An Evaluation of a Cluster-Based Architecture for Peer-to-Peer Information Retrieval

Iraklis A. Klampanos and Joemon M. Jose

Department of Computing Science
University of Glasgow
United Kingdom

Abstract. In this paper we provide a full-scale evaluation of a cluster-based architecture for P2P IR, focusing on retrieval effectiveness. We observe that there is a significant difference in performance between the architecture we examine and a centralised index. After inspecting our experimental methodology and our results, we provide evidence that suggests that this discrepancy is due to the information clustering algorithms employed throughout. The construction errors of the resource descriptions as well as the failure of the clustering mechanisms to discover the structure of the smallest of peer-collections lead to erroneous query routing. We proceed further to show experimentally how content replication and relevance-feedback mechanisms can help to alleviate the problem.

1 Introduction

Information retrieval (IR) over peer-to-peer (P2P) networks is a challenging problem that is frequently referred to in the IR literature ([1,2,3,4,5], etc.). A number of architectures have been proposed that address various instantiations of this problem. It is clear that different applications of P2P networks will pose different challenges for IR. Popular applications of P2P IR include digital libraries, open information-sharing and others ([1,2] etc.). Information clustering is often used by various studies as an architectural component or as a tool for achieving realistic evaluation environments. However, the application of clustering in P2P IR may lead to errors in the cluster centroids. These errors are caused by the inadequate information that describes the constituent objects. However, the effects of this problem have not been studied within the context of P2P IR and so we do not know the extent of the problem, let alone which solutions could be applied in order to amend it. These are the issues that this paper contributes insight and solutions for.

In this paper we provide a wide-scale experimental evaluation of a cluster-based P2P IR architecture [2], using a set of testbeds that were devised for this purpose [6]. Through clustering, this architecture attempts to organise the shared content into semantically-related peer-groups. The testbeds employed are totally independent of the experimental evaluation process itself. As our initial effectiveness results are poor, we provide insight into what may be causing this behaviour and we propose solutions that we justify experimentally.

In the next section we present the cluster-based architecture our study is based on as well as the experimental testbeds we use for our experiments. In Section 3 we present

R. Wagner, N. Revell, and G. Pernul (Eds.): DEXA 2007, LNCS 4653, pp. 380–391, 2007.

an initial evaluation that is targeted on retrieval performance. In Section 4 we narrow down our evaluation on a near-optimal (for retrieval purposes) subset of the original testbeds. We use this smaller collection in order to focus on various individual aspects of the architecture, isolate potential problems and suggest potential solutions. Finally, in Section 5 we present our conclusions and provide pointers for future work.

2 Related Work

2.1 A P2P IR Architecture

We base our evaluation on an architecture [2] that employs clustering at two levels: first, in order to derive usable resource description vectors from the participating information providers and, subsequently, in order to generate content-aware peer-groups (CAGs). The ultimate goal is to form groups of peers that share similar content. The main hypothesis behind this organisation is that it can, potentially, increase the retrieval effectiveness through selective query routing, *i.e.* bypassing irrelevant information sources. Content-based network organisation also increases efficiency since it avoids uninformed query-routing strategies, such as query flooding.

Another property of this architecture is that it is hybrid (*i.e.* there exist super-peers with additional administrative responsibilities) and service-oriented (please refer to [2] for the exact services that are identified). For our evaluation purposes peers are either hubs, *i.e.* peers that are responsible for managing connections and routing messages, or information providers, *i.e.* peers that share documents with the rest of the network.

2.2 Testbeds for Evaluating P2P IR

The evaluation of P2P IR systems is an intimidating task due to the potential size of the network and the total volume of the shared information. An additional challenge is posed by factors having to do with the distribution of documents among the peer-collections, the concentration of relevant documents in the evaluation testbeds etc. Different potential applications of P2P IR technologies exhibit different such properties. Since these factors, generally, affect retrieval performance, they have to be taken into account during evaluation.

We performed our evaluation using the testbeds proposed in [6]. These testbeds are based on TREC's WT10g collection 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:

ASISWOR. This testbed is designed to reflect the properties of open information-sharing environments. It exhibits a steep power-law distribution of documents. In this testbed, each web-domain of WT10g corresponds to a peer-collection.

