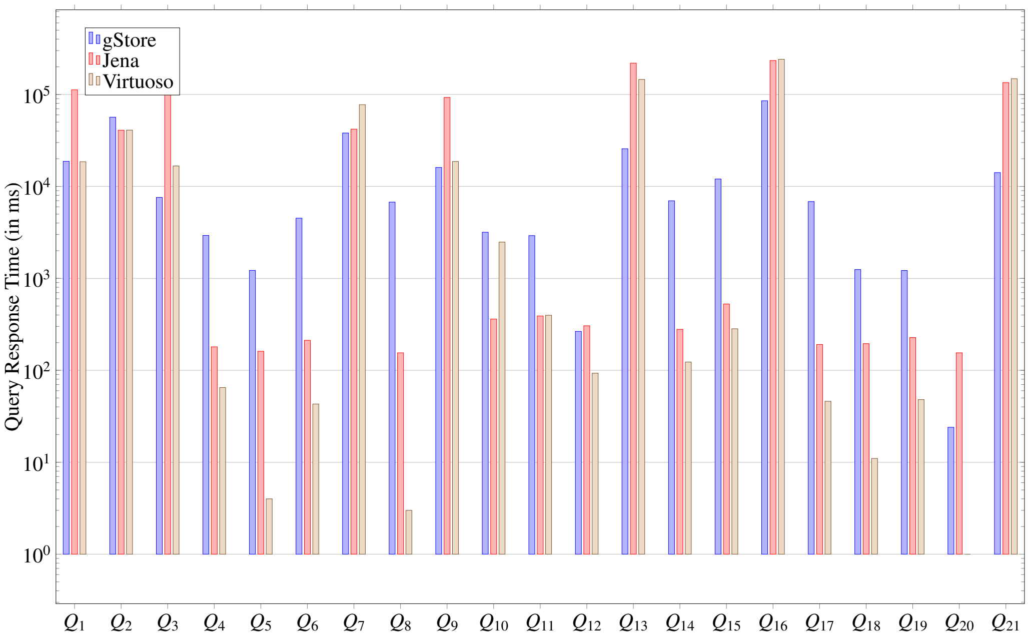
Figure 5: Query Performance over LUBM 10M and 100M



