Challenges to be Addressed During Benchmarking SPARQL Federated Engines

Maribel Acosta
Karlsruhe Institute of Technology

Maria-Esther Vidal
Universidad Simón Bolívar

Agenda

1 Parameters that impact on performance of federated query engines

2 Benchmarking SPARQL federated query engines:
A use case

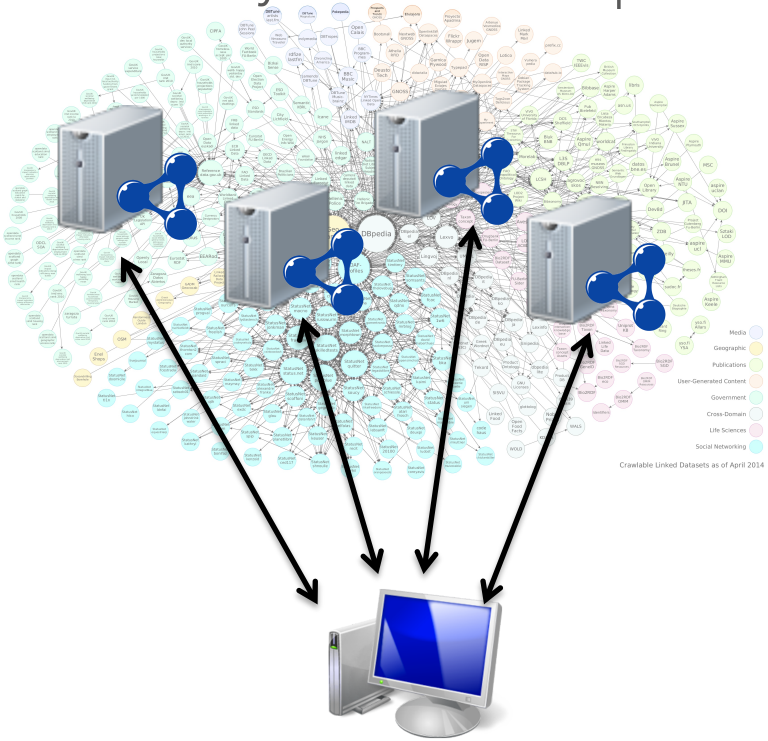
3 Formalization of the query decomposition problem

4 Lessons learned

1 PARAMETERS THAT IMPACT ON PERFORMANCE OF FEDERATED QUERY ENGINES

Federated Querying Processing

Publicly available endpoints



Federated
query processing

Current SPARQL Federated Engines

FedX
Linked Data in
a Federation

ARQ

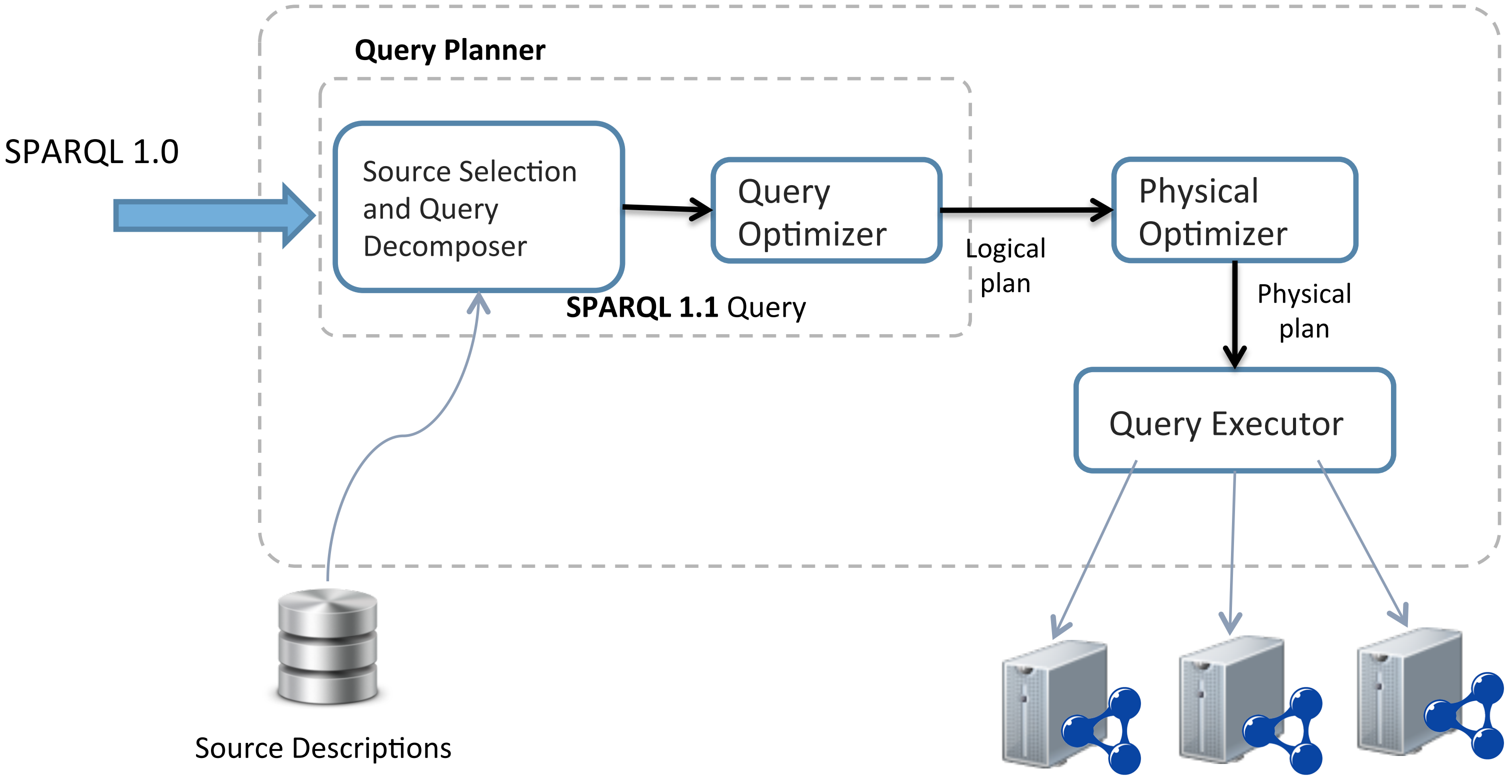
ANAPSID

SPARQL-DQP

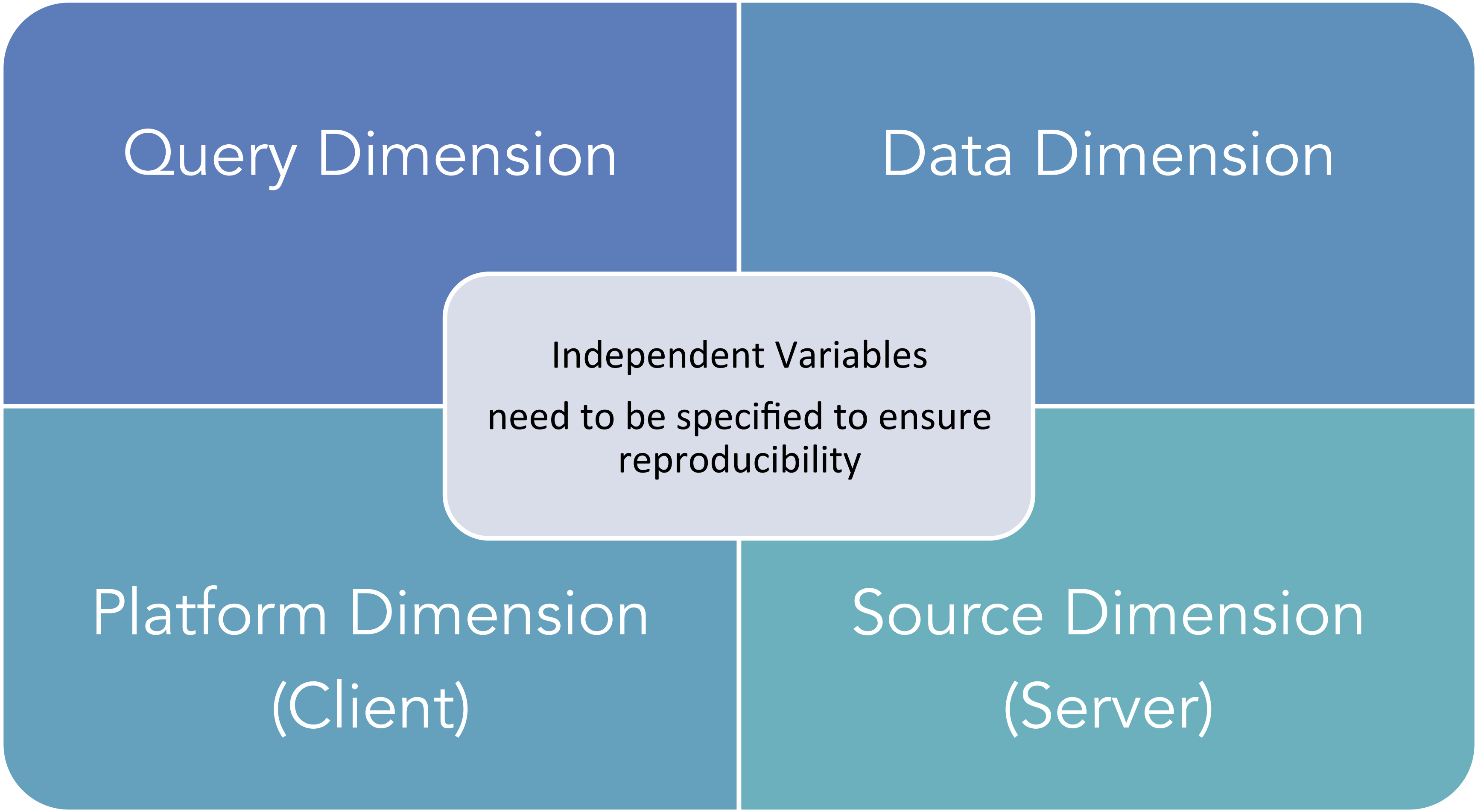
SPLENDID

avalanche

Federated Engines: Architecture



Benchmarking Dimensions



Query Dimension

Data Dimension

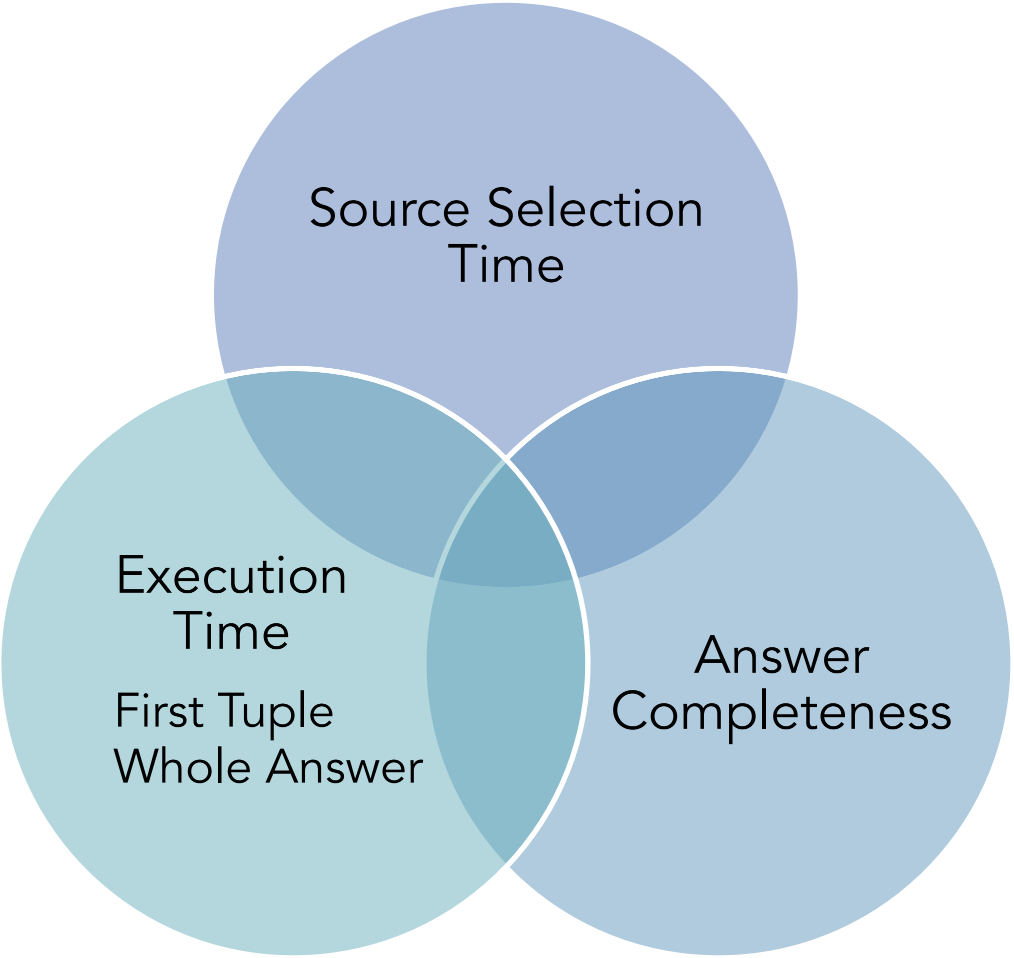
Independent Variables

need to be specified to ensure reproducibility

Platform Dimension (Client)

Source Dimension (Server)

Dependent/Observable Variables



Source Selection Time

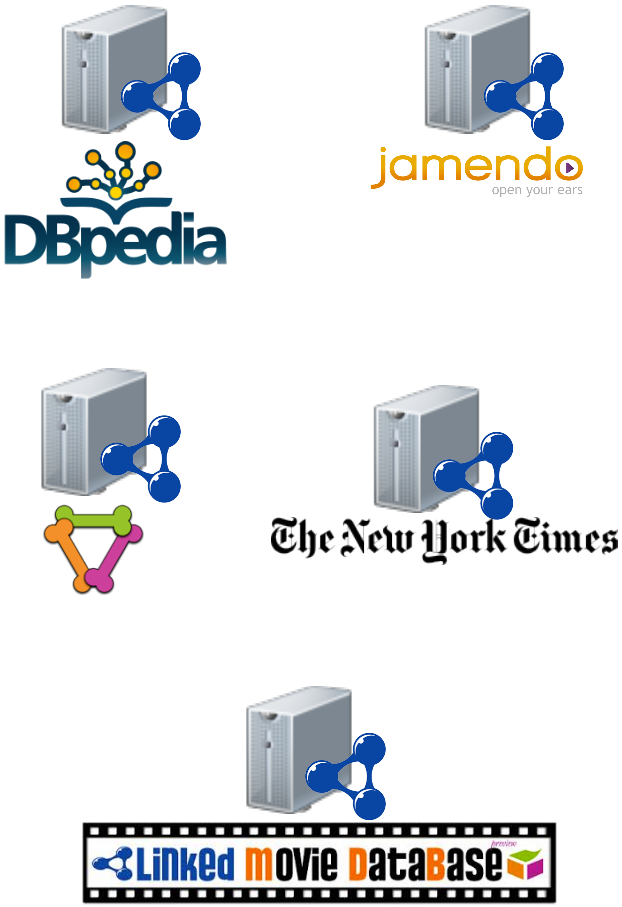
Execution Time

First Tuple
Whole Answer

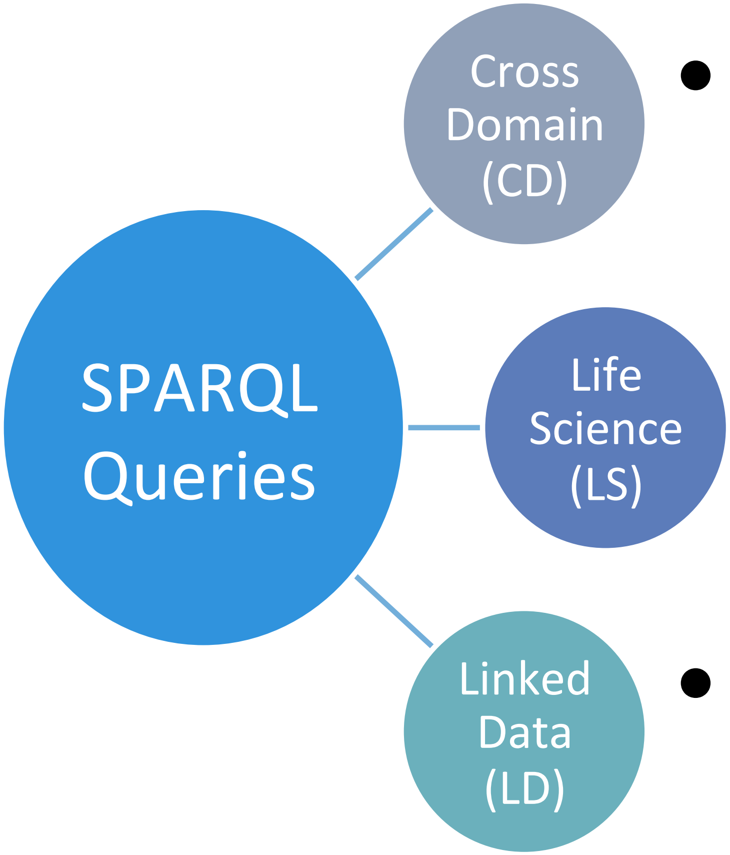
Answer Completeness

FedBench Benchmark (1)

	Collections
Cross Domain	DBpedia subset
	NY Times
	LinkedMDB
	Jamendo
	GeoNames
	SW Dog Food
Life Science	SP2Bench 10M
	KEGG
	Drugbank
	ChEBI



FedBench Benchmark (2)