UWOR. This is a testbed designed to address P2P IR in environments where the documents are uniformly distributed across the participating information providers. Such environments may include strict DRM environments, networks of devices with restricted resources etc.

DLWOR. This testbed aims to reflect a digital-library setting. The number of collections are less than in the ASISWOR, making the individual collections larger in average.

DLLC. This is a testbed originally proposed and made available by Lu and Callan in [1] and it also addresses the problem of P2P IR in digital libraries.

3 Initial Evaluation

3.1 Methodology and Parameters

The evaluation we present in this paper is simulation-driven. For simulating this architecture one has to take under consideration a number of parameters that affect its behaviour. These parameters, having to do with content representation and network topology, are presented in the following sections.

Content Descriptions. In the proposed architecture, content descriptions are used at two stages: by information providers that advertise their content to hubs and by hubs that organise the network performing some kind of clustering. In this study, content descriptions are either term-frequency (TF) or binary vectors [1].

Network Topology. The topology of the network depends primarily on how the hubs group the information providers. For the evaluation of this architecture we implemented two different approaches to this organisation. The first is to cluster the clusters of the information providers using single-pass clustering in its simplest form (*Simple* topology). The second alternative is to use a fixed number of CAGs, as attractors for the information providers. For the experiments that follow, we used the largest relevant document for each topic of WT10g as CAG attractors (*Fixed* topology).

3.2 IR-Related Results

For our evaluation we assessed the underlying document collection (WT10g) against the standard 100 TREC topics as a centralised index. Even though these results are not directly comparable to the results from the P2P architecture, they provide a point of reference for discussion. The results of the centralised index run are presented in Table 1.

Table 1. IR effectiveness for WT10g as a centralised index

Topics	*Relevant*	*Retrieved*	*Rel. Retrieved*	*P@10*
100	5,980	97,048	3,817	0.2960

***Simple* Single-Pass Topology.** For the Simple topology, a simple single-pass clustering algorithm [8] was used to cluster peer-centroids into CAGs. For this, we did not cap the number of CAGs to be created. The results for IR effectiveness can be seen in Table 2.

[1] Even though binary vectors are thought to lead to worse IR effectiveness, it has been reported [7] that there is no evidence to suggest that are inferior to TF vectors for clustering.

Table 2. IR effectiveness across non-replication testbeds for the Simple topology

Testbed	*Threshold*	*CAGs*	*Topics*	*Relevant*	*Retrieved*	*Rel. Retrieved*	*P@10*
ASISWOR	0.05	57	46	3,562	7,525	83	0.0196
	0.1	145	22	1,050	3,184	27	0.0000
	0.2	559	16	954	1,979	41	0.0125
UWOR	0.05	70	30	2,248	5,900	23	0.0233
	0.1	203	28	2,064	5,050	49	0.0250
	0.2	523	10	437	1,400	14	0.0600
DLWOR	0.05	44	35	2,051	5,300	36	0.0057
	0.1	126	17	952	2,700	9	0.0059
	0.2	471	16	1,112	1,850	48	0.0063
DLLC	0.05	17	20	1,226	3,076	12	0.0100
	0.1	64	14	745	1,776	26	0.0286
	0.2	272	9	606	887	15	0.0111

Table 3. IR effectiveness across non-replication testbeds for the Fixed topology

Testbed	*Threshold*	*Topics*	*Relevant*	*Retrieved*	*Rel. Retrieved*	*P@10*
ASISWOR	0.05	61	3,987	10,912	158	0.0393
	0.1	37	2,328	6,829	56	0.0189
	0.2	15	773	2,277	19	0.0067
UWOR	0.05	55	3,725	10,100	101	0.0164
	0.1	37	2,320	6,500	28	0.0108
	0.2	14	761	2,300	13	0.0000
DLWOR	0.05	59	3,892	10,950	182	0.0492
	0.1	37	2,328	6,900	61	0.0054
	0.2	13	759	2,000	26	0.0231
DLLC	0.05	56	3,800	9,150	152	0.0286
	0.1	34	2,272	5,700	52	0.0206
	0.2	13	621	1,600	23	0.0615