(a) lubm200M

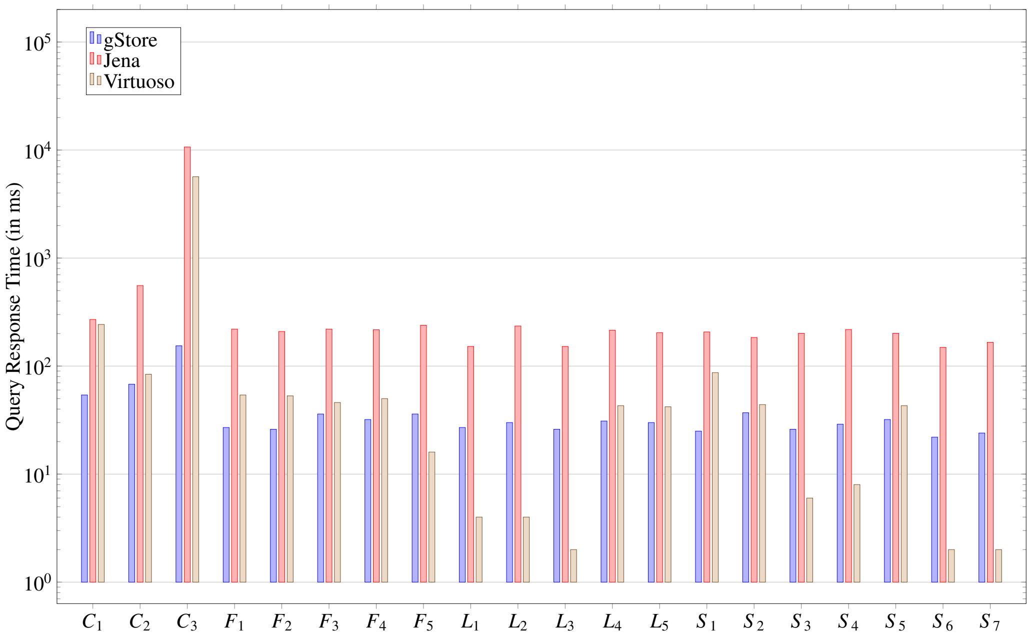
(b) lubm300M

Figure 6: Query Performance over LUBM 200M and 300M

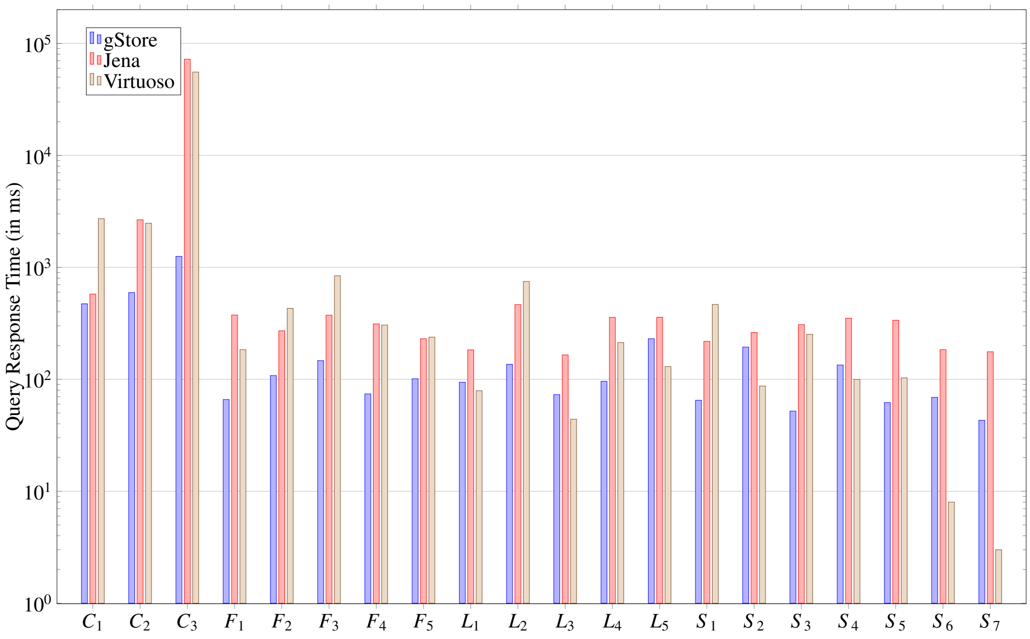


(a) lubm500M

Figure 7: Query Performance over LUBM 500M

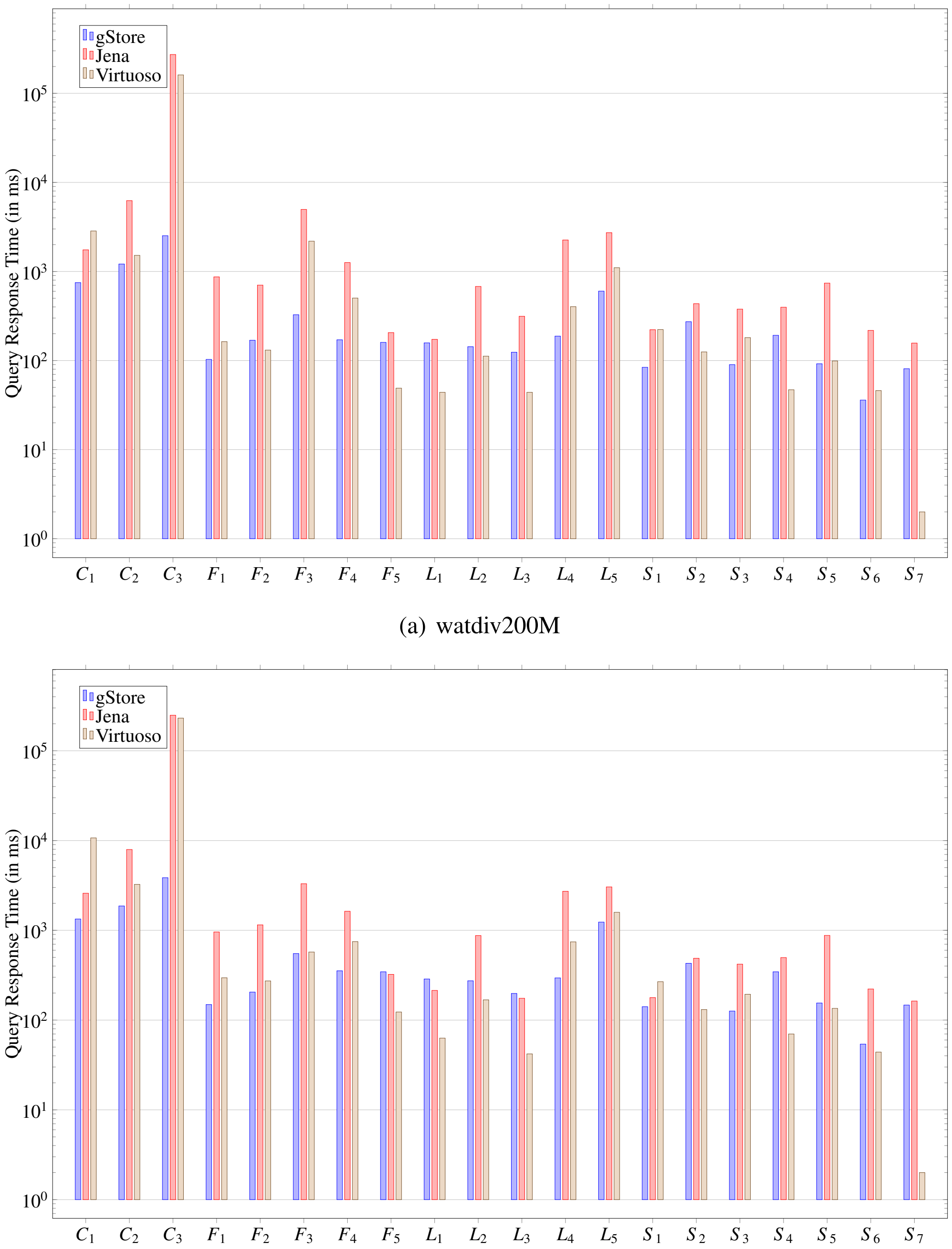


(a) watdiv10M



(b) watdiv100M

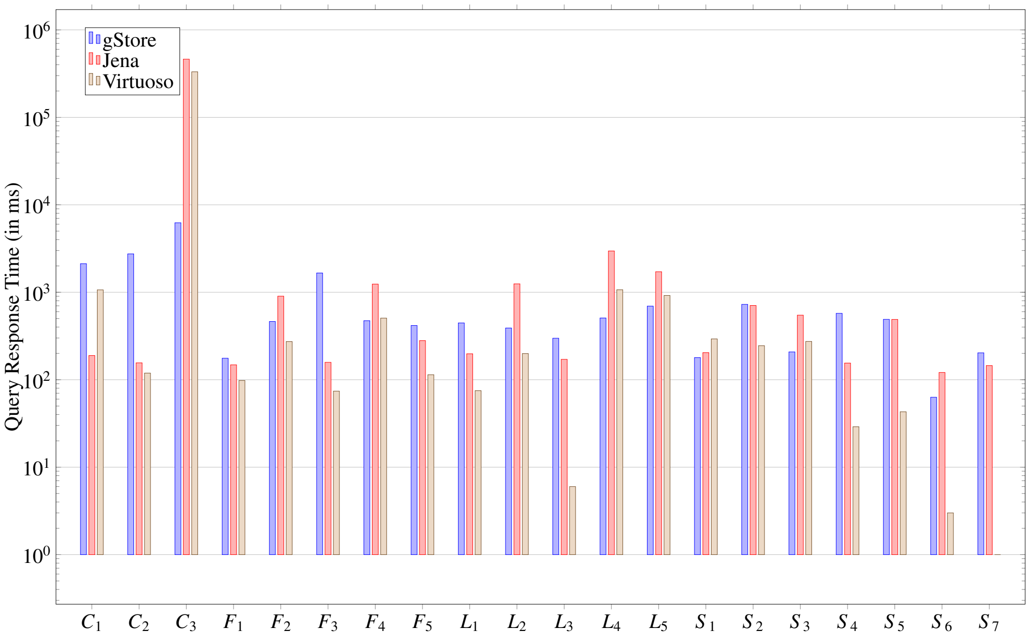
Figure 8: Query Performance over WatDiv 10M and 100M



(a) watdiv200M

(b) watdiv300M

Figure 9: Query Performance over WatDiv 200M and 300M



(a) watdiv500M

Figure 10: Query Performance over WatDiv 500M

4 Modification

We provide insertion and deletion in gStore v0.4.0. You can either insert/delete from a given RDF file, or just run sparql queries to insert/delete something. If you want to modify something, you need to delete it and reinsert. The cost of insertion and deletion are recorded in table 8, where the time unit is ms. Generally, the update cost is related with the original database size. Here we build the database first from a dataset with 99550 triples, then we do updates of different scales.

Triples(lubm)	insert	delete
10	389	396
100	400	484
1000	459	1430
10000	985	10303

Table 8: Insertion and Deletion

To sum up, updates can be handled efficiently now, i.e. 1000 triples can be inserted or removed in a second. As you can see in the table above, the time of insert process varies slightly while the time of remove process varies greatly, when the num of updates grows. The reason may be that gStore has to recalculate the signatures in remove process, while in the insert process gStore only has to take an OR bit-wise operation.

However, bugs do exist in insertion/deletion. When testing on lubm66M, the answer of q1.sql and q2.sql are not all right if the operation order is: build, delete, insert, query. More precisely, a few results are lost, though the proportion is really small. We do not care about the efficiency of insert/delete, but the correctness is a must, which means we will try to fix this bug as quickly as possible.

5 Conclusion

