



A case study of distributed information retrieval architectures to index one terabyte of text

Fidel Cacheda ^a  , Vassilis Plachouras ^b  , Iadh Ounis ^b  

[Show more](#) 

 Share  Cite

<https://doi.org/10.1016/j.ipm.2004.05.002> 

[Get rights and content](#) 

Abstract

The increasing number of documents to be indexed in many environments (Web, intranets, digital libraries) and the limitations of a single centralised index (lack of scalability, server overloading and failures), lead to the use of distributed information retrieval systems to efficiently search and locate the desired information. This work is a case study of different architectures for a distributed information retrieval system, in order to provide a guide to approximate the optimal architecture with a specific set of resources. We analyse the effectiveness of a distributed, replicated and clustered architecture simulating a variable number of workstations (from 1 up to 4096). A collection of approximately 94 million documents and 1 terabyte (TB) of text is used to test the performance of the different architectures. In a purely distributed information retrieval system, the brokers become the bottleneck due to the high number of local answer sets to be sorted. In a replicated system, the network is the bottleneck due to the high number of query servers and the continuous data interchange with the brokers. Finally, we demonstrate that a clustered system will outperform a replicated system if a high number of query servers is used, essentially due to the reduction of the network load. However a change in the distribution of the users' queries could reduce the performance of a clustered system.

Introduction

The number of documents available on the Web is increasing daily, from several thousands in 1993 to more than 4 billion indexed by Google (2004) in 2004. There are several challenges faced by Web Information Retrieval (IR) systems in this highly dynamic environment. One major challenge is how to efficiently locate the desired information into all available data. Recent research in Web IR has led to the development of

highly effective search engines that allow users to locate relevant or useful documents. A second challenge at the system level is how to design retrieval engines that can process a massive number of documents, while handling a considerable number of queries simultaneously.

Retrieval systems based on a single centralised index are subject to several limitations: lack of scalability, server overloading and failures (Hawking & Thistlewaite, 1999). Therefore, given these facts, it seems more appropriate to turn to the Distributed Information Retrieval (DIR) approach for the storage and search processing.

In a distributed search environment, there are usually two basic strategies for distributing the inverted index over a collection of query servers (Ribeiro-Neto & Barbosa, 1998; Tomasic & Garcia-Molina, 1993). One strategy is to partition the document collection so that each query server is responsible for a disjoint subset of documents in the collection (called *local inverted files* in Ribeiro-Neto & Barbosa (1998)). The other option is to partition based on the index terms so that each query server stores inverted lists corresponding to only a subset of the index terms in the collection (called *global inverted files* in Ribeiro-Neto & Barbosa (1998)). In the local inverted file organisation, a search term will be broadcasted to all the query servers, each of which would return a disjoint list of relevant documents. In the global inverted file organisation, only some query servers will receive some query terms, each of which would return a list of relevant documents for each term. Performance studies described in (Tomasic & Garcia-Molina, 1993) indicate that the local inverted file organisation uses system resources effectively, provides good query throughput in most cases and is more resilient to failures.

From the collection point of view, a distributed information retrieval system could follow a single-collection model or a multi-collection model (Callan, 2000). In the single-collection model, the collection of documents is indexed and distributed as a whole. In a multi-collection model, the whole collection of documents is divided into sub-collections that are indexed independently, where each sub-collection could have the inverted index distributed. A central server stores the descriptions of each sub-collection, and a sub-collection selection service will identify the most relevant sub-collection(s) for each query. The query is sent to the most relevant sub-collections and the results are merged considering that different rankings schemes could have been used.

This work is a case study of different architectures for a distributed information retrieval system. The SPIRIT collection (94,552,870 documents and 1 terabyte (TB) of text) (Jones et al., 2002) is used for the simulation of the distributed IR system. We partition the collection of documents using a local inverted file strategy, and we test the response times for different configurations. Although the obtained timings depend on the specific system simulated (Plachouras, Ounis, Amati, & van Rijsbergen, 2002), the trends and conclusions should be independent of the system used. In this way, we intend to provide a guide to estimate the optimal architecture of a distributed IR system to index and search a very large collection like SPIRIT.

The improvements in the performance of a single-collection model are examined in a distributed and replicated system. Moreover, the effects of a multi-collection model are tested through a clustered system. Using the query topics provided in (Spink, Jansen, Wolfram, & Saracevic, 2002), we define a clustered system and compare its performance with a replicated system in order to measure the effect of changes in the query topics through time.

We start by presenting the related work in Section 2. In Section 3, we describe our simulation model, starting with the analytical model for a basic, not distributed, IR system. Next, a generic collection model is described and the key components of a distributed IR system are analysed. In Section 4, we describe the simulations performed for the different architectures: distributed, replicated and clustered system, and the results obtained. Finally, the main conclusions and the future work are presented in Section 5.

