Approximate Query Answering Using Data Warehouse Striping

Jorge Bernardino[1], Pedro Furtado[2], Henrique Madeira[2]

[1] Institute Polytechnic of Coimbra, ISEC, DEIS, Apt. 10057
P-3030-601 Coimbra, Portugal
jorge@isec.pt
[2] University of Coimbra, DEI, Pólo II
P-3030-290 Coimbra, Portugal
{pnf, henrique}@dei.uc.pt

Abstract. This paper presents an approach to implement large data warehouses on an arbitrary number of computers, achieving very high query execution performance and scalability. The data is distributed and processed in a potentially large number of autonomous computers using our technique called data warehouse striping (DWS). The major problem of DWS technique is that it would require a very expensive cluster of computers with fault tolerant capabilities to prevent a fault in a single computer to stop the whole system. In this paper, we propose a radically different approach to deal with the problem of the unavailability of one or more computers in the cluster, allowing the use of DWS with a very large number of inexpensive computers. The proposed approach is based on approximate query answering techniques that make it possible to deliver an approximate answer to the user even when one or more computers in the cluster are not available. The evaluation presented in the paper shows both analytically and experimentally that the approximate results obtained this way have a very small error that can be negligible in most of the cases.

1 Introduction

Data warehousing refers to "a collection of decision support technologies aimed at enabling the knowledge worker (executive, manager, analyst) to make better and faster decisions" [6]. A data warehouse is a global repository that stores large amounts of data that has been extracted and integrated from heterogeneous, operational or legacy systems. OLAP is the technique of performing complex analysis over the information stored in a data warehouse [8]. The data warehouse coupled with OLAP enable business decision makers to creatively analyze and understand business trends since it transforms operational data into strategic decision making information. Typical warehouse queries are very complex and ad hoc and generally access huge volumes of data and perform many joins and aggregations. Efficient query processing is a critical requirement for data warehousing because decision support applications typically require interactive response times.

In this paper, we assume that a multidimensional database is based on a relational data warehouse in which the information is organized as a star schema [12]. A star schema is composed by a set of dimension and fact tables where the fact table accounts for most of the space occupied by all the tables in the star for most of the cases [13]. However, this table is highly normalized, which means that it really represents the most effective (concerning storage space) relational way to store the facts. The dimension tables are very denormalized, but these tables usually represent a small percentage of the space in the star. The star schema is also particularly optimized for the execution of complex queries that aggregate a large amount of data from the fact table.

We use a new round-robin data partitioning approach for relational data warehouse environments proposed in [5]. This technique, called data warehouse striping (DWS), takes advantage of the specific characteristics of star schemas and typical data warehouse queries profile to guarantee optimal load balance of query execution and assures high scalability. In DWS, the fact tables are distributed over an arbitrary number of workstations and the queries are executed in parallel by all the workstations, guaranteeing a nearly linear speedup and significantly improving query response time.

In spite of the potential dramatic speedup and scaleup that can be achieved by using the DWS technique, the fact that the data warehouse is distributed over a large number of workstations (nodes) greatly limits the practical use of the technique. The probability of having one of the workstations in the system momentarily unavailable cannot be neglected for a large number of nodes. The obvious solution of building the system using fault tolerant computers is very expensive and will contradict the initial assumption of DWS technique of using inexpensive workstations with the best cost/performance ratio. Nevertheless, high availability is required for most data warehouses, especially in areas such as e-commerce, banks, and airlines where the data warehouse is crucial to the success of the organizations.

In this paper, we propose a new approximate query answering strategy to handle the problem of temporarily unavailability of one or more computers in a large data warehouse implemented over a large number of workstations (nodes) using the DWS technique. In the proposed approach the system continues working even when a given number of nodes are unavailable. The partial results from the available nodes are used to return an estimation of the results from the unavailable nodes. Currently, we provide approximate answers for typical aggregation queries providing the user with a confidence interval about the accuracy of the estimated result. The analytic and experimental study presented in this paper show that the error introduced in the query results can be really very small.

