I.J. Information Technology and Computer Science, **2013, 10, 85-91**
Published Online September 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2013.10.09

Web Crawler Based on Mobile Agent and Java Aglets

Md. Abu Kausar
Dept. of Computer & System Sciences, Jaipur National University, Jaipur, India
E-mail: kausar4u@gmail.com

V. S. Dhaka
Dept. of Computer & System Sciences, Jaipur National University, Jaipur, India
E-mail: vijaypal.dhaka@gmail.com

Sanjeev Kumar Singh
Dept. of Mathematics, Galgotias University, Gr. Noida, India
E-mail: sksingh8@gmail.com

***Abstract*—** With the huge growth of the Internet, many web pages are available online. Search engines use web crawlers to collect these web pages from World Wide Web for the purpose of storage and indexing. Basically Web Crawler is a program, which finds information from the World Wide Web in a systematic and automated manner. This network load farther will be reduced by using mobile agents.

The proposed approach uses mobile agents to crawl the pages. A mobile agent is not bound to the system in which it starts execution. It has the unique ability to transfer itself from one system in a network to another system. The main advantages of web crawler based on Mobile Agents are that the analysis part of the crawling process is done locally rather than remote side. This drastically reduces network load and traffic which can improve the performance and efficiency of the whole crawling process.

***Index Terms*—** World Wide Web, Search Engine, Mobile Crawler, Aglets, Web Crawler, Mobile Agent

I. Introduction

The web is very dynamic and 53% of its contents change daily [11], to maintain the up to date pages in the group, a crawler needs to revisit the websites many times. Due to more revisit, the property like CPU cycles, disk space, and network bandwidth etc., it will become overloaded and due to this type of overloads sometime a web site may collapse. Study [12] report that the current web crawlers have downloaded and indexed billion of pages and about 41% of current internet traffic and bandwidth spending is due to the web crawlers. Nevertheless, the maximum web scope of any well-known search engine is not more than 16% of the current web size.

Using mobile agent i.e. mobile crawlers, the method of selection and filtration of web pages can be done at servers rather than search engine side which can reduce network load caused by the web crawlers [9].

Search engines use crawlers that visit a Website, read the information on the actual site, read the web site's meta tags and also follow the links that the web site connects to performing indexing on all linked sites as well. The crawler returns all information back to a central repository, where the data is indexed. The crawler periodically returns to the web sites to check for any information that has been changed.

The paper is organized as follows: the structure and working of Search Engine is detailed in section 2, related work regarding search engine are presented in section 3, section 4 describes the Mobile Agent Based Crawling, Aglet Life Cycle Model described in section 5, section 6 describes the benefits of Mobile Agent in crawling, section 7 describes the whole working of proposed system, and conclusions are made in the last section 8.

II. Structure and Working of Search Engine

The basic structure of crawler based search engine is shown in Fig. 1. The main steps in any search engine are:

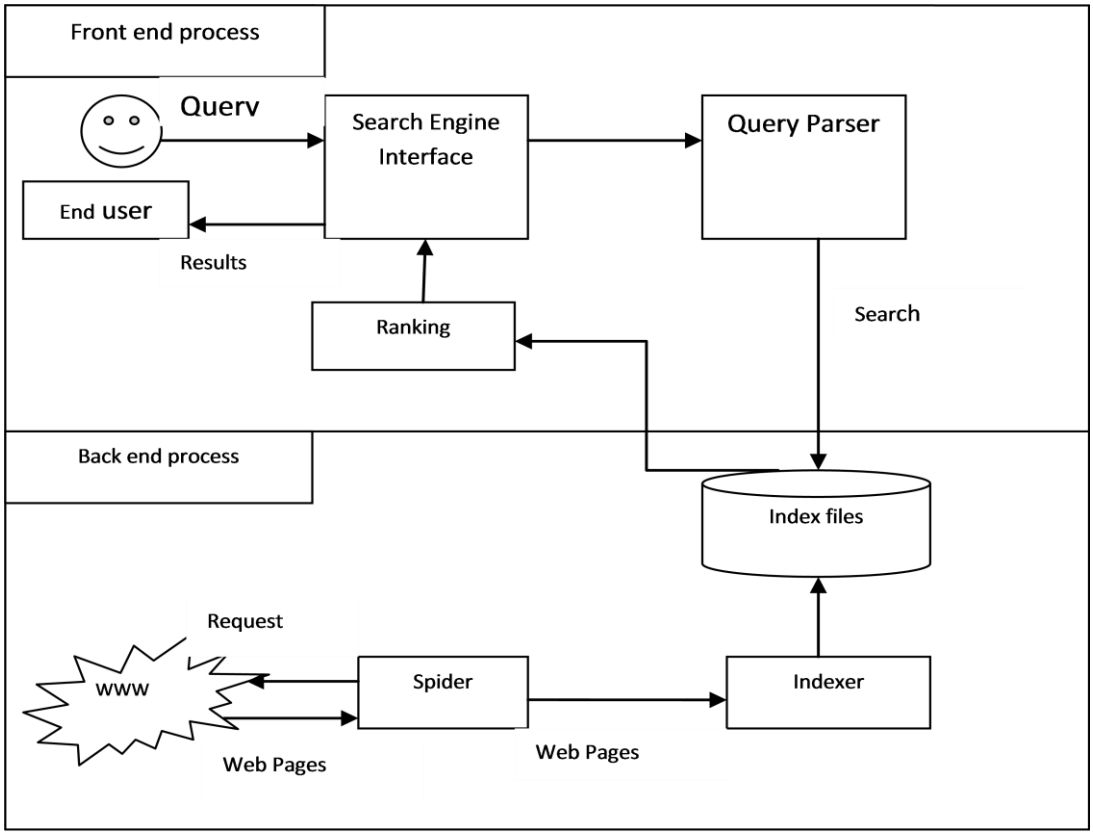

Fig. 1: Working steps of search engine

Every search engine depends on a crawler to provide the grist for its operation. This operation is performed by special software, called Crawlers. Web crawler is a program/software or programmed script that browses the WWW in a systematic, automated manner on the search engine's behalf. The programs are given a starting set of URLs called seed URL, whose pages they retrieve from the Web. The web crawler extracts URLs appearing in the retrieved pages, and provides this information to the crawler control module. This module determines which links to visit next, and feeds the links to visit back to the crawlers.

2.1 Maintaining Repository

All the data of the search engine is stored in a repository as shown in the figure 1. All the searching is performed through that database and it needs to be updated frequently. During a crawling process, and after completing crawling process, search engines must store all the new useful pages that they have retrieved.

2.2 Indexing

Once the web pages are stored in the repository, the next job of search engine is to make index of stored data. The indexer module extracts all the words from every web page, and records the URL where each word occurred. The result is a usually very large that can provide all the URLs that point to web pages where a given word occurs.

2.3 Querying

This module deals with the user queries. The responsibility of query engine module is for receiving and filling search requests from users. The search engine relies deeply on the indexes, and sometimes on the page repository.

2.4 Ranking

Since the user query results in a large number of results, it is the work of the search engine to display the most suitable results to the user. To do this efficient searching, the ranking of the results are performed. The ranking module has the task of sorting the results such that results near the top are the most likely ones to be what the user is looking for. Once the ranking is done by the Ranking module, the final results are displayed to the user.

III. Related Work

Search engine has three major parts named as indexer, crawler and query engine. Web crawlers are programs that traverse the web on the behalf search engine, and

follow links to achieve different web pages to download them. Beginning with a set URLs called seed url, crawler will extract URLs from retrieved pages, and store pages in a repository. The downloaded web pages are indexed and stored in the search engine repository. This continuous updating of repository makes a search engine more consistent source of relevant and updated information. Details of crawler are discussed by [14].

The crawler has to deal with two main responsibilities i.e. downloading the new web pages and remaining the earlier downloaded web pages fresh. However, freshness can only be assured by simply revisiting all the web pages more often without placing unnecessary load on the internet. With the available bandwidth which is neither infinite nor free, it will become necessary to crawl the web pages in a way that is not only scalable but also efficient, if several reasonable measure of quality or freshness is to be continued.

IV. Mobile Agent Based Crawling

A mobile agent is a self-directed program that acts on behalf of its owner. According to its path, it visits hosts which are linked together via a network. A mobile agent is created, sent, finally received and evaluated in its owner's home framework. At a visited host, a mobile agent is executed in a working context. Building of web index using mobile agents is called as mobile crawler. The ability of a mobile crawler is to migrate to the web server before the actual crawling process is started on that web server. Mobile crawlers are capable to go to the resource which needs to be accessed in order to take benefit of local data access. After accessing a resource, mobile crawlers shifted to the next server, carrying the web crawling result in the memory. Mobile crawlers are transferred to the site of the source where the data is located in order to filter out any unnecessary data locally before transferring it back to the search engine [15]. These mobile crawlers can decrease the network load caused by the crawlers by falling the amount of data transported over the network. Using this approach filter those web pages that are not modified using mobile crawlers but retrieves only those web pages from the remote server that are really modified and perform the filtering of not modified web pages without downloading the pages. This mobile crawlers move to the web servers, and perform the downloading of web documents, processing, and extraction of keywords and after compressing, transmit the results back to the central search engine.