The column entitled *Threshold* corresponds to the threshold that was used for the document clustering as well as for the query routing that took place after the topology was created. The column *CAGs* shows the number of CAGs that were created with the given threshold. The column *Topics* shows the number of topics that were successfully routed to the network for matching. This number depends on the routing threshold. The initiating hub only routes a query to a CAG if its similarity to the CAG's centroid is higher than this threshold. The column *Relevant* shows the number of relevant documents for the number of topics that responses were given for. This comes from the relevance assessments provided by TREC for WT10g. The column *Retrieved* shows the number of documents that were retrieved in total, while *Rel. Retrieved* shows the number of relevant documents that were retrieved. Last, *P@10* is the precision achieved for the first 10 results in the result list, averaged over all the topics that got evaluated.

From this table we can see that there is a significant difference in retrieval effectiveness when compared to the results we obtained for the centralised index of Table 1. Even though these results may seem rather poor, one has to keep in mind a number of factors that are known to affect retrieval. First, WT10g is a web collection and therefore its documents cannot be expected to be of the same quality as the ones in other collections of documents such as collections of journal articles. Another important factor is the lengths of the documents. In the web, most documents are very small. This affects matching and, more importantly for this architecture, clustering. Very small documents (like very large documents) are harder to relate to other documents and classify automatically. Therefore, these results are not as surprising as they may seem at first, especially since no measures have been taken to counteract the aforementioned issues.

***Fixed* Topology.** For this topology we created a fixed number of CAGs based on the 100 TREC topics and their relevance assessments. We took the largest relevant documents for all the topics and used them as attractors for the rest of the documents. This gave us a topology of 94 CAGs – 2 topics have no relevant documents while 4 more did not attract any other documents apart from themselves. The retrieval effectiveness results we obtained are shown in Table 3. In this table, *Threshold* corresponds to the routing thresholds only, since we did not threshold similarity during the CAGs creation. The rest of the columns have the same meaning as their counterparts in Table 2, explained in the previous Section.

It can be seen in Table 3 that the effectiveness for this topology is very low and comparable to that exhibited by the Simple topology presented in the previous Section. This may seem unexpected as a result. Indeed, we included this alternative topology expecting to achieve significantly higher retrieval effectiveness, especially since the attractor documents were based on the topics that we would eventually evaluate against. This is a strong hint that there is a more important factor involved that impedes effectiveness. We believe that this factor has solely to do with the formation of cluster centroids and we will be analysing it further in Section 4.

4 Evaluating on an Optimal Testbed

In this section we re-assess the architecture using a small and near-optimal testbed based on the ASISWOR testbed of Section 2.2. We used ASISWOR as a base for our near-optimal testbed because it addresses openly available information-sharing environments and, as such, it is arguably the most generally applicable environment for the given architecture. We choose a smaller and more manageable testbed in order to better analyse and understand the P2P IR architecture and, therefore, to discover its pathological sources in a better controlled environment.

4.1 Characteristics and Conditions

Testbed Characteristics. The minimal ASISWOR testbed is near-optimal for the IR-based evaluation we will be presenting because it has a very high concentration of relevant documents. It was derived by randomly removing non-relevant documents from peer-collections also randomly picked. It consists of 4834 documents in total, spanning 1316 peers. 2267 of these documents are the relevant documents of the 100 standard

TREC topics while the rest were left intentionally in order to preserve some minimal distortion.

The, relatively to the total number of documents, large number of peer-collections ensured some skewness in the document distribution. This skewness is an important property that makes the ASISWOR testbed realistic and so even partially retaining it in the minimal testbed is important. The maximum number of documents a peer-collection has is 137, while 71% of the collections have 1 or 2 documents.

Minimal ASISWOR as a Centralised Collection. Similarly to the previous section, we provide the testbed's IR behaviour as a centralised corpus. These results are shown in Table 4.

