The Aqua Approximate Query Answering System

Swarup Acharya Phillip B. Gibbons Viswanath Poosala Sridhar Ramaswamy*

{swarup,gibbons,poosala}@research.bell-labs.com, sridhar@epiphany.com
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 along with quality guarantees by rewriting the queries to run on these synopses. Finally, Aqua keeps the synopses up-to-date as the database changes, using fast incremental maintenance techniques.

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 built the Approximate QUery Answering (Aqua) system. Aqua is a system designed to provide fast, approximate answers to primarily aggregate, but also to 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, called *synopses*, 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. A key feature of Aqua is that the system provides probabilistic error/confidence bounds on the answer, based on Hoeffding and Chebychev bounds [AGPR99]. Currently, the system handles arbitrarily complex SQL queries applying aggregate operations (`avg, sum, count, etc.`) over the data in the warehouse.

The high-level architecture of the Aqua system is shown in Figure 1. Initially, Aqua takes as an input the space available for the Aqua synopses and if available, hints on important query and data characteristics from the warehouse administrator.[1]

*This work was done while the author was at Bell Labs. Current affiliation is Epiphany Inc., 2300 Geng Road, Suite 200, Palo Alto CA 94303.

[1]Work is also in progress to automatically extract this information from a query workload and adapt the statistics dynamically.

SIGMOD '99 Philadelphia PA

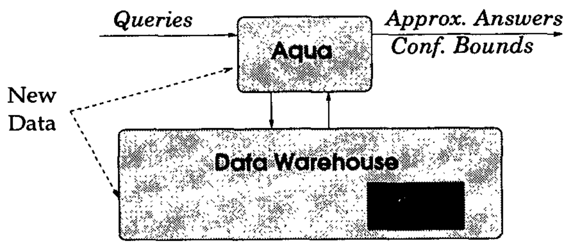

Figure 1: The Aqua architecture.

This information is then used to precompute a suitable set of synopses on the data, which are stored as regular relations in the DBMS. These synopses are also incrementally maintained up-to-date as more data is added to the warehouse.

Aqua provides approximate answers by rewriting and executing the user query over the summary synopses. The rewriting also involves suitably scaling the results of certain operators within the query. Finally, the query and the approximate answer are analyzed (as part of the query itself) to provide guarantees on the quality of the answer, and report error bounds.

3 Technical 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 techniques appear in [GMP97, AGPR99, AGP99]. We briefly highlight three of them below.

Join Synopses. In [AGPR99], 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. Instead, we proposed the use of *join synopses*, precomputed samples of a small set of distinguished joins, in order to compute approximate answers for multi-table aggregate queries. We showed that for queries with foreign-key joins, Aqua can provide good quality approximate answers to multi-table aggregate queries using a very small number of join synopses. ∎

Congressional Samples. In [AGP99], we have demonstrated the drawback of pre-computed uniform random samples to effectively answer *group-by* queries, a key component of drill-down and roll-up analysis in OLAP. We showed that, to be effective for group-by queries, non-uniform *biased sampling* should be used. We proposed techniques based on biased sampling which minimize the error when the grouping attributes are known beforehand. Additionally, we introduced a hybrid technique, called *congressional samples*, which is effective for queries with arbitrary group-bys (including none). ∎

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

(a) Original query

```
select 100*sum(l_quantity),
  sum_error(l_quantity)
  from L0synopsis
  where o_orderstatus = F
```

(b) Rewritten query

Figure 2: Query rewriting using join synopses.

