Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik

Evaluating Conjunctive Triple Pattern Queries over Large Structured Overlay Networks

Erietta Liarou, Stratos Idreos, and Manolis Koubarakis 2006.

Waled Almukawi.

Outline

- Motivation
- Terminologies
  - Conjunctive Queries
  - RDF
  - DHT
- System Model & Data Model
- Query Chain Algorithm
- Spread By Value Algorithm
- Experiments
- Future Works
- References

Motivation

- One of the most challenging and interesting open problems in the frontiers of P2P and Semantic Web is how to evaluate queries expressed in RDF query languages on top of P2P networks .
- Previous Rdfpeers used only atomic formula or disjoint RDF triple pattern.
- The conjunctive queries are interesting, since they provide a flexible methods that can deal effectively with a full functionality of existing RDF query .

Conjunctive Queries

- The conjunctive queries are simply the fragment of **first order logic** given by the set of **atomic formula** using **conjunction** ∧.
- A conjunctive query q formula :

?x1, . . . , ?xn : (s1, p1, o1) ^ (s2, p2, o2) ^ · · · ^ (sn, pn, on)

where **?x1, . . . , ?xn** are answer variables, each **(si, pi, oi)** is a triple pattern, and each variable ?xi appears in at least one triple pattern **(si, pi, oi)**.

Conjunctive queries (2)

- Example:

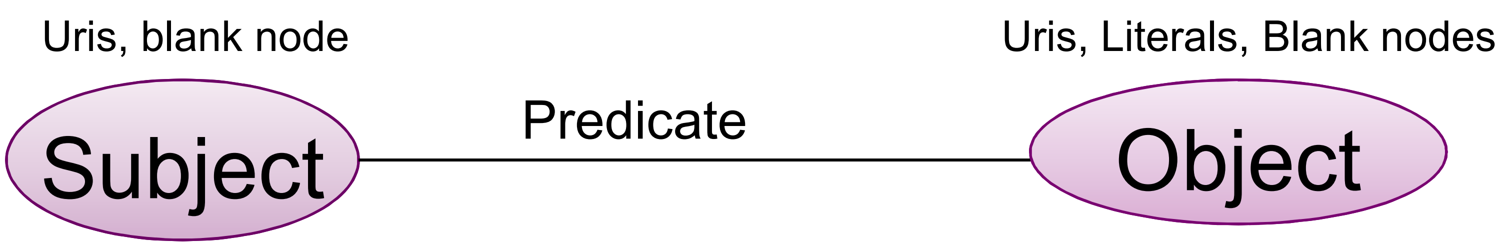
Q:ans(x,y):- country(Id,x),capital(y,x),continent(x,“Europa“)

Q‘:ans(x,y):-(?x,“type“,“country“),(?x,“has-capital“,?y),
(?x, continent,“Europa“).

Resource Description Framework (RDF)

- RDF is a framework for describing Web resources, such as the title, author, content …e.t.c.

- RDF Statements is The combination of a Resource, a Property, and a Property value forms a Statement (known as the subject, predicate and object of a Statement).



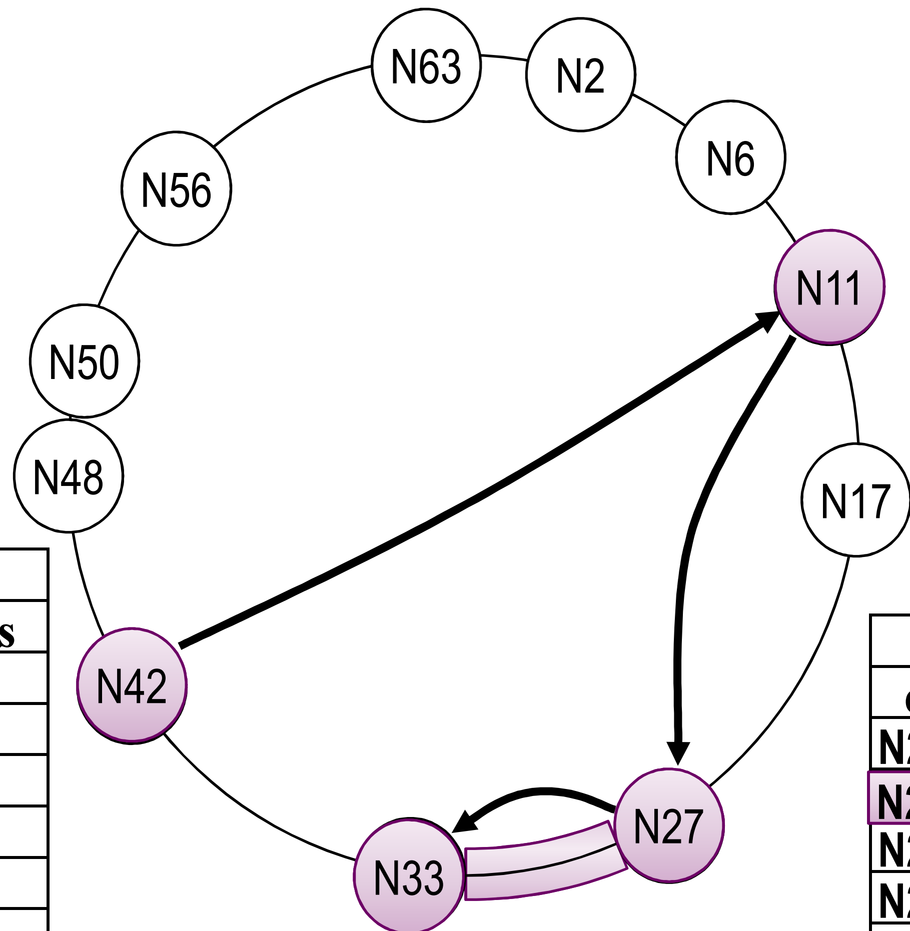
Distriuted Hash Tables (DHT)

- DHTs are an important class of P2P networks that offer distributed hash table functionality, and allow one to develop scalable, robust and fault-tolerant distributed applications.
- Distribute data over a large P2P network
- Can Quickly find any given item ni Log N nodes.

