6.1 CONCLUSION
In this thesis, the complete work done is divided into two parts. In the first part a novel architecture for incremental parallel crawlers has been proposed whereas in second part methods have been developed which detect whether two versions of a web document have been changed thereby helping to refresh the web documents to keep the repository up-to-date at Search engines side.

The novel architecture proposed for incremental parallel web crawler, helps to solve the following challenging problems which are still faced by almost every Search engine while running multiple crawlers in parallel for downloading the web documents for its repository.

- Overlapping of web documents
- Quality of downloaded web documents
- Network bandwidth/traffic

6.1.1 OVERLAPPING OF WEB DOCUMENTS
Overlap problem occurs when multiple crawlers running in parallel download the same web document multiple times due to the reason that one web crawler may not be aware of another having already downloaded the page.

In the proposed architecture the entire downloading process of web documents is performed under the coordination of Multi Threaded server and therefore no URLs have been assigned simultaneously to more than one client crawler executing in parallel. By applying this approach, server has the global image of the entire downloaded web documents and thus, overlapping problem is reduced.
In some cases, it happens that to increase the reliability, the same documents are maintained at multiple servers and thereby multiple copies of the same documents may be downloaded through different URLs. To avoid the duplication of same documents in the repository some very effective change detection methods have been developed.

6.1.2 QUALITY OF DOWNLOADED WEB DOCUMENTS
The quality of downloaded documents can be ensured only when web pages of high relevance are downloaded by the crawlers. Therefore, to download such relevant web pages by earliest, multiple crawlers running in parallel must have global image of collectively downloaded web pages.

In the proposed architecture the ranking algorithm developed, computes the relevance of the URLs to be downloaded on the basis of global image of the collectively downloaded web documents. It computes relevance on the basis of the forward link count as well as back link count whereas the existing method is only based on back link count. The advantage of the proposed ranking method over existing one is that it does not require the image of the entire Web to know the relevance of a URL as “forward link count” is directly computed from the downloaded web pages whereas “back link count” is obtained from in-built repository.

6.1.3 NETWORK BANDWIDTH/TRAFFIC
In order to maintain the quality, the crawling process is carried out using either of the following approaches.

- Crawlers can be generously allowed to communicate among themselves or
- They can not be allowed to communicate among themselves at all.

In the first approach network traffic will increase because crawlers communicate among themselves more frequently to reduce the overlap problem whereas in second approach, if they are not allowed at all to communicate then as a result same web document may be downloaded multiple times thereby consuming the network bandwidth. Thus, both approaches put extra burden on the network traffic.
The proposed architecture helps to reduce the overlapping and because all communications take place through MT Server, there is no direct communication requirement among client crawlers. Both of the above facilities help in reducing network traffic significantly.

6.1.4 CHANGE DETECTION METHODS FOR REFRESHING WEB DOCUMENTS

In the second part of the thesis, the change detection methods have been developed to refresh the web document by detecting the following major types of changes.

- Structural changes
- Presentation changes
- Content level changes

The structural change detection methods also help in detecting presentation changes as well. As the presentation of web documents gets modified through insertion/deletion/modification of the tag structure, the methods proposed for structural change detection also work efficiently and guarantees to detect the structural as well as presentation changes.

Two different schemes, document tree based and document fingerprint based have been developed for structural change detection. The document tree based scheme works efficiently and guarantees to detect structural changes. Apart from providing details about the structural changes within documents, it also helps in locating the area of major/minor change as well. For instance, the information about at what level how many nodes have been inserted/deleted is also reported. In document fingerprint based scheme, two separate fingerprints in the form of strings for each version of web document are generated on the basis of their tag structure. The first fingerprint contains the set of first characters of the tags in the order they appear in the web page whereas the second fingerprint contains the set of last characters of the tag in the order they appear in the web page. Time and space complexity in document tree based scheme is higher than the fingerprint based scheme. Generation and comparison of fingerprints in the form of strings, require less time than to generate and compare two trees. Also, the space required to store the node details (as node consists of multiple fields) in the document tree is high in
comparison to space required to store fingerprints. Though fingerprint based scheme is efficient in terms of space and time complexity, it also guarantees to detect structural changes even at micro level but the details about the changes can not be reported, for which document tree based scheme is better.

Similar to structural change detection scheme, two different schemes, Root Mean Square based and Checksum based have been developed for content level change detection. Both schemes are based on ASCII principle of symbols and both generate checksum for entire web page and also for different paragraphs. The checksum for entire web page helps to draw the picture about content level changes at page level whereas through paragraph checksum, changes at micro level i.e. paragraph level can be detected. Though, both the schemes are efficient and guarantee to detect changes at micro level also but the former scheme considers changes in contents uniformly whereas the later scheme assigns different weightage to different contents present in the web page. In this work, the font size has been considered for assigning the weightage. If the contents having bigger font size gets changed, it contributes more in comparison to contents having smaller font size.

6.2 FUTURE WORK
The work done in this thesis can be extended with the following list of possible future research issues with respect to incremental parallel web crawler:

1. In this thesis, work has been done to identify Structural, Presentation and Content level changes. Behavioral changes have not been discussed and have been left for future work.
2. Ideally, changes in the link of an image hyperlink can be detected through the structural change detection methods discussed but in case the image itself is replaced or modified then that can not be detected using the methods discussed. Work needs to be carried out to implement the proposed method for image change detection [128].
3. The architecture along with change detection methods can be synchronized with the frequency of change of web documents to get better results, as web documents from various domains change at different intervals.