



**UNIVERSITY**  
*of*  
**GLASGOW**

Cacheda, F. and Carneiro, V. and Plachouras, V. and Ounis, I. (2005)  
Performance comparison of clustered and replicated information retrieval  
systems. *Lecture Notes in Computer Science 4425*:pp. 124-135.

<http://eprints.gla.ac.uk/3760/>

# Performance Comparison of Clustered and Replicated Information Retrieval Systems

Fidel Cacheda<sup>1</sup>, Victor Carneiro<sup>1</sup>, Vassilis Plachouras<sup>2</sup>, and Iadh Ounis<sup>3</sup>

<sup>1</sup> Department of Information and Communication Technologies, University of A Coruña  
Facultad de Informática, Campus de Elviña s/n, 15071 A Coruña, Spain  
{fidel, viccar}@udc.es

<sup>2</sup> Yahoo! Research  
Ocata 1, 1st floor, 08003 Barcelona, Spain  
vassilis@yahoo-inc.com

<sup>3</sup> Department of Computing Science, University of Glasgow  
Glasgow, G12 8QQ, UK  
ounis@dcs.gla.ac.uk

**Abstract.** The amount of information available over the Internet is increasing daily as well as the importance and magnitude of Web search engines. Systems based on a single centralised index present several problems (such as lack of scalability), which lead to the use of distributed information retrieval systems to effectively search for and locate the required information. A distributed retrieval system can be clustered and/or replicated. In this paper, using simulations, we present a detailed performance analysis, both in terms of throughput and response time, of a clustered system compared to a replicated system. In addition, we consider the effect of changes in the query topics over time. We show that the performance obtained for a clustered system does not improve the performance obtained by the best replicated system. Indeed, the main advantage of a clustered system is the reduction of network traffic. However, the use of a switched network eliminates the bottleneck in the network, markedly improving the performance of the replicated systems. Moreover, we illustrate the negative performance effect of the changes over time in the query topics when a distributed clustered system is used. On the contrary, the performance of a distributed replicated system is query independent.

**Keywords:** distributed information retrieval, performance, simulation.

## 1 Introduction

The information available over the Internet has increased spectacularly in the last years, and we can expect that it will continue growing at the same rate, at least in the short term. Simultaneously, Web search engines have grown in importance as the users need to find, recover, and filter all the information available in this environment. Therefore, Web search engines must manage a large amount of information, and make it possible for users to locate the information required in a very short time, while simultaneously dealing with a large number of queries.

Information Retrieval (IR) systems based on a single centralised index present several problems, such as the lack of scalability, or server overloading and failures [11],

which make them unsuitable for highly loaded systems, such as Web search engines. For this reason, the methods based on the distribution of the documents index for searching and storage are widely used. For example, the Google web search service is based on a distributed and replicated IR architecture [1].

A distributed IR system is made up of two components: the brokers (dispatchers or receptionists) and the query servers. The brokers receive the queries from the users, distribute them to the query servers, and send back the final results to the user. The query servers hold the distributed index, process the queries, and send their partial results back to the brokers for the final merging.

An inverted index can be distributed over a collection of servers following two main strategies: global inverted files (term partitioning), or local inverted files (document partitioning). In this work, we will focus on the local inverted file strategy as it has been found to be more efficient than the global inverted file strategy [17] [21].

The index distribution is necessary to deal with a high volume of data (probably because it cannot be indexed by a single machine) and to keep the response times low. When the volume of queries handled by the system must be increased (e.g. because multiple users simultaneously send queries to a Web search engine), then the IR system must be parallelised to process multiple queries concurrently. Two main types of systems can be defined to increase the query throughput: replicated and clustered.

A replicated system is composed of one or more distributed IR systems. Each distributed system indexes the whole collection, and all the distributed systems that have been replicated have the same number of query servers. The brokers, in this case, must decide initially which replica will process the query, and then broadcast the query to all the query servers in the replica. The brokers must balance the load through all the replicas to obtain an optimal performance.