Section snippets

Related work

Related work for distributed IR systems includes the evaluation of the architecture performance, data partitioning, caching and multiprocessor systems. However, the work on architecture performance is the most directly related to this paper.

Harman, McCoy, Toense, and Candela (1997) showed the feasibility of a distributed IR system, by developing a prototype architecture and performing user testing. However, they used a very small text collection (less than 1 GB data), and did not analyse ...

Simulation model

To explore the performance of different architectures for a distributed IR system, we implemented a discrete event-oriented simulator using the JavaSim simulation environment (Little, 2001).

The simulation model defined in this work is divided into three parts. Initially an analytical model has been developed for the simulation of a simple IR system based on the WT10g collection and the queries from the topic-relevance task of TREC10 (Hawking & Craswell, 2001), using a single server. Next, a ...

Simulation results

This section describes the results of several experiments, in which we used the simulation model described in the previous section. The objective is to investigate different approaches for the distribution and replication of the collection using a cluster of query servers, and compare the performance between the different configurations.

All the simulations are based on the 1 TB SPIRIT collection model. We model the queries with the skewed query model and, following a worst case scenario, we ...

Conclusions

In this paper, we have described different architectures for a distributed IR system, analysing the optimal design (i.e. number of brokers) and estimating the maximum performance achieved with multiple configurations (from 1 up to 4096 query servers). We have studied the performance of a distributed,

replicated and clustered system, and we have established the bottlenecks and limitations of each possible configuration. Our work extends previous works with respect to both the size of the ...

Acknowledgments

The work of the first author has been partially supported by the Comisión Interministerial de Ciencia y Tecnología (CICYT) of the Spanish government, under project TIC2001-0547 and by the Fundación Caixa Galicia (Beca curso 2002/2003 para Estudios de Postgrado en Universidades y en Centros de Investigación de Excelencia Académica).

The work of the second and third authors is funded by a UK Engineering and Physical Sciences Research Council (EPSRC) project grant, number GR/R90543/01. The ...

[Recommended articles](#)

References (22)

K. Risvik *et al.*

[Search engines and web dynamics](#)

Computer Networks (2002)

P. Bailey *et al.*

Engineering a multi-purpose test collection for web retrieval experiments

Information Processing and Management (2003)

F.J. Burkowski

Retrieval performance of a distributed database utilising a parallel process document server

B. Cahoon *et al.*

Performance evaluation of a distributed architecture for information retrieval

J. Callan

Distributed information retrieval

T.R. Coevreur *et al.*

An analysis of performance and cost factors in searching large text databases using parallel search systems

Journal of the American Society for Information Science (1994)

Google, (2004). Homepage. Google Inc. Retrieved 21 April, 2004. Available:...

D. Harman *et al.*

Prototyping a distributed information retrieval system that uses statistical ranking

Information Processing and Management (1997)

D. Hawking

Scalable text retrieval for large digital libraries

Lecture Notes in Computer Science (1997)

Hawking, D., & Craswell, N., (2001). Overview of the TREC-2001 Web track. In E. M. Voorhees, & D. K. Harman (Ed.),...



View more references

Cited by (29)

MapReduce indexing strategies: Studying scalability and efficiency

2012, Information Processing and Management

[Show abstract](#) ✓

Analyzing imbalance among homogeneous index servers in a web search system

2007, Information Processing and Management

Citation Excerpt :

...As an outcome of our analysis, we verified that the idealized scenario, that supposes balanced execution times as a consequence of an uniform data distribution among homogeneous index servers, is unlikely to be found in practice. This is an important contribution because it sheds light on the usual assumption of balanced execution times taken by many theoretical models in the literature to simplify their modeling task (Cacheda et al., 2004; Chowdhury & Pass, 2003; Ribeiro-Neto & Barbosa, 1998; Tomasic & Garcia-Molina, 1993). Our findings derive from a comprehensive experimental analysis using an information retrieval testbed and real data obtained from a real-world search engine...

[Show abstract](#) ✓

Opinion mining: reviewed from word to document level ↗

2013, Social Network Analysis and Mining

A cascade ranking model for efficient ranked retrieval ↗

2011, SIGIR 11 Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval

An effective statistical approach to blog post opinion retrieval ↗

2008, International Conference on Information and Knowledge Management Proceedings

Digital libraries ↗

2007, Annual Review of Information Science and Technology



[View all citing articles on Scopus ↗](#)

[View full text](#)

Copyright © 2004 Elsevier Ltd. All rights reserved.



All content on this site: Copyright © 2026 Elsevier B.V., its licensors, and contributors. All rights are reserved, including those for text and data mining, AI training, and similar technologies. For all open access content, the relevant licensing terms apply.