gStore can go well with RDF datasets which are in N-Triples format and TTL format, while the other database management systems may come across some questions. In addition, gStore outperforms other systems on many SPARQL queries. What is more, gStore is highly extensively because it uses graph model instead of relational model.

However, there are also some shortcomings for gStore:

1. RDF datasets in XML format are not supported
2. the disk cost is high
3. gStore is sometimes slower

Besides, gStore v0.3.0 does not generate solutions for satellites which are not selected. This will speed up the query answering process, while not keeping so many duplicates in the result set. For example, in below query, let's assume that ?s has only one unique answer, but ?o1 and ?o2 both have 10,000 answers. In previous versions of gStore, there are 100,000,000 records in the result set because we have to find the answer of ?s and generate the solutions for ?o1 and ?o2, even if only ?s is selected in the sparql query. However, in the v0.3.0, we find the answer of ?s and return directly. In this case, there won't be so many duplicates in the result set as before, but this is ok.

Currently, gStore needs 40G memory and 50G disk to run LUBM300M. Meanwhile, the time of build process is also speeded up, as well as the query process. As for the build process, the bottleneck is the time of building vstree now.

What is more, we have enabled the gStore to run dbpedia_en_large, a dataset with 800M triples. Maybe gStore will run into crash if building other datasets with

800M triples, but it do well in this dataset.

```
select  ?s  where
{
  ?s  <linkTo> ?o1  .
  ?s  <produce> ?o2  .
}
```

Out of question, the performance of gStore can be improved a lot later. The future work is listed below:

1. fix the problem in insertion/deletion
2. support datasets of 1 billion triples in a single machine(only 500 million now)
3. add unit testing for the whole system(only black-box testing now)
4. do code level optimization(for example, large loop in Join module)
5. speed up the table join process using pipeline

6 Appendix

6.1 WatDiv queries

These queries come from [3].