A clustered system is divided into groups of computers (or clusters), where each group operates as an autonomous distributed IR system. Each cluster can be composed of a different number of query servers. Each cluster is responsible for one disjoint part of the whole collection of documents, and each cluster could use distribution and replication to store its respective index. In this case, a broker must determine the appropriate cluster for each query and then submit the query to it. A clustered system must be configured a-priori based on the distribution of the queries that the IR system expects to receive. For example, if 40% of the queries submitted to the IR system are related to “*Entertainment*”, we may assign 40% of our resources (e.g. query servers) to the “*Entertainment*” cluster in order to improve its response time and throughput. This implies that a change in the queries distribution may affect the overall system performance. For example, if the number of “*Entertainment*” queries drops to 20%, this cluster may improve its performance, probably at the expense of other clusters.

In this paper, we present a detailed performance analysis, based on simulations, of a clustered system compared to a replicated system. We also study the effect of changes in the query topics over time, based on work by Spink, Jansen, Wolfram and Saracevic [22].

The performance analysis in distributed IR is used to study different configurations and measure different parameters, usually considering a fixed set of resources. Two parameters are usually considered in the performance analysis of a distributed IR system: response time and throughput [19]. In the former, we are interested in the average time to answer a query when the system is idle. In the latter, we are interested

in the maximum query processing rate that the system is able to achieve. This is especially interesting when designing a large-scale IR system (e.g. hundreds of computers) in order to determine the optimal configuration for some fixed benchmarks<sup>1</sup>, and to detect and prevent possible bottlenecks.

Many previous articles have studied different performance parameters of pure distributed IR systems, such as [8], [10], [14] or [21], to name but a few. On the other side, several previous articles examined the effects of different parallelisation techniques in a distributed IR system. Tomasic and Garcia-Molina [23] simulated a small group of servers and studied the effect of multiprogramming on the throughput using various inverted index organisations. Frieder and Siegelmann [9] studied the organisation of the data to improve the performance of parallel IR systems using multiprocessor computers. Lu and McKinley [16] analysed the effects of partial replication to improve the performance in a collection of 1TB. Moffat, Webber, Zobel and Baeza-Yates [18] presented a replication technique for a pipelined term distributed system, which significantly improves the throughput over a basic term distributed system.

In [5] and [6], the authors analysed the performance of a distributed, replicated and clustered system using a simple network model. They identified two main bottlenecks: the brokers and the network. The high load on the brokers was due to the number of local answer sets to be sorted. The network bottleneck was due to the high number of query servers and the continuous data interchange with the brokers, especially in a replicated IR system. The analysis of the clustered systems indicated that the best throughput was achieved when a great number of query servers was used, outperforming a replicated system. However, the clustered systems must be configured a-priori based on the queries distribution that the IR system is expected to receive.

In [4], a more realistic network simulation model is presented, and the authors described some solutions for the main bottlenecks of a distributed IR system. They showed that the use of a switched network reduces the saturation of the interconnection network. They also showed that the brokers' bottleneck can be improved by reducing the number of partial results sent by the query servers (with a negligible probability of changing the system's precision and recall retrieval performances), or by using a hierarchical distributed broker model.

The main objective of this paper is to compare the performance of a replicated and clustered IR system, both in terms of throughput and response time, using the extended simulation model introduced in [4], and to compare the obtained results with those previously reported in [5] and [6].

The paper is organised as follows. The simulation model is described in Section 2. Section 3 describes the simulations performed for the clustered and replicated systems and the results obtained. A discussion of the results obtained is presented in Section 4. The main conclusions of the work and possible future research directions are presented in Section 5.

## 2 Simulation Model

The simulation model of a distributed IR system used in this work is based on the work described in [4], where the authors implemented a discrete event-oriented

---

<sup>1</sup> An example of fixed benchmarks is that the maximum response time should be one second per query and the minimum throughput should be twenty queries per second.

simulator using the JavaSim simulation environment [15]. The defined simulation model represents a local inverted file strategy (see Section 1). All the queries are stored in a global queue, which is controlled by one or more central brokers. Each broker will take one query and will send it to all the query servers through a network [21]. Each query server then processes the whole query locally, obtains the answer set for that query, ranks the documents, selects a certain number of documents from the top of the ranking and returns them to the broker. The broker collects all the local answer sets and combines them into a global and final ranked set of documents.