SPARQL Queries

Cross Domain (CD)

- Seven Queries

Life Science (LS)

- Seven Queries

Linked Data (LD)

- Eleven Queries

Query Dimension

Query Domain	Source Selection Time	Execution Time	Answer Completeness
Query Plan Shape	✔	✔	✔
#Triple Patterns	✔	✔	✔
#Instantiations and their position		✔	
Join Selectivity		✔	
#Intermediate results		✔	
Answer size		✔	
Usage of query language expressivity	✔	✔	
#General properties	✔	✔	✔

Query Dimension: Example

“Kegg compound identifiers and among their drugs, those that have a substrate that is an enzyme.”

```
q0: Select * WHERE
{?d drugbank:keggCompoundId ?c.
 ?e bio2rdf-kegg:xSubstrate ?c. }
 ?e rdf:type bio2rdf-kegg:Enzyme}
```

Triple Bound to General Predicate

Query Dimension: Example

"Kegg compound identifiers and among their drugs, those that have a substrate that is an enzyme."

```
q0: Select * WHERE
{?d drugbank:keggCompoundId ?c.
 ?e bio2rdf-kegg:xSubstrate ?c. }
 ?e rdf:type bio2rdf-kegg:Enzyme}
```

Triple Bound to General Predicate

```
q1: Select * WHERE
{?d drugbank:keggCompoundId ?c.
 ?e bio2rdf-kegg:xSubstrate ?c. }
 ?e rdf:type bio2rdf-kegg:Enzyme.
 ?d owl:sameAs ?d1}
```

Triples Bound to General Predicate

Query Dimension: Example

"Kegg compound identifiers and among their drugs, those that have a substrate that is an enzyme."

```
q0: Select * WHERE
{?d drugbank:keggCompoundId ?c.
 ?e bio2rdf-kegg:xSubstrate ?c. }
 ?e rdf:type bio2rdf-kegg:Enzyme}
```

Triple Bound to General Predicate

```
q1: Select * WHERE
{?d drugbank:keggCompoundId ?c.
 ?e bio2rdf-kegg:xSubstrate ?c. }
 ?e rdf:type bio2rdf-kegg:Enzyme.
 ?d owl:sameAs ?d1}
```

Triples Bound to General Predicate

```
q2: Select * WHERE
{?d drugbank:keggCompoundId ?c.
 ?e bio2rdf-kegg:xSubstrate ?c. }
 ?e rdf:type bio2rdf-kegg:Enzyme.
 ?d owl:sameAs ?d1.
 ?d1 rdf:type dbpedia-owl:Drug}
```

Triples Bound to General Predicate

Query Dimension: Example

“Kegg compound identifiers and among their drugs, those that have a substrate that is an enzyme.”

```
q0: Select * WHERE
{?d drugbank:keggCompoundId ?c.
 ?e bio2rdf-kegg:xSubstrate ?c. }
?e rdf:type bio2rdf-kegg:Enzyme}
```

Triple Bound to General Predicate

```
q1: Select * WHERE
{?d drugbank:keggCompoundId ?c.
 ?e bio2rdf-kegg:xSubstrate ?c. }
?e rdf:type bio2rdf-kegg:Enzyme.
?d owl:sameAs ?d1}
```

Triples Bound to General Predicate

```
q2: Select * WHERE
{?d drugbank:keggCompoundId ?c.
 ?e bio2rdf-kegg:xSubstrate ?c. }
 ?e rdf:type bio2rdf-kegg:Enzyme.
 ?d owl:sameAs ?d1.
 ?d1 rdf:type dbpedia-owl:Drug}
```

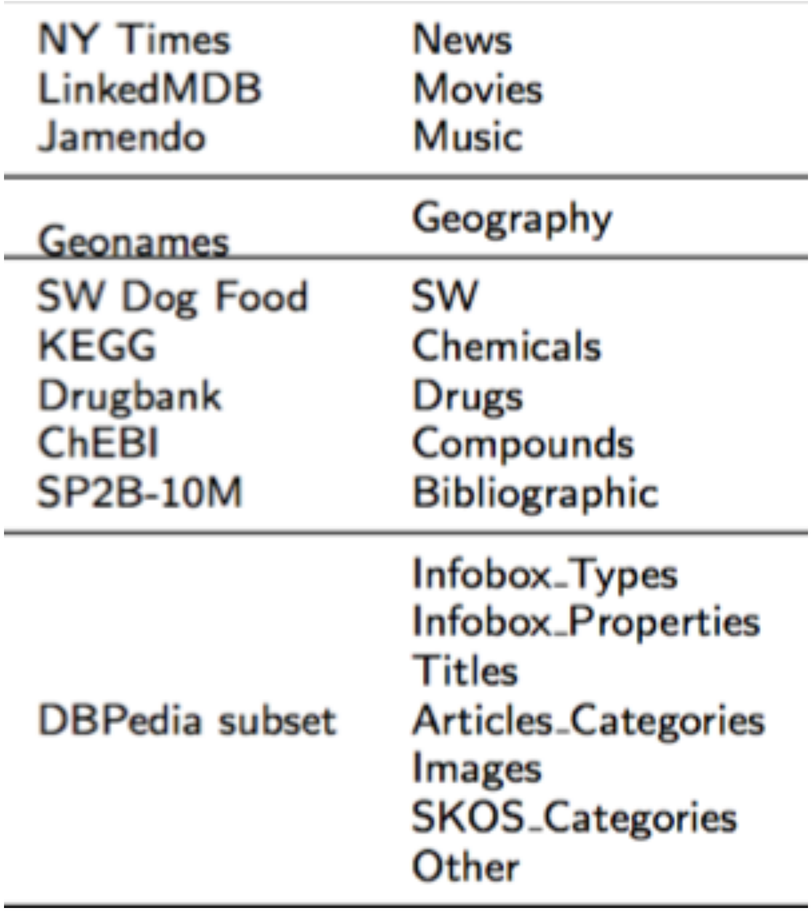
Triples Bound to General Predicate

```
q3: Select * WHERE
{?d drugbank:keggCompoundId ?c. ?e bio2rdf-kegg:xSubstrate ?c. }
 ?e rdf:type bio2rdf-kegg:Enzyme.
 ?d owl:sameAs ?d1.?d1 rdf:type dbpedia-owl:Drug.
 ?d1 rdf:type ?t1. ?d1 rdfs:label ?d2}
```

Triples Bound to General Predicate

Query Dimension: Example

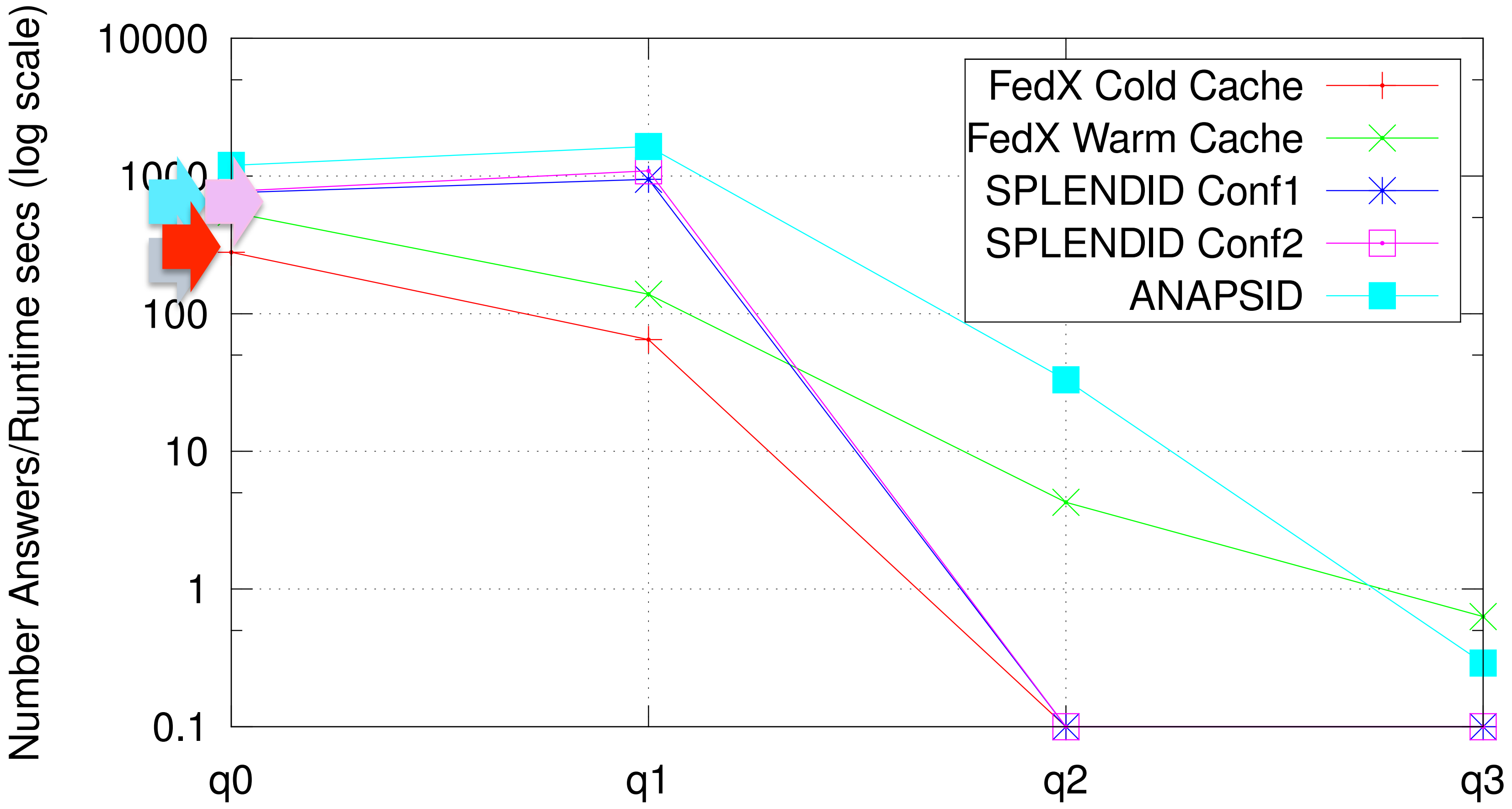
26 Local SPARQL Endpoints



NY Times	News
LinkedMDB	Movies
Jamendo	Music
Geonames	Geography
SW Dog Food	SW
KEGG	Chemicals
Drugbank	Drugs
ChEBI	Compounds
SP2B-10M	Bibliographic
DBPedia subset	Infobox_Types
	Infobox_Properties
	Titles
	Articles_Categories
	Images
	SKOS_Categories
	Other

- **26 Virtuoso endpoints**
  - Timeout set up to 240 secs, or 71,000 tuples.
- **Metrics**
  - Throughput
- **Federated Engines**
  - FedX [Schwarte et al 2011]
  - SPLENDID [Gorlitz and Staab 2011]
  - ANAPSID [Acosta et al. 2011]
- **Experimental Environment**
  - Linux Mint machine. Intel Pentium Core2 Duo 3.0 GHz.
  - 8GB RAM.

Federated Engine Performance

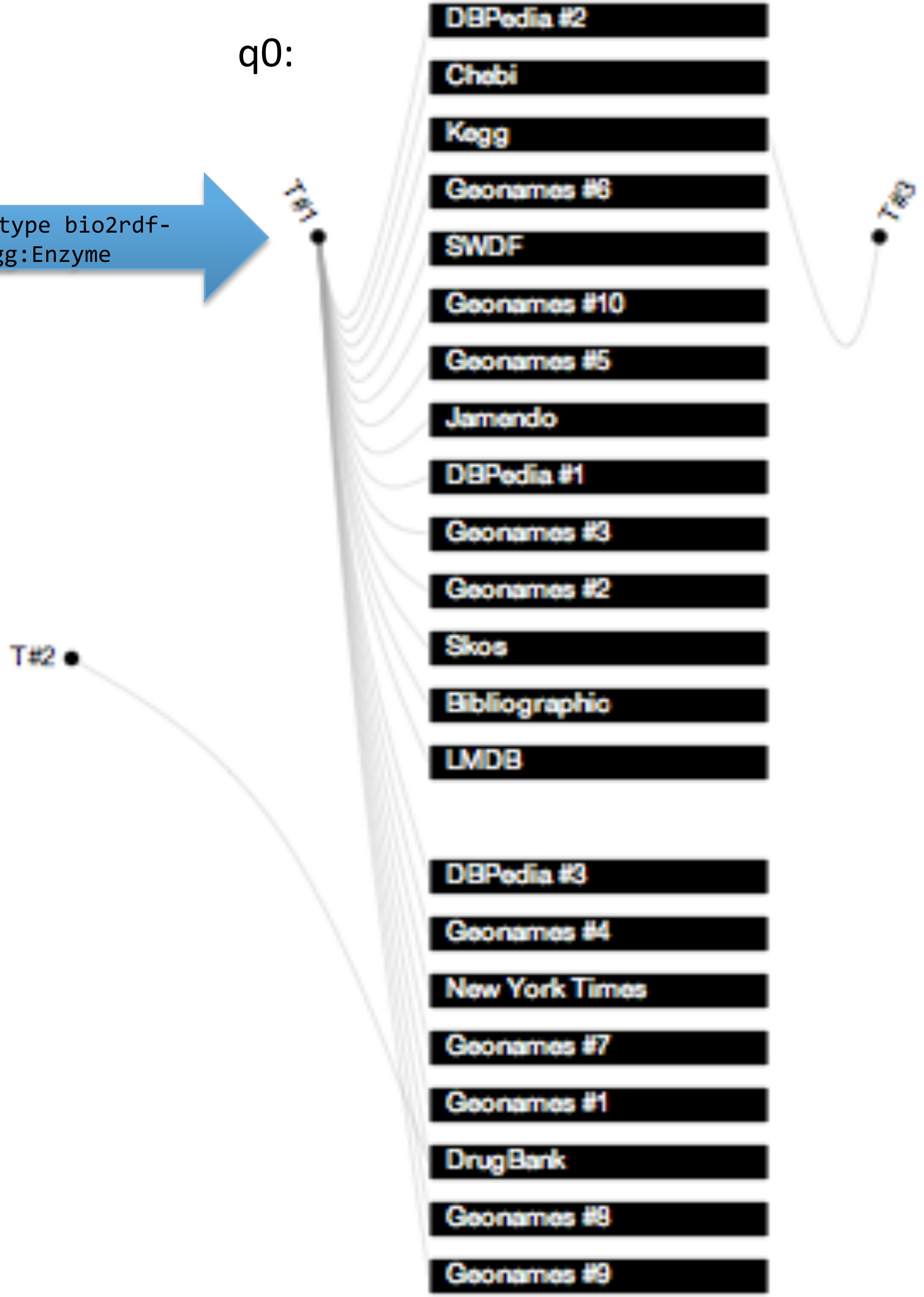


Queries

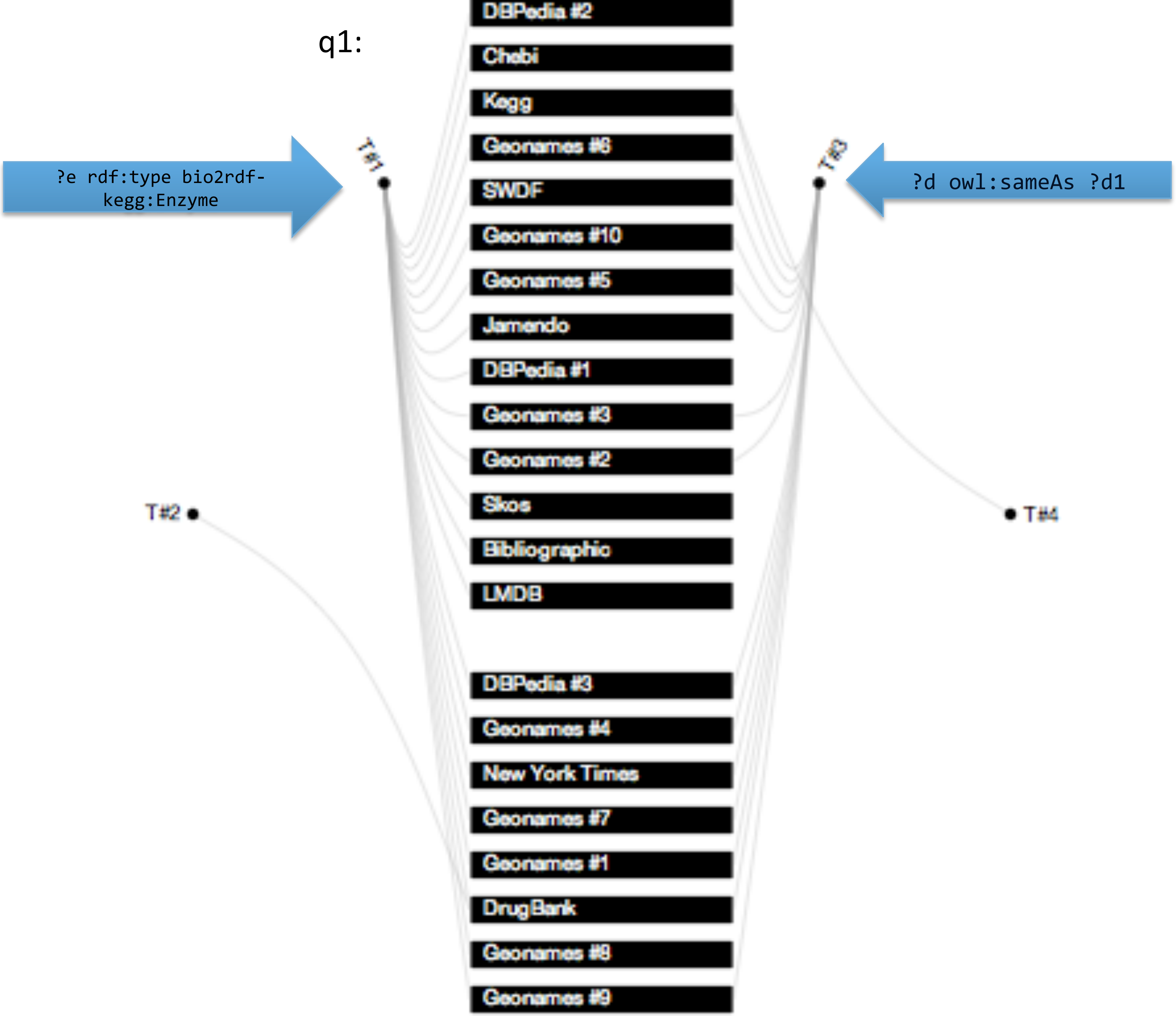
(Query Complexity)

q0:

?e rdf:type bio2rdf-kegg:Enzyme



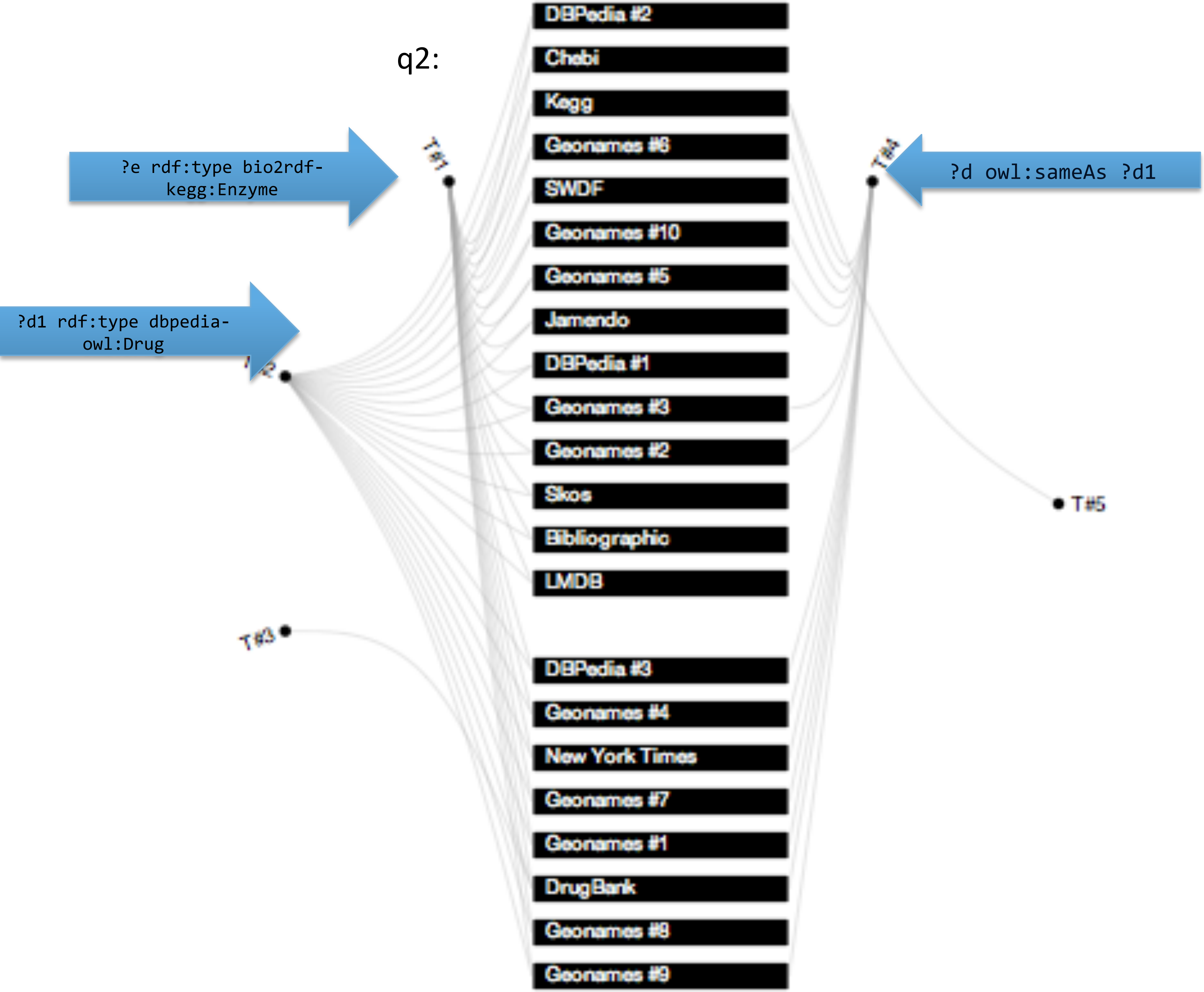
q1:



?e rdf:type bio2rdf-kegg:Enzyme

?d owl:sameAs ?d1

q2:

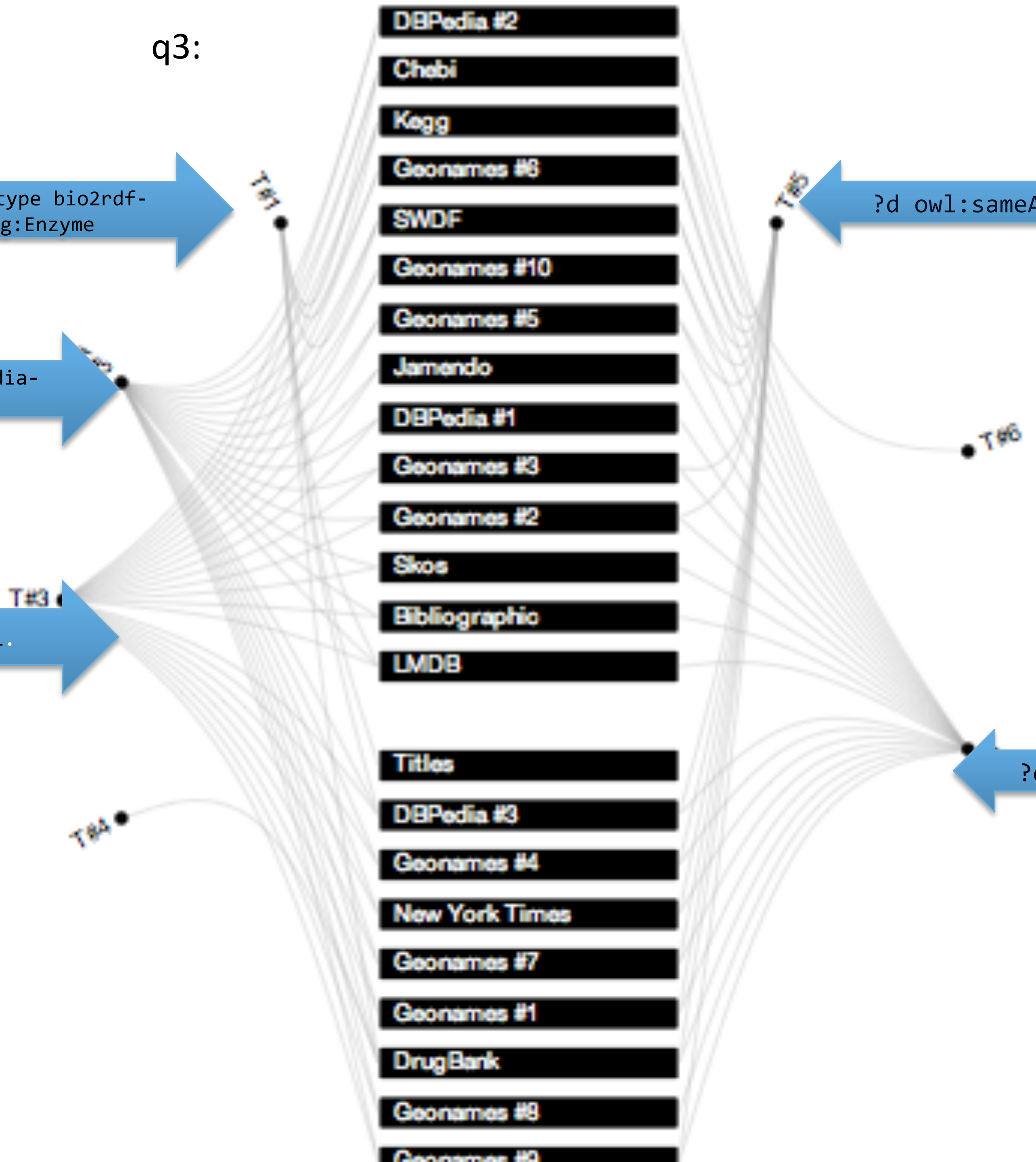


?e rdf:type bio2rdf-kegg:Enzyme

?d1 rdf:type dbpedia-owl:Drug

?d owl:sameAs ?d1

q3:



?e rdf:type bio2rdf-kegg:Enzyme

?d1 rdf:type dbpedia-owl:Drug

?d1 rdf:type ?t1.

?d owl:sameAs ?d1

?d1 rdfs:label ?d2

Data Dimension

Data Domain	Source Selection Time	Execution Time	Answer Completeness
Dataset size		✔	
Data frequency distribution		✔	
Type of partitioning	✔	✔	✔
Data Endpoint Distribution	✔	✔	✔

Data Fragmentation

Fragment 1

Fragment 2

Fragment 3

Empty Intersection between fragments

Data Fragmentation: Horizontal

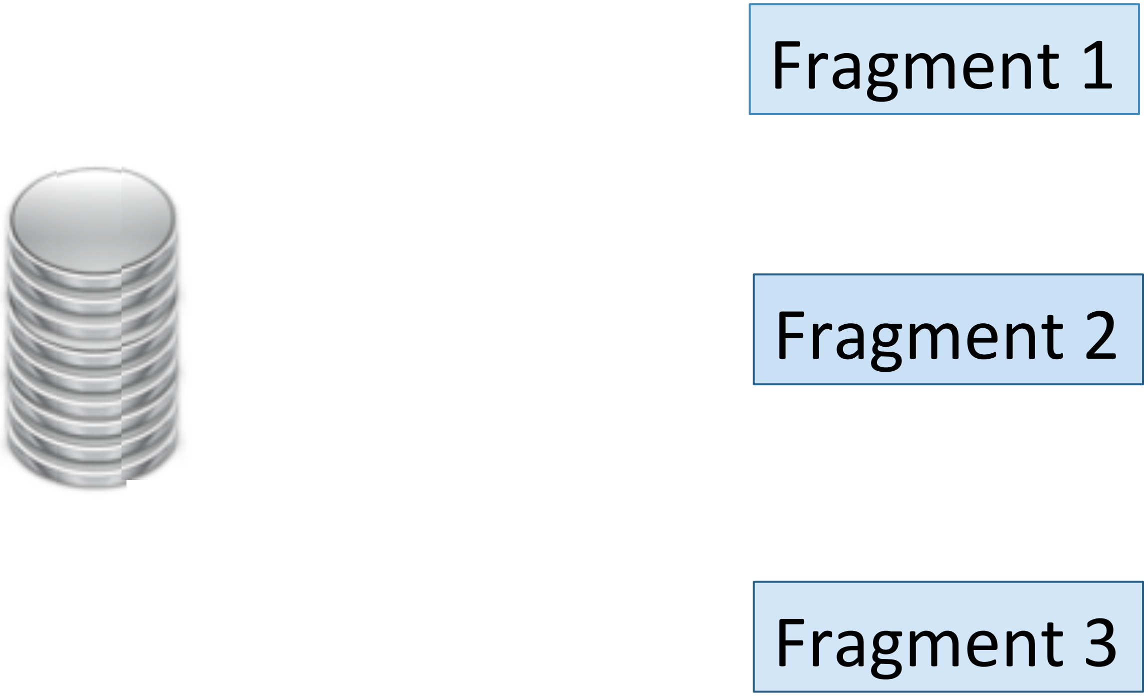
Fragment 1

Fragment 2

Fragment 3

- Each fragment may contain triples of many predicates.
- Triples of one predicate can belong to different fragments.
- May impact on the completeness of a query answer.

Data Fragmentation: Vertical



Fragment 1

Fragment 2

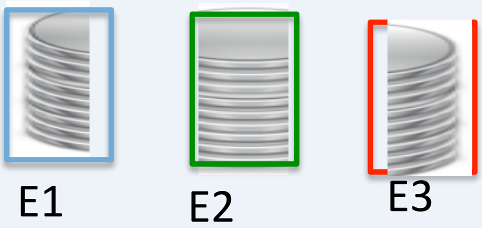
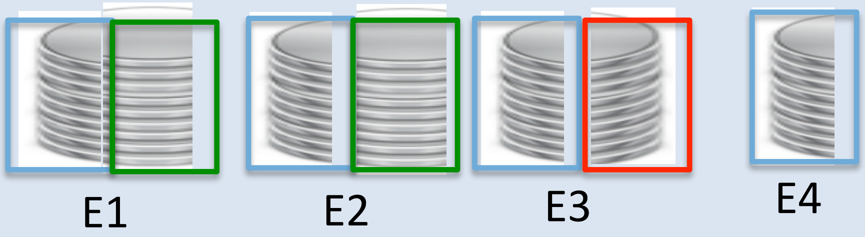
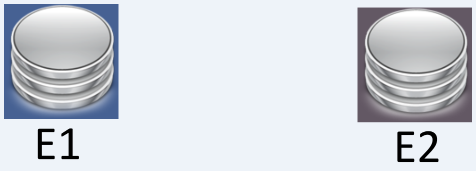
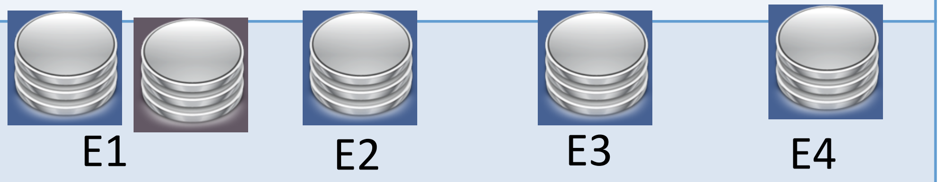
Fragment 3

- Each fragment contains all the triples of at least one predicate in the original dataset.
- May impact on query performance.

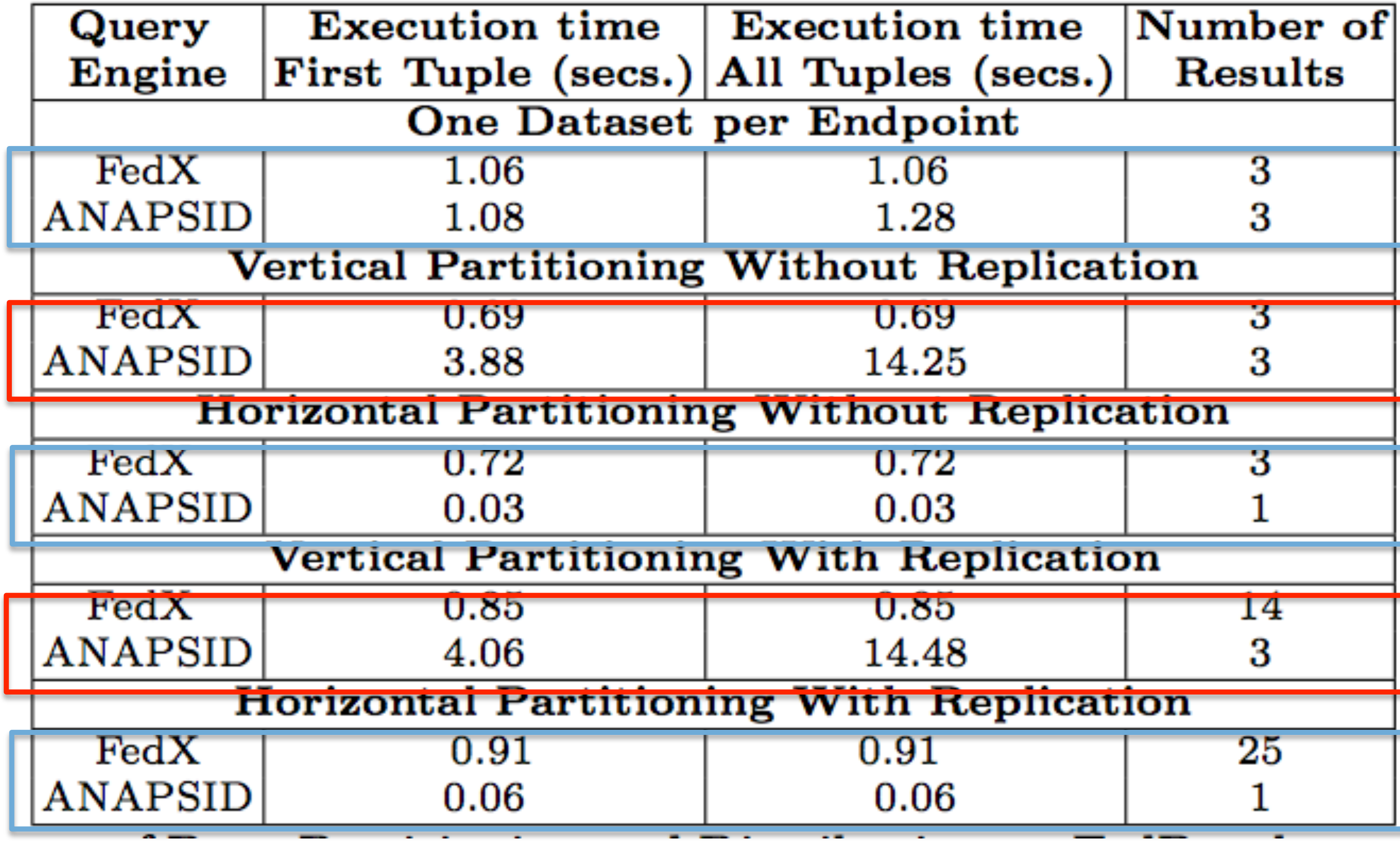
Data Fragmentation & Replication Effects

Triples of three predicates were stored in fragments.

