

# Linked Data Management

## Capítulo 9: Experiences with Virtuoso Cluster RDF Column Store

Jean Souza

18 de outubro de 2016

# Proposta do artigo

- Avanços no suporte a escalabilidade com Virtuoso e seus recursos para armazenamento de colunas RDF
- Avaliação do resultado no Berlin SPARQL Benchmark
- Amostra da avaliação: micro benchmark
  - 10 bilhões de triplas
- Avaliação experimental: completo (transacional -Explore, analítica - BI workloads)
  - 50 e 150 bilhões de triplas

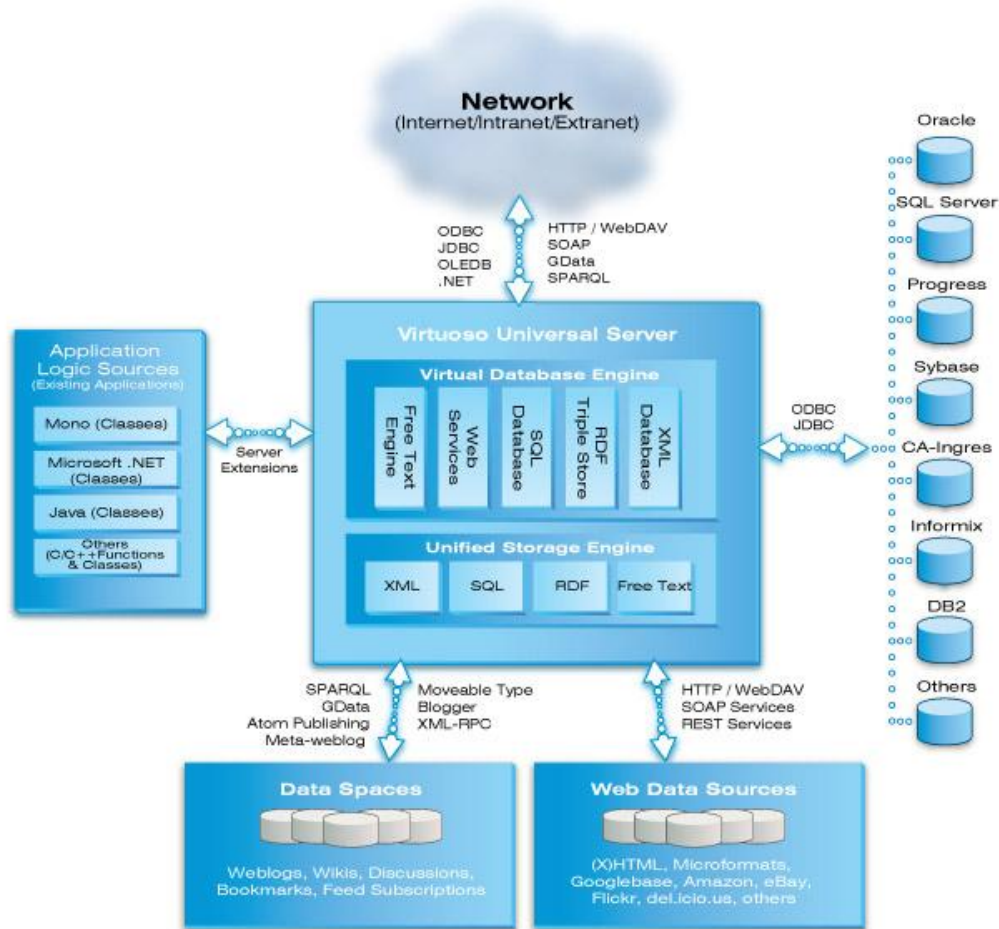
# Berlin SPARQL Benchmark (BSBM)

- Desenvolvido em 2008
- Uma das primeiras fontes públicas e aberta
- Compara desempenho de sistemas de armazenamento que estão expostos ao SPARQL

## Virtuoso

- Solução moderna para acesso a dados, integração e gerenciamento de banco de dados relacionais.
- Possibilita ser executado em modo cluster
  - Banco de dados lógico
  - Servido por um conjunto de processos de propagação
  - Distribuído por um conjunto de máquinas