6.1.1 C1.sql

```
SELECT ?v0 ?v4 ?v6 ?v7 WHERE
{
  ?v0 <http://schema.org/caption> ?v1 .
  ?v0 <http://schema.org/text> ?v2 .
  ?v0 <http://schema.org/contentRating> ?v3 .
  ?v0 <http://purl.org/stuff/rev#hasReview> ?v4 .
  ?v4 <http://purl.org/stuff/rev#title> ?v5 .
  ?v4 <http://purl.org/stuff/rev#reviewer> ?v6 .
  ?v7 <http://schema.org/actor> ?v6 .
  ?v7 <http://schema.org/language> ?v8 .
}
```

6.1.2 C2.sql

```
SELECT ?v0 ?v3 ?v4 ?v8 WHERE
{
  ?v0 <http://schema.org/legalName> ?v1 .
  ?v0 <http://purl.org/goodrelations/offers> ?v2 .
  ?v2 <http://schema.org/eligibleRegion> <http://db.uwaterloo.ca/~
        galuc/wsdbm/Country5> .
  ?v2 <http://purl.org/goodrelations/includes> ?v3 .
  ?v4 <http://schema.org/jobTitle> ?v5 .
  ?v4 <http://xmlns.com/foaf/homepage> ?v6 .
  ?v4 <http://db.uwaterloo.ca/~galuc/wsdbm/makesPurchase> ?v7 .
  ?v7 <http://db.uwaterloo.ca/~galuc/wsdbm/purchaseFor> ?v3 .
  ?v3 <http://purl.org/stuff/rev#hasReview> ?v8 .
  ?v8 <http://purl.org/stuff/rev#totalVotes> ?v9 .
}
```

6.1.3 C3.sql

```
SELECT ?v0 WHERE
{
  ?v0 <http :// db.uwaterloo . ca/~galuc / wsdbm/likes> ?v1  .
  ?v0 <http :// db.uwaterloo . ca/~galuc / wsdbm/friendOf> ?v2 .
  ?v0 <http :// purl . org / dc/terms / Location> ?v3  .
  ?v0 <http :// xmlns.com/foaf / age> ?v4  .
  ?v0 <http :// db.uwaterloo . ca/~galuc / wsdbm/gender> ?v5 .
  ?v0 <http :// xmlns.com/foaf / givenName> ?v6 .
}
```

6.1.4 F1.sql

```
SELECT ?v0 ?v2 ?v3 ?v4 ?v5 WHERE
{
  ?v0 <http :// ogp.me/ns#tag> <http :// db.uwaterloo . ca/~galuc / wsdbm/
          Topic103> .
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> ?v2 .
  ?v3 <http :// schema.org/  trailer > ?v4  .
  ?v3 <http :// schema.org/keywords> ?v5  .
  ?v3 <http :// db.uwaterloo . ca/~galuc / wsdbm/hasGenre> ?v0 .
  ?v3 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> <http://db.
          uwaterloo . ca/~galuc / wsdbm/ProductCategory2> .
}
```

6.1.5 F2.sql

```
SELECT ?v0 ?v1 ?v2 ?v4 ?v5 ?v6 ?v7 WHERE
{
  ?v0 <http :// xmlns.com/foaf / homepage> ?v1 .
  ?v0 <http :// ogp.me/ns# title > ?v2 .
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> ?v3 .
  ?v0 <http :// schema.org/ caption> ?v4 .
  ?v0 <http :// schema.org/ description > ?v5 .
  ?v1 <http :// schema.org/ url> ?v6 .
  ?v1 <http :// db.uwaterloo.ca/~galuc/wsdbm/hits> ?v7 .
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/hasGenre> <http://db.
        uwaterloo.ca/~galuc/wsdbm/SubGenre35> .
}
```

6.1.6 F3.sql

```
SELECT ?v0 ?v1 ?v2 ?v4 ?v5 ?v6 WHERE
{
  ?v0 <http :// schema.org/ contentRating> ?v1 .
  ?v0 <http :// schema.org/ contentSize> ?v2 .
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/hasGenre> <http://db.
        uwaterloo.ca/~galuc/wsdbm/SubGenre59> .
  ?v4 <http :// db.uwaterloo.ca/~galuc/wsdbm/makesPurchase> ?v5 .
  ?v5 <http :// db.uwaterloo.ca/~galuc/wsdbm/purchaseDate> ?v6 .
  ?v5 <http :// db.uwaterloo.ca/~galuc/wsdbm/purchaseFor> ?v0 .
}
```

6.1.7 F4.sql

```
 select  ?v0 ?v1 ?v2 ?v4 ?v5 ?v6 ?v7 ?v8 where {
?v0 <http :// xmlns.com/foaf/homepage>  ?v1 .
?v2 <http :// purl.org/ goodrelations / includes>  ?v0 .
?v0 <http :// ogp.me/ns#tag>  <http :// db.uwaterloo.ca/~galuc/wsdbm/
        Topic60> .
?v0 <http :// schema.org/ description > ?v4 .
?v0 <http :// schema.org/ contentSize> ?v8 .
?v7 <http :// db.uwaterloo.ca/~galuc/wsdbm/likes> ?v0 .
?v1 <http :// schema.org/ url> ?v5 .
?v1 <http://db.uwaterloo.ca/~galuc/wsdbm/hits>  ?v6 .
?v1 <http://schema.org/language>  <http://db.uwaterloo.ca/~galuc/wsdbm/Language0> .
}
```