From the brokers' point of view, the time to process the  $i$ th query (named  $t_i$ ) is divided into three phases: the time to process the query in the query servers ( $P_1$ ), the time to receive all the partial answers from the query servers to the broker ( $P_2$ ) and the merging and ranking of the final results by the broker ( $P_3$ ). Therefore, the processing time for a query  $q_i$  is given by:

$$t_i = \overbrace{\max(t_{i,j})}^{P_1} + \overbrace{\max(ra_{i,j})}^{P_2} + \overbrace{\sum_j tc(tr_{i,j})}^{P_3}$$

where the following parameters are introduced:

- $q_i$ : vector of keywords for the  $i$ th query.
- $t_{i,j}$ : total time (in milliseconds) to complete the processing of query  $q_i$  at query server  $j$ .
- $ra_{i,j}$ : time to receive the local answer set for query  $q_i$  from the query server  $j$ .
- $tr_{i,j}$ : number of documents from the top ranking in query  $q_i$  returned as the local answer set for query server  $j$ , where  $tr_{i,j} \leq tr_{max}$ , and  $tr_{max}$  is the maximum number of top ranked documents in the local answer (we consider the top 1000 documents only).
- $tc(n)$ : time to merge and sort  $n$  documents, which is computed following the logarithmic model:  $tc(n) = tc_0 + tc_1 \times n + tc_2 \times \ln(n)$ , as described in [4].

From the query servers' point of view, the time to process the  $i$ th query by the  $j$ th query server (named  $t_{i,j}$ ) is divided into five phases: the time to receive the query from the broker ( $P_{1,1}$ ), the initialisation time ( $P_{1,2}$ ), the seek time ( $P_{1,3}$ ), the reading time from disk ( $P_{1,4}$ ), and the ranking of the partial results ( $P_{1,5}$ ).

$$t_{i,j} = \overbrace{rq_{i,j}}^{P_{1,1}} + \overbrace{ti}^{P_{1,2}} + \overbrace{k_i \times ts}^{P_{1,3}} + \overbrace{\sum_{k \in q_i} d_{k,j} \times tr}^{P_{1,4}} + \overbrace{tc(r_{i,j})}^{P_{1,5}}$$

where these new parameters are introduced:

- $rq_{i,j}$ : time to receive the query  $q_i$  for the query server  $j$ .
- $ti$ : initialisation time, including memory allocation and output display, if necessary.
- $k_i$ : number of keywords in query  $q_i$ .
- $ts$ : average seek time for a single disk.
- $tr$ : average time to read the information about one document in an inverted list and to do its processing (seek time is excluded).
- $d_{k,j}$ : number of documents of the inverted list for keyword  $k$  on query server  $j$ .
- $r_{i,j}$ : number of results obtained for query  $q_i$  on query server  $j$ .

The Terrier<sup>2</sup> IR system described in [20] is used to estimate the parameters for the analytical model, obtaining the following values:  $t_i = 62.335\text{ms}$ ,  $t_s = 0.03\text{ms}$ ,  $t_r = 1.15\mu\text{s}$ ,  $t_{c_0} = -470$ ,  $t_{c_1} = 0.0$ ,  $t_{c_2} = 62$  [4]. The document model parameters ( $d_{k,j}$  and  $r_{i,j}$ ) are simulated from the SPIRIT collection, which consists of 94,552,870 documents and 1 terabyte (TB) of text [13]. Each query is generated as a sequence of  $K$  terms ( $t_1, \dots, t_k$ ), independently and identically distributed, following the skewed query model [12]. The skewed query model sets the probability of a term occurring in a query proportional to its frequency in the vocabulary, and provides more realistic queries than the uniform query model [6].

The network parameters ( $rq_{i,j}$  and  $ra_{i,j}$ ) that determine the transmission times among the hosts cannot be estimated using an analytical model, as they depend directly on the network load of each moment. Therefore, a network simulation model is defined.

In [5] and [6], the network simulation model was based on a shared access local area network (LAN), where the transmission media is shared out among all the hosts, which must compete to access the media and send their transmissions. This network simulation model had certain limitations (e.g. not considering the maximum number of hosts connected to the LAN or the maximum size of the network) that reduced the capacities of the simulated IR systems.