The role of mobile crawlers and data retrieval architecture as established by mobile crawlers given in Fig. 2.

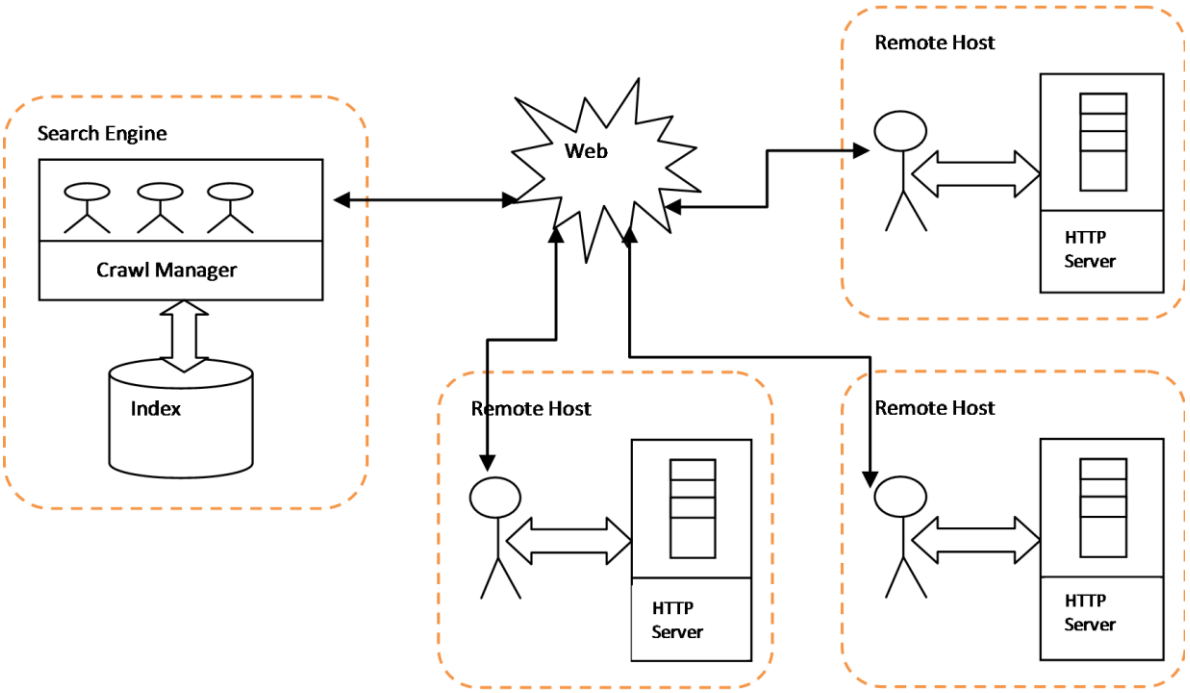

Fig. 2: Mobile Based Crawling Approach

4.1 Aglet Architecture

An Aglet is a Java-based mobile agent system [5]. The Aglets architecture consists of two layers and APIs that define interfaces for accessing their functions. The Aglets runtime layer is the implementation of the Aglet API, and defines the activities of the API components, such as AgletProxy and AgletContext. It provides the fundamental functions for aglet to be created, managed, and dispatched to remote hosts. The communication layer is mainly responsible for transferring a serialized agent to a destination and receiving it.

4.1.1 Aglets Runtime Layer

Aglets runtime layer applies Aglets interfaces such as AgletProxy and AgletContext. It also includes of a core framework and subcomponents. The core framework offers the following mechanisms which is essential for aglet execution:

- Serialization and De-serialization of aglets
- Class loading and transfer
- Reference management and garbage collection

The subcomponents are designed to be extensible and customizable because these services may vary depending on environments.

PersistenceManager: The PersistenceManager is liable for storing the serialized agent, consisting of the aglet's code and state into a constant medium like hard disk.

CacheManager: The CacheManager is responsible for maintaining the bytecode used by the aglet and it transfer when the aglet moves to the next destination, the CacheManager caches all bytecode even after the matching class has been defined.

