FedX: Busca otimizada sobre base de dados Federada

Hugo Paulino Bonfim Takiuchi

Curitiba - 2018

Artigo Base

FedX: Optimization Techniques for Federated Query Processing on Linked Data

Autores: Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, Michael Schmidt

Tópicos

- Introdução
- Trabalhos Relacionados
- FedX
- Implementação
- Avaliação

Introdução

- Acessos bases RDF através de endpoints SPARQL (HTTP)
- Bases Heterogêneas (Decentralizado)
- Princípios dos dados ligados (Ontologia)
- Usuário visualiza as bases com uma federação
- Sistemas atuais apresentam desempenho ruim:
  - Não suportam SPARQL completamente
  - Utilizam pré-processamento

Introdução

- Otimização: diminuir o número de request(acessos) sobre as bases para responder a consulta
  - latência na rede
  - quantidade de dados transferidos (resultados intermediários)

Introdução

- Contribuição do artigo:
  - Otimização para busca federada:
    - novo método de juntar os resultados intermediários (grupo de triplas)
    - Não utiliza pré-processamento (ASK SPARQL e cache)
  - Fedx: motor para busca SPARQL federada sobre bases de dados RDF, proporcionando integração virtual dos dados
  - Comparação do FedX com outros motores em base de dados reais

Trabalhos Relacionados

- Fontes:
  - Bottom-up: encontra fontes durante o processamento da consultada
  - Top-down: as fontes relevantes são conhecidas antecipadamente

Trabalhos Relacionados

- DARQ:
  - busca federada em (distribuídos) SPARQL endpoints
  - Utiliza service description (metadados)
  - Seleção de recursos baseado em predicados
  - Predicados tem que estar ligados (bound)

Trabalhos Relacionados

- SemWIQ:
  - Seleção de fontes baseadas nas informações sobre os tipos de dados
  - Catálogo de dados (metadados)
  - Todos os sujeitos são variáveis
  - Tipos dos sujeitos devem ser conhecidos

FedX

- BGP (busca conjuntiva): grafos formados pelos padrões de triplas
- Otimizações sobre os subgrafos das subconsultas (OPTIONAL)
- FedX é top-down
- Utiliza SPARQL 1.0

FedX

- Técnicas de busca:
  - triplas são enviadas e avaliadas em todos os endpoints SPARQL
    - Problema: Retorna resultados irrelevantes
    - (?a, sameAs, ?b)
  - busca analisa padrão por padrão (Nested Loop Join)
  - Resultado intermediário usados para a próxima subconsulta
    - Problema: Muitos requests remotos