6.1.8 F5.sql

```
 select  ?v0 ?v1 ?v3 ?v4 ?v5 ?v6 where {
?v0 <http://purl.org/goodrelations/includes>  ?v1 .
<http://db.uwaterloo.ca/~galuc/wsdbm/Retailer661> <http://purl.org/goodrelations/offers>  ?v0 .
?v0 <http://purl.org/goodrelations/price> ?v3 .
?v0 <http://purl.org/goodrelations/validThrough>  ?v4 .
?v1 <http://ogp.me/ns#title>  ?v5 .
?v1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?v6 .
}
```

6.1.9 L1.sql

```
SELECT ?v0 ?v2 ?v3 WHERE
{
  ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/subscribes> <http://db.uwaterloo.ca/~galuc/wsdbm/Website38303> .
  ?v2 <http://schema.org/caption> ?v3 .
  ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/likes> ?v2 .
}
```

6.1.10 L2.sql

```
SELECT ?v1 ?v2 WHERE
{
  <http://db.uwaterloo.ca/~galuc/wsdbm/City131> <http://www.geonames.org/ontology#parentCountry> ?v1 .
  ?v2 <http :// db.uwaterloo.ca/~galuc/wsdbm/likes> <http :// db.uwaterloo.ca/~galuc/wsdbm/Product0> .
  ?v2 <http :// schema.org/ nationality > ?v1  .
}
```

6.1.11 L3.sql

```
SELECT ?v0 ?v1 WHERE
{
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/likes> ?v1  .
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/subscribes> <http :// db.uwaterloo.ca/~galuc/wsdbm/Website20769> .
}
```

6.1.12 L4.sql

```
 select  ?v0 ?v2 where
{
?v0 <http :// ogp.me/ns#tag>  <http :// db.uwaterloo.ca/~galuc/wsdbm/Topic117> .
?v0 <http :// schema.org/caption> ?v2  .
}
```

6.1.13 L5.sql

```
 select  ?v0 ?v1 ?v3 where
{
?v0 <http :// schema.org/ jobTitle >   ?v1  .
?v0 <http :// schema.org/ nationality > ?v3  .
<http :// db.uwaterloo.ca/~galuc/wsdbm/City22>  <http :// www.geonames.org/ontology#parentCountry>  ?v3 .
}
```

6.1.14 S1.sql

```
SELECT ?v0 ?v1 ?v3 ?v4 ?v5 ?v6 ?v7 ?v8 ?v9 WHERE
{
  ?v0 <http://purl.org/goodrelations/includes> ?v1 .
  <http://db.uwaterloo.ca/~galuc/wsdbm/Retailer391> <http://purl.org
        /goodrelations/offers> ?v0 .
  ?v0 <http://purl.org/goodrelations/price> ?v3 .
  ?v0 <http://purl.org/goodrelations/serialNumber> ?v4 .
  ?v0 <http://purl.org/goodrelations/validFrom> ?v5 .
  ?v0 <http://purl.org/goodrelations/validThrough> ?v6 .
  ?v0 <http://schema.org/eligibleQuantity> ?v7 .
  ?v0 <http://schema.org/eligibleRegion> ?v8 .
  ?v0 <http://schema.org/priceValidUntil> ?v9 .
}
```

6.1.15 S2.sql

```
SELECT ?v0 ?v1 ?v3 WHERE
{
  ?v0 <http://purl.org/dc/terms/Location> ?v1 .
  ?v0 <http://schema.org/nationality> <http://db.uwaterloo.ca/~galuc
        /wsdbm/Country23> .
  ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/gender> ?v3 .
  ?v0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://db.
        uwaterloo.ca/~galuc/wsdbm/Role2> .
}
```

6.1.16 S3.sql

```
SELECT ?v0 ?v2 ?v3 ?v4 WHERE
{
  ?v0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://db.
        uwaterloo.ca/~galuc/wsdbm/ProductCategory12> .
  ?v0 <http :// schema.org/ caption> ?v2 .
  ?v0 <http :// db.uwaterloo.ca/~galuc/wsdbm/hasGenre> ?v3 .
  ?v0 <http :// schema.org/ publisher> ?v4 .
}
```

6.1.17 S4.sql

```
select ?v0 ?v2 ?v3 where
{
?v0 <http :// xmlns.com/foaf/age> <http :// db.uwaterloo.ca/~galuc/
        wsdbm/AgeGroup1> .
?v0 <http :// xmlns.com/foaf/familyName> ?v2 .
?v3 <http :// purl.org/ontology/mo/ artist > ?v0 .
?v0 <http :// schema.org/ nationality > <http :// db.uwaterloo.ca/~galuc/
        wsdbm/Country1> .
}
```

6.1.18 S5.sql

```
select ?v0 ?v2 ?v3 where
{
?v0 <http :// www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://db.
        uwaterloo.ca/~galuc/wsdbm/ProductCategory3> .
?v0 <http :// schema.org/ description > ?v2 .
?v0 <http :// schema.org/keywords> ?v3 .
?v0 <http :// schema.org/language> <http :// db.uwaterloo.ca/~galuc/
        wsdbm/Language0> .
}
```

