TRINITY Project
Microsoft Research Asia

Rebeca Schroeder

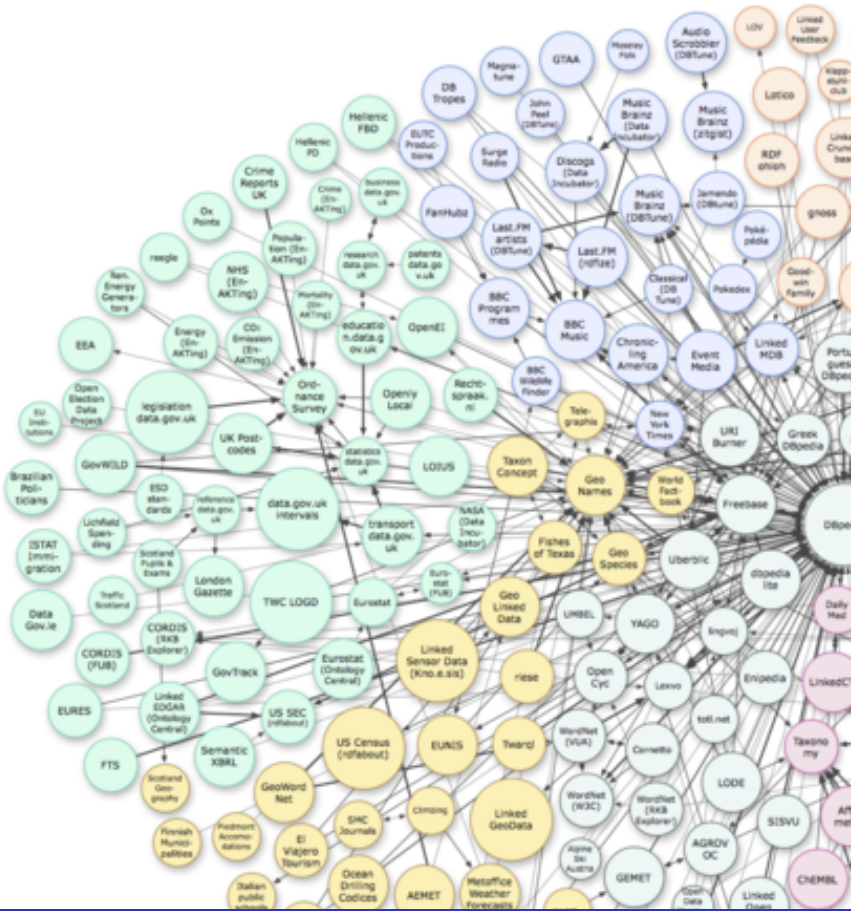
Universidade Federal do Paraná

Curitiba, 17 de setembro de 2013

Desafios da era Big Data

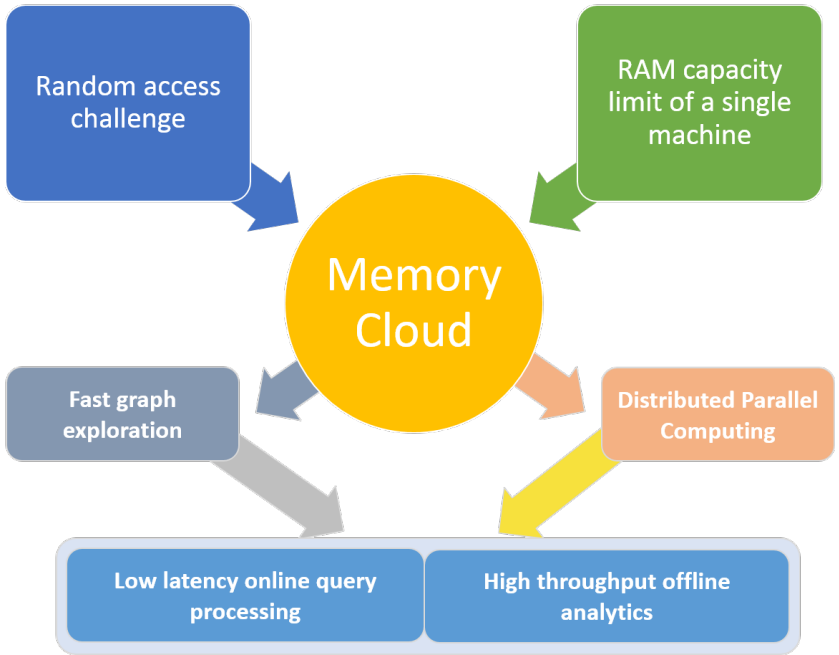
BIG DATA e seus 3 V's:

- **V**ariedade:
  - Dados complexos e semanticamente ricos
  - Fim do "*One size fits all*"
- **V**olume:
  - Grandes repositórios, Redes sociais, grafo da Web
- **V**elocidade



Trinity

Sistema de grafos de propósito geral:

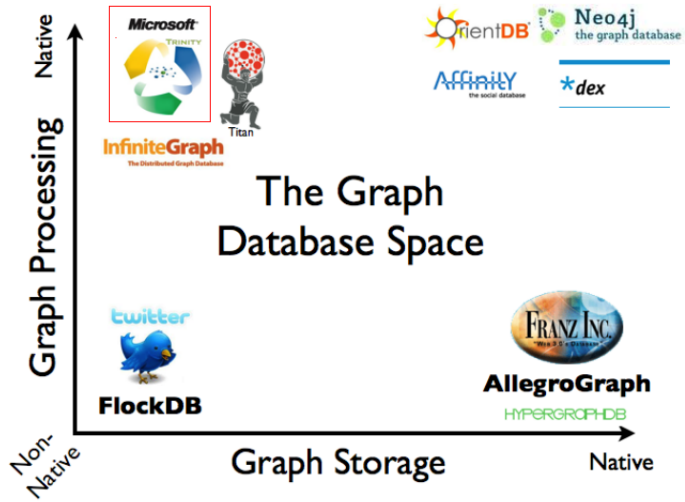


"A graph database can be defined as it can find its neighbor's or its child's nodes without using indexes. If a database system can do that, then we can call it a native graph database system" (Haixun Wang - membro da equipe Trinity)[7]

Trinity

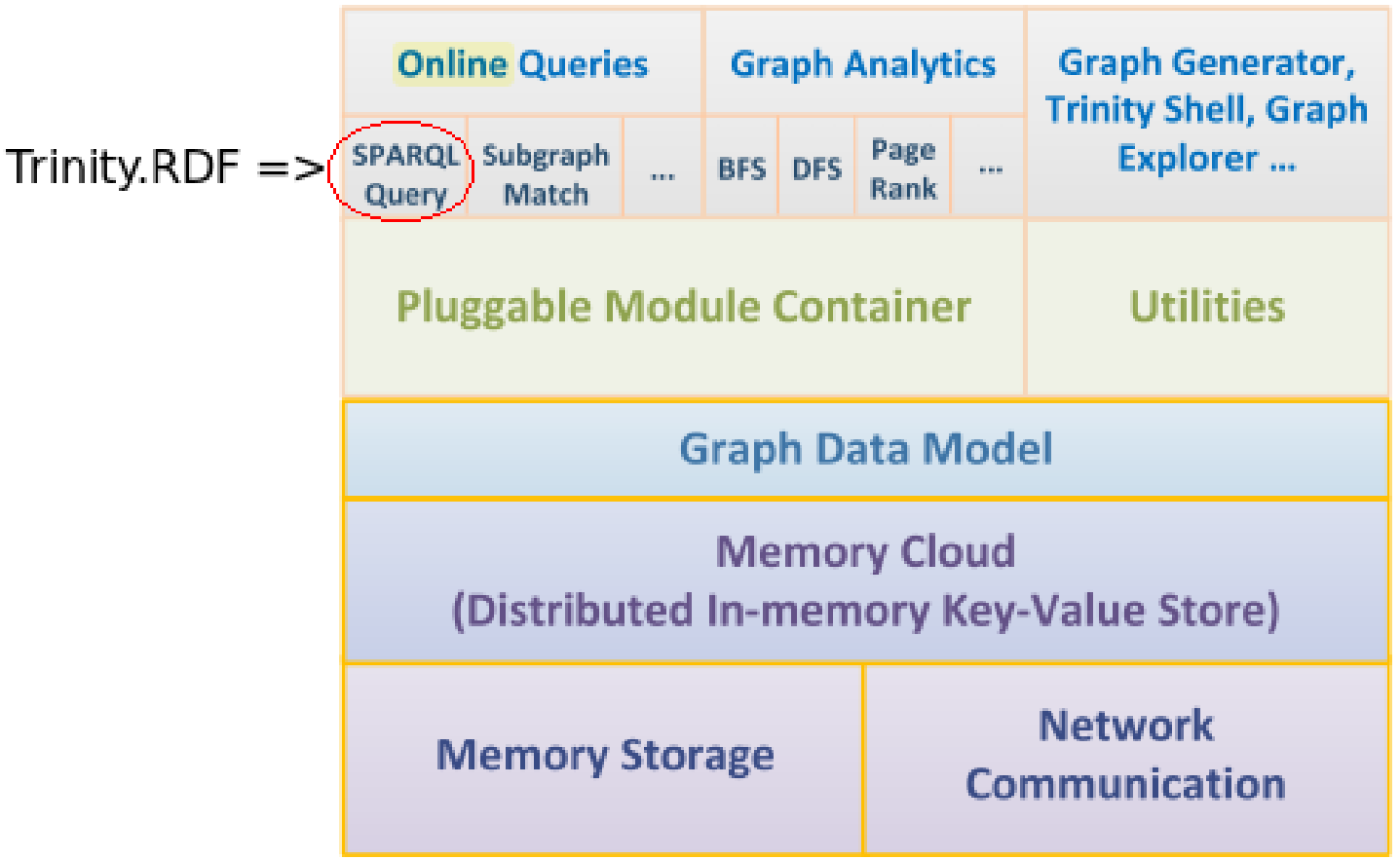
Classificação:

- Cargas de trabalho: *online* e *offline*
- Processamento: paralelo e distribuído
- Método de acesso: *graph exploration*
- Modelo lógico: nativo
- Armazenamento: *memory cloud*
- Transacional: não
- Indexação: por predicados no *Trinity.RDF*
- Linguagem de consulta: SPARQL no *Trinity.RDF*



[4]

Trinity - Visão Geral



[5]

Agenda

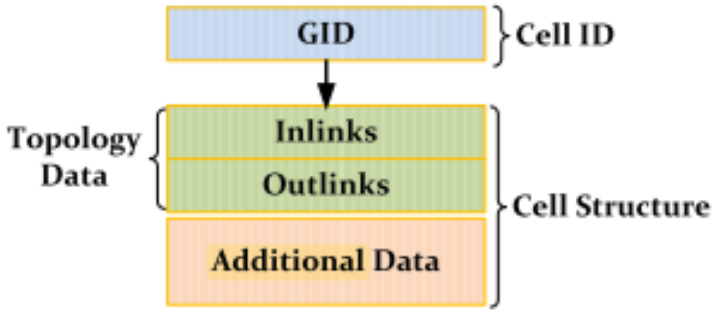
Modelo de Dados

Modelo Lógico:

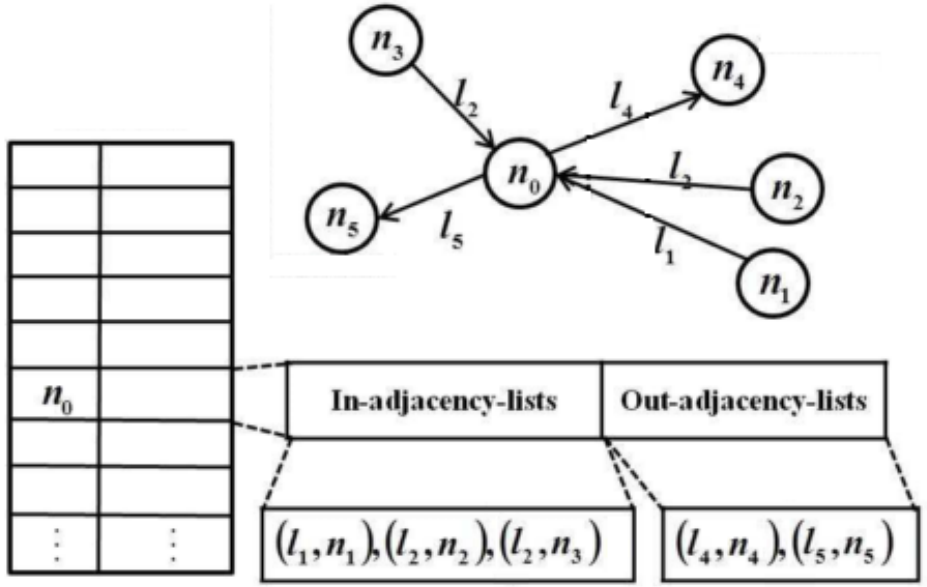
- Suporta grafos direcionados, não-direcionados e hipergrafos
- Elementos: nós e aresta direcionadas