With the aim of improving the limitations of this initial network model, a new model was defined in [4], equivalent to a switched network FastEthernet 100BASE-T at 100Mbps. The switched LAN is the evolution of the shared access networking technology and it is based on a device named switch, which centralises the communication among the hosts. In this way, the switch will reduce the transmission conflicts, because a host only has to compete with other hosts that want to communicate with the same destination, increasing the effective network speed.

Using this new network model, a more extensive and realistic simulation model is defined, where the hosts are interconnected via one or more switches, depending on the number of hosts to be interconnected (assuming that each switch has a capacity for 64 hosts). Moreover, the overhead estimation is carried out exhaustively, taking into account the different headers of the communication protocols, IP fragmentation, and even the propagation delay [4]. The design of this new network model has also extended the capacity to represent multicast messages. The multicast messages allow sending one message to multiple recipients, instead of sending one message to each recipient (unicast messages). In a distributed IR system based on local inverted files, multicast messages are especially useful to reduce the number of messages required to distribute the queries to the query servers from the brokers.

In [4], an extended description of the switched network simulation model can be found, along with a detailed comparison of the real IR system with the simulation model, confirming their correspondence. A brief description of the network simulation model can also be found in the short article [3]. In all the experiments reported in this paper, this new switched network simulation model is used in order to obtain realistic conclusions when simulating and comparing the clustered and replicated systems.

---

<sup>2</sup> A core version of the Terrier system can be downloaded from <http://ir.dcs.gla.ac.uk/terrier>

### 3 Experiments

The objective of the experiments in this paper is to compare the performance of a replicated and clustered IR system, both in terms of throughput and response time, using a realistic setting based on the switched network simulation model described in the previous section. In [5] and [6], the main conclusions showed that a clustered system will outperform a replicated system if a notable number of query servers is used (e.g. 1024). These experiments, however, were based on a shared access network, which produced the saturation of the network in the replicated system. Moreover, only four replicas were defined in the considered replicated system.

In the new experiments conducted in this paper, we provide a detailed comparison between a replicated system and a clustered system using 1024 query servers and a switched network supporting multicast. In addition, we consider replicated systems with up to 32 replicas.

#### 3.1 Experimental Setting

For the replicated system we examine different configurations for the 1024 query servers: 1, 2, 4, 8, 16 and 32 replicas (with 1024, 512, 256, 128, 64 and 32 query servers per replica, respectively). The optimal number of brokers required for the replicated system is calculated as  $3R$ , where  $R$  is the number of replicas, as shown in [4]. The optimal number of brokers is the minimum number of brokers necessary to obtain the best throughput and response time performance (there are no improvements by further increasing this number).

As mentioned in Section 1, a clustered system must be configured a-priori based on the distribution of queries that the IR system is likely to receive. For the configuration of the clustered system in the experiments reported in this section, we used the work by Spink et al. [22], where a set of real Web queries is categorised into 11 different topics considering three different years: 1997, 1999 and 2001. Table 1 provides a summary of the 11 topics and the percentage of queries through the different years.

We assume that each topic is indexed in a different cluster. The SPIRIT collection [13] is divided into 11 sub-collections with an inverted file of approximately the same size that is 8.5 million documents in each sub-collection. Therefore, the 11 defined clusters will index the same number of documents, although using a different number of servers. This setting is selected because we are more interested in the configuration of each cluster, rather than the distribution of the topics. Hence, the configurations of the clusters will fit the same throughput curve, generating a simpler simulation model.

In the reported simulations, the number of queries is fixed to 200 and the queries will retrieve 3 million documents on average. The base sub-collection of 8.5 million documents has been distributed over  $N$  query servers using a switched network and three brokers, where  $N = 1, 2, 4, 8, 16, 32, 64, 128, 256$  and  $512$ . In Table 1, the column *Configuration* describes the query servers assigned to each topic. The first number represents the number of the distributed query servers, and the second, the number of replicas in each cluster.

