Aqua: A Fast Decision Support System Using Approximate Query Answers

Swarup Acharya Phillip B. Gibbons Viswanath Poosala

Information Sciences Research Center
Bell Laboratories
600 Mountain Avenue
Murray Hill NJ 07974

Abstract

Aqua is a system for providing fast, approximate answers to aggregate queries, which are very common in OLAP applications. It has been designed to run on top of any commercial relational DBMS. Aqua precomputes *synopses* (special statistical summaries) of the original data and stores them in the DBMS. It provides approximate answers (with quality guarantees) by rewriting the queries to run on these synopses. Finally, Aqua also incrementally keeps the synopses up-to-date as the database changes.

1 Motivation

Traditional query processing has focused solely on providing exact answers to queries, in a manner that seeks to minimize response time and maximize throughput. However, in large data recording and warehousing environments, providing an exact answer to a complex query can take minutes, or even hours, due to the amount of computation and disk I/O required.

There are a number of scenarios in which an exact answer may not be required, and a user may prefer a fast, approximate answer. For example, during some drill-down query sequences in ad-hoc data mining, initial queries in the sequence are used solely to determine what the interesting queries are. An approximate answer can also provide feedback on how well-posed a query is. Moreover, it can provide a tentative answer to a query when the base data is unavailable. Another example is when the query requests numerical answers, and the full precision of the exact answer is not needed, e.g., a total, average, or percentage for which only the first few digits of precision are of interest (such as the leading few digits of a total in the millions, or the nearest percentile of a percentage).

Motivated by these concerns, we have developed the Approximate QUery Answering (Aqua) system. Aqua is a system designed to provide fast, approximate answers to aggregate as well as set-valued queries. The work is tailored to data warehousing environments. Our goal is to provide an estimated response in orders of magnitude less time than the time to compute an exact answer, by avoiding or minimizing the number of accesses to the base data.

2 Architecture

Aqua is designed as a module that sits on top of any SQL-compliant DBMS managing a data warehouse. Aqua precomputes statistical summaries on the relations in the warehouse. Currently, the statistics take the form of various types of samples and histograms, and are stored as regular relations inside the warehouse; they are also incrementally maintained up-to-date as the base data is updated.

Aqua answers user queries using the pre-computed summaries. Approximate answers are provided by rewriting the user query over the summary relations and executing the new query. The rewriting involves suitably scaling the results of certain operators within the query. Finally, the

**Proceedings of the 25th VLDB Conference,
Edinburgh, Scotland, 1999.**

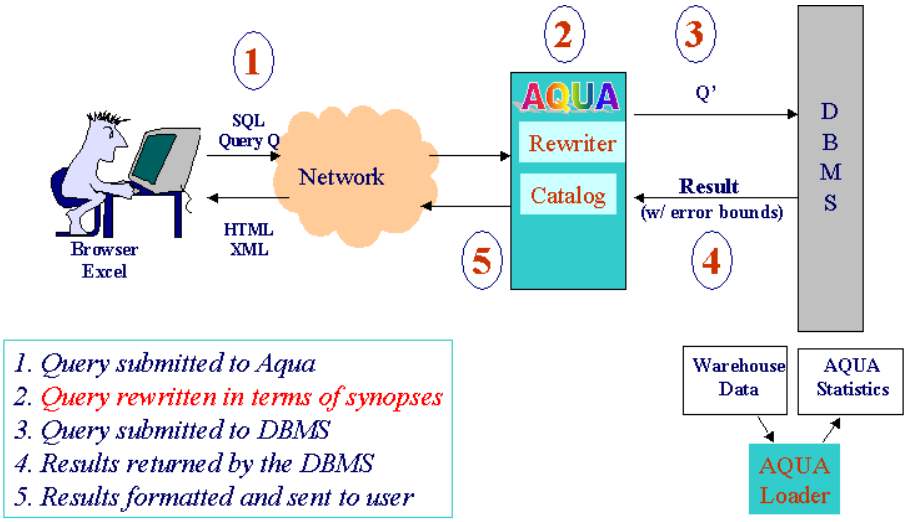

Figure 1: The Aqua architecture.

query and the approximate answer are analyzed to provide guarantees on the quality of the answer, and report error bounds. The high-level architecture of Aqua is depicted in Figure 1, along with the steps taken during query processing. As new data arrives, Aqua maintains the synopses up to date, with few or no accesses to the original data.

