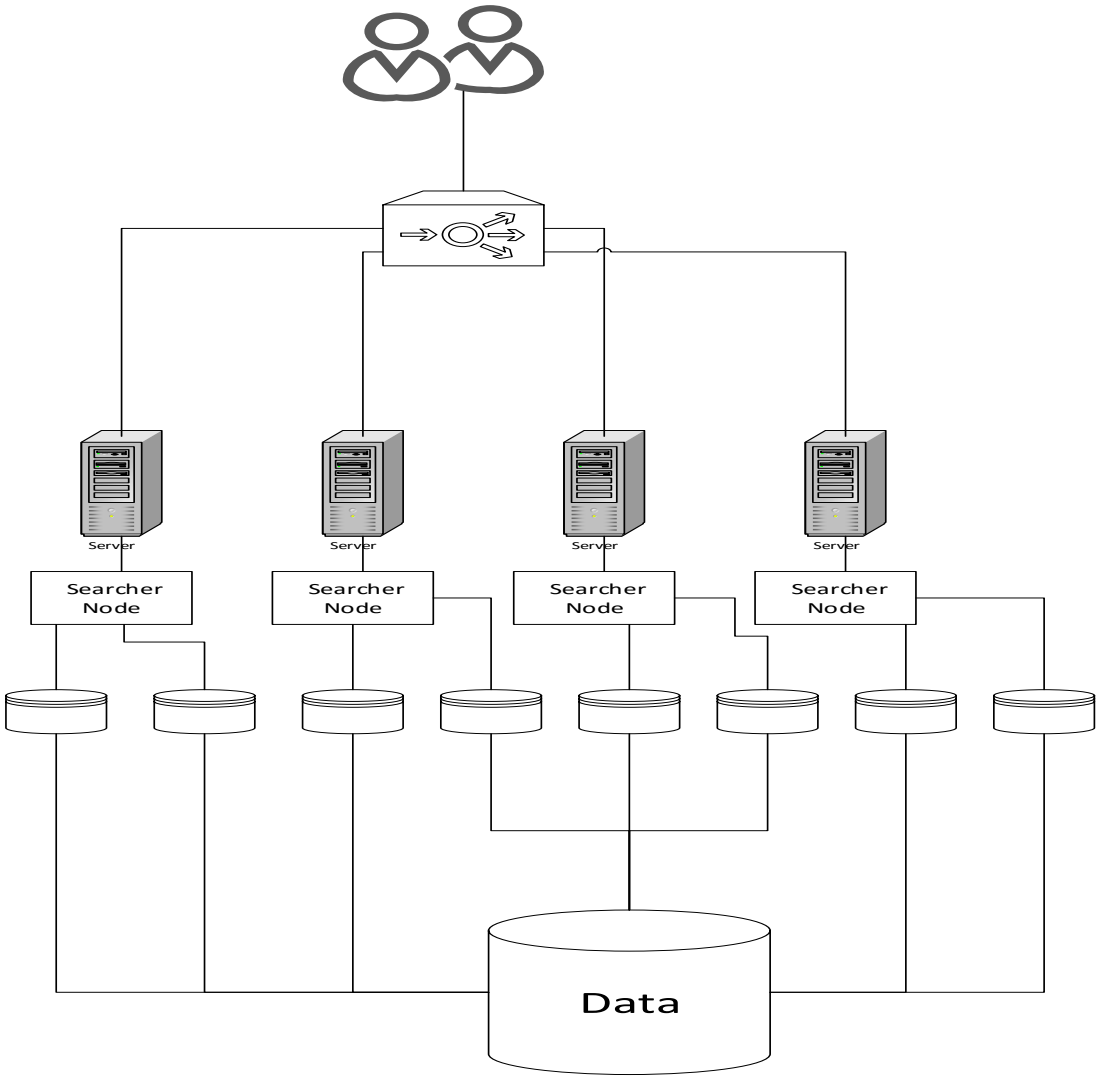
Abstract

Rather than blindly returning anything that contains the text as keyword, Semantic Search takes into account the context of your search as well as the underlying meaning of the documents to be searched. Semantic Search is defined as search for information based on the intent of the searcher and contextual meaning of the search terms, instead of depending on the dictionary meaning of the individual words in the search query. Semantic search systems consider various points including context of search, location, intent, and variation of words, synonyms, generalized and specialized queries, concept matching and natural language queries to provide relevant search results.

