Migration of Search Engine Process into the Cloud

Akassh A Mishra, Chinmay Kamat
Dept of Computer Engineering
Sardar Patel Institute of Technology, Andheri
Mumbai, India

ABSTRACT
The basic elements of a search engine process are crawling, storage, indexing and ranking algorithms. The current approach towards design of search engines is monolithic and infrastructure-heavy. In present paper, we discuss a modularized and a light-weight approach towards the search engine process using the merits of cloud computing. The cloud-based search architecture enables customization of the search process as per requirements of the stakeholders. The new approach provides a cloud-based platform for low-cost, effective and personalized search models. This approach overcomes the pitfalls in traditional Search Engine Optimization and has tremendous scope for future development.

General Terms
Search Engine Optimization, Cloud Computing, Security, Pattern Recognition.

Keywords
search engine process optimization; crawling; indexing; search algorithms; cloud computing; cloud architecture

1. INTRODUCTION
In the world of Web 2.0, the adage "content is king" remains a prevailing theme. With seemingly endless content available online, the "findability" of content becomes a key factor. Search engines are the primary tools people use to find information on the web. Searches are performed using keywords. When you enter a keyword or phrase, the search engine finds matching web pages and show you a search engine results page (SERP) with recommended web pages listed and sorted by relevance. Though it used to be difficult to obtain diverse content, there are now seemingly endless options competing for an audience's attention. As a result, search engines have gained popularity by helping users quickly find and filter the information they want. Google, Yahoo, Bing and Ask have emerged as the most popular search engines in the recent past. Most users have formed searching habits to gain the information they need, as there is no single website that caters to all their needs. Google logs an estimated 2 billion searches per day and an estimated 300 million users use the search facility provided by Google on a daily basis. This number is set to rise in the future.

Cloud computing is an upcoming technology. The only thing that is restraining cloud for globalizing is the issue of security. The issue is one cannot determine how the data flows once it is sent to cloud. These issues will be solved in near future and cloud will become omnipresent technology.

With this background in mind, the presented paper aims at marrying the search engine optimization and ranking process with the benefits of cloud technology. The first part of the paper explains the architecture and working of conventional search engines along with the various ranking parameters. In the latter half, we propose a cloud-based architecture to migrate the search engine process into the cloud. The advantages and drawbacks of both approaches – traditional as well as cloud based are also mentioned

2. TRADITIONAL SEARCH ENGINE

2.1 Need of Search Engine
In the evolved internet, traffic redirection to any Web Product is primarily via the popular search engines. The products on the internet serve all kinds of consumers without any preference to the internet savvy population. Hence the importance of gaining visibility via the Search Engines is extremely critical to ensure that the product is able to capitalize of a wider consumer base. A study by China Internet Network Information Centre in 2009 states that the number of internet users in China stands at 298 million with annual increase of 59.49 million people and an annual growth rate of 34%. A study by ITU states that 82.5% of the British population had access to the internet in 2010 as compared to 26.2% in 2000. [1] About 70% of the website flow rate comes from major search engines through which users acquire various kinds of information regarding their life, study and work. It is obviously that search engines have become an integral part of users' life while enterprises have found the marketing role of search engine during their network marketing, too.

2.2 Components of Search Engine
The study of various components[6] that constitute a search engine help us understand the various processes involved in search engine rankings and the inter-dependency between them. In the latter part of the presented paper, we aim to replicate the same process using the advancement in cloud computing technology.

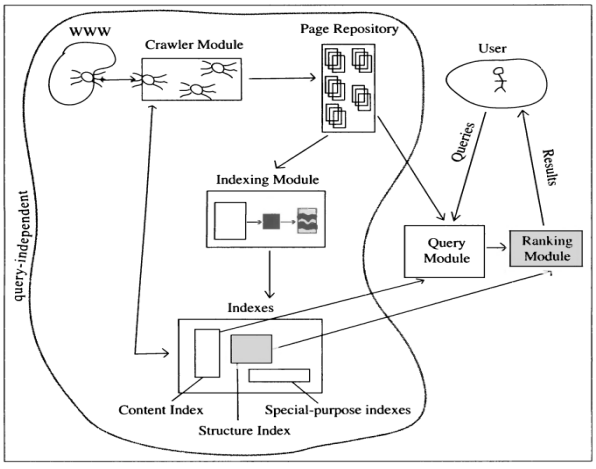

Figure 1 Components of Search Engine

2.2.1 Crawler Module

The web document collection lives in a cyber warehouse - a virtual entity that is not limited by geographical constraints and can grow without limit. Search engines must do the data collection and categorization tasks on their own. Hence, all web search engines have a crawler module. This module contains the software that collects and categorizes the web's documents. The crawling software creates virtual robots, called spiders that constantly scour the Web gathering new information and web pages and returning to store them in a central repository.

