Chapter 14

Federated Query Processing over Linked Data

Luiz Brito da Rosa. Msc

Introdução

- A internet vem evoluindo e está saindo da Web de documentos para uma Web de Dados.
- Os princípios de Linked Data foram formuladas com a visão de criar um espaço de dados conectados globalmente.
- O objetivo é de integrar dados semanticamente semelhantes.
- *Linking Open Data*, um projeto que visa ligar dados RDF distribuídos na Web.

Introdução

- Atualmente, o *Linked Open Data cloud* compreende mais de 200 conjuntos de dados que estão interligados por links RDF
- Abrangendo vários domínios que vão desde as Ciências da Vida a vários domínios específicos.
- Juntar informações fornecidas por diferentes fontes.
- É necessário estratégias de processamento de consulta eficiente.
- O maior desafio encontra-se na distribuição natural dos dados.

Categorias

Em relação a este desafio da distribuição de dados, temos a abordagens do estado da arte que pode ser dividido em três categorias principais:

- **MQP - Materialization-based query processing**
  - processamento de consulta baseada em materialização
- **LQP - Lookup-based query processing**
  - processamento de consulta baseado em pesquisa
- **FQP - Federated query processing**
  - processamento de consulta federada

MQP - Materialization-based query processing

- Adotando a ideia de armazenamento de dados, uma abordagem comumente utilizada para o processamento de consultas em cenários de integração em larga escala é integrar conjuntos de dados relevantes em um local, fazendo um armazenamento centralizado de triplas.
- Subconjuntos de Linked Open Data cloud :
  - *LOD cloud cache*
  - *Factforge*
  - *Linked Life Data (23 fontes de dados Biomédico)*
  - OpenPHACTS – vários bancos de dados no relacionados ao espaço farmacêutico.

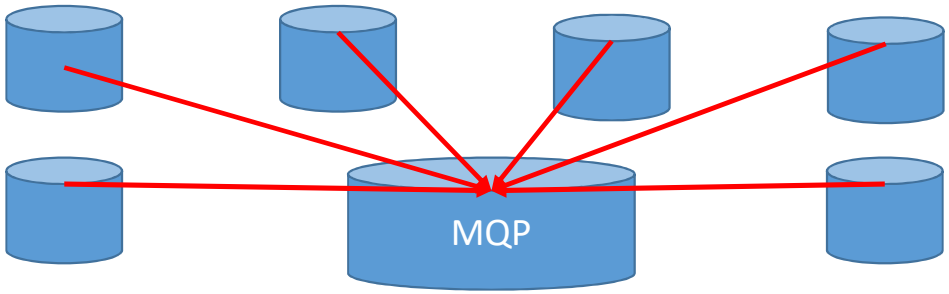
MQP - Materialization-based query processing

Vantagens:

- A maior vantagem, é claro, é que toda a informação está disponível localmente;
- Processamento eficiente de consulta, otimização;
- Baixo tempo de resposta das consultas.

Desvantagens:

- Não há garantia de que os dados são *atualizados* de modo que temos de cuidar de atualizações recentes de fontes originais ou substituindo a cópia local.



LQP - Lookup-based query processing

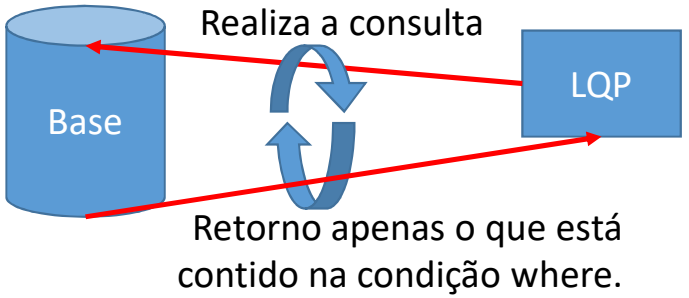
- LQP descarrega os dados das fontes apenas após a inicialização da consulta;
- De acordo com o que está buscando é realizado o download dos dados para processar.
- Durante o processamento da consulta, os dados pertinentes são baixados de forma iterativa;
- Realiza HTTP GETs em URIs que fazem parte dos resultados da consulta intermediária;

LQP - Lookup-based query processing

- As triplas baixadas são utilizadas para avaliar partes da consulta e identificar URIs adicionais, que não serão referenciadas na próxima iteração.
- Este princípio é proposto em combinação, não só com consultas SPARQL, mas também com uma linguagem de consulta declarativa baseado em expressões regulares.
- Uma alternativa para dereferenciamento de URIs é comparar as informações contidas na consulta para estatísticas pré-computadas (índices), identificando as fontes relevantes e fazendo o download dos dados em paralelo em uma única etapa.