FedX

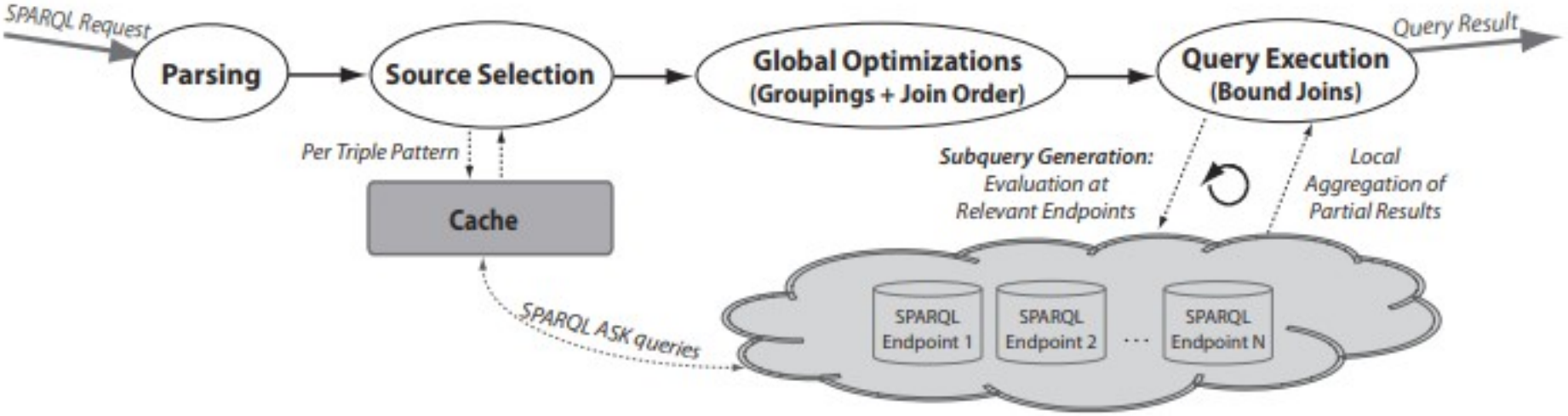


Fig. 1: Federated Query Processing Model

FedX

- Parsing

```
SELECT ?drug ?title WHERE {
   ?drug drugbank:drugCategory drugbank-category:micronutrient .
   ?drug drugbank:casRegistryNumber ?id .
   ?keggDrug rdf:type kegg:Drug .
   ?keggDrug bio2rdf:xRef ?id .
   ?keggDrug purl:title ?title .
}
```

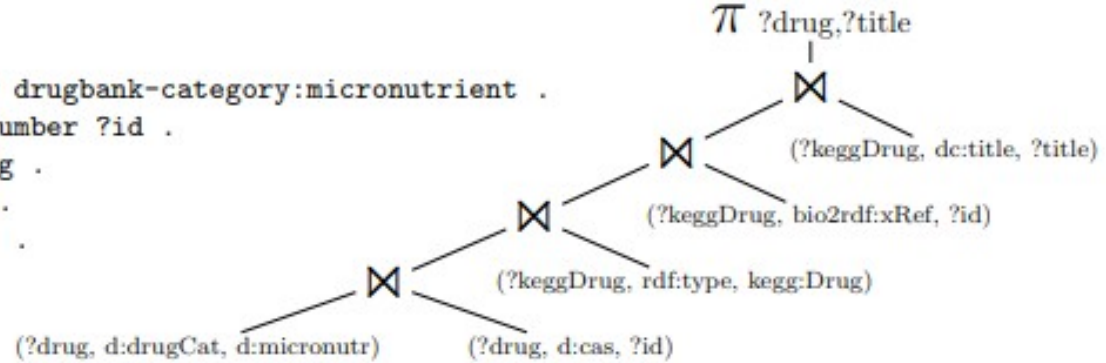


Fig. 2: Life Science Query 6 and the Corresponding Unoptimized Query Plan.

FedX

- Seleção de fontes
  - ASK SPARQL
    - Problema: (?S rdf:type ?0)
    - Resolvido durante a otimização
  - Cache local
- SERVICE para SPARQL 1.1
    - Problema: saber a fonte de antecipado

FedX

- Ordem de junção
  - determina a quantidade de resultados intermediários
  - triplas (conjunto de triplas) de menor custo
  - Mais variáveis locais maior o custo (não seletiva)
  - Armazenadas em uma lista de menor custo primeiro
  - Preferências para grupos exclusivos

FedX

- Ordem de junção

Algorithm 1 Join Order Optimization

order(*joinargs*: list of n join arguments) {

```
1:  left ← joinargs
2:  joinvars ← ∅
3:  for i = 1 to n do
4:      mincost ← MAX_VALUE
5:      for all j ∈ left do
6:          cost ← estimateCost(j, joinvars)
7:          if cost < mincost then
8:              arg ← j
9:              mincost ← cost
10:         end if
11:     end for
12:     joinvars ← joinvars ∪ vars(arg))
13:     result[i] ← arg
14:     left ← left − arg
15: end for
16: return result }
```

FedX

- Grupos Exclusivos

Definition 1. *Let $t_1 \ldots t_n$ be a set of triple patterns (corresponding to a conjunctive query), $S_1 \ldots S_n$ be distinct data sources, and S_t the set of relevant sources for triple pattern t. For $s \in \{S_1, \ldots, S_n\}$ we define $E_s := \{t \mid t \in \{t_1..t_n\}$ s.t. $S_t = \{S\}\}$ as the exclusive groups for source S, i.e. the triple patterns whose single relevant source is S.*

