An Evaluation of a Cluster-based Testbed for Peer-to-Peer Information Retrieval

Saloua Zammali
Dept. of Computer Science and Mathematics
Faculty of Sciences of Tunis
Tunis, Tunisia
Email: zammalisalwa@gmail.com

Khedija Arour
Dept. of Computer Science
National Institute of Applied Sciences and Technology of Tunisia
Tunis, Tunisia
Email: Khedija.arour@issatm.rnu.tn

***Abstract*—Peer-to-Peer (P2P) systems present an advantageous way to provide and share services [1]. Hence, P2P are the major technology of access upon various resources on Internet. Hence, P2P are the major technology of access upon various resources on Internet. A particularly intriguing class of distributed applications consists in Information Retrieval (IR) systems. The issue of Peer-to-Peer Information Retrieval (P2PIR) is being tackled by researchers attempting to provide valuable insights and to propose solutions to use it successfully. Nearly, all published studies have been evaluated by simulation means, using well known document collections (usually acquired from TREC). This practice leads to two problems: First, there is little justification in favor of the document distributions used by relevant studies and second, since different studies use different experimental testbeds, there is no common ground for comparing the solutions proposed. In this paper, we propose, *CB* a testbed for P2PIR based on P-Kmeans. CB, a cluster-based testbed, allows to distribute documents. This work marks the start of an effort to provide more realistic evaluation environments for P2PIR systems as well as to create a common ground to compare the current and future architectures.**

***Keywords-Testbed*; *P2P systems*; *Information retrieval*.**

I. INTRODUCTION

Peer-to-Peer (P2P) systems present an advantageous way to provide and share services [1]. Hence, P2P are the major technology of access upon various resources on Internet. Hence, P2P are the major technology of access upon various resources on Internet. A particularly intriguing class of distributed applications consists in Information Retrieval (IR) systems. The issue of Peer-to-Peer Information Retrieval (P2PIR) is being tackled by researchers attempting to provide valuable insights and to propose solutions to use it successfully. Nearly, all published studies have been evaluated by simulation means, using well known document collections (usually acquired from TREC). On the IR side, in a P2P network, the distribution of documents is, to a significant scale, a result of the previous location and retrieval. However, this also depends on the application specification and/or on other non-functional requirements that may be imposed (such as copyright considerations, etc.). Defining and simulating user behaviour, especially in a very large distributed system, is a complex and intimidating task. The problem with such approaches is a twofold. Firstly, there are cases where the documents distribution does not successfully reflect the application scenario and therefore such evaluation results are hardly conclusive. Secondly, each individual considers a different testbed for experimental evaluation, the mutual comparison and the quantification of performance improvements become an impossible task.

Organising documents according to their content, and consequently, achieving more accurate and effective retrieval is, arguably, one of the principal goals of IR research. Document clustering has been a particularly active research field within the Information Retrieval (IR) community [2][3][4][5]. The reason behind this, apart from a natural human tendency [6], is that by clustering, documents relevant to the same topics tend to be grouped together (the Cluster Hypothesis [2]). Addressing these issues, we propose a testbed, suitable for the evaluation of P2PIR systems.

This paper is organized as follows. Section 2 defines the notion of testbed and Section 3 reviews related work about testbed in P2P retrieval. In Section 4, we detail our proposal and we are showing our first experimental results in Section 5. Section 6 concludes and gives some open issues.

II. NOTION OF TESTBED

- Centralized Information Retrieval Testbed:
  Dekhtyar [7] defines IR testbed by the following formula:
  Testbed = DataSet + Tasks + Answers + Evaluation measure + Data Formats. Indeed, a testbed must provide the documents and the queries to be raised on these documents. The answers to the queries are often data provided by experts, together with the relevance judgements. Evaluation measures are the tools which the testbed uses in order to test the relevance of the IR algorithms. Data Formats, relates to the existence of testbed under various formats of possible data.
- Distributed Information Retrieval Testbed:
  In a distributed context, new information must be defined; how to distribute the data on the various nodes of a network and which replication law to apply? In addition, we define the elements which a distributed testbed must provide:
  Distributed Testbed = documents collection+ queries collection + documents and queries distribution method

 ISBN: 978-1-61208-124-3

among peers + documents and queries replication method among peers + evaluation metrics+ queries responses.

