A Non-Linear Cost Model for Multi-Node OLAP Systems

Jorge Loureiro[1] Orlando Belo[2]

[1] Departamento de Informática, Escola Superior de Tecnologia, Instituto Politécnico de Viseu, Campus Politécnico de Repeses, 3505-510 Viseu, PORTUGAL
jloureiro@di.estv.ipv.pt

[2] Departamento de Informática, Escola de Engenharia, Universidade do Minho, Campus de Gualtar, 4710-057 Braga, PORTUGAL
obelo@di.uminho.pt

Abstract. Answering performance to business queries, mainly of aggregated nature, known as On-Line Analytical Processing queries, depends heavily on the proper selection of multidimensional structures, known as materialized subcubes or views. As user's queries profiles change, these structures have to be recalibrated, once elected the new appropriated selection through a cube view selection algorithm. In the very core of all these algorithms is the estimation of the cost of answering queries and maintenance, given M materialized OLAP selected structures that may be of a distributed nature. This paper introduces a new extended cost model that supports both centralized or multi-node OLAP architecture, query and maintenance cost of non-linear nature and incremental or integral (from scratch) maintenance.

1 Introduction

The aggregated nature of almost all *On-Line Analytical Processing* (OLAP) queries implies that pre-computing and materializing of aggregated queries answers, denoted as materialized views or subcubes [2], is a *sine qua non* performance condition in OLAP systems. In fact, when a user poses a query, if a suitable aggregated view is available, its use may decrease enormously the time to obtain an answer. The reading of a few records may be sufficient, avoiding a total scan and ulterior aggregation of a possibly immense fact table. But, as the number of aggregated views is huge, its complete materialization would be very expensive, as it requires storage space and especially time to refresh all subcubes, when the base relations were updated. Then, among all possible subcubes we have to choose the ones (M-materialized set) that reveal themselves as the most valuable, attending to the queries' profile: that's the well known cube view selection problem, which is characteristically NP-hard. To know the cost of a given set of queries and M, we need a way of simulate an OLAP system, using a model, that includes all contributing elements to cost estimation and corresponding computing equations. The complexity of the model is related with the features that are to be included in the simulation system. Here, we are interested on the inclusion of a broad set of characteristics able to support: 1) a distributed OLAP architecture, here denoted Multi Node OLAP (M-OLAP) architecture, where *n* OLAP servers share data, being spatially close or disperse; 2) non-linearities, as the existence of indexes or sorts may disturb the direct relation between the number of tuples scanned and the cost of query answering or generating a subcube (extending the linear model introduced in [3]); 3) the possibility to estimate both query and maintenance cost; 4) the incremental or integral (from scratch) maintenance process, selecting the maintenance type to perform at subcube granularity level.

2 Distributed non-Linear Generalized Cost Model

Figure 1 shows the proposed M-OLAP architecture and the distributed non-linear generalized cost model. Summarizing, we have, not only the intra-node dependencies, but also inter-node dependencies, as the communication links allows to compute a subcube using others in different nodes [1]. As we can see, the model has several weights. In each vertex of the *lattice* we have: *S*, the subcube's size, a value for the scan cost; f_q and f_u, query and update frequencies; query utility extent, *eq* (the mean query usage extent of any subcube, caused by OLAP restriction clauses that limit the range of scanning and aggregation operations), and *eu*, the update extent, no more than the impact of the changes in the base relations in each subcube (the ratio of the number of updated cells in the refreshed subcube and its size). Each edge has two weights: wq_{ij}, the query processing cost of subcube s_i from s_j and wu_{ij}, the update processing cost of subcube s_i from s_j. Query cost estimation of Cq_{ij} (query cost of subcube *i* using *j*) is the cost *S* of subcube j, added by *wq*'s corresponding to the arrows of the shortest path between j and i. In physical terms, one may consider *S* as the scan cost (the one incurred when *i*=*j*), whose further aggregations imply an extra cost (*wq*'s), which may be different depending on the cube *i* and the sorts or indexes in *j*. Concerning to incremental maintenance cost, it may be computed simply summing the *wu*'s that are now supposed to be the cost of preparing a delta [7] using *j* and integrating it into *i*. This model may also be used in the case of an integral maintenance (generation of subcubes from scratch), using query cost equation added of integration costs. Concerning to communication costs, two scenarios might be posed: distributed OLAP in the same physical space or at disperse geographic points. Intuitively, the former is simpler and an instance of the last. In this paper we're going to consider the simpler case. Then, the communication cost of transmitting data between nodes *i* and *j* can be assumed to be linear, as follows: $CC_{ij} = Np * Sp / BD + La$, where *BD* is binary debit, *La* is the latency and *Np* is the number of data packets to transfer. The processing and communication costs discussed above have different referential units: records and seconds, respectively. Time will be our cost unit, as it allows the estimation of the answering time that settles users' satisfaction and productivity and the maintenance constraint is also specified in time (length of the refreshing window). Its conversion is straightforward: the processing power of each node where the scan (and aggregation) or integration operations occur may be used: this way we may convert records into time spend to process them. Concerning to the costs of transmitting a subcube, in fact, each record is actually a cell, usually a value (e.g. total sales), that may have a size (in bytes) of $8<sb<16$ which allows to compute the number of bits to transmit.