`skos:subject, owl:sameAs, nytimes:latest_use`

Vertical Partitioning Without Replication (Three fragments)	
Vertical Partitioning With Replication (Three fragments)	
Horizontal Partitioning Without Replication (Two fragments)	
Horizontal Partitioning With Replication (Two fragments)	

Data Fragmentation & Replication Effects (FedBench LD10)

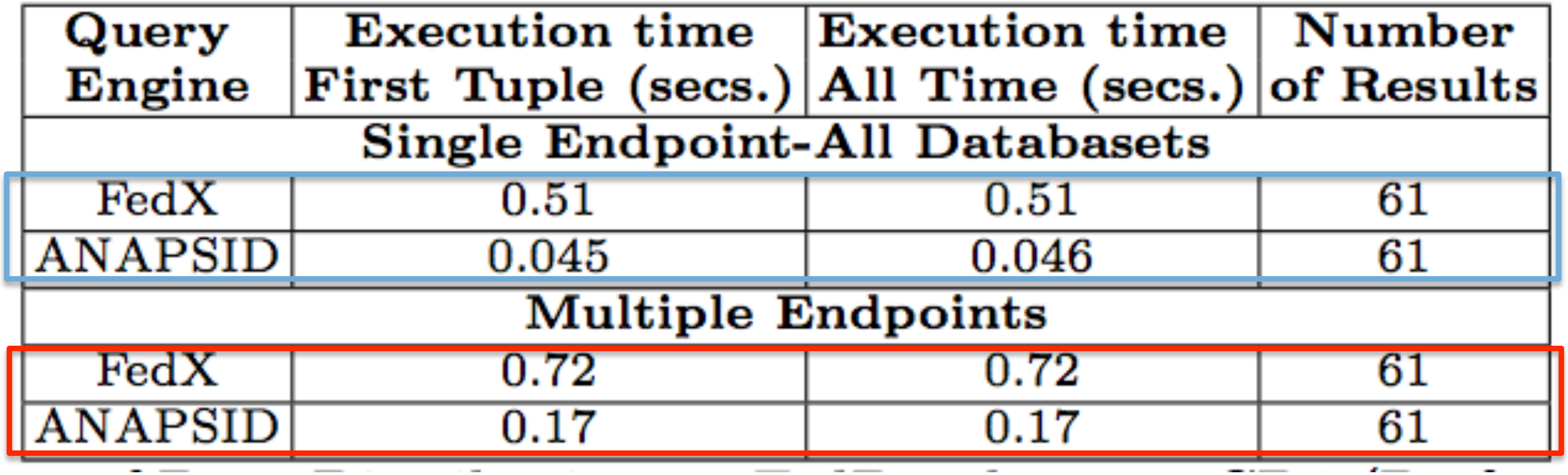


	Query Engine	Execution time First Tuple (secs.)	Execution time All Tuples (secs.)	Number of Results
	One Dataset per Endpoint			
✓	FedX	1.06	1.06	3
	ANAPSID	1.08	1.28	3
	Vertical Partitioning Without Replication			
✗	FedX	0.69	0.69	3
	ANAPSID	3.88	14.25	3
	Horizontal Partitioning Without Replication			
✓	FedX	0.72	0.72	3
	ANAPSID	0.03	0.03	1
	Vertical Partitioning With Replication			
✗	FedX	0.85	0.85	14
	ANAPSID	4.06	14.48	3
	Horizontal Partitioning With Replication			
✓	FedX	0.91	0.91	25
	ANAPSID	0.06	0.06	1

Data Distribution

- All FedBench datasets distributed in one endpoint versus distributed in different endpoints.

FedBench CD1 (Perfect Network)



	Query Engine	Execution time First Tuple (secs.)	Execution time All Time (secs.)	Number of Results
	Single Endpoint-All Databasets			
✓	FedX	0.51	0.51	61
	ANAPSID	0.045	0.046	61
	Multiple Endpoints			
✖	FedX	0.72	0.72	61
	ANAPSID	0.17	0.17	61

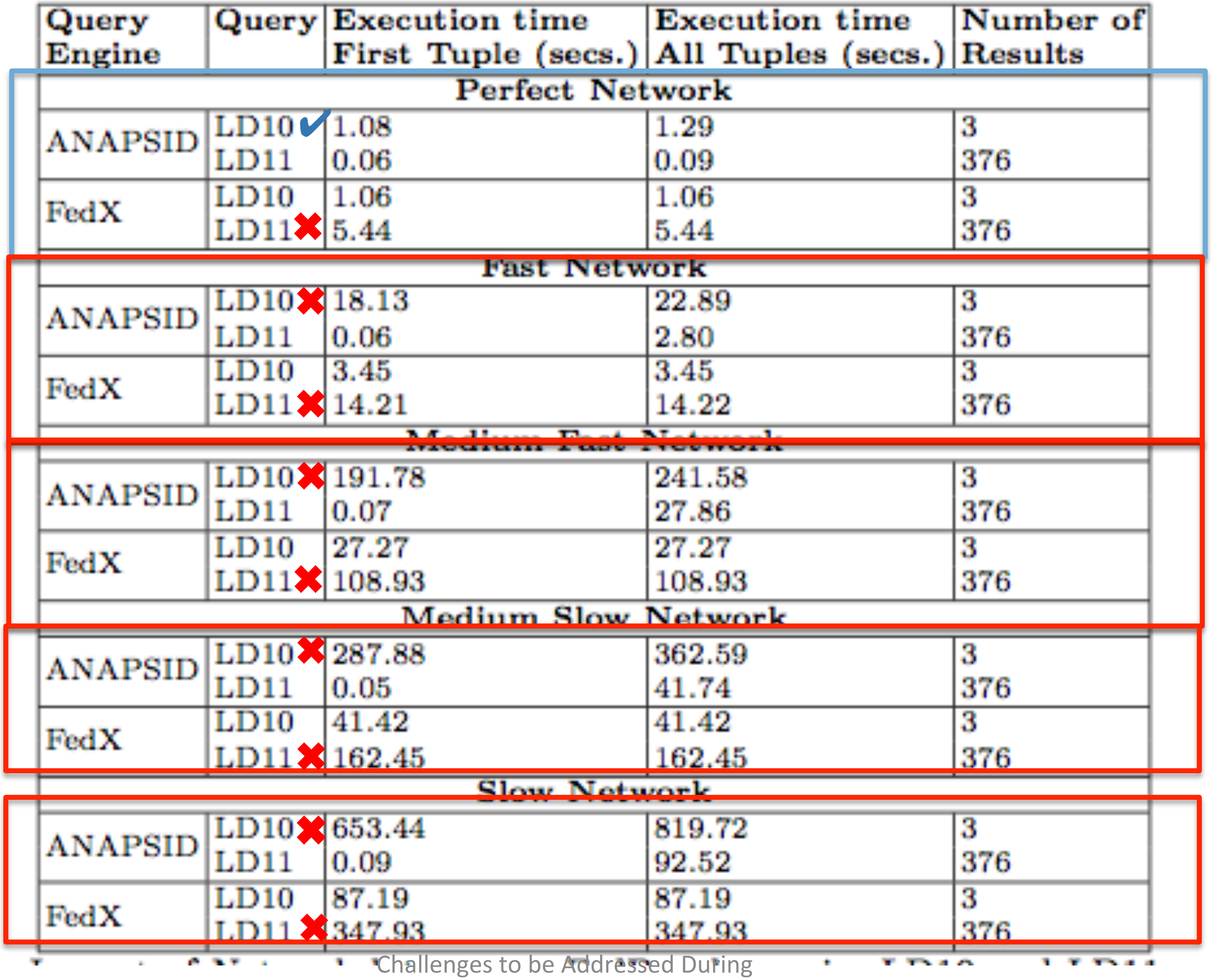
Platform Dimension (Client)

Platform Domain	Source Selection Time	Execution Time	Answer Completeness
Cache on/off	✔	✔	
RAM available	✔	✔	
#Processors	✔	✔	

Source Dimension (Server)

Endpoint Domain	Source Selection Time	Execution Time	Answer Completeness
#Endpoints	✔	✔	✔
Endpoint Type	✔	✔	
Relation Graph/ Endpoint		✔	✔
Network Latency	✔	✔	✔
Initial Delay	✔	✔	
Message size		✔	✔
Transfer Distribution	✔	✔	✔
Answer Size Limit Timeout		✔	✔

Source Dimension



Query Engine	Query	Execution time First Tuple (secs.)	Execution time All Tuples (secs.)	Number of Results
		Perfect Network		
ANAPSID	LD10 ✓	1.08	1.29	3
	LD11	0.06	0.09	376
FedX	LD10	1.06	1.06	3
	LD11 ✗	5.44	5.44	376
		Fast Network		
ANAPSID	LD10 ✗	18.13	22.89	3
	LD11	0.06	2.80	376
FedX	LD10	3.45	3.45	3
	LD11 ✗	14.21	14.22	376
		Medium Fast Network		
ANAPSID	LD10 ✗	191.78	241.58	3
	LD11	0.07	27.86	376
FedX	LD10	27.27	27.27	3
	LD11 ✗	108.93	108.93	376
		Medium Slow Network		
ANAPSID	LD10 ✗	287.88	362.59	3
	LD11	0.05	41.74	376
FedX	LD10	41.42	41.42	3
	LD11 ✗	162.45	162.45	376
		Slow Network		
ANAPSID	LD10 ✗	653.44	819.72	3
	LD11	0.09	92.52	376
FedX	LD10	87.19	87.19	3
	LD11 ✗	347.93	347.93	376

2 BENCHMARKING SPARQL FEDERATED ENGINES: A USE CASE

Experimental Study

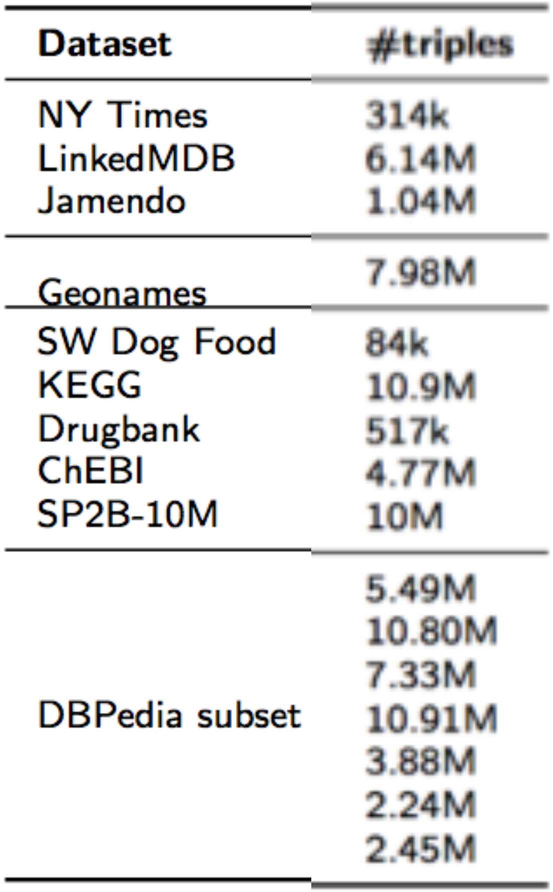
- **Queries**
  - 25 FedBech queries
  - Eleven additional complex queries*
    - Between 6 and 48 triple patterns.
    - Decomposed into up to 8 sub-queries.
    - Different SPARQL Operators.
    - General Predicates: rdf:type, owl:sameAs, rdfs:seeAlso
- **Virtuoso 6.1 Endpoints**
  - Timeouts 240 secs.
  - 71,000 tuples

- **Timeout 1,800 secs**
- **Experimental Environment**
  - Linux Mint machine.
  - Intel Pentium Core2 Duo 3.0 GHz.
  - 8GB RAM.
  - 133MHz DDR3
- **Federated SPARQL Engines**
  - ANAPSID 2.0 [Acosta et al 2011]
  - FedX 2.0 [Schwarte et al 2011]
  - ARQ
  - Virtuoso 6.1 SPARQL endpoints

* http://www.ldc.usb.ve/~mvidal/FedBench/queries/ComplexQueries

Experimental Study

- Network Latency
  - Perfect Network
  - Message 16KB
- Metrics
  - Total Execution Time*
  - Spearman's Rho correlation.
  - *Pyhton time.time()
- Data Distribution
  - Complete:
    - FedBench collections were stored into a single graph, one endpoint.
  - Federated:
    - Fedbench collections were stored in ten Virtuoso endpoints
  - Data collections were downloaded on December 2011.
- Each query was run 10 times and average is reported.



Dataset	#triples
NY Times	314k
LinkedMDB	6.14M
Jamendo	1.04M
Geonames	7.98M
SW Dog Food	84k
KEGG	10.9M
Drugbank	517k
ChEBI	4.77M
SP2B-10M	10M
DBPedia subset	5.49M
	10.80M
	7.33M
	10.91M
	3.88M
	2.24M
	2.45M

Experimental Protocol (1)

- Configuration 1: ANAPSID Complete Distribution
- Configuration 2: ANAPSID Federated Distribution
- Configuration 3: ARQ Complete Distribution
- Configuration 4: ARQ Federated Distribution
- Configuration 5: FedX Complete Distribution
- Configuration 6: FedX Federated Distribution

Experimental Protocol (2)

CD1: 0.78

CD2: 0.37

CD3: 20.92

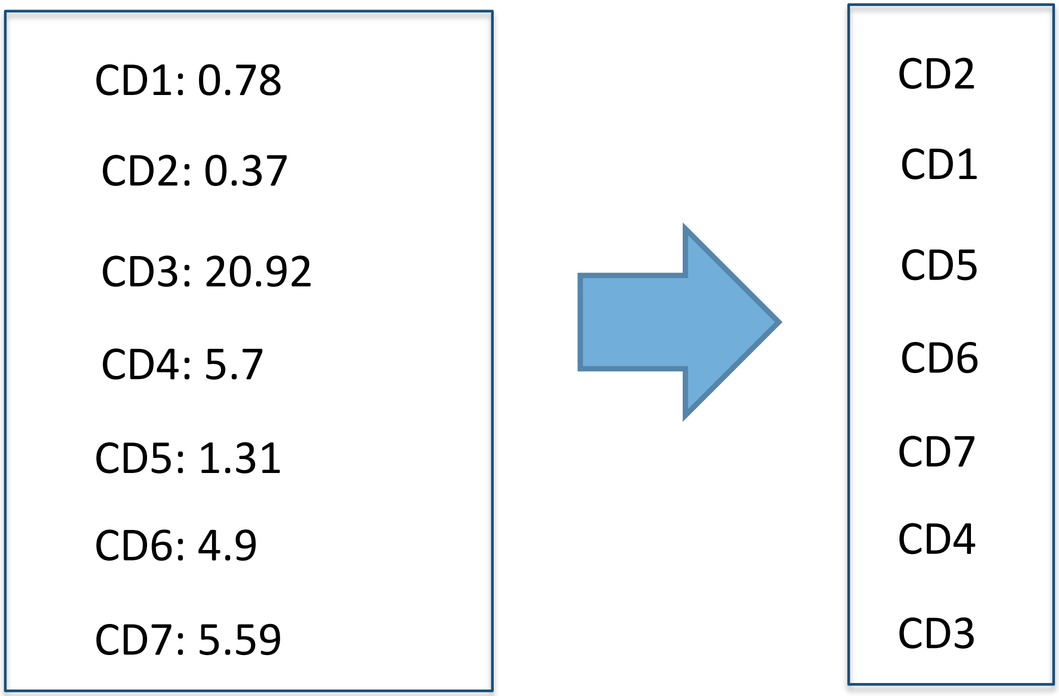
CD4: 5.7

CD5: 1.31

CD6: 4.9

CD7: 5.59

Experimental Protocol (2)



CD1: 0.78

CD2: 0.37

CD3: 20.92

CD4: 5.7

CD5: 1.31

CD6: 4.9

CD7: 5.59

CD2

CD1

CD5

CD6

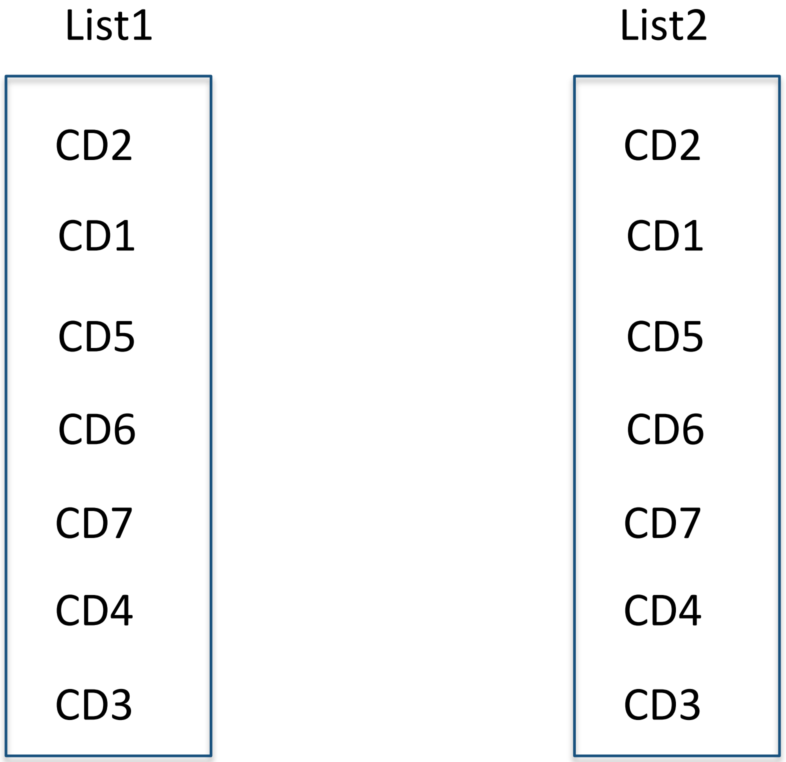
CD7

CD4

CD3

For each configuration, a list of queries is created in ascending order according to execution time