Based on this notion of testbed, we propose in this paper, a cluster-based testbed. Before presenting the main features of our testbed, it is important to present a brief state of the art of some existing testbeds in a centralized and distributed context.

III. Background and Related Work

A. Testbeds for centralized systems

For centralized Information Retrieval, there exist an important number of standard centralized benchmarks, such as the yearly competitions conducted by the Text Retrieval Conference, or TREC [8], DMOZ [9], etc.

TREC, co-sponsored by the National Institute of Standards and Technology (NIST) and U.S. Department of Defense, was started in 1992 as part of the TIPSTER Text program [10]. Its purpose was to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval approach. In 2001 and 2002, the conference organized evaluating campaigns segmentation, indexing and searching content in the video [11]. For each version of TREC, NIST provides a collection of test. However, TREC is only available to registered participants of the conference. Other benchmarks repeatedly deployed in the literature include the Initiative for the Evaluation of XML Retrieval, or INEX [12], benchmark. The test collection consists of a set of XML documents, topics and relevance assessments. The topics and the relevance judgments are obtained through a collaborative effort from the participants. On the on-line topic submission, retrieval result submission, relevance judgment task, and evaluation metrics will be provided by INEX. Relevance assessments will be provided by the participating groups using INEXs on-line assessment system.

B. Testbeds for decentralized systems

In recent years, distributed information retrieval systems based on Peer-to-Peer (P2P) architectures have been increasingly attracting attention [13][14][15][16][17][18][19]. These systems usually consider a collaboration of peers where each peer stores a subset of the globally available documents. Being influenced by information retrieval in centralized systems, a substantial fraction of authors in the field of distributed IR evaluate their approaches by partitioning well known centralized IR testbed collections, such as the one provided by TREC, into (overlapping or disjoint) fragments. However, the assignment of documents to peers is not standardized and is performed differently by the authors, rendering the comparison of experimental results a bothersome task. Also, built testbeds is a challenge in distributed information retrieval systems and in particular in P2PIR systems.

IV. P-KMeans: a cluster-based testbed for P2PIR

In P2P network, each peer usually has a homogeneous collection of documents representing the interests of its user. Intuitively, clustering similar documents will help to discover useful resources and prune the searching space. Therefore, clustering similar documents will benefit information retrieval in P2P systems.

Clustering algorithms partition a set of objects, documents in our case, into groups called clusters. The classical algorithm K-means was introduced and drawing by Hartigan [20]. This algorithm is a classification tool that allows reserve a set of data in k homogeneous classes; k is fixed by the user. It affects each object, randomly, to a region and we iterate as follows: the centers of the different groups are recalculated and each object is assigned to a new group, based on the nearest center. Convergence is reached when the centers (also called (centroids)) are fixed [21].

But k-means also has drawbacks, among which we can mention: the method does not scale to large data collections. Indeed, most traditional methods of clustering are easily affordable but can not be applied to large collections of data. Their space complexity is often too great. It follows that it is interesting to seek an algorithm that is based on K-Means to enjoy these benefits and that adapts to a large-scale distributed environment. To obtain a semantics distribution on different network nodes, we first apply the algorithm K-Means on the document collection. We noticed that K-Means takes into account that small collections. Following this finding, we used an empirical study to determine the maximum number of documents processed by both K-Means. The results of this study is that this algorithm treats up to 5000 objects (*i.e.*, documents). For this, we thought about implementing a clustering algorithm (Peers KMeans or *P-KMeans*). The objective of this algorithm is to define a method of distributing documents on peer and overcome the main drawback of the algorithm K-Means. P-KMeans algorithm takes as input a documents collection $\mathcal{DF}$, the number of peers in the network k, the number of documents in $\mathcal{DF}$ and the number of documents processed at each iteration. It will output the set of k-clusters $\mathcal{P}_k$.