Modelo de armazenamento:

- Célula: par chave-valor representa um nó e suas arestas



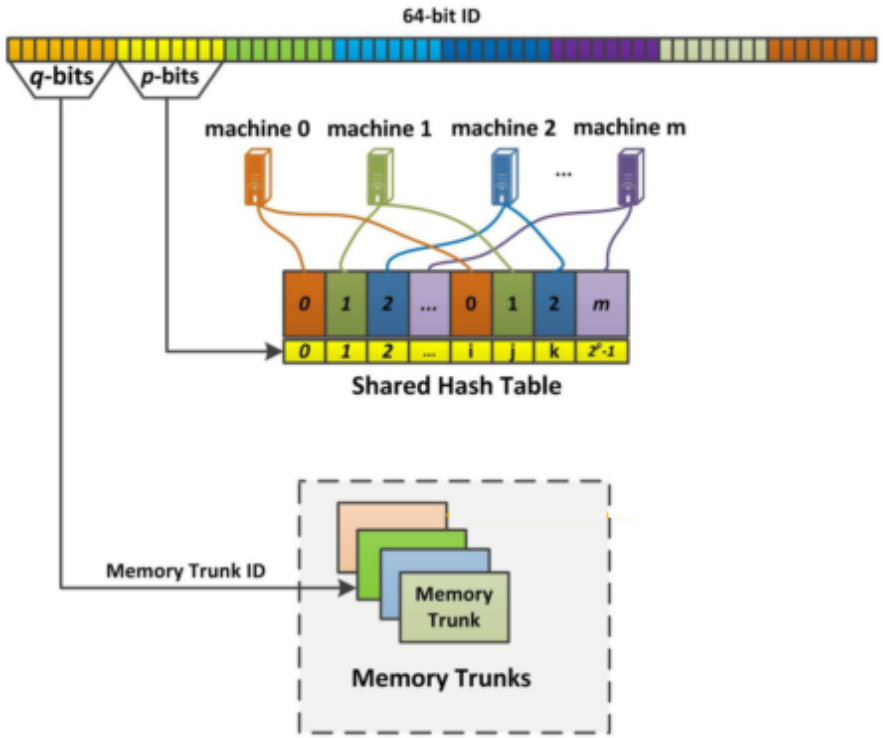
[5]