**Table 1.** Distribution of queries across general topic categories, and the used configurations for the simulated clustered system. The column Configuration describes the query servers assigned to each topic. The first number represents the number of the distributed query servers, and the second one represents the number of replicas in each cluster.

Topics	1997	1999	2001	Configuration
<i>Entertainment</i>	19.64 %	7.73 %	6.65 %	67 * 3
<i>Pornography</i>	16.54 %	7.73 %	8.56 %	56 * 3
<i>Commerce</i>	13.03 %	24.73 %	24.76 %	66 * 2
<i>Computers</i>	12.24 %	11.13 %	9.65 %	63 * 2
<i>Sciences</i>	9.24 %	8.02 %	7.55 %	48 * 2
<i>People</i>	6.43 %	20.53 %	19.75 %	66 * 1
<i>Society</i>	5.44 %	4.43 %	3.96 %	56 * 1
<i>Education</i>	5.33 %	5.52 %	4.55 %	55 * 1
<i>Arts</i>	5.14 %	1.33 %	1.16 %	53 * 1
<i>Non-English</i>	3.84 %	7.03 %	11.36 %	39 * 1
<i>Government</i>	3.13 %	1.82 %	2.05 %	32 * 1

The clustered system is configured in accordance with the distribution of the topics of the year 1997. The replications for the most popular topics are maximised, but the number of query servers in each replica is kept as close as possible to 64 in order to obtain an appropriate response time. Indeed, in [4], the authors studied the improvement obtained with a switched network and the figures showed that with less than 64 query servers the performance of the system decreases importantly. The number of brokers is selected taking into account the sum of the replicas in each cluster (i.e.  $R=18$  replicas), and calculating the optimal number of brokers as  $3R$ , as described in [4]. For completeness, we also report results with the optimal number of brokers  $2R + 1$ , suggested in [5] and [6].

In all our experiments, as stressed in Section 1, the performance is measured using the throughput and the response time. The throughput is measured considering that the system is operating in batch mode and that there is a processing queue of 200 queries. The response time is measured assuming that the queries will arrive to the IR system following an Exponential distribution [7], with mean 500 milliseconds and simulating 200 queries.

### 3.2 Replicated and Clustered System Comparison

The results obtained for the simulated clustered and replicated systems are presented in Table 2. The column *Replicated* describes the configurations for the studied replicated system. The first number represents the number of replicas, and the second one represents the number of the distributed query servers. In all the replicated system configurations, the optimal number of brokers is defined as  $3R$ . The shaded cells represent the optimal configurations for the replicated and clustered systems. In this case, the optimal configuration is the one that achieves the best trade-off between the minimal response time and the maximal throughput.



Regarding the replicated systems, we observe that the throughput increases as the number of replicas increases due to the higher level of parallelism in the system. At the same time, the response time of the system is decreasing as the replication increases, except for the last configuration (32x32), where the reduced distribution of the index in each replication (only 32 query servers) increases the response time to 2658 milliseconds per query (17% more than the optimal configuration).

On the other hand, Table 2 shows that the throughput of the clustered system is maximised if 3*R* brokers are used. This configuration outperforms the 2*R* + 1 configuration in terms of throughput, extending the conclusions obtained in [4] for the replicated system to the clustered system. The number of brokers does not have an important repercussion on the response time as the queries are processed nearly sequentially, which leads to minimal parallelisation. In a clustered system, the number of replicas (*R*) is calculated as the sum of all the replicas through all the clusters (e.g. 18 replicas, obtained from the configuration in Table 1).

Comparing the two types of systems, the results show that a replicated system with 16 replicas will achieve a better throughput and response time than the clustered system defined. In both cases the level of parallelism achieved is quite similar (with 16 and 18 parallel groups, respectively). The main benefit achieved with the clustered system is a reduction in the network traffic, which is crucial if the network is the main bottleneck of the system. However, the switched network has solved this problem improving markedly the performance of the replicated system, which is able to outperform the clustered system.

This result suggests that the distribution must be used to reduce the response times and the replication must be used to increase the query throughput of the system. For example, this is the approach used by the Google web search service [1].