- Conjunto de triplas que apresentam apenas uma mesma fonte relevante S
- Um conjunto para cada fonte Sn

FedX

- Grupos Exclusivos
  - Triplas do grupo são envidas juntas uma única vez
  - Diminui a quantidade de request

FedX

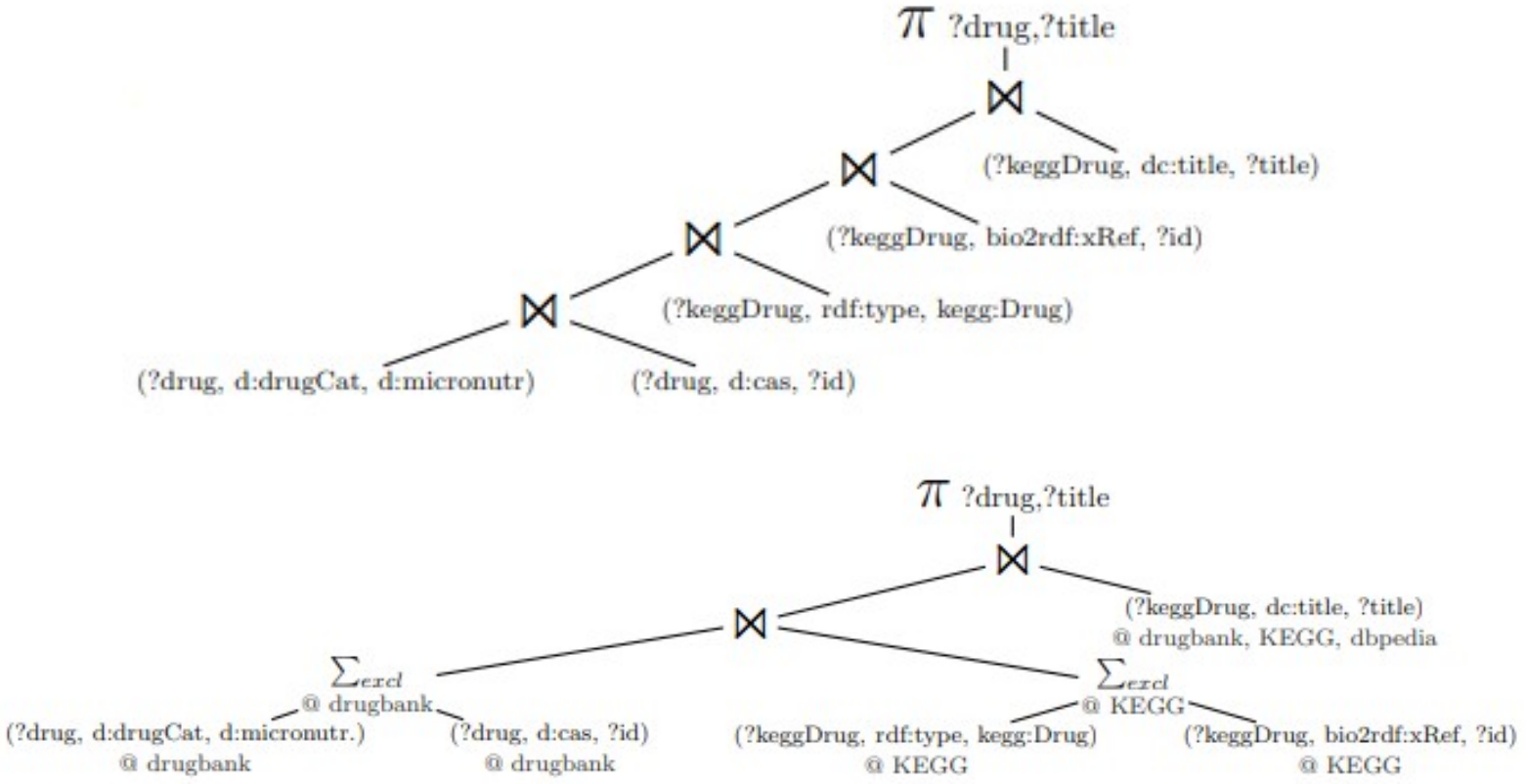


Fig. 3: Execution Plan of Life Science Query 6 (Including Optimizations)

