An Approach of Content Management based on Structured Peer-to-Peer Networks

Luis Enrique Colmenares Guillén and Angel Omar Mendoza Rojas

Facultad de Ciencias de la Computación FCC de la
Benemérita Universidad Autónoma de Puebla BUAP
Apdo. Postal J-32, Ciudad Universitaria, Puebla, México.
lecolme@cs.buap.mx, aomendoza87@gmail.com

Abstract. In this work, first evaluates routing Dht and the integration with metadata in query; in the future during the second phase would evaluate the proposal in dynamic scenes considering churn-rate and during the third phase would implement the application in platforms of development for P2P networks, that offer some applications in management of contents such as: collaborative, distributed and scalable. The main contribution of this first stage is the improvement obtained by the Distributed Hash Table (DHT) when adding metadata in the query. Semantic Routing reduces the number of hops averages to locate an object. This routing aid Dht-bamboo is used to improve the search of documents, because it obtains the interest of participants that could have on the documents. The metadata give us expressiveness Dht. In Dht-bamboo, the metadata consumes more bandwidth in the communication. The number of metadata within query increases the cost of communication between all the participants of the collaborative application.

1 Introduction

The three architectures more used of content delivery systems are: Wide World Web (WWW), Content Delivery Network (CDN) and Peer to Peer (P2P). The WWW and CDN are client-server architectures. In the WWW, the objects are static and small and the requests are usually frequent. The excess of requests can create unavailability of servers. A solution to this problem would be to have a great number of servers and high resources of storage, like for example, the CDN [1] that contains many objects of audio and video. One disadvantage of CDN is that represent greater costs in the storage and communication. Another solution is using the P2P networks because they are decentralized, distributed, and autonomous.

There are three considerations that are important for the management content:

1. The growth of non structured data. The structured data is kept in a data warehouse; data-marts and/or applications from databases. The non structured data are: Audio, video, text without structure, such as the body of an email or a document made by a text processor, spreadsheets or presentations.
2. The management of content in the network edge. The document management and the content management use metadata to classify and to search

© G. Sidorov, M. Aldape, M. Martinez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 42, 2009, pp. 251-262

Received 18/03/09
Accepted 27/04/09
Final version 06/05/09

information. The document management includes the information lifecycle, from its creation to the visualization by the final users. The management of contents is oriented to the reusability of the content.

3. The collaboration of Internet inside and between users from universities is frequent and useful in nowadays. The collaborative applications such as BSCW [3] and MOODLE [4] are client-server architectures in the WWW, and all the contents are in the server; the participants are dependent of the robustness of the server in order to have the active contents. Nevertheless, the disadvantages of these centralized applications are that they represent a single point of fault. Groove [5] is a system P2P that provides collaborative tools like chat, forums of discussion, interchange of files and calendars.

Now, the prototype makes the basic functions of a system of content management and the creation and search of the content with mechanisms of caching and routing. This work is organized as follows: in section 2, the description of the requirements of the system, in section 3, the architecture of the management of content for a P2P network based on collaboration. In section 4: a case of study, the used platform and the evaluation, and in the section 5, gives some conclusions of the contributions.

2 Architecture Requirements

In this section outlined the requirements that differentiate our system from the existing content management systems. The requirements in order to improve the delivery content are the following: The data types to share, the use of open standards and the management of content. Insert new: control of the content, indexes and searches P2P, content-based on interest and management of content off-line.

2.1 Non Structured Data

The characteristics of the data or contents are: they have small size, they are not structured and they change frequently, although, the contents might have greater size, and greater use of the bandwidth and storage. A discussion about data not structured is in [2].

2.2 Open Standards and Communication

The use of structured P2P networks based on Distributed Hash Table (Dht) as OpenDht [8] and Bamboo [9] is event-driven and has a single-threaded programming style [9]. Many programmers can be familiarized with this style of programming when using Graphics User Interface (GUI) libraries such as Java Swing or the Gimp toolkit (GTK++). When code is event-driven written it is easy to understand and it can generate high concurrence. The configuration file of bamboo is similar to XML files. Each node of the network Bamboo has a file with these characteristics.