The rest of the paper is organized as follows. In the next section, we give an overview of related work and discuss the problems associated with approximate answering in data warehouses. Section 3 briefly describes the DWS approach and section 4 discusses approximate query answering using DWS in the presence of node failures. Section 5 analyzes the experimental results and the final section contains concluding remarks and future work.

2 Related Work

Statistical techniques have been applied to databases in different tasks for more than two decades (e.g. selectivity estimation [14]). Traditionally, researchers are interested in obtaining exact answers to queries, minimizing query response time and maximizing throughput. In this work, we are interested in analyzing and giving solutions to the failure of one or more workstations in a DWS system. Thus, it has some similarities with approximate query answering research, where the main focus is to provide fast approximate answers to complex queries that can take minutes, or even hours to execute. In approximate query answering the size of the base data is minimized using samples, which is analogous to the failure of a workstation inhibiting the access to the part of the base data that resides in that workstation. A survey of various statistical techniques is given by Barbara et al [4].

Recently, there has been a significant amount of work on approximate query answering [1, 10, 16]. One of the first works was [11] where the authors 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. The Approximate Query Answering (AQUA) system [9] provides approximate answers using small, pre-computed synopses of the underlying base data. Other systems support limited on-line aggregation features; e.g., the Red Brick system supports running COUNT, AVG, and SUM [11].

Our work is also related to distributed processing in data warehouses. The fact that many data warehouses tend to be extremely large in size [6] and grow quickly means that a scalable architecture is crucial. In spite of the potential advantages of distributed data warehouses, especially when the organization has a clear distributed nature, these systems are always very complex and have a difficult global management [2]. On the other hand, the performance of distributed queries is normally poor, mainly due to load balance problems.

In this context, the DWS concept provides a flexible approach to distribution, inspired in both distributed data warehouse architecture and classical round-robin partitioning techniques. The data is partitioned in such a way that the load is uniformly distributed to all the available workstations and, at the same time, the communication requirements between workstations is kept to a minimum during the query computation phase. This paper marries the concepts of distributed processing and approximate query answering to provide a fast and reliable relational data warehouse.

3 Data Warehouse Striping

Star schemas provide intuitive ways to represent the typical multidimensional data of businesses in a relational system. In the data warehouse striping (DWS) approach, the data of each star schema is distributed over an arbitrary number of workstations having the same star schema (which is the same schema of the equivalent centralized version). The dimension tables are replicated in each machine (i.e., each dimension has exactly the same rows in all the workstations) and the fact data is distributed over

the fact tables of each workstation using a strict row-by-row round-robin partitioning approach (see Figure 1). Each workstation has *1/N* of the total amount of fact rows in the star, with *N* being the number of workstations.

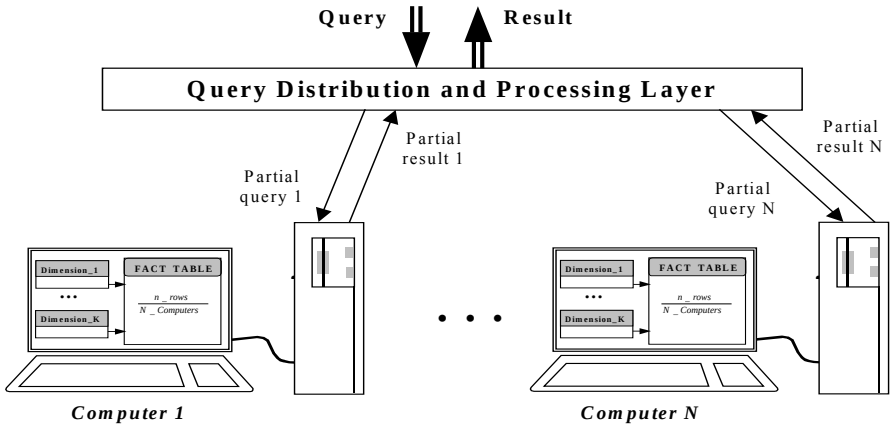


Fig. 1. Data Warehouse Striping

Most of the queries over a star schema can be transformed into *N* independent partial queries due to their nature (and because the fact rows are partitioned in a disjoint way: i.e., the rows stored in a workstation are not replicated in other workstation).