LQP - Lookup-based query processing

- Vantagem:
  - Versão atualizada, logo dados recentes;
  - Nível básico de cooperação;
  - É Baseado em dereferenciamento de URI's;
- Desvantagem:
  - Pode não ter uma garantia na integridade dos resultados;
  - Pode ter uma grande quantidade de servidores ou pequenos servidores;
  - Não sabemos o tempo de resposta;
  - Quando usando o cache, o mesmo pode estar desatualizado.



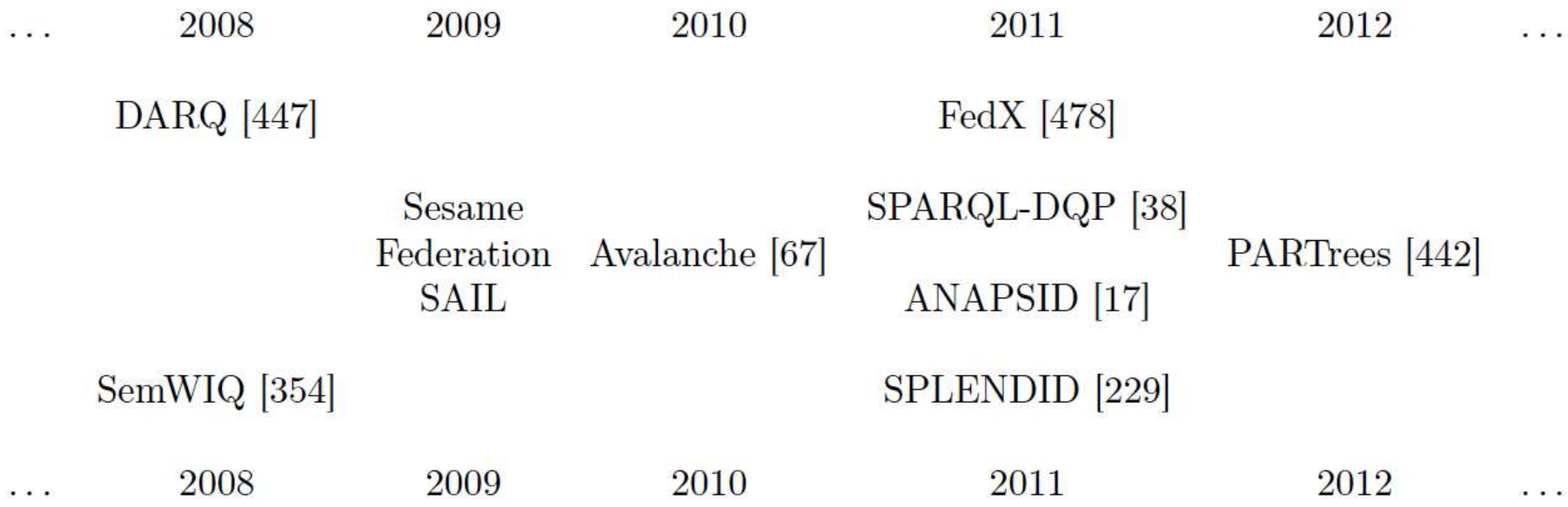
FQP - Federated query processing

- FQP implementa o conceito de integração virtual;
- Várias fontes heterogêneas podem ser consultados como se residindo no mesmo banco de dados.
- Uma integração virtual de terminais SPARQL evita a maior parte das desvantagens da MQP e LQP.
- Partes da consulta são “jogadas” para as fontes e avaliados em seus dados locais, e apenas os resultados intermediários são buscados e combinados em uma máquina central.

FQP - Federated query processing

- O processamento de consultas nesta configuração geralmente segue os seguintes passos.
  - Em primeiro lugar, a consulta precisa ser analisada, convertida numa forma canónica e analisadas para uma possivelmente correção sintática e semântica.
  - Segundo, a consulta é otimizada considerando aspectos como cardinalidade dos resultados intermediários, as fontes relevantes, se divide em conjuntos alternativos de subconsultas, etc.
  - Terceiro, as subconsultas são enviados para as fontes (terminais SPARQL), onde são avaliadas nos dados locais.
  - Finalmente, os resultados são obtidos a partir das fontes e o resultado final é computado.