2.2.2 Page Repository

The spiders return with new web pages, which are temporarily stored as full, complete web pages in the page repository. The new pages remain in the repository until they are sent to the indexing module, where their vital information, such as keywords and Meta tags, is stripped to create a compressed version of the page.

2.2.3 Indexing Module

The indexing module takes each new uncompressed page and extracts only the vital descriptors, creating a compressed description of the page that is stored in various indexes. The uncompressed page is then removed from the page repository.

2.2.4 Indexes

The indexes hold the valuable compressed information for each webpage. In the content index; content such as keyword, title, and anchor text for each webpage, is stored in a compressed form using an inverted file structure. Further valuable information regarding the hyperlink structure of pages is stored in compressed form in the structure index. The crawler module sometimes accesses the structure index to find uncrawled pages. Special purpose indexes, such as indexes on pdf files or ppt files are maintained to satisfy specialized task queries.

The four modules mentioned above namely crawler, page repository, indexing module and indexes are said to be query-independent modules. These modules operate independently of users and their queries. In contrast the query module and the ranking module are dependent on the user query and are said to be query-dependent modules.

2.3 Query Dependent Modules

2.3.1 Query Module

The query module converts a user's natural language query into a language that the search system can understand and consults the various indexes in order to answer the query. For example, the query module consults the content index and its inverted file to find which pages use the query terms. These pages are called the relevant pages. Then the query module passes the set of relevant pages to the ranking module.

2.3.2 Ranking Module

The ranking module takes the set of relevant pages and ranks them according to some criterion. The outcome is an ordered list of web pages such that the pages near the top of the list are most likely to be what the user desires. The ranking module is the most important component of the search process because the output of the query module often results in thousands of relevant pages. The ordered list filters the less relevant pages to the bottom, making the list of pages more manageable for the user. This ranking which carries valuable, discriminatory power is arrived at by combining two scores, the content score (derived using on-site parameters) and the popularity score (derived using off-site factors).

3. RANKING

3.1 Search Engine Ranking Parameters

The ranking module filters relevant pages and ranks them in order of the overall score assigned to them. The ranking parameters are those critical points on which the ranking score is assigned. Ranking Parameters are broadly classified as on-site and off-site parameters.

On-site parameters refer to changes made in the developmental code of the site to help it rank better in Search Engines. Some of the major on-site parameters include:

- URL of the website
- Title of the website
- Description meta tag
- Keyword meta tag
- Density of keyword on the document
- Proximity of keywords to each other
- Keywords using H1 tag and other header tags
- Keywords using alt tags for graphics
- Keywords using Bold and Italics
- HTML validation at (typically at www.w3.org)
- Uniqueness of content as compared to other websites
- Spelling and Grammar

Off-site parameters refer to the interaction of the website with other websites and search engines. Google assigns strength to the site in the form of "Page Rank" on the basis of off-site parameters. Some of the major off-site parameters include:

- Number of websites linking back to the web site
- Page Rank of the web site
- Anchor texts of the links pointing to the web site
- The rate at which inbound links have been accumulated
- Number and quality of directories a page is listed in (e.g.: DMOZ or Yahoo)
- IP address and relationship to other IP addresses
- Current age of the domain

3.2 Ranking Algorithm – Page Rank

The Page Rank algorithm which is named after one of Google's founders Larry Page is used to identify the importance of web pages divided in the range 1-10, where 10 represents full score (Higher PR value) and 1 represents lower PR value. The Page Rank of a page Pi, denoted r(Pi), is the sum of the Page Ranks of all pages pointing into Pi

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|}, \qquad (1)$$

Where BP is the set of pages pointing into Pi and | Pj | is the number of out links from page Pj.
Suppose page A has its pages- p_1, $p_2 \ldots p_n$ linked to it, then page A has its PR value as follows:

$$PR(A)=(1-d)+d\{PR(p_1)/C(p_1)+ +PR(p_n)/C(p_n)\} \quad (2)$$

Where, d=damping factor PR=Page Rank and C=count of outgoing links from a page.

Based on the above factors, inferences for better off-page score are as follows:

- Higher the number of inbound links, better for ranking.
- Links from sites pertaining to the same topic as your site have more weight-age.
- Links from sites having low outbound links are more beneficial.
- Links from sites having authority (High PR) contribute more to the PR of your site.
- Links should be built gradually over a period of time.