Although this is not the main point of this paper, it is important to mention that DWS achieves an optimal speedup. We have made experiments with 3, 5 and 10 workstations using a set of typical queries from APB-1 benchmark [3] and we obtained an average speedup of 3.01, 5.11 and 11.01, respectively. These results show an optimal speedup for all configurations. In fact, the speedup is higher than the theoretical value, because the centralized data warehouse that was used as the reference experiment worked near the workstation memory and I/O limits. Although the speedup increases as the number of workstations used in a DWS system increases, the probability of unavailability of a subset of those workstations also increases proportionally.

In practice, this means that DWS could not be used with a large number of workstations because the DWS will be unstable. This problem therefore would impair the use of this simple technique. However, we propose a radical solution to this problem, which allows even ordinary computers to be used as DWS computing units (without any special characteristic of hardware or software fault tolerance). In the proposed solution, the system continues running normally, even if one or more workstations are unavailable, and an approximate answer is computed from those that are available. Additionally, confidence intervals are also returned with the estimated result.

We experimentally analyzed our proposal with different configurations (different number of workstations) and in different scenarios of unavailability. The extreme case of unavailability will also be analyzed (when the user can only access his own workstation), as the approximate result obtained this way can still be useful when s/he is in the phase of "digging" the data and the precision of the result is not a crucial issue.

4 Approximate Query Answering in the Presence of Node Failures

Our proposal consists of providing the user with an answer even when one of the machines in the DWS system has a failure. In this case the answer will be approximate, because some partial results of the failed machines are unavailable. DWS is working normally with approximate answers until we manually recover the workstation. However, we show that this solution is acceptable for the following reasons:

- The error is very small, as will be shown in this paper and a small error is not relevant for the decision support activities in most of the cases;
- It is possible to provide the user with a confidence interval that gives him/her an idea of the amount of error in the approximate result.

For queries using aggregation functions an approximate answer is simply an estimated value for the answer given together with an accuracy value in the form of confidence intervals. We provide confidence intervals based on large sample bounds [10].

4.1 Estimated Values

In this section, we will see how DWS compute the approximate aggregation values when one or more workstations cannot contribute to the final result. Consider the number of workstations used in DWS to be $N = N_u + N_a$, where N_u is the number of workstations that are unavailable and N_a is the number of workstations that are available, contributing to compute the estimated aggregation value. If the aggregations functions to compute are average, sum or count and one or more workstations are unavailable the approximate average, sum and count are simply given by

$$AVERAGE_{estimated} = \frac{SUM_a}{COUNT_a},\ SUM_{estimated} = SUM_a \frac{N}{N_a},\ COUNT_{estimated} = COUNT_a \frac{N}{N_a} \quad (1)$$

where SUM_a and $COUNT_a$ represents the partial sum and count from the available workstations and N is the number of workstations used in the DWS system. Intuitively, the overall estimated average is equal to the average taken from the available nodes. These are the formulas that will be used in our experiments to compute the estimated values of the queries.

4.2 Analysis of the Error Incurred in DWS Estimations

When one or more workstations are unavailable, approximate query answers must be given. Although it is not possible to return exact answers in those cases, the estimation is extremely accurate for an important subset of typical query patterns consisting of aggregations of values into categories. The estimation is based in statistical inference using the available data as samples. We assume that the random sample is taken from an arbitrary distribution with unknown mean μ and unknown variance σ^2. We make the additional assumption that the sample size n_s is large enough (n_s>30) so that the Central Limit Theorem can be applied and it is possible to make inferences concerning the mean of the distribution [7]. As σ is unknown, we replace it with the estimate

s, the sample standard deviation, since this value is close to σ with high probability for large values of n_s. Thus, bounds on the confidence interval for the mean of an arbitrary distribution are given by $\bar{x}$ ±Error, where

$$\text{Error} = \pm z_{\alpha/2} \times \frac{s}{\sqrt{n_s}} \sqrt{\frac{n - n_s}{n - 1}} \quad . \tag{2}$$

In this expression, s is the standard deviation of the sample and n is the population size. The term $z_{\alpha/2}$ is the corresponding percentile in the normal distribution. This expression shows that the error decreases significantly as the sample size n_s increases and eventually reaches extremely small values for very large sample sizes ($n_s \approx n$).