2.3 Overlay peer-to-peer Networks

The overlay networks are divided into three different topologies of peer-to-peer: unstructured, structured and hierarchical. In the peer-to-peer unstructured they have a centralized manager index; the disadvantage is that they represent a single point of failure. All peers are equal and peers do not have restrictions imposed by the topology. In the unstructured P2P networks [7], the distributed searches are flooding, and have the disadvantage of saturating the network of messages, increasing some times, their delivery time. In the cases, where the documents are rare to locate, the search time could end before finding the document. A characteristic that contributes to the success of the unstructured P2P networks, is that they support versatility in query or partial-match [10], that is to say, within query the objects are described by their types and properties (for example: Title, composer, interpreter).

In hierarchical networks, the peers are divided into two groups: The Hubs (called SuperPeers or SuperNodes), which are dedicated servers with large memory and computational power and the leaf-peers responsible to store the contents (because the central server does not store contents). The Hub server recognizes the resources that are for sharing and are available.

Finally, the structured networks have a regular topology, circles, torus or cubes. In addition they use distributed hash tables (Dht). Within Dht, the mechanism search (lookup) provides a model with deterministic searches that hide to the user: request of routing, costs of churn-rate, load balance and availability.

Some works, that have used efficiently Dht in the networks overlay, are [11], [12], [13], [9], and [14].

Two main disadvantages of the P2P structured systems based on Dht are identified:

1. The requests are randomly sent towards peers without associating the content, that is to say, the use of keywords lacks of flexibility in the mechanism of lookup and routing.
2. The indices in the hash tables present transitivity, because continuously peers are connected and disconnected of the network, affecting the robustness of the system.

2.4 Content Classification and Control

A Content Management System (CMS) should be able to manage content coming from two main sources: participants and repositories. Moreover, the system should access a database, remote web server, or XML files.

Metadata offer a good way to classify any piece of information and this makes it possible to search for content in a very simple and efficient way.

2.5 Indices and Searches P2P

In the search methods, it is fundamental to have an appropriate mechanism of indices and an optimization of queries that are propagated by the indices. The engineering of the indices is the heart of the search methods of P2P. Now, the applications use indices distributed in the P2P network. The known example is Gnutella [7] that has semantic indices. In many applications, the keywords are associated with a document, an administrative domain or a key in a database.

The indices in Dht are semantic-free with data-centric references [15]. These references are based on the name of the data, without interest in the location. The semantic indices capture the relation of the objects. While, that Dht semantic-free guarantees that they can find a key, even if it there is no relation with the content.

The query is a keyword that represents content or a document. The keywords or metadata have been studied in size and number. For example, the queries that contain, keywords in the Web, demonstrated in an analysis that the communication costs are increased as the keywords grow, indicating that they do not scale well [16]. Nevertheless, the indexed in keywords is possible in structured P2P networks [17], because the size of the metadata is usually single-key. Some Works [18] and [19], include metadata in the request of routing, improving the precision of the search.

2.6 Content-based Interest

The content-based interest is a characteristic has been linked by affinities of social networks that currently remain a challenge to solve [26].

2.7 Management of Content Off-line

Many implementations like Bamboo, Kelips [20], or [21] use an algorithm broadcast as they can be epidemic, rumor, among others. For example: the nodes that are affected in their tables of Dht when a node is connected or disconnected, sends a message (a rumor), to their near neighbors like their distant neighbors, propagating in the entire network the rumor and updating the corresponding indices within Dht.

At this point, should remark that the Architecture is still in a prototyping phase, and only some of the requirements above have been addressed:

1) The non structured data is displayed as indices; these indices associate a document creating a document list.
2) Routing was added so that it determines the preferences of the objects and allows that the requests are based on interests. A caching mechanism with a LRU policy was designed and the size of caching is static.
3) Open standards for communication between nodes are used. A distribution of Zipf for requests of documents is used. One Dht of a structured P2P network is installed in a computers cluster.
4) Requirements 2.4 and 2.7 have not been implemented yet.

