Meta Search Engine Architecture and Guided Google as the Best Example for Meta Search Engine

Akshatha. G,
1st M.Tech, CNE,
BNMIT, Bangalore.

Tirupati Naga Vaishnavi,
1st M.Tech, VLSI Design & Embedded Systems,
RVCE, Bangalore.

Abstract – Information Retrieval on the internet is gaining very much importance in the day to day life. Now-a-days there are millions of websites available on the internet and there is a competition between them for providing best, fast & accurate results to the customers as soon as possible. This purely depends on the search engines, how fast they process the search queries and provide the results. Meta search engines are such systems that can provide effective information by accessing multiple existing search engines such as Dogpile, Meta-crawler, etc. But most of the cannot successfully operate on heterogeneous & fully dynamic web environment. This paper presents a web service architecture for meta search engine to fulfill the need of heterogeneous and dynamic web environment. Of all the meta-search engines, Google guide is the best example. The goal of this applications are to help ease and guide the searching efforts of web users towards their desired objective. The Guided Google serves as an advanced interface to the actual google.com search engine.

Index terms- Meta search engine, web service, google cluster, google API, Guided Google.

I. INTRODUCTION

Now-a-days, internet usage has become a priority requirement for every person. People depend on internet for different activities such as browsing, ecommerce, e-banking etc. Though automatic information retrieval (IR) existed before world wide web (www), post-internet era has made it indispensable. IR is a sub field of computer science consent with presenting relevant information, gathered from online information sources to users in response to search queries. Various types of IR tools have been created, solely to search engines (SE), other useful tools are deep-web search portals, web – directories & Meta- search engines (MSE).

The dependence of the internet is growing day by day for search of file, videos, meaning etc. Google is one of the popular search engines, supports web service that allows users to search. A main objective of web applications is to provide fast and accurate results. As we expect the results of the search to be fast and accurate it purely depends on the search engines how fast they evaluate the queries.

Characteristics of Google search engine:-
• Google is one of the five most popular websites in the world.
• Google web search that lets you find other sites on web based.
• Google also provides specialised search through videos, news items etc..
• Google provides internet services that let you create blogs, send mail, and publish web pages.

It can be said that Google focuses on a page’s relevance and not on the number of responses. Only a small number of web users actually know how to utilize the true power of Google. Most average web users make searches based on irrelevant query keywords or sentences, which returns unnecessary, or worse, inaccurate results.

Google has been providing access to its services via various interfaces such as the Google Toolbar and wireless searches. And now the company has made its index available to other developers through a Web services interface. This allows the developers to programmatically send a request to the Google server and get back a response. The main idea is to give the developers access to Google’s functionality so that they can build applications that will help users make the best of Google. The Web service offers existing Google features such as searching, cached pages, and spelling correction. Currently the service is still in beta version and is for non-commercial use only. It is provided via SOAP (Simple Object Access Protocol) over HTTP and can be manipulated in any way that the programmer pleases.

This paper explores the web services, Meta Search Engine & the architecture of the Guided Google. In section 2, we propose the existing background of the web services, In section 3, we discuss the existing architecture of a Meta Search Engine. Section 4 we discuss the advantages of existing meta service engines. section 5, we discuss about the Cluster & Google cluster architecture, Google API. Section 6 we discuss the guided Google architecture. In Section 7, we propose design and implementation of Google cluster. In section 8, evaluation of different searching techniques. In section 9, finally presents the conclusions and the future work.
II. WEB SERVICE

Web service protocol is designed for providing service via web. Web Services are emerging as the fundamental building blocks for creating distributed, integrated and interoperable solutions across the Internet. They represent a new paradigm in distributed computing that allow applications to be created from multiple Web Services dispersed across the web originating from various sources regardless of where they reside or how they were implemented.

The format of the messages exchanged between a client and a Web service is specified by a standard called SOAP. The Simple Object Access Protocol (SOAP) is defined by the W3C as a lightweight protocol for exchange of information in a decentralized, distributed environment. SOAP is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application defined data types, and a convention for representing remote procedure calls and responses.

Web service messages are sent across the network in an XML format defined by the W3C SOAP specification. In most Web services, there are two types of SOAP messages: requests and responses. When a SOAP request is received, the Web service performs an action based on the request and returns a SOAP response. In many implementations, SOAP requests are similar to function calls with SOAP responses returning the results of the function call. With SOAP, a communication between Web services is possible and structured and each participant knows how to send or receive the corresponding SOAP Message. The final step to complete the communication architecture of Web services is to define how to access a service once it is implemented.