The distribution of the fact table rows into N workstations is considered pseudo-random because a round-robin approach is used. As a result, we assume the values in any aggregation pattern to be evenly distributed by all the workstations. For simplicity, we consider that an average is being computed over each aggregation group. We also consider that N_u workstations are unavailable (cannot contribute with partial values to the final query result). Some reasonable assumptions can be made concerning the values taken by these variables,

- $1 < N \leq 100$
- N_u is typically a small fraction of N

Consider also an aggregation of n_g values into one of the group results, with n_g being reasonably large (e.g. $n_g \geq 100$). The number of values available in the sample when N_u workstations are unavailable is $n_g - n_g/N \times N_u = n_g\,(1 - N_u/N)$ and the error is:

$$\text{Error} = \pm z_{\alpha/2} \times \frac{s}{\sqrt{n_g\,(1 - N_u / N)}} \sqrt{\frac{n_g - n_g\,(1 - N_u / N)}{n_g - 1}} \approx \pm z_{\alpha/2} \times \frac{s}{\sqrt{n_g\,(1 - N_u / N)}} \sqrt{\frac{N_u}{N}} \tag{3}$$

This value is typically extremely small for the type of queries considered, because the number of values aggregated into a specific group (n_g) is at least reasonably large (e.g. more than 100 values) and the fraction N_u/N is usually small. In other words, a sufficiently large sample is usually available, resulting in very accurate estimations. In these formulas we are concerning about the mean of the distribution but if we would like to compute the sum or count is only multiply the formulas above by the number of elements in each group (n_g). Figure 2 shows the 99% interval for the error taken as a function of the fraction of workstations that are unavailable (x axis) and considering also different numbers of values aggregated in a group. These results were obtained by considering the standardized normal distribution N(0,1).

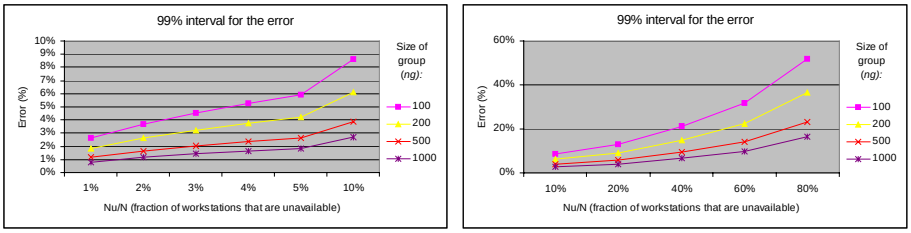


Fig. 2. 99% Confidence intervals for the error

The results of the left picture show that the error is very small when the fraction of workstations that are unavailable is reasonably small or the number of values aggre-

gated into the group is sufficiently large. Additionally, although the increase in the fraction of workstations that are unavailable results in larger errors, as shown in the right picture, those errors are not very large in many cases and decrease significantly as n_g increases. For instance, it can be seen from the left picture that the error is extremely small when the fraction of unavailable workstations is less or equal to 10% and the number of values aggregated into the group is larger than 200.

5 Experimental Results

In these experiments we evaluate the error and test if it is within the estimated confidence intervals in a large variety of cases. We are also interested in analyzing the influence of group-by queries with different sizes of groups.

The TPC-H benchmark [15] was used as an example of a typical OLAP application. It consists of a suite of business oriented ad hoc queries illustrating decision support systems that examine large volumes of data, execute queries with a high degree of complexity and give answers to critical business questions. Therefore, in accordance with the above criteria, we concentrated our attention on queries Q1, Q6 and Q7 of TPC-H benchmark. These queries have the characteristics shown in Table 1, where the first column represents the number of rows processed in average for each group, the second column show the number of groups and the third represents the average group selectivity when the data warehouse is centralized in one workstation.

Table 1. Characteristics of queries Q1, Q6 and Q7

	number of rows/group	number of groups	selectivity
Q1	1,479,417	4	14.6 %
Q6	114,160	1	1.9 %
Q7	1,481	4	0.025 %

5.1 Experimental Testbed

The experimental evaluation of approximate query answering in DWS is conducted in a workstation environment where all are linked together in an Ethernet network with Oracle 8 database management system installed in each of them.