Scalability is also one of the strong factors which determine its feasibility over large data sets. For this end, shared nothing architecture can provide the flexibility needed. To sum it up, the project objective is to develop a search engine providing search based on underlying semantics core concepts of the query and make it scalable by taking advantage of the shared nothing architecture.

In this project, we have tried to solve the problem of searching data across multiple nodes using semantic context and managing the constant influx of data generated in real time distributed environments, making it searchable almost instantaneously. The difficulty lies in the semantic content extraction, distributed environment setting and constant data generation. In our system, semantic context extraction considers synonyms, concept matching and lemmatization. For mapping the documents and making them searchable, we have used the vector space model for document representation using term frequency- inverse document frequency (tf-idf) metric. The extraction of semantic concepts of the query is done during the parsing phase of query as well as the documents. The architecture of the search engine consists of central controller and load balancer and connected to it are the sharded nodes searching the data stored locally in the databases maintaining each of their respective indices.

Project Design



Architectural Layout

The architecture of our system is given above. When the user fires a search query to the engine the query passes through the load balancer or the query redirector which sends the query to all the nodes which have registered. These nodes are independent of each other and maintain their own database having their own indices. This structure solves the scalability issue that many similar systems face. The arrangement of nodes in such a way is termed as the sharded or “shared-nothing” architecture design that we have gone in for achieving the performance enhancements goals that were set.

Once the query is received by the node, the search head takes over to check the whether the query's correct result is available in the node. After the results are obtained in the nodes, they are sent back to the load balance or query redirector which as the job of not only redirecting but also to merging the results that arrive from the independent nodes. Once all the nodes send their responses, they are merged and a uniformed search result is returned back to the user.

The architecture also supports the addition of new data. This is achieved by allowing the new data to be temporarily stored into a directory which results in activation of a trigger that helps in deciding into which shard/node the data is to be sent for storage. Some of the parameters that are used to decide where the data is to be stored are: current state of node, amount of free space etc. The optimal node is selected as the one on top of priority queue having the maximum value of the score calculated based on the parameter after they have been normalized.

Once the optimal node is found the data is transferred. Once the data reaches the node it needs to be indexed in order to be used for searching. For this each node which is online, in regular intervals the data which is stored is indexed so as to reflect any new data into the index list.

With respect to the search specifics, different parsers are made available for parsing different type of files. Supported files are PDF, HTML, Text files. The searching procedure takes place as follows:

1. Tokenizing the query.
2. Stemming the query to its root can be replaced with Lemmatization (stemming the query after considering the context).
3. Matching query and calculating score using tf-idf calculation.

Stanford Lemmatizer is used in this context to tag the POS in the statement. The POS which are considered to have little information are not considered or they are not converted to their base forms.

Technologies Used

Lucene

Lucene is an open source, highly scalable text search-engine library available from the Apache Software Foundation. You can use Lucene in commercial and open source applications. Lucene's powerful APIs focus mainly on text indexing and searching. It can be used to build search capabilities for applications such as e-mail clients, mailing lists, Web searches, database search, etc. Web sites like Wikipedia, TheServerSide, jGuru, and LinkedIn have been powered by Lucene.

Lucene also provides search capabilities for the Eclipse IDE, Nutch (the famous open source Web search engine), and companies such as IBM®, AOL, and Hewlett-Packard. Lucene has been ported to many other programming languages, including Perl, Python, C++, and .NET. As of 30 Jul 2009, the latest version of Lucene in the Java™ programming language is V2.4.1.

Features of Lucene:

- Has powerful, accurate, and efficient search algorithms.
- Calculates a score for each document that matches a given query and returns the most relevant documents ranked by the scores.
- Supports many powerful query types, such as PhraseQuery, WildcardQuery, RangeQuery, FuzzyQuery, BooleanQuery, and more.
- Supports parsing of human-entered rich query expressions.
- Allows users to extend the searching behavior using custom sorting, filtering, and query expression parsing.
- Uses a file-based locking mechanism to prevent concurrent index modifications.
- Allows searching and indexing simultaneously.