Table 4. The retrieval effectiveness of minimal ASISWOR as a centralised collection

#Topics	*Relevant*	*Retrieved*	*Rel_Retrieved*	*P@10*
100	5980	47710	4596	0.6900

4.2 Evaluation Results

The overall retrieval effectiveness results are presented in Table 5. These results show that the IR effectiveness of the architecture is still at very significant odds compared to its centralised counterpart (Table 4). The sources of this discrepancy include the following:

1. The testbed does not encapsulate any structure to be found by the clustering mechanisms of the architecture.
2. The clustering mechanisms fail to discover the structure in the testbed.
3. The routing fails to locate enough relevant sources for the query to get forwarded to.

However, for this study instead of discussing these issues further, we will take them for granted, as a property of a realistic environment for P2P IR. Instead, we will pursue potential solutions that might help us to counter them[2].

Table 5. Results on retrieval effectiveness

	#Topics	*#CAGs*	*Relevant*	*Retrieved*	*Rel_Retrieved*	*P@10*
S-P – **0.0**	87	1	5475	3550	740	0.2678
S-P – **0.05**	19	10	1976	747	123	0.1737
S-P – **0.1**	16	30	1573	78	54	0.2562
S-P – **0.2**	16	139	1335	62	40	0.2125
FIXED	89	89	5530	16162	551	0.0596

[2] Additional experimental evidence, not presented herein, suggests that the fundamental assumptions made by both the architecture and the minimal-ASISWOR testbed hold. Hence they were omitted from this paper.

4.3 Compensating for Distortion

In Section 3 we showed experimentally that a two-level clustering, especially on small collections, can potentially limit the retrieval effectiveness of our P2P IR architecture. However, in our treatment, we neglected to look into a feature present in other architectures and indeed a very important feature for P2P networks in general, namely replication [4,9,10,1]. In this section we will look into whether replication can improve retrieval effectiveness. We will also look into term-weight adjustment, as this can result from relevance-feedback. Without arguing for a particular relevance-feedback implementation, we will show that weight-adjusted resource descriptions can increase the retrieval effectiveness in cluster-based P2P environments.

Replication. In order to assess the effect of replication on the P2P IR architecture, we implemented a replication strategy based on hypothetical popularities for the standard TREC topics. In our implementation, popularity is represented by a real number within the range $[0, 1)$ with 0 representing a topic that is not popular at all. The relevant documents to the topics are replicated to a number of peers according to their corresponding topic's popularity value, *i.e.* a document whose topic is popular has more chances to reside to another peer-collection etc. In order to calculate these popularities we used an inverse power law. Where, according to power-law, $y = \alpha x^k$, in our case, a popularity score s_t, for a topic t, is given by $s = \alpha / r^k$, where α is a constant that determines the popularity score for the most popular topic, r is the rank of the topic with 1 being the most popular and k is the exponent that determines the skewness of the output values. Once all topics have been assigned a popularity score, our algorithm iterates over all relevant documents and peer-collections and replicates documents randomly, according to their topic's score. This technique allows us to introduce realistic replication, scaled-down to the number of topics that we experiment on. For our experiments we took $\alpha = 0.9$ and $k = 2$. The α value ensures that no document gets replicated to all the peer-collections, while the k value ensures that the trend of the popularities is not too steep so as to get meaningful replication for at least some of the topics.

For our experiments we created seven minimal testbeds with different arrangements of replicated content. This was done because of the element of randomness involved in the replication process described in the previous paragraph. The IR effectiveness results can be seen in Table 6. Comparing this table to Table 5 we notice two important differences: first, the effectiveness in the testbeds with replication is higher than in the testbed without. In particular, after the introduction of replication, for the testbeds used, we get an average P@10 of 0.4071, while in the testbed without replication, for the same threshold, *P@10* is 0.2562. On the other hand we notice that the number of topics that get to be answered (column *#Topics*) in the testbeds with the replication is much smaller (average of 1.86) than its corresponding figure for the testbed without replication (16). These two artifacts show that there is a significant improvement in effectiveness when replication is introduced, but only for the popular topics. In fact, the rest of the topics do not even get to be answered, *i.e.* their similarity to any CAG description falls below the threshold. This behaviour can be explained by looking into the cosine similarity measure that is used. When more similar documents, about a particular topic, are included in a cluster centroid (or a resource description for our purposes),

Table 6. Results on retrieval effectiveness on testbeds with replication. These results were obtained for a threshold of 0.1. The size of the original minimal testbed (before introducing replication) is 4834 documents.