SecurityManager: The SecurityManager is responsible for protecting aglet platforms and aglets from malicious entities. It catches every security-sensitive operation and verifies whether the caller is permitted to perform it. There is only one instance of SecurityManager in the system which cannot be altered once it is installed.

4.1.2 Communication Layer

The Aglets runtime has no communication mechanism for transferring the serialized data of an aglet to destinations. The Aglets runtime employs the communication API that abstracts the communication between agent systems. This API defines techniques for creating and transferring agents, tracking agents, and managing agents in an agent system and protocol-independent way. The current Aglets use the Agent Transfer Protocol (ATP) as the default implementation of the communication layer. ATP is modeled on the HTTP protocol, and is an application level protocol for transmission of mobile agents. To facilitate remote communication between agents, ATP also holds message-passing.

V. Aglet Life Cycle Model

Aglet is a library written in Java which was introduced by IBM to support the development of mobile agent. The execution environment within which Aglets are executed is referred to as the Aglet's Context and is responsible for enforcing the security restrictions of the mobile agent.

The different states in Aglet life cycle [6] are as follows:

- **Created:** a brand new aglet is born,- its state is initialized, its main thread starts executing
- **Cloned:** a twin aglet is born - the current state of the original is duplicated in the clone
- **Dispatched:** an aglet travels to a new host - the state goes with it
- **Retracted:** an aglet, previously dispatched, is brought back from a remote host - its state comes back with it
- **Deactivated:** an aglet is put to sleep - its state is stored on a disk somewhere
- **Activated:** a deactivated aglet is brought back to life - its state is restored from disk
- **Disposed:** an aglet dies - its state is lost forever

The Aglet Life cycle state diagram is shown in Fig. 3.

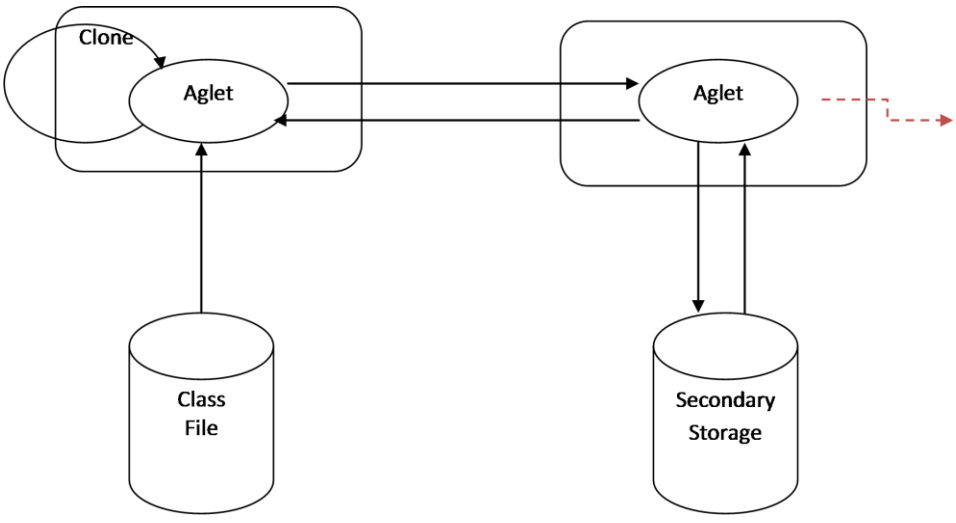

Fig. 3: Aglet Life Cycle

VI. Benefits of using Mobile Agent

1. Reduce the Network Load: Due to HTTP request or response model, downloading the contents from a Web server involves major overhead due to request messages which have to be sent for each Web page individually. Using a mobile crawler we can reduce the HTTP overhead by transferring the crawler to the source of the data. The web crawlers then issue all HTTP requests locally with respect to the HTTP server. This approach still needs one HTTP request per web document but there is no need to send out these requests over the network anymore. A mobile crawler therefore saves bandwidth by reducing Web traffic caused by HTTP requests.
2. Remote Page Selection: Traditional crawlers implement the data shipping approach of database systems because they download the whole database before they can issue queries to identify the relevant portion. In contrast to this, mobile crawlers implement the query shipping approach of database systems because all the information required to identify the related data portion is moved directly to the data source together with the mobile crawler. After the query executed remotely, only the query result is moved over the network and can be used to set up the desired index without requiring any further analysis.
3. Remote Page Filtering: Remote page filtering expands the idea of remote page selection to the contents of a Web page. The goal behind remote page filtering will allow the crawler to manage the granularity of the data it retrieves. Depending on the ratio of relevant to irrelevant information, major portion of network bandwidth are exhausted by transmitting ineffective data. A mobile crawler overcomes this difficulty since it can filter out all irrelevant page portions keeping only information which is relevant with respect to the search engine the crawler is working for. Remote page filtering is especially useful for search engines which use a specialized representation for Web pages (e.g., URL, title, modification date, keywords) instead of storing the complete page source code.
4. Remote Page Compression: In order to decrease the amount of data that is to be transmitted back to the crawler controller, we introduce remote page compression as another fundamental feature of mobile crawlers. To decrease the bandwidth required to transfer the crawler along with the data it holds back to the search engine, the mobile crawler reduces its size before transmission.