Look up for an Item in the DHT is performed in O(Log N) nodes.

System Model (ref:[4])

Key 28



finger table		
distan	succ.	keys
N11+0	N11	11
N11+1	N17	12
N11+2	N17	13
N11+4	N17	15
N11+8	N27	19
N11+16	N27	27
N11+32	N48	43

finger table		
distan	succ.	keys
N42+0	N42	42
N42+1	N48	43
N42+2	N48	44
N42+4	N48	46
N42+8	N50	50
N42+16	N63	58
N42+32	N11	10

finger table		
distan	succ.	keys
N27+0	N27	27
N27+1	N33	28
N27+2	N33	29
N27+4	N33	31
N27+8	N42	35
N27+16	N48	43
N27+32	N63	59

Data Model

- Each network node is able to describe in RDF the resources that it wants to make available to the rest of the network .
- It creates and inserts the RDF triples

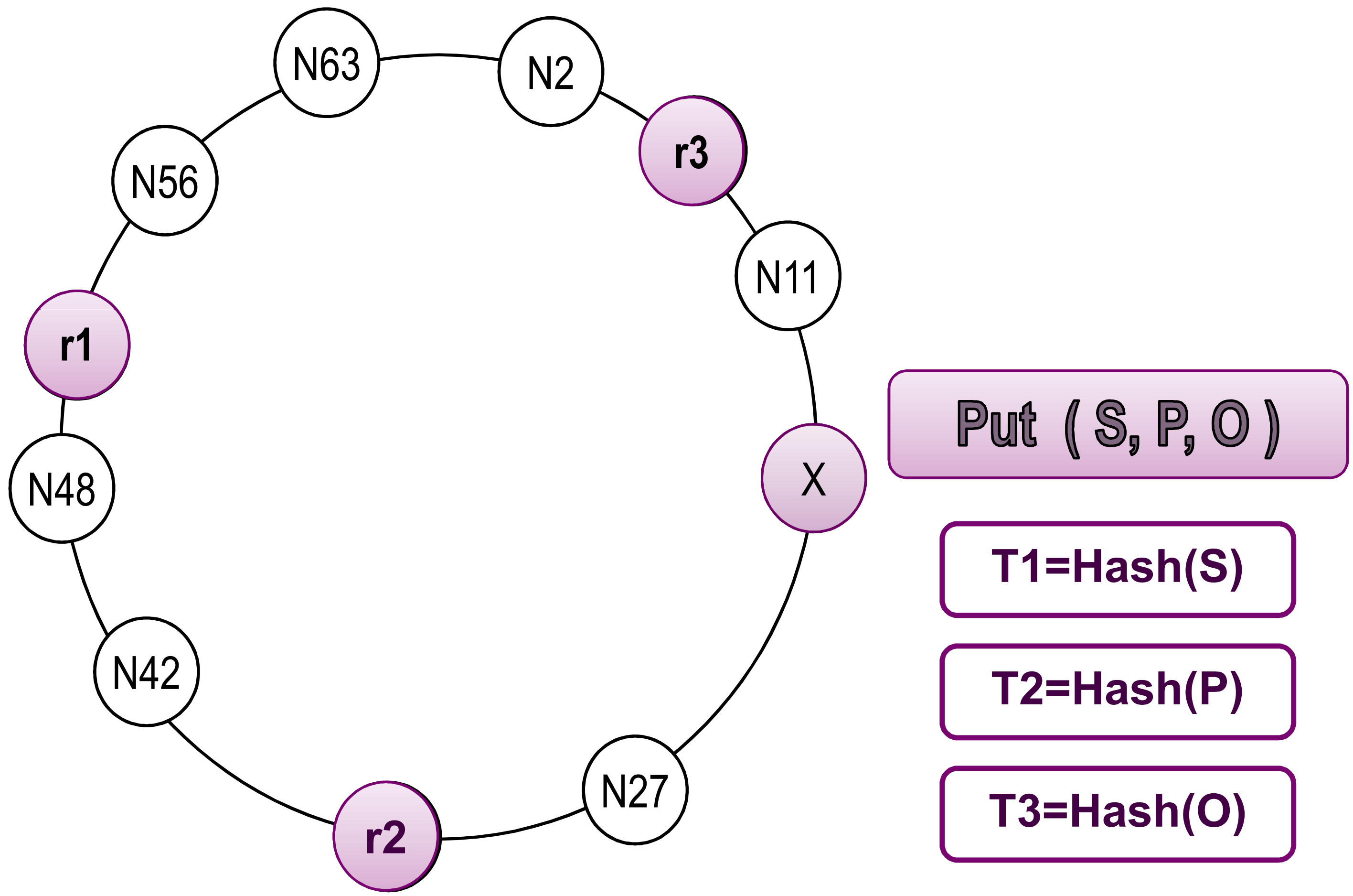
  (Subject, Predicate, Object).
  - The subject of a triple identifies the resource that the statement is about.
  - The predicate identifies a property or a characteristic of the subject.
  - The object identifies the value of the property.