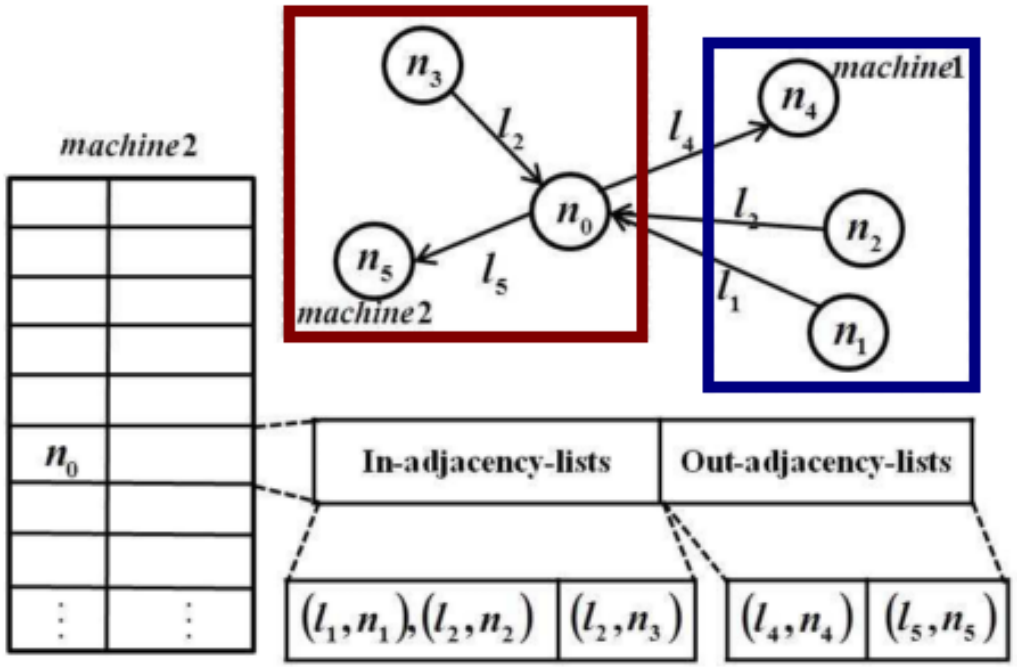
Memory Cloud

Repositório chave-valor em memória:

- Armazenamento e Acesso a células
- Replicação
- Controle de concorrência



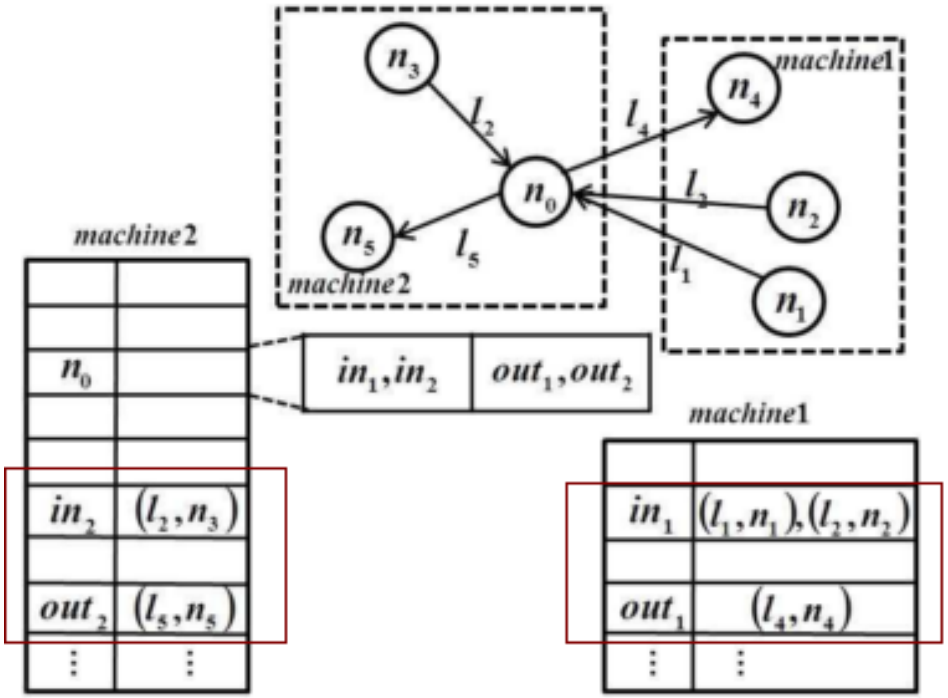
Memory Cloud



[5]