The TPC-H was implemented with a scale factor of 1 for the test database, which corresponds, to approximately 1 GB for the database size. This corresponds to a big fact table LINEITEM (6,001,215 rows) and the dimension tables ORDERS (1,500,000 rows), CUSTOMER (150,000 rows), SUPPLIER (10,000 rows) and NATION (25 rows).

In these experiments we apply our technique to one workstation, simulating a centralized data warehouse and denote it as CDW (Centralized Data Warehouse), and to N=5, 10, 20, 50 and 100 workstations, which corresponds to DWS-5, DWS-10, DWS-20, DWS-50 and DWS-100, respectively.

The use of N workstations was simulated by dividing the *n_fact_rows* (6,001,215) of LINEITEM fact table, into N partial fact tables (LINEITEM_1,…, LINEITEM_N). Each

workstation has *n_fact_rows*/*N* rows and the dimensions are replicated in each workstation. For example, DWS-100 simulates the use of 100 workstations (*N*=100) having 100 partial fact tables (LINEITEM_1, … ,LINEITEM_100) with each one having 600,121±1 fact rows, while the dimensions are equivalent to those of the CDW system.

5.2 Approximate query answers

In these experiments we evaluated the error obtained with typical OLAP queries when some of the workstations are unavailable and proved that we can give very tight confidence intervals such that users know about the accuracy of the result. The influence of group-by queries with different sizes of groups will also be analyzed.

Estimation accuracy. The unavailability of each individual workstation was simulated by not taking into account the partial results that corresponded to the failed workstation. Finally, we compute the average and maximum of all these errors for each configuration. For example, using the DWS-100 configuration, we determine the error when one of the 100 workstations is unavailable and determine the average and maximum of these errors.

The average and maximum error obtained for Q1, Q6 and Q7 queries of TPC-H benchmark using DWS-5, DWS-10, DWS-20, DWS-50 and DWS-100 and considering only one unavailable workstation are illustrated in Figure 3, where the *x* axis represents the number of workstations used.

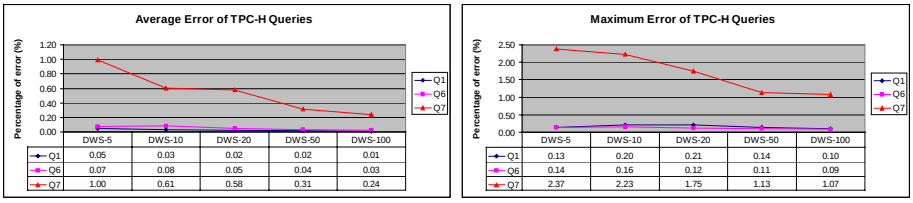


Fig. 3. Average and maximum error for Q1, Q6 and Q7 when one workstation is unavailable

As can be observed, the average error obtained for these queries is extremely small because we only simulate the unavailability of one of the workstations that comprise each configuration. Additionally, this error decreases when we use more workstations, due to the fact that the number of missing rows is smaller. The average error obtained for Q7 query is larger than the error corresponding to the other queries because the average number of rows aggregated by group is smaller. This is due to the fact the query Q7 has a higher selectivity (as shown in Table 1), meaning less elements in each aggregation group, which in case of failure of one workstation has more impact in the precision of the result obtained.

The maximum error is higher than average error because it is the worst-case. We compute the maximum error obtained for each query and for each group. However, as illustrated in the right picture of Figure 3, this error was always smaller than 2.5% even when 1/5 of workstations were unavailable.

In the results shown before only one workstation was unavailable, but we are also interested in studying the results when the number of unavailable workstations is much larger. For instance, in DWS-5 we can simulate the failure of 2, 3 or 4 workstations, which corresponds to an unavailability of 40%, 60% or 80%, respectively. The average and maximum error for all possible combinations is shown in Figure 4 for queries Q1, Q6 and Q7, where the *x* axis represents the number of workstations unavailable.