Experimental Protocol (3)



List1	List2
CD2	CD2
CD1	CD1
CD5	CD5
CD6	CD6
CD7	CD7
CD4	CD4
CD3	CD3

Spearman's Correlation: 1

List1 perfect monotone function of List2

Experimental Protocol (3)

List1	List2
CD2	CD3
CD1	CD4
CD5	CD7
CD6	CD6
CD7	CD5
CD4	CD1
CD3	CD2

Spearman's Correlation: -1

List1 perfect monotone function of List2

List1 increases

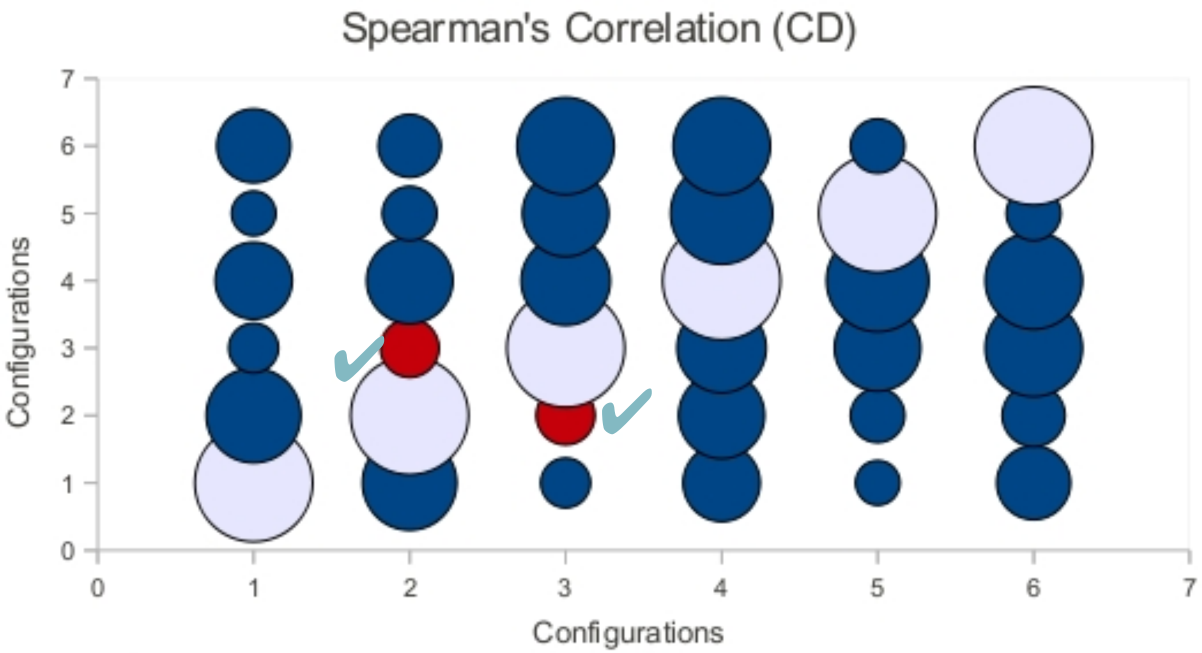
List2 decreases

Experimental Protocol (4)

List1	List2
CD2	CD1
CD1	CD2
CD5	CD4
CD6	CD3
CD7	CD5
CD4	CD6
CD3	CD7

Spearman’s Correlation: 0.96

Experimental Evaluation: Federated Engine Trend – CD



Configuration 1: ANAPSID Complete Distribution

Configuration 2: ANAPSID Federated Distribution

Configuration 3: ARQ Complete Distribution

Configuration 4: ARQ Federated Distribution

Configuration 5: FedX Complete Distribution

Configuration 6: FedX Federated Distribution

○ Correlation 1.0

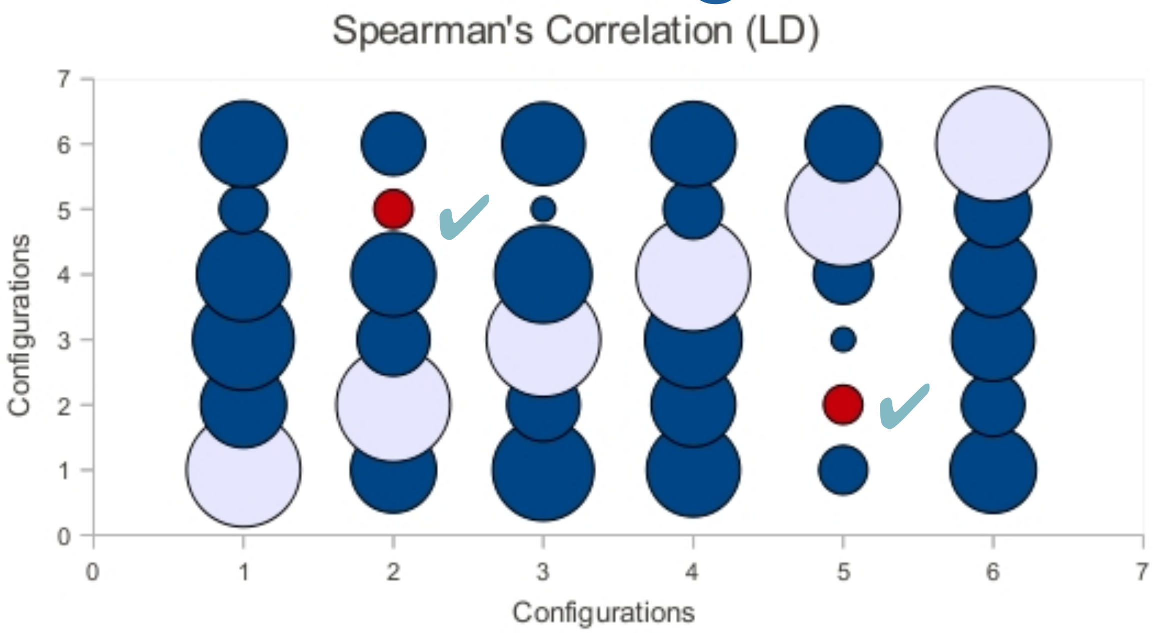
● Positive Correlation

● Negative Correlation

ANAPSID Federated Distribution and ARQ Complete Distribution exhibit opposite performance

ANAPSID and FedX Federated Distribution exhibit positive correlation

Experimental Evaluation: Federated Engine Trend – LD



Configuration 1: ANAPSID Complete Distribution
Configuration 2: ANAPSID Federated Distribution
Configuration 3: ARQ Complete Distribution
Configuration 4: ARQ Federated Distribution
Configuration 5: FedX Complete Distribution
Configuration 6: FedX Federated Distribution

○ Correlation 1.0

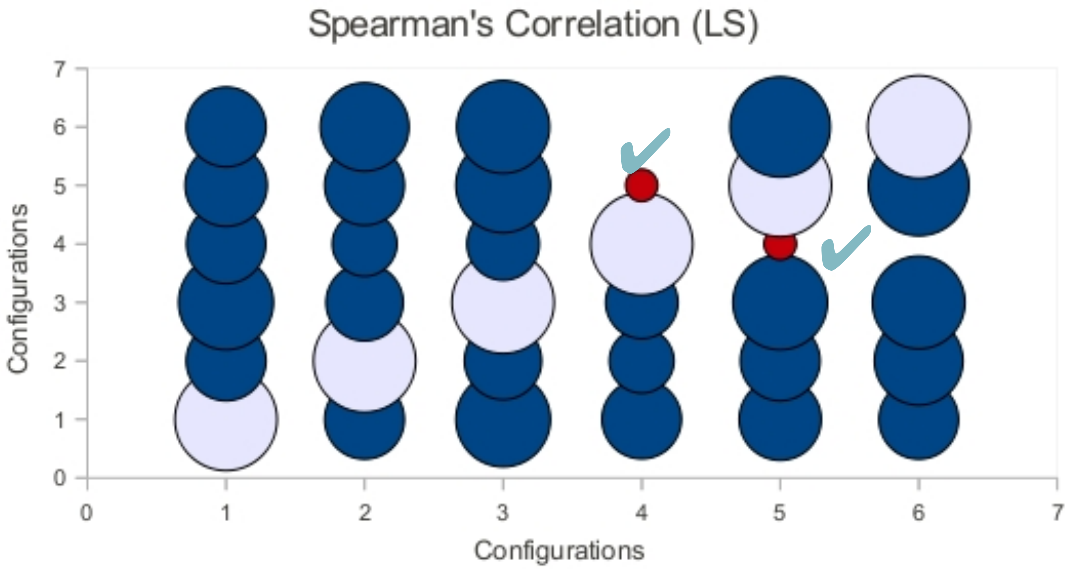
● Positive Correlation

🔴 Negative Correlation

ANAPSID Federated Distribution and FedX Complete Distribution exhibit opposite performance

ANAPSID and FedX Federated Distribution exhibit positive correlation

Experimental Evaluation: Federated Engine Trend – LS



○ Correlation 1.0

● Positive Correlation

● Negative Correlation

Configuration 1: ANAPSID Complete Distribution

Configuration 2: ANAPSID Federated Distribution

Configuration 3: ARQ Complete Distribution

Configuration 4: ARQ Federated Distribution

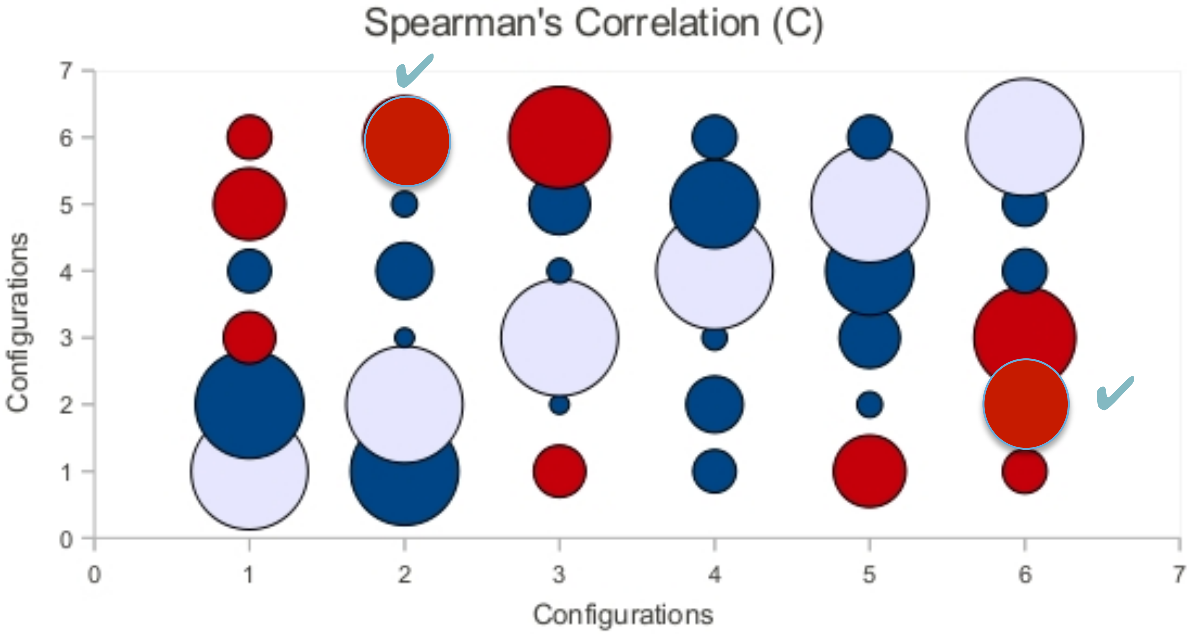
Configuration 5: FedX Complete Distribution

Configuration 6: FedX Federated Distribution

ARQ Federated Distribution and FedX Complete Distribution exhibit opposite performance

ANAPSID and FedX Federated Distribution exhibit positive correlation

Experimental Evaluation: Federated Engine Trend – Complex



○ Correlation 1.0

🔵 Positive Correlation

🔴 Negative Correlation

Configuration 1: ANAPSID Complete Distribution

Configuration 2: ANAPSID Federated Distribution

Configuration 3: ARQ Complete Distribution

Configuration 4: ARQ Federated Distribution

Configuration 5: FedX Complete Distribution

Configuration 6: FedX Federated Distribution

ANAPSID and FedX Federated Distribution exhibit opposite performance

Set of queries do not seem to be general enough to effectively measure the performance of existing federated engines, and ensure generality of the results.

3 FORMALIZATION OF THE QUERY DECOMPOSITION PROBLEM

Motivation

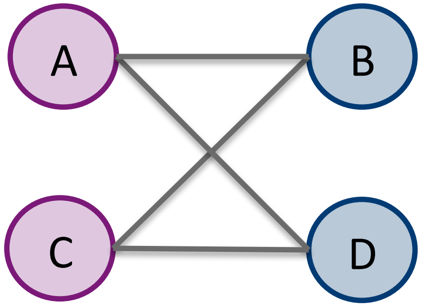
- Cast the federated **Query Decomposition Problem** into the **Vertex Coloring Problem**.

- **SPARQL queries** are mapped to **vertex coloring graphs**.

- Determine the **complexity** of benchmark queries, and explain the **observed variables**.

The Vertex Coloring Problem (1)

- Coloring the vertices of a graph such that no two adjacent vertices share the same color.



- Minimizes the number of colors for a given graph.

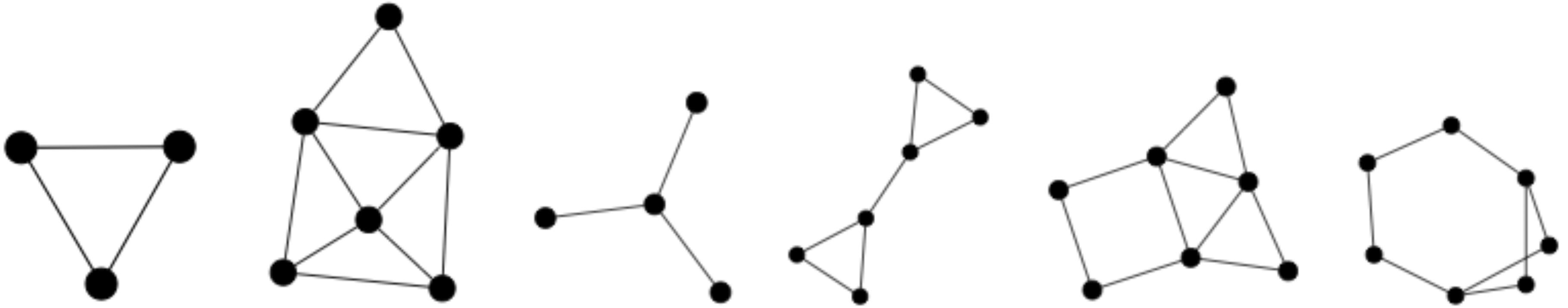
- NP-hard problem [Garey 79].

The Vertex Coloring Problem (2)

A Solution …

- DSATUR is an approximation that can find optimal solutions.

- Depends on the shape of the input graph.