III. METASEARCH

Metasearch is to utilize multiple other search systems (called component search systems) to perform simultaneous search. A metasearch engine is a search system that enables metasearch. To perform a basic metasearch, a user query is sent to multiple existing search engines by the metasearch engine; when the search results returned from the search engines are received by the metasearch engine, they are merged into a single ranked list and the merged list is presented to the user.

Meta Search Engines can be classified into two types. a) General purpose metasearch engine and b) Special purpose Meta Search Engines. The former aims to search the entire Web, while the latter focuses on searching information in a particular domain (e.g., news, jobs).

1) Major Search Engine Approach: This approach uses a small number of popular major search engines to build a metasearch engine. Thus, to build a general-purpose metasearch engine using this approach, we can use a small number of major search engines such as Google, Yahoo!, Bing (MSN) and Ask. Similarly, to build a special purpose Meta Search Engine for a given domain, we can use a small number of major search engines in that domain.

2) Large scale Metasearch engine approach: In this approach, a large number of mostly small search engines are used to build a Metasearch engine. For example, to build a general-purpose metasearch engine using this approach, we can conceivably utilize all documents driven search engines on the Web. Such a metasearch engine will have millions of component search engines. Similarly to build a special purpose metasearch engine for a given domain with this approach, we can connect to all the search engines in that domain. For instance, for the news domain, tens of thousands of newspaper and news-site search engines can be used.
this case, the search engine selector is probably not needed. However, if the number of component search engines is large, as in the large scale Meta Search Engine scenario, then sending each query to all component search engines will be an inefficient strategy because most component search engines will be useless with respect to any particular query.

Passing a query to useless search engines may cause serious problems for efficiency. Generally, sending a query to useless search engines will cause waste of resources to the metasearch engine server, each of the involved search engine servers and the Internet. The problem of identifying potentially useful component search engines to invoke for a given query is the search engine selection problem, which is sometimes also referred to as database selection problem, server selection problem, or query routing problem. Obviously, for metasearch engines with more component search engines and/or more diverse component search engines, having an effective search engine selector is more important.

Web Service:
A Meta search engine is a search engine that collects results from other search engine. Web service offers such functionality and then presents a summary of that information as the results of a search. Most search engines available on the Web provide only a browser based interface; however, because Web services start to be successful, some of those search engines offer also an access to their information through Web services.

SOAP, the Simple Object Access Protocol was developed to enable a communication between Web services. It was designed as a lightweight protocol for exchange of information in a decentralized, distributed environment. SOAP is an extensible, text based framework for enabling communication between diverse parties that have no prior knowledge of each other. This is the requirement a transport protocol for Web services has to fulfill. SOAP species a mechanism to perform remote procedure calls and therefore removes the requirement that two systems must run on the same platform or be written in the same programming language.

Result Extractors:
After a component search engine processes a query, the search engine will return one or more response pages. A typical response page contains multiple (usually 10) search result records, each of which corresponds to a retrieved Web page, and it typically contains the URL and the title of the page, a short summary (called snippet) of the page content, and some other pieces of information such as the page size. Fig 2 shows the upper portion of a response page from the Google search engine. Response pages are dynamically generated HTML documents, and they often also contain content unrelated to the user query such as advertisements (sponsored links) and information about the host Web site. Since different search engines often format their results differently, a separate result extractor is usually needed for each component search engine.

Although experienced programmers can write the extractors manually, for large-scale metasearch engines, it is desirable to develop techniques that can generate the extractors automatically.

Result Merger:
After the results from the selected component search engines are returned to the metasearch engine, the result merger combines the results into s single ranked list. The ranked list of search result records is then presented to the user, possible 10 records on each page at a time, just like more search engines do. Many factors may influence how result merging will be performed and what the outcome will look like. A good result merger should rank all returned results in descending order of their desirability.

IV. ADVANTAGES OF WEB SERVICE META SEARCH ENGINE

We attempt to provide a comprehensive analysis of the potential advantages of metasearch engines over search engines. We will also focus on the comparison of metasearch engine and search engine.