Apache Tomcat

Apache Tomcat is an open-source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements several JavaEE specifications including JavaServlet, JavaServer Pages (JSP), JavaEL and WebSocket and provides a pure JavaHTML web server environment for Java code to run.

Java Script

JavaScript is a dynamic programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also used in server-side network programming with runtime environments such as Node.js, game development and the creation of desktop and mobile applications. With the rise of the single-page web app and JavaScript-heavy sites, it is increasingly being used as a compile target for source-to-source compilers from both dynamic languages and static languages. In particular, Emscripten and highly optimized JIT compilers, in tandem with asm.js that is friendly to AOT compilers like Odin Monkey, have enabled C and C++ programs to be compiled into JavaScript and execute at near-native speeds, making JavaScript be considered the "assembly language of the web", According to its creator and others.

JavaScript is classified as a prototype-based scripting language with dynamic typing and first-class functions. This mix of features makes it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

Despite some naming, syntactic, and standard library similarities, JavaScript and Java are otherwise unrelated and have very different semantics. The syntax of JavaScript is actually derived from C, while the semantics and design are influenced by the Self and Scheme programming languages.

JQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

The Spring Framework

The Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application.

Spring enables you to build applications from "plain old Java objects" (POJOs) and to apply enterprise services non-invasively to POJOs. This capability applies to the Java SE programming model and to full and partial Java EE.

How the Spring Framework benefits developers:

- Make a Java method execute in a database transaction without having to deal with transaction APIs.
- Make a local Java method a remote procedure without having to deal with remote APIs.
- Make a local Java method a management operation without having to deal with JMX APIs.
- Make a local Java method a message handler without having to deal with JMS APIs.

Java Servlet

A Java servlet is a Java programming language program that extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly implement applications hosted on Web servers. Such Web servlets are the Java counterpart to other dynamic Web content technologies such as PHP and ASP.NET.

Servlets are most often used to:

- Process or store a Java class in Java EE that conforms to the Java Servlet API, a standard for implementing Java classes which respond to requests. Servlets could in principle communicate over any client–server protocol, but they are most often used with the HTTP protocol. Thus "servlet" is often used as shorthand for "HTTP servlet". Thus, a software developer may use a servlet to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets can maintain state in session variables across many server transactions by using HTTP cookies, or URL rewriting.
- To deploy and run a servlet, a web container must be used. A web container (also known as a servlet container) is essentially the component of a web server that interacts with the servlets. The web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

The Servlet API, contained in the Java package hierarchy javax.servlet, defines the expected interactions of the web container and a servlet.

A Servlet is an object that receives a request and generates a response based on that request. The basic Servlet package defines Java objects to represent servlet requests and responses, as well as objects to reflect the servlet's configuration parameters and execution environment. The package javax.servlet.http defines HTTP-specific subclasses of the generic servlet elements, including session management objects that track multiple requests and responses between the web server and a client. Servlets may be packaged in a WAR file as a web application.