(a) Cycle (b) Wheel (c) Star (d) Cactus (e) Polygon Tree (f) Necklace

Mapping of the Query Decomposition Problem into the Vertex Coloring Problem

- **Nodes** correspond to **triple patterns** in the query
- **Edges** connect two nodes if:
  - It is not possible to perform a JOIN between the two triple patterns
  - The triple patterns cannot be answered by the same endpoint

Example: Decomposing a Federated Query

```
@PREFIX foaf:<http://xmlns.com/foaf/0.1/>
@PREFIX geonames:<http://www.geonames.org/ontology#>

SELECT ?name ?location WHERE {
t1 ?artist foaf:name ?name .
t2 ?artist foaf:based_near ?location .
t3 ?location geonames:parentFeature ?germany .
t4 ?germany geonames:name 'Federal Republic of Germany' .}
```

t1 @Jamendo
SWDF
DBpedia
Bibliographic

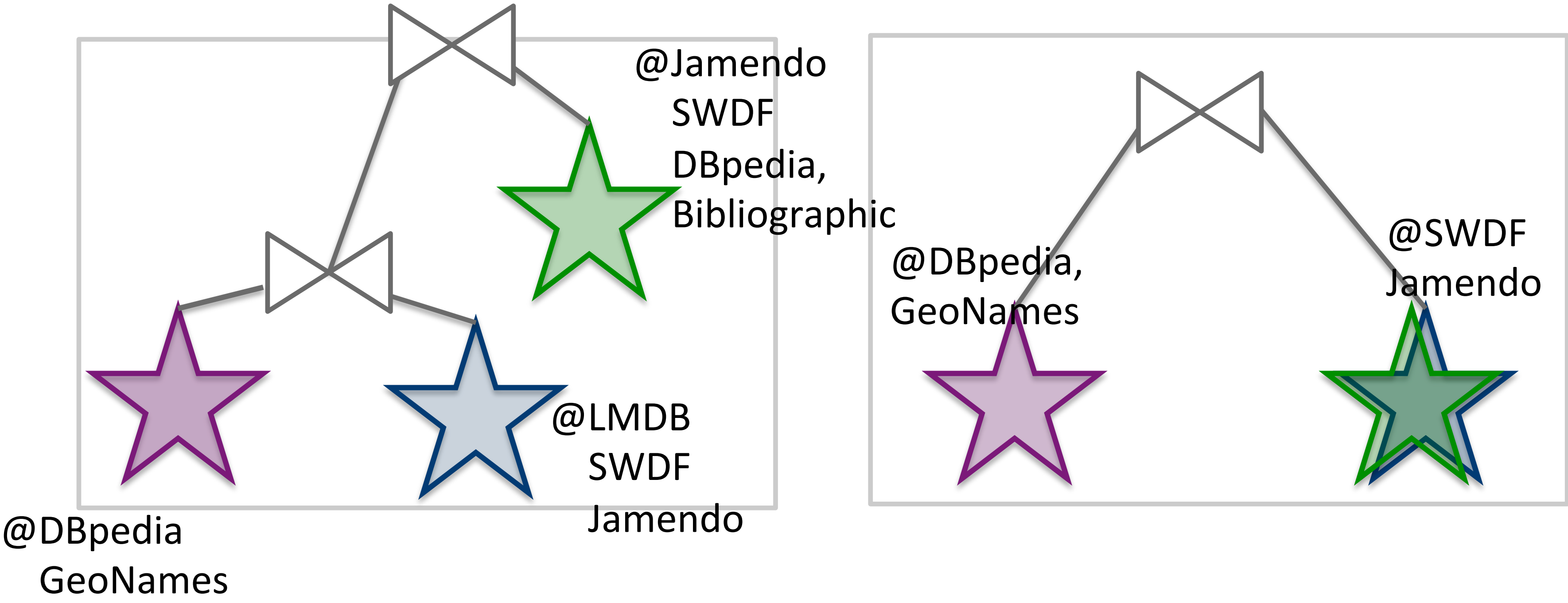
t2 @LMDB
SWDF
Jamendo

t3 @DBpedia
GeoNames

t4 @DBpedia
GeoNames

Example: Decomposing a Federated Query

Query Decompositions



@Jamendo
SWDF
DBpedia,
Bibliographic

@LMDB
SWDF
Jamendo

@DBpedia
GeoNames

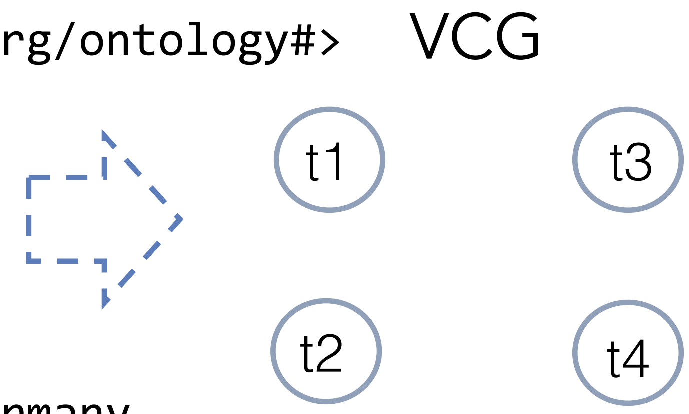
@DBpedia,
GeoNames

@SWDF
Jamendo

Example: Decomposing a Federated Query

```
@PREFIX foaf:<http://xmlns.com/foaf/0.1/>
@PREFIX geonames:<http://www.geonames.org/ontology#>

SELECT ?name ?location WHERE {
t1 ?artist foaf:name ?name .
t2 ?artist foaf:based_near ?location .
t3 ?location geonames:parentFeature ?germany .
t4 ?germany geonames:name 'Federal Republic of Germany’ .}
```



t1 @Jamendo
SWDF
DBpedia
Bibliographic

t2 @LMDB
SWDF
Jamendo

t3 @DBpedia
GeoNames

t4 @DBpedia
GeoNames

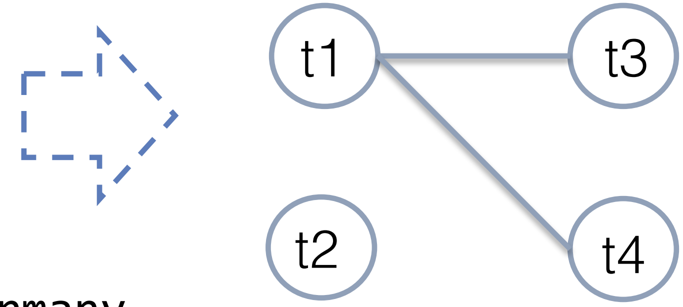
Example: Decomposing a Federated Query

t1 does **not share** a variable with t3
t1 does **not share** a variable with t4

```
@PREFIX foaf:<http://xmlns.com/foaf/0.1/>
@PREFIX geonames:<http://www.geonames.org/ontology#>

SELECT ?name ?location WHERE {
t1 ?artist foaf:name ?name .
t2 ?artist foaf:based_near ?location .
t3 ?location geonames:parentFeature ?germany .
t4 ?germany geonames:name 'Federal Republic of Germany’ .}
```

VCG



t1 @Jamendo
SWDF
DBpedia
Bibliographic

t2 @LMDB
SWDF
Jamendo

t3 @DBpedia
GeoNames

t4 @DBpedia
GeoNames

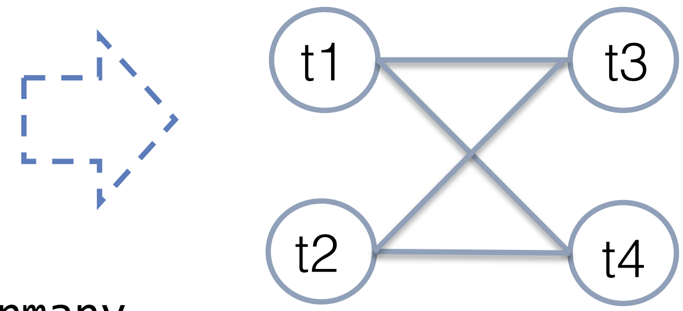
Example: Decomposing a Federated Query

t2 does **not share** an Endpoint with t3, t4
t2 does **not share** a variable with t4

```
   @PREFIX foaf:<http://xmlns.com/foaf/0.1/>
   @PREFIX geonames:<http://www.geonames.org/ontology#>

   SELECT ?name ?location WHERE {
t1 ?artist foaf:name ?name .
t2 ?artist foaf:based_near ?location .
t3 ?location geonames:parentFeature ?germany .
t4 ?germany geonames:name 'Federal Republic of Germany’ .}
```

VCG



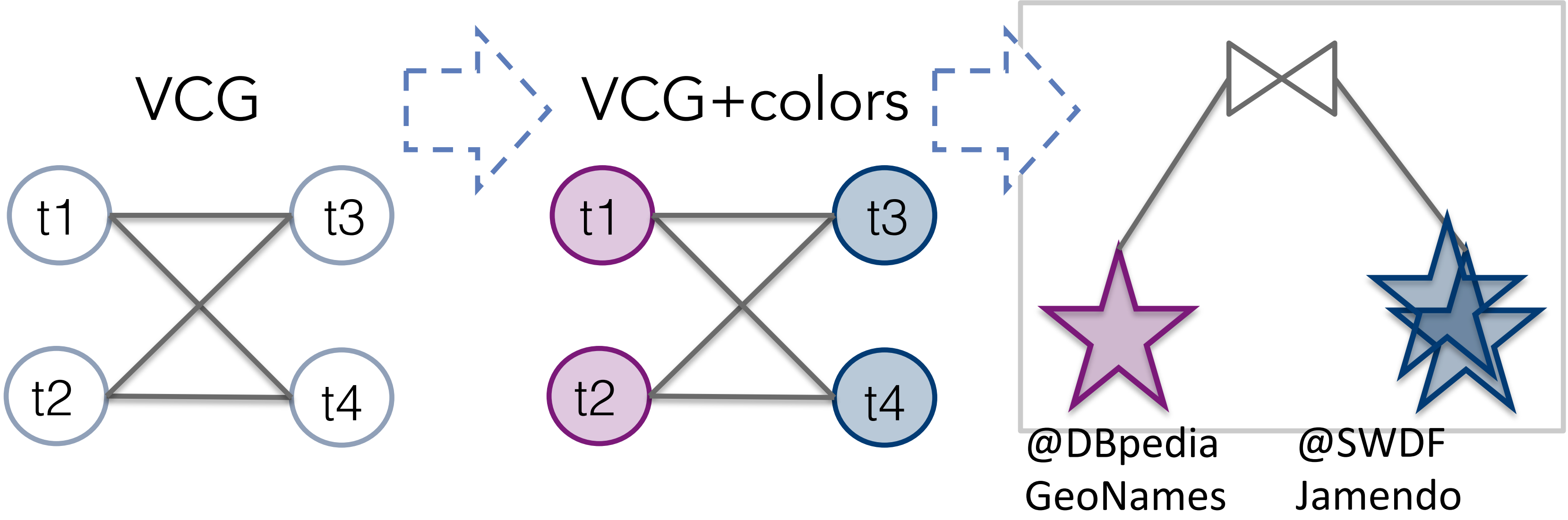
t1 @Jamendo
SWDF
DBpedia
Bibliographic

t2 @LMDB
SWDF
Jamendo

t3 @DBpedia
GeoNames

t4 @DBpedia
GeoNames

Example: Decomposing a Federated Query



Query Complexity Analysis with the VCG

Query 1: CD6 FedBench

```
@PREFIX foaf:<http://xmlns.com/foaf/0.1/>
@PREFIX geonames:<http://www.geonames.org/ontology#>

SELECT ?name ?location WHERE {
   ?artist foaf:name ?name .
   ?artist foaf:based_near ?location .
   ?location geonames:parentFeature ?germany .
   ?germany geonames:name 'Federal Republic of Germany’ .}
```

Query Complexity Analysis with the VCG

Query 2

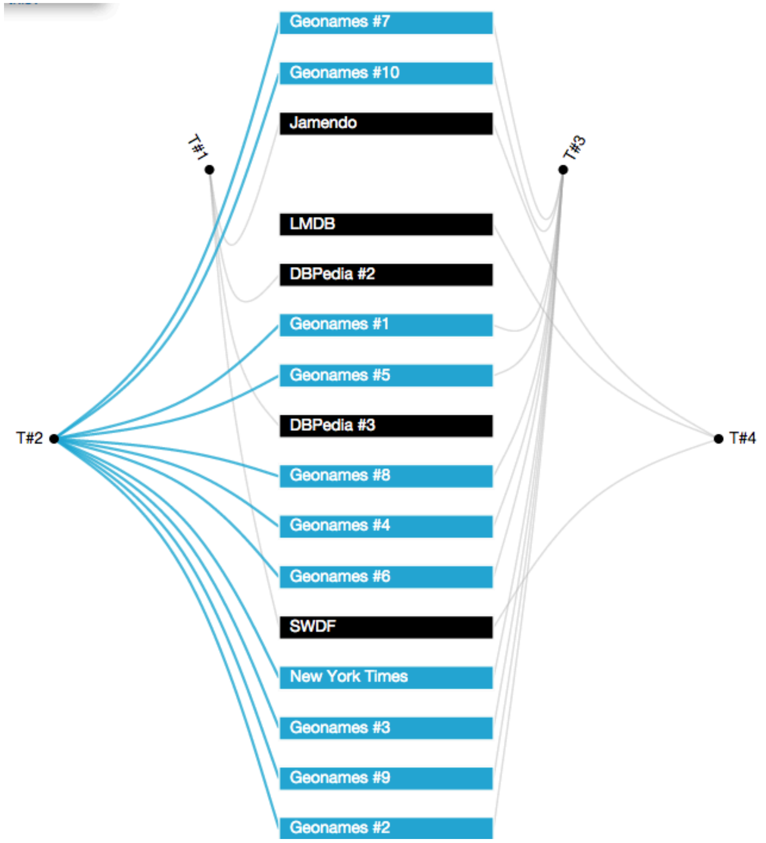
```
SELECT DISTINCT ?drug ?drug1 ?drug2 ?drug3 ?drug4  ?d1 WHERE {
?drug1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/drugCategory> <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugcategory/antibiotics> .
?drug2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/drugCategory> <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugcategory/antiviralAgents> .
?drug3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/drugCategory> <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugcategory/antihypertensiveAgents> .
?drug4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/drugCategory> <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugcategory/anti-bacterialAgents> .
?drug1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/target> ?o1 .
?o1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/genbankIdGene> ?g1 .
?o1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/locus> ?l1 .
?o1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/molecularWeight> ?mw1 .
?o1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/hprdId> ?hp1 .
?o1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/swissprotName> ?sn1 .
?o1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/proteinSequence> ?ps1 .
?o1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/generalReference> ?gr1 .
?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/target>?o1 .
?drug2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/target> ?o2 .
?o1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/genbankIdGene> ?g2 .
?o2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/locus> ?l2 .
?o2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/molecularWeight> ?mw2 .
?o2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/hprdId> ?hp2 .
?o2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/swissprotName> ?sn2 .
?o2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/proteinSequence> ?ps2 .
?o2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/generalReference> ?gr2 .
?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/target>?o2 .
?drug3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/target> ?o3 .
?o3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/genbankIdGene> ?g3 .
?o3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/locus> ?l3 .
?o3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/molecularWeight> ?mw3 .
?o3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/hprdId> ?hp3 .
?o3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/swissprotName> ?sn3 .
?o3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/proteinSequence> ?ps3 .
?o3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/generalReference> ?gr3 .
?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/target>?o3 .
?drug4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/target> ?o4 .
?o4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/genbankIdGene> ?g4 .
?o4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/locus> ?l4 .
?o4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/molecularWeight> ?mw4 .
?o4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/hprdId> ?hp4 .
?o4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/swissprotName> ?sn4 .
?o4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/proteinSequence> ?ps4 .
?o4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/generalReference> ?gr4 .
?drug <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/target>?o4 .
OPTIONAL{
    ?I1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/interactionDrug2> ?drug1 .
    ?I1 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/interactionDrug1> ?drug .
    ?I2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/interactionDrug2> ?drug2 .
    ?I2 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/interactionDrug1> ?drug .
    ?I3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/interactionDrug2> ?drug3 .
    ?I3 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/interactionDrug1> ?drug .
    ?I4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/interactionDrug2> ?drug4 .
    ?I4 <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/interactionDrug1> ?drug .}}
```

48 triple patterns

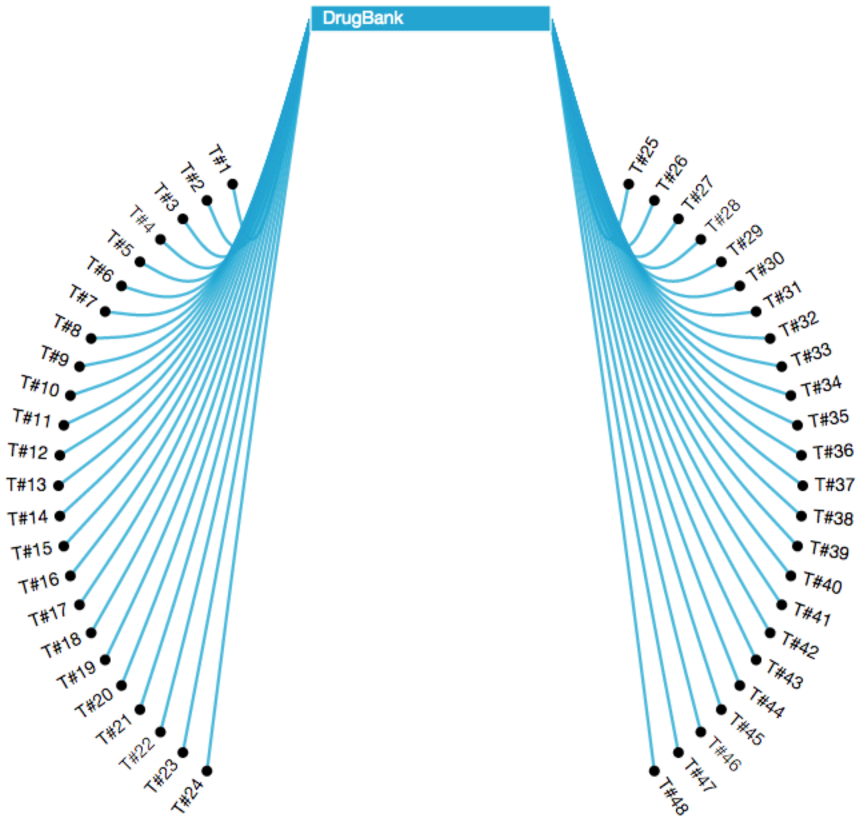
Query Complexity Analysis with the VCG

Sources

Query 1



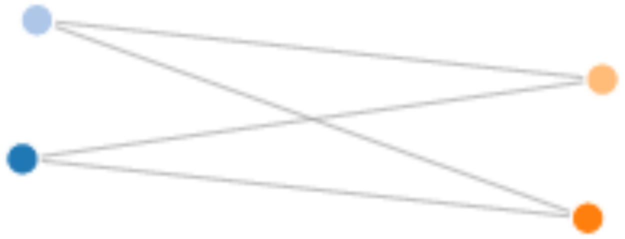
Query 2



Query Complexity Analysis with the VCG

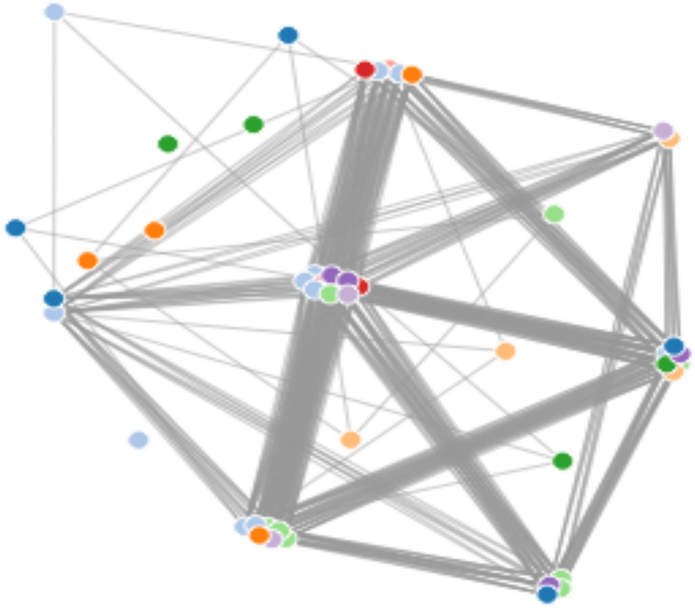
Vertex Coloring Graphs

Query 1



(Bipartite Graph)