Query Chain Algorithm

- The main idea of QC is that the query is evaluated by a chain of nodes.
- Intermediate results flow through the nodes of this chain
- Finally the last node in the chain delivers the result back to the node that submitted the query.
- Distribute the Load Balance by
  - Split the query into the triple patterns that it consists of.
  - Evaluate each triple pattern separately to a deferent node.
  - Intermediate result flow through these nodes as triple and partially satisfy the query.

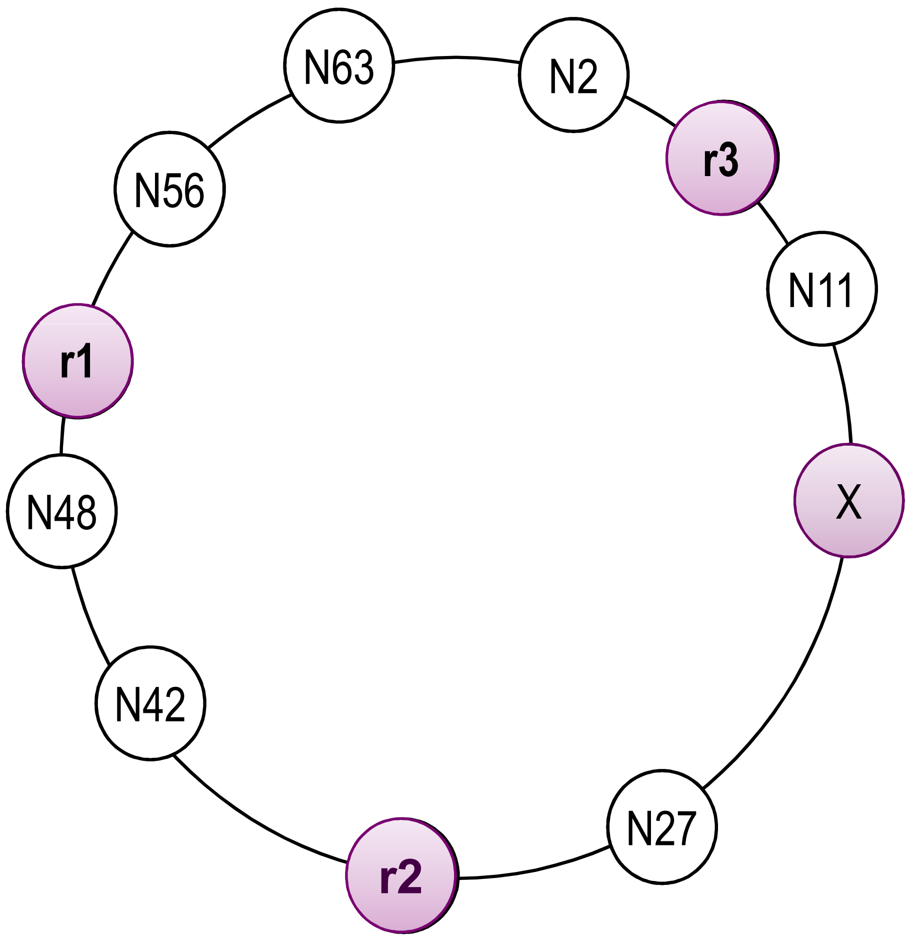
Query Chain Algorithm

- **Indexing a new triple**



Query Chain Algorithm

- **Indexing a new triple**



**Put (s1, P1, s2),
(s2,p2,s3),
(s3,p3,o3).**

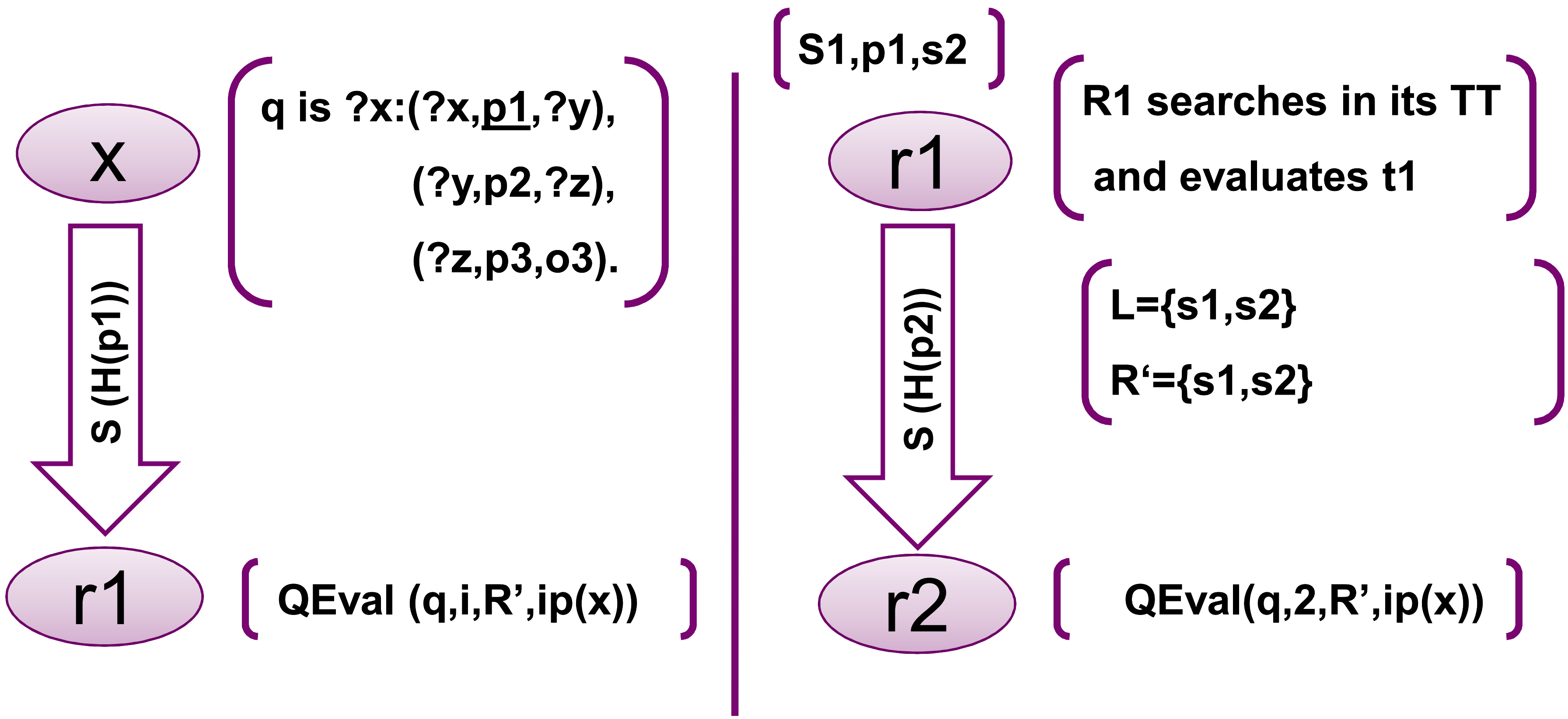
T1=Hash(p1)

T2=Hash(p2)

T3=Hash(o3)

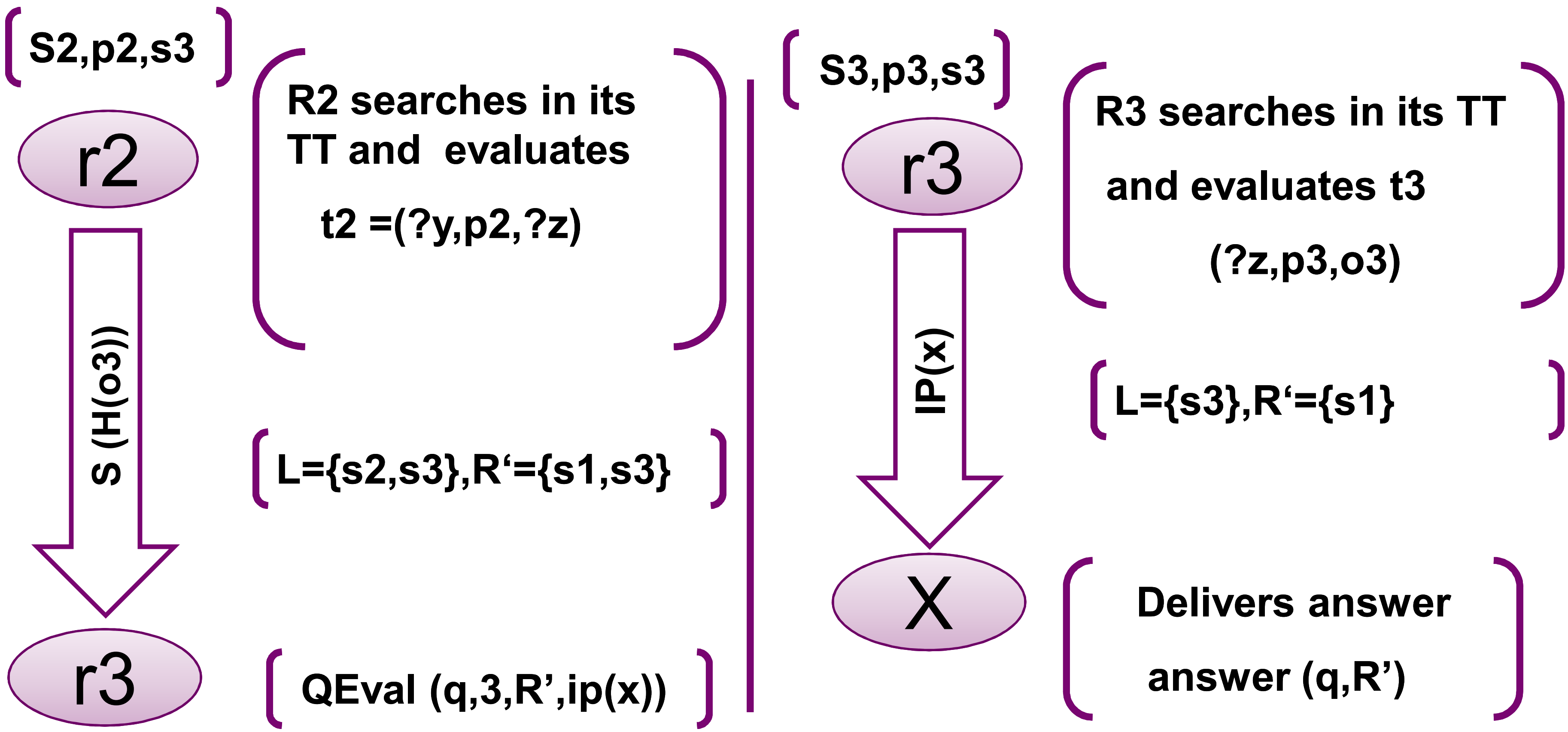
Query Chain Algorithm

- **Evaluating a query**



Query Chain Algorithm

- **Evaluating a query**



Query chain algorithms

- **Local processing at each chain node**

n

QEval (q,i,R,ip(x))