3 An Approach of Management of Content for a Network P2P based on Collaboration

Dht-Bambo allows the communication between the nodes, the high level architecture of open-Dht is used, Fig. 1. The nodes OpenDht act like gateway, each node executes the code of OpenDht allowing the communication via RPC to any user who belongs to the network. The users (student or teacher) are nodes that can be situated outside of the infrastructure of Open-Dht. The nodes Open-Dht participate in routing and storage of Dht.

The three phases of our architecture that provide management of content in the context of structured P2P based in Dht are: 1. Communication begins with caching indexed 2. Searches with keyword in routing, make simultaneous requests for the location of content. 3. Replication with a broadcast algorithm to update content.

3.1 Communication

In this phase, the system initializes the collaborative architecture. We identified two roles that can be defined in the same PC. The role of participant is in the physical network and the role of participant-Dht, which belongs to the Open-Dht. Two zones are created when you enter the collaborative architecture: zone of replication of the participants and caching indexed of the participants-Dht. In Fig. 2, the indexed of a participant-Dht is the zone of caching; this indexing is a local index where the documents ordered by popularity are located. For example, a metadata "a" is associated with the node in the form of IP 192.168.0.1 within query, the request that makes the node open-Dht, this is composed of metadata and IP address and is called metadata-node. This metadata-node is in the participants who previously are asking for theme "a". The access to the network of contents is simple, by means of script in Perl all the participants-Dht are created. The participants have the infrastructure to communicate with other participants via RPC in Internet. The phase of communication allows during the disconnection the information of the active session to be stored. When the participant connects again, automatically the metadata-nodes update all the participants who belong to their shared areas.

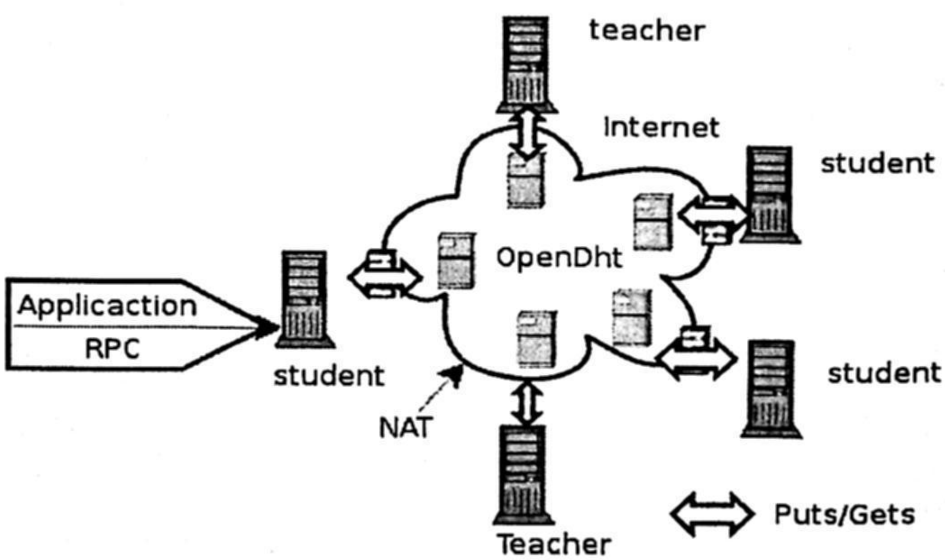

Fig. 1. Collaborative Architecture with OpenDht.

