8

Overview of the Full-Text Document Retrieval Benchmark

Samuel DeFazio

Digital Equipment Corporation

8.1 Introduction

For most of recorded history, textual data have existed primarily in hard-copy format, and the related document retrieval process was essentially a manual task, possibly involving the assistance of cross-reference catalogs. By the mid-1960s, work was under way at the University of Pittsburgh to develop computer-assisted legal research systems [Harrington, 1984–85]. Also, during this period of time, computer-based document retrieval systems were beginning to emerge in commercial firms; for example, InfoBank at the *New York Times* [Harrington, 1984–85]. The most distinguishing characteristics of such systems include full-text Boolean search logic and support for proximity expressions (e.g., phrases). With this technology, termed full-text retrieval (FTR), documents are selected from a database in terms of *content*, rather than with predefined keywords or subject categories. For example, suppose that we were interested in locating articles about benchmarking full-text document retrieval systems. To formulate a search expression that would specify the desired content, we could select keywords (e.g., *benchmark, performance*) and phrases (e.g., *document retrieval*, *full-text retrieval*, *information retrieval*) which would likely be found within relevant documents. The reader should note that this simple example illustrates one important shortcoming of FTR systems: The inherent ambiguity of natural language makes FTR query formulation imprecise. Although FTR systems lack closed methods to formulate

queries, this technology tends to be significantly less time-consuming than manual document retrieval, especially for large databases.

Since the early work on computer-based document retrieval systems, considerable progress has been made toward applying FTR technology to harness the information content of textual data. Progress is particularly evident within the information industries. For example, FTR software is the foundation for electronic information services such as Dialog, LEXIS/NEXIS, and BRS. During the 1980s, information service companies automated the literature search process for many important commercial areas of intellectual discourse. During the 1990s, deployment of FTR systems is likely to increase dramatically within industry, government, and academia. To support this growth, IDC expects shipments of FTR system software products to exceed two million copies per year by 1995 [IDC, 1991].

The primary advantage of the FTR scheme is that documents can be located according to content; however, providing this capability is costly in terms of computing resources. As document databases grow, the computing power needed to support content searches tends to increase at least linearly [DeFazio & Hull, 1991]. Thus, when the database grows by a factor of *N*, the computing power must increase correspondingly in order to maintain the same search response time. This property derives primarily from the underlying software technology. Rather than exhaustively searching the raw text, FTR systems normally employ surrogate file structures to improve response time [Faloutsos, 1985]. To effectively support proximity searching, these surrogate files usually contain an entry for each token (e.g., word, number, date, time) occurrence in the document database. Consequently, the surrogate files, search processing time, and answer sets generally grow in proportion to the document database size. As such, when both the number of searches and the database size grow by a factor of *N*, the demand for computing power to maintain response time tends to increase by a factor of $N^{2.}$

Although FTR technology has existed for some time, the computer industry lacks a widely accepted, standard benchmark that measures the performance and price/performance of full-text document retrieval systems. Assuming that a large number of FTR software products will be acquired during the 1990s [IDC, 1991], having a uniform method to compare the performance of such systems appears highly desirable. The full-text document retrieval benchmark presented below is designed to provide this capability. Conceptually, this benchmark is similar to TPC-B in that the focus is performance of the "document retrieval engine" for some hardware configuration. Concentrating on the performance of this system component is justified since the related terminal workload for such applications tends to be much smaller and, therefore, less significant. That is, the FTR software for content searching typically generates most of the resource demands exhibited by full-text document retrieval systems. Thus, throughput for the benchmark is defined as partition searches completed per minute (SPM). Price/performance is computed as dollars per SPM, where dollars represents the total five-year ownership cost for the system.

To date, the benchmark has been validated at Sequent Computer Systems, Inc., with several commercially available FTR products. The initial validation tests indicated that the specification

is complete and workable. Since those activities were not audited, we are only able to summarize the related results. Our preliminary findings indicate that operating on Sequent Symmetry 750 platforms, some FTR products can handle multiple gigabyte databases and sustain the maximum throughput rate as defined by the benchmark.

The following sections contain an overview of the benchmark's business, database, and system models, along with descriptions of the associated transactions, response time requirements, workload generation procedure, and performance measures.