6.1.19 S6.sql

```
select ?v0 ?v1 ?v2 where
{
```

```
?v0 <http://purl.org/ontology/mo/conductor> ?v1 .
?v0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?v2 .
?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/hasGenre> <http://db.uwaterloo.ca/~galuc/wsdbm/SubGenre1> .
}
```

6.1.20 S7.sql

```
select ?v0 ?v1 ?v2 where
{
?v0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?v1 .
?v0 <http://schema.org/text> ?v2 .
<http://db.uwaterloo.ca/~galuc/wsdbm/User2094> <http://db.uwaterloo.ca/~galuc/wsdbm/likes> ?v0 .
}
```

6.2 LUBM queries

This queries come from two places. $q1 \sim q7$ come from [1] and [3]. $q8 \sim q21$ come from [2] and [4].

6.2.1 q1.sql

```
select ?x where
{
  ?x      <rdf:type>      <ub:GraduateStudent>.
  ?y      <rdf:type>      <ub:University>.
  ?z      <rdf:type>      <ub:Department>.
  ?x      <ub:memberOf>  ?z.
  ?z      <ub:subOrganizationOf>  ?y.
  ?x      <ub:undergraduateDegreeFrom>   ?y.
}
```

6.2.2 q2.sql

```
select  ?x  where
{
  ?x        <rdf : type>        <ub:Course>.
  ?x        <ub:name>           ?y .
}
```

6.2.3 q3.sql

```
select  ?x  where
{
  ?x        <rdf : type>        <ub:UndergraduateStudent>.
  ?y        <rdf : type>        <ub: University >.
  ?z        <rdf : type>        <ub:Department>.
  ?x        <ub:memberOf>   ?z.
  ?z        <ub:subOrganizationOf>   ?y .
  ?x        <b:undergraduateDegreeFrom>       ?y .
}
```

6.2.4 q4.sql

```
elect  ?x  ?y1  ?y2  ?y3  where
{
  ?x        <ub:worksFor>    <http :// www.Department0.University0.edu>.
  ?x        <rdf : type>        <ub: FullProfessor >.
  ?x        <ub:name>           ?y1 .
  ?x        <ub:emailAddress>           ?y2 .
  ?x        <ub:telephone>    ?y3 .
}
```

6.2.5 q5.sql

```
select ?x where
{
  ?x      <ub:subOrganizationOf>  <http :// www.Department0.
        University0.edu>.
  ?x      <rdf:type>      <ub:ResearchGroup>.
}
```

6.2.6 q6.sql

```
select ?x ?y where
{
  ?y      <ub:subOrganizationOf>  <http :// www.University0.edu>.
  ?y      <rdf:type>      <ub:Department>.
  ?x      <ub:worksFor>   ?y.
  ?x      <rdf:type>      <ub: FullProfessor >.
}
```

6.2.7 q7.sql

```
select ?x ?y ?z where
{
  ?x      <rdf:type>      <ub:UndergraduateStudent>.
  ?y      <rdf:type>      <ub: FullProfessor >.
  ?z      <rdf:type>      <ub:Course>.
  ?x      <ub:advisor>    ?y.
  ?x      <ub:takesCourse>        ?z.
  ?y      <ub:teacherOf>  ?z.
}
```

6.2.8 q8.sql

```
select ?X where
```

```
{
  ?X      <rdf:type>      <ub:GraduateStudent>.
  ?X      <ub:takesCourse>          <http :// www.Department0.
          University0.edu/GraduateCourse0>.
}
```

6.2.9 q9.sql

```
select  ?X ?Y ?Z where
{
  ?X      <rdf:type>      <ub:GraduateStudent>.
  ?Y      <rdf:type>      <ub: University >.
  ?Z      <rdf:type>      <ub:Department>.
  ?X      <ub:memberOf>   ?Z.
  ?Z      <ub:subOrganizationOf>   ?Y.
  ?X      <ub:undergraduateDegreeFrom>     ?Y.
}
```

6.2.10 q10.sql

```
select  ?X where
{
  ?X      <rdf:type>      <ub: Publication >.
  ?X      <ub: publicationAuthor >   <http :// www.Department0.
          University0.edu/ AssistantProfessor0 >.
}
```

6.2.11 q11.sql

```
select  ?Y1 ?Y2 ?Y3 where
{
  ?X      <rdf:type>      <ub: FullProfessor >.
  ?X      <ub:worksFor>   <http :// www.Department0.University0.edu>.
  ?X       <ub:name>        ?Y1.
  ?X       <ub:emailAddress>         ?Y2.
  ?X       <ub:telephone>   ?Y3.
}
```

6.2.12 q12.sql

```
select ?X where
{
  ?X       <ub:memberOf>   <http :// www.Department0.University0.edu>.
}
```

6.2.13 q13.sql

```
select ?X where
{
  ?X       <rdf:type>       <ub:UndergraduateStudent>.
}
```

6.2.14 q14.sql

```
elect ?X ?Y where
{
  ?X       <rdf:type>       <ub:Student>.
  ?Y       <rdf:type>       <ub:Course>.
  ?X       <ub:takesCourse>           ?Y.
  <http :// www.Department0.University0.edu/ AssociateProfessor0 >      <
           ub:teacherOf>   ?Y.
}
```