4. PROPOSED INTEGRATION OF SEARCH ENGINE WITH CLOUD ARCHITECTURE

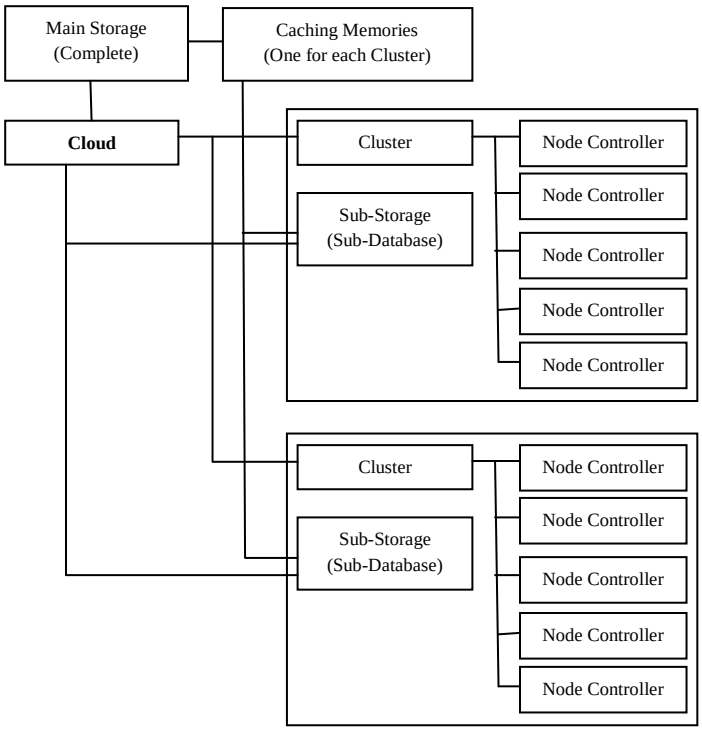

Figure 2. Proposed Architecture

The present paper proposes an architecture which will help in further optimization of search engine based on cloud architecture; it also contains few modifications from current cloud architecture. The components in proposed architecture are Cloud Controller, Main-Storage, Fast Cache Memories, Cluster Controllers, Sub-Storages and Node Controllers. The Descriptions of the above named components are as follows:

4.1 Cloud Controller

Cloud Controller forms an integral part of proposed architecture. It is the brain of the cloud Search engine architecture. Various decisions regarding data distribution, data buffering, authentication, etc… is taken care by this cloud controller. It also acts as an Organizer and Synchronizer when it comes to inter-cloud communication. For good, efficient and effective cloud architecture careful design of cloud controller is of utmost importance.

4.2 Main Storage

All the links and Data that is accumulated with the help of crawler will be stored in the Main Storage. Various redundancy techniques can be applied on this main storage so as to protect the data. Main storage is analogous to Secondary storage in Computer Architecture. Main Storage can be considered a wealth that a Cloud search engine possesses.

4.3 Fast Cache Memories

Fast Cache Memories acts like a mediator between the Sub-storage and Main Storage. Its size lies in between the size of main storage and Sub Storage. For each cluster controller there will be a Fast Cache Memory belonging to that cluster controller and its size would be more than that of sub-storage present in that cluster. As the name suggest it will be useful for caching all the recently used data by the node in a particular cluster. This will help in increasing the efficiency of the cloud search engine architecture.

4.4 Cluster Controller

Cluster Controller is the controller of all the node controllers present in that particular cluster. Cluster controller is responsible for managing all the node controllers and also manages the distribution of data within all the node controllers so as there is a dividend belonging to a particular node controller. Cluster Controller plays an important role in making the cloud searching architecture work as a layered architecture. Cluster Controller is the one of the significant component of cloud search engine architecture.

4.5 Sub-Storage

Sub-storage is a customized part of main storage. It is a part of Main Storage that is accessible to a particular cluster and is invisible to other clusters. Data present in Sub-storage is uniformly distributed over the node controllers by cluster controller which leads to an efficient data handling and gives result in an optimum time frame.

4.6 Node Controller

Node Controllers are the Controllers which work in the lowest controlling level. It's responsible for controlling the distribution of data among the nodes and also responsible for mapping for distribution to a particular node on low level, Node Controller forms the basic entity of cloud search engine.

5. WORKING OF PROPOSED CLOUD SEARCH ENGINE

Working of Search Engine is divided into two part, query dependent module and query independent module.

5.1 Query Independent Module

Query Independent Module consists of Crawler, indexer and Repository. Query Independent module does not depend upon the user but is an ongoing continuous process. Each Component of query independent module works inside the cloud controller. Except for repository all the other components of Query Independent Module lies within the Cloud controller. All the data which crawler brings gets stored in the repository which is present in the main storage. Indexing module will be present in cloud controller while Indexes will be present in Main Storage.

5.2 Query Dependent Module

Query Dependent module consists of Query Module and a Ranking Module. Each Component on query dependent module forms a Part of Cluster. Cluster controller contains query module with all the ranking algorithms and parameters. While ranked pages are stored in sub-storage according to the personalization of data present in main storage.