VII. Working of Proposed System

The Proposed system works as follows:

In the beginning process the mobile agent is dispatched to the remote server to crawl the web pages locally. First time, the HTML pages of different web sites are downloaded by the mobile agent onto the Client Site. These web pages are properly indexed and their full contents are stored in database at the Client Site. From the next time, the Crawl Manager generates the mobile agent for each Remote Site separately. The mobile agent moves to the Remote Site to crawl the pages allocated to it. At the remote site, the Mobile agent searches the pages recursively and then retrieves each page one by one in which their URL's are given in the database and it also download web page size.

The complete working of the proposed system is shown in the flow chart in Fig.4

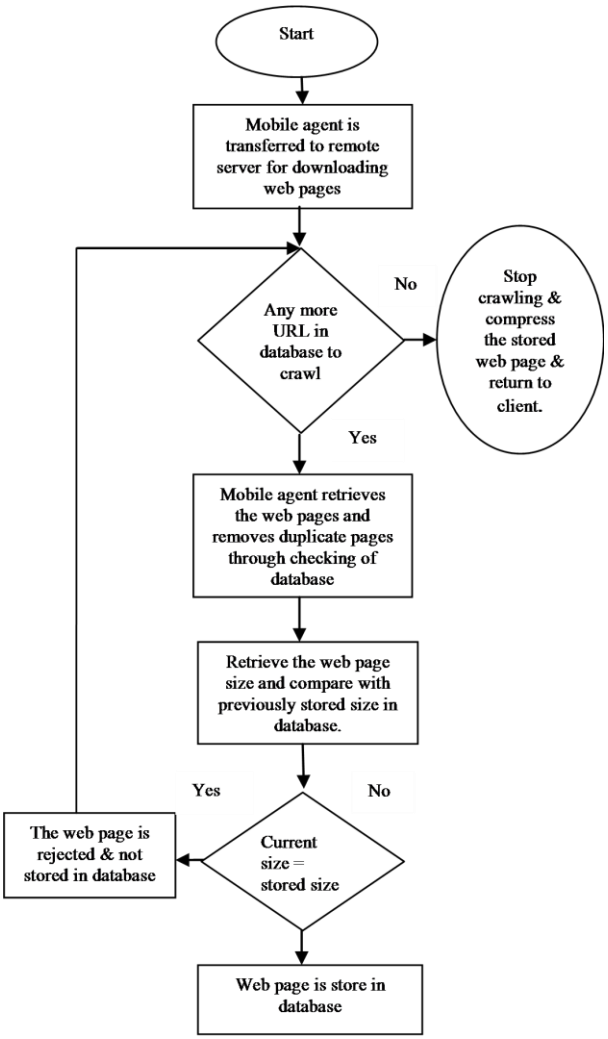

Fig. 4: Working of proposed system

VIII. Conclusion

In this paper we proposed a Model for Web Crawling based on Java Aglets. Web Crawling based on Mobile Agent will yield high quality pages. The crawling process will migrate to host or server to start downloading.

The Mobile agent based Web Crawler can filter out the web pages that have not been modified since last crawl. This technique can reduce the usage of CPU cycles at the remote site. The proposed mobile crawler system based on Java Aglets will reduce the traffic on the network and saved CPU cycles significantly as compared to Traditional Crawlers.

Acknowledgment

The author especially thanks to mention the financial help provided by MANF JRF under University Grant Commission (UGC), New Delhi. No. F1-17.1/2011/MANF-MUS-BIH-3287.

References

[1] Web Crawler, "The Web Crawler Search Engine", Web site, http://www.webcrawler.com.