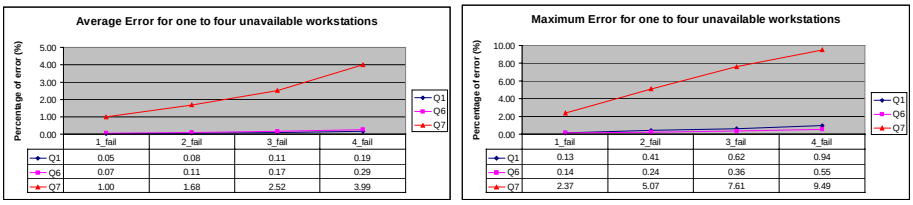


Fig. 4. Average and maximum error on DWS-5 when one to four workstations are unavailable

In these experiments, using DWS-5 configuration, the error increases with the number of workstations that are unavailable (as expected). However, this error is not very large in average, as it does not exceed 4% (Q7 query) or even less. Furthermore, the average error is not higher than 0.3% for queries Q1 and Q6, which is a very good approximation of the result for most of the cases. These very good results are due to the fact that our partitioning strategy is pseudo-random resulting in a uniform random distribution of the data over the workstations.

The maximum error obtained using all the possible combinations of workstation unavailability is about 10%. It must be pointed out this is the extreme case of unavailability because the user is only accessing his own workstation. Interestingly, the maximum error of Q1 query is higher than the maximum error of Q6 even though Q1 aggregates more rows in average than Q6 (see Table 1). However, this is due to the influence of a very small group in Q1. This group has 38,854 rows, which is a much smaller number of rows than those from query Q6 (see Table 1). Therefore, we could conclude that the precision of our results is highly influenced by the number of rows processed in each group of a group-by query. However, even in the case of unavailability of 80% of the workstations we obtain an error less than 10% in the worst case meaning that approximate results are not harmful.

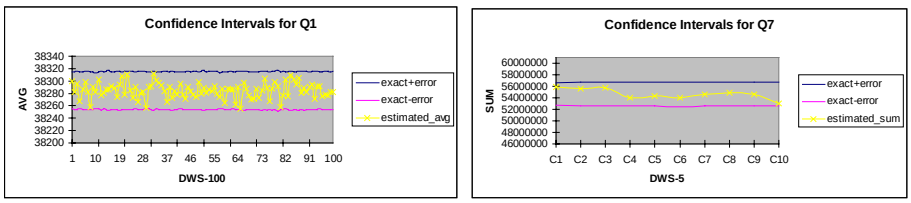
Confidence intervals. We provide accuracy measures as confidence intervals for each estimate, for some confidence probability.

The next figures analyze the confidence interval that is returned using our technique for queries Q1 and Q7 using various configurations of DWS. Each graphic shows three curves, two of them representing the sum of the exact value with the confidence interval (*exact_value+error* and *exact_value-error*), corresponding to the upper and lower curves in the figures. The middle curve is the *estimated_value* obtained with the respective configuration.

Figure 5(a) shows the confidence interval for query Q1 using DWS-100 and the aggregation *avg(l_extendedprice)* for one of the groups of the result. As we are simulating the unavailability of only one workstation, the *x* axis legend indicates which

workstation is unavailable and the *y* axis shows the value of the aggregation as well as the confidence interval bounds. This example shows that the confidence intervals do a good job determining boundaries for the error. These intervals are computed using the formula 3 of section 4.2, with a probability of 99%.

Figure 5(b) shows the confidence intervals for query Q7 and all possible combinations of unavailability of three workstations using DWS-5. The value computed by query Q7 is the aggregation *sum(volume)*. The *x* axis represents all possible combinations of unavailability of three workstations. The query Q7 returns four groups, but for simplicity is only shown the result of one in the figure 5(b). In this case confidence intervals are computed with a probability of 95%.



(a) Unavailability of one workstation

(b) Combinations of unavailability of three workstations

Fig. 5. Confidence interval for queries Q1 and Q7 using DWS-100 and DWS-5

The experimental results show that confidence intervals are very useful to deliver convenient error bounds for a given confidence level and the errors are very small. Thus, we can give the user very tight confidence intervals of the approximate answers when one or more workstations are unavailable. The artificial nature of the TPC-H benchmark data could influence the results but we argue that this highly-accurate answers are mainly due to our round robin data partitioning which provides randomness of our facts which would not be the case if we have used range partitioning.