Increased Search Coverage:
Metasearch engine can search any document that is indexed by at least one of the search engines. Metasearch engine with multiple major search engines as components will have larger coverage than any single component search engine. Different search engines often employ different document representation and result ranking techniques, and as a result, they often return different sets of top results for the same user query. Thus, by retrieving from multiple major search engines, a metasearch engine is likely to return more unique high quality results for each user query.

Better Content Quality:
The quality of the content of a search engine can be measured by the quality of the documents indexed by the search engine. General-purpose metasearch engines implemented using the large-scale metasearch engine approach has a better chance to retrieve more up-to-date information than major search engines and metasearch engines that are built with major search engines.

Good Potential for Better Retrieval Effectiveness:
More unique results are likely to be obtained, even among those highly ranked ones, due to the fact that different major search engines have different coverage and different document ranking algorithms. The result-merging component of the metasearch engine can produce better results by taking advantage of the fact that the document collection of major search engines has significant overlaps. This means that many shared documents have the chance to be ranked by different search engines for any given query. In general, if a document if retrieved by more search engines, the document is more likely to be relevant.
Better Utilization of Resources:
Metasearch engine use component search engines to perform basic search. This allows them to utilize the storage and computing resources of these search engines. Among the available search engines guided google is the best example which is discussed here.

V. GOOGLE CLUSTER, GOOGLE CLUSTER ARCHITECTURE AND GOOGLE API

Google
The dependence of the internet is growing day by day for search of file, videos, meaning etc. Google is one of the popular search engines, supports web service that allows users to search. A main objective of web applications is to provide fast and accurate results. As we expect the results of the search to be fast and accurate it purely depends on the search engines how fast they evaluate the queries.

Cluster
A web cluster is a distributed system of computers or servers. In other words, it’s a group of computers which appear to the outside world as a single entity. Clustered hosting is a type of web hosting that spreads the load of hosting across multiple physical machines, or node. This means that you can upload a site to the cluster, make a change to the site, and it’s totally transparent to the end user. That the site might have been moved from one server to another within the cluster.

Google Cluster
Amenable to extensive parallelization, Google’s web search application lets different queries run on different processors by partitioning the overall index, also lets a single query use multiple processors. To handle this workload, Google’s architecture uses a cluster of more than 15,000 commodity-class PC servers with fault-tolerant software. The servers are based on Intel Architecture processors and linked using Intel® PRO/100 Ethernet adapters. The servers, which are equipped with 256MB to 1GB of RAM, run Linux operating system and have a suite of custombuilt applications. This architecture makes it much more cost effective than a comparable system built out of a smaller number of high-end servers. It achieves superior performance at a fraction of the cost of a system built from fewer, but more expensive, high-end servers.

The Google cluster is further divided into smaller clusters that is made up of a few thousand machines, and is geographically distributed to protect Google against catastrophic data centric failures. Each of these smaller clusters will be assigned queries based on the user’s geographic proximity to it. This is a form of load balancing, and it helps to minimize the round-trip time for a query; hence, giving greater response time. The Google software architecture has been enhanced to support processing of queries raised by external third party applications, which is achieved by providing Web service interface as discussed below.

VI. Guided Google Architecture
An integrated architecture of Guided Google along with its interaction between the user and the Google server is illustrated in Figure. The users access the Guided Google site, which is developed using JSP (JavaServer Pages) and hosted on a Tomcat server. The Tomcat server acts as both a stand-alone web server and servlet container. As seen in Figure 1, Guided Google will send and receive information from the Google.com server via SOAP. As this process is hidden, the users access it just like accessing any HTML-based web site.

VII. DESIGN AND IMPLEMENTATION
Guided Google is a web based guided search tool that is developed using Forte for Java and coded in JSP. As mentioned earlier, this tool is developed based on the assumption that an average web user makes searches based on imprecise query keywords or sentences, which in turn leads to unnecessary or inaccurate results. In this work, we
demonstrate that with a simple tweak or manipulation of existing Google functions, we can help guide users to get better search results.

The Guided Google functionalities can be grouped into two categories. The first one is for combinatorial keyword searching, and the second one is for keyword searching based on the servers hosting them. Each of these will be discussed in detail later in this section. A simple illustration of how this application is structured is discussed below.

The layered organisation of Guided Google software module is shown in Figure 4. There is a main page (index.jsp) that is used to interface with all the other .jsp files. The normal and combinatorial keyword search functions are supported by the normalSearch.jsp program file. The host based search function, on the other hand, is supported by serverBased.jsp and serverBased2.jsp program files. The first module isolates the first five unique domains from the results and the second module refines the search based on the selected host and displays the results in an expandable tree menu. Please refer to Figures 11 and 12 for a better illustration of how this works.