8.2 Business Model

This benchmark models multiuser FTR systems that locate and retrieve documents in large (i.e., one or more gigabyte) collections of textual data. We refer to an application of this type as a **document retrieval service** (DRS). Users maintain accounts with the DRS and sign on for service from terminals, PCs, or workstations. The DRS provides read-only access to the document database. Customers can select documents from the database using FTR queries, display results, and transfer text to their terminals. The DRS does not support end-user operations that modify the database. The model assumes that all maintenance functions are handled by the database administrator.

This business model encompasses a large number of full-text document retrieval applications. Examples include commercial information retrieval services such as Dialog and BRS, competitive analysis systems, technical document libraries, customer support and problem-reporting systems, and litigation support applications. In effect, the DRS model accommodates almost any document-based, multiuser application that supports full-text search capabilities.

8.3 Database Model

Logically, the database is structured as a collection of document **partitions**. Each partition contains a set of documents that are stored as variable length records. A record includes the entire text of one document and is represented as an entry in the **text file**. As shown in the following figure, a partition may also have an associated structure called the **search file**. The search file, if present, contains an "index" that the FTR software uses to improve the response time for locating documents. In this context, an index is any surrogate file structure that the FTR software uses to avoid exhaustively scanning the textual data.

The benchmark specification requires that the test database contain only documents which were authored by people. As such, documents in the test database may not be machine generated. The basic reason for requiring a database population such as this is to help ensure uniformity in the generated workload. With respect to token usage patterns, large collections of naturally

written text are statistically indistinguishable [Zipf, 1965]. Given this, when actual documents are used to build the test database, the generated workloads should be nearly identical for any body of text which conforms to the specification. Unfortunately, it is unclear whether this assertion can be made for a generated database. In the absence of such knowledge, the benchmark demands that "real" text be used to populate the test database.

8.4 System Model

The benchmark is based on the assumption that the DRS operates, logically, as a document server. In this model, users are represented by client processes that submit search and retrieval transactions to the DRS. Search transactions locate documents of interest, and each retrieval transaction fetches the text of one document. The DRS server handles search and retrieval transactions by performing read-only operations on the database. In the client process, each transaction is considered an atomic unit of work. The DRS server, however, may decompose transactions into smaller units of work.

8.5 Search Transactions

Search transactions are the means by which users locate documents, and they represent the major source of work for the benchmark. A transaction of this type contains a search expression that specifies the desired documents in terms of *content*. For this benchmark, a search expression is composed of terms and Boolean connectors (i.e., AND, OR, AND NOT). Each search term may be either a simple token (e.g., word) or a proximity operator (i.e., Phrase, WithinSentence, WithinParagraph). The output from a search transaction is an **answer set** that contains the unique identification, or *docid*, for each document which satisfies the related search expression.

The following example represents, with an SQL style syntax, the informal search expression described above for locating documents related to benchmarking full-text retrieval systems.

```
SELECT      docid
FROM        Document_Database
WHERE       WithinParagraph("benchmark", "performance")
AND         Phrase("document retrieval")
OR          Phrase("full-text retrieval")
OR          Phrase("information retrieval")
```

As illustrated, proximity operators provide scope for the associated tokens. For example, the search term Phrase("full-text retrieval") locates documents that contain the phrase "full-text retrieval." By contrast, a search expression containing the terms "full-text" AND "retrieval"

finds documents in which the tokens "full-text" and "retrieval" appear anywhere in the related text.

The benchmark specifies that the input for search transactions be randomly generated. That is, terms and connectors are independently selected and uniformly distributed over the respective ranges. Search expressions may have from 1 to 50 tokens. Under this range of values, a search expression for the benchmark contains, on average, 25 tokens.