$L = \pi\ X\ (\ \sigma F\ (TT)\ )$

e.g : q1=(?s1,p1,o1)

$L=\pi\ 1(\sigma 2=p1 \land 3=o1(TT)\ )$

$R'=\pi\ Y(R \bowtie L\ )$

n'

QEval (q, i+1, R', ip(x))

n_k

QEval (q,k, R', ip(x))

$R'=\pi\ Y(R \bowtie \pi\ X\ (\ \sigma F\ (TT)\ )$

Answer (q,R')

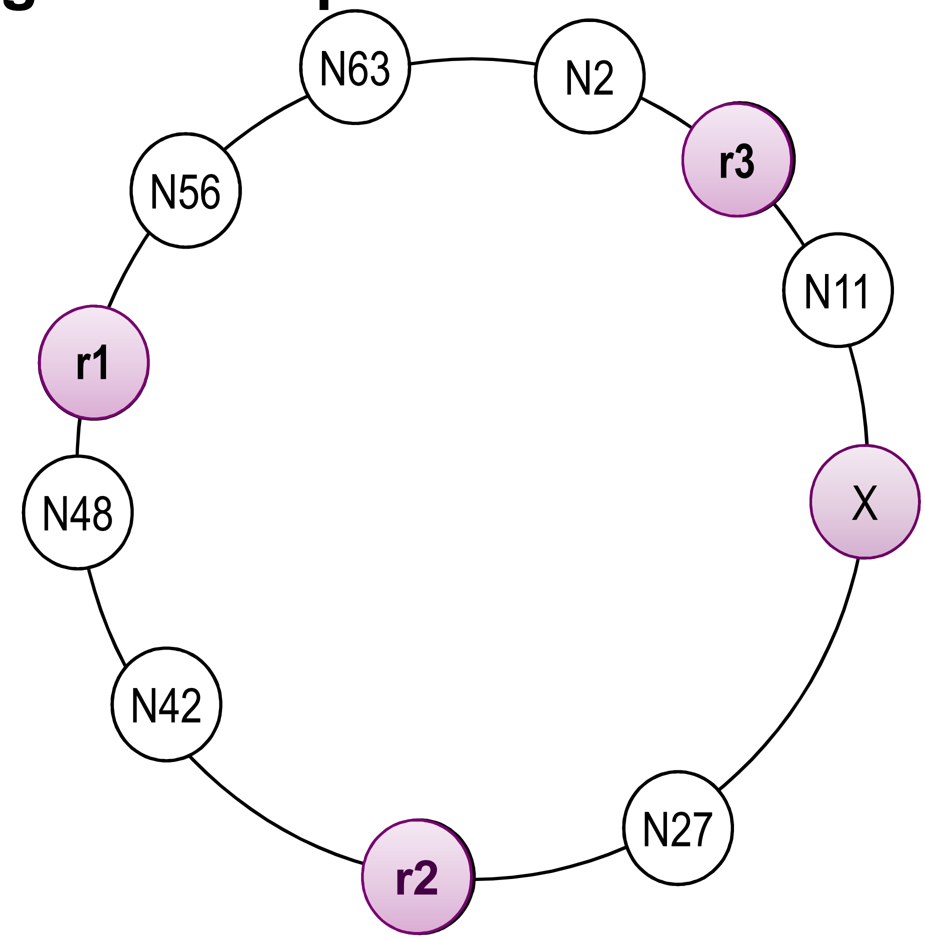
Spread By Value Algorithm

- In SBV the triple pattern t will be stored at the successor nodes of the identifiers Hash(s), Hash(p), Hash(o), Hash(s + p), Hash(s + o), Hash(p + o) and Hash(s + p + o).

- SBV exploits the values of matching triples found while processing the query incrementally.

SBV algorithm

- **Indexing a new triple**



Put (s1, p1, s2),
(s2,p2,s3),
(s3,p3,o3),
(s1,p1,o1) .

T1=Hash(p1)

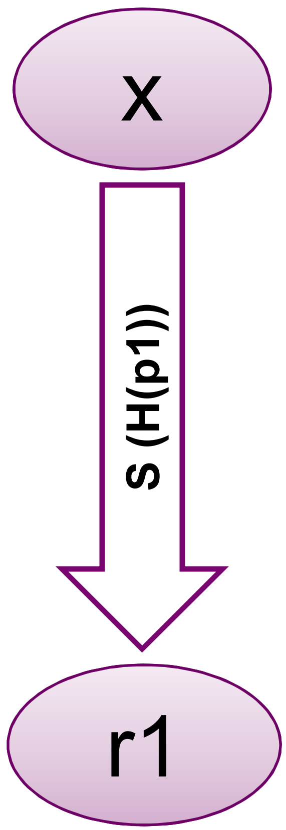
T2=Hash(s2+p2)

T3=Hash(s3+o3)

T4=Hash(p1)

SBV Algorithm

- **Evaluating a query**



q is ?x:(?x,p1,?y),
(?y,p2,?z),
(?z,p3,o3).