The document distribution algorithm operates in three stages:

- Clustering of documents:
  All documents in the $\mathcal{DF}$ collection is partitioned according to the number of documents processed at each iteration n. The pseudo-code for the partitioning is given by Algorithm 2. The notations used are summarized in Table I. The partitioning algorithm takes a documents collection (*i.e.*, $documentsDF$), the number of documents in $\mathcal{DF}$ and the number of documents processed at each iteration as input. It produces the subsets of documents (*i.e.*, $DocDef$). For any subset

k	:	peers number in network.
n	:	♯documents processed at each iteration.
df_i	:	subset of documents.
$\mathcal{N}$	:	♯documents in $\mathcal{DF}$.
$\mathcal{DF}$	:	documents collection.
$\mathcal{P}_k$	:	k-clusters.
$DocDef$	:	set of all documents.
$clusterFiles$	:	set of files containing documents clusters.
$centroidFiles$	:	set of files containing centroids clusters.
$clusterCentroidsFiles$	:	set of files containing centroids clusters.

Table I
P-KMeans ALGORITHM NOTATIONS.

df_i of $DocDef$, we apply *adaptedKMeans* algorithm (line 8-9) that takes df_i and the number of k peers in the network as input. It produces df_i documents groups (*i.e.*, $clusterFiles$) and these centroids groups (*i.e.*, *centroidFiles*).

- Clustering of centroids:
  Centroids (*i.e.*, *centroidFiles*) already generated previously, are grouped by K-Means algorithm (line 10) to produce centroids cluster (*clusterCentroidsFiles*).
- Mapping between document clustering and centroids clustering:
  This step is the intermediate step between the clustering of documents and the clustering of centroids to obtain the conclusion of clustering documents. The pseudo-code for this step is given by algorithm 3. The notations used are summarized in Table I. The mapping algorithm takes as input all documents clusters ($clusterFiles$) and all centroids clusters ($centroidFiles$), it produces k-clusters set (*i.e.*, P_k).

Algorithm 1: P-KMEANS

```
1  Algorithm: P-KMEANS(DF, k, N, n)
   Input:
   DF: documents collection.
   k: peers number in network.
   N: documents number in DF.
   n: documents number at each iteration.
   Output:
   P_k.
2  begin
3      DocDef ::= partitionDF(DF, N, n);
4      centroidFiles := {∅};
5      clusterFiles := {∅};
6      clusterCentroidsFiles := {∅};
7      foreach df_i ∈ DocDef do
8          clusterFiles := clusterFiles ∪
           adaptedKMeans(df_i, k);
9          centroidFiles := centroidFiles ∪
           adaptedKMeans(df_i, k);
10     clusterCentroidsFiles :=
       KMeans(centroidFiles, k);
11     P_k := mapping(clusterCentroidsFiles,
       clusterFiles);
12     return (P_k)
13 end
```

Algorithm 2: PARTITIONDF

```
1  Algorithm: partitionDF(DF, N, n) Input:
   DF: documents collection.
   N: documents number in DF.
   n: documents number traits at each iteration.
   Output:
   DocDef.
2  begin
3      DocDef := {∅};
4      for (i=0 ; i ≠ N/(n − 1); i++) do
5          df_i := Partition(DF, n, i) ;
6          DocDef := DocDef ∪ df_i ;
7      return DocDef
8  end
```