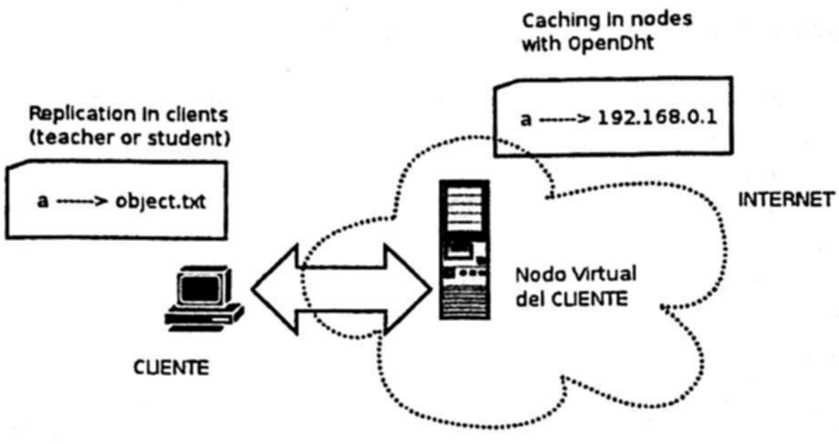

Fig. 2. Communication of clients and OpenDht nodes

3.2 Searches

In this phase, there are two mechanisms to make precise searches, first provides additional routing and second within query, there are metadata that represent the keyword for a location based on the content of the Theme. The first mechanism is called search of the participant-Dht, the participant uses different routings, and in the second mechanism the participants locate contents in their shared areas.

3.2.1 Search of the Participant-Dht

There are two routings in our architecture: routing of Dht-bamboo and routing semantic based on caching, see Fig. 3. For Dht Routing (DR), use routing of bamboo, is one Dht based on Pastry [11]. Dht-bamboo [9] assigns to each node, a key. The main mechanism is lookup; this has a greedy protocol that comes progressively near to the key in each hop. Each node maintains a set of neighbors who are used to send contents. These neighbors are: a) Near neighbors, which are numerically near and b) distant neighbors, that guarantees to choose some of them, for a network of *n* nodes, has a maximum of $O(\log n)$ hops and the number of neighbors does not exceed $O(\log n)$. In bamboo, the near neighbors are called leaf set and the distant neighbors routing table.

A new class of bamboo is created, Semantic Routing (SR) works with the Dht. The mechanism of lookup, now, has three ways: leaf-set, routing table, and SR.

Info-semantic is defined, as a pair of data (metadata, node). These data, represent the metadata = key and value = IP node. This space of memory that uses info-semantic is caching. The key is the theme and the value is the node that has the theme.

At each request, the node, keeps in its zone from caching the metadata-node. The participants, who have interest in the document, keep the connection metadata-node. And in next requests, the content is given in a smaller number of hops. The participant, who makes for the first time a request of an object, uses the DR. Info-semantic is limited in each node, that is, the node can be full and use the DR. If a node makes a request and the object that is requested is not found in caching, also the DR is used. The substitution policy is made in order to eliminate at least the most recently used (LRU), and updating caching in each participant node.

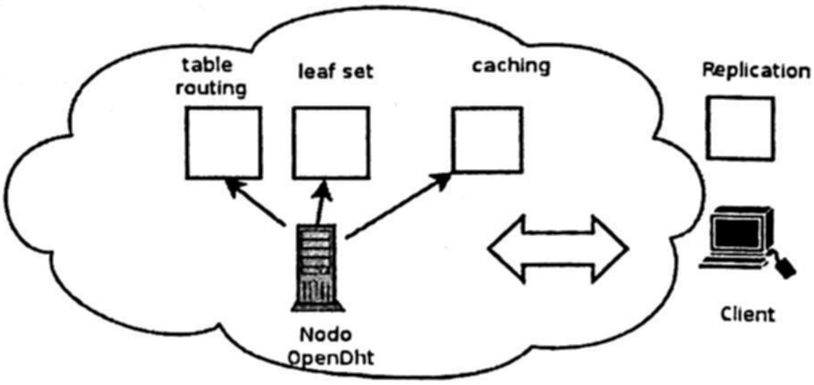

Fig. 3. Routings

3.2.2 Search in the Participant

The mechanism search in the participant is similar to the search in a system of files or a page Web. It is a continuation of the search in the participant-Dht. That is to say, when the way in routing was chosen, it traverses through different participants-Dht. The participant-Dht who has the content is chosen, so the content in its shared areas is located and success in the search is obtained. The Dublin Core schema [25] provides a consolidated and nearly standardized way for documents classification.