5.3 Actual working of Cloud Search Engine

Whenever user fires a query its goes to a particular cluster depending on the type and data set required by the query and that is determined by cloud controller. Whenever a data match is found then the query result is returned immediately to cloud controller. This form of layered architecture helps in efficiently managing the search engine.

6. BENEFITS AND DRAWBACKS

6.1 Benefits of Cloud Search Engine

- When used appropriately, a layered design can lessen the overall impact of changes to the application.
- Allows you to modify a component without disturbing the next one
- Design scalable and maintainable rapidly
- Increase security level of an application

6.2 Drawbacks of Cloud Search Engine

- Security is still an issue to use Cloud Architecture.
- Inter-Cloud Communication is still not possible.
- Lack of Robustness and Platform Dependent.
- Open Standard still an issue in Cloud Computing.

7. COMPARATIVE STUDY OF EXISTING ARCHITECTURE WITH PROPOSED ONE

Cloud computing has been a disruptive technology, which has challenged the way we think about data and services. A search engine architecture based in the cloud will serve to challenge the perspective through which we look at search engines.

Search engines, especially web based search engines, are monoliths. The internal architecture for commercial search engines is varied, each search engine having their own non-standardized interfaces and architecture. As a result, the search engine sphere is dominated by a top few engines and all extensions of the search facility are through the APIs provided by them. There is no scope for manipulating or tweaking the process to suit individual requirement, or using only a specific set of modules. For an individual or organization looking to build their own searching process based on their own criteria, or looking to use personalized search as a service, the options in the broader market are extremely limited.

Search architecture in cloud would add make the search engine process light-weight. As computation and data would be stored in the cloud, the end user would be decoupled from the processing requirements. The architecture would also foster modularity and allow the users to customize the search process to suit their requirements. The user could choose to not use certain modules (say) fast cache memories or sub-storage depending on the needs of his search process. The customized search could also make use of the query dependent module only, working on a dataset which has been previously collected using the query dependent module. Specific search engines targeting a micro-domain could be employed using indexes for that particular domain. And finally, value added cloud based search engines with could also be provided to the consumers as PaaS (platform-as-a-service). Thus, a whole lot of variations in the search process are made possible with the new architecture.

Table 1. Comparative Study of Architectures

Advantages over existing architectures	**Disadvantages**
Modularity	Lack of standards
Flexibility	Platform dependency
Customization	Security Issues
Availability on a service basis	Inter Cloud Operations
Scalability	
Cost efficient	

8. FUTURE SCOPE OF PROPOSED ARCHITECTURE

Security is the only aspect that is holding back the cloud computing to become the need of today. But, this issues is solvable and cloud computing will eventually result in one of the global technology of future.

Monolithic design of search engine has some drawbacks which are overcome by Layered architecture which are provided by cloud. So, changes in any module or component can be done without disturbing the other components.

Personalization and customization of search result is easily possible on each cluster components giving a better scope of partial data sharing in future.

Due to the above factors we can say that the proposed architecture has tremendous application and bright future scope.

9. CONCLUSION

Thus, in the present paper we have proposed an architecture which gives a better approach to search engines by converting them from Monolithic Architecture to Layered Architecture. Merger of Cloud computing technology with search engine optimization gives a better understand and helps in distribution and personalization of data. Future scope of proposed architecture has already been discussed and also, in near future cloud computing will be the most useful and admired technology. So, Merger of Search Engine with Cloud Architecture will be a significant step in future.

10. REFERENCES

[1] Mo Yunfeng, "A Study on Tactics for Corporate Website Development Aiming at Search Engine Optimization", 2010 Second International Workshop on Education Technology and Computer Science.

[2] Chengling Zhao, Jiaojiao Lu, Fengfeng Duan," Application and Research of SEO in the Development of Web2.0 Site", 2009 Second International Symposium on Knowledge Acquisition and Modeling.

[3] Sean A. Golliher, "Search Engine Ranking Variables and Algorithms", SEMJ.ORG Volume1 Supplemental Issue August 2008.

[4] Nan Yang, "Strategy of the Search Engine Marketing for High Tech Enterprises", 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)

[5] Google, Google's Search Engine Optimization Starter Guide, 2008.

[6] Langville A., Meyer C., Google's Page Rank and Beyond, Princeton University Press,2006

[7] "Service-Oriented Computing and Cloud Computing: Challenges and Opportunities". IEEE Internet Computing.

[8] Buyya, Rajkumar; Chee Shin Yeo, Srikumar Venugopal "*Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*"

[9] Danielson, Krissi "Distinguishing Cloud Computing from Utility Computing"

[10] Bernstein, David; Ludvigson, Erik; Sankar, Krishna; Diamond, Steve; Morrow, Monique *Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability*. IEEE Computer Society. pp. 328–336