3 Aqua Technical Results and Operational Details

There are several technical problems arising in answering approximate queries. We have identified and solved a few of them, and incorporated the solutions into Aqua. Many of these results appear in [GMP97, GM98, AGPR99]. The key features of Aqua are as follows:

- Novel incremental maintenance techniques for keeping histograms and samples up-to-date in the presence of database updates.
- Improved error bounds based on a novel subsampling scheme.
- Strategies for allocating space among various summary statistics.
- Biased samples for improving the accuracy for queries with group-by operations.
- Improved sampling techniques (concise and counting samples) that use less space than traditional samples.

We have shown that schemes for providing approximate answers to multi-table queries that rely on using random samples of base relations alone suffer from serious disadvantages. We have developed an approach, which we call *join synopses*, that overcomes these disadvantages. We have shown both theoretically and empirically that join synopses provide highly-accurate answers and tighter confidence bounds than sampling from base relations.

In a recent work [AGP99], we demonstrate the drawback of uniform samples to effectively answer *group-by* queries, a key component of drill-down and roll-up analysis in OLAP. We propose new biased sampling techniques to address this handicap. Incorporation of these techniques into Aqua have shown their utility in making group-by queries significantly more accurate in practice.

As an illustration of query processing in Aqua, we present a key component, the query rewrite process using a simple example (details in [AGPR99]). Figure 2 gives an example of this rewriting that takes into account join synopses. The query is based on the schema for the TPC-D benchmark. When the query is submitted to Aqua, it identifies the join being computed in the query and rewrites the query to refer to the appropriate join synopsis. Specifically, the table names `lineitem` and `order` are replaced by the Aqua table names `bs_lineitem` and `js_order`. In this example, the resulting join synopsis is a 1% sample of the join between `lineitem` and `order`, so the `sum` aggregate in the select clause is scaled by 100. The rewritten query submitted to the warehouse is shown in Figure 2(b). (Calculation of error bounds is not shown here for simplicity.)

Aqua also provides a web-based interface to allows users to pose queries. A screenshot of the interface is

```
select sum(l_quantity)
  from lineitem, order
  where l_orderkey = o_orderkey
  and o_orderstatus = F
```

(a) Original query

```
select 100*sum(l_quantity)
  from bs_lineitem, js_order
  where l_orderkey = o_orderkey
  where o_orderstatus = F
```

(b) Rewritten query

Figure 2: Query rewriting to use join synopses.