Other than that, several java bean components are used to support tasks that are needed to be performed multiple times. GoogleConnectionBean.java is used to establish the initial connection to the Google server. PermuterBean.java is used to calculate the permutation for the keywords when the user selects the combinatorial option. And ServerBean.java is used to store the URLs of unique domains which are obtained after the execution of the serverBased.jsp file. This allows serverBased2.jsp to access those URLs when it is executed to refine the search, based on the hosts.

To simplify the deployment, we package the entire application modules in a WAR (Web Archive) file. The hosting environment needs to be running a web server such as Tomcat 3.3 that supports JSP. The instructions for deployment of gg.war and googleapi.jar packages on the hosting server are specified in Appendix A. However, different web servers or even different versions of Tomcat may require different deployment steps.

The remainder of this section describes the implementation and operation flow of the two functions supported in Guided Google. First of all, the entry page (as illustrated in Figure 6) consists of a query text box, a license key text box, and options as to whether the user wants to use the combinatorial or search by host functions. The users are encouraged to key in their own license key if they have registered with Google. But if they do not have one, leaving the text box ‘null’ would allow them to use our license key by default, which is currently limited to Google’s stipulation of 1000 queries a day.

If the user submits a query keyword without selecting any of the two optional functions, Guided Google will perform a normal search (see Figure 7), as how Google.com (see Figure 8) would have done. But when one of the functions is selected, the user can get better search results as demonstrated in the next section.

![Fig 6: Guided Google entry page](image)

**Combinatorial Search**

It is widely known that the arrangement of keywords in the search query makes a difference in the search results of the Google Search Engine. Based on the assumption that novice web users are not familiar with the construction of effective keywords for their search queries, Guided Google provides a function that will automatically calculate the permutation and make different combinations of the keywords used. For example, if the users were to key in the term “grid computing”, it will automatically generate two combinations, which are “grid computing” and “computing grid”. The results of these two queries are very different (see Figure 9 and Figure 10). It should be noted that the quotes in search queries do make a difference. In Google search, the words in quotes mean that they have to occur in that particular order, in the search results. In accordance with this, if the search query is placed in quotes, the result of the combinations will also be reflected in quotes as demonstrated in the next section.

**Search by Host**

This function is slightly more complex in nature as it allows the users to search for a keyword based on the first five hosts that are hosting it. By default, Guided Google is configured to return only the first five results of a search. It is possible that there may be fewer than five hosts returned by the search. This can also happen when Guided Google automatically removes identical hosts from the list.

The ‘arrow’ sign shown beside each link in Figure 11 indicates that it is expandable. Clicking on the arrow triggers another query, but that will be restricted to query search within the selected host’s web pages. This can be achieved by manipulating Google’s query syntax (e.g., “site:some_host_domain”). There are many other query syntaxes that can be very useful, depending on how the programmer manipulates them. Here, we are just showing a
simple application of this in action. However, this function is not always guaranteed to give more accurate results. The main purpose of this function is to allow the user to have a different perspective of searching. It gives a lateral way of looking at the results. With the search results obtained, the users will hopefully get a better idea of what they are searching for, and hence learn to issue more accurate query keywords.

Other Google query syntax such as “inurl:”, “allintext:” and “allinlinks:” can also be included. Each one of these can be manipulated in various ways to produce different outcomes.

VIII. EVALUATION

This section focuses on evaluating the search performance of Guided Google compared to the default Google. For simplicity, we use a common search term, which is “grid computing”. This term is used throughout so that the different results can be compared.

Normal Search

In this case, “grid computing” (note that the keyword is in quotes) is submitted as the query keyword, without any options chosen, as shown in Figure 7. The results of the search are the same as a normal search done on the Google.com site (see Figure 8), except for results 3 and 5. They seem to have a different rank in the actual Google search site. Theoretically, the ranking should be the same, since we have not done any manipulation to the search. Nevertheless, the results are still the same, even though the listing sequence is slightly different in this case.

Combinatorial Search

In a combinatorial search, the term “grid computing” is passed through a permutation function which gives all unique combinations of it. In this case, it will also produce “computing grid”. Permutation increases in the order of factorial, so 3 keywords would mean a combination of 6 different search queries. A combinatorial search with larger number of keywords leads to exhaustive search results and it becomes hard to find relevant resources. In such situation, a normal search would have produced very accurate search results.