3.3 Replication

The works of replication like Beehive [22] or Symmetrical Replication [23] that need O(1) messages when the nodes enter/leave the network, do not relate the content from node, that is to say, they relate the content to numerical approaches like Dht. Bamboo supports an epidemic algorithm during the update of indices in the tables of the nodes when churn-rate exists. When extending this algorithm, and updating the new zone of caching, the new information is available in the nodes.

4 Case Study: The Magazine Collaborative

The motivation of this work is to have a collaborative magazine in the network overlay structured P2P, where the participants are students or professors. These participants are from different universities from all the country that can be located in physical or administrative limits, with security mechanisms: firewall, router, NAT, among others.

The work of professors and students is the creation of content where they can collaborate and share contents. This magazine is scalable, distributed and decentralized. The storage of the contents is directly proportional to the use of the themes and the degree of collaboration. The collaborative magazine cannot be accessed by certain documents because the participants do not have sufficient bandwidth for the communication of messages or when a document is highly acceded.

When a participant has decided to provide the content to another participant, these users have a relation. Regularly, those relations are created in run time and can change on the time. Each participant has a shared zone, where the files are associated by interest.

The users have interest in the contents. This allows creating content based on interests in a zone of caching. The zone of caching has a fixed size. When added caching to the users, improved response time, although to have a new difficulty, the update of caching that is solved including a policy of Least Recently Used (LRU) caching.

The users have permissions to make changes to contents. In addition, the system maintains coherence and consistency of the contents, even though the user is off-line. In the storage, Dht supports the primitive, *put* (key, value) and *get* (key) within routing. The idea is extended this algorithm and measure the impact that it generates to have more content in the node, in static and dynamic scenes.

4.1 Platform

The platform to measure our architecture this made in Bamboo-DHT [9] and the language scripting is Perl; the code is within a cluster of 16 nodes. The nodes of cluster have dual microprocessors AMD Opteron 64, 1.4 GHz, connected by Ethernet Gigabit, RAM memory of 2 GB a hard disk SCSI of 36 GB and OS fedora Linux Core 6x86-64. The simulations are execution-driven. All the file logs are stored in each participant of the network overlay P2P in run time.

4.2 Evaluation

For the evaluation three modules or classes in Bamboo were added. The first module, is caching, that assigns cache in each participant-Dht and allows the node to choose between two routings. The second module makes simultaneous requests in parallel. This allows making requests of queries from different participant nodes at the same time. The third module is the replication of content, and this in phase of elaboration. This will consist basically of the extension of the epidemic algorithm of Dht-bamboo to update the contents in caching of each node.

In the evaluation the following parameters are considered in the tables:

1.Requests based on the law of Zipf (simultaneously from different participant nodes).

2. Numbers of nodes with info-semantic: Use a local index and create this zone. Info-semantic is the zone of caching.

3. Limit of the Caching in the node: The zone of caching is limited; it uses a policy LRU and the objects from most popular to least popular.

4. Structured Metadata: The filtration of the searches of contents is made with metadata that represent a theme. The queries allow a unique key within routing of Dht. This association key-metadata within routing allows precise searches.

4.2.1 Static Scenes

The evaluations are made both routing SR and DR. In Table 1, shows the parameter that was used for static scenes with routing Dht-bamboo. The number of tests that were made was approximately 100 by each scene, although from test 45 the graphical one shows a uniform behavior with respect to the growth of the average of the number of hops.

The Table 2 shows the parameter that was used for static scenes with SR. A metadata in query of Dht is used; in addition caching that was made vary in the size of each node or participant. The number of objects is important, if are a small number of requests, 5, 10 or 20 requests. The objects will vary of 100, 250, 500, 750, and 1000 with duration of the object of one week (7*24*3600) or a day (24*3600) guaranteeing that the objects always are in the network bamboo. The communications between nodes of the cluster are stable in approximately 85% of the nodes. This guarantees 85 % of objects. In the number of requests: 10000, 12500, 15000, 20000, observed that the limits of the average are reduced. The size of caching used are: 5, 10, 15, 20. The use of caching represents use more the Dht in SR. In scenes 1, 2, 3 of table 2, using small caching, would more use of Dht-bamboo. In experiments 4, 5, 6 used greater caching, would use less the Dht-bamboo that is reflected in a smaller number of hops, as is in Fig. 5.

