CHAPTER 6

Chapter 6

Summary and conclusion

The subject of this work is signature based indexing structure for information retrieval on object-oriented data bases. An introduction to the area focused is presented in Chapter 1. The use of signatures in information retrieval, the various methods of signature generation and the research openings using signature-based methods have been reported. Further the use of signatures in object-oriented data bases and the need for the current work have also been stressed.

Related work carried out in the proposed research track is the core content of chapter 2. Signature file based techniques were initially applied on text data. So, the chapter commences with the discussion of various text retrieval methods used and justifies the use of signature file based method as a better option. Different methods to extract signatures are discussed next, among which superimposed coding - the best option has been substantiated from literature. Various physical representations of signature files like SSF, BSSF, CBSSF, S-tree, Multilevel signature file, Signature graph and Signature tree with their efficacy and shortcomings have been discussed. The motivation for this work was derived from the observations made from [CHEN 06]. The other related work of the proposed research track namely signature-based object-oriented query processing has been discussed. Since signature file based techniques are well suited for representing

set-valued attributes and nested object-hierarchy, further move of the research work was fixed in this track and the relevant reports from literature have been analyzed.

Signature file based access methods initially applied on text have now been used to handle set-oriented queries in OODBs. Most of the proposed methods use either efficient search method or tree based intermediate data structure to filter data objects matching the query. Use of search techniques retrieves the objects by sequentially comparing the positions of 1s in it. Such methods take longer retrieval time. On the contrary tree based structures traverse multiple paths making comparison process tedious. To bring better performance among the signature file representations a B+ tree based structure called Signature Declustering tree has been presented in Chapter 3. Using this for a given query signature all the matching signatures can be retrieved cumulatively in a single node. Also for signature insertion and query searching an optimal search path is used to speed up the process.

The analytical and experimental results with comparative analysis of Signature tree [CHEN 06] show that considerable search time is saved. Also the query response time is lesser than signature insertion time in SD-tree based processing. The SD-tree structure results in shorter tree for a given database size and hence the search cost is also reduced. The space overhead in SD-tree may be higher due to the presence of binary prefixes in higher order signature nodes; however when the number of bits used to represent the database size n exceeds the signature length which is normally true for large databases, SD-tree shows better performance. In general SD-tree shows potential

improvement of various complexities and hence is an effective structure for signature-based applications.

Signatures are preferred in handling set-oriented operations because they are of fixed length and hence convenient for index structures, set comparison operators on signatures can be easily implemented by efficient bit operations and they are more space efficient compared to the conventional set representation. Hence applying SD-tree on set-oriented (inclusive) queries was the topic of chapter 4. It has been observed that SD-tree which is retrieving all matching signatures in a single access handles both types of queries efficiently. Further the advantages of SD-tree like its flexibility and fast retrieval time is well-retained which promotes efficient handling of object-oriented queries.

Objects can be nested by storing the child OIDs in the complex attributes of the parent objects. The nesting structure of objects can easily be represented using signatures. Since the signature of an object is the superimposed signature of its constituent attributes signatures, the embedded values of nested hierarchy as well is easily reflected in signatures. In chapter 5, SD-tree's behavior in nested object hierarchy environment is studied. To handle nested object queries the signatures of each class in the class hierarchy are stored in separate SD-tree structures forming a hierarchy. In each level except the target class the only matching OIDs' signatures obtained from the previous level are inserted in SD-tree for further searching. The object query is stored in a tree like structure to promote parallel comparison as in [CHEN 04]. Use of SD-tree structure cuts down irrelevant branches as well as the number of checked nodes. This improves the

search time complexity and adds more to optimality. The limitation observed is that the space overhead is more in SD-tree due to heavier signature nodes in the upper level and signature number replication for all set bits of the signature.

The contribution of this thesis is that the proposed data structure is an efficient way of retrieving information from large databases. It can be well adapted to different query styles of OODBs.

The results of any research work surely will be a starting point of another. Only sky is the limit for improvements of any work in any track. This work also has paved way for further enhancements. Some are listed below:

- The SD-tree structure could further be improved by using 0s when the signature weight is more than 50%, so that number of signature nodes accessed for insertion and search is optimal.
- By providing a means of accessing OIDs from the structure for handling complex queries.
- To implementing the system to run on real and large OODB.
- The system can further be tested for point and range queries in the OO hierarchy.
- The system could be tested for nested queries of type MX.