FedX

- Bound Join
  - vários mapeamentos dentro de um único bloco
  - Diminui quantidade de request proporcional ao tamanho do bloco
  - Mapeamento liga um resultado a uma nova subconsulta
  - UNION SPARQL
  - Otimiza o Nested Loop Join

FedX

- Bound Join Exemplo

Tripla: (?S, name, ?O)

Sequencia de entrada: [?S=Person1,?S=Person2,?S=Person3]

Fonte: t1=(Person1, name, ‘Peter’)

t2=(Person3, name, ‘Andreas’)

a) Expected Result

?S	?O
Person1	Peter
Person3	Andreas

b) SPARQL subquery

```
SELECT ?O_1 ?O_2 ?O_3 WHERE {
  { Person1 name ?O_1 } UNION
  { Person2 name ?O_2 } UNION
  { Person3 name ?O_3} }
```

c) Subquery result

?O_1	?O_2	?O_3
Peter		
		Andreas

Fig. 4: Sample execution for bound join processing of (?S, name, ?O)

Implementação

- JAVA
- Sesame
  - Análise e busca sobre uma base RDF
  - Parsing de dados RDF
  - Acesso através de API
  - Repositórios locais
- Paralelismo tratado com Scheduler (subconsultas em uma FIFO)

Avaliação

- Benchmark: FedBench
  - simular busca federada sobre base de dados reais
- Bases: Cross Domain (CD) e Life Science (LS)
- Motores: DARQ e Alibaba
- Resultado:
  - Requests: 170,579 (DARQ), 93,248 (AliBaba), 23 (FedX) para consuta CD3

Avaliação

- **a) Query Characteristics**

Cross Domain (CD)			
	#Tp.	*#Src*	*#Res*
1	3	2	90
2	3	2	1
3	5	5	2
4	5	5	1
5	4	5	2
6	4	4	11
7	4	5	1

Life Science (LS)			
	#Tp.	*#Src*	*#Res*
1	2	2	1159
2	3	4	333
3	5	3	9054
4	7	2	3
5	6	3	393
6	5	3	28
7	5	3	144

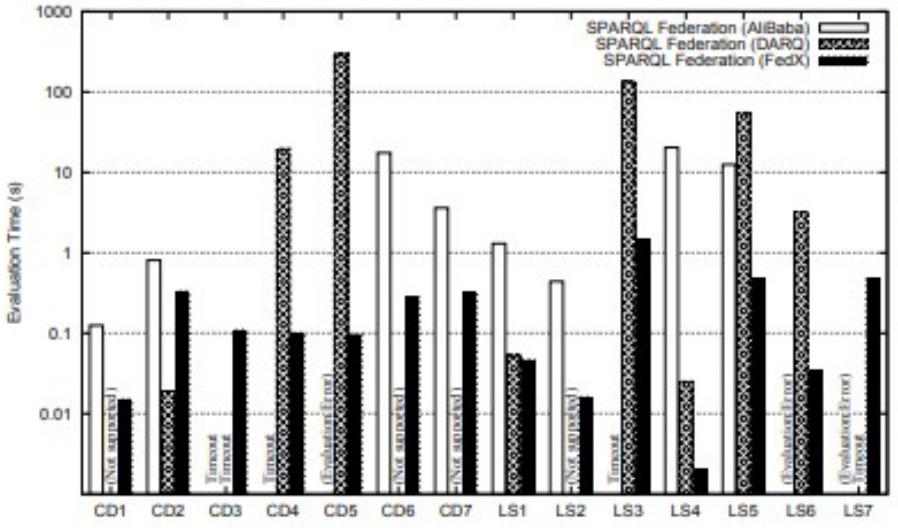
b) Datasets

	#Triples	DARQ *SD*
DBpedia	43.6M	01:05:46
NYTimes	335k	00:00:09
LinkedMDB	6.15M	01:07:39
Jamendo	1.05M	00:00:20
GeoNames	108M	n/a
KEGG	1.09M	00:00:18
Drugbank	767k	00:00:12
ChEBI	7.33M	00:01:16