Table 1. Routing Dht-bamboo parameters.

Scenes: Dht-bamboo	No.Nodes (parameter 2)	No. Objects	No.Requests (parameter 1)	Without *caching*, without metadata
1	500	100	10000	-------------
2	500	250	12500	-------------
3	500	500	15000	-------------
4	500	750	15000	-------------
5	500	1000	20000	-------------

Table 2. Routing Semantic parameters.

Scenes: info-semantic (parameter 4)	Nodes (parameter 2)	Objects	Requests (parameter 1)	Size of caching (parameter 3)
1	500	100	10000	5
2	500	250	12500	5
3	500	500	15000	5
4	500	750	15000	10
5	500	1000	20000	15
6	500	1000	20000	20

In Fig. 6, so as in Fig. 4 and Fig. 5, the minimum, 3er quartile and the average of the number of hops by searching each object were found. The graph of Fig. 6 shows the following: The five more popular objects of the average, the total of objects represent 95% of the total requests. The requests are sent from 500 nodes of a total of 500 available nodes. Each node sends n requests of m available object. Each experiment of 500 nodes uses 500*n requests. In x-axis, the objects are ordered from the most popular to the least popular, indicated by the named percentage and with their labels from left to right. In the y-axis, the number of total hops is shown.

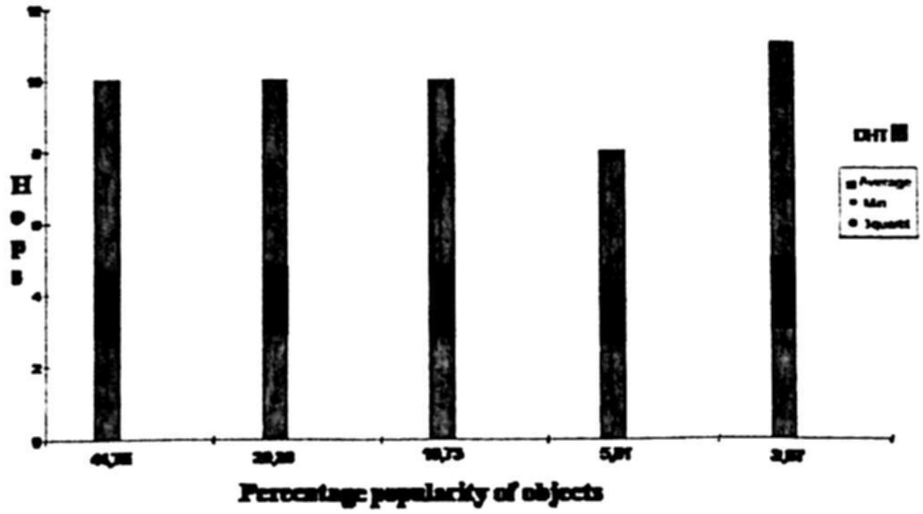

Fig. 4. Dht-bamboo Routing

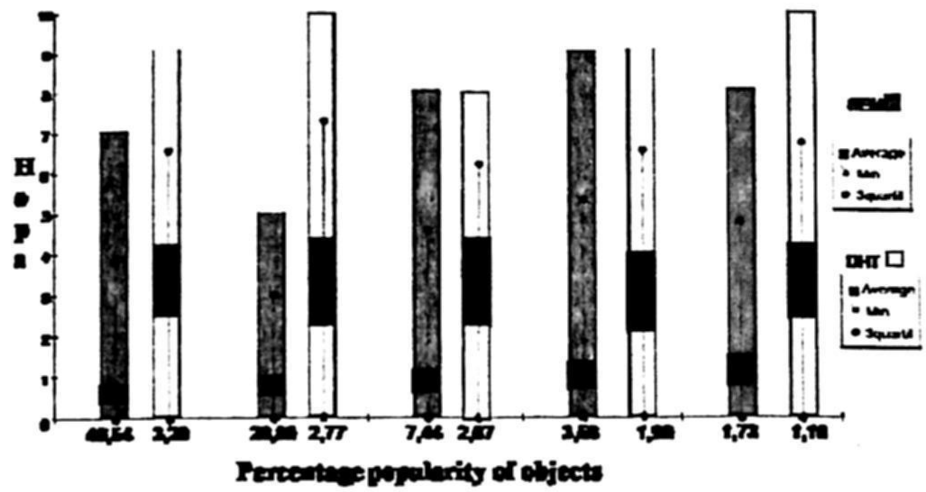

Fig. 5. Semantic Routing

In Fig. 6, there are five most popular objects with three bars, respectively. The two first bars of each object represent semantic routing proposed and it is a combination of both routings. The last bar of each object represents routing of Dht-Bamboo without alterations. The bars are divided in quartiles; the maximum number of hops for a same object is the high part of the bar. The third quartile represents the superior part of the line that is within the bar, the average or second quartile represents the high part of the black picture and the first quartile is the minimum