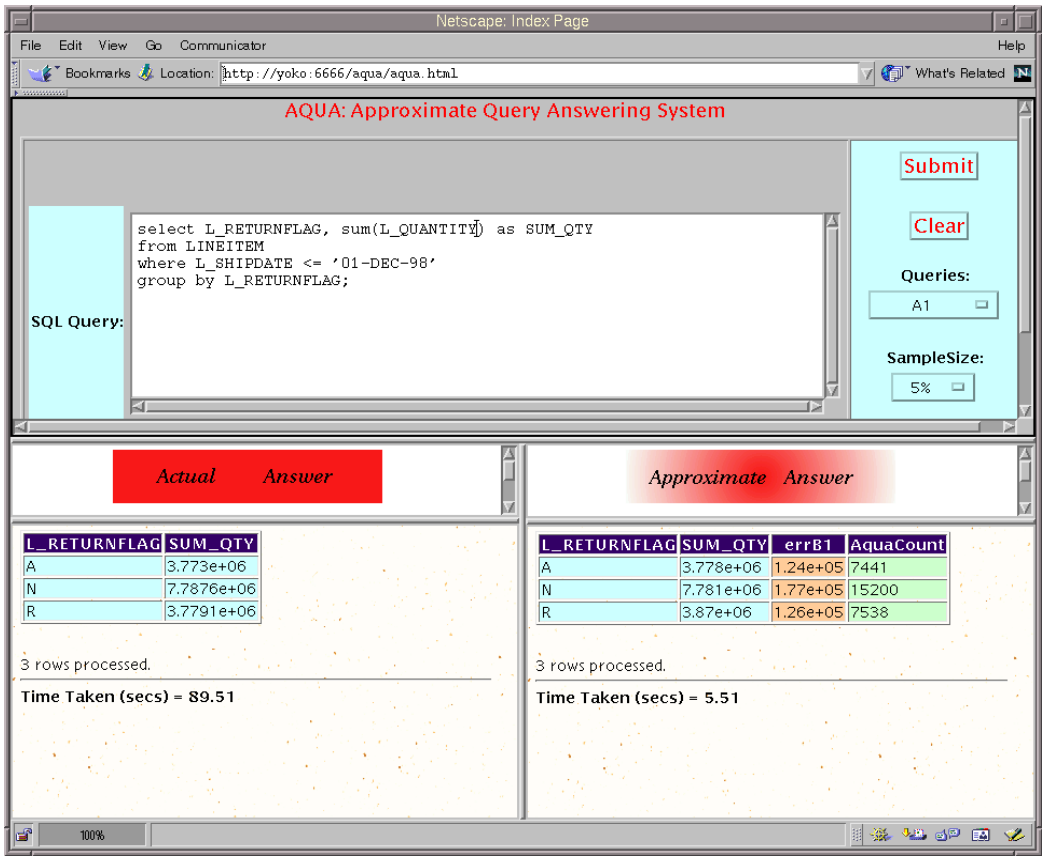

Figure 3: Aqua user interface

shown in Fig 3. It shows the actual answer and the approximated answer generated by Aqua for a simplified version of Query $Q1$ of the TPC-D suite along with the running times. The results are for a 0.1 scalefactor TPC-D database (100 MB). The approximate answer window also shows the error bound calculated for each group (errB1), along with a count of the tuples used to generate the estimate (AquaCount).

4 Related Work

Statistical techniques have been applied in databases for more than two decades now, but primarily inside a query optimizer for selectivity estimation. However, the application of statistical techniques to approximate query answering has started receiving attention only very recently. Hellerstein *et al.* [HHW97] proposed a framework for approximate answers of aggregation queries called *Online Aggregation*, in which the base data is scanned in random order at query time and the approximate answer is continuously updated as the scan proceeds. Unlike Aqua, this work involves accessing the base data at query time, thus being more costly; but at the same time, this approach provides the fully accurate answer gradually. Other systems support limited on-line aggregation features; e.g., the Red Brick system supports running COUNT, AVG, and SUM. Since the scan order used to produce these aggregations is not random, the accuracy can be quite poor. In the *Approximate*

query processor, developed by Vrbsky and Liu [VL93], an approximate answer to a set-valued query is any superset of the exact answer that is a subset of the cartesian product. Recently, Ioannidis and Poosala have developed a robust numerical measure for computing the error in an approximate set-valued query answer and also provided histogram-based techniques for answering complex queries [IP99]. There has also been some work on using histograms and wavelets to approximate the data cube for providing approximate answers to aggregate queries [PG99, VWI98].

5 Details of the Demo

The main focus of the demo will be on the ability of Aqua to provide quick and high-quality approximate answers to queries of varying complexities. The data warehouse will consist of the popular TPC-D benchmark data loaded into a commercial DBMS. The queries include many of the benchmark queries as well as some generated by us, chosen to demonstrate the benefits of various statistical techniques in Aqua. In particular, we will show that join synopses outperform base samples for queries with joins and that biased samples outperform uniform samples for group-by queries. We will also show some of the key features of the Aqua front-end, such as the ability to execute a query using any one of the available sets of statistics.

Acknowledgments

The authors acknowledge the contribution of Sridhar Ramaswamy in the initial design and development of the Aqua prototype.

References

[AGP99] S. Acharya, P. B. Gibbons, and V. Poosala. Congressional samples for approximate answering of group-by queries. Technical report, Bell Laboratories, Murray Hill, New Jersey, February 1999.

[AGPR99] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. Join synopses for approximate query answering. In *Proc. of ACM SIGMOD Conf*, June 1999.

[GM98] P. B. Gibbons and Y. Matias. New sampling-based summary statistics for improving approximate query answers. *Proc. of ACM SIGMOD Conf*, pages 331–342, June 1998.

[GMP97] P. B. Gibbons, Y. Matias, and V. Poosala. Fast incremental maintenance of approximate histograms. *Proc. of the 23rd Int. Conf. on Very Large Databases*, pages 466–475, August 1997.

[HHW97] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. In *Proc. ACM SIGMOD International Conf. on Management of Data*, pages 171–182, May 1997.

[IP99] Y. Ioannidis and V. Poosala. Histogram-based techniques for approximating set-valued query-answers. *Proc. of the 25rd Int. Conf. on Very Large Databases*, September 1999.

[PG99] V. Poosala and V. Ganti. Fast approximate answers to aggregate queries on a data cube. *International working conference on Scientific and Statistical Database Management*, July 1999.

[VL93] S. V. Vrbsky and J. W. S. Liu. Approximate—a query processor that produces monotonically improving approximate answers. *IEEE Trans. on Knowledge and Data Engineering*, 5(6):1056–1068, 1993.

[VWI98] J. S. Vitter, M. Wang, and B. R. Iyer. Data cube approximation and histograms via wavelets. *Proc. of the CIKM*, pages 96–104, 1998.