6 Conclusions and Future Work

In this paper, we have proposed a distributed data warehousing strategy that is able to answer typical OLAP queries when component workstations are unavailable. Data Warehouse Striping (DWS) is a scalable technique that divides data warehouse facts into a number of workstations to solve data warehouse limitations related to heavy storage loads and performance problems. With the proposed modular approach, simple workstations without any special hardware or software fault tolerance can be used and very accurate approximate answers are returned even when a substantial number of the component workstations are unavailable. We have proposed a formula to quantify estimation error of the answer and proved that this error is very small when the fraction of workstations that are unavailable is reasonably small or the number of values aggregated into the groups is sufficiently large.

The proposed technique is a cost-effective solution that could be applied in almost all types of organizations, taking advantage of the availability of computer networks to

distribute the data and their processing power, avoiding the need of very expensive servers. The experimental results show a linear or even super linear speedup of DWS, due to the fact that, when we distribute the data, we are working with more manageable amounts of data that do not stress memory and computing resources so much.

The experimental results of this paper have also shown that the DWS technique provides approximate query answers with very small errors, even when most of the workstations are unavailable. The confidence intervals are promising, as the technique is able to return strict confidence intervals with important information to the user concerning the amount of error of the estimations. We propose a more complex statistical analysis as future work.

References

1. Acharaya, S., Gibbons, P.., Poosala, V.: Congressional Samples for Approximate Answering of Group-By Queries. ACM SIGMOD Int. Conf on Management of Data, (2000) 487-498
2. Albrecht, J., Gunzel, H., Lehner, W.: An Architecture for Distributed OLAP. Int. Conference on Parallel and Distributed Processing Techniques and Applications PDPTA, (1998)
3. APB-1 Benchmark, Olap Council, November 1998, www.olpacouncil.org
4. Barbara, D., et al.: The New Jersey data reduction report. Bulletin of the Technical Committee on Data Engineering, 20(4) (1997) 3-45
5. Bernardino, J., Madeira, H.: A New Technique to Speedup Queries in Data Warehousing. In Proc. of Chalenges ADBIS-DASFAA, Prague (2000) 21-32
6. Chauduri, S., Dayal, U.: An overview of data warehousing and OLAP technology. SIGMOD Record, 26(1), (1997) 65-74
7. Cochran, William G.: Sampling Techniques, 3rd edn, John Wiley & Sons, New York, 1977.
8. Codd, E.F., Codd, S.B., Salley, C.T.: Providing OLAP (on-line analitycal processing) to user- analysts: An IT mandate. Technical report, E.F. Codd & Associates (1993)
9. Gibbons, P. B., Matias Y.: New sampling-based summary statistics for improving approximate query answers. ACM SIGMOD Int. Conf. on Management of Data (1998) 331–342
10. Haas, P. J.: Large-sample and deterministic confidence intervals for online aggregation. In Proc. 9th Int. Conference on Scientific and Statistical Database Management (1997) 51-62
11. Hellerstein, J.M., Haas, P.J., Wang, H.J.: Online aggregation. ACM SIGMOD Int. Conference on Management of Data (1997) 171-182
12. Kimball, Ralph: The Data Warehouse Toolkit. Ed. J. Wiley & Sons, Inc (1996)
13. Kimball, Ralph, Reeves, L., Ross, M., Thornthwalte, W.: The Data Warehouse Lifecycle Toolkit. Ed. J. Wiley & Sons, Inc (1998)
14. Selinger, P., et al.: Access Path Selection in a Relational Database Management System. ACM SIGMOD Int. Conf. on Management of Data (1979) 23-34
15. TPC Benchmark H, Transaction Processing Council, June 1999, www.tpc.org
16. Vitter, J., Wang, M.: Approximate computation of multidimensional aggregates of sparse data using wavelets. ACM SIGMOD Int. Conf. on Management of Data (1999) 193-204