Albert-Ludwigs-University Freiburg
Department of Computer Science
Computer Networks and Telematics

SS 2009

Triple Distribution, Resoning and Load Balancing in DHT Based RDF Stores

Aldarwich Yaser

29. Juli 2009

Supervised by Prof. Christian Schindelhaue and Liaquat Ali

This paper addresses Resource Description Framework(RDF)[1]and the challenges encountered in structuring RDF stores based on distributed hash tables (DHTs), then it shows how the reasoning and integration can be applied using forward-chaining, also this paper describes the load-balancing issue and the efficient way to solve such a problem by using overlay tree.

1 Motivation

The growth of information in modern society made it more difficult to find relevant information that supports people at their tasks. Therefore Centralized databases will become incapable of handling large number of triples. P2P based distributed databases offer better scalability and integration of many different data resources.

2 Introducation

2.1 RDF

RDF stands for Resource Description Framework, everything in RDF is represented by triples of the form (S,P,O) used for presenting information on the web[1] like: Berlin is Capital of Germany ,
Berlin is the Subject, Capital is Predicate and Germany is Object.

2.2 DHT

Is a class of decentralized distributed system that provides a lookup service, it is used to Solve the item location problem in a distributed network of nodes [1][2]. Two main Operations in DHT are :
• Put(K, x): given key k (for a data item), map the key onto a node.
• Get(K): Return the data item with a given a key.
The major advantage of DHT based stores is to gather data from different resources into virtual global database.

3 Triple dissemination

This section describes the mechanism of storing and querying the data to/from DHT, the basic idea for storing RDF in DHT is as follows: each triple should be stored at three locations on the DHT ring; these locations (nodes) are determined by calculating the hash value of the Subject, P and O. Insert and look up operations can be done in O(log N) step, where N is the number of nodes in network. The pastryprotocol is used rout the message to responsible node using prefix matching mechanism[5][4].

4 Node Architecture

Each node hosts multiple RDf databases, That serve different purposes described as follows: local triples database, Received triples database, Replica database and Generated triples[9].

4.1 Local triples

Stores the RDF triples that originate from particular node, all local triples are systematically distributed to the nodes in the network by calculating the hash value of the subjects, predicates and objects then sending the triples to the nodes responsible for the corresponding parts of the DHT space[9].

4.2 Received Triples

receive and store the triples from the other nodes.

4.3 Replica Database

It is stored in node whose id is the nearest to target hash value determined by hash function. The purpose of this replica is to support the network when nodes depart or crash[9].

4.4 Generated Database

Finally, each node hosts a database for generated triples that originate from applying forward chaining on received triples[9].

5 Reasoning

The inferencing is based on forward chaining, that is one of the two main methods for reasoning, it helps to infer new knowledge from existing or available information, it allows us also to extract information in case the description does not exactly match a query.
Example:
Alex fatherof Christian
father subpropertyof relatives
⇒ Alex relatives of Christian

6 Triple dissemination in DHT

The main part of this paper is described in details in two sections. First, Triple dissemination in DHT, Each node contributes its local triples to the network. Generated triples are created by applying forward chaining on received triples databases then all those generated triples are stored into the generated triples database after that, they are disseminated like local triples to the respective nodes and replicated from there on[8].

7 Load balancing

The second main section is Load balancing, the Major criticism against DHT based RDF stores is Load balancing because many collisions are unavoidable. As DHT stores many triples with predicate rdf:type and each one has at least one type and triples hashed by (S,P,O), then the node responsible for the hash value of rdf:type is subjected to high storage load because of transitive relation like rdfs:subClassOf that creates many triples with Predicate rdf:type . The problem of overloading node can be solved by constructing overlay tree for discrete positions on DHT and introducing a remote triples database for storing triples in overlay tree structure, each node can have several remote triple databases and can offer capacity to several overloaded nodes[8]. Example: we have four nodes numbered from 1 to 4, if first node gets overloading, it shift half of frequent collision triples from its Local database to remote triples database, half of triples stay in local part and second half move to node 2, external part of node 1 is linked to new remote triples database of node 2 through a reference which can be an IP address of destination node. Node 2 had no external part of remote triples, but only local part. Node 2 became overloaded, it recursively splits its data and shifts half of them to node

3. Now, node 1 recognizes that it is still overloaded. So, it splits the local part of received triples by creating new remote triples database for storing half of the triples and second half is linked and stored to another remote triples database on another node. By this way we can reduce the load of nodes. If node 1 gets a query for an ID storied in the remote triple database, the query is routed directly into both branches or to all branches.Query reaches its target in O(log N+d) steps ,where N states for number of nodes in the DHT network and d is the depth of the overlay tree[9].

8 Reparation of node failure

The triples are affected by many various events during their life cycle like departing or joining, these events may leads in some cases into different problems like Node crashing, that causes the expiry of triples and vanishing the branches of network. To avoid such a problem and ensure the stability of connection of per to per network, the DHT layer has to repair the routing tables also the modification of replication strategy is required. Each node has two replicas, when node crashes a replica has to cover the area of vanished node as follows: when a vanished node receive query, the first replica receives the query without processing it. Rather, the query is routed to the root node, which performs then the load balancing among itself and replica[9].

9 Handling RDFs rules in load balancing

9.1 Problem of RDF rules

As node is overloaded, the triples are splited into other nodes **Example:**
a, rdfs:domain, x
u, a, v

9.2 Solution

Make copy of most common rdfs schema into each replica

10 Conclusion

Advantage of storing RDF in DHT is, that all triples with common S,P or O can be looked up at one node, no need to collect them from all data sources disseminated over the network. P2p based distributed database offer beter scalability and source integration. RDFs provide real power by inferring new information from explicit information. Overlay tree is the solution for overloading problem.

11 References

1. http://www.w3.org/TR/rdf-schema
2. http://www.w3schools.com
3. http://www.videolectures.net
4. http://cone.informatik.uni-freiburg.de
5. http://peersim.sourceforge.net
6. http://infolab.stanford.edu
7. http://www.edutella.org/edutella.shtml
8. Battre,heine,Kao:Top k RDF query evaluation in p2p network
9. Dominic Battre, Felix Heine, Andre Hnig, Odej Kao : On Triple Dissemination, Forward-chaining, and Load Balancing in DHT Based RDF Stores