# Virtuoso



## Virtuoso

- Inicialmente desenvolvido para SGBDs orientado a transações linha a linha
- Posteriormente suporte a RDF e inferência através de SPARQL
- Atualmente devido ao suporte RDF tendo em vista sua alta eficiência de espaço de compressão no armazenamento das triplas o Virtuoso também trabalha coluna a coluna (Virtuoso 7)

# Coluna a coluna e linha a linha

Subject	Property	Object
Isaac	rdf:type	student
Isaac	Age	26
Rachel	rdf:type	post-doc
Isaac	Year	3rd
Rachel	Office	925
Rachel	Age	29

The following property table can be created:

Subject	rdf:type	Age	Year	Office
Isaac	student	26	3rd	NULL
Rachel	post-doc	29	NULL	925

# Esquema armazenamento de RDF

- O Virtuoso considera (S)ujeito - (P)redicado - (O)bjeto pertencentes a um (G)rafo.
- Conjunto é um “quadrângulo” e não tripla
- O G representa um grafo nomeado por uma URI

```
create table RDF_QUAD (  
  G IRI_ID_8,  
  S IRI_ID_8,  
  P IRI_ID_8,  
  O any,  
  primary key (P, S, O, G) column  
)
```



# Exemplo dump grafo nomeado

```
$ /opt/virtuoso/bin/isql llll
Connected to OpenLink Virtuoso
Driver: 06.01.3127 OpenLink Virtuoso ODBC Driver
OpenLink Interactive SQL (Virtuoso), version 0.9849b.
Type HELP; for help and EXIT; to exit.
SQL> dump_one_graph ('http://daas.openlinksw.com/data#', './data_', 1000000000);
Done. -- 1438 msec.
```

# Exemplo Inserção através protocolo HTTP

```
curl -i -d "INSERT IN GRAPH <http://mygraph.com>
{ <http://www.openlinksw.com/dataspace/kidehen@openlinksw.com#this>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  <http://rdfs.org/sioc/ns#User> .
  <http://www.openlinksw.com/dataspace/kidehen@openlinksw.com#this>
  <http://www.w3.org/2000/01/rdf-schema#label>
  <Kingsley Uyi Idehen> .
  <http://www.openlinksw.com/dataspace/kidehen@openlinksw.com#this>
  <http://rdfs.org/sioc/ns#creator_of>
  <http://www.openlinksw.com/dataspace/kidehen@openlinksw.com/weblog/kidehen@openlinksw.com%27s%20BLOG%20%5B127%5D/1300-
}" -u "demo:demo" -H "Content-Type: application/sparql-query" http://localhost:8890/DAV/home/demo/test/myrq
```