The main conclusion of the above set of experiments is that the performance of a clustered system (both in throughput and response time) does not improve the performance obtained by the best replicated system. This result is related to the use of a switched network. The switched network has eliminated the bottleneck in the network, markedly improving the performance of the replicated systems. On the other hand, the main advantage of a clustered system is the reduction of network traffic, which is less relevant when a switched network model is used.

**Table 2.** Throughput (queries/second) and response time (milliseconds) for the clustered and replicated systems (with the optimal number of brokers as 3*R*), using a switched network supporting multicast

<i>Replicated</i>	Response		<i>Clustered</i>		Response	
	Throughput	Time	<i>Brokers</i>	<i>Year</i>	Throughput	Time
<i>1x1024</i>	0.70	4247.83	<i>3R</i>	<i>1997</i>	7.60	2404.11
<i>2x512</i>	1.38	4257.67	<i>3R</i>	<i>1999</i>	3.23	2828.11
<i>4x256</i>	2.69	3231.22	<i>3R</i>	<i>2001</i>	3.59	2960.87
<i>8x128</i>	5.03	2354.92	<i>2R+1</i>	<i>1997</i>	7.17	2380.20
<i>16x64</i>	8.47	2274.09	<i>2R+1</i>	<i>1999</i>	3.11	3165.59
<i>32x32</i>	12.92	2658.93	<i>2R+1</i>	<i>2001</i>	3.43	2863.65

### 3.3 Query Topics Change

As we described in the introduction, a clustered system must be configured a-priori based on the distribution of the queries that the IR system expects to receive. In our experiments above, the clustered system was configured based on the queries distribution for the year 1997 (Table 1, second column).

In this section, we study the effect of changes in the topics distribution over time in the performance of clustered systems. Obviously, the performance of a replicated system is query independent. Therefore, the performance values obtained for a replicated system do not change: 8.47 queries per second and 2.27 seconds per query on average, for the optimal configuration (See Table 2, row *16x64*). However, the performance of a clustered system when the queries distribution varies can be severely affected, as we will discuss below.

Indeed, in the experiments reported in Table 2, we also used the query distribution for the years 1999 and 2001 (Table 1, third and forth column respectively) to simulate the queries in the clustered system. The results (see Table 2) show more than 50% throughput reduction in the *3R* configuration (from 7.60 queries per second to 3.23 and 3.59 queries per second for the years 1999 and 2001, respectively) and in the *2R + I* configuration (from 7.17 queries per second to 3.11 and 3.43 queries per second for the years 1999 and 2001, respectively), and an increase in the response time between 17% and 33%, in both configurations, for the years 1999 and 2001.

The negative effect on the response time is less marked, because each cluster has been configured with enough query servers per replica to obtain an appropriate response time. On the other hand, the changes in the popular topics imply that smaller clusters will receive more queries, decreasing their throughput, while larger clusters will receive fewer queries and may have idle periods, decreasing the overall throughput of the system.

In [5] and [6], the throughput worsening due to the changes in the topics distribution was also confirmed, although the distribution for the year 2001 was the baseline and the reduction in the performance was mainly manifested in the year 1997, with nearly no changes with the year 1999.

The important repercussion of the changes in the queries distribution in the performance of the clustered systems entails a permanent monitoring of the queries submitted to the IR system and, if necessary, the modification of the clusters configuration. This point raises new research concerns, such as the estimation of the threshold in the topics change where it is worth modifying the configuration, looking for a balance between the performance improvement and the cost of modifying the cluster configuration of the IR system.

## 4 Discussion

We have shown that the performance obtained (both in terms of throughput and response time) for a clustered system does not improve over the one obtained by the best replicated system.

This result, which in some way contradicts the conclusions in [5] and [6], is related to the use of a switched network. The main benefit of a clustered system is the reduction of network traffic. However, the switched network has eliminated the bottleneck

in the network, markedly improving the performance of the replicated systems, which exceeds that of the clustered systems. It is interesting to state that the clustered system results represent a best-case performance as we have assumed that documents can be split in non-overlapping clusters and queries can be assigned univocally to one cluster.