	Size	*#Topics*	*Relevant*	*Retrieved*	*Rel_Retrieved*	*P@10*
Testbed 0	$68,469$	3	316	600	169	0.3667
Testbed 1	$81,081$	3	316	600	186	0.3333
Testbed 2	$65,199$	2	310	400	177	0.5000
Testbed 3	$75,454$	1	269	200	13	0.4000
Testbed 4	$101,168$	1	269	200	101	0.5000
Testbed 5	$23,729$	1	269	200	112	0.6000
Testbed 6	$153,885$	2	59	350	20	0.1500
Average	$81,283.57$	1.86	258.29	364.29	111.14	0.4071

the similarity between this centroid and any topic other than the heavily replicated ones decreases. In this particular case, this decrease pushes the similarity below the lowest acceptable threshold, hence the small number of topics that get answered. Even though this seems to be a drawback, we believe it is to be expected in a large and widely available P2P information-sharing environment, where the potential number of topics are in the millions, not just one hundred. We believe that in such an environment the system could work sufficiently well for the majority of the users.

Relevance Feedback. For experimenting, we use the relevance assessments in order to alter the CAG centroids instead of the queries (the standard relevance-feedback application). Our goal is to counter-balance the noise that is introduced by the two-level clustering, by filtering, not augmenting, the document vectors. We assume that a relevance-feedback mechanism exists, which allows the aforementioned modification of resource-description vectors. For an original term weight t_i of a CAG centroid, t_i becomes $t_i + 0.5$ if t_i is relevant (*i.e.* being a term of a document that is relevant to any of the TREC topics); otherwise t_i becomes $t_i - 0.5$. In other words, the terms that describe relevant documents, collectively, to any of the topics, get promoted by 50% of their original weight while the rest get demoted by the same percentage. For these experiments we only adjusted the CAG centroids. Alternatively we could have also adjusted the cluster centroids that form the resource descriptions of the information-providers. This would lead to the re-clustering of these peers into new CAGs and possibly to better performance. However, while the creation of the CAG descriptions is a responsibility of the network, the creation of the cluster descriptions is a responsibility of the participating information providers. We did not want to directly adjust the cluster descriptions of the information providers since the architecture assumes that they are autonomous and trusted.

Evaluating this adjustment on the minimal ASISWOR testbed gives the results in Table 7.In Table 8 we summarise the difference in performance between the two different flavours of the architecture. From this table, apart from the aforementioned difference in P@10, we also note that the architecture with the hypothetical relevance-feedback mechanism manages to address more topics than the basic one. Beside this difference between the architectures one can observe that more topics are addressed for the higher

Table 7. Results on retrieval effectiveness with relevance-feedback term-weighting on the resource descriptions

	#Topics	*#CAGs*	*Relevant*	*Retrieved*	*Rel_Retrieved*	*P@10*
S-P – **0.05**	39	10	3543	1540	404	0.2667
S-P – **0.1**	32	30	3329	997	293	0.2844
S-P – **0.2**	49	138	4294	823	314	0.2510

Table 8. Comparison of IR effectiveness between the basic and relevance-feedback architecture

Threshold	0.05		0.1		0.2	
	Basic	*RelFbk*	*Basic*	*RelFbk*	*Basic*	*RelFbk*
#Topics	19	39	16	32	16	49
P@10	0.1737	0.2667	0.2562	0.2844	0.2125	0.2510

threshold of 0.2. In this case, this is a desirable fact, since the effective routing of more topics does not hinder the overall performance (as measured by P@10) of the system.

Applying Weight-Adjustment along with Replication. Having observed how the retrieval effectiveness increases when using relevance-feedback-based weight adjustment and replication separately, in this section we look into the effectiveness when both these mechanisms are applied. For the experiments presented below we adjusted the weights of the CAGs in the small testbeds we used in Section 4.3. The results in effectiveness are summarised in Table 9.