It should be noted that the combinatorial search function need to be used when the users are interested in more variety of results. When the users are not too sure of what they are looking for—for example, “ski holiday” or “jet boat”—they may find it worth searching in different sequences. This function automates the ‘combinations portion’ and searches for all the combinations. Figures 9 and 10 illustrate the results of two combinatorial searches for “grid computing”. Users should use a normal search, and save them the trouble of wading through the different combination of results, if they are already sure of what they are looking for.

It can be seen in Figure 9, that the search results are the same as the normal search. Nothing has been changed there that would cause a difference in the results. Figure 10, however, shows a whole different set of returns. Although results 2 to 5 may not be in the top ten returns of the original search, they are still relevant search results. We have made a comparison of the This actually gives the users a wider perspective of their search result options because users do not usually search through pages 5 and above, since most of them consider these pages irrelevant to their search. This function gives the users a chance to sample search results from different ranking levels.
In searching based on hosts, the term “grid computing” is submitted to Google.com and the results are retrieved. Then the first 5 unique domains will be isolated and displayed to the user, as shown in Figure 11. Clicking on the arrows by the domains will submit another query to the Google site, (how this works was explained in the previous section) further refining the search based on the selected domain as illustrated in Figure 12.

As mentioned in the previous section, this search is not guaranteed to give better search results. But we believe that the results are just as accurate, if not better than the normal search. This is because we are picking the first 5 unique domains that are returned in the first phase of the search. Refining the search based on these domains (this is the second phase of the search) produces results that are not too far off from being accurate. In fact, searching these domains leads to more specialized search results, since we are refining the search on a domain that has already been given a high rank by Google.

The illustration in Figure 11 shows that the hosts returned are the same as the domains of the first five results of the normal search. Figure 12 shows a refined search on the “www.gridcomputingplanet.com” domain. Apart from the first result, the other four are different from the results of a normal search. These four, however, are more specific on the “grid computing” topic, since they were obtained from a single domain. The users are now able to see much more specialized results.

As illustrated, the results returned by the ‘Combinatorial keyword searching’ and ‘Searching by Hosts’ are very different. Even within combinatorial search, Figures 9 and 10 have highlighted the differences in results just by swapping the combination of search queries. Searching by Hosts, though it returns results that are not the highest in rank, is useful when the users want to search for a term that is related to a certain host only.
IX. CONCLUSION

The goal of this proposal is to shed some light on the reasons that make building meta search engines, especially large scale meta search engines, difficult & challenging. This paper proposes a robust meta search engine which can communicate to heterogeneous platform using advanced web service techniques. As much progress has been made in advanced meta search engine technology, several challenges need to be addressed before truly large scale meta search engine can be effectively built & managed. This paper proposed a guided meta-search engine, called “Guided Google”, which provides meta-search capability developed using the Google Web Services. It guides and allows the user to view the search results with different perspectives. This is achieved through simple manipulation and automation of the existing Google functions. Our meta-search engine supports search based on “combinatorial keywords” and “search by hosts”. A detailed evaluation demonstrates how one can harness the capability of Google cluster architecture through its programmable Web services by creating advanced search features at a third party user application level.

REFERENCES


Appendix A: Guided Google Deployment

Instructions (on Windows)

1. Download the appropriate jakarta-tomcat-<version> binary archive file.
2. Expand the archive into some directory (say C:\). This should create a new subdirectory named "jakarta-tomcat-<version>".
3. Set the environment variable JAVA_HOME to point to the root directory of your JDK hierarchy. set JAVA_HOME=c:/jdk1.3.1
   PATH=%JAVA_HOME%;bin;%PATH%
4. Set the TOMCAT_HOME environment variable. set TOMCAT_HOME=c:/jakarta-tomcat-<version>
5. Place the gg.war file into the webapps directory (c:\jakarta-tomcat-3.3.1\webapps). When Tomcat is started, it will automatically expand and deploy that file.
6. Place the googleapi.jar file into the tomcat library folder. (c:\jakarta-tomcat-3.3.1\lib\apps)
7. Start Tomcat. This is done by running the bin\startup.bat file in the bin directory of Tomcat.
8. To load the main page, type http://localhost:8080/gg/index.jsp into the browser window.
9. To stop Tomcat, just run the bin\shutdown.bat file.