Information Processing and Management 49 (2013) 51–66

Contents lists available at SciVerse ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

# Document replication strategies for geographically distributed web search engines

Enver Kayaaslan [a,1], B. Barla Cambazoglu [b,*], Cevdet Aykanat [a]

[a] Computer Engineering Department, Bilkent University, Ankara, Turkey
[b] Yahoo! Research, Barcelona, Spain

ARTICLE INFO

*Article history:*
Received 16 September 2010
Received in revised form 7 September 2011
Accepted 2 January 2012
Available online 2 February 2012

*Keywords:*
Web search
Distributed information retrieval
Document replication
Query processing
Query forwarding
Result caching

ABSTRACT

Large-scale web search engines are composed of multiple data centers that are geographically distant to each other. Typically, a user query is processed in a data center that is geographically close to the origin of the query, over a replica of the entire web index. Compared to a centralized, single-center search engine, this architecture offers lower query response times as the network latencies between the users and data centers are reduced. However, it does not scale well with increasing index sizes and query traffic volumes because queries are evaluated on the entire web index, which has to be replicated and maintained in all data centers. As a remedy to this scalability problem, we propose a document replication framework in which documents are selectively replicated on data centers based on regional user interests. Within this framework, we propose three different document replication strategies, each optimizing a different objective: reducing the potential search quality loss, the average query response time, or the total query workload of the search system. For all three strategies, we consider two alternative types of capacity constraints on index sizes of data centers. Moreover, we investigate the performance impact of query forwarding and result caching. We evaluate our strategies via detailed simulations, using a large query log and a document collection obtained from the Yahoo! web search engine.



## 1. Introduction

We consider a large-scale web search engine architecture with multiple, geographically distributed data centers (Baeza-Yates, Gionis, Junqueira, Plachouras, & Telloli, 2009; Cambazoglu, Plachouras, & Baeza-Yates, 2009). In this architecture, each data center crawls and maintains the documents that are served by the web sites in its geographical region (Cambazoglu, Plachouras, Junqueira, & Telloli, 2008). User queries are routed to data centers according to the regions they originate from. For example, a data center in Madrid crawls the web sites in Spain and processes the queries submitted from Spain. As we will discuss next, this architecture leads to two extremes for the placement of the web index and query processing.

At one extreme, a global index is built over the entire web collection, and this index is replicated on all data centers. Queries are processed on the entire web index, and hence search result qualities are identical to those of a centralized search architecture. However, this approach does not scale well since the global web index needs to be constructed from a distributed document collection and periodically maintained. Moreover, this approach requires major hardware investments and results in high power consumption, which is an important issue for commercial search engines. Finally, processing queries

* Corresponding author. Address: Yahoo! Research, Avda. Diagonal 177, 8th Floor, 08018 Barcelona, Spain. Tel.: +34 93 183 8830; fax: +34 93 183 8901.
*E-mail addresses:* enver@cs.bilkent.edu.tr (E. Kayaaslan), barla@yahoo-inc.com (B.B. Cambazoglu), aykanat@cs.bilkent.edu.tr (C. Aykanat).
[1] This work is conducted during the author's internship at Yahoo! Research Barcelona.

0306-4573/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2012.01.002

over an entire web index may be too costly to satisfy the tight response time constraints of large-scale web search engines (Cambazoglu, Zaragoza, et al., 2010).

At the other extreme, each data center builds a regional web index on its local crawl and processes its queries over this partial (local) index. This approach is highly scalable because partial indexes are locally maintained and less resources are needed for query processing (alternatively, queries can be processed faster). However, as processing of queries is limited to a partial index, some high-quality or best matching documents that are indexed by non-local data centers may be missing in search results. This may lead to not affordable losses in search result qualities, negatively impacting the user satisfaction and potentially the revenues of the search engine.

A search engine architecture based on selective replication of documents on data centers emerges as a feasible mid-ground between these two extremes. The main idea in selective replication is to identify the documents that are of interest to the users of each geographical region and replicate the documents on the data centers according to the user interest. If this can be wisely done, queries can be locally processed in regional data centers, reducing the search quality loss relative to the second extreme and providing better scalability compared to the first extreme. Selective replication can be further coupled with selective forwarding of queries between data centers so that documents that are missing in the local top $k$ results (with respect to the global top $k$ results) can be retrieved from non-local data centers, preventing any search quality loss (Baeza-Yates et al., 2009; Cambazoglu, Varol, Kayaaslan, Aykanat, & Baeza-Yates, 2010).

In this paper, we propose strategies for selectively replicating documents in a geographically distributed search engine setting. Our strategies identify the documents that are of interest to the users of certain geographical regions, based on the occurrence frequencies of documents in past search results. The identified documents are then replicated and indexed on non-local data centers so that future queries can be efficiently and effectively processed.

The outline of the paper is as follows. In Section 2, we provide the details of the search engine architecture that we consider in this work and provide formal definitions for two variants of the document replication problem we aim to solve. Section 3 describes the datasets used in our work and the setup of our simulations. In Sections 4–6, we propose various replication algorithms, each optimizing different performance metrics under different constraints and assumptions. We report the experimental results about the performance of our algorithms in the associated sections. In Section 7, we investigate the impact of query forwarding on the performance. We survey the related work in Section 8. The paper is concluded in Section 9.

## 2. Preliminaries

### 2.1. Architecture

We consider a search engine architecture composed of multiple data centers. In this architecture, each data center crawls and stores documents belonging to a disjoint subset of the Web. Each data center then builds a local web index over its crawled documents, independent of the other data centers. We assume that IP addresses (or countries at a higher granularity) are statically assigned to data centers according to their geographical proximity. Each data center is responsible for processing queries that originate from its subset of IPs and is said to be the local data center for those queries.

In our architecture, certain documents are replicated on non-local data centers. Hence, in addition to its local index, each data center maintains a replicated index, built over its non-local documents. The replication pattern of documents is periodically determined based on the frequencies with which documents appear in the search results generated by individual data centers.

Queries are evaluated as follows.[2] A user query is first processed in the local data center associated with the user, over both local and replicated indexes. A local top $k$ result set is formed based on the estimated relevance scores of documents (Cambazoglu & Aykanat, 2006). At this point, this result set may be immediately returned to the user. Alternatively, the query may be forwarded to a set of non-local data centers, hoping to retrieve some documents whose scores are higher than that of the lowest scoring document in the locally computed top $k$ set. Forwarded queries are concurrently processed over the local indexes of non-local data centers, whose top $k$ results are returned to the local data center. These results are then merged in decreasing order of scores and the top $k$ results are returned to the user.

Fig. 1 illustrates the process. In the figure, each data center is represented by a large box. The patterns indicate the original assignment of documents to data centers. The box in the top row represents the local documents of a data center. The boxes in the bottom row represent non-local documents that are locally replicated. The directed arcs show contributions of different document collections to the final search results.

In this architecture, if a query is only locally processed, the search result quality may deteriorate as some of the documents that appear in the global top $k$ result set may not be available in the local data center. On the other hand, if the query is forwarded to non-local data centers, the query response time increases due to the network latency between the local and non-local data centers (the workload may also increase). Readers may refer to (Cambazoglu, Varol, et al., 2010) for more details about the architecture.

[2] We assume that the global collection statistics are made available to all data centers so that the scores generated by different data centers are comparable.