6.2.15 q15.sql

```
select  ?X where
{
  ?X      <rdf:type>       <ub:UndergraduateStudent>.
  ?Y      <rdf:type>       <ub:Department>.
  ?X      <ub:memberOf>   ?Y.
  ?Y      <ub:subOrganizationOf>  <http :// www.University0.edu>.
  ?X      <ub:emailAddress>         ?Z.
}
```

6.2.16 q16.sql

```
select  ?X ?Y ?Z where
{
  ?X      <rdf:type>       <ub:UndergraduateStudent>.
  ?Z      <rdf:type>       <ub:Course>.
  ?X      <ub:advisor>     ?Y.
  ?Y      <ub:teacherOf>   ?Z.
  ?X      <ub:takesCourse>           ?Z.
}
```

6.2.17 q17.sql

```
select  ?X where
{
  ?X      <rdf:type>       <ub:GraduateStudent>.
  ?X      <ub:takesCourse>           <http :// www.Department0.
          University0.edu/GraduateCourse0>.
}
```

6.2.18 q18.sql

```
select  ?X where
```

```
{
  ?X       <rdf:type>        <ub:ResearchGroup>.
  ?X       <ub:subOrganizationOf>   <http :// www.University0.edu>.
}
```

6.2.19 q19.sql

```
select  ?X ?Y where
{
  ?Y       <rdf:type>        <ub:Department>.
  ?X       <ub:worksFor>    ?Y.
  ?Y       <ub:subOrganizationOf>   <http :// www.University0.edu>.
}
```

6.2.20 q20.sql

```
select  ?X where
{
  <http :// www.University0.edu>     <ub:undergraduateDegreeFrom>     ?X
           .
}
```

6.2.21 q21.sql

```
select  ?X where
{
  ?X       <rdf:type>        <ub:UndergraduateStudent>.
}
```

6.3 DBpedia queries

These queries are written by us, imitating queries in other benchmarks.

6.3.1 q0.sql

```
select ?v0 where
{
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> <http:// dbpedia.org/ class /yago/LanguagesOfBotswana> .
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type>   <http :// dbpedia . org/ class /yago/LanguagesOfNamibia> .
  ?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> <http:// dbpedia.org/ontology/Language> .
}
```

6.3.2 q1.sql

```
select ?v0 where
{
  ?v0 <http :// dbpedia . org/ontology/associatedBand> <http :// dbpedia . org/ resource /LCD_Soundsystem> .
}
```

6.3.3 q2.sql

```
select ?v2 where
{
  <http :// dbpedia . org/ resource /!! Destroy−Oh−Boy!!> <http://dbpedia. org/ property / title > ?v2 .
}
```

6.3.4 q3.sql

```
select ?v0 ?v2 where
{
  ?v0 <http :// dbpedia . org / ontology / activeYearsStartYear > ?v2 .
}
```

6.3.5 q4.sql

```
select ?v0 ?v1 ?v2 where
{
  ?v0 <http :// dbpedia . org / property / dateOfBirth> ?v2 .
  ?v1 <http :// dbpedia . org / property / genre> ?v2 .
}
```

6.3.6 q5.sql

```
select ?v0 ?v1 ?v2 ?v3 where
{
  ?v0 <http :// dbpedia . org / property / familycolor > ?v1 .
  ?v0 <http :// dbpedia . org / property / glotto > ?v2 .
  ?v0 <http :// dbpedia . org / property / lc> ?v3 .
}
```

6.3.7 q6.sql

```
select ?v0 ?v1 ?v2 ?v3 ?v4 ?v5 ?v6 ?v7 ?v8 ?v9 where
{
  ?v0 <http :// dbpedia . org / property / dateOfBirth> ?v1 .
  ?v0 <http :// dbpedia . org / property / genre> ?v2 .
  ?v0 <http :// dbpedia . org / property / instrument> ?v3 .
  ?v0 <http :// dbpedia . org / property / label > ?v4 .
  ?v0 <http :// dbpedia . org / property / placeOfBirth> ?v5 .
  ?v6 <http :// dbpedia . org / property / name> ?v7 .
  ?v6 <http :// dbpedia . org / property / occupation> ?v8 .
  ?v6 <http :// dbpedia . org / property / placeOfBirth> ?v5 .
  ?v6 <http :// dbpedia . org / property / instrument> ?v3 .
  ?v6 <http://dbpedia.org/property/notableInstruments> ?v9 .
}
```

6.4 BSBM queries

6.4.1 q0.sql

```
select ?v0 where
{
?v0 <http :// www4.wiwiss.fu−berlin.de/ bizer /bsbm/v01/vocabulary/
            rating2 > "6"^^<http://www.w3.org/2001/XMLSchema#integer
            > .
}
```

6.4.2 q1.sql

```
select ?v0 ?v1 where
{
?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> ?v1 .
}
```

6.4.3 q2.sql

```
select ?v0 ?v1 where
{
?v0 <http :// www.w3.org/2000/01/rdf−schema#label> ?v1 .
?v1 <http :// purl . org/dc/elements /1.1/ publisher > ?v2 .
}
```