Moreover, we have illustrated the negative performance effect of the changes over time in the topics distribution, when a clustered system is used, as opposed to the performance of a replicated system, which is query independent. As a consequence, the configuration of a clustered system needs to be modified according to the topics distribution and their drift over time. This may prove to be a practical problem in the Web context, where the information needs of the users and their interest may markedly vary over time, for example, in reaction to contemporary events and concerns [2].

When building a large-scale IR system, the index distribution is necessary to deal with a high volume of data (and to keep the response times low) and the parallelisation is necessary to process multiple queries concurrently. The results obtained in this work suggest that the replicated systems are the best option for the parallelisation in terms of performance (throughput and response time) and stability through the time (as it is query independent).

Finally, it is important to mention that there could be other factors not represented in these simulations that could improve the performance of the clustered systems. For example, one benefit of the clustered systems is the reduction in the size of the collection indexed by each cluster. This may produce a reduction in the number of relevant results obtained by the IR system, since the final results must be associated with the relevant cluster for the query. In all the experiments reported in this paper, both types of systems retrieve exactly the same number of relevant documents for each query. Nevertheless, in a real clustered system, some documents associated with other less relevant clusters will not be included and this could reduce the overall response time. It is however difficult to precisely model this as it depends on factors such as the size of the cluster, the length of the query and its type. In addition, the number of returned documents may affect precision and recall. As a consequence, in comparing the clustered and replicated systems, we simulated systems that return the same number of documents making the comparison fair in terms of precision and recall measures.

## 5 Conclusions

In this work, we have presented a detailed study of a clustered system and several replicated systems, comparing their performance in terms of throughput and response time. Using the extended simulation network model introduced in [4] to represent a switched network, we can perform a more accurate and realistic evaluation of the two types of distributed IR systems.

We showed that the throughput and response time for a clustered system do not improve the values obtained by the best replicated system. Moreover, the performance of a replicated system is independent of the changes over time in the query topics whilst there is a negative effect on the clustered system performance. This implies that the configuration of a clustered system needs to be adapted dynamically to the topics distribution. A possible future work would be to investigate how to automatically define the optimal configuration of a clustered system over time.

It is also important to consider that there are other factors that have not been taken into account, such as the reduction in the number of relevant documents, which may improve the performance of clustered systems. Another future work would be to study this effect on the response time and its repercussion in terms of precision and recall.

This work suggests that the replicated IR systems should be used to obtain a better degree of parallelism and throughput. In this sense, in our future work, we would like to study different solutions to reduce data interchange through the interconnection network and the workload on the brokers for the replicated and distributed IR systems.

## Acknowledgements

The work of the first and second authors has been partially supported by the Spanish government under project TS12005-07730.

The work of the third and fourth authors is funded by a UK Engineering and Physical Sciences Research Council (EPSRC) project grant, number GR/R90543/01. The project funds the development of the Terrier Information Retrieval framework (url: <http://ir.dcs.gla.ac.uk/terrier>).

We would also like to thank Mark Sanderson and Hideo Joho for giving us access to the 1TB dataset used for the SPIRIT Project.

## References

1. Barroso, L.A., Dean, J., Hölzle, U.: Web search for a planet: The Google cluster architecture. *IEEE Micro*, 23(2), (2003) 22-28.
2. Beitzel, S.M., Jensen, E.C., Chowdhury, A., Grossman, D., Frieder, O.: Hourly Analysis of a Very Large Topically Categorized Web Query Log. In Proc. of the 27<sup>th</sup> Conf. on Research and Development in Information Retrieval. New York: ACM Press. (2004) 321-328
3. Cacheda, F., Carneiro, V., Plachouras, V., Ounis, I.: Performance Network Analysis for Distributed Information Retrieval Architectures. In Proc. of 27<sup>th</sup> European Conf. on Information Retrieval Research (ECIR'05), LNCS Vol. 3408, (2005) 527-529.
4. Cacheda, F., Carneiro, V., Plachouras, V., Ounis, I.: Performance Network Analysis for Distributed Information Retrieval Architectures. *Information Processing and Management Journal*, published on-line (2006)
5. Cacheda, F., Plachouras, V., Ounis, I.: Performance Analysis of Distributed Architectures to Index One Terabyte of Text. In Proc. of 26<sup>th</sup> European Conf. on Information Retrieval Research, LNCS Vol. 2997, (2004) 394-408.
6. Cacheda, F., Plachouras, V., Ounis, I.: A Case Study of Distributed Information Retrieval Architectures to Index One Terabyte of Text. *Information Processing and Management Journal*, 41(5), (2005) 1141-1161
7. Cacheda, F., Viña, A.: Experiences retrieving information in the World Wide Web. In Proc. of the 6th IEEE Symposium on Computers and Communications. IEEE Computer Society, (2001) 72-79
8. Cahoon, B., McKinley, K.S.: Performance evaluation of a distributed architecture for information retrieval. In Proc. of 19<sup>th</sup> ACM-SIGIR International Conf. on Research and Development in Information Retrieval, New York: ACM Press. (1996) 110-118.
9. Frieder, O., Siegelmann, H. T.: On the Allocation of Documents in Multiprocessor Information Retrieval Systems. In Proc. of the 14<sup>th</sup> ACM-SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press. (1991) 230-239