Tokens are selected for a search expression from the database vocabulary (i.e., the set of unique tokens in the database). Since token usage patterns in large collections of documents are known to follow a "Zipf" distribution [Zipf, 1965], random selection over the entire vocabulary would provide search expressions that differ from what one would expect to observe for FTR applications. To address this, the benchmark specification requires the vocabulary to be segmented into *high use*, *moderate use*, and *low use* tokens. This segmentation is performed on the **search vocabulary**, which is a subset of the database vocabulary with the numeric tokens and **noise words** (i.e., the 50 most frequently occurring tokens such as "or," "and," "of," "the," "a") removed. The *high use* segment contains that portion of the search vocabulary which generates 90 percent of the token occurrences. Zipf [1965] has shown that for large collections of text, the *high use* vocabulary (including noise words) is usually fewer than 10,000 tokens. The *low use* segment generates 5 percent of the token occurrences within the search vocabulary and corresponds to the least frequently used portion of the tokens. This segment contains most of the search vocabulary (typically about 90 percent of the tokens) and tends to be dominated by proper nouns, acronyms, misspelled words, and so on. The *moderate use* segment has those tokens that fall between the *low use* and *high use* vocabularies. The benchmark requires this segment to contain that portion of the search vocabulary which represents 5 percent of the token occurrences. Using these segments, tokens are selected for search expressions by first randomly choosing the segment, then randomly picking a token within that segment.

The search transaction is based on the notion of a traditional "Boolean query" augmented with proximity operators. Clearly, there are many other methods that can be used to specify the desired document content for a search transaction. It is not our intent to argue the merit of this approach; the commercial marketplace has already done so. That is, the overwhelming majority of commercially available products for document retrieval employ underlying Boolean search mechanisms which support, among other things, proximity operators. Thus, the benchmark as designed is applicable to a large number of existing products and, therefore, could benefit a significant portion of the customer base for document retrieval technology.

8.6 Retrieval Transactions

Retrieval transactions take a *docid* as input and return the full text of the related document. The acts of searching a database and retrieving documents from the associated answer set are not necessarily performed consecutively. As such, the benchmark does not attempt to relate search

and retrieval transactions. The benchmark requires that the *docid* for a retrieval transaction be randomly selected over the range of possible values for the test database.

8.7 Response Time Requirements

The benchmark requires 90 percent of the search transactions to be completed within 20 seconds. Thus, the DRS application is forced to provide search transaction response times which are commensurate with the large amount of work that is required [DeFazio & Hull, 1991]. Relatively speaking, retrieval transactions are not very labor intensive. As such, the benchmark requires 90 percent of the retrieval transactions to be completed within 2 seconds. Consequently, the DRS application must ensure that users can obtain documents for display in a reasonable amount of time.

8.8 Workload Generation

The benchmark attempts to generate a realistic DRS workload by employing a process which is based on the following assumptions:

1. The complexity of search expressions, with respect to the number of tokens, must vary significantly.
2. The DRS workload must contain a relatively uniform mix of search and retrieval transactions.
3. The database size must scale with the number of search transactions.

These assumptions are cast into the benchmark specification by means of the following parameters:

1. **Search Expression Size**. The range of tokens for a search expression is from one to 50 tokens.
2. **Transaction Mix.** The ratio of retrieval transactions to search transactions in the workload is fixed at 10 to 1.
3. **Scaling**. The database increases in size by one partition for each 50 search transactions completed per minute. A partition contains 1 GB (i.e., 10^9 bytes) of text and 200,000 documents.

To vary the complexity of search transactions, the benchmark specifies that the number of tokens be randomly selected over the stated range. According to Haskin [1982], the average

number of tokens per search is approximately ten, but the variance tends to be high. Also, most DRS applications provide some form of query augmentation such as synonyms, stems, or wildcards. The net "logical" effect of all these features is to expand the number of tokens entered by the user. For example, given the wildcard "comput*", the search expression would be expanded to contain related tokens in the database vocabulary such as compute, computer, computing, and so on. Since such features are highly application specific, the benchmark does not include any such requirement for the generation of search expressions. The rationale is that by defining the range of tokens per search to be from one to 50, essentially the same behavior is obtained at significantly less complexity.

In practice, a DRS typically processes multiple retrieval transactions per search. Also, the average number of document retrievals executed by a DRS per search transaction tends to be relatively stable over time. The benchmark models this workload characteristic by requiring a fixed ratio of ten retrieval transactions for each search transaction.

Scaling for the benchmark is based on the search transaction rate. That is, the database grows by one partition (i.e., 1 GB of text) for each 50 search transactions completed per minute. The target for each search transaction is the entire database. To improve the performance of search transactions, the database may be physically partitioned. Thus, the related FTR system software may issue multiple database transactions for each search transaction as defined by the benchmark.