Query 2



Query Complexity Analysis with the VCG

Decompositions

Query 1

Exclusive Groups & ANAPSID

#T1

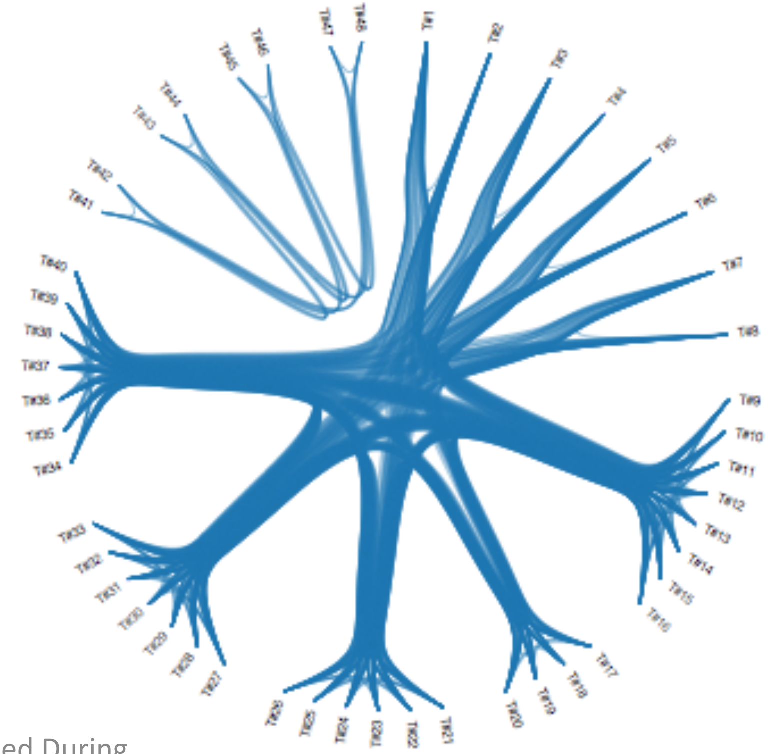
#T2

#T3

#T4

Query 2

Exclusive Groups



Query Complexity Analysis with the VCG

Decompositions

Query 1

Exclusive Groups & ANAPSID

#T1

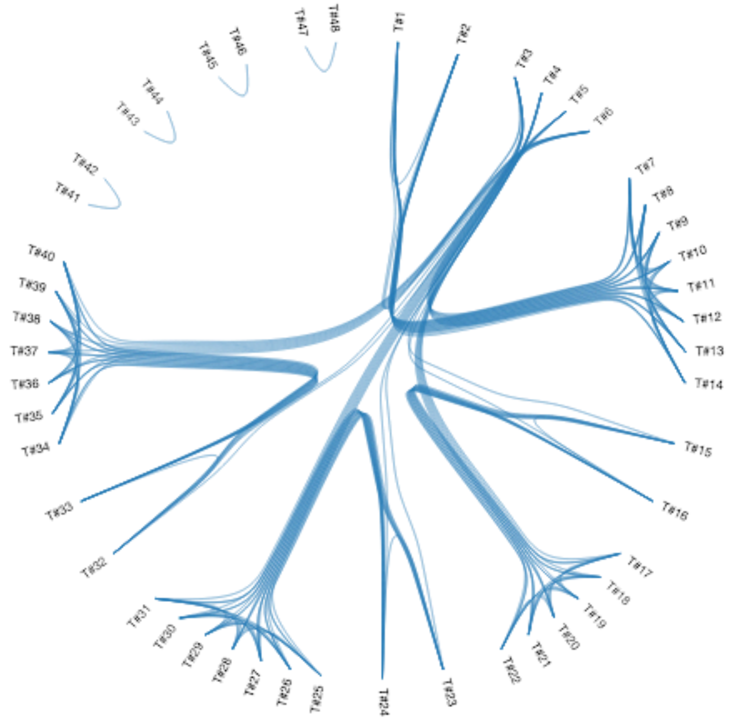
#T2

#T3

#T4

Query 2

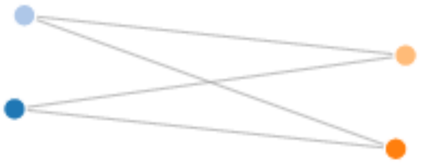
ANAPSID



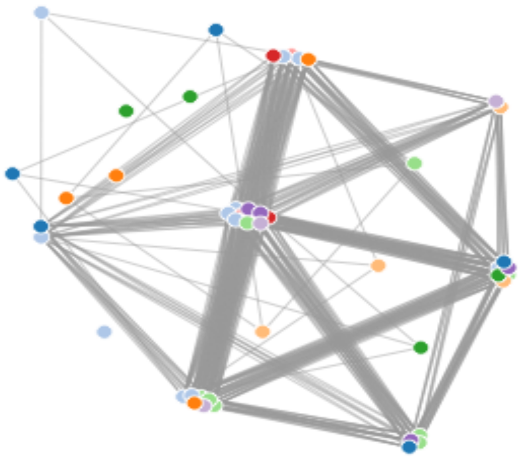
Query Complexity Analysis with the VCG

Conclusions from the example

- Query 1 is simple

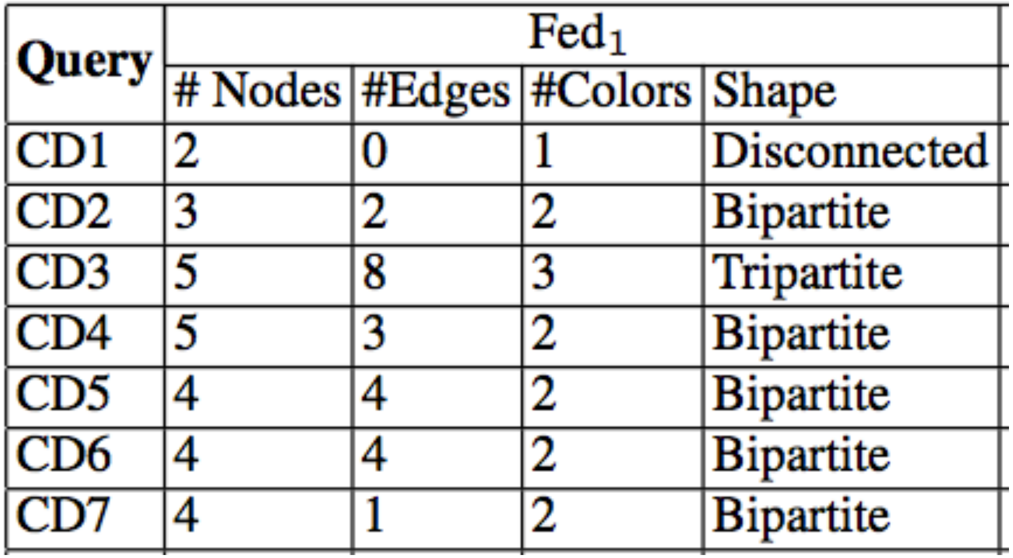


- Query 2 is complex



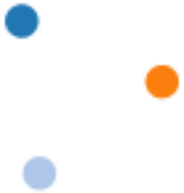
Analyzing FedBench with the VCG (1)

Cross Domain



Query	Fed$_1$			
	# Nodes	#Edges	#Colors	Shape
CD1	2	0	1	Disconnected
CD2	3	2	2	Bipartite
CD3	5	8	3	Tripartite
CD4	5	3	2	Bipartite
CD5	4	4	2	Bipartite
CD6	4	4	2	Bipartite
CD7	4	1	2	Bipartite

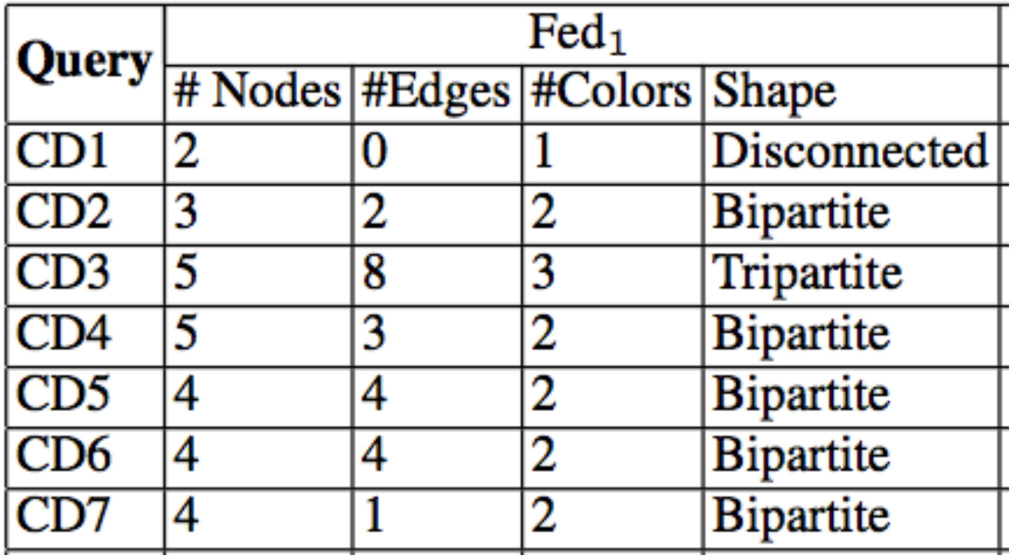
✔



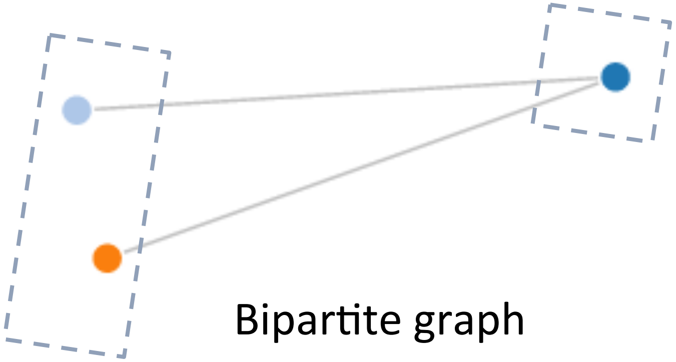
Disconnected graph

Analyzing FedBench with the VCG (1)

Cross Domain



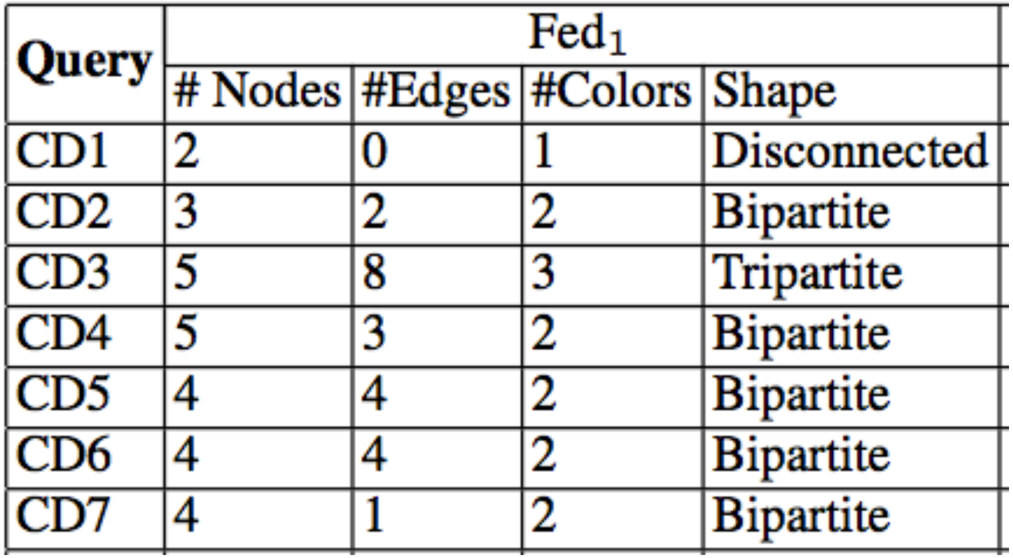
Query	Fed$_1$				
	# Nodes	#Edges	#Colors	Shape	
CD1	2	0	1	Disconnected	✓
CD2	3	2	2	Bipartite	✓
CD3	5	8	3	Tripartite	
CD4	5	3	2	Bipartite	
CD5	4	4	2	Bipartite	
CD6	4	4	2	Bipartite	
CD7	4	1	2	Bipartite	



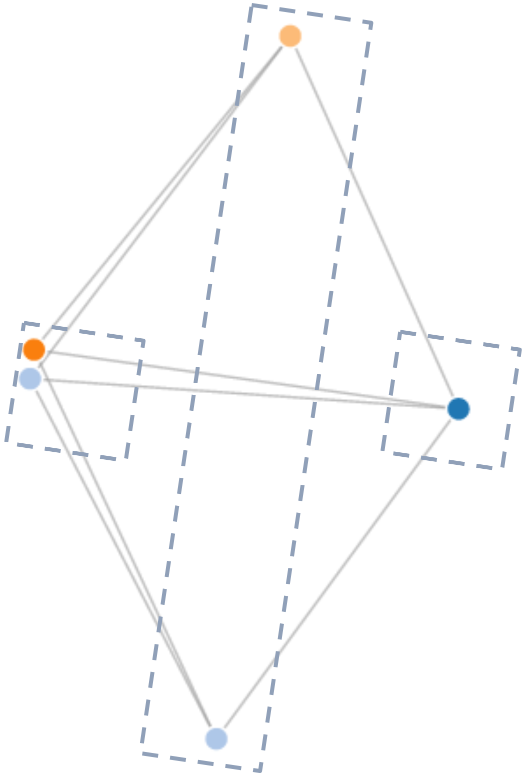
Bipartite graph

Analyzing FedBench with the VGC (1)

Cross Domain



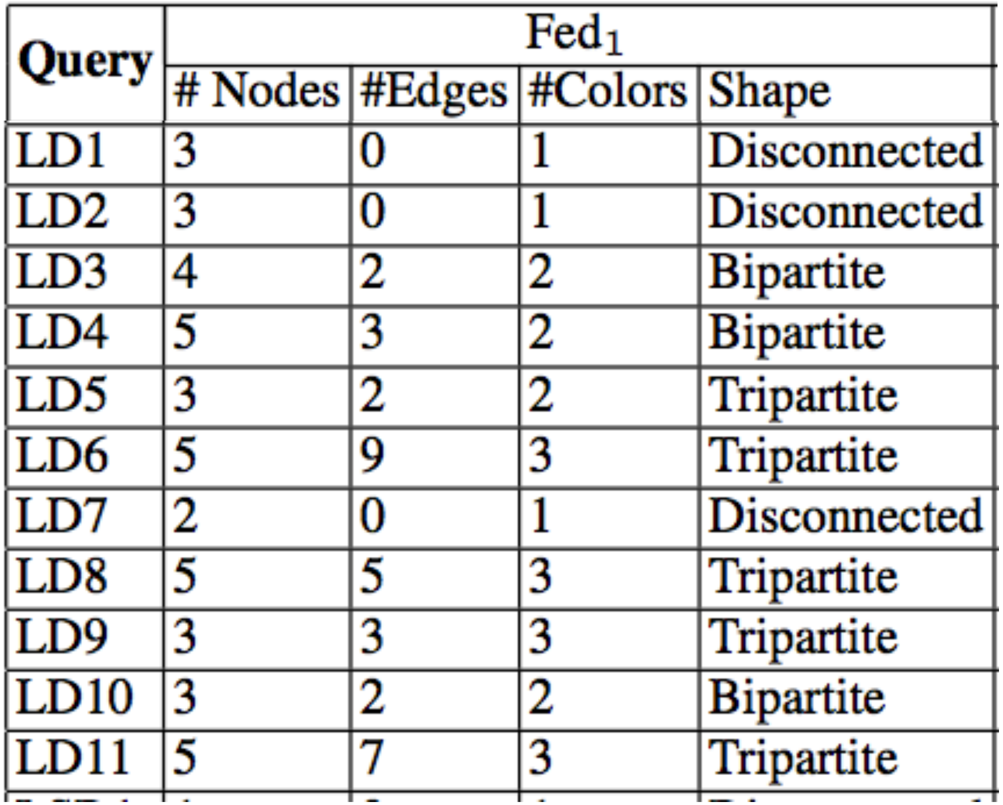
Query	Fed$_1$				
	# Nodes	#Edges	#Colors	Shape	
CD1	2	0	1	Disconnected	✓
CD2	3	2	2	Bipartite	✓
CD3	5	8	3	Tripartite	✓
CD4	5	3	2	Bipartite	✓
CD5	4	4	2	Bipartite	✓
CD6	4	4	2	Bipartite	✓
CD7	4	1	2	Bipartite	✓