Incremental Maintenance. In [GMP97], we have provided efficient techniques for the incremental maintenance of equi-depth and compressed histograms [PIHS96] on a relation. These techniques made use of precomputed uniform samples (called *backing samples*) of the data. In the same paper, we also provided techniques for the incremental maintenance of the samples. In [AGPR99] and [AGP99], we showed how to incrementally maintain join synopses and congressional samples, respectively. ∎

As an illustration of query processing using Aqua, we describe next a key component, the query rewrite process, using a simple example (further details are 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. Here, `L0synopsis` is a 1% sample of the join between the `lineitem` and `order` tables. When the query is submitted to Aqua, it identifies the join being computed in the query and then rewrites the query to refer to the `L0synopsis` table instead of the base tables. The rewritten query submitted to the warehouse is shown in Figure 2(b). Since the join synopsis is a 1% sample of the join, the rewritten query scales the sum by 100. In this figure, the error bound is encapsulated for simplicity by the `sum_error` function.

Aqua also provides a web-based interface to allow users to pose queries. A screen-shot of the interface is shown in Fig 3. It shows the actual answer

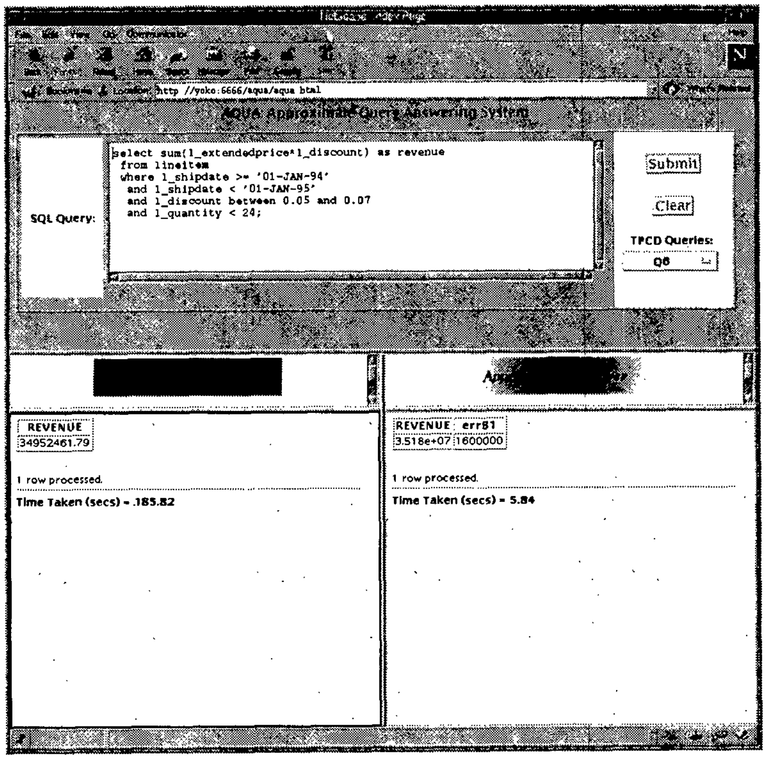

Figure 3: Aqua user interface

and the approximate answer generated by Aqua for Query $Q6$ of the TPC-D benchmark along with the running times. Both answers are shown for comparison. Typically, Aqua users will request only the fast approximate answer, resorting to the (much slower) actual answer only when the full precision of the answer is required.

4 Conclusions

The need for approximate query answering is increasing due to the rapidly growing demand for OLAP and data analysis on larger and larger data warehouses. Aqua addresses this need by providing an efficient decision support system based on approximate query answering. Aqua maintains summary synopses of the warehouse relations and rewrites user queries to run off these synopses. Since these synopses are significantly smaller than the base warehouse data, there is usually an order of magnitude or more improvement in response time. Additionally, Aqua provides guaranteed error bounds on the quality of the estimated answers. Our tests show that the tradeoff in the loss of accuracy to the time saved in Aqua is extremely favorable to the users.

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, March 1999.

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

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

[PIHS96] V. Poosala, Y. E. Ioannidis, P. J. Haas, and E. J. Shekita. Improved histograms for selectivity estimation of range predicates. In *Proc. ACM SIGMOD International Conf. on Management of Data*, pages 294–305, June 1996.