Brute Force Information Retrieval Experiments using MapReduce

by Djoerd Hiemstra and Claudia Hauff

MIREX (MapReduce Information Retrieval Experiments) is a software library initially developed by the Database Group of the University of Twente for running large scale information retrieval experiments on clusters of machines. MIREX has been tested on web crawls of up to half a billion web pages, totaling about 12.5 TB of data uncompressed. MIREX shows that the execution of test queries by a brute force linear scan of pages, is a viable alternative to running the test queries on a search engine's inverted index. MIREX is open source and available for others.

Research in the field of information retrieval is often concerned with improving the quality of search systems. The quality of a search system crucially depends on ranking the documents that match a query. To get the best documents ranked in the top results for a query, search engines use numerous statistics on query terms and documents, such as the number of occurrences of a term in the document, the number of occurrences of a term in the collection, the number of hyperlinks pointing at a document, the number of occurrences of a term in the anchor texts of hyperlinks pointing at a document, etc. New ranking ideas are tested off-line on query sets with human rated documents. If such ideas are radically new, experimentally testing them might require a non-trivial amount of coding to change an existing search engine. If, for instance, a new idea requires information that is not currently in the search engine's inverted index, then the researcher has to re-index the data or even recode parts of the system's indexing facilities, and possibly recode the query processing facilities that access this information. If the new idea requires query processing techniques that are not supported by the search engine (for instance sliding windows, phrases, or structured query expansion) even more work has to be done.

Proud researchers and their cluster

Instead of using the indexing facilities of the search engine, we propose to use MapReduce to test new retrieval approaches by sequentially scanning all documents. Some of the advantages of this method are: 1) Researchers spend less time on coding and debugging new experimental retrieval approaches; 2) It is easy to include new information in the ranking algorithm, even if that information would not normally be included in the search engine's inverted index; 3) Researchers are able to oversee all or most of the code used in the experiment; 4) Large-scale experiments can be done in reasonable time.

MapReduce was developed at Google as a framework for batch processing of large data sets on clusters of commodity machines. Users of the framework implement a mapper function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reducer function that processes intermediate values associated with the same intermediate key. For the example of simply counting the number of terms occurring across the entire collection of documents, the mapper takes as input a document URL (key) and the document content (value) and outputs pairs of term and term count in the document. The reducer then aggre-

gates all term counts of a term together and outputs the number of occurrences of each term in the collection. Our experiments are made of several such MapReduce programs: We extract anchor texts from web pages, we gather global statistics for terms that occur in our test queries, we remove spam pages, and we run a search experiment by reading web pages one at a time, and on each page we execute all test queries. Sequential scanning allows us to do almost anything we like, for instance sophisticated natural language processing. If the new approach is successful, it will have to be implemented in a search engine's indexing and querying facilities, but there is no point in making a new index if the experiment is unsuccessful. Researchers at Google and Microsoft have recently reported on similar experimental infrastructures.

When implementing a MapReduce program, users do not need to worry about partitioning of the input data, scheduling of tasks across the machines, machine failure, or interprocess communication and logging: All of this is automatically handled by the MapReduce runtime. We use Hadoop: an open source implementation of Google's file system and MapReduce. A small cluster of 15 low cost machines suffices to run experiments on about half a billion web pages, about 12.5 TB of data if uncompressed. To give the reader an idea of the complexity of such an experiment: An experiment that needs two sequential scans of the data requires about 350 lines of code. The experimental code does not need to be maintained: In fact, it should be retained in its original form to provide data provenance and reproducibility of research results. Once the experiment is done, the code is filed in a repository for future reference. We call our code repository MIREX (MapReduce Information Retrieval EXperiments), and it is available as open source software from: http://mirex.sourceforge.net

MIREX is sponsored by the Netherlands Organization for Scientific Research NWO, and Yahoo Research, Barcelona.

Links:

MIREX: http://mirex.sourceforge.net
Database Group:
http://db.cs.utwente.nl
Web Information Systems Group:
http://wis.ewi.tudelft.nl

Please contact:

Djoerd Hiemstra
University of Twente, The Netherlands
E-mail: hiemstra@cs.utwente.nl