Tripartite graph

Analyzing FedBench with the VCG (2)

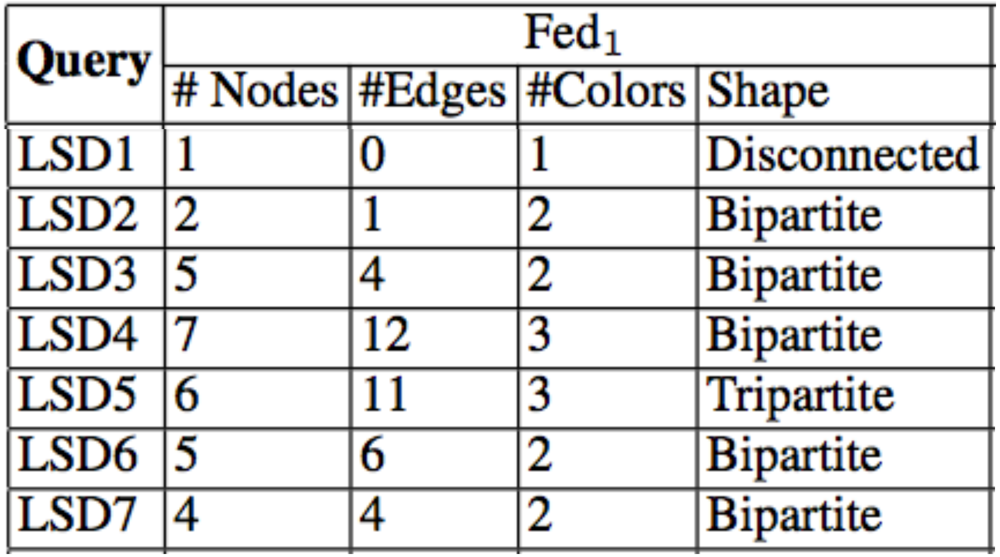
Linked Data



Query	Fed_1				
	# Nodes	#Edges	#Colors	Shape	
LD1	3	0	1	Disconnected	✓
LD2	3	0	1	Disconnected	✓
LD3	4	2	2	Bipartite	✓
LD4	5	3	2	Bipartite	✓
LD5	3	2	2	Tripartite	✓
LD6	5	9	3	Tripartite	✓
LD7	2	0	1	Disconnected	✓
LD8	5	5	3	Tripartite	✓
LD9	3	3	3	Tripartite	✓
LD10	3	2	2	Bipartite	✓
LD11	5	7	3	Tripartite	✓

Analyzing FedBench with the VCG (3)

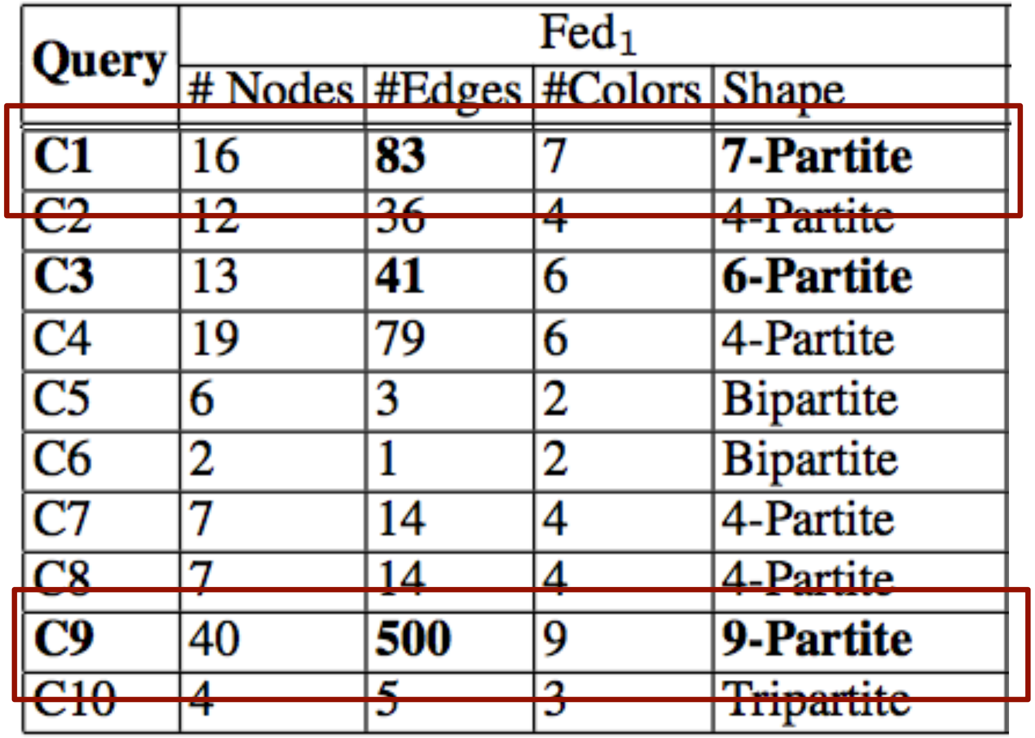
Life Science Data



Query	Fed_1				
	# Nodes	#Edges	#Colors	Shape	
LSD1	1	0	1	Disconnected	✓
LSD2	2	1	2	Bipartite	✓
LSD3	5	4	2	Bipartite	✓
LSD4	7	12	3	Bipartite	✓
LSD5	6	11	3	Tripartite	✓
LSD6	5	6	2	Bipartite	✓
LSD7	4	4	2	Bipartite	✓

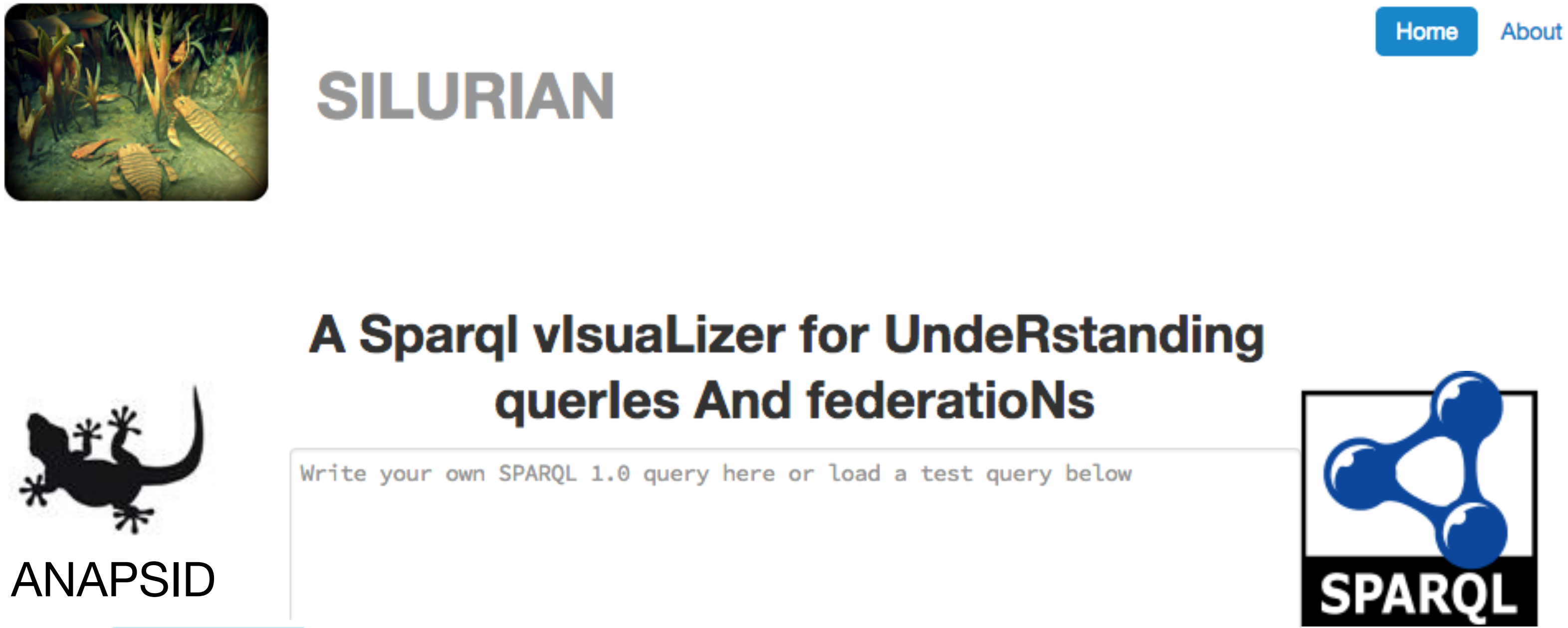
Analyzing Complex Queries with the VCG

Complex Queries



Query	Fed$_1$			
	# Nodes	#Edges	#Colors	Shape
C1	16	**83**	7	**7-Partite**
C2	12	36	4	4-Partite
C3	13	**41**	6	**6-Partite**
C4	19	79	6	4-Partite
C5	6	3	2	Bipartite
C6	2	1	2	Bipartite
C7	7	14	4	4-Partite
C8	7	14	4	4-Partite
C9	40	**500**	9	**9-Partite**
C10	4	5	3	Tripartite

C1 and C9 are complex queries that none of the engines was able to execute in less than **30 minutes!**



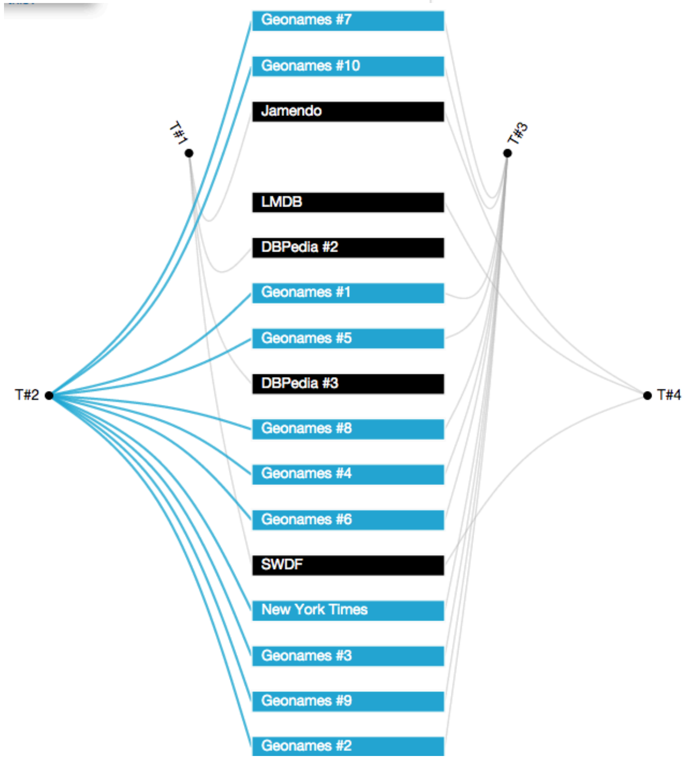
SILURIAN

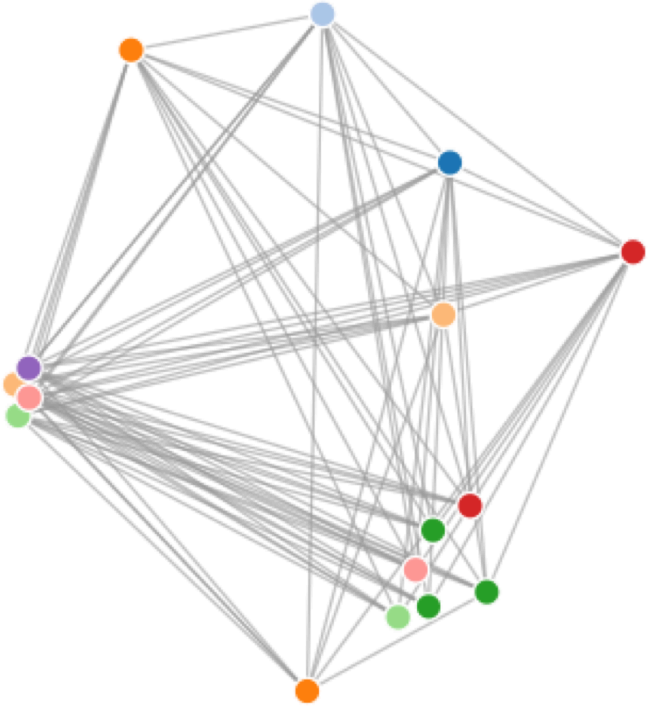
A Sparql vIsuaLizer for UndeRstanding querIes And federatioNs

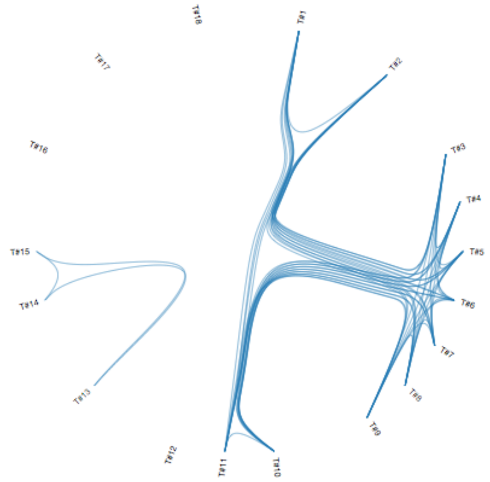
Write your own SPARQL 1.0 query here or load a test query below

ANAPSID

SPARQL





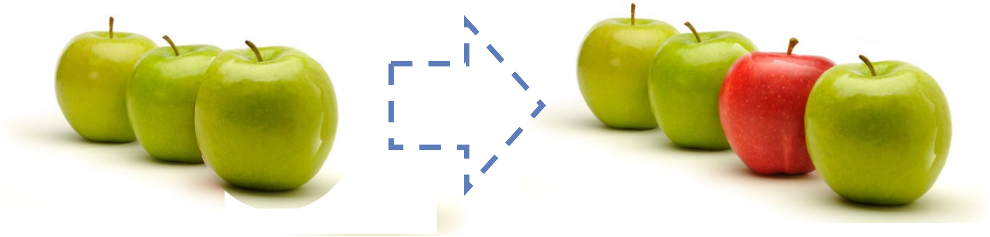


4 LESSONS LEARNED

Lessons Learned (1)

1. Testbeds should provide the **mechanisms to test/stress** the engines in each of the **different dimensions** that affect the execution of SPARQL queries.

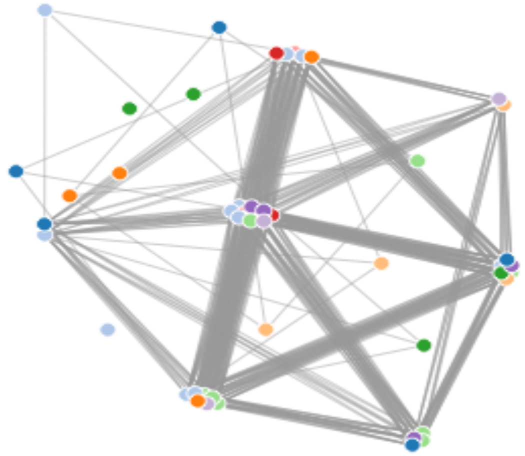
2. Testbeds should specify not only data and tests, but **values for different parameters.** This allows for:
    - Reproducibility of experiments
    - Generality of results
    - Avoid hiding red apples!



Lessons Learned (2)

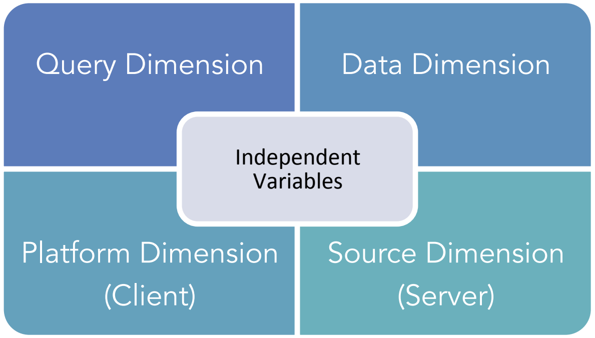
3. **Formalization of the different problems**, e.g., the query decomposition problem, allow to explain:
   - Properties of the queries/tests *(simple vs. complex)*
   - The impact on the observed variables

4. Including **challenging tests** in testbeds allow to identify **open problems**.

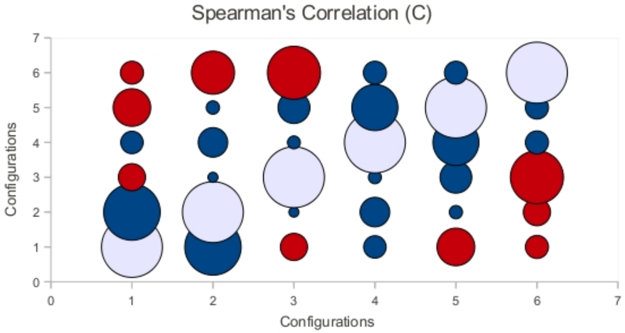


Summary

- We studied different parameters that may affect the behavior of federated SPARQL engines.



- We evaluated existing Federated SPARQL engines using FedBench.



- We formally studied the complexity of SPARQL query decompositions with the VCG

Future Work

Extend existing benchmarks to consider the parameters studied in this evaluation.

References

- G. Montoya, M.E. Vidal, O. Corcho, E. Ruckhaus, C. Buil-Aranda et al. **Benchmarking Federated SPARQL Query Engines: Are Existing Testbeds Enough?** ISWC 2012.

- S. Castillo, M.E. Vidal, M. Acosta, G. Montoya, G. Palma. **A Vertex Coloring Based Approach for Decomposing SPARQL Federated Queries**. University Simon Bolivar, Technical Report, 2013.