Particionamento do grafo

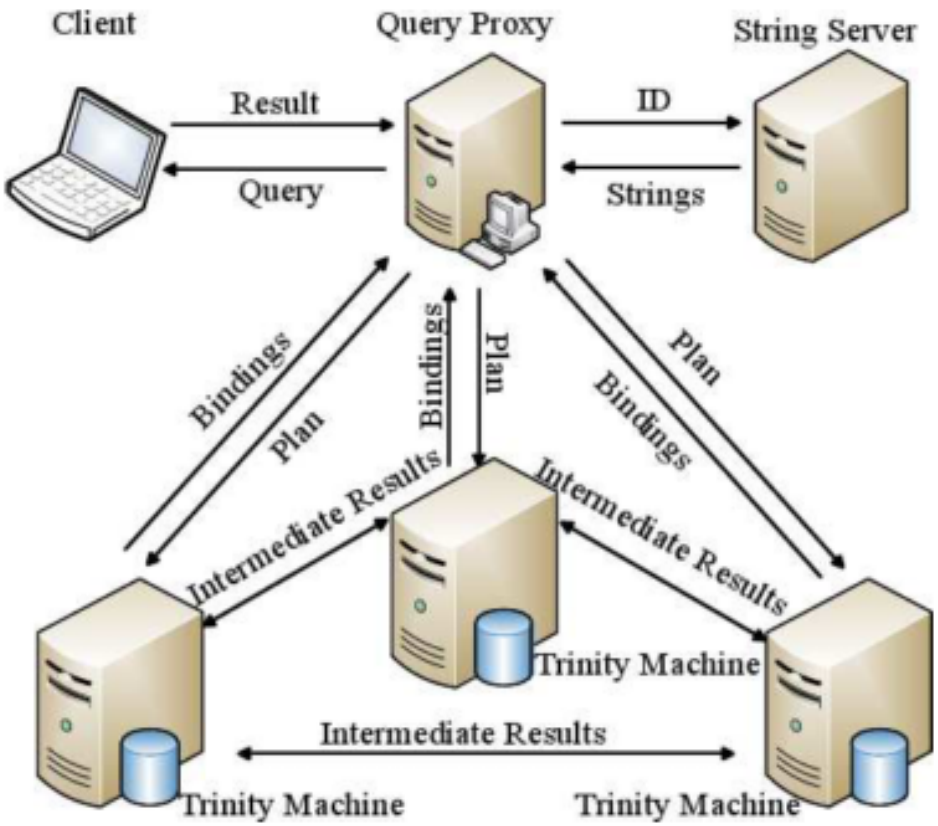
- Grau de nós segue uma distribuição com a de Pareto
- Aplica um limiar sobre grau de nós
- Agrupa as adjacências dos nós que excedem o limiar



Indexação

- Índices locais (para cada nó que excede o limiar do grau):
  - Ordena pares (predicado, id-no) da lista de adjacência
  - Cria um índice de agregação sobre o predicado
- Índice global:
  - Habilita identificar nós que estão conectadas a um predicado
  - Cada máquina i armazena pares no formato (predicate, $\langle subject - list_i, object - list_i \rangle$)

Arquitetura de Processamento



[6]

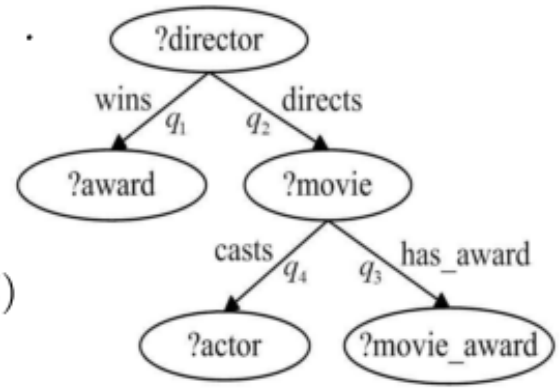
Exploração do Grafo

Exploração é desempenhada paralelamente sobre cada máquina:

1. Consulta é decomposta em uma sequêcia de padrões de triplas
2. Encontra os casamentos para padrões de forma sequencial

```
SELECT ?movie, ?actor WHERE{
 ?director wins ?award .
 ?director directs ?movie .
 ?movie has_award ?movie_award .
 ?movie casts ?actor .}
```

- q_1: (*?director wins ?award*)
- q_2: (*?director directs ?movie*)
- q_3: (*?movie has_award ?movie_award*)
- q_4: (*?movie casts ?actor*)