# Exemplo resultado através de consulta SPARQL

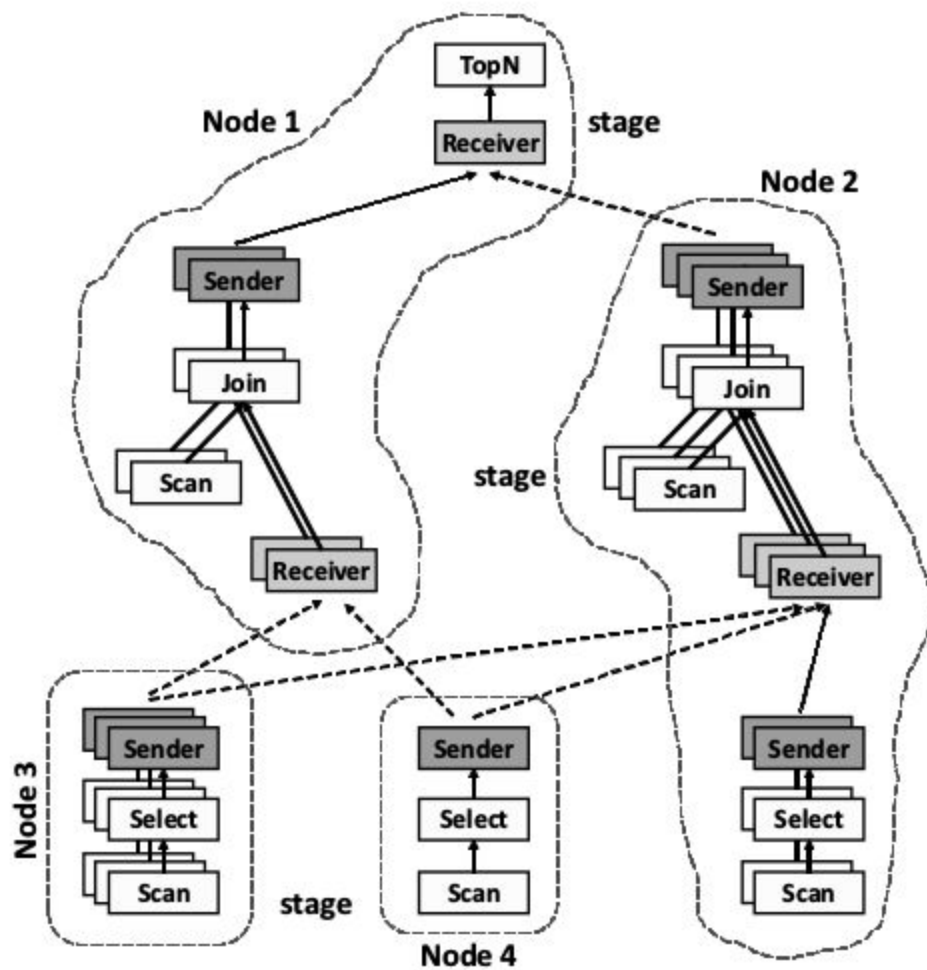
Grafo: <http://mygraph.com>

Query: SELECT \* WHERE {?s ?p ?o}

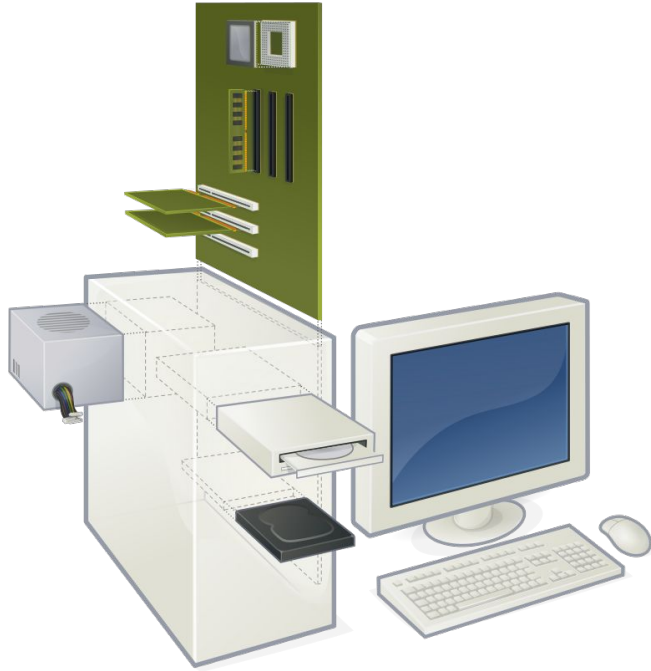
Sujeito	Predicado	Objeto
<a href="http://www.openlinksw.com/data-space/kidehen@openlinksw.com#this">http://www.openlinksw.com/data-space/kidehen@openlinksw.com#this</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://rdfs.org/sioc/ns#User">http://rdfs.org/sioc/ns#User</a>
<a href="http://www.openlinksw.com/data-space/kidehen@openlinksw.com#this">http://www.openlinksw.com/data-space/kidehen@openlinksw.com#this</a>	<a href="http://www.w3.org/2000/01/rdf-schema#label">http://www.w3.org/2000/01/rdf-schema#label</a>	Kingsley
<a href="http://www.openlinksw.com/data-space/kidehen@openlinksw.com#this">http://www.openlinksw.com/data-space/kidehen@openlinksw.com#this</a>	<a href="http://rdfs.org/sioc/ns#creator_of">http://rdfs.org/sioc/ns#creator_of</a>	<a href="http://www.openlinksw.com/data-space/kidehen@openlinksw.com/weblog/kidehen@openlinksw.com%27s%20BLOG%20%5B127%5D/1300">http://www.openlinksw.com/data-space/kidehen@openlinksw.com/weblog/kidehen@openlinksw.com%27s%20BLOG%20%5B127%5D/1300</a>