MySQL Database

MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Implementation

Welcome page

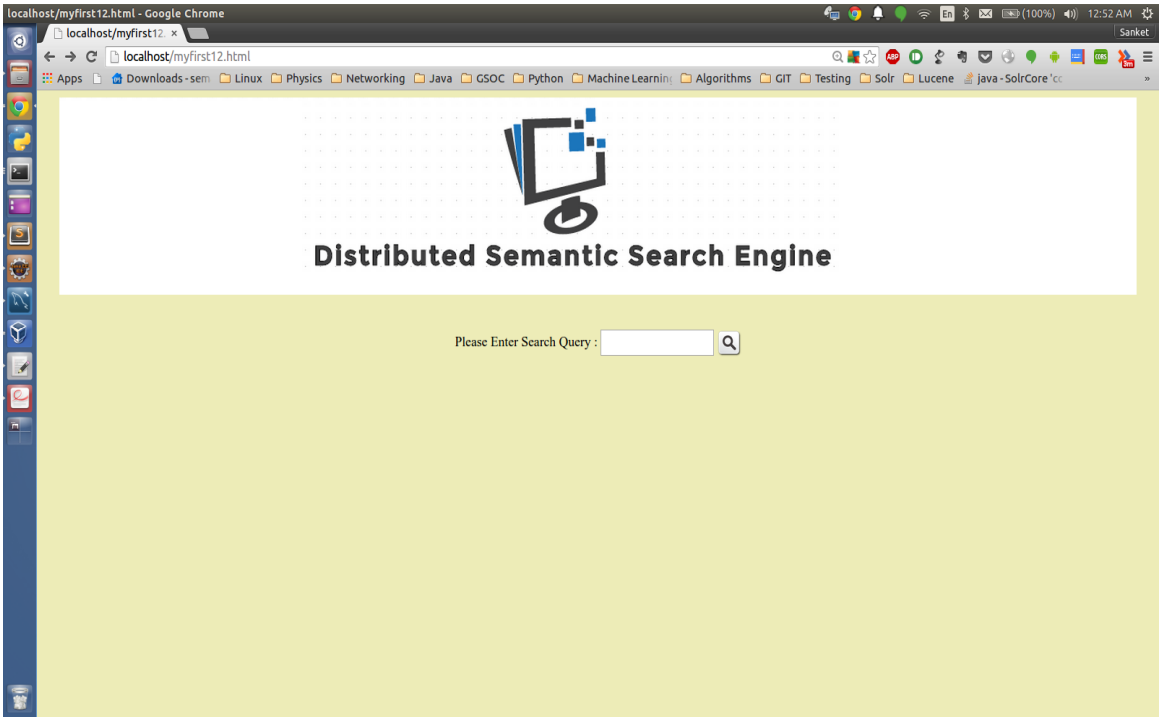


Figure. Welcome page

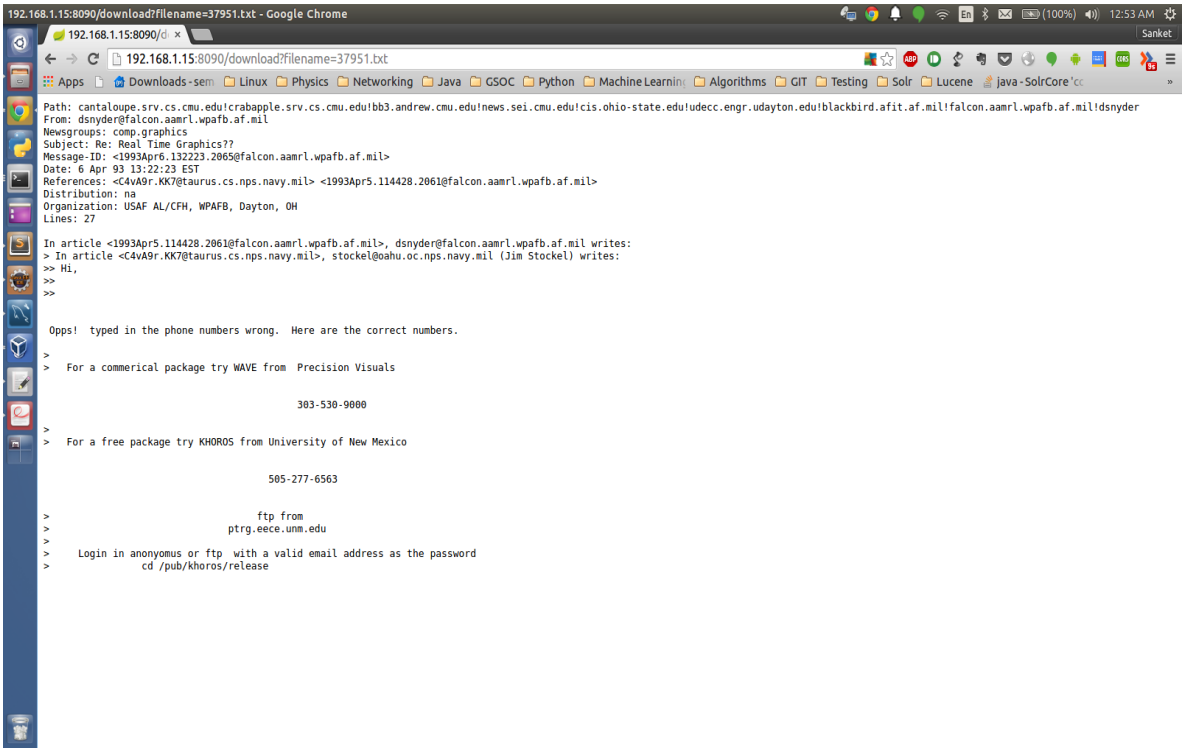
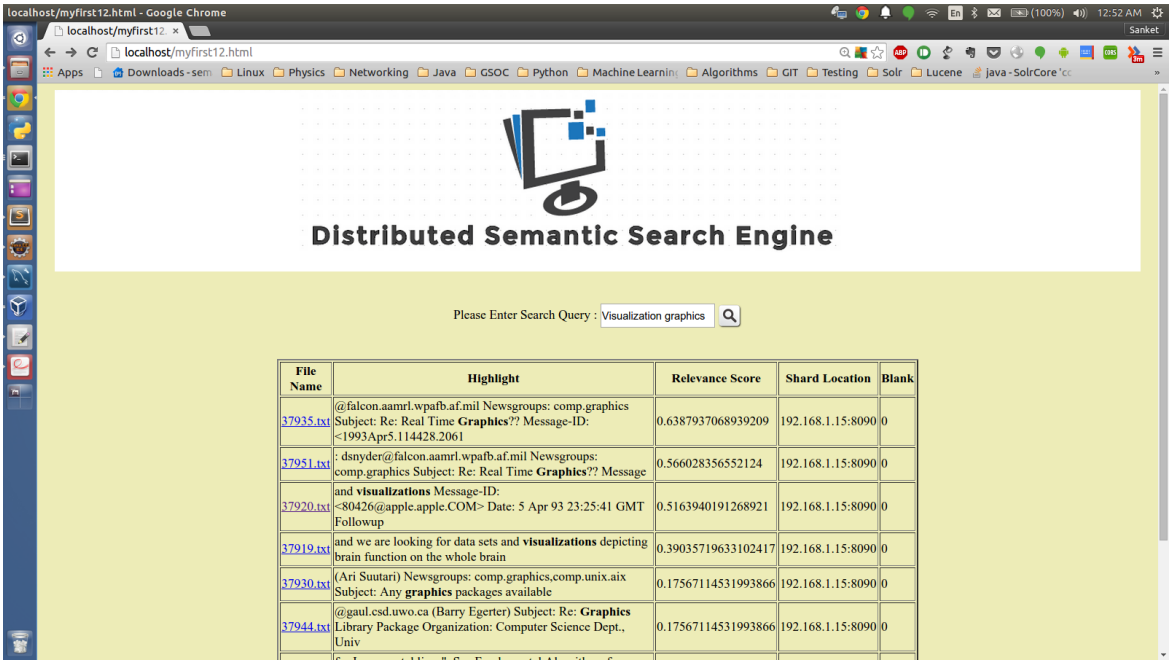


Figure. Starting Server

Results Displayed



File Name	Highlight	Relevance Score	Shard Location	Blank
37935.txt	@falcon.aamrl.wpafb.af.mil Newsgroups: comp.graphics Subject: Re: Real Time **Graphics**?? Message-ID: <1993Apr5.114428.2061	0.6387937068939209	192.168.1.15:8090	0
37951.txt	: dsnyder@falcon.aamrl.wpafb.af.mil Newsgroups: comp.graphics Subject: Re: Real Time **Graphics**?? Message	0.566028356552124	192.168.1.15:8090	0
37920.txt	and **visualizations** Message-ID: <80426@apple.apple.COM> Date: 5 Apr 93 23:25:41 GMT Followup	0.5163940191268921	192.168.1.15:8090	0
37919.txt	and we are looking for data sets and **visualizations** depicting brain function on the whole brain	0.39035719633102417	192.168.1.15:8090	0
37930.txt	(Ari Suutari) Newsgroups: comp.graphics,comp.unix.aix Subject: Any **graphics** packages available	0.17567114531993866	192.168.1.15:8090	0
37944.txt	@gaul.csd.uwo.ca (Barry Egerter) Subject: Re: **Graphics** Library Package Organization: Computer Science Dept., Univ	0.17567114531993866	192.168.1.15:8090	0

Figure. Result Set