Ferramentas



DARQ (Distributed ARQ)

- Permite consulta com processamento distribuído sobre um conjunto de terminais SPARQL registrados.
- As fontes de dados são descritas usando descrições de serviços que fornecem informações sobre as capacidades de uma fonte, ou seja, as restrições expressas com expressões de filtro SPARQL regulares.
- Estas restrições podem expressar, por exemplo, que uma fonte de dados armazena apenas dados sobre tipos específicos de recursos.
- Para fontes que suportam apenas padrões de acesso limitado, por exemplo, permitindo pesquisas sobre dados pessoais somente quando o usuário pode especificar o nome da pessoa de interesse, a descrição do serviço inclui tais padrões que devem ser incluídos na consulta.
- Para fornecer o otimizador de consulta com as estatísticas, descrições de serviço que contém o número total de triplas fornecidos por uma fonte de dados e informações, opcionalmente, para cada recurso, por exemplo, o número de triplas com um predicado específico e as seletividades (ligado sujeito / objeto) de um tripla teste padrão com um predicado específico.

SemWIQ.

- Usa estatísticas básicas para otimizar consultas SPARQL em uma configuração federada.
- Essas estatísticas incluem uma lista de classes e o número de instâncias de uma fonte de dados que fornece para cada classe, bem como uma lista de predicados e suas ocorrências.
- O sistema exige que cada entidade consultada tem um tipo afirmado, isto é, requer informações de tipo para cada variável objeto de uma consulta.
- Para otimização de consultas, o federador analisa a consulta, explorando estatísticas e tipos de informações para determinar fontes de dados registradas relevantes. O plano resultante é executado enviando subconsultas às fontes.

Sesame Federation SAIL.

- A Federação SAIL permite combinar virtualmente vários conjuntos de dados em um único conjunto de dados e foi um dos primeiros quadros federação disponíveis.
- A consulta do usuário é avaliada através da distribuição de subconsultas para os diferentes membros da federação.
- Os resultados parciais são então agregados localmente e devolvido ao utilizador.
- Não usa qualquer estratégia de seleção de fonte e, portanto, envia subqueries a todos os membros da federação, causando significativa sobrecarga de comunicação.

Avalanche

- É um sistema federado que decompõe uma consulta em chamadas “moléculas” (subconsultas).
- Em contraste com a utilização de estatísticas pré-computadas ou padrão de consultas SPARQL ASK, fontes relevantes são identificados usando um motor de busca da Web Semântica ou outro repositório online que fornece estatísticas sobre fontes.
- As fontes são consultadas para coletar informação estatística, por exemplo, a cardinalidade de variáveis não ligadas.
- Com base nas estatísticas retornadas o Avalanche calcula combinações de fontes e “moléculas” cujos dados em combinação fornece resultados da consulta.
- O processo de execução de consulta proposta considera a seletividade de módulos para que os módulos seletivos sejam avaliadas primeiro. Execução da consulta é interrompida depois de ter devolvido o primeiro k únicas.

FedX

- É uma estrutura para acesso transparente às fontes de dados através de uma federação.
- Ele estabelece uma camada de federação que minimiza o número de pedidos por jobs, e usando técnicas de otimização algébricas baseadas em regras, com o objetivo de avaliar partes das consultas;
- Selecionando primeiro as execuções dos primeiros filtros para reduzir o tamanho dos resultados intermediários;
- FEDX aplica pipelining para computar os resultados o mais rápido possível e faz uso de sofisticadas estratégias de junção de execução com base no Semi-joins distribuídos.
- O sistema identifica ainda situações em que uma consulta pode ser dividida em grupos chamados exclusivos.
- Todas estas técnicas de optimização são aplicados automaticamente e não necessita de qualquer interação com o utilizador.

Optimization Techniques

- Em uma configuração federada com fontes de dados distribuídas, é importante otimizar a consulta de tal forma que o número de solicitações intermediárias seja minimizadas, ao mesmo tempo em que garante a rápida execução das solicitações individuais;
- E com base em um padrões de gráficos básicos (*basic graph patterns* - BGP) que é um conjunto de padrões de triplas, sendo uma tripla (sujeito, predicado, objeto) com variáveis em zero ou mais posições.