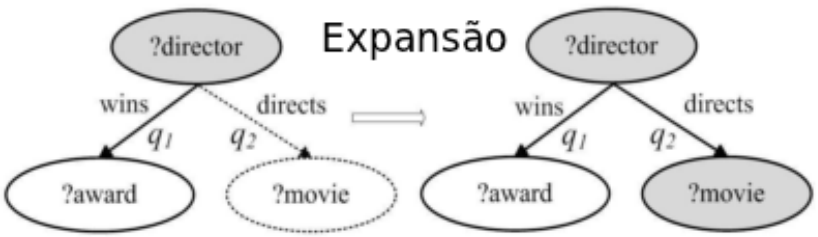


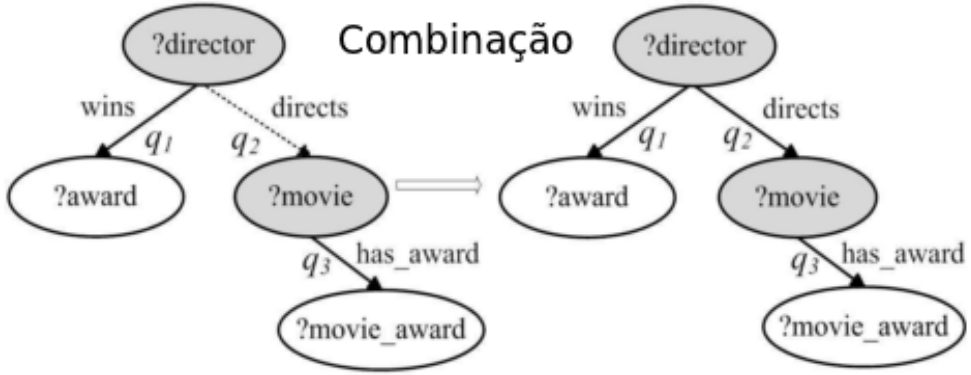
[6]

Otimização do Plano de Consulta

Objetivo: reduzir quantidade de resultados intermediários

- Melhor sequência de exploração das triplas
- O custo para cada padrão é proporcial ao tamanho de seu resultado
- Encontra a sequência ótima por programação dinâmica
- O grafo de exploração é construído a partir da sua expansão ou combinação com outros grafos





Trabalhos Relacionados

Exploração versus Junção

- Tempo de resposta para consultas em ms
- *Dataset* com 15 milhões de triplas

	D1	D2	D3	D4	D5	D6	D7	D8
Trinity.RDF	**7**	220	**5**	**7**	**8**	**21**	**13**	**28**
RDF-3X (In Memory)	15	**79**	14	18	22	34	68	35
BitMat (In Memory)	335	1375	209	113	431	619	617	593
RDF-3X (Cold Cache)	522	493	394	498	366	524	458	658
BitMat (Cold Cache)	392	1605	326	279	770	890	813	872

[6]

Trabalhos Relacionados

Exploração versus Junção

- Tempo de resposta para consultas em ms
- *Dataset* com 3,17 bilhões de triplas

	S1	S2	S3	S4	S5	S6	S7
Trinity.RDF	**12**	10	**31**	**21**	**23**	**33**	**27**
RDF-3X (Warm Cache)	108	8407	27428	62846	32	260	238
RDF-3X (Cold Cache)	5265	23881	41819	91140	1041	3065	1497
MapReduce-RDF-3X (Warm Cache w/o MapReduce)	132	**8**	4833	6059	24	1931	2732
MapReduce-RDF-3X (Cold Cache w/o MapReduce)	2617	661	13755	18712	801	4347	7950
MapReduce-RDF-3X (MapReduce)	N/A	N/A	39928	39782	N/A	33699	33703

[6]

Trabalhos Relacionados

	Native graphs	Online query processing	Memory based exploration	Distributed parallel processing
Neo4j	Yes	Yes	No	No
HyperGraphDB	No	Yes	No	No
FlockDB	No	Yes	No	Yes
MapReduce	No	No	No	Yes
PEGASUS	No	No	No	Yes
Pregel	No	No	No	Yes
Trinity	Yes	Yes	Yes	Yes

[5]

Referências

1 Angles, R., Gutierrez, C. Survey of Graph Database Models. TR-2005-10.

2 Microsoft Research Asia. Trinity Manual. 2012

3 Microsoft Research Asia. Trinity homepage. http://research.microsoft.com/en-us/projects/trinity/default.aspx

4 Robinson, I. et al. Graph Databases. O'Reilly. 2013

5 Shao, B. et al. The Trinity Graph Engine. SIGMOD'13

6 Zeng, K. et al. A Distributed Graph Engine for Web Scale RDF Data. VLDB'13

7 Wang, H. Trinity on Hanselminutes weekly audio talk show. 2013 Li, Yatao