QEval(q, V, u, IP(x)),
r1 evaluates (?x,p1,?y)

SBV Algorithm

- **Evaluating a query**

(?x,p1,?y) ,(?y,p2,z3)

t1=(s1,p1,s2)

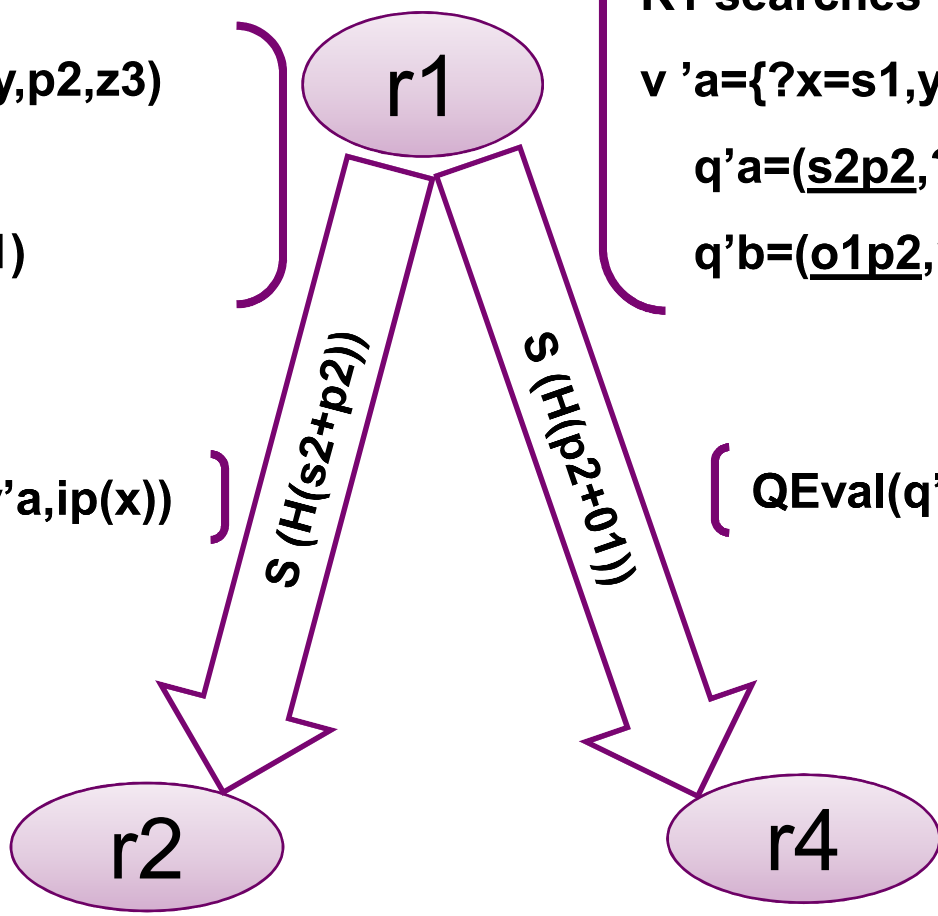
t4=(s1,p1,o1)

R1 searches in its TT and finds

v ’a={?x=s1,y=s2},v’b={?x=s1,y=o1}

q’a=(s2p2,?z)(?z,p3,o3)

q’b=(o1p2,?z)(?z,p3,o3)