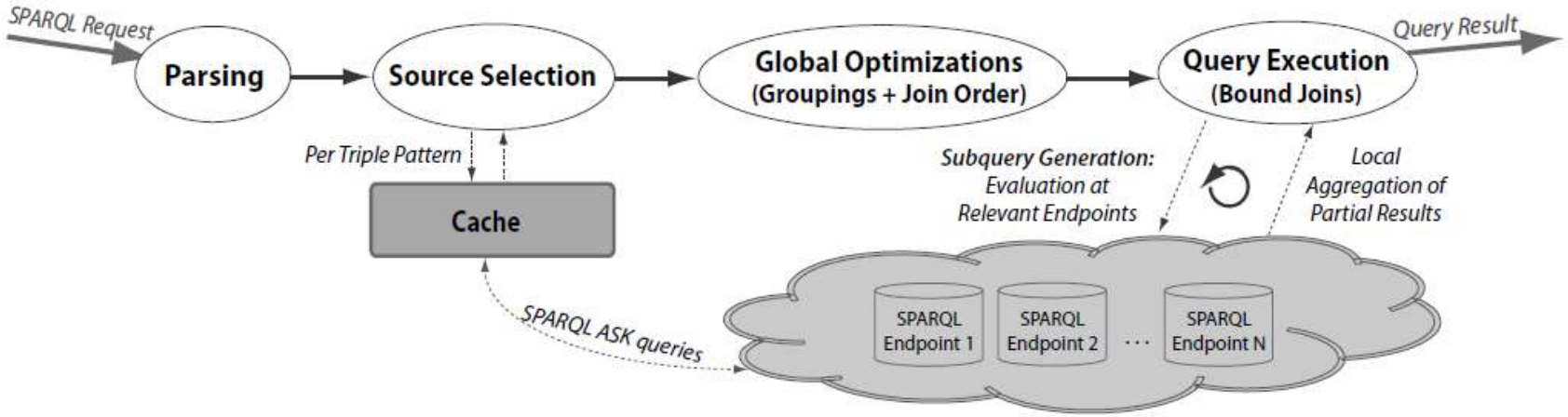
Optimization Techniques

- Avaliação de um consulta SPARQL:

1 - Todos os padrões de triplas são avaliados individualmente e completamente contra cada ponto final na federação e o resultado da consulta é construído localmente no servidor;

2 -Um motor avalia a consulta iterativamente padrão por padrão, ou seja, começando com um único padrão de tripla e substituindo mapeamentos a partir do padrão na etapa de avaliação subsequente, assim, avaliar a consulta em um aninhado *nested loop join fashion* (NLJ).

Federated Query Processing Model



Source Selection

- Os padrões de triplas de uma consulta SPARQL precisam ser avaliados somente nas fontes de dados que podem contribuir com resultados.
- Para identificar essas fontes relevantes, o FedX propõe uma técnica efetiva, que não requer metadados pré-processados: antes de otimizar a consulta, o mecanismo envia consultas SPARQL ASK para cada padrão de tripla para os membros da federação e, com base nos resultados, anota cada padrão. Na consulta com sua (s) fonte (s) relevante (s).
- Embora essa técnica possivelmente superestime o conjunto de fontes de dados relevantes (por exemplo, para (? S, rdf: tipo,? O) qualquer fonte de dados provavelmente coincidirá durante a seleção da fonte, mas durante a avaliação de junção com mapeamentos reais substituídos por? S e? O Pode não haver resultados), em consultas práticas, muitos padrões de triplas são específicos para uma única fonte de dados.
- Observe também que o FedX usa um cache para lembrar a informação de proveniência binária (ou seja, se a fonte S é relevante / irrelevante para um padrão de tripla) para minimizar o número de consultas ASK remotas.

Join Ordering

- A ordem de união determina o número de resultados intermediários e é, portanto, um fator altamente influente para o desempenho da consulta.
- Para a configuração federada, o FedX usa um otimizador de junção baseado em regras, que ordena uma lista de argumentos de junção (ou seja, padrões de triplas ou grupos de padrões de triplas) de acordo com uma estimativa de custo baseada em heurística.
- O algoritmo utiliza uma variação da técnica de contagem variável proposta em e está representada no Algoritmo.
- Seguindo uma abordagem iterativa, ele determina o argumento com o menor custo dos itens restantes (linha 6-12) e o anexa à lista de resultados (linha 14).
- Para a estimativa de custo (linha 7), o número de variáveis livres é contado considerando variáveis já ligadas, isto é, as variáveis que são ligadas por um argumento de junção que já está ordenado na lista de resultados.
- Além disso, o FedX aplica uma heurística que prefere grupos exclusivos pois estes em muitos casos podem ser avaliados com a maior seletividade.