10. Hawking, D.: Scalable text retrieval for large digital libraries. *Lecture Notes in Computer Science*, Vol. 1324. (1997) 127-146
11. Hawking, D., Thistlewaite, P.: Methods for Information Server Selection. *ACM Transactions on Information Systems*, Vol. 17(1). (1999) 40-76
12. Jeong, B., Omiecinski, E.: Inverted File Partitioning Schemes in Multiple Disk Systems. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 6(2). (1995) 142-153
13. Jones, C. B., Purves, R., Ruas, A., Sanderson, M., Sester, M., van Kreveld, M., Weibel, R.: Spatial information retrieval and geographical ontologies an overview of the SPIRIT project. In *Proc. of the 25<sup>th</sup> ACM-SIGIR Conf. on Research and Development in Information Retrieval*. New York: ACM Press. (2002) 387-388
14. Lin, Z., Zhou, S.: Parallelizing I/O intensive applications for a workstation cluster: a case study. *ACM SIGARCH Computer Architecture News*, Vol. 21 (5). (1993) 15-22
15. Little, M. C.: *JavaSim User's Guide. Public Release 0.3, Version 1.0*. University of Newcastle upon Tyne. Retrieved 1 June, 2003. <http://jasim.ncl.ac.uk/manual/jasim.pdf>
16. Lu, Z., McKinley, K.: Partial collection replication versus caching for information retrieval systems. In *Proc. of the 25<sup>th</sup> ACM-SIGIR Conf. on Research and Development in Information Retrieval*. New York: ACM Press. (2000) 248-255
17. Moffat, A., Webber, W., Zobel, J.: Load Balancing for Term-Distributed Parallel Retrieval. In *Proc. of the 29<sup>th</sup> ACM-SIGIR Conf. on Research and Development in Information Retrieval*. New York: ACM Press. (2006) 348-355
18. Moffat, A., Webber, W., Zobel, J., Baeza-Yates, R.: A pipelined architecture for distributed text query evaluation. *Information Retrieval*, published on-line. (2006)
19. Moffat, A., Zobel, J.: What does it mean to "measure performance"? In *Proc. of the 5<sup>th</sup> International Conf. on Web Information Systems*, LNCS Vol. 3306. (2004) 1-12
20. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Lioma, C.: Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proc. of ACM SIGIR'06 Workshop on Open Source Information Retrieval*. (2006)
21. Ribeiro-Neto, B., Barbosa, R.: Query performance for tightly coupled distributed digital libraries. *Proc. 3<sup>rd</sup> ACM Conf. on Digital Libraries*. New York: ACM Press. (1998) 182-190
22. Spink, A., Jansen, B. J., Wolfram, D., Saracevic, T.: From e-sex to e-commerce: Web search changes. *IEEE Computer* Vol. 35(3). (1998) 107-109
23. Tomasic, A., Garcia-Molina, H.: Performance of inverted indices in shared-nothing distributed text document information retrieval systems. In *Proc. 2<sup>nd</sup> Inter. Conf. on Parallel and Distributed Info. Systems*. San Diego, California: IEEE Computer Society. (1993) 8-17