V. EXPERIMENTS

A. Experimental Environment

- PeerSim simulator:
  To evaluate the approach proposed in this paper, we have chosen to use the PeerSim [22] simulator which is an open source tool written in Java. It has the advantage of being dedicated to the study of P2P systems. It has an open and modular architecture allowing it to be adapted to specific needs. More precisely we use an extension of PeerSim developed by the RARE project [23]. This extension can be seen as a PeerSim specialization for information retrieval.
- Source Data:
  As a data set, we used "BigDataSet", produced under the RARE project [23]. It was obtained from a statistical analysis on data collected from the Gnutella [24] system and data TREC collection, which allows us to perform simulations in real conditions. BigDataSet is composed of a set of documents (25000), a queries set (4999) , a set of peers (500) and a queries distribution on peers. It provides XML files describing the system nodes and the documents they possess, as well as queries which will be launched on the network [25].
- Routing Algorithms
  Routing models used here, are Gnutella and LPS. Gnutella is a system that used a simple constrained flooding approach for search. A query was forwarded to a fixed number of neighbors until its time-to-live (TTL) in terms of forwarding steps was exhausted or a loop was detected.

Algorithm 3: MAPPING

Algorithm: *mapping*($clusterCentroidsFiles$, $clusterFiles$)
Input:
$clusterCentroidsFiles$: set of files containing centroids clusters.
$clusterFiles$: set of files containing documents clusters.
Output:
$DocDef$.
begin
  $DocDef := \{\emptyset\}$;
  foreach ($cluster_i \subset clusterCentroidsFiles$) **do**
    foreach ($c \in cluster_i$) **do**
      D_c :=extractCentroidDocs(c, $clusterFiles$);
      $c := D_c$
  return $\mathcal{P}_k$;
end

LPS is an algorithm for routing queries based on learning implicit behavior of users that is deducted from queries history [26].

- Evaluation Metrics
  In an IR system, the system's success or rejection is based on how effectiveness is measured. Recall (R), Precision (P) and F-score (the harmonic mean of precision and recall) measures have been widely used as fundamental measures to test the effectiveness of IR systems [27]. Let RDR, the number of relevant documents returned. Let RD, the number of relevant documents. Let DR, the number of documents returned. These measures are defined as follows:

$$R = \frac{RDR}{RD} \quad (1)$$

$$P = \frac{RDR}{DR} \quad (2)$$

$$F - score = 2 * \frac{P * R}{P + R} \quad (3)$$

- Initialize simulation parameters
  The simulation, of both algorithms *LPS* and *Gnutella*, is based on the parameters:
  - *TTL* (Time To Live): Maximum depth of research, initialized to 5.
  - *Pmax*: Maximum number of peers which the query should be propagated to.
  - *Overlay size*: Number of peers in the network, initialized to 500.
  - *Replication degree*: We used the same *Zipf* replication degree that is 40.

B. Testbeds for Evaluating

We performed our evaluation using the testbeds proposed in [28]. These testbeds are based on "BigDataSet" collection, produced under the RARE project [23], and are designed to address a number of P2P IR applications through different document distributions and concentrations of relevant documents. The individual testbeds used are the following:

- **UBZR**: This testbed is designed for the simulation of systems where the documents are distributed uniformly across the peer population.
- **RBZR**: In this testbed, documents assignment is done in a completely random manner.
- **SB**: This testbed aims to reflect a P2PIR scenario. Relevant documents are distributed among a small number of peers. Each peer usually has a homogeneous collection of documents representing the interests of its user.

C. Experimental Results

Our experiments aim to determine the impact of different testbeds on routing algorithms performance. We compared *CB* (based on P-Kmeans algorithm) testbed with *UBZR*, *RBZR* and *SB*. Figures 1 and 3 show the results for Gnutella algorithm when applying the different testbeds. Figure 2 and 4 shows the results for LPS algorithm under different testbeds. Former tests presented here are, in our opinion, very encouraging. By comparing our testbed with existing ones, we evaluate that our testbed is competitive.

A search algorithm is substantially in influence by used type of distribution. Indeed, a semantics data distribution, such the case of *CB*, gives the best results compared to other testbeds. Indeed, distribute data according to thematic, such as *CB* testbed, sought may be beneficial both for flooding routing algorithm (case of Gnutella) and a semantic algorithm (case of LPS) and thus with recall and F-score.

