ABSTRACT

In present days, the computerized systems such as e-commerce portals, web applications, high-end desktop applications and mobile applications store and process huge volume of data. Powerful database management systems are required to process these data. The users of these applications access data which are distributed on various sites. Different levels of transparencies should be provided to hide complexity at network, database, application, operating system and location levels. The database transparency is required because data are stored in multiple heterogeneous databases. Apart from this, the physical structures such as indexes, materialized views, partitions, etc. should be designed properly. Poorly designed physical structures degrade the transaction performance, while properly designed physical structures improve the performance. In this thesis, the research work is done to improve concurrent transaction execution in multiple heterogeneous databases by analyzing and recommending new indexes for tables. The tool Table Index Evaluator and Recommender (TIER) has been developed which takes set of queries as input from different databases, parses the queries, calculates total frequency of fields used in conditions and finds out ratio of usage for these fields. The parser module of TIER is based on the mathematical model which generates parse matrix on the basis of set of inputted queries and workload. After parsing the queries, it separates out fields and tables referred in various clauses of these queries. It also collects statistics from the system catalog. On the basis of collected statistics and frequency of fields referred in queries, it suggests unique, bitmap and composite indexes. TIER also provides the module to compare performance of a query before and after applying the recommended indexes. It helps database administrator to take decision whether or not to apply recommended indexes. To test the performance of transactions after applying recommended indexes, TPC-H benchmark of benchmark factory is used. Huge volume of data from this benchmark factory is loaded into multiple heterogeneous databases. Different jobs for different sets of queries are created in benchmark factory to perform scalability test. These jobs are based on different user loads (concurrent users). Many iterations of the same job is performed. This process is repeated up to three times. After executing different runs of these jobs for various user loads, the data is generated by benchmark in the form of average response time for different transactions before and after applying recommended indexes. To check whether the performance is improved or degraded after applying recommended indexes, the paired t-test is performed on these data. The null hypothesis rejected after the test, which shows that the performance is improved in terms of response time after implementing the recommended indexes.