6.4.4 q3.sql

```
select ?v0 ?v1 ?v2 where
{
?v0 <http :// purl . org/dc/elements /1.1/ publisher > ?v1 .
?v0 <http :// purl . org/dc/elements /1.1/ date> ?v2 .
}
```

6.4.5 q4.sql

```
select  ?v0 ?v1 ?v2 where
{
?v0 <http :// www.w3.org/2000/01/rdf−schema#label> ?v2 .
?v1 <http :// www.w3.org/2000/01/rdf−schema#subClassOf> ?v2 .
}
```

6.4.6 q5.sql

```
select  ?v0 ?v1 ?v2 ?v3 where
{
?v0 <http :// www.w3.org/2000/01/rdf−schema#label> ?v1 .
?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> ?v2 .
?v3 <http :// www.w3.org/2000/01/rdf−schema#label> ?v1 .
?v3 <http :// purl . org/dc/elements /1.1/ publisher > <http :// www4.wiwiss
          .fu−berlin.de/ bizer /bsbm/v01/ instances /
            StandardizationInstitution1  > .
}
```

6.4.7 q6.sql

```
select  ?v0 ?v1 ?v2 ?v3 ?v4 where
{
?v0 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> ?v1 .
?v0 <http :// purl . org/dc/elements /1.1/ date> ?v2  .
?v3 <http :// www.w3.org/2000/01/rdf−schema#comment> ?v4 .
<http :// www4.wiwiss.fu−berlin.de/ bizer /bsbm/v01/ instances /
          ProductType2> <http :// www.w3.org/2000/01/rdf−schema#
          comment> ?v4 .
?v3 <http :// www.w3.org/1999/02/22−rdf−syntax−ns#type> ?v1 .
}
```

6.4.8 q7.sql

```
select ?v0 ?v1 ?v2 ?v3 ?v4 where
{
?v0 <http://purl.org/dc/elements/1.1/title> ?v1 .
?v0 <http://purl.org/dc/elements/1.1/publisher> <http://www4.wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/dataFromRatingSite11/RatingSite11> .
?v0 <http://purl.org/stuff/rev#text> ?v4 .
?v2 <http://purl.org/stuff/rev#reviewer> ?v3 .
?v2 <http://purl.org/dc/elements/1.1/date> "2008-04-16"^^<http://www.w3.org/2001/XMLSchema#date> .
?v2 <http://purl.org/stuff/rev#text> ?v4
}
```

6.4.9 q8.sql

```
select ?v0 ?v1 ?v2 ?v3 ?v4 ?v5 ?v6 where
{
?v0 <http://www.w3.org/2000/01/rdf-schema#subClassOf> ?v1 .
?v0 <http://purl.org/dc/elements/1.1/date> ?v2 .
?v3 <http://purl.org/dc/elements/1.1/publisher> ?v4 .
?v3 <http://www.w3.org/2000/01/rdf-schema#subClassOf> ?v1 .
?v5 <http://purl.org/dc/elements/1.1/publisher> ?v6 .
?v5 <http://purl.org/dc/elements/1.1/date> ?v2 .
}
```

6.4.10 q9.sql

```
select ?v0 ?v1 ?v2 ?v3 ?v4 ?v5 ?v6 ?v7 ?v8 where
{
```

```
?v0 <http :// purl . org/dc/elements /1.1/ date> "2000-07-17"^^<http://
            www.w3.org/2001/XMLSchema#date> .
?v0 <http :// purl . org/dc/elements /1.1/ publisher > ?v1 .
?v0 <http :// www.w3.org/2000/01/rdf-schema#label> ?v2 .
?v0 <http :// www.w3.org/2000/01/rdf-schema#comment> ?v8 .
?v3 <http :// www.w3.org/2000/01/rdf-schema#subClassOf> <http://www4
            .wiwiss.fu-berlin.de/bizer/bsbm/v01/instances/ProductType2> .
?v3 <http :// purl . org/dc/elements /1.1/ publisher > ?v1 .
?v3 <http :// www.w3.org/2000/01/rdf-schema#label> ?v4 .
?v3 <http :// www.w3.org/1999/02/22-rdf-syntax-ns#type> ?v7 .
?v5 <http :// www.w3.org/2000/01/rdf-schema#label> ?v2 .
?v5 <http :// purl . org/dc/elements /1.1/ publisher > ?v6 .
?v5 <http :// www.w3.org/1999/02/22-rdf-syntax-ns#type> ?v7 .
}
```

References

[1] M. Atre, V. Chaoji, M. J. Zaki, and J. A. Hendler. Matrix "bit" loaded: a scalable lightweight join query processor for rdf data. In *International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, Usa, April*, pages 41–50, 2010.

[2] Y. Guo, Z. Pan, and J. Heflin. Lubm: A benchmark for owl knowledge base systems. *Web Semantics Science Services and Agents on the World Wide Web*, 3(23):158–182, 2005.

[3] P. Peng, L. Zou, M. T. Özsu, L. Chen, and D. Zhao. Processing SPARQL queries over distributed RDF graphs. *VLDB J.*, 2016.

[4] L. Zou, M. T. Zsu, L. Chen, X. Shen, R. Huang, and D. Zhao. gstore: a graph-based sparql query engine. *Vldb Journal International Journal on Very Large Data Bases*, 23(4):565–590, 2014.