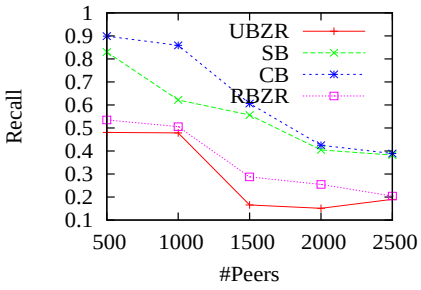

Figure 1. Relation between Recall and Nbr of peers according to different evaluation testbeds for Gnutella

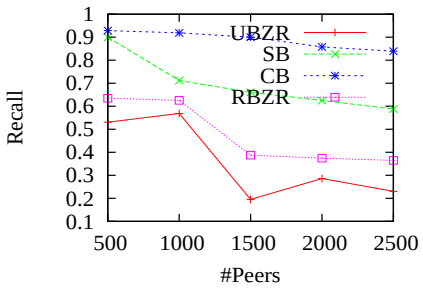

Figure 2. Relation between Recall and Nbr of peers according to different evaluation testbeds for LPS

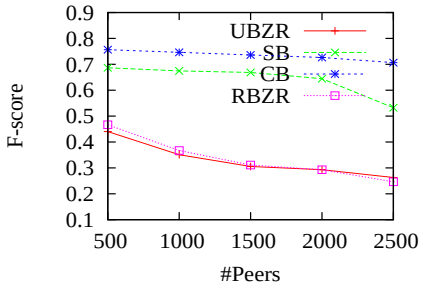

Figure 3. Relation between F-score and Nbr of peers according to different evaluation testbeds for Gnutella

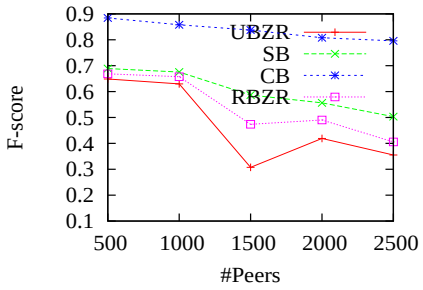

Figure 4. Relation between F-score and Nbr of peers according to different evaluation testbeds for LPS

VI. Conclusion and Future Works

The field of information retrieval is very experimental in nature. We identify the need to create testbeds for information retrieval experimentation. We propose *CB*, a testbed for P2PIR, based on clustering algorithm (P-Kmeans).
In our testbed, recall and F-score (harmonic mean) are implemented as two instances of the evaluation element. Finally, this work can be followed by the use of other collections (*e.g*, INEX, DMOZ, etc.) and development of more realistic distribution methods, by building a real centralized collection (documents, queries and relevance judgments) from P2P network data.

References

[1] D. L. Lee, D. J. Zhao, and Q. Luo, "Information retrieval in a peer-to-peer environment," in *Proceedings of the 1st international conference on Scalable information systems*, New York, NY, USA, 2006.

[2] C. J. van Rijsbergen, *Information Retrieval*. London: Butterworths, 1979.

[3] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of American Statistical Association*, vol. 58, no. 301, pp. 236–244, March 1963.

[4] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *The Computer Journal*, vol. 26, no. 4, pp. 354–359, 1983.

[5] Q. He, "A review of clustering algorithms as applied in IR," Ph.D. dissertation, University of Illinois, 1999.

[6] R. M. Cormack, "A review of classification," *Journal of the Royal Statistical Society*, vol. Series A, no. 134, pp. 321–353, 1971.

[7] A. Dekhtyar and J. Hayes, "Good Benchmarks are Hard To Find: Toward the Benchmark for Information Retrieval Applications in Software Engineering," in *IEEE International Conference on Software Maintenance (ICSM 2007)*, France, October 2007, pp. 1–3.

[8] TREC, "Trec web site," http://trec.nist.gov/, 18/February, 2010.

[9] DMOZ, "Dmoz web site," http://www.dmoz.org/, 18/February, 2010.

[10] D. K. Harman, "Overview of the first text retrieval conference (trec1)," in *Proceedings of the First Text REtrieval Conference (TREC1), NIST Special Publication*, 1993, pp. 207–500.

[11] S. Robertson, "Introduction to the special issue: Overview of the TREC routing and filtering tasks," *Information Retrieval*, vol. 5, no. 2-3, pp. 127–137, 2002.

[12] INEX, "Inex web site," http://inex.is.informatik.uni-duisburg.de/, 18/February, 2010.

[13] J. Lu and J. Callan, "Content-based retrieval in hybrid peer-to-peer networks," in *Conference on Information and knowledge management (CIKM 2003)*, New Orleans, USA, 2003, pp. 199–206.

[14] K. Aberer, P. Cudr-Mauroux, M. Hauswirth, and T. V. Gridvine, "Pelt: Building internet-scale semantic overlay networks," in *nternational Semantic Web Conference (ISWC 2004)*, 2004.

[15] S. Idreos, M. Koubarakis, and C. Tryfonopoulos, "P2p-diet: An extensible p2p service that unifies ad-hoc and continuous querying in super-peer networks," in *Special Interest Group on Management of Data (SIGMOD 2004)*, 2004.

[16] H. Nottelmann, G. Fischer, A. Titarenko, and A. Nurzenski, "An integrated approach for searching and browsing in heterogeneous peer-to-peer networks," in *Heterogeneous and Distributed Information Retrieval (HDIR 2005)*, 2005.

 ISBN: 978-1-61208-124-3

[17] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer, “Minerva: Collaborative p2p search,” in *Very Large Data Bases (VLDB 2005)*, 2005, pp. 1263–1266.

[18] F. C. Acuna, C. Peery, R. P. Martin, and T. D. Nguyen, “Planetp: Using gossiping to build content addressable peer-to-peer information sharing communities,” 2003.

[19] T. Suel, C. Mathur, J. wen Wu, J. Zhang, A. Delis, M. Kharrazi, X. Long, and K. Shanmugasundaram, “Odissea: A peer-to-peer architecture for scalable web search and information retrieval,” in *Web and Databases (WebDB 2003)*, 2003.

[20] H. Bock, “Origins and extensions of the k-means algorithm in cluster analysis,” *Electronic Journal for History of Probability and Statistics*, vol. 4, no. 2, pp. 1–18, December 2008.

[21] S. Siersdorfer and S. Sizov, “Automatic document organization in a p2p environment,” in *European Conference on Information Retrieval (ECIR 2006)*, London, 2006, pp. 265–276.

[22] Peersim, “The peersim simulator,” http://peersim.sf.net, 18/February, 2010.

[23] RARE, “Le projet rare (routage optimisé par apprentissage de requêtes),” in *http ://www-inf.int-evry.fr/defude/RARE/*, 2010.

[24] Gnutella, “Gnutella web site,” http://www.gnutella.com/, 18/February, 2010.

[25] R. Mghirbi, K. Arour, Y. Slimani, and B. Defude, “A Profile-Based Aggregation Model in a Peer-To-Peer Information Retrieval System,” in *Data Management in Grid and P2P Systems (Globe 2010)*, Spain, September 2010, pp. 148–159.

[26] T. Yeferny and K. Arour, “LearningPeerSelection: A Query Routing Approach for Information Retrieval in P2P systems,” in *International Conference on Internet and Web Applications and Services (ICIW 2010)*, Spain, May 2010, pp. 235–241.

[27] M. Renda and U. Straccia, “Web metasearch: rank vs. score based rank aggregation methods,” in *Symposium on Applied computing (SAC 2003)*, New York, 2003, pp. 841–846.

[28] S. Zammali and K. Arour, “P2PIRB: Benchmarking Framework for P2PIR,” in *Data Management in Grid and P2P Systems (Globe 2010)*, Spain, September 2010, pp. 100–111.

 ISBN: 978-1-61208-124-3