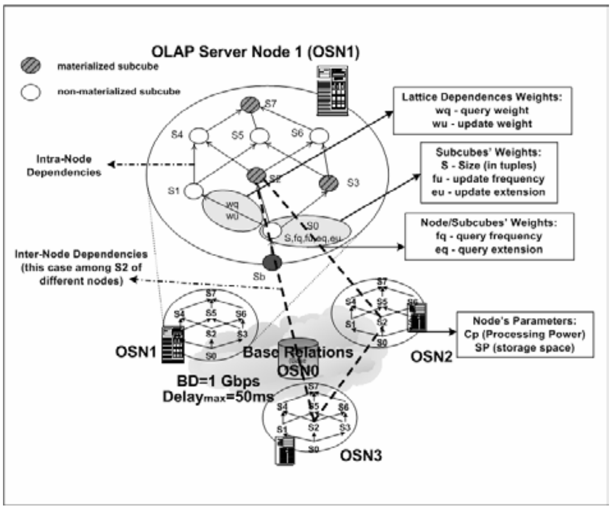

Figure 1. Multi-Node architecture and generalized cost model.

Then, the final equation that allow to compute the query cost is:

$$Cq(Q,M)=\sum_{q_i\in Q}(\min((|Anc(q_i,M)|+\min\sum_{v\in(Anc(q_i)\to q_i)} wq_v)/Cp(Node_{Anc_{q_i}})+CC_{Node(Anc_{q_i}\to q_i)} * eq_{q_i})*fq_{q_i} \quad \text{(eq.1).}$$

Concerning to incremental maintenance, we have to split *wu*'s costs, as scanning and aggregating may occur in one node and integration in a different one. W_u corresponds now only to scan and aggregation operations costs; integration costs (corresponding to delta size) are $Si*eu_{si}$. Then:

$$Cm(M)=\sum_{S_i\in M}\min(\sum_{v\in(Anc(s_i)\to s_i)} wu_v/Cp(No_{AncSi})+CC_{Node(Anc_{q_i}\to q_i)})+S_i*eu_{s_i}/Cp(No_{S_i})*fu_{si} \quad \text{(eq.2),}$$

Finally, if the subcube is to be computed from scratch (integral maintenance):

$$Cm(M)=\sum_{S_i\in M}(\min((|Anc(S_i,M)|+\min\sum_{v\in(Anc(s_i)\to s_i)} wq_v)/Cp(No_{AncSi})+CC_{Node(Anc_{Si}\to S_i)})+S_i/Cp(No_{S_i}))*fu_{S_i} \quad \text{(eq.3).}$$

One example of query and maintenance cost estimation using the proposed distributed non-linear generalized cost model would be elucidative and may be found in [5], although for the dispersed geographic M-OLAP.

3 Conclusions and Future Work

This paper introduces a distributed non-linear generalized cost model that addresses the estimation of query and maintenance cost of a M-OLAP architecture. This architecture extends the centralized OLAP structures to real distributed multidimensional structures using a number of OLAP server nodes (potentially high) interconnected by a heterogeneous communication network, capturing the known advantages of distributed databases. This model supports a broad set of characteristics, namely: 1) an heterogeneous multi-node OLAP support with effective OLAP cube distribution; 2) an explicit inclusion of node processing power and communication network's parameters; 3) the inclusion of non-linearities on the model; 4) the incremental or integral maintenance defined at subcube level; and 5) a real effective maintenance cost computing in time units, immediately allowing the feasibility of a given cube distribution. Given its generality, this model may be used as the base framework to develop suitable cost computing algorithms, with corresponding broad OLAP systems cases appliance. The M-OLAP architecture allows the parallel processing of many tasks. This feature may be considered in the design of cost estimation algorithms. The control of parallelization, mainly the conflict resolution, implies further complexity, but the estimated values will be probably more accurate. Using a super-pipeline scheme to simulate the parallel tasks processing, we developed query and maintenance algorithms [5], where the parallelization of maintenance cost tasks allows a refreshing process that may be viewed as a wave effect. With those algorithms in hands we plan, in the near future, to design and develop distributed cube selection algorithms. Some work already done with greedy, genetic [4] and discrete particle swarm algorithms [6], although using a linear cost model, was encouraging and opens good perspectives to this future work.

Acknowledgements

The work of Jorge Loureiro was supported by a grant from PRODEP III, acção 5.3 – Formação Avançada no Ensino Superior, Concurso N.º 02/PRODEP/2003.

References

1. Bauer, A. and Lehner, W.: On Solving the View Selection Problem in Distributed Data Warehouse Architectures. In: Proceedings of the 15th International Conference on Scientific and Statistical Database Management (SSDBM'03), IEEE (2003) 43-51.
2. Gupta, A., and Mumick, I.S.: Materialized Views: Techniques, Implementations and Applications. The MIT Press (1999).
3. Harinarayan, V., Rajaraman, A., and Ullman, J.: Implementing Data Cubes Efficiently. In: Proc. of ACM SIGMOD, Montreal, Canada, (1996), 205-216.
4. Loureiro, J., and Belo, O.: Selecting and Allocating Distributed OLAP Cubes: An Evolutionary Approach. *Technical Report*. Department of Informatics, University of Minho. 2006.
5. Loureiro, J., and Belo, O.: Cost Estimation Algorithms with Parallel Task Execution Simulation for Multi-Node OLAP Cubes with Parallel Task Execution Simulation. *Technical Report*. Department of Informatics, University of Minho. 2006.
6. Loureiro, J., and Belo, O.: A Discrete Particle Swarm Algorithm for OLAP Data Cube Selection. To appear in: Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS 2006), Paphos – Cyprus, 23-27 May (2006).
7. Mumick, I., Quass, D., and Mumick, B.: Maintenance of Data Cubes and Summary Tables in a Warehouse. In: Peckham J (ed): Proceedings of ACM SIGMOD International Conference of Management of Data, Tucson, Arizona (June 1997) 100-111.