hop that was obtained by requests of a same object that represents the final part of the line. The average of hops (superior limit of the black box) diminishes when the object is more popular. We observed in the graph that improvement in the number of hops by routing formed by DR and SR are the two initial bars. Routing of DR represents the third bar of each popular object; the average is over the two bars.

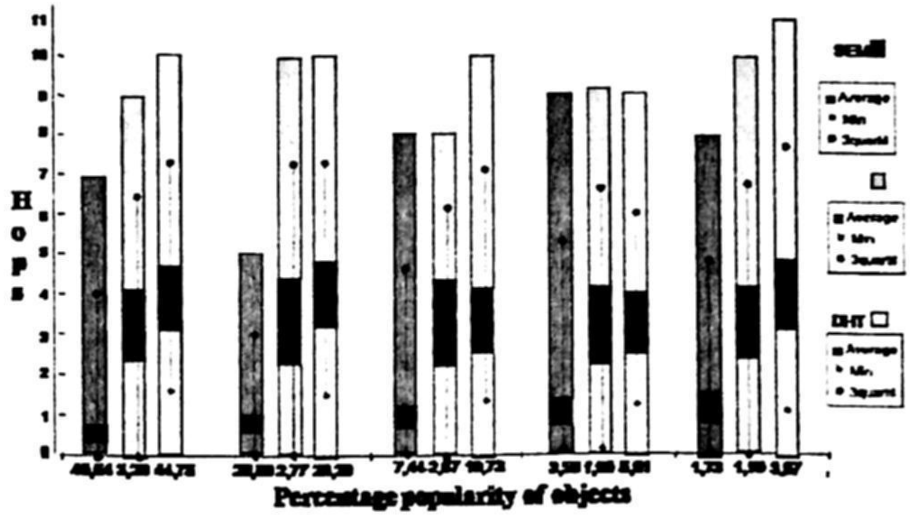

Fig. 6. Comparison of Routings

5 Conclusions

The main contribution of this work is to improve the routing of the Dht-Bamboo, with semantic routing that reduces the number of hops average. The culmination of this work will have three contributions, now had fulfilled the two first: 1. a mechanism of content delivery that allows possible routings to reach the content, when adding semantic routing in Dht-bamboo, obtained a reduction in the number of hops averages to find an object and 2. A search mechanism using semantic information, giving expressivity to Dht, single-key was used and it does not affect the costs of the communication between nodes of Dht, and 3. The last contribution will end with the class of bamboo that extends the broadcast algorithm, in addition, the evaluation in dynamic scenes, with parameters likes churn-rate and limitation of resources.

References

1. Akamai Technologies, Inc. [Online] http://www.akamai.com/. (2009).
2. Robert Blumberg, Shaku Atre. The Problem with Unstructured Data. DM Review Magazine. (2003).
3. BSCW (Basic Support for Cooperative Work) http://bscw.fit.fraunhofer.de/ (2009).
4. Modular Object-Oriented Dynamic Learning Environment (Moodle) http://moodle.org (2009).
5. The Microsoft Office Groove. http://office.microsoft.com/es-es/groove/default.aspx/ (2009).
6. "The emergence of Distributed Content Management and Peer-to- Peer Content Networks", by Gardner Consulting, (2001).
7. T. Kleinberg and R. Manfredi, http://www.gnutella.com, (2009).