Join Ordering

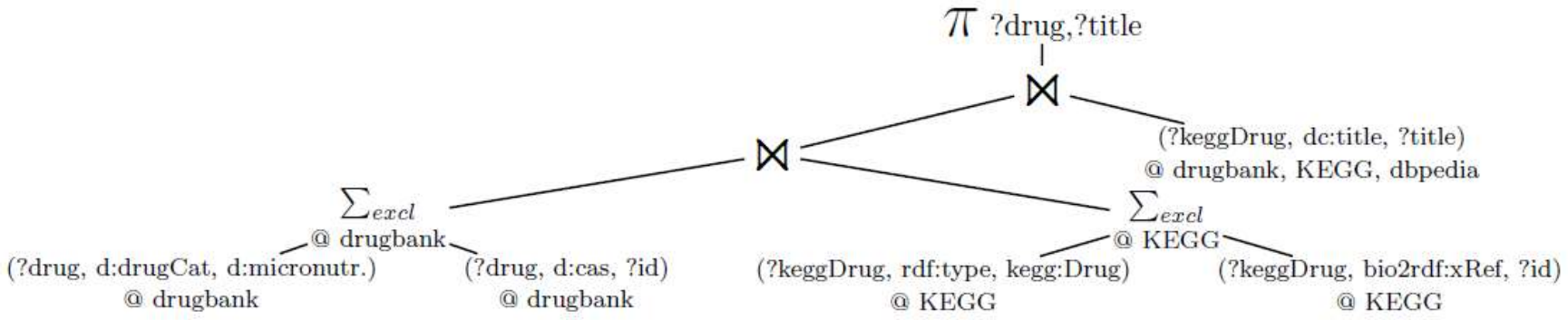
Algorithm 14.1 Join order optimization

```
1: function ORDER(joinargs: list of n join arguments)
2:     left ← joinargs
3:     joinvars ← ∅
4:     for i = 1 → n do
5:         mincost ← MAX_VALUE
6:         for all j ∈ left do
7:             cost ← ESTIMATECOST(j, joinvars)
8:             if cost < mincost then
9:                 arg ← j
10:                mincost ← cost
11:            end if
12:        end for
13:        joinvars ← joinvars ∪ VARS(arg)
14:        result[i] ← arg
15:        left ← left − arg
16:    end for
17:    return result
18: end function
```

Exclusive Groups

- Alto custo no processamento de consultas federadas resulta da execução local de junções no servidor, em particular quando as junções são processadas em um modo de loop aninhado.
- Para minimizar esses custos, o FedX apresenta os chamados grupos exclusivos, que desempenham um papel central no otimizador FedX.
- Um grupo exclusivo consiste em um conjunto de padrões de triplas que possuem a mesma fonte relevante.
- Esse grupo de padrões de triplas pode ser avaliado (combinado) em uma consulta conjuntiva na fonte relevante, já que nenhum outro parâmetro pode influenciar os resultados.

Exclusive Groups



Bind Joins

- Ao computar as associações numa forma de anel aninhado em bloco, isto é, como uma semi join distribuído, é possível reduzir o número de pedidos por um fator equivalente ao tamanho de um bloco, e seguir referido como uma sequência de entrada.
- A idéia geral dessa otimização é agrupar um conjunto de mapeamentos em uma única subconsulta usando as construções de SPARQL UNION.
- Esta subconsulta agrupada é então enviada para as fontes de dados relevantes numa única solicitação remota.
- Finalmente, algum pós-processamento é aplicado localmente para manter a correção.

Bind Joins

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

Conclusão

- Já foram publicados primeiros relatórios sobre a aplicação com êxito de técnicas de federação a federações RDF de larga escala numa gama trimestral de mil milhões.
- Apesar dos sucessos na aplicação de federações em grande escala, é justo dizer que as implementações atuais ainda têm um longo caminho a percorrer para serem tão confiáveis como abordagens centralizadas para o processamento de consultas.
- E a grande questão que tem que ser resolvida é a transferência de dados.

Perguntas?

Questionário:

1 - O Federated Query Processing over Linked é baseado em 3 categorias principais, discorra sobre o beneficio de utilizar cada uma das **categorias** apresentadas.

2 - Atualmente em RDF há várias **técnicas** de otimização de federação implementadas, com base no que foi apresentado discorra, qual foi a técnica mais interessante no seu ponto de vista e explique os pontos favoráveis da mesma.

Luizbrt989@gmail.com