# Avaliação da consulta

- Unidade básica para avaliação da consulta
  - Fragmento Distribuído (DFG)
    - Sequência de junção
    - Passos consecutivos podem não estar na mesma partição
    - Troca de mensagem entre todas as fatias (stages) que possuem dado referente a operação solicitada



# Micro Benchmark - Amostra de consulta



2 máquinas configuradas

- Xeon E5-2630
- 192g RAM

2 processos por máquina

- 12 fatias de dados
- 12 threads
- 10 bilhões de triplas

# Avaliação da amostra de consulta - Micro Benchmark

- Query

```
SELECT ?productType (COUNT(?review) AS ?reviewCount) {  
  ?product bsbm:producer ?producer .  
  ?producer bsbm:country <http://downlode.org/rdf/iso-3166/countries#AT> .  
  ?review bsbm:reviewFor ?product .  
  ?review rev:reviewer ?reviewer .  
  ?reviewer bsbm:country <http://downlode.org/rdf/iso-3166/countries#US> .  
  ?product a ?productType .  
  ?productType a bsbm:ProductType . }  
GROUP BY ?productType  
ORDER BY DESC 2;
```

# Avaliação da amostra de consulta - Micro Benchmark

- Resultados

	seconds	cpu%	msg/sec	bytes/msg
Single:	7.0	1578	n. a.	n. a.
Single 10K vec	32.5	1745	n. a.	n. a.
2-Cluster	4.96	3458	942K	1.6K
2-Cluster no DFG	10.94	585	16K	132K
2-Cluster 500K vec	4.3	2851	599	2.6K
2-Cluster 10K vec	20.3	2411	7.7M	0.3K

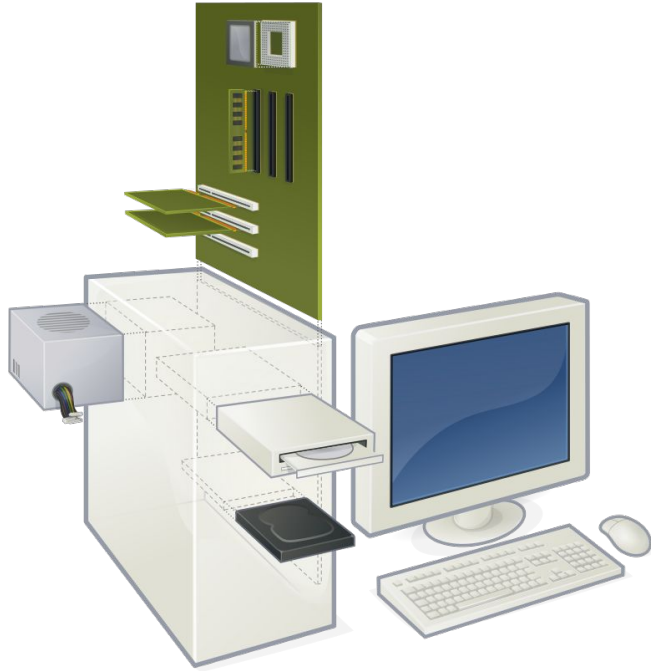
TABLE 9.1: BSBM: BI Q1 micro-benchmark: vector size and DFG



# Avaliação Experimental

- Avaliações do Virtuoso existentes
  - BSBM version 1 (July 2008) with 100 million triple datasets
  - BSBM version 2 (Nov 2009) with 200 million triple datasets
  - BSBM version 3 (Feb 2011) with 200 million triple datasets
- Avaliação proposta
  - 750 vezes mais dados

# Avaliação Experimental



8 máquinas configuradas

- Linux
- 2 CPUs (8 núcleos, hyperthreading, 2GHz, Sandy Bridge (arquitetura))
- 256GB de RAM
- 3 HDs magnéticos (SATA) em RAID 0 (180MB taxa de transf.)
- Alta largura de banda  
50 a 150 bilhões de bytes