Table 9. Results on retrieval effectiveness on testbeds with replication and weight adjustment based on relevance-feedback. These results were obtained for a threshold of 0.1.

	#Topics	*Relevant*	*Retrieved*	*Rel_Retrieved*	*P@10*
Testbed 0	10	1064	2000	253	0.3000
Testbed 1	14	1302	2650	527	0.2786
Testbed 2	19	1564	3500	476	0.3263
Testbed 3	21	1940	3900	598	0.3190
Testbed 4	18	1385	3600	616	0.2889
Testbed 5	19	1430	3650	663	0.2842
Testbed 6	21	1557	4200	680	0.2333
Average	17.43	1463.14	3357.14	544.71	0.2900

Comparing this to the results of Table 6, showing the retrieval effectiveness when only replication has been used, we notice that the introduction of relevance-feedback (the use of better aligned vectors to the topics) helps routing more topics than when we just used replication. In actual numbers, the average number of topics effectively routed when only replication was used is 1.86, while when both relevance-feedback and replication is used, for the same replication testbeds, the corresponding figure is 17.43. On the other hand the overall effectiveness, as measured by P@10, falls by about 11%. Because the gain in the number of topics that get routed is disproportionate to the loss of retrieval effectiveness we conclude that the use of both replication and relevance-feedback would probably benefit most P2P IR applications; however, this would still depend on the application requirements, with some applications preferring more precise results over wider query penetration.

Comparison. In Table 10 we present a comparison in retrieval performance across all variations of the minimal testbed. From these results we conclude that the term weighting adjustment, that could be accomplished by relevance feedback, is the most effective means to overcome the loss of information caused by clustering. We derive this conclusion after observing that, even though retrieval effectiveness does not fall – it actually increases – more topics are routed to the network. Even though not experimentally verified we anticipate that if network clusters were to change according to the new term-weights, retrieval effectiveness and query penetration would increase even more.

Replication significantly improves the effectiveness for the few topics that are popular, even though it impedes penetration. From our experiments it appears that replication and weight-adjustment complement each-other and so they could yield meaningful results if used together. However, since we only use the standard 100 TREC topics, the popular topics, used for replication, end up being very few and so we will not be expanding on it any further.

Table 10. Comparison in effectiveness across all variations of the minimal ASISWOR testbed

	Basic	*Relevance-Feedback*	*Replication*	*Both*
#Topics	16	**32**	1.86	17.43
P@10	0.2562	0.2844	**0.4071**	0.2900

4.4 Adjusting Term-Weights on Large Testbeds

So far, we have demonstrated two main points. First, that the use of clustering for large-scale P2P IR, at least on testbeds that have similar properties to ours, proves to be ineffective due to the loss of information inherent to the creation of cluster centroids. The second point is that two effective ways to amend this problem is by either introducing (or by using existing) replication and/or introducing some relevance-feedback mechanism that would help overcome the noise in the network resource descriptions. The second point has still to be demonstrated in a larger evaluation environment than the small ASISWOR-based testbed that we have used so far in this chapter.

In this section we present the retrieval effectiveness achieved in the original testbeds when weight-adjustment is used. These results can be seen in Table 11 and they demonstrate that even though the overall effectiveness across all testbeds does not rise beyond approximately 5% (for the case of DLWOR), an important difference in favour of the use of term-weighted resource description emerges, namely that the query penetration almost doubles across all the testbeds. This effect becomes more significant as the number of the topics that get routed rises alongside the retrieval performance.

The results of Table 11 confirm the results derived from earlier experiments using the minimal ASISWOR testbed of Section 4.3. The use of weighted vectors as resource descriptions, as opposed to using the original term-frequency vectors, appear to increase retrieval performance while it greatly enhances the query penetration of the network.

Table 11. Retrieval effectiveness in the original P2P IR testbeds in both the basic and the weight-adjustment configurations. The routing threshold is set to 0.1 while the adjusted vectors were only used for routing and not for peer-clustering.