Table 1: Query characteristics of benchmark queries (a) and datasets used (b): Number of triple patterns (#Tp.), data sources (#Src) and results (#Res); number of triples included in datasets (#Triples) and preprocessing time for DARQ Service Description (SD) in hh:mm:ss

Avaliação

a) Benchmark Results



	AliBaba	DARQ	FedX
CD1	0.125	x	0.015
CD2	0.807	0.019	0.330
CD3	>600	>600	0.109
CD4	>600	19.641	0.100
CD5	#	294.890	0.097
CD6	17.499	x	0.281
CD7	3.623	x	0.324
LS1	1.303	0.053	0.047
LS2	0.441	x	0.016
LS3	>600	133.414	1.470
LS4	20.370	0.025	0.001
LS5	12.504	55.327	0.480
LS6	#	3.236	0.034
LS7	#	>600	0.481

x not supported

evaluation error

Fig. 5: Experimental Results of Cross Domain (CD) and Life Science (LS) Queries in SPARQL Federation: a) Benchmark Results of AliBaba, DARQ, and FedX. b) Influence of Caching `ASK` Requests for Source Selection. c) Total Number of Requests sent to Endpoints; Parentheses Indicate Timeouts after 10min or Evaluation Errors. d) Comparison of Join Operator Implementations in the SPARQL Federation: *Controlled Worker Join* (CJ), *Controlled Worker Bound Join* (CBJ). All Runtimes in Seconds.

Avaliação

b) Caching in FedX

	No Caching	Caching
CD1	0.044	0.015
CD2	0.374	0.330
CD3	0.219	0.109
CD4	0.134	0.100
CD5	0.131	0.097
CD6	0.508	0.281
CD7	0.449	0.324
LS1	0.062	0.047
LS2	0.038	0.016
LS3	2.202	1.470
LS4	0.018	0.001
LS5	0.633	0.480
LS6	0.063	0.034
LS7	0.686	0.481

c) Number of Requests

	AliBaba	DARQ	FedX CJ	FedX CBJ
CD1	27	x	7	7
CD2	22	5	2	2
CD3	(93,248)	(170,579)	63	23
CD4	(372,339)	22,331	69	38
CD5	(117,047)	247,343	35	18
CD6	6,183	x	2,457	185
CD7	1,883	x	1,508	138
LS1	13	1	1	1
LS2	61	x	38	18
LS3	(410)	101,386	14,221	2059
LS4	21,281	3	3	3
LS5	16,621	2,666	6,537	458
LS6	(130)	98	315	45
LS7	(876)	(576,089)	5,027	485

d) Join Operators

	CJ	CBJ
CD1	0.016	0.015
CD2	0.349	0.330
CD3	0.203	0.109
CD4	0.134	0.100
CD5	0.115	0.097
CD6	1.560	0.281
CD7	1.336	0.324
LS1	0.053	0.047
LS2	0.025	0.016
LS3	5.435	1.470
LS4	0.001	0.001
LS5	2.146	0.480
LS6	0.103	0.034
LS7	1.763	0.481

Fig. 5: Experimental Results of Cross Domain (CD) and Life Science (LS) Queries in SPARQL Federation: a) Benchmark Results of AliBaba, DARQ, and FedX. b) Influence of Caching `ASK` Requests for Source Selection. c) Total Number of Requests sent to Endpoints; Parentheses Indicate Timeouts after 10min or Evaluation Errors. d) Comparison of Join Operator Implementations in the SPARQL Federation: *Controlled Worker Join* (CJ), *Controlled Worker Bound Join* (CBJ). All Runtimes in Seconds.