[2] Berners-Lee, Tim, "The World Wide Web: Past, Present and Future", available at: http://www.w3.org/People/Berners-Lee/1996/ppf.html. MIT (1996).

[3] Brin S. and Page L., "The Anatomy of a Large Scale Hypertextual Web Search Engine," Technical Report, Stanford University, Stanford, CA (1997)

[4] Cho J., Garcia-Molina H., and Page L., "Efficient Crawling Through URL Ordering," Technical Report, Computer Science Department, Stanford University, Stanford, CA, USA (1997)

[5] Lange D. and Oshima M., Programming and Deploying Java Mobile Agents with Aglets, Addison Wesley (1998)

[6] Evens Jean, Tu Jiao, Ali R Hurson, and Thomas E. Potok. "SAS: A Secure Aglet Server", Computer Security Conference 2007, Myrtle Beach, SC, USA (2007)

[7] Internet World Stats. Worldwide internet users, available at: http://www.internetworldstats.com (accessed on March 5, 2013)

[8] Koster M., "Guidelines for Robot Writers", A Web Document, http://wsw.nexor.co.uk/mak/doc/robots/guidelines.html

[9] Fiedler J. and Hammer J., "Using Mobile Crawlers to Search the Web Efficiently," International Journal of Computer and Information Science, vol. 1, no. 1, pp. 36-58 (2000)

[10] Junghoo Cho and Hector Garcia-Molina "Parallel Crawlers". Proceedings of the 11th international conference on World Wide Web WWW '02", May 7–11, 2002, Honolulu, Hawaii, USA (2002)

[11] Nath, R.; Bal, S. and Singh, M., "Load Reducing Techniques an the Websites and other Resources: A comparative Study and Future Research Directions," Computer Journal of Advanced Research in Computer Engineering, vol. 1, no. 1, pp. 39-49 (2007).

[12] B. Kahle, "Archiving the Internet," Scientific American (1996)

[13] Shkapenyuk, V. and Suel, T., "Design and implementation of a high performance distributed web crawler", In Proceedings of the 18th International Conference on Data Engineering (ICDE), San Jose, California. IEEE CS Press, pp. 357-368 (2002).

[14] Md. Abu Kausar, V S Dhaka and Sanjeev Kumar Singh. "Web Crawler: A Review." International Journal of Computer Applications 63(2), pp. 31-36, USA (2013).

[15] Michael S. Greenberg and Jennifer C. Byingfon, Theophany Holding, David G. Harper, Tuffs University, "Mobile Agent and Security", IEEE Communications Magazine (1998).

Authors' Profiles

Md. Abu Kausar: received his BCA degree from T. M. Bhagalpur University, Bhagalpur in 2002, Master in Computer Science from G. B. Pant University of Agriculture & Technology, Pantnagar, Uttrakhand, India in 2006 and MBA (IT) from Symbiosis, Pune, India in 2012. He has received Microsoft Certified Technology Specialist (MCTS). At present, he is pursuing Ph.D in Computer Science from Jaipur National University, Jaipur, India and he is receiving UGC MANF JRF Fellowship during Ph.D Programme. He is having 6 years of experience as a Software Developer. His research interest includes Information Retrieval and Web Crawler.

Dr. V. S. Dhaka: is a young and dynamic technocrat with 10 years of intensive experience in industry and academia. He is M.Tech and Ph.D in computer Science from Dr. B R Ambedkar University, Agra, India. With more than 32 publications in international journals and paper presentations in 27 conferences/seminars, he always strives to achieve academic excellence. He has been awarded by the employers with "Employee of the Quarter Award", "Mentor of the year award" and with

letters of appreciations for his commitment, advocacy and mentor-ship. He has organized several Conferences, Seminars and Workshops.

Dr. Sanjeev Kumar Singh: is working as Assistant Professor in Department of Mathematics at Galgotias University, Gr. Noida, India. He earned his M. Sc. and Ph.D. degrees with major in Mathematics and minor in Computer Science from G.B. Pant University of Agriculture and Technology, Pantnagar, Uttrakhand, India. Before that he completed B.Sc. (Physics, Mathematics & Computer Science) from Lucknow University, Lucknow.

He is having more than nine years of teaching and research experience. Besides organizing three national conferences, he has published several research papers in various International/National journals of repute and several national and international conferences and workshops. His areas of interest include Mathematical Modeling, Differential Geometry, Computational Mathematics, data mining & Information Retrieval.