8. S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu. OpenDHT: a Public DHT Service and its Uses. SIGCOMM' 05, Philadelphia, Pennsylvania, USA, august 21-26, (2005).
9. The Bamboo Distributed Hash Table. A Robust, Open-Source DHT. http://www.bamboo-dht.org y http://opendht.org (2009).
10. E. Cohen, A. Fiat, and H. Kaplan. A Case for Associative Peer to Peer Overlays. ACM SIGCOMM Computer communication review, vol. 33, issue 1 pp 95-100. ISSN: 0146-4833, (2003).
11. Rowstron, A., Druschel, P.: Pastry: Scalable, Distributed Object Location and Routing for Large-scale peer-to-peer Systems. In IFIP/ACM Int. Conf. On Distr. Systems Platforms (middleware) (2001) 329-350.
12. Stoica, I., Morris, R. Karger, D., Kaashoek, M. F., Balakrishnan, H.: Chord: a Scalable Peer-to-Peer Lookup Service for Internet Applications, In Conf. On applications, Technologies, Architectures, and protocols for comp. Communications (2001) 149-160.
13. Zhao, B. Y., Kubiatowicz, J. D., Joseph, A. D.: Tapestry: An Infrastructure for Fault-Tolerant Wide Area Location and Routing. TR UCB/CSD-01-1141, UC Berkeley (2001).
14. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, and S.: A Scalable Content Addressable Network. In: ACM SIGCOMM 2001. (2001) 161-172.
15. S. Shenker. The Data-Centric Revolution in Networking. Proceedings of the 29th VLDB Conference, Berlin, Germany, (2003).
16. J. Li, B. T. Loo, J. M. Hellerstein, M. F. Kaashoek, D. Karger, and R. Morris. On the Feasibility of Peer-to-Peer Web Indexing and Search. In Proc. IPTPS'03, (2003).
17. A. T. Clements, D. R. K. Ports, and D. R. Karger. Arpeggio: Metadata Searching and Content Sharing with Chord. In Proc. IPTPS' 05, (2005).
18. P. Triantafillou and I. Aekaterinidis. Content-based Publish-Subscribe over Structured P2P Networks. DEBS' 04, (2004).
19. Y. Choi and D. Park. Mirinae: A peer-to-peer Overlay Network for Large-scale Content-Based Publish-Subscribe Systems. In proc. of ACM conference NOSSDAV' 05, Washington, USA, (2005).
20. I. Gupta, K. Birman, P. Linga, A. Demers, R. van Renesse. Kelips: Building an Efficient and Stable P2P DHT through Increased Memory and Background Overhead. Proceedings of the 2nd International Workshop on Peer-to-Peer Systems, IPTPS' 03,(2003).
21. S. El-Ansary, L. Onama, P. Brand, and S. Haridi. Efficient Broadcast in Structured P2P Networks. In Proc. of the 2nd. Int. workshop on Peer-to-Peer Systems, IPTPS' 03, (2003).
22. V. Ramasubramanian, E. Gün, Beehive: O (1) Lookup Performance for Power-law Query . Distributions in Peer-to-Peer Overlays. Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation, (2004).
23. Ghodsi A., Alima L., and Haridi S. Symmetric Replication for Structured Peer-to-Peer Systems. In the 3rd Int. Workshop on Databases, Information Systems and Peer-to-peer computing, Trondheim, Norway (2005).
24. Adamic, L.A. "Zipf, Power-laws and Pareto– a ranking tutorial", http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html (2009).
25. http://dublincore.org/, (2009).
26. H. Garcia-Molina, "Web Information Management Past, Present and Future". In the Proceedings of the International Conference on Web Search and Web Data Mining, Palo Alto, California, USA, 2008, pp. 1-10.