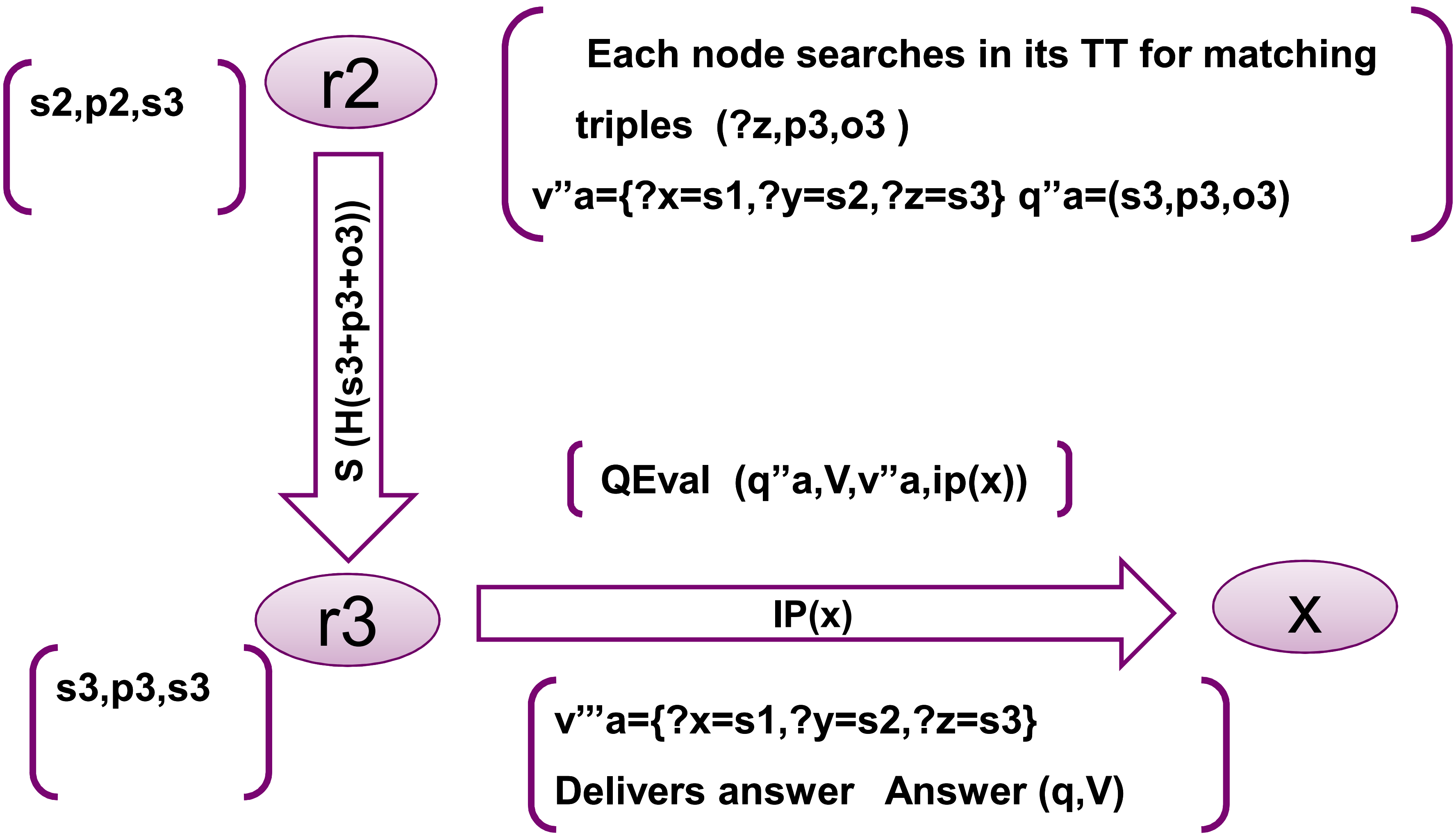


QEval(q’a,V,v’a,ip(x))

QEval(q’b,V,v’b,ip(x))

SBV Algorithm

- Evaluating a query

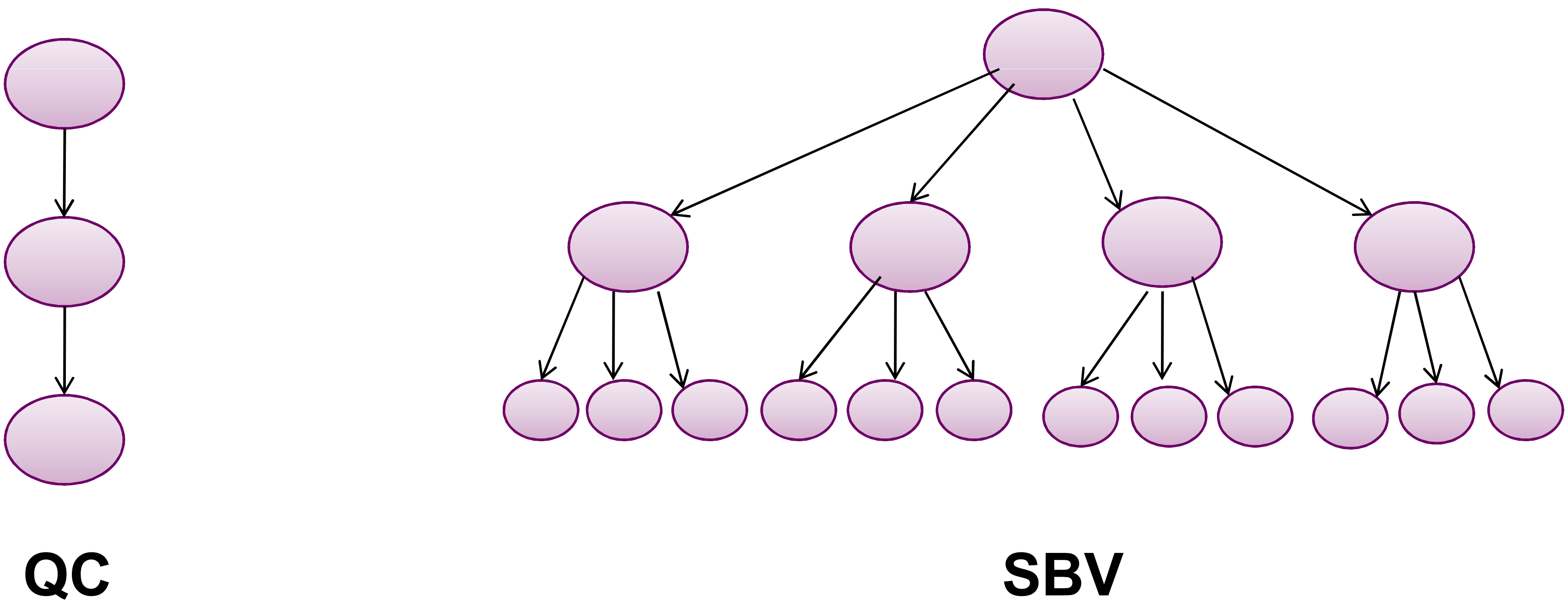


Comparing the query chains in QC and SBV

q is ?x : (s1,p1,?x),(?x,p2,?y),(?y,p3,o3).

?x has four matching values for (s1,p1,?x) .