8.9 Performance Metrics

The benchmark includes throughput and price/performance metrics. Conceptually, the notion of throughput for the benchmark is multidimensional. One dimension of the throughput metric relates to search transactions completed per minute, and the other relates to database size. More formally, the benchmark defines throughput, denoted SPM, as partition searches completed per minute. The throughput value for a benchmark run is obtained by using the following computation:

throughput = [search transactions completed per minute] * [database size in partitions]

Using the search transaction rate as a throughput computation factor is rooted in the notion that performance is "linked" to finding documents. That is, the essence of this benchmark is the selection of documents from the database using FTR technology to process Boolean search expressions. The amount of work associated with each search transaction is proportional to the database size [DeFazio & Hull, 1991]. Since the benchmark scales in both database size and number of search transactions, the workload tends to grow quadratically. For example, 100 search transactions accessing a 2-GB database generate four times the amount of work associated

with 50 search transactions that target a 1-GB database. The throughput metric reflects this property of FTR technology by including database size as a factor in the computation.

Notice that retrieval transactions are not used in the throughput computation. Conceptually, retrieval transactions are included as part of the workload only to ensure that while search processing is taking place, the system under test can deliver acceptable response times for requests to display, download, or print documents.

Price/performance for the benchmark is

dollars/throughput

where dollars is the five-year cost of ownership for the system and throughput is defined as above. Assuming both quadratically increasing resource demands and related hardware costs, different price/performance values can be compared directly, without needing intimate knowledge of the benchmark.

It should be recognized that the throughput metric defined above differs significantly from the standard benchmarks issued by the TPC. Given the nature of FTR technology, this difference appears necessary. Our formulation of throughput is certainly open for discussion, and possibly refinement over time based on experience gained from using the benchmark.

8.10 Benchmark Specification Style and Content

The full-text document retrieval benchmark specification has been submitted to the TPC for consideration. The format and content of the benchmark are consistent with other TPC standards. As with the TPC standard benchmarks, the specification is designed to minimize ambiguity at the expense of formality and to be complete. That is, one should be able to develop driver software and run the benchmark based strictly on the specification. Sample software to assist users in correctly interpreting the specification is presented in Appendix A and Appendix B of the benchmark. This software is not part of the benchmark specification.

Acknowledgments

Early in 1991, work on the full-text document retrieval benchmark was initiated at Sequent Computer Systems, Inc. Since then, many knowledgeable individuals have reviewed and helped me refine the specification. I would like to take this opportunity to express my sincere gratitude to:

David Becker, Mead Data Central
Garth Boyd, Mead Data Central

Michael Cation, Verity
Peter Chellone, Oracle
Thomas Couvreur, Chemical Abstracts Service
Afsaneh Eshghi, Sequent Computer Systems
Aki Fleshler, Sequent Computer Systems
Hector Garcia-Molina, Stanford University
Jim Gray, Digital Equipment Corporation
Charles Greenwald, Mead Data Central
Clifford Reid, Verity
Michael Squires, Sequent Computer Systems
Anthony Tomasic, Stanford University

I hope that the contributions of these individuals will ultimately be rewarded by the emergence of a generally accepted standard benchmark, derived from this work, for measuring the performance of full-text document retrieval systems.

References

DeFazio, S., & Hull, J. (1991). Toward servicing textual database transactions on symmetric shared memory multiprocessors. *Proceedings of the Fourth International Workshop on High Performance Transaction Processing Systems*. Asilomar Conference Center, September, 1991.

Faloutsos, C. (1985). Access methods for text. *ACM Computing Surveys*, *17*(1), March, 1985, pp. 49–74.

Harrington, W. J. (1984–85). A brief history of computer-assisted legal research. *Law Library Journal*, *77:*453. (1984–85), pp. 543–556.

Haskin, R. (1982). Hardware for searching very large databases. *Proceedings on Database Engineering*.

IDC (1991). *Workgroup application systems, full-text retrieval systems market review and forecast, 1989/90–1995*. Framingham, MA: International Data Corp., February, 1991.

Zipf, G. (1965). *The psycho-biology of language: An introduction to dynamic philology*. Boston: Houghton Mifflin, 1935; Cambridge, MA: MIT Press, 1965.