# Avaliação Experimental

- Gerador de dados BSBM single-thread
  - Levaria semanas para produzir 150 bilhões de triplas
  - Alteração do gerador de dados
  - Geração de dados escalável, execução em diferentes máquinas em paralelo
- Avaliação proposta
  - 750 vezes mais dados

# Avaliação Experimental

- Gerador de dados BSBM modificado

nr triples	Size (.ttl)	Size (.gz)	Database Size	Load Time
50 Billion	2.8 TB	240 GB	1.8 TB	6h 28m
150 Billion	8.5 TB	728 GB	5.6 TB	n/a

- 150 bilhões não foi possível medir o tempo de carregamento
- Tiveram problemas com hardware
- Intervenção manual

# Avaliação Experimental

- Métrica utilizada para avaliação
  - AQET : Tempo médio de execução de consulta (segundos)
  - $AQET(q) = (\text{tempo de execução total de } q) / (\text{número de execuções de } q)$

# Avaliação Experimental

- Consulta (Explore - transacional)
- Procura semelhanças a um determinado produto

```
SELECT DISTINCT ?product ?productLabel
WHERE {
  ?product rdfs:label ?productLabel .
  FILTER (%ProductXYZ% != ?product)
  %ProductXYZ% bsbm:productFeature ?prodFeature .
  ?product bsbm:productFeature ?prodFeature .

  %ProductXYZ% bsbm:productPropertyNumeric1 ?origProperty1 .
  ?product bsbm:productPropertyNumeric1 ?simProperty1 .
  FILTER (?simProperty1 < (?origProperty1 + 120) &&
          ?simProperty1 > (?origProperty1 - 120))
```

# Avaliação Experimental

- Resultados da consulta

	50 Billion triples		150Billion triples	
	Single-Client	4-Clients	Single-Client	4-Clients
runtime	931s (100 runs)	15s (1run)	1894s (100 runs)	29s (1 run)
Tput	4.832M	11.820M	7.126M	18.386M
	AQET	AQET	AQET	AQET
Q1	0.066s	0.415s	0.113s	0.093s
Q2	0.045s	0.041s	0.066s	0.086s
Q3	0.112s	0.091s	0.111s	0.116s
Q4	0.156s	0.102s	0.308s	0.230s
Q5	3.748s	6.190s	8.052s	9.655s
Q7	0.155s	0.043s	0.258s	0.360s
Q8	0.100s	0.021s	0.188s	0.186s
Q9	0.011s	0.010s	0.011s	0.011s
Q10	0.147s	0.020s	0.201s	0.242s
Q11	0.005s	0.004s	0.006s	0.006s
Q12	0.014s	0.019s	0.013s	0.010s

# Conclusão

- O artigo apresentou o recurso de armazenamento e "clusterização" por coluna disponível no Virtuoso.
- Examinou seu desempenho usando benchmark BSBM
- Esta nova arquitetura de cluster permite o desempenho da gestão de RDF em uma escala sem precedentes
- Resultados apresentados marcam 750 vezes mais do que foi registrado nos resultados anteriores



# Conclusão

Apresentam uma rápida execução de consultas de pesquisa de índices (Explore - transacional) e também consultas analíticas complexas (BI)

# Exercícios

1. Conforme script de criação de tabela no Virtuoso, selecione a alternativa que representa o trecho de código das colunas que formam uma tripla RDF.

A) create table RDF\_QUAD

B) primary key (P, S, O, G)

C) S IRI\_ID\_8, P IRI\_ID\_8, O any

```
create table RDF_QUAD (  
  G IRI_ID_8,  
  S IRI_ID_8,  
  P IRI_ID_8,  
  O any,  
  primary key (P, S, O, G) column  
)
```

## Exercícios

2. O que é o Berlin SPARQL Benchmark ?
  - A) BSBM é uma linguagem de consulta em banco de dados transacional.
  - B) Compara desempenho de sistemas de armazenamento que estão expostos ao SPARQL
  - C) Um Benchmark para comparar diferentes linguagens de programação