?y has three matching values for each value of ?x .



Optimizing network traffic

- IP cache (IPC) used by our algorithms to significantly reduce **network traffic**.
- Another effect of IPC, is that we reduce the **routing load** incurred by nodes in the network.

Experiments

- A simulator for chord network with java code was done.

- Our metrics are:

  1) The amount of network traffic that is created.

  2) how well the query processing load and storage load are distribution among the network nodes.

Experiments

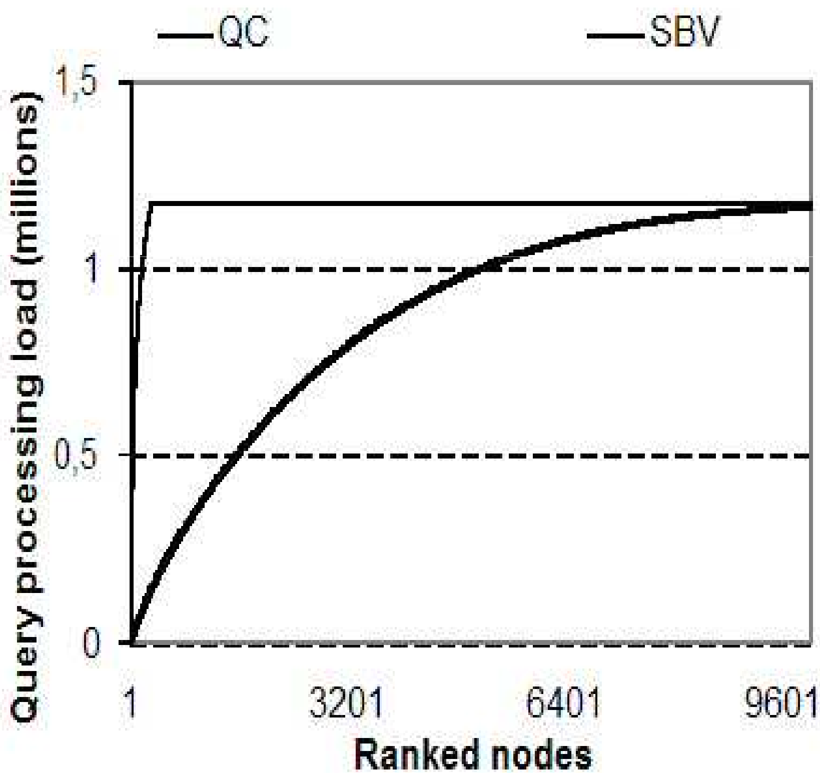
- Some Experimental evaluating of the two algorithms ,and the result are showed in the following figures

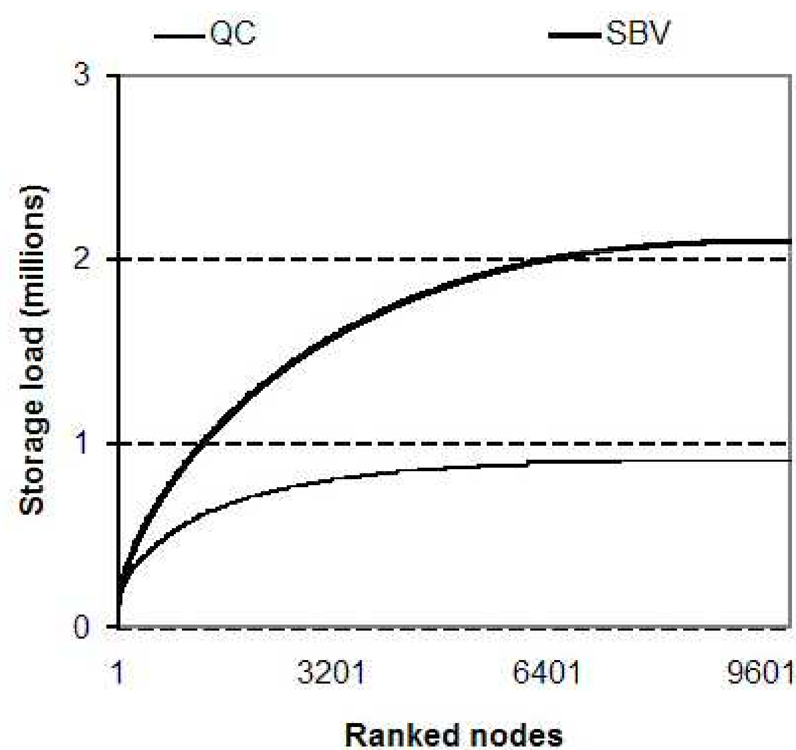
(a) Traffic cost

(b) IPC effect

Experiments

- Query processing and storage load distribution





Future work

- Order of triple choosed to evaluate the query.
- RDFpeers reasoning.

Reference

1) E. Liarou, S. Idreos, and M. Koubarakis. Evaluating Conjunctive Triple Pattern Queries over Large Structured Overlay Networks. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo.

2) E. Liarou, S. Idreos, and M. Koubarakis. Publish-Subscribe with RDF Data over Large Structured ,Overlay Networks. In DBISP2P '05.

3) Book : *Foundations of Databases*, Addison-Wesley, 1995. ISBN 0-201-53771-0.

4) I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable peer-to-peer lookup service for internet applications.

5) E. Prud'hommeaux and A. Seaborn. SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/, 2005.

THANK YOU FOR YOUR ATTENTION.