Testbed	*Configuration*	*#Topics*	*Relevant*	*Retrieved*	*Rel. Ret*	*P@10*
ASISWOR	Basic	22	1050	3184	27	0.0000
	RelFbk	40	3497	7358	72	0.0175
UWOR	Basic	28	2064	5050	49	0.0250
	RelFbk	36	2907	6750	91	0.0361
DLWOR	Basic	17	952	2700	9	0.0059
	RelFbk	**44**	3595	7800	160	**0.0591**

While the effectiveness in both the minimal testbed we used in this section as well as in the large testbeds of Section 3 is by far worse than in a centralised index alternative, the findings of this Chapter are important for future studies and systems as they provide solid experimental evidence suggesting that relevance-feedback is a promising and natural evolution of current P2P IR technologies. Especially the automatic topological adaptation of a P2P network based on feedback seems to be promising as far as retrieval effectiveness is concerned.

5 Conclusions and Future Work

In this paper we presented a full-scale evaluation of a cluster-based P2P IR architecture, focusing on retrieval effectiveness. The architecture [2] we considered uses a two-level clustering in order to organise the shared content of the participating peers, taking no assumptions on the actual document distributions or other properties of the overall shared content. For our experiments we used a set of testbeds [6], which are based on TREC's WT10g. The use of a number of testbeds offers a more holistic view on the behaviour of the architecture we evaluate.

Our findings are the following: Employing a two-level clustering for P2P IR, especially in an open information-sharing environment, seems to amplify issues having to do with clustering itself, therefore resulting in poor retrieval performance. In particular, the noise in the resource descriptions created through clustering impedes standard IR practices such as query-routing based on cosine similarity. Building on this conclusion, we proposed replication and relevance-feedback as potential solutions for this problem and showed experimentally, using a small and manageable testbed, that both mechanisms can improve retrieval effectiveness for cluster-based P2P IR. Finally, we replicated our findings on the large testbeds we used originally. The results show that the performance of the architecture gets improved, mainly, through the significant increase of its query penetration rate.

Obvious pointers for future work include distributed relevance-feedback algorithms, devising replication strategies targeted at IR performance as well as studying the adaptability of the network given real-time changes of resource descriptions from an efficiency viewpoint.

References

1. Lu, J., Callan, J.: Content-based retrieval in hybrid peer-to-peer networks. In: Proceedings of the twelfth international conference on Information and knowledge management, pp. 199–206. ACM Press, New York (2003)
2. Klampanos, I.A., Jose, J.M.: An architecture for information retrieval over semi-collaborating peer-to-peer networks. In: Proceedings of the 2004 ACM Symposium on Applied Computing, vol. 2, pp. 1078–1083. Nicosia, Cyprus (2004)
3. Tang, C., Xu, Z., Mahalingam, M.: Peersearch: Efficient information retrieval in peer-peer networks. Technical Report HPL-2002-198, Hewlett-Packard Labs (2002)
4. Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: ICS, New York (2002)
5. Nottelmann, H., Fuhr, N.: Comparing different architectures for query routing in peer-to-peer networks. In: Proceedings of the 28th European Conference on Information Retrieval Research (ECIR 2006) (2006)
6. Klampanos, I.A., Poznański, V., Jose, J.M., Dickman, P.: A suite of testbeds for the realistic evaluation of peer-to-peer information retrieval systems. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 38–51. Springer, Heidelberg (2005)
7. Tombros, A.: The Effectiveness of Hierarchic Query-Based Clustering of Documents for Information Retrieval. PhD thesis, Department of Computing Science, University of Glasgow (2002)
8. van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworths, London (1979)
9. Plaxton, C.G., Rajaraman, R., Richa, A.W.: Accessing nearby copies of replicated objects in a distributed environment. In: Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures, pp. 311–320. ACM Press, New York (1997)
10. Cuenca-Acuna, F.M., Martin, R.P., Nguyen, T.D.: Planetp: Using gossiping and random replication to support reliable peer-to-peer content search and retrieval. Technical Report DCS-TR-